

**High End Computing Revitalization Task Force  
(HECRTF), Inter Agency Working Group (HECIWG)  
File Systems and I/O Research Workshop HECIWG  
FSIO 2006**

**Rob Ross DOE/Office of Science ANL  
Evan Felix DOE/Office of Science PNL  
Bill Loewe DOE/NNSA LLNL  
Lee Ward DOE/NNSA SNL  
James Nunez DOE/NNSA LANL  
John Bent DOE/NNSA LANL  
Ellen Salmon NASA  
Gary Grider DOE/NNSA LANL**

Executive Summary .....	2
Compelling Case Information and Background .....	5
The Eight Areas of Needed R&D .....	10
Background .....	11
Frequently Used Terms .....	13
Research Themes Identified From the Workshops .....	14
Metadata .....	14
Measurement and Understanding .....	20
Quality of Service .....	21
Security .....	22
Next-Generation I/O Architectures .....	24
Communications and Protocols .....	33
Management and RAS .....	34
Archive .....	35
Assisting Standards .....	38
Assisting Research and Education .....	38
Data Availability .....	39
Education, Community, and Center Support .....	41
Availability of Computational Resources .....	43
Conclusion .....	45
References .....	49
APPENDIX A (HECURA and SciDAC2 FSIO projects) .....	52
APPENDIX B (HECIWG FSIO 2006 attendees) .....	68
APPENDIX C (Inter-Agency HPC FSIO R&D Needs Document) .....	73

## Executive Summary

The need for immense and rapidly increasing scale in scientific computation drives the need for rapidly increasing scale in storage capability for scientific processing. Individual storage devices are rapidly getting denser while their bandwidth is not growing at the same pace. In the past several years, initial research into highly scalable file systems, high level Input/Output (I/O) libraries, and I/O middleware was conducted to provide some solutions to the problems that arise from massively parallel storage. To help plan for the research needs in the area of File Systems and I/O, the inter-government-agency document titled “HPC File Systems and Scalable I/O: Suggested Research and Development Topics for the Fiscal 2005-2009 Time Frame” [Appendix C] was published which led the High End Computing Interagency Working Group (HECIWG) to designate this area as a national focus area starting in FY06. To collect a broader set of research needs in this area, the first HECIWG File Systems and I/O (FSIO) workshop was held in August 2005 in Grapevine, TX. Government agencies, top universities in the I/O area, and commercial entities that fund file systems and I/O research were invited to help the HECIWG determine the most needed research topics within this area. The HECIWG FSIO 2005 workshop report can be found at <ftp://ftp.lanl.gov/public/ggrider/HEC-IWG-FS-IO-Workshop-08-15-2005/HECIWG-FSIO-Workshop-Document-FY05.pdf>. All presentation materials from this workshop can be found at <http://www.nitrd.gov/subcommittee/hec/workshop/>

The workshop attendees helped

- catalog existing government funded and other relevant research in this area,
- list top research areas that need to be addressed in the coming years,
- determine where gaps and overlaps exist, and
- recommend the most pressing future short and long term research areas and needs necessary to help advise the HECIWG to ensure a well coordinated set of government funded research

The recommended research topics are organized around these themes: metadata, measurement and understanding, quality of service, security, next-generation I/O architectures, communication and protocols, archive, and management and RAS. Additionally, university I/O center support in the forms of computing and simulation equipment availability, and availability of operational data to enable research, and HEC involvement in the educational process were called out as areas needing assistance.

With the information from the Multi-agency I/O Document [Appendix C] and the HECIWG FSIO 2005 workshop, a number of activities occurred during 2006:

- In the area of R&D
  - a National Science Foundation (NSF) HEC University Research Activity (HECURA) solicitation for university research in the FSIO area was written based on the eight areas of research identified during the HECIWG FSIO 2005 workshop
  - the NSF/HECURA solicitation was conducted resulting in 62 proposals from over 80 universities

- from a careful analysis of the proposals, 19 HECURA awards were made
- the DOE Office of Science awarded two SciDAC2 FSIO projects
- In the area of providing computational resources
  - the DOE Office of Science INCITE program for supplying computing clusters
  - NSF infrastructure program for providing computing infrastructure
- In the area of providing operational data to enable research
  - LANL release of failure, event, and usage data
  - PSC, Library of Congress, and HP working on data release
  - Consortia for failure data release is forming up.

Additionally, due to the success of the HECIWG FSIO 2005 workshop and subsequent activities, a permanent HECIWG FSIO advisory group was formed to help continue to advise the HECIWG in how best to coordinate FSIO activity.

The HECIWG FSIO advisory group held the HECIWG FSIO 2006 workshop in August 20-22 in Washington DC. The location was picked to encourage more HECIWG agency participation, and indeed three more HECIWG agencies were represented. The workshop was again, attended by top university, government HEC, and industry FSIO R&D professionals. The goals for this workshop were:

- To update everyone on the
  - 19 HECURA and two SciDAC2 FSIO research activities
  - programs available to get computing resources
  - activities to make HEC site operational data available to enable research.
- To solicit input on
  - remaining gaps in the needed research areas
  - gaps in providing center support such as providing
    - computational resources
    - operational data available for research
    - getting the HEC community involved in the educational process.

The information gathered at this and the previous workshop will be used to advise the HECIWG on how to facilitate better coordinated government funded R&D in this important area in the coming years.

The results of the HECIWG FSIO 2006 are presented in this document. From the 70 original attendees of the 2005 workshop, the attendance grew to 100 in 2006. The workshop was well received and accomplished its goals. Attendees were quite happy about the HECURA and SciDAC2 activities and were extremely interested in the other support activities, especially the availability of operational data and the HEC involvement in the educational process.

Analysis of the 2006 workshop identified gaps by the HECIWG is summarized below:

### **Research Gaps**

The most important research gaps that remain are:

- Revolutionary approaches to metadata for scaling, searching, extensibility, and data lifetime management
- Automated self tuning, management, configuration, etc. (good research starting but very difficult topic)
- Next generation virtual file system (VFS) interface for operating systems
- Scaling to multi-million way parallelism for small I/O operation workloads
- Understanding complex system workloads

### **Research Enabling and Infrastructure Gaps**

The most important research enabling and infrastructure gaps that remain are:

- Computing site failure, usage, event, and performance data, including recommended standard formats, data to collect, and tools for collecting and using these data.
- Availability of simulation and virtualization tools, for simulation of disks, disk arrays, storage area networks, and even file systems
- Midrange testbed for I/O research, something larger than a small university cluster but smaller than a large INCITE or NSF supercomputer center, where reconfiguration could be done for computer science research, would be of value as well for enabling research.
- Benchmark clearinghouse

### **Education Gaps**

The most important education gaps that remain are:

- K-12 and undergraduate program including tours of HEC sites
- HEC sites and industry provided lectures in class rooms
- Harness the gaming industry excitement in curricula
- More HEC site and industry internships
- Promote recurring NSF research funding in I/O and Storage
- Promote recurring NSF education funding in I/O and Storage
- Promote I/O and Storage grand challenge and top 10 problems

### **Standardization Gaps**

The most important standardization gaps that remain are:

- Information sharing between file systems and applications
- Archive interfaces

### **Productization Gaps**

The most important productization gaps are:

- Exploitation of hybrid storage devices (disk and non volatile memory)
- Exploitation of active storage technology

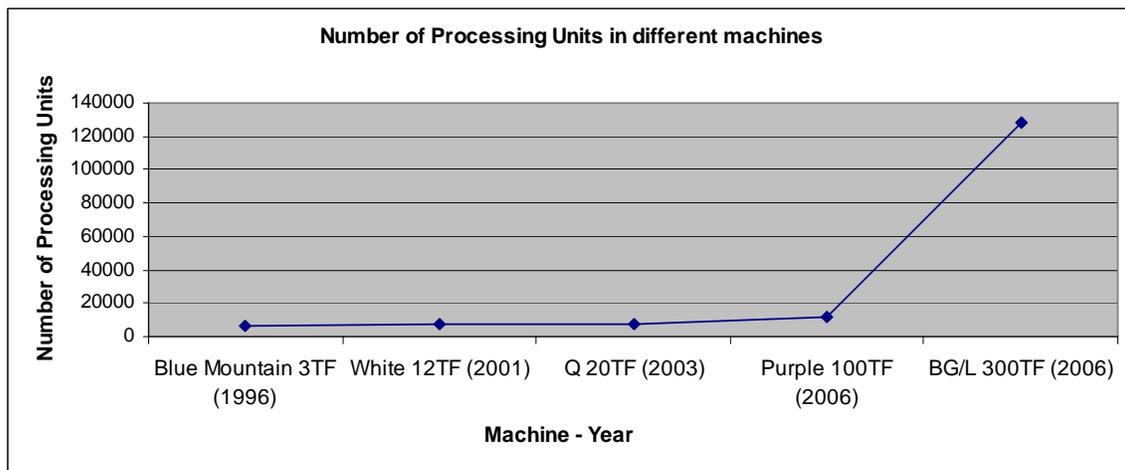
- Testing of storage stripe widths beyond 10,000 and with million way client parallelism
- Commercial parallel archiving products
- Long term security key management and electronic signing and tracking of data

The workshop presentation materials can be found at <http://www.nitrd.gov/subcommittee/hec/workshop/> . It was obvious from opinions gathered that the work conducted by the HECIWG FSIO group is sorely needed and that the job of the group is far from done. In summary, the HECIWG workshop activity has served to solidify the FSIO community and is serving the HEC community and the entire FSIO community well.

## Compelling Case Information and Background

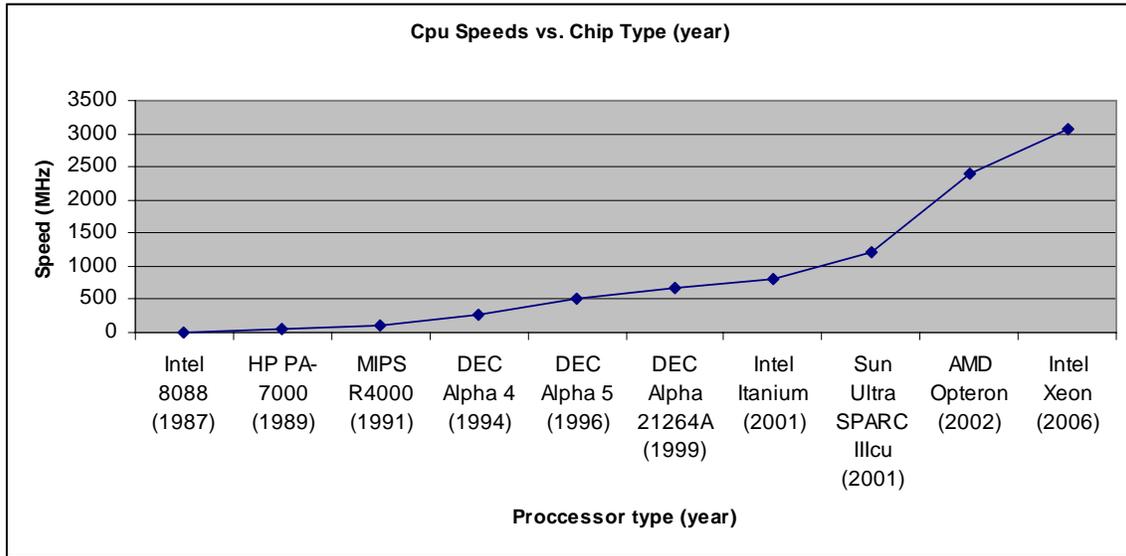
Historical and projected future trends can be used to explain why I/O and file system research is needed and which particular areas of research need to be pursued.

Although processor clock speeds have grown drastically over the last two decades, it appears that the rapid increases in processor clock rates are slowing. The microprocessor industry is exploring and even beginning to deploy processor architectures that have many more processing units per chip or board to continue to meet the processing power growth demand. This implies that scientific applications will have to begin to rely more heavily on multi-process/task parallelism at a greater scale than ever before. When single processors were getting much faster each year, applications could gain advantage over time by keeping a constant number of processes per task but this appears to be no longer true. In order for applications to continue to gain speed up, it will now require the use of more processing elements over time. The compute capabilities of machines anticipated for scientific computing is growing rapidly. The following graph illustrates the corresponding growth in the number of processors we have and expect to see used to build these large scientific computers.

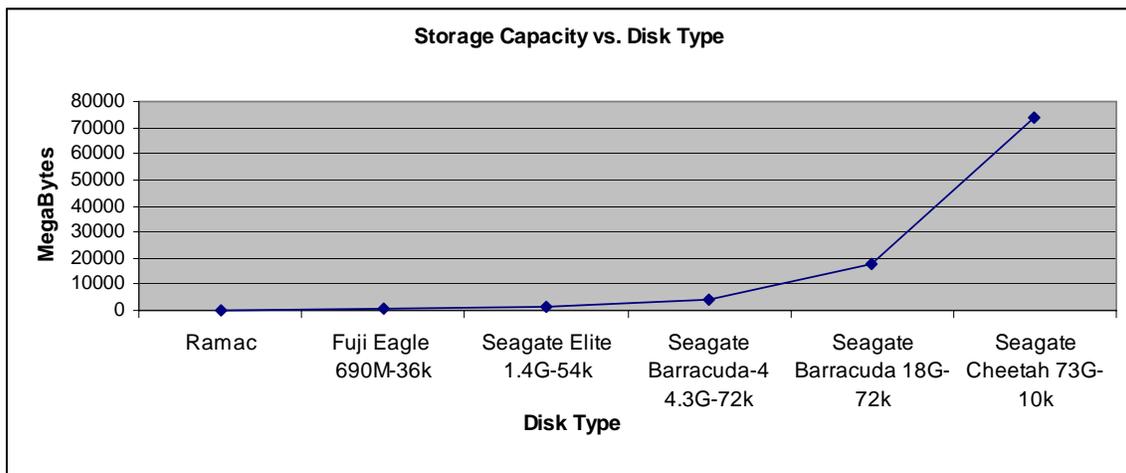


Additionally, it is important to note that the amount of memory per processor has gone down over the last few years from 2-4 GB/processor on most ASC machines to .25 GB/processor on BG/L. Memory per Teraflop (TF) has gone down from around 1 Terabyte (TB) per TF to about .1 TB per TF.

In the past 50 years, the performance of individual processors has gone up by 4-5 orders of magnitude.

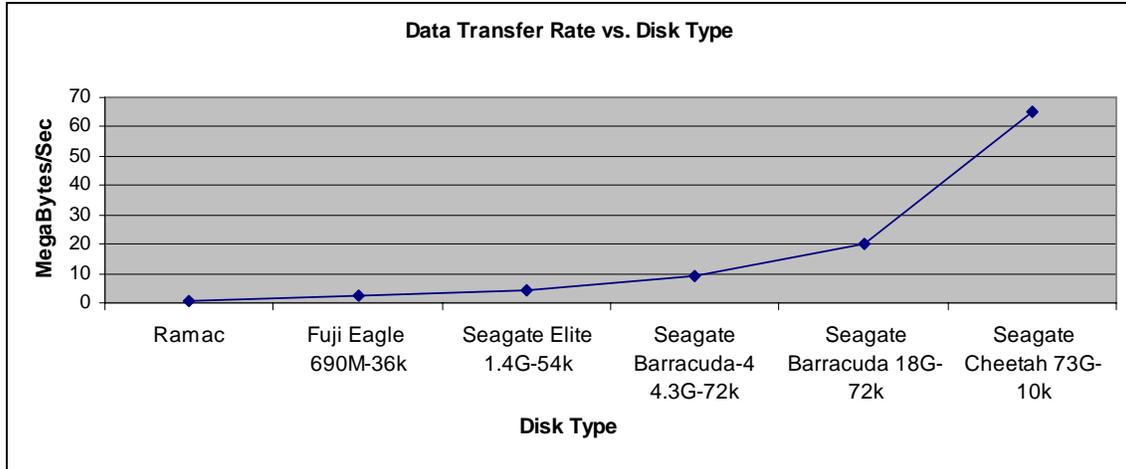


We've also seen that disk drives have become far more dense over the past 5 decades. The following graph shows the relative density of the original disk drive, the IBM RAMAC 1956 versus a recent state of the art Seagate 15k RPM 2.5 inch 73 gigabyte disk drive.

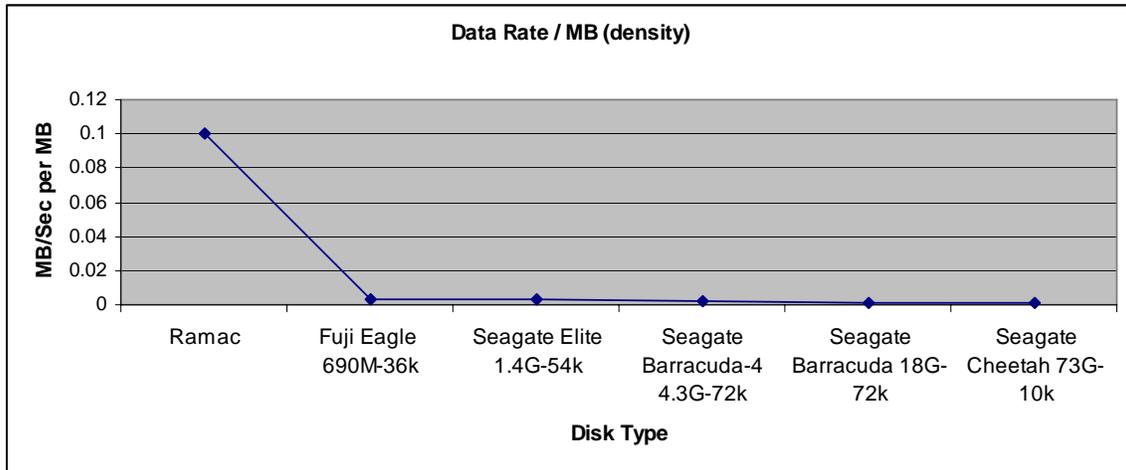


As you can see, disk capacity has grown at an amazing rate: five orders of magnitude in 50 years; approximately the same change that CPU speed has gone through.

In the following graph, the disk drive data transfer rate speed up over the past 50 years is shown.

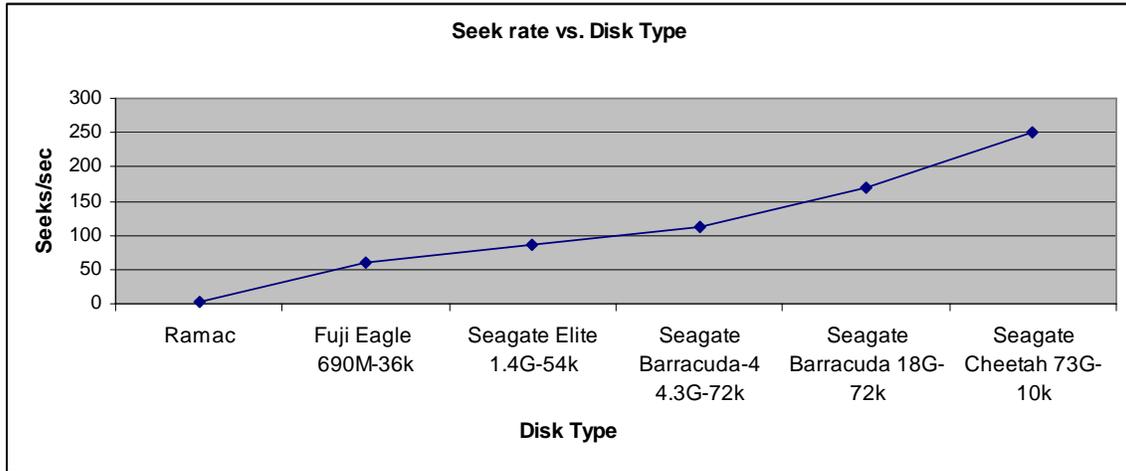


As you can see, transfer rate has gone up a lot as well, but only about two orders of magnitude. This is an important observation. Another way to look at this phenomenon is in the graph below which shows disk transfer rates normalized to byte of storage capacity (which would be roughly equivalent to normalizing to processing power as well given the similarities between the growth of disk capacity and CPU speed).

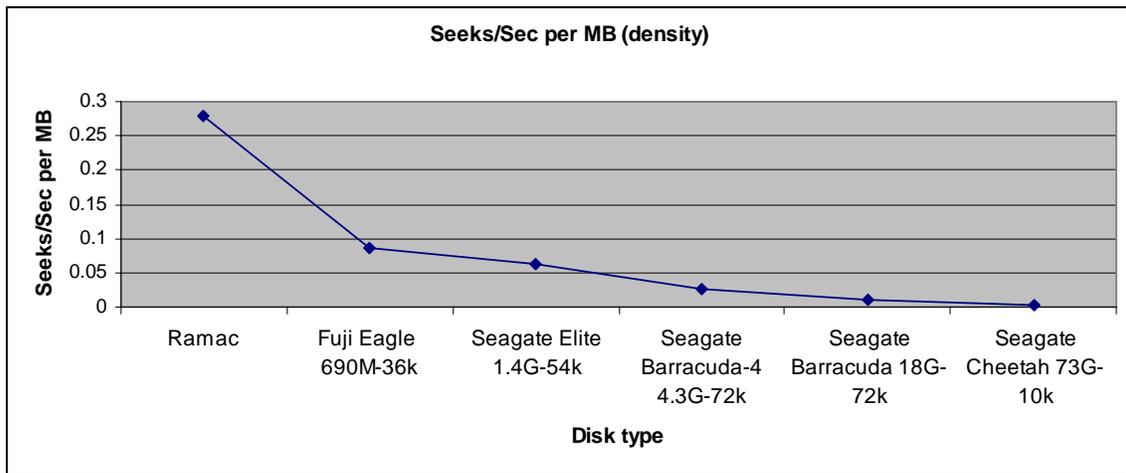


As you can see, the data rate performance of disk drives has gone down by about two orders of magnitude. **This means that it takes two orders of magnitude more disk drives per CPU to do the same relative workload in a balanced way than 50 years ago!**

Disk agility has also gotten much better over the last 50 years. The following graph shows the seek performance increase achieved.



Again, the seek performance has gone up by about two orders of magnitude. This is also an important observation. Another way to look at this phenomenon is in the graph below which shows disk seek rates normalized to byte of storage capacity (which would be roughly equivalent to normalizing to processing power as well given the similarities between the growth of disk capacity and CPU speed).



As you can see, the seek rate performance of disk drives has gone down by about two orders of magnitude. **This means that it takes two orders of magnitude more disk drives per CPU to do the same relative workload in a balanced way than 50 years ago!**

All of this information has some interesting effects on the I/O and File Systems services:

- The number of disk drives needed to build a balanced system over time has gone up drastically. The largest sites today currently have in the 20,000 disk drive range. It seems quite likely that we will see sites with well over 100,000 disk drives soon and perhaps over a million when machines reach multiple petaflops.

- Since the number of processing elements is going up rapidly and the amount of memory per processing element is doing down, the I/O system must orchestrate collecting memory from far more memories to be sent to far more disk drives, due to the slow growth in disk bandwidth.
- As the numbers of processing elements goes up, the required file system metadata operations per second goes up, which again implies orchestrating metadata requests from more clients to more disks than ever due to the slow growth in disk agility.

The areas identified at the HECIWG FSIO 2005 workshop were: metadata, measurement and understanding, quality of service (QoS), security, next-generation I/O architectures, communication and protocols, archive, and management and RAS. Using the above background, the following is an examination of these eight areas.

- Metadata – Metadata operations will involve orchestration of more clients to more disks.
- Measurement and understanding – Measuring and understanding performance as more and more disks and clients are involved becomes more difficult.
- Quality of Service – mixed workloads in an environment with hundreds of thousands of processors will make providing determinism extremely difficult.
- Security – Security is an age old issue, scale is one area that makes security difficult, but there are many others, especially need to know issues. Security for persistent data is difficult even without scale.
- Next-generation I/O architectures – Next generation I/O architectures will be needed to orchestrate movement of data and metadata from so many processing elements and so many disks.
- Communication and protocols – Communications and protocols will need to be used in these immensely scaled up environments. Protocols can be used to help in the huge orchestration effort.
- Archive – Archiving integrated to file systems and I/O stacks at these scales is a difficult task.
- Management and RAS – Management and RAS in an environment with 100,000 or more disks and million way parallelism will be extremely difficult. Adding to the difficulty of management is the added difficulty of having to deal with long term persistent data. Failure of processors typically means a job is re-run. Failure of a disk may mean loss of valuable data or information.

Two major questions sum up most of the I/O and File System's plight.

1) How do you manage 100,000 mechanical disk drive devices and their associated environment, both hardware and software? This includes, RAS, QoS for usage that varies by seven orders of magnitude, management without requiring an army of administrators, security, etc.

2) How do you productively use 100,000 mechanical disk drive devices and their associated environment? This includes middleware, high level libraries, file systems, dealing with massive in-flight data, etc.

It is no surprise that the above trend information describes well most of the existing and new I/O and file systems issues identified at the workshop.

Despite these troubling trends in processor and storage, an examination of the quantity of computer science research in I/O and file systems compared to other areas of the HEC environment reveals that the I/O and file systems area has been greatly neglected. This neglect helps explain why investments in research in this area are needed so acutely and why the HECURA and SciDAC2 I/O projects were so well received by the HECIWG FSIO 2006 workshop attendees. The area of I/O has really been largely overlooked both in the area of how to manage enormous scale I/O systems and how to productively use such systems.

## **The Eight Areas of Needed R&D**

The HECIWG FSIO 2005 workshop recommended investment in both evolutionary and revolutionary research in eight areas.

- Metadata - Investigation into metadata issues is needed, especially in the areas of scalability, extensibility, access control, reliability, availability, and longevity for both file and archival systems. Additionally, consideration for very revolutionary ideas such as new approaches to name spaces and use of novel storage devices needs to be explored.
- Measurement and understanding - Research into measurement and understanding of end-to-end I/O performance is needed including evolutionary ideas such as layered performance measurement, benchmarking, tracing, and visualization of I/O related performance data. Also, more radical ideas like end-to-end modeling and simulation of I/O stacks and the use of virtual machines for large scale I/O simulation need to be explored.
- Quality of Service - QoS is a ripe topic for research especially in the area of providing prioritized, deterministic performance in the face of multiple, complex, parallel applications running concurrently with other non-parallel workloads. More revolutionary ideas such as dynamically adaptive end-to-end QoS throughout the hardware and software I/O stack are equally important.
- Security - Aspects of security such as usability, long term key management, distributed authentication, and dealing with security overhead are all good topics for research. There is also room for more difficult research topics such as novel new approaches to file system security including novel encryption end-to-end or otherwise that can be managed easily over time. The need for standardization of access control list mechanisms is also needed and investigation into a standard API for end-to-end encryption could be useful.
- Next-generation I/O architectures - There is great need for research into next-generation I/O architectures, including evolutionary concepts such as extending the POSIX I/O API standard to support archives in a more natural way, access awareness, and HEC/high concurrence. Studies into methods to deal with small, unaligned I/O

and mixed-size I/O workloads as well as collaborative caching and impedance matching are also needed. Novel approaches to I/O and File Systems also need to be explored including redistribution of intelligence, adaptive and reconfigurable I/O stacks, user space file systems, data-aware file systems, and the use of novel storage devices.

- Communications and protocols - In the area of file system related communications and protocols, evolutionary items such as exploitation of Remote Direct Memory Access (RDMA), Object Based Secure Disk (OBSD) extensions, Network File System Version 4 (NFSv4) extensions, and parallel Network File System (pNFS) proof-of-concept implementations as well as more revolutionary exploration of server to server communications are needed.
- Archive – In the area of archive, the interfaces to the file systems and I/O stacks in HEC systems and long term care for the massive scale of an archive in the HEC environment are difficult areas needing more research than they have received before.
- Management and RAS - In the area of management, reliability and availability at scale, management scaling, continuous versioning, and power management are all needed research topics. Additionally, more revolutionary ideas like autonomies, use of virtual machines, and novel devices exploitation need to be explored.

In addition to the eight research areas, more focused and complete government research investment needs to be made in the file systems and I/O middleware area of HEC, given its importance and its lack of sufficient funding levels in the past, as compared to other elements of HEC.

Scalable I/O is perhaps the most overlooked area of HEC R&D, and given the information generating and processing capabilities being installed and contemplated. It is a mistake to continue to neglect this area of HEC. One of the primary purposes of this document is to present the areas in need of new and continued investment in R&D and standardization in this crucial area of HPC file systems and scalable I/O which should be pursued by the government and that were identified at the HEC-IWG File Systems and I/O Research Workshop.

## Background

To help plan for the research needs in the area of File Systems and I/O, the inter-government-agency document titled “HPC File Systems and Scalable I/O: Suggested Research and Development Topics for the Fiscal 2005-2009 Time Frame” [Felix] was published which led the High End Computing Interagency Working Group to designate this area as a national focus area starting in FY06. To collect a broader set of research needs in this area, the first HECIWG File Systems and I/O (FSIO) workshop was held in August 2005 in Grapevine TX. Government agencies, top universities in the I/O area, and commercial entities that fund file systems and I/O research were invited to help the HECIWG determine the most needed research topics within this area. The HECIWG FSIO 2005 workshop report can be found at <ftp://ftp.lanl.gov/public/ggrider/HEC-IWG-FS-IO-Workshop-08-15-2005/HECIWG-FSIO-Workshop-Document-FY05.pdf> . All presentation materials from this workshop can be found at <http://www.nitrd.gov/subcommittee/hec/workshop/>

The workshop attendees helped

- catalog existing government funded and other relevant research in this area,
- list top research areas that need to be addressed in the coming years,
- determine where gaps and overlaps exist, and
- recommend the most pressing future short and long term research areas and needs necessary to ensure a well coordinated set of government funded research

The recommended research topics are organized around these themes: metadata, measurement and understanding, Quality of Service, security, next-generation I/O architectures, communication and protocols, archive, and management and RAS. Additionally, university I/O center support in the forms of computing and simulation equipment availability, availability of operational data to enable research, and HEC involvement in the educational process were called out as areas needing assistance.

With the information from the HECIWG document mentioned above and the HECIWG FSIO 2005 workshop, a number of activities occurred during the past year:

- In the area of R&D
  - a National Science Foundation (NSF) HEC University Research Activity (HECURA) solicitation for university research in the FSIO area was written based on the eight areas of research identified during the HECIWG FSIO 2005 workshop
  - the NSF/HECURA solicitation was conducted resulting in 62 proposals from over 80 universities
  - from a careful analysis of the proposals, 19 HECURA awards were made
  - the DOE Office of Science awarded two SciDAC2 FSIO projects
- In the area of providing computational resources
  - the DOE Office of Science INCITE program for supplying computing
  - NSF infrastructure program for providing computing infrastructure
- In the area of providing operational data to enable research
  - LANL release of failure, event, and usage data
  - PSC, Library of Congress, and HP working on data release
  - Consortia for failure data release is forming up.

Additionally, due to the success of the HECIWG FSIO 2005 workshop and subsequent activities, a semi permanent HECIWG FSIO advisory group was formed to help continue the advise the HECIWG on how best to coordination FSIO activity.

The HECIWG FSIO advisory group recently held the HECIWG FSIO 2006 workshop in August 20-22 in Washington DC. The location was picked to encourage more HECIWG agency participation, and indeed three more HECIWG agencies were represented. The workshop was again, attended by top University, government HEC, and industry FSIO R&D people. The goals for this workshop were:

- To update everyone on the
  - 19 HECURA and two SciDAC2 FSIO research activities
  - programs available to get computing resources
  - activities to make operational data available to enable research
- To solicit input on
  - remaining gaps in the needed research areas
  - gaps in providing center support such as providing
    - computational resources
    - operational data available for research
    - getting the HEC community involved in the educational process

The information gathered at this and the previous workshop will be used to facilitate better coordinated government funded R&D in this important area in the coming years.

## Frequently Used Terms

*ASC* – The Advanced Simulation and Computing program supports the Department of Energy’s National Nuclear Security Administration simulation based stockpile stewardship.

*File system* – A combination of hardware and software that provides applications access to persistent storage through an application programming interface (API), normally the Portable Operating System Interface (POSIX) for I/O.

*Global* – refers to accessible globally (by all), often implies all who access a given resource see the same view of the resource

*HECRTF* – High End Computing Revitalization Task Force, an effort to make the US more competitive in high end computing

*HECIWG* – High End Computing Inter-agency Working Group, an inter US government agency working group to help coordinate government funding of R&D activities in the HEC area

*HECURA* – High End Computing University Research Activity – A government funded university research activity in the HEC area

*FSIO* – File Systems and I/O

*Higher level I/O library* – software libraries that provide applications with high level abstractions of storage systems, higher level abstractions than parallelism, examples are the Hierarchical Data Formats version 5 library (HDF5) ([http://www-rcd.cc.purdue.edu/~aai/HDF5/html/RM\\_H5Front.html](http://www-rcd.cc.purdue.edu/~aai/HDF5/html/RM_H5Front.html)) and parallel Network Common Data Formats library (PnetCDF) ([http://www-unix.mcs.anl.gov/parallel-netcdf/sc03\\_present.pdf](http://www-unix.mcs.anl.gov/parallel-netcdf/sc03_present.pdf))

*I/O* – input/output

*I/O Middleware* – software that provide applications with higher level abstractions than simple strings of bytes, an example is the Message Passing Interface – I/O library (MPI-IO) (<http://www-unix.mcs.anl.gov/romio>)

*Metadata* – information that describes stored data; examples are location, creation/access dates/times, sizes, security information, etc.

*Parallel* – multiple coordinated instances, such as streams of data, computational elements

*POSIX* - Portable Operating System Interface (POSIX), the standard user interfaces in the UNIX based and other operating systems (<http://www.pasc.org/plato/>)

*QoS* – Quality of Service

*SAN* – Storage Area Network, network for connecting computers to storage devices

*Scalable* – decomposition of a set of work into an arbitrary number of elements, the ability to subdivide work into any number of parts from 1 to infinity

*WAN* – Wide Area Network, refers to connection over a great distance, tens to thousands of miles

## **Research Themes Identified From the Workshops**

During the workshop, a number of research themes emerged. The recommended research topics are organized around these themes: metadata, measurement and understanding, quality of service, security, next-generation I/O architectures, communication and protocols, archive, and management and RAS. The following subsections describe the research appropriate to each theme area. Each subsection will cover both evolutionary and more revolutionary research topics that were identified as needing additional research attention. Input from both the 2005 and 2006 workshops will be provided for a complete picture of the area. Additionally, the list of HECURA and SciDAC2 I/O awards are described in Appendix A of this document.

### ***Metadata***

#### **HECIWG FSIO 2005 Problem Definition**

The 2005 workshop identified a number of research needs for metadata management. The 2006 HECURA call funded four strong proposals in response. These proposals addressed direct scalability, augmented metadata for IO scalability, applicability and benchmarking of alternative approaches, roles, semantic awareness, extensibility, and fundamental algorithms of implementation.

### ***Scalability***

#### ***Evolution***

##### ***Problem Definition***

Clustered file systems seem to be converging on an architecture that employs a centralized metadata service to maintain layout and allocation information among multiple, distinct movers. While this has had significant, positive impact on the scalability in the data path, it has been at the expense of the scalability of the metadata service. The transaction rate against the metadata service has increased, as has the amount of information communicated between the metadata service component and its clients has increased. These trends indicate that distributed metadata storage is key to successful scalability.

##### ***2005-2006 Progress***

The 2006 HECURA call produced a strong response in this area. As was noted, at the 2005 workshop, performance is critical for any petaFLOPS

attempt and scalable performance is paramount. All four, funded, proposals directly addressed this issue in various ways: peering and distribution of metadata information, on-store layouts, applicability of object-based storage to the problem, enabling scalability performance in consistency and coherency guarantees, and fundamental algorithms for efficient implementation of classic name spaces.

## ***Revolution***

### *Problem Definition*

In the near-term we are likely to see relatively simple metadata distribution schemes used to allow for more concurrency in metadata operations for cluster file systems with a moderate number of metadata storage devices. In the longer term, it will be necessary to extend these schemes in order to facilitate the use of very large numbers of metadata storage devices. Techniques for discovery of file objects and mapping of tree-based name spaces onto in a very large metadata space are just two potential research areas for long-term study of metadata scalability. Continued scaling implies an increase in in-flight data and metadata and adapting to this change is an area of great interest for scaling research as well.

### *2005-2006 Progress*

This topic was partially addressed in the responses to the 2006 HECURA call; however, not in any real, direct, way. Autonomics will provide for the management of a large number of storage devices but no direct examination of those devices with respect to file system or user metadata was proposed. Similarly, efficiency improvements for generic metadata was proposed in the fundamental algorithms response but scalability was not discussed. Many of the proposals discussed scalability but none offered novel architectural approaches. It is probably true that the existing approaches haven't been sufficiently, critically, examined. New approaches, offering other options would be welcome also.

## ***Extensibility***

### *Evolution*

#### *Problem Definition*

HEC applications are increasingly creating, storing, and relying on derived data and provenance information as part of the discovery process. At the moment, additional databases or files are used to store this information. At the same time, extended attribute support is becoming a common feature of local file systems. Additional work is needed to understand appropriate storage mechanisms for these user-created attributes within cluster and parallel file systems and to create interfaces to this data.

### *2005-2006 Progress*

One response in the FY'06 call addressed part of this extensibility issue. It proposes to examine whether and how object-based devices are to be leveraged in this area. Another call proposing work in fundamental implementation will also be applicable, although the researchers did not make specific note of this.

## ***Revolution***

### *Problem Definition*

As extendible metadata and data transparency become more common, it is likely that the file system will know increasingly more about the data being stored. With respect to metadata, an important issue will be how to store semantic information alongside file data in a manner that is accessible and understandable by the file system itself.

### *2005-2006 Progress*

One funded proposal in the FY'06 HECURA call did mention semantic indexing. However, only as to how content might be efficiently indexed and searched. Application at scale, in a large distributed machine, or environ was not contemplated. The proposal was more fundamental, an algorithm. Perhaps a follow-on proposal will be able to hybridize two or more approaches to extend the work into a distributed environment.

## ***Metadata and Archiving***

### ***Evolution***

#### *Problem Definition*

Archiving of cluster file systems is problematic for a number of reasons. One key factor is the vast number of individual items (e.g. files) that must be archived. Efficiently storing the metadata for these objects in a manner that is also efficiently accessed on streaming storage is an unsolved challenge.

#### *2005-2006 Progress*

There were proposals in the FY'06 call addressing the wedding of file systems with archival storage. It is difficult to predict the suitability and fortunes of the described approaches as this problem is, first, not new and, second, has been worked on many times previously without significant effect. Work always looks promising and good results are produced but these demonstrated benefits always seem to have issues when the work is contemplated for incorporation into existing and contemplated product from industry.

### ***Revolution***

#### *Problem Definition*

While simply archiving large volumes of data stored on cluster file systems is a challenge in itself, tighter coupling of storage system and archival system is desirable. One challenge in bringing file systems and

archival systems together is the need for additional metadata describing locations of data, which might in fact not even exist on the file system at that point in time. Additional challenges to archiving of file system data come when name space changes occur. Capturing novel file system organizations on archival storage will require new approaches.

#### *2005-2006 Progress*

No response produced the revolutionary approach discussed here. No new name space organization and, therefore, no issue introduced with respect to on-line versus archival name space integration appeared.

### ***Access Control Lists***

#### ***Evolution***

##### *Problem Definition*

Access Control Lists (ACLs) are a widely-implemented mechanism for limiting access to file system objects. The challenge in applying ACLs to cluster file systems is in their distributed nature. As we move away from centralized metadata storage to distributed metadata storage, efficiently verifying permissions will become more complicated and communication-intensive. Novel approaches to distributing ACLs and maintaining consistency of ACLs in a distributed environment are necessary to prevent these checks from becoming an artificial I/O bottleneck.

This item is closely related to security, of course. Here, however, we are talking about the issue in terms of performance and scalability. No proposal targeted at metadata management directly responded. While a couple of the funded proposals for security discuss researching the topic, none do so in terms of performance. The issue may be unavoidable, so progress in all of the metadata and the security responses should be tracked with this in mind.

#### *2005-2006 Progress*

One response did attempt to address the usability of security which begins to get at the heart of this issue. Also the POSIX HECEWG, an attempt to enhance the POSIX I/O API by the HEC community, effort is starting to address this.

#### ***Revolution***

##### *2005-2006 Progress*

No response produced a revolutionary approach to this issue.

### ***Data Transparency***

#### ***Evolution***

##### *Problem Definition*

Traditionally, file systems have operated in terms of streams of bytes. However, today's file systems are accessed by numerous, often

heterogeneous systems. In order to store data in a platform-independent manner, high-level libraries are used to convert data prior to storage on the file system. Augmenting file systems to understand basic data format semantics would allow these types of operations to be moved into the file system proper allowing the file system to make decisions on the best format for physical storage and likely reducing the overhead of data recoding.

#### *2005-2006 Progress*

No response in the list of funded activities for the 2006 HECURA call addressed this issue. Further, without researchers directly addressing it, it seems unlikely that any progress will be forthcoming. The topic itself is difficult to grasp and, given the lack of any response, it is probably best if the research community is educated and a specific request for research in the area is made, perhaps, in a later call.

### ***Name Spaces***

#### ***Revolution***

##### *Problem Definition*

In order to assist applications with managing enormous amounts of data, application programmers and data management specialists are calling for the ability to store and retrieve data in organizations other than the age-old file system tree-based directory structure. Data format libraries currently provide some of this function but are not at all well-mated to the underlying file system capabilities. Databases are often called upon to provide these capabilities but they are not designed for petabyte or exabyte scale stores with immense numbers of clients. Exploratory work in providing new metadata layouts and finding data in these new layouts is vital to address this identified need.

#### *2005-2006 Progress*

One response in the 2006 HECURA call briefly mentioned indexing and incorporation of semantic awareness. How that might be exposed or used is unclear. No response offered to examine the utility of such a mechanism. No response contemplated examination in the high-end computing arena with its scaling needs.

The revolutionary approach that the 2005 workshop thought was needed has not, so far, been proposed.

### ***Hybrid Devices***

#### ***Revolution***

##### *Problem Definition*

New storage technologies such as MEMS, MRAM, FLASH, and others all provide storage that is faster than spinning disk but at a higher cost. While it will be some time before such technologies supplant disk (if ever), these

technologies are very amenable to use as metadata storage or metadata cache spaces. Integrating these devices into file system infrastructure holds the promise of increased metadata rates.

#### *2005-2006 Progress*

The HECURA 2006 call produced no responses in this area. Perhaps the right input is not being solicited. Attendees at the 2005 and 2006 workshops leaned heavily toward file system research. While the storage industry was represented, no presentation or discussion mentioned any examination of the utility of hybrid devices.

#### ***Overall***

There were a number of funded HECURA projects that will attempt to address metadata issues.

- Collaborative Research: Petascale I/O for High End Computing [Maccabe]
- Collaborative Research: Techniques for Streaming File Systems and Databases [Bender]
- Applicability of Object-Based Storage Devices in Parallel File Systems [Wyckoff]
- Collaborative Research: SAM<sup>2</sup> Toolkit: Scalable and Adaptive Metadata Management for High-End Computing [Jiang]
- Improving Scalability in Parallel File Systems for High End Computing [Ligon]

#### **HECIWG FSIO 2006 Analysis and Update**

Additional research is needed to address longer-term concerns in the area of metadata storage and management.

Neither the HECURA call nor the 2006 workshop produced much in the way of truly revolutionary approaches for high-end computing. This is not to say that the evolutionary work is somehow deficient or lacking. However, one would expect that a vibrant research community would produce occasional fresh examinations of old problems. This has not occurred. Maybe this should not be surprising. The HECURA call marks something of a restart of research in high-end I/O and so, researchers probably have not had sufficient time or regained sufficient understanding of the issues to begin thinking outside of "the box".

It should be very useful to look, again, at all of these proposals in a year or more for new issues brought to light and contemplated actions. It's been a long time since this research community has been offered the opportunity to remove existing constraints and contemplate radical change.

#### **Identified Gaps**

- Revolutionary scaling
- Revolutionary extensibility
- Revolutionary name spaces

- Revolutionary File System/Archive metadata integration
- Revolutionary Hybrid devices exploitation
- Revolutionary Data Transparency

## ***Measurement and Understanding***

### **HECIWG FSIO 2005 Problem Definition**

Research tools for measurement and understanding of parallel file system and end-to-end I/O performance are needed for advances in future file systems. In parallel application building and tuning, there are a multitude of correctness and performance tools available to applications. In the area of scalable I/O and file systems, however, there are few generally applicable tools available. Tools and benchmarks for use by application programmers, library developers, and file system managers would be an enormous aid.

There is a need for research into evolutionary ideas such as layered performance measurement, benchmarking, tracing, and visualization of I/O related performance data. More radical ideas to be explored include end to end modeling and simulation of I/O stacks and the use of virtual machines for large scale I/O simulation. With research into these ideas, a future generation of high performance file systems could be understood and more efficiently pursued.

### **2005-2006 Progress**

There were several success stories within the HECIWG FSIO community in measurement and understanding. First, the NSF HECURA program has funded many proposals targeting the measurement and understanding of parallel file systems and end-to-end I/O performance. From this funding, we expect exciting results from the following:

- A collaborative venture between Alok Choudhary's group at Northwestern and Mahmut Kandemir at Penn State exploring scalable I/O middleware [Choudhary]
- The Arpaci-Dusseaus' group at Wisconsin providing an analysis of formal failure models for storage systems [Arpaci-Dusseaus]
- Tzi-Cker Chiueh's work on quality of service guarantees for scalable parallel storage systems will be predicated on a strong initial foundation of measurement and understanding of this problem area [Chiueh]
- Priya Narasimhan at Carnegie Mellon has been funded to research automated problem analysis of large scale storage systems [Narasimhan]
- Purdue's Mithuna Thottethodi will be investigating performance models and systems optimizations for disk-bound applications [Thottethodi]
- Erez Zadok at SUNY – Stony Brook is exploring many aspects of HEC file systems such as tracing, replaying, profiling and analyzing [Zadok]

Additionally, a group from Carnegie Mellon University has recently completed important work [Mesnier] in this area by developing a trace replay tool which automatically discovers causal events within HEC parallel applications. Replay tools such as this are

important as they can drastically speed-up the research cycle by allowing researchers access to a wider variety of application behaviors and can potentially allow the I/O patterns of private applications (e.g. classified government codes) to be publicly released and studied.

### **HECIWG FSIO 2006 Analysis and Update**

Discussion at the workshop indicates that there remain several important, currently unaddressed, gaps in this space. The attendees felt that current work, although highly important, is not looking at system workload in a realistic enterprise environment in which additional consideration must be given to questions arising from aging, reconfiguration, and workloads consisting of multiple heterogeneous applications. Further, the group feels that developing standards for HEC I/O benchmarks is important and that these benchmarks must account for the realistic enterprise environment challenges listed above such as aging. Also, testbeds for I/O research should be made available so that the cost of entry to do research in this area can be lowered. One final identified gap is a lack of cutting edge visualization tools to analyze large-scale I/O traces. Such a tool could help identify complex causal dependencies and would prove highly valuable for analysis of HEC I/O.

### **Identified Gaps**

- Understanding system workload in enterprise environment
- Standards for HEC I/O benchmarks
- Testbeds for I/O research
- Applying cutting edge visualization/analysis tools to large scale I/O traces

## ***Quality of Service***

### **HECIWG FSIO 2005 Problem Definition**

Quality of service (QoS) can be defined as features of a storage architecture which allow a user or administrator to recommend policies for data movement during Input/Output operations. These QoS policies can reach a broad range of integration into software, file systems, and hardware devices. Policies such as guaranteed I/O performance, specific redundancy requirements, or I/O priority settings will allow the system to perform optimally for a given work profile. Further research into areas such as adaptive QoS systems, end-to-end solutions, hardware support, and cross-system integration will revolutionize storage systems that will be created in the next few years. These research topics will bring the storage systems to a point where users, systems, or entire clusters can be insulated from each other, while using the same storage infrastructure. This will also allow for predictable I/O performance and response time for the users.

### **2005-2006 Progress**

There were a few successes based on this problem statement in the HECIWG FSIO community. Four projects were funded that address the QoS needs of the community. These projects are discussed in more detail in Appendix A

- Quality of Service Guarantee for Scalable Parallel Storage Systems [Chiueh]

- Active Data Systems [Reddy]
- Exploiting Asymmetry in Performance and Security Requirements for I/O in High-end Computing [Sivasubramaniam]
- End-to-End Performance Management for Large Distributed Storage [Brandt]

### **HECIWG FSIO 2006 Analysis and Update**

The topic of quality of service was discussed again at the 2006 workshop, focusing mainly on the areas that were not addressed in current research and in the HECURA funded and research. The high priority areas were robust availability, defining QoS as it relates to HEC, and policy management.

Robustness in the QoS area is an emerging new emphasis in this area and consequently has not been worked on extensively. For this work to move forward research needs to be done in the area of providing feedback from components of the system to the different layers of QoS subsystems. Protocols for this work will also need to defined and prototyped to determine their usability.

The High End Computing community needs to better define what QoS means in relation to the specific needs of the HEC community. QoS requirements are very different for HEC as opposed to real-time visualization, data capture, and multi-media applications and systems. Research in this area is required to define what is needed to define QoS for HEC and then determine ways to evaluate the systems. A set of benchmarks testing each area of QoS for HEC is one way to do this evaluation.

Policy management is a key component to a useable and robust QoS system. It must support the security, authorization, and policy control. These systems need to implement a system that can be managed automatically without high operator overhead. Research needs to be done in the areas of policy implementation, authorization, distributed QOS needs, and using computing bounds, estimated or measured.

### **Identified Gaps**

- Robust availability
- Defining QoS as it relates to High End Computing
- Policy Management for Quality of Service

## **Security**

### **HECIWG FSIO 2005 Problem Definition**

In the 2005 workshop, the topic of security was recognized as one of growing importance with several areas needing attention. The bulk of these issues were recognized to relate to security usability, functionality, and overhead.

In the area of usability of security, it was recognized that if security is not easy to both use and understand, it will not be used. Ease-of-use of APIs and interfaces is an area of research for file system and I/O researchers to study, simplifying the use of security features for file systems and I/O. Standardization and validation, in turn, would become important as APIs and interfaces emerge.

In the area of security functionality, the topics of key management, distributed authentication/authorization, and end-to-end encryption APIs were noted:

- Long-term key management is an area in security that needs research and offers opportunities in industry. Security that incorporates encryption of data at rest requires carefully considered long-term key management schemes. Some issues include handling encrypted data at rest for which the encryption algorithm has become easily breakable, the protection of keys that allow for flexible use, and the management and longevity of keys over long periods of time and throughout natural events.
- Security systems must be able to handle authentication and authorization issues in a completely distributed world, often with hundreds of thousands of entities needing protection. Further, collaboration between multiple sites and organizations is becoming more prevalent where security solutions are necessary to flexibly handle these distributed file system security situations in terms of scale, distance, and flexibility.
- End-to-end encryption needs a robust, accepted application program interface that enables all data on the storage service and moving across the network to reside and move in encrypted form. Such an interface would provide for the encryption and decryption of data within the client-side operating system.

In the area of performance overhead, security within file systems and I/O comes at a price. There will always be an overhead associated with providing security, of course, but given the massive scale that HEC environments represent, as well as geographically-dispersed HEC sites, building security solutions that have acceptable overhead is a difficult, but important, task. There is a need for research into security overhead for HEC security applications.

### **2005-2006 Progress**

To address some of the work needed in the area of security, two proposals were funded in the 2006 NSF HECURA Awards:

The areas of security usability and the trade-offs of security overhead versus performance are being pursued through Pennsylvania State University research on security framework [Sivasubramaniam]. The aim of this work is to provide a security solution that can easily accommodate different points in the security-performance space, offering different levels of security for clustered SAN-based architectures for different environments. Rather than attempt to accommodate each environment with a customized system, the framework would be tunable for performance or security based on site policies.

Another area of security being researched is that of long-term protection of stored data. Research at the University of Minnesota addresses scalable, global, and secure (SGS) online storage, building on the existing Lustre and Panasas object-based file systems [Du]. The work will investigate transparent, end-to-end encryption for high-performance backup and archival functions. Further, this work will investigate the area of long-term

protection of cryptographic keys, including loss/recovery of keys, user/group membership changes, and retrieval of old data.

### **HECIWG FSIO 2006 Analysis and Update**

The aforementioned research activities tackle several areas (usability, performance, end-to-end encryption, and key management) from the original problems noted in the 2005 workshop. There is certainly more work to be done in the area of security, however, as security issues are an ever-growing challenge. In a world with the issue of data stored on mobile laptops and portable hard drives, as well as the certainty of malicious parties bent on circumventing security measures, there are significant opportunities for novel designs and approaches.

In the 2006 workshop, additional areas of security research were discussed with existing topics from the previous year's workshop. The top ideas for research from the 2006 workshop included:

- Tracking the path of information as it flows through the system, determining where data has been and whether it has been compromised in-flight or at-rest
- Additional research into long-term key/algorithm management issues. In particular, as long-term persistence of security for storage holds different challenges than other types of security which can be more transient or sessioned in nature
- Performance and scalability overhead issues with security features
- Research on resilient security, including quick recovery from compromise
- Usability, addressing improved interfaces/APIs for ease-of-use
- A need for standards for secure deletion, balancing performance versus disk overwrite for deletion
- Understanding composition of security as it applies to end-to-end security
- Exploring the capability of quality of security as it pertains to HEC environments
- Developing methods for searching and indexing encrypted data

### **Identified Gaps**

- Tracking the path of information as it flows through the system
- Additional research into long-term key/algorithm management issues
- Impact of security overhead on performance and scalability

### ***Next-Generation I/O Architectures***

#### **HECIWG FSIO 2005 Problem Definition**

I/O stacks and architectures have been static for some time now forcing developers to adopt awkward solutions in order to achieve target I/O rates. Changes in the storage model could result in significant gains in performance and usability; however, there is little incentive for vendors to make major changes given that most new acquisitions require interoperability with legacy codes and systems.

Short and medium term efforts in this area should include the definition of extensions to the POSIX I/O API to support high-end computing (including support for access awareness, small and unaligned access, and mixed workloads), mechanisms to better integrate archival storage with on-line storage, research into system-wide collaborative caching, and impedance matching across the I/O stack. In the longer term, additional effort could help redefine the I/O stack itself, such as moving intelligence lower into the I/O stack, eliminating independent client-side caches, integrating application-specific capabilities, or incorporating semantic awareness into the I/O architecture. At the same time, user-space interfaces, novel devices and hybrid architectures, and peer-to-peer technologies provide challenges and opportunities in this space.

## *Interfaces*

### *Evolution*

#### *Problem Definition*

The POSIX file access calls were simply not designed for high-end computing outside of an explicitly shared memory model. The environment for which it was designed assumed that file descriptors, synchronization, and buffer management were all supported in local, directly accessed, memory by very low-latency operations. In today's high-end computing, the dominant solution is a cluster or multi-programmed parallel machine, and these architectures directly expose a distributed memory. Instead of the current situation (vendors receiving exemptions from the POSIX standard, and acquisitions therefore having to be non-POSIX compliant for performance reasons), we need to move to a POSIX standard set that supports typical HEC file systems and file systems I/O access patterns.

One type of enhancement would address the lack of support for collective I/O. For instance, a collective open could mitigate considerable startup times on very large clusters. Locking at the process group level might allow coherence to be maintained between applications without the overhead of traditional locking interfaces and implementations.

Another area for enhancement is in how applications describe accesses. Currently applications are very limited in their ability to describe access to disjoint regions in the "stream of bytes" that make up a file's data. By allowing applications to describe more complex accesses, we can significantly reduce the number of small transfers, converting them into a smaller number of large transfers instead.

#### *2005-2006 Progress*

An effort has begun under the Open Group to define HEC Extensions to the POSIX interface, an important step in adoption of extensions in vendor products. However, none of the funded proposals from the 2006 HECURA call addressed this area.

## ***Revolution***

### *Problem Definition*

Active disk, the ability to move part of an application near and on to the disk, has been an active area of research. However, no good interface and set of semantic rules has come along that would make it generally useful. Currently, it would seem that all solutions in this arena are restricted to modifications to support specific applications. A design that provides a “sandbox” so that multiple, unique applications can leverage the promise of active disk simultaneously would be welcome.

### *2005-2006 Progress*

One funded proposal addressed this area specifically and is a good start at defining the operating environment for processing on disk in conjunction with traditional workloads. More follow-on work will be necessary to fully understand how active disks will fit into the larger I/O picture.

Other revolutionary work related to interfaces is covered in the I/O Software Stacks section, below.

## ***Access Patterns***

### ***Evolution***

#### *Problem Definition*

File system designs tend to require tuning to efficiently support either large or small transfers. Unfortunately, they do not seem amenable to supporting both simultaneously. Worse, some applications attempt both during different phases of their processing. Something adaptive is clearly called for to avoid performance penalties both for the bulk streaming I/O (where bandwidth dominates) and for the smaller transactions that are latency sensitive. Modern self-describing data organizations such as are employed by HDF and CDF/netCDF sometimes scatter attributes throughout the file, interposed with the data, which can result in these mixed workloads. It is insufficient to simply handle small and large transfers. We must also be able to handle these in a directed, or vectored, fashion.

#### *2005-2006 Progress*

This 2006 HECURA call produced a very strong response in this area. Funded proposals cover data layouts for more efficient access, small and mixed I/O optimizations, and access pattern recognition. Further concepts and proposals addressing these challenges are addressed in the Caching and Coherence section, below.

## ***I/O Software Stacks***

### ***Evolution***

### *Problem Definition*

The existing IO software stack is deep and composed of different, sometimes disparate, modules. For instance, any distributed file system will rely on networking components. A fresh look at this stack, end-to-end, could address bottlenecks, especially when disparate modules call on each other. At the least, it would be highly desirable to have a standard method indicate a long latency path was in use so that normal timeouts were not employed. This is a problem with today's hierarchical storage management systems when part of the path is over a WAN. Object based storage systems will have a similar concern since their logical evolution is to have objects appear "equal" regardless of physical constraints (unequal file system behavior as well as distance).

### *2005-2006 Progress*

Funded proposals specifically address how object storage can best fit into the I/O stack, the use of anticipatory I/O scheduling to better manage resources, and how caching at various I/O layers could be coordinated to improve overall performance. No proposals directly address this issue of varying latencies in the I/O path in WAN environments.

## ***Revolution***

### *Problem Definition*

Many large engineering and physics simulations are burdened by CPU data cache coherency semantics. It's always been known that the ability to turn these off, where possible, enhances observed performance. Similarly, in the I/O world, a distributed application often does not require the services of a local buffer cache. Changes to the file system could be made that would react to or enable an application to remove these high-overhead but low value components from the control or data path could be usefully leveraged.

In the HEC space, we have clearly outgrown the decades old initiator-target I/O paradigm, yet the requirements for performance, data integrity, and coherency remain. Achieving a revolutionary approach to I/O is constrained within the respected paradigm.

File system solutions in the high end have relied on a core stack from commodity file systems design for workstations and servers. This could be redesigned in order to add to, remove from, or alter the placement of existing file systems components in the software stack. For instance, a local buffer cache could be removed in favor of a collaborative cache maintained by a distributed application for its own use. Early research indicates a benefit here; however, the only implementation has been in the presence of the local host buffer cache. Other components could be reexamined, in order to find potentially better placement within the stack.

### *2005-2006 Progress*

Revolutionary concepts in I/O stacks were partially covered by the 2006 HECURA responses. Work is funded in the areas of I/O graphs as part of the I/O stack, collective I/O, and tunable consistency. Additional work in communication protocols also addresses I/O stack concerns.

## ***File Systems***

### ***Revolution***

#### *Problem Definition*

While parallel file systems are a common component of HEC systems, very little work in recent years has focused on parallel file system architectures or optimizing these systems for HEC workloads.

Existing solutions utilize the network as a communications channel but there is power in the network far beyond that. While some solutions go so far as to generate multiple simultaneous transfers, there is not much that is fundamentally different from the classic initiator-target model. This method of using the network is highly portable but ignores the real power in high-end networks. Fresh approaches, such as peer-to-peer solutions, use new paradigms to reorganize storage. Such solutions would directly incorporate geographical distance in their cost function and significantly lower the bar that prevents massive replication, enhancing fault characteristics, or directly leverage other attractive properties in the network.

While many research file systems utilize components in user space, the practice is uncommon for production file systems. Performance data in the high end, at least, would seem to suggest that user-space file systems are a practical approach in general. Benefits from the eased development and debugging effort might, conceivably, offset the slightly higher call latencies. One of the historic reasons for preventing user space file system activity has been the data integrity concern – research into mechanisms (shared secrets, others) that could allay those concerns and permit user space and system space file system behaviors to co-exist is desirable.

File systems move bytes. Some government agencies believe that a file system augmented with knowledge of the data stored and transported could go much further. For instance, a file system that understood the machine word format used on the machine where the data was originally deposited, could reformat for a heterogeneous network of machines when required. As well, understanding the relationships between records accessed by an application could allow the file system to do a better job when storing the data or use much more intelligent prefetch strategies when retrieving it. Providing a method for the definition of such

associations could be useful. Then, researching how changes within the file system might leverage such information would be appropriate.

Long-lived data will need to convey format many years into the future, potentially. The concern is not just word lengths and endian issues but such things as floating point formats (mostly resolved) and the representations of complex math components. A generic API that could stand the test of time is desirable. Efficiency in performance, CPU cycle consumption, and storage will remain issues.

#### *2005-2006 Progress*

Work such as the PVFS project and the Light Weight File System (LWFS) project continue to push the boundaries of what is possible with user-space file systems and provide necessary infrastructure for additional research in parallel file systems.

Work funded by the 2006 HECURA call addresses the use of advanced network capabilities in the I/O system, server-to-server communication, autonomies, server co-scheduling, and content addressable storage. These concepts cover a wide space of possible file system designs and promise to uncover viable new architectures with characteristics better suited to HEC. No proposals specifically addressed this concept of long-lived data. This could be construed as an archiving issue, however, and not specifically a file system issue.

### ***Caching and Coherence***

#### ***Evolution***

##### *Problem Definition*

Modern operating systems inevitably buffer I/O transfers. While this has proved optimal in performance for locally attached storage, it presents problems when the goal is to efficiently use remote storage that is byte granular. The client operating system will typically employ buffers of fixed size. An application that does not fill a buffer when writing, can place the operating system in the uncomfortable position of having to read a full buffer, update the content and then prematurely flush the modified content back to the stable store. If a buffer system was available that was variable length or naturally supported modified sub-regions then byte-granular stable stores such as object-based disk could be efficiently leveraged.

Another related goal is to minimize the number of buffering steps required for data. It is not always done in current software stacks and that is a concern both from a performance and a memory usage perspective.

A file's address space is unqualified even when accessed by the several clients in a cluster or MPP machine. While this is an accepted well-

understood paradigm, it amounts to globally shared memory without any hardware assists; something long-ago recognized as sub-optimal. The core problem appears to be that a globally coherent view must be maintained. Many high-performance applications do not require such a thing but, the service section in a high-end machine would.

The metadata service relies upon lock services to accomplish the globally coherent views. By definition, such a thing is provided by the cluster, or MPP, service section. Usually, these are a magnitude, or more, smaller in size than the compute client section as they are simple overhead. They enable the compute section but do not directly contribute. Worse, the available lock algorithms do not seem to scale, so even if a larger service section was available, it would consume itself while trying to manage locks. Clearly, a renewed interest in scalable lock services is needed. Solutions involving the compute partition, to augment the power of the metadata service and relaxing the API semantics with respect to coherency, could lessen the load on the metadata service section.

#### *2005-2006 Progress*

A number of 2006 HECURA proposals cover aspects of caching and coherence, from collaborative caching (using the caches of multiple clients in concert) and multi-level caching (efficiently leveraging caches at multiple levels in the I/O stack) to enhanced prefetching algorithms to better fill caches. These efforts fit well with other work in I/O software stacks and interfaces. Improvements to metadata services that leverage atomic operations were also covered but in general, no solutions were proposed that attempt to leverage transactions in file systems or otherwise attempt to move away from the lock-based coherence paradigm. Some work discussed under Metadata is also applicable to the problems addressed here. Additional work is called for in this area.

### ***Novel Hardware***

#### ***Revolution***

##### *Problem Definition*

Storage devices have not changed fundamentally in 50 years. The advancements possible in allocation policy and layout given a radical change in the access latency to bandwidth ratios seem attractive to explore. New devices with promise along these lines as well as hybrids combining existing storage solutions could spark renewed interest in, and value from, these core file system areas.

The focus, today, is on capacity. Obviously, there is a need. However, the increasing capacities also come at a price. The greater probability of faults on a single unit now jeopardizes RAID systems at rebuild time. One avenue that is being explored in depth is to use the aggregate to offset the error probabilities. Others could be in the individual disk units themselves.

### *2005-2006 Progress*

Funded proposals cover new storage organizations to more efficiently use local storage in I/O systems and ways to integrate object storage and active storage into the I/O system. However, none of the funded proposals addressed the growing need for new redundancy schemes; additional work in this particular area is needed to fill critical caps.

## ***Archival Storage Evolution***

### *Problem Definition*

The directed graph mandated by the POSIX name space extends well into hierarchical storage systems. However, while solutions such as X/DSM allow a useful transition, for the user, from one part of the storage hierarchy to another, IO access to the file address space is problematic. Users unexpectedly encounter long delays while data is copied to or from the high-overhead portions of the storage hierarchy, for instance.

The presence of near-line and off-line storage encourages the user to view the storage system as infinite. This is in direct conflict with site policies normally. Inevitably, administrators require a gate or hurdle the user must encounter so that it is understood that scratch, ephemeral, and redundant data should not be placed into the deeper, or archival, layers of the hierarchy. In the past, this has been accomplished by adding manual steps to the process or imposing quotas. Both of those methods, however, are counterproductive in a name space that should, or could, span multiple layers of the storage hierarchy. This will become even more of a concern as the name space is expected to be global within an enterprise.

Many sites have archival mandates where some data lives forever. For these sites, moving from one product to another is difficult as a migration of this data from the old product to the new must be performed. Then, too, the amount of this data is forever growing which, as time goes by, makes the problem ever more difficult. Even the identification of such data sets is not incorporated. While migration is not addressed at all, industry does address mandated archival needs by providing special write-once file system and hardware solutions. This is unnatural, though, as the partitioning of the name space is necessarily tied to policy. A more natural solution would allow policy to be applied to storage without regard to file location in the name space.

### *2005-2006 Progress*

Only one 2006 HECURA proposal covered archives. While archival storage sits at the edge of what we traditionally consider HEC I/O concerns, more effort is needed in this area to maintain the viability of archival storage in HEC environments.

### ***Overall***

As part of SciDAC2, the SciDAC2 Petascale Data Storage Institute was created and the SciDAC Scientific Data Management Center for Enabling Technologies was continued. Both of these entities are focused on key problems in scientific computing at scale.

Many of the NSF HECURA funded projects address problems in this space as well:

- The role of I/O graphs and metabots in I/O architectures [Maccabe]
- How new organizations such as streaming B-trees might impact storage organization [Bender]
- Alternative I/O middleware organizations [Choudhary]
- Multi-level caching and alternative consistency semantics [Ma]
- The potential role of active networks in the I/O stack [Chandy]
- How tools might extract I/O patterns from applications for the purpose of prefetching and other optimizations [Chiueh]
- The role of server-to-server communication in parallel I/O systems [Ligon]
- The real-world implications of integrating active storage with traditional storage systems [Reddy]
- Advanced scheduling schemes, including anticipatory scheduling and co-scheduling [Shen]
- The server-side push concept in parallel I/O systems [Sun]
- Power-aware I/O architectures [Thottethodi]
- The convergence of object storage and parallel file systems [Wyckoff]

### **HECIWG FSIO 2006 Analysis and Update**

The projects above hit upon a large number of key issues in this area. However, there are still a number of outstanding issues. The 2006 workshop identified a number of areas where additional work is warranted. Underlying file system abstractions topped the list of areas that are not adequately covered by existing work. This area includes next-generation virtual file system (VFS) layers for operating systems, alternative name spaces and file system organizations, and transactional file systems (and integration of database concepts in general). Advances in autonomic storage systems were seen as critical, particularly given the near-term construction of I/O systems incorporating tens to hundreds of thousands of devices. Novel devices and hybrid I/O architectures, noted as an important issue in the 2005 workshop, still need additional attention. Additionally, the specter of HEC systems having multi-million way parallelism with the need for very small I/O operations coming from all processes and how to best deal with this issue was brought up.

One area that did not receive much attention in the responses to the 2006 HECURA call and was identified as critical by government agencies was high-level I/O libraries. While one could consider the proposed work in I/O software stacks to address this indirectly, a more substantial effort focused on data formats and interfaces for HEC could improve

performance, increase usability, and provide a long-lived data format, a combination that is very compelling.

### **Identified Gaps**

- Underlying file system abstractions
  - Next-generation virtual file system interface
  - Alternative naming and organization schemes
  - Convergence of database and file system technologies, such as a transactional file system
- Self-assembling, self-reconfiguring, self-healing storage components
- Architectures using  $10^4$ - $10^5$  storage components
- Hybrid architectures leveraging emerging storage technologies
- HEC systems with multi-million way parallelism doing small I/O operations

### ***Communications and Protocols***

#### **HECIWG FSIO 2005 Problem Definition**

One of the most important factors in the resulting performance and functionality of a parallel file system is the communication protocol that ties the system together. Typically file system developers build their protocol from scratch on top of low-level networking protocols such as SCSI or IP and tune their protocol to match their architecture and expected workloads. However, much of the functionality in parallel file systems is similar across implementations. Understanding the key components of parallel file system communication and how new technologies fit into these protocols is critical to effective parallel file system designs in the future.

In the near term, research into most effective integration of networking technologies such as advanced network adapters and alternative low-level protocols will help make best use of upcoming networks. Integrating object-based storage concepts into I/O protocols will make for a more efficient mapping of client I/O operations to device operations. Understanding how to best leverage the new features of NFSv4 and 4.1 (including the pNFS capabilities) will be an important step in the direction of improved usability and lowered development costs.

As parallel file systems continue to evolve, communication between I/O servers begins to play an ever-greater role. Our understanding of this type of communication is very limited and better understanding of how this communication differs from client communication, how we might leverage aggregate communication concepts from message passing or group communication concepts from the peer-to-peer community could revolutionize storage architectures.

## **2005-2006 Progress**

Some of these concepts are now being actively researched as a result of NSF HECURA funding:

- How collaborative caching might be best integrated into parallel I/O systems [Choudary]
- The integration of active networks into the I/O stack, bringing a new communication paradigm into the picture [Chandy]
- How servers might play a more active role in initiation of I/O operations [Sun]
- The impact of network placement and migration [Thottlethodi]
- How active storage devices could be first-class citizens in a parallel file system [Wyckoff]
- Alternatives for server-to-server and client-to-client communication in parallel file systems [Ligon]

## **HECIWG FSIO 2006 Analysis and Update**

The projects listed above hit on many of the key problems in parallel file system protocols and the newly-created SciDAC2 Petascale Data Storage Institute is active in NFSv4 and pNFS efforts. However, a few areas were noted as needing additional coverage at the 2006 workshop. Improving the interface between applications and the underlying file system with respect to expected access patterns was seen as particularly critical. While some facilities for these types of hints are available on some systems, often this information is lost in the I/O stack well before it hits the parallel file system.

Along these same lines, providing mechanisms through which the file system can report more information back to the application was also seen as very important. This could be data that could be used to better align high-level data structures to storage, optimize network transfers, or better understand I/O system behavior. Finally, investigations into how to properly incorporate one-sided communication models (e.g. RDMA) into I/O systems was seen as a potential win in the long term.

### **Identified Gaps**

- Expanding interface to file system in respect to expected access patterns
- Mechanisms for file system to report information back to the application
- Incorporating one-sided communication models into I/O systems

## ***Management and RAS***

### **HECIWG FSIO 2005 Problem Definition**

Management and RAS (Reliability, Availability, and Serviceability) are both obvious areas affected by immense scale. The number of storage devices, associated hardware, and software needed to provide the needed scalable file system service in a demanding and mixed workload environment of the future will be extremely difficult to manage given current technology. Advances must be made in massive scale storage management to enable management survival with future file system deployments. Additionally, RAS

at scale is another major issue. Given that future file systems will be based on tens of thousands or more mechanical devices with an extremely complicated software stack deployed at scale, it is likely that failure will be more the norm than the exception.

### **2005-2006 Progress**

Several of the funded NSF HECURA projects are directly addressing this important challenge:

- An analysis of formal failure models for storage systems [Arpaci-Dusseau]
- Improving scalability for HEC parallel file systems [Ligon]
- Automated problem analysis of large scale storage systems [Narasimhan]

Additionally, there was significant progress in this area in terms of completed work. Bianca Schroeder published an analysis [Schroeder] of machine failures at Los Alamos National Lab using publicly provided data (<http://institute.lanl.gov/data/lanldata.shtml>) recently made available by Gary Grider's group at Los Alamos National Laboratory. Bianca's analysis makes important contributions and contradicts the previous conventional wisdom in terms of failure predictability. This leads itself into her current work in using this new understanding of failure patterns to improve HEC checkpointing. The availability of the Los Alamos data set should continue to pay similar dividends as other groups attempt different analyses.

### **HECIWG FSIO 2006 Analysis and Update**

Although the workshop attendees were happy with the funded proposals and the work accomplished thus far, there was unanimous agreement that this problem space is particularly challenging and that large amounts of effort must continue to be expended in this space. Specifically, the group feels that HEC researchers would be well served by delving into the research in the modeling community. By emulating the complexity of those models, useful failures models could be developed to aid the HEC community research management and RAS. Further, the group agreed that more widespread dissemination and analysis of reliability information (such as the Los Alamos failure data) is particularly important.

### **Identified Gaps**

- Dissemination of reliability information
- More formal failure analysis
- Semantics for asynchronous events completion
- Autonomics

### **Archive**

#### **HECIWG FSIO 2005 Problem Definition**

The 2005 workshop identified I/O and file system areas of particular concern to high-performance archives supporting HEC systems. The scale and longevity of data in HEC archives adds some particular slants to these research areas:

- The complicating factors of RAS at the very large scale are a foremost concern for deep archives on disk.
- Long-lived archives experience extremes in namespace size, making efficient storage, management, and retrieval of file system metadata imperative and research into new namespace technologies attractive. Content-addressable storage and similar technologies show promise in finding, tracking, and managing large archives over long periods, but more work is needed.
- The longevity of archive files makes more imperative the ability to set and enforce policy to manage the data. Policy is also important in the lifecycle movement of data among layers in the archive hierarchy.
- The X/DSM POSIX file system interface for offline data is more than ten years old; modern, high-performance archives call for a new generation replacement.
- Long-term archives must contend with migration to new generations of hardware; emerging technologies such as object-based storage architectures may be particularly well suited for optimizing such movement and for enabling larger scale parallelism in the archive system.

### **2005-2006 Progress**

During 2005-2006, the HECURA File System and I/O research awards supported efforts in a number of areas directly pertinent to archives that support high-end computing.

Security requirements for archive are addressed in the research on long term (data and key) management in a high-performance hierarchical archive environment [Du]. Another area focused on flexible object-driven policy for combinations of security and performance needs [Sivasubramaniam].

The more extreme RAS needs of archive systems are addressed by research into more complete failure and problem analysis, both from the viewpoint of more complete thus more insight into failure-handling problems [Arpaci-Dusseau], as well as automating problem analysis [Narasimhan]. In addition, studies supporting monitoring, profiling, and analyzing large storage systems [Zadok] should contribute both to enhanced RAS and improved archive performance.

Research into adaptive I/O stack frameworks, including multi-layer coordination, holds especial promise in improving coordination of resources and layers in hierarchical archives [Ma].

Several awards focused on research with significant foci on metadata and scaling, including:

- Scalable, adaptive metadata management contending with access patterns, load balancing and caching in parallel and distributed filesystem environments [Jiang]
- Investigating active caching, buffering, communication, and autonomies in support of scalable metadata operations, improvements in handling small unaligned data accesses, and auto-configuration [Ligon]
- Exploring parallel versions of new methods that show promise for significant performance increases in indexing and scanning (meta)data [Bender]

Many other research awards included aspects that are central to current and future concerns for archive. Support for scalable archive metadata operations would be encompassed in the exploration of the suitability of object based storage for parallel file systems [Wyckoff]. Offline techniques for deriving metadata [Maccabe] and support for filtering and other active functions “in the data path” [Maccabe, Wyckoff, Reddy] could be used for data “scrubbing”, extraction, and transformation or conversion to new archive formats.

In addition, the collaborative and education opportunities provided by the two SCIDAC2 I/O projects [Gibson, Shoshani] and the LANL//UCSC educational institute on scientific data management will advance areas of interest to archive as will the efforts on HEC extensions to POSIX standards.

### **HECIWG FSIO 2006 Analysis and Update**

At the 2006 workshop, the archive gap discussion focused on the following areas:

- Standard, transparent, interoperable means to deal with archives and archive file systems:
  - Standardizing archive attributes
  - Developing POSIX standards for archive interfaces, including policies on files and directories, as well as the ability to search part of an object
  - Possibility of developing a new standard VFS layer for archives, or some other standard means, that would provide user-transparent, interoperable ways to deal with archives and archive file systems as an alternative to modifying existing POSIX interfaces to accommodate special needs of archives
- Alternate access methods to archives and their components: these ranged from higher-level aspects contending with archive content (e.g., map reduction techniques such as those employed by the Google API) to lower-level functions such as alternate interfaces for accessing and addressing/naming storage hardware and objects
- Reliability is becoming paramount because of proliferation of devices in archives but RAS techniques often conflict with HEC needs; e.g., proactive, prophylactic device “scrubbing” is at odds with the need to minimize power consumption
- Growing interest in parallel archives, especially as they can be aligned with “Information Lifecycle Management” activities, from commercial HEC sites
- Automating the generation of archive attributes and content-metadata
- Management of versions and snapshots in archives including compression and navigation techniques

### **Identified Gaps**

- Standardizing archive attributes and automated generation of archive attributes
- POSIX Standards for archive interfaces
- Standardized VFS layer for archive
- Alternative access methods involving indexing
- Long term disk device reliability
- Commercial parallel archives
- Managing versioning in archives

## ***Assisting Standards***

Over the past decade, the HEC community has had a role in the formation and adoption of various FSIO related standards. The most notable are:

- The ANSI T10 1355D specification for Object Based Storage Devices (OBSD), the standard which is currently in draft 1, nearing draft 2, specifies the interfaces to object storage devices. The HEC FSIO community has provided input and funding to various parts of this specification development and will continue to do so. One avenue of HECIWG FSIO involvement in this effort is the SciDAC2 Petascale Data Storage Institute.
- The IETF NFSv4 standard including the new pNFS portion of the NFSv4.1 minor revision of the NFSv4 specification. The HEC FSIO community has provided input and funding to various parts of this specification development. One of the prime sites for NFSv4 activity has been funded partially by DOE. The pNFS, the age old idea of separation of data and control, concept came directly from HEC needs and pressures in the late 1990's. The HECIWG FSIO team with the HEC FSIO community will continue to provide input and funding of various kinds to this effort. One avenue of HECIWG FSIO involvement in this effort is the SciDAC2 Petascale Data Storage Institute.
- The newly formed Open Group HEC Extensions to the POSIX standards work has also been an outcome of HECIWG FSIO and the HEC I/O community work. This effort initially is focusing on extending the POSIX I/O API to accommodate the needs of HEC, in particular for highly parallel/cooperating/concurrent applications. The HECIWG FSIO team and the HEC I/O community will continue this valuable work. One avenue of HECIWG FSIO involvement in this effort is the SciDAC2 Petascale Data Storage Institute.

The HECIWG FSIO group and the HEC I/O community will no doubt continue to make progress on affecting these and other standards efforts. At the HECIWG FSIO 2006 workshop, an update was given on NFSv4.1/pNFS and the POSIX I/O API efforts. These presentations are available at the conference web site <http://www.nitrd.gov/subcommittee/hec/workshop/>.

### **Identified Gaps**

- Continued support of POSIX HEC Extensions efforts including prototype implementations
- Continued support for OBSD standards including keeping a reference implementation up to date
- Continued support for NFSv4, especially pNFS prototype work

## ***Assisting Research and Education***

In addition to recommended research topics discussed in detail in the previous sections, the HECIWG FSIO 2005 also called for the HEC FSIO community to provide university I/O center support in the forms of computing and simulation equipment availability,

availability of operational data to enable research, and HEC involvement in the educational process were called out as areas needing assistance. The following sections discuss things that have been done in the past year in these areas as well as the input received in these areas at the HECIWG FSIO 2006 Workshop.

### **Identified Gaps**

- Availability of computing equipment
- Availability of simulation tools
- Availability of trace and performance data
- Availability of failure and RAS data

### ***Data Availability***

#### **HECIWG FSIO 2005 Problem Definition**

At the HEC 2005 workshop, participants identified access to central clearinghouses of HEC data as necessary to understand and make progress on problems seen at HEC sites. Data that would be useful to these researchers include trace data from real HEC applications, synthetic applications that approximate the behavior of real HEC applications, historical data about failure rates of HEC systems, and very basic machine and environment information.

Challenges in providing this data are both political and technical. Politically, it may be impractical to provide data which may be classified to outside entities. Technical challenges in providing traces or synthetic applications involve the level of detail to provide. As the focus of this workshop is on HEC I/O, one simple answer would be to provide data only about the file system activity of the applications. However, this detail may be insufficient as other aspects of the application's behavior may influence their interaction with the file system. For example, a trace of an HEC application run on a machine with a large buffer cache may show less file system activity due to caching than the same application run on a machine with a smaller buffer cache. Additionally, storage and distribution solutions for this clearinghouse data need to be designed and implemented.

#### **2005-2006 Progress**

In response to the requests made for data from the HEC 2005 workshop, substantial progress has been made with the public release of data and more is in progress. Los Alamos National Lab has released general machine information, failure, event, and usage data for 22 of their premier supercomputers from the past nine years. In some cases, the records released cover the entire life of the machines from initial use to decommissioning. In addition to releasing over 23,000 failure, event, and usage records, LANL released some papers on computer failures and are publicly available at <http://institute.lanl.gov/data/lanldata.shtml>.

LANL researchers also made available to the public a synthetic application that mimics the I/O profiles of many of their important, but classified, applications. The synthetic benchmark is available at <http://public.lanl.gov/jnunez/>.

#### **HECIWG FSIO 2006 Analysis and Update**

During the 2006 workshop, two presentations were given on two technical reports from Carnegie Mellon derived from the released data and the synthetic application. Bianca Schroeder presented joint work with Garth Gibson on their technical report “A Large-scale Study of Failures in High-performance-computing Systems” [Schroeder] and Mike Mesnier presented “//TRACE: Parallel Trace Replay with Approximate Causal Events” [Mesnier]. During this session, the formation of the Computer Failure Data Repository was announced as a central place to store and make available failure, usage, and event data and contain pointers to sites that already make this data available. In addition, it was announced that there are several other organizations that are planning to release data including HP Labs, the Library of Congress, and Pittsburgh Supercomputing, and that LANL will be releasing synthetic and real application I/O traces.

After the presentations, the Data Availability discussion was broken up into two groups: one focusing on failure, usage, and RAS data and the other on traces and performance data.

In the area of traces and performance data, participants identified the following as gaps in making forward progress:

- lack of analysis tools
  - Instead of writing new analysis tools, we can leverage others efforts by reviewing existing tools and look into extending these existing tools to apply to HEC I/O.
- set of benchmarks that exhibit certain categories of behavior
  - Applications are evolving, which change their I/O patterns. Benchmarks that are general or cover a category of behavior has more of a chance of being relevant than very specific benchmarks.
- specify and use common formats, collection programs, obfuscation programs
  - Trace formats must be widely extensible because the number of things to trace will evolve over time with experience and requests from researchers using the traces.
  - There is a tremendous amount of data that should be captured that make the traces and benchmark results more relevant. System-wide profiler-like tools are needed to capture metadata describing traces in terms of application domain, number of procs, etc.
- common clearinghouse for traces, synthetics, benchmarks, and performance data
  - Consolidate the number of sites that provide data or provide pointers to those sites in one location
  - The owner of the repository should be neutral. Both SNIA and USENIX have expressed interest or put effort in this area and Los Alamos National Lab and Carnegie Mellon University have data repositories

In the area of failure, usage and RAS data availability, besides expressing concern over the anonymity of the data and how it will be used by vendors and their marketing, participants identified the following as gaps in making forward progress:

- need for standards, best practices, and guidance on what and how to capture relevant data
- evaluate existing models
  - Common trace formats etc. from the commercial computing world
  - Can we learn from the medical industry on anonymous data
  - Commercial autonomies work
- work with cluster vendors/industry/users to capture any of this data from the thousands of clusters in the world
- Sell this as a pre-competitive research topic
- Engage the vendors on how do we get proactive failure handling
- There may be no way to ever get vendors to give this stuff away

#### **Identified Gaps**

- Availability of data from HEC sites
- Lack of analysis tools
- Need for standards, best practices, and guidance on what and how to capture relevant data

### ***Education, Community, and Center Support***

#### **HECIWG FSIO 2005 Problem Definition**

At the HECIWG FSIO 2005 workshop, there was a recognition that the HEC FSIO community should find ways of supporting students working in the general area of I/O as well as students working more specifically on I/O within HEC. Investment to support the research of these students was considered worthwhile both because they may provide important research while still in school as well as by cultivating these students such that they may continue to work on HEC I/O problems following their graduation and, with any luck, become the next generation of HEC I/O experts.

#### **2005-2006 Progress**

There were a few success stories within the HECIWG FSIO community to help address this important need of education, community, and center support.

- Probably the most notable of the accomplishments was the NSF/HECURA research funding effort in the FSIO area. This is one of the first government funding efforts directed at the FSIO area and sent a signal to universities and vendors that the HECIWG FSIO effort is serious and capable of producing benefits.
- The SciDAC2 FSIO awards both have outreach to universities and industry in their scope. The PDSI SciDAC Institute calls out directly working with

- universities to raise awareness and understanding of the FSIO problems and to encourage advancement of curricula and other educational endeavors in the HEC FSIO area.
- An educational institute for scalable scientific data management was put in place by LANL with UCSC. This institute provides funding for shaping curricula and student fellowships in the area of HEC FSIO.
  - Of course, there were numerous joint papers and other joint research done between HEC sites, vendors, and universities as well.

### **HECIWG FSIO 2006 Analysis and Update**

At the 2006 workshop, this topic was broken up into two areas: curricula needs and center/community support.

#### ***Curricula needs***

A number of very good ideas were brought out in this area. The top ideas are as follows:

- HEC site and industry provided lectures in the class room
- Providing access to large scale computing resources for curricula work (this topic is covered in more detail in the next section about availability of computational resources)
- Find a way to harness the gaming industry excitement in the curricula program
- More HEC site and industry internships
- Flashy K-12 and undergraduate shows and tours of HEC sites
  - It would be very helpful if decommissioned equipment could be sent to high schools and even middle schools so that youth could see the inside of a disk drive, or see a working tape robot or other interesting show and tell items. Most high school students have never seen the inside of a computer, or a disk drive, or a tape cartridge. This activity might generate interest in pre-college and even undergraduate students.
- Curricula development funding to assist in
  - Developing new storage or storage centric courses
  - Developing materials, problem sets and answers, text book
  - Inject storage topics into existing classes
  - National Lab/Industry input/endorsement of this curricula building activity

#### ***Center/Community support***

A number of very good ideas were brought out in this area. The top ideas are as follows:

- Data to study (addressed before in data availability section )
- Large testbed environments and simulation environments to be used for research (addressed in availability of computational resources section)
- Promote a separate research storage related program at NSF that is ongoing, not just a one shot HECURA program.
- Provide more industry/HEC site internships.
- Conduct marketing of the importance of the FSIO and HEC FSIO efforts
  - Come up with an appealing name
  - Provide some awards or recognition for achievement in this field

- Industry luminary speaking on a popular media like NOVA
- Exploit ACM and other societies, job fairs, and journals like ACM Queue
- Flashy technology, how it works, shows at K-12 and undergraduate level
- Come up with a clean message as why storage/IO matters
- Engage the Other Discipline (MIS, Business, CE, EE, etc) as part of the community to gain broader recognition
- Use technology roadmaps as motivators
- Fully exploit NSF and other educational funding opportunities.
- Promote a Grand Challenge for information management.
- Engage the ACM K-12 Program, promote programming contests etc.
- Sponsor a top ten open problem in storage
- Lobby industry leaders to lobby government funding sources/agencies for a sustained investment in the storage and FSIO area.

### **Identified Gaps**

- HEC site and industry provided lectures in the class room.
- Find a way to harness the gaming industry excitement in the curricula program.
- More HEC site and industry internships.
- Flashy K-12 and undergraduate program including shows and tours of HEC sites.
- Promote a separate research storage related program at NSF that is ongoing, not just a one shot HECURA program.
- Conduct marketing of the importance of the FSIO and HEC FSIO efforts
- Promote a Grand Challenge for information management.
- Sponsor a top ten open problem in storage
- Lobby industry leaders to lobby government funding sources/agencies for a sustained investment in the storage and FSIO area.

### ***Availability of Computational Resources***

#### **HECIWG FSIO 2005 Problem Definition**

One of the most frequently echoed problems expressed at the 2005 workshop was the difficulty faced by many researchers; lack of access to real HEC applications and computing platforms. The researches felt that government investment is needed to support efforts that allow them access to both the physical and virtual infrastructure that they need in order to participate in and conduct HEC FSIO research.

Many would-be HEC researchers, particularly those in academia, lack access to the large parallel computers typically used in HEC applications. Therefore, the development and maintenance of open testbeds would enable these researchers to contribute in the area of HEC. There are at least four possible ways in which this physical infrastructure could be provided.

- The first is through direct acquisition: by providing large amounts of funding, new testbeds could be directly purchased. However, the cost of purchasing very large parallel systems could be prohibitively expensive.

- A second approach would be to develop sharing and sandboxing mechanisms by which “guests” could use the computational resources at other institutions. The sharing mechanisms would need to ensure appropriate priority schemes such that the guests would not pre-empt the compute time of the presumably more important local users.
- A third approach is using a virtual machine such that one computer architecture is made to appear like another through the use of complex software. One disadvantage of early virtual machine systems was a significant performance cost, but more recent work in this area has reduced this penalty. To date however, there is no virtual machine system for large parallel architectures; as such, this could be a project worthy of government investment. However, creating a parallel architecture from a physical architecture which is not parallel is problematic and it could probably only be done with large performance penalties. However, projects dealing with non-performance related aspects such as fault tolerance, functionality, and correctness would be feasible.
- A fourth approach is developing *simulation platforms*. With realistic timing models, these simulation platforms could even provide realistic performance evaluations. However, simulated systems cannot run unmodified applications as can virtual machines. Simulations can provide insight into future system development by exploring trends and simulating systems that do not currently exist.

### **2005-2006 Progress**

There were a few success stories within the HECIWG FSIO community to help address the availability of computational resources to enable research. Three ways for university researchers to gain access to large scale HEC computational resources were provided.

- The DOE Office of Science 2007 INCITE Program - Now in its fourth year, the Innovative and Novel Computational Impact on Theory and Experiment (INCITE) program using ORNL/LBNL/PNNL/ANL world class resources can be sought by researchers.
- The NSF has a research infrastructure funding program that can be applied for by researchers.
- Also, there is always the possibility of collaboration with HEC sites such as national labs on jointly interesting problems. Frequently, this results in getting university researchers access or exposure to large computational resources. The SciDAC2 teams would be good candidates for such collaboration.

### **HECIWG FSIO 2006 Analysis and Update**

While this topic was not discussed directly at the 2006 workshop, it did come up in several settings, especially in the area of center/community support and curricula needs breakout sessions.

The following were the most prevalent comments.

- The INCITE program might work pretty well for simulation and modeling activities but given the non mainstream computing resources being offered and the lack of system privileges available, the INCITE program may not be a very complete answer.

- There is a need to get access both for research and for educational/class activities.
- It would be very helpful if decommissioned equipment could be sent to high schools and even middle schools so that youth could see the inside of a disk drive, or see a working tape robot or other interesting show and tell items. Most high school students have never seen the inside of a computer, or a disk drive, or a tape cartridge. This activity might generate interest in pre-college and even undergraduate students.

### **Identified Gaps**

- Availability of computational resources for research (destructive) and educational (non-destructive) activities
- Disk and system simulators
- Virtual parallel machines

### **Working Group Gaps Analysis**

The high end computational needs can be provided by INCITE/NSF supercomputing sites. The low end clusters can be provided by NSF infrastructure and industry. Providing midrange testbeds for I/O research, something larger than a small university cluster but smaller than a large INCITE or NSF supercomputer center, where reconfiguration could be done for computer science research would be of value as well for enabling research. Additionally providing virtualization software and disk simulator software to enable research would be of great value as well.

### **Conclusion**

In the near future, sites will deploy supercomputers with hundreds of thousands processors routinely. Million-way parallelism is around the corner and, with it, bandwidth needs to storage will go from tens of gigabytes/sec to terabytes/sec. Online storage requirements to support work flows for efficient complex science will begin to approach the exabyte range. The ability to handle a more varied I/O workload ranging seven orders of magnitude in performance characteristics, extremely high metadata activities, and management of trillions of files will be required. Global or virtual enterprise wide-area sharing of data with flexible and effective security will be required. Current extreme-scale file system deployments already suffer from reliability and availability issues, including recovery times from corruption issues and rebuild times. As these extreme-scale deployments grow larger, these issues will only get worse. It will possibly be unthinkable for a site to run a file system check utility, yet it is almost a given that corruption issues will arise. Recovery times need to be reduced by orders of magnitude, and these types of tools need to be reliable, even though they may rarely be used. The number of storage devices needed in a single coordinated operation could be in the tens to hundreds of thousands, requiring integrity and reliability schemes that are far more scalable than available today. Management of enterprise-class global parallel file/storage systems will become increasingly difficult due to the number of elements involved, which will likely approach 100,000 spinning disks with widely varying workloads. The challenges of the future are formidable.

The publishing of the document, “HPC File Systems and Scalable I/O: Suggested Research and Development Topics for the Fiscal 2005-2009 Time Frame” [Appendix C] and the designation of file systems and I/O as a national focus area beginning in FY06 laid the framework for the HECIWG FSIO 2005 Workshop. This workshop helped identify, categorize, and prioritize the needed research in this area of HEC. Using the research topics document and the HECIWG 2005 Workshop document, the HECIWG organization and the HEC I/O community made major progress in the 2005-2006 timeframe. Some of the progress is summarized as follows.

- In the area of R&D
  - a National Science Foundation (NSF) HEC University Research Activity (HECURA) solicitation for university research in the FSIO area was written based on the eight areas of research need from the HECIWG FSIO 2005 workshop
  - the NSF/HECURA solicitation was conducted resulting in 62 proposals from over 80 universities
  - 19 HECURA awards were made from a careful analysis of the proposals
  - the DOE Office of Science awarded two SciDAC2 FSIO projects
- In the area of providing computational resources
  - the DOE Office of Science INCITE program for supplying computing clusters
  - NSF infrastructure program for providing computing infrastructure
- In the area of providing operational data to enable research
  - LANL release of failure, event, and usage data
  - PSC, Library of Congress, and HP working on data release
  - Consortia for failure data release is forming up

Additionally, due to the success of the HECIWG FSIO 2005 workshop and subsequent activities, a permanent HECIWG FSIO advisory group was formed to help continue to advise the HECIWG in the coordination of FSIO activity.

The HECIWG FSIO 2006 workshop in August 20-22 in Washington DC had the following goals.

- To update everyone on the
  - 19 HECURA and two SciDAC2 FSIO research activities
  - programs available to get computing resources
  - activities to make operational data available to enable research
- To solicit input on
  - remaining gaps in the needed research
  - gaps in providing I/O and Storage Research Center support such as providing
    - computational resources
    - operational data available for research
    - getting the HEC community involved in the educational process

The information gathered at this and the previous workshop will be used to facilitate better coordinated government funded R&D in this important area in the coming years.

The HECIWG FSIO 2006 workshop was well received and accomplished its goals. Attendees were quite happy about the HECURA and SciDAC2 activities and were extremely interested in the other support activities, especially the providing of computer site operational data and the HEC involvement in the educational process. The workshop presentation materials can be found at <http://www.nitrd.gov/subcommittee/hec/workshop/>. It was obvious from opinions gathered that this HECIWG FSIO work is a sorely need activity and that the job of the HECIWG FSIO group is far from done. Many research gaps were identified and lots of good ideas about how to expand the HEC FSIO effort were proposed.

Analysis of the 2006 workshop identified gaps by the HECIWG is summarized below:

### **Research Gaps**

The most important research gaps that remain are:

- Revolutionary approaches to metadata for scaling, searching, extensibility, and data lifetime management
- Automated self tuning, management, configuration, etc. (good research starting but very difficult topic)
- Next generation virtual file system (VFS) interface for operating systems
- Scaling to multi-million way parallelism for small I/O operation workloads
- Understanding complex system workloads

### **Research Enabling and Infrastructure Gaps**

The most important research enabling and infrastructure gaps that remain are:

- Computing site failure, usage, event, and performance data, including recommended standard formats, data to collect, and tools for collecting and using these data.
- Availability of simulation and virtualization tools, for simulation of disks, disk arrays, storage area networks, and even file systems
- Midrange testbed for I/O research, something larger than a small university cluster but smaller than a large INCITE or NSF supercomputer center, where reconfiguration could be done for computer science research, would be of value as well for enabling research.
- Benchmark clearinghouse

### **Education Gaps**

The most important education gaps that remain are:

- K-12 and undergraduate program including tours of HEC sites
- HEC sites and industry provided lectures in class rooms
- Harness the gaming industry excitement in curricula
- More HEC site and industry internships
- Promote recurring NSF research funding in I/O and Storage
- Promote recurring NSF education funding in I/O and Storage
- Promote I/O and Storage grand challenge and top 10 problems

### **Standardization Gaps**

The most important standardization gaps that remain are:

- Information sharing between file systems and applications
- Archive interfaces

### **Productization Gaps**

The most important productization gaps are:

- Exploitation of hybrid storage devices (disk and non volatile memory)
- Exploitation of active storage technology
- Testing of storage stripe widths beyond 10,000
- Commercial parallel archiving products
- Long term security key management and electronic signing and tracking of data

The HEC FSIO activities are off to a great start, but much work remains.

The HECIWG organization thanks the workshop coordinators and participants for helping to conduct and participating in both the 2005 and 2006 workshops. The workshops have both been extremely successful.

The HECIWG FSIO community would like to thank the NSF, DOD, DARPA, and DOE for making the HECURA and SciDAC2 FSIO awards possible.

The HECIWG organization would also like extend a special thank you to:

- Almadena Chtchelkanova at NSF,
- Fred Johnson, Thuc Hoang, Robert Meisner at DOE, and
- John Grosh formerly of DARPA

for their unwavering support of the HECIWG FSIO activity.

## References

Arpaci-Dusseau, Remzi H., Andrea C. Arpaci-Dusseau, Benjamin R. Liblit, Miron Livny, and Michael M. Swift. "NSF 06-503: Formal Failure Analysis for Storage Systems." High End Computing University Research Activity NSF 06-503 (2006)

Bender, Michael A. and Martin Farach-Colton. "Collaborative Research: Techniques for Streaming File Systems and Databases." High End Computing University Research Activity NSF 06-503 (2006)

Brandt, Scott A., Darrell D. E. Long, and Carl Maltzahn. "End-to-End Performance Management for Large Distributed Storage." High End Computing University Research Activity NSF 06-503 (2006)

Chandy, John A. "Active Storage Networks for High End Computing." High End Computing University Research Activity NSF 06-503 (2006)

Chiueh, Tzi-Cker. "Quality of Service Guarantee for Scalable Parallel Storage Systems." High End Computing University Research Activity NSF 06-503 (2006)

Choudhary, Alok N., Mahmut T. Kandemir and Rajeev S. Thankur "Collaborative Research: Scalable I/O Middleware and File System Optimizations for High-Performance Computing." High End Computing University Research Activity NSF 06-503 (2006)

Du, David H., Yongdae Kim and David J. Lilja. "Integrated Infrastructure for Secure and Efficient Long-Term Data Management." High End Computing University Research Activity NSF 06-503 (2006)

Felix, Evan, Gary Grider, Rob Hill, Bill Loewe, Rob Ross, Lee Ward. "HPC File Systems and Scalable I/O: Suggested Research and Development Topics for the fiscal 2005-2009 time frame." 10 Oct. 2006 <ftp://ftp.lanl.gov/public/ggrider/HEC-IWG-FS-IO-Workshop-08-15-2005/FileSystems-DTS-SIO-FY05-FY09-R&D-topics-final.pdf>

Gibson, Garth, Evan Felix, Gary Grider, Peter Honeyman, William Kramer, Darrell Long, Philip Roth, Lee Ward. PetaScale Data Storage Institute, SciDAC2 10 Oct. 2006 <http://www.scidac.gov/compsci/PDSI.html>

Jiang, Hong, Yifeng Zhu, David R. Swanson, and Jun Wang. "Collaborative Research: SAM^2 Toolkit: Scalable and Adaptive Metadata Management for High-End Computing." High End Computing University Research Activity NSF 06-503 (2006)

Ligon, Walter B. "Improving Scalability in Parallel File Systems for High End Computing." High End Computing University Research Activity NSF 06-503 (2006)

Ma, Xiaosong, Anand Sivasubramaniam, Yuanyuan Zhou, John M. Blondin and Vincent W. Freeh. "Collaborative Research: Application-adaptive I/O Stack for Data-intensive

Scientific Computing.” High End Computing University Research Activity NSF 06-503 (2006)

Maccabe, Arthur B., Karsten Schwan, Patrick G. Bridges, Greg S. Eisenhauer, Ada Gavrilovska, Patrick A. Widener and Matthew Wolf. “Collaborative Research: Petascale Storage for High End Computing.” High End Computing University Research Activity NSF 06-503 (2006)

Mesnier, Mike, Gregory R. Ganger, James Hendrix, Julio Lopez, Raja R. Sambasivan, Matthew Wachs. “//TRACE: Parallel Trace Replay with Approximate Causal Events” Carnegie Mellon University Parallel Data Lab Technical Report CMU-PDL-06-108, September 2006.

Narasimhan, Priya, Chuck Cranor and Gregory R. Ganger. “Toward Automated Problem Analysis of Large Scale Storage Systems.” High End Computing University Research Activity NSF 06-503 (2006)

Reddy, A.L. Narasimha. “Active Data Systems.” High End Computing University Research Activity NSF 06-503 (2006)

Schroeder, Bianca and Garth Gibson. “A Large-scale Study of Failures in High-performance-computing Systems.” Proceedings of the International Conference on Dependable Systems and Networks (DSN2006), Philadelphia, PA, USA, June 25-28, 2006.

Shen, Kai. “Concurrent I/O Management for Cluster-based Parallel Storages.” High End Computing University Research Activity NSF 06-503 (2006)

Shoshani, Arie, Ilkay Altinas, Alok Choudhary, Terence Critchlow, Bill Gropp, Chandrika Kamath, Wei-Keng Liao, Bertram Ludaescher, Jarek Nieplocha, Steve Parker, Rob Ross, Doron Rotem, Nagiza Samatova, Rajeev Thakur, Jeff Vetter, Mladen Vouk. Scientific Data Management Center for Enabling Technologies, SciDAC2 10 Oct. 2006 <http://www.scidac.gov/compsci/SDM.html>

Sivasubramaniam, Anand and Patrick D. McDaniel. “Exploiting Asymmetry in Performance and Security Requirements for I/O in High-end Computing.” High End Computing University Research Activity NSF 06-503 (2006)

Sun, Xian-He. William D. Gropp and Rajeev S. Thakur. “HECURA: The Server-Push I/O Architecture for High End Computing.” High End Computing University Research Activity NSF 06-503 (2006)

Thottethodi, Mithuna S., Vijay S. Pai, Rahul T. Shah, T. N. Vijaykumar and Jeffrey S. Vitter. “Performance Models and Systems Optimization for Disk-Bound Applications.” High End Computing University Research Activity NSF 06-503 (2006)

Wyckoff, Pete. "Applicability of Object-Based Storage Devices in Parallel File Systems."  
High End Computing University Research Activity NSF 06-503 (2006)

Zadok, Erez, Ethan L. Miller and Klaus Mueller. "File System Tracing, Replaying,  
Profiling, and Analysis on HEC Systems." High End Computing University Research  
Activity NSF 06-503 (2006)

## APPENDIX A (HECURA and SciDAC2 FSIO projects)

### HECURA I/O Projects

- Collaborative Research: Petascale I/O for High End Computing;
  - Maccabe, Arthur B/ Schwann, Karsten; UNM/ Georgia Tech Research Corporation – GA Inst of Tech
  - HECIWG topics: Metadata, Next generation I/O architectures
  - Keywords
    - Higher level I/O abstractions via I/O graphs
    - Flexible metadata management by metabots
    - Rich metadata
    - Lightweight file systems

#### Motivation

Data-intensive HPC applications are becoming increasingly important, adding substantial challenges to the already daunting input/output requirements of MPP codes. A well-known example is the interpretation of data from seismic exploration. In these applications, I/O problems occur both from the large data volumes produced by seismic sensing and from the fact that this data must be manipulated to fit simulation requirements for translating the time series data from multiple sensor locations into a format ready for 3-D subsurface reconstruction. Similarly, in online collaboration systems, visualizations require conversion and/or filtering to meet client needs.

#### Problem Statement and Solution Approach

The difficulties faced by scientists and engineers in attaining high performance I/O for data-intensive MPP applications are exacerbated by the low level of abstraction presented by current I/O systems. This research will create higher level I/O abstractions for developers. Specifically, the SSDS framework we propose models I/O as I/O Graphs that 'connect' application components with input or output mechanisms like file systems based on metadata constructed offline by autonomous metabots. SSDS enhances the I/O functionality available to end users in several ways. I/O Graphs can be programmed to realize application-specific I/O functionality, such as data filtering and conversion, data remeshing, and similar tasks. Their management is automated, including the mapping of their logical graph nodes to underlying physical MPP and distributed machine resources. I/O performance in SSDS will be improved by integrating the computational I/O actions of I/O Graphs with the backend file systems that store high volume data and with the I/O actions already taken by applications, and by moving metadata management offline into metabots.

The purpose of the new functionality inherent in SSDS is to help developers carry out complex I/O tasks. Technical topics to be addressed to realize this goal include the development of automated methods for deploying graph nodes to the physical sites that perform I/O functions, of dynamic management methods that maintain desired

levels of QoS for those I/O functions that require it (e.g., when accessing remote sensors). A key aspect of this work is the automation of I/O Graph creation and deployment. XML-based interfaces will make it easy for developers to provide information about the structure of I/O data, and to specify useful data manipulations. Efficient representations of metadata will enable both in-band and out-of-band data manipulation, to create I/O Graphs that best match current I/O needs and available machine resources. New offline techniques will derive metadata that can be used to enrich I/O graphs and more generally, meta-information about the large data volumes produced and consumed by MPP applications. Finally, this work will improve flexibility for I/O in future MPP machines, where virtualization techniques coupled with new chip (i.e., multicore) and interconnect technologies will make it easier to construct multi-use MPP platforms capable of efficiently performing both computational and I/O tasks.

The implementation of the SSDS system and its I/O Graph model will impact a substantial HPC user community, due to its planned integration with the Lightweight File System (LWFS) currently under development at Sandia National Laboratories (SNL). This file system and its SSDS extensions will be deployed on large-scale machines at Sandia to demonstrate scalability and application utility. SSDS will be integrated with the file formats and file systems used by other groups at Sandia and at Oak Ridge National Laboratories (ORNL) (with whom we are also collaborating). In fact, Georgia Tech and UNM have a long history of collaboration with SNL and ORNL. Finally, our team has been working with the vendors of future processor technology: with IBM as part of the PERCS project, and with Intel, to better understand the implications for the high performance domain of future processor virtualization techniques

- Collaborative Research: Techniques for Streaming File Systems and Databases;
  - Bender, Michael A/Farach-Colton, Martin; SUNY at Stony Brook/Rutgers Univ New Brunswick
  - HECIWG topics: Metadata, Next generation I/O architectures, and File System and related Communication Protocols
  - Keywords
    - Streaming B-trees and variants for efficient data layout on disk, databases

The performance of many high-end computing applications is limited by the capacity of memory systems to deliver data. As processor speeds increase, I/O performance continues to lag. Thus, I/O is likely to remain a critical bottleneck for high-end computing.

The researchers propose to address core problems on how to organize data on disk to optimize I/O, thus re-examining decades-old questions in the face of new applications, new technology, and new techniques. Specifically, the researchers propose to build prototypes of their streaming B-tree and variants for a file system or database. Streaming B-trees index and scan data at rates one-to-two orders of magnitude faster than traditional B-trees; they use cache-oblivious techniques to achieve platform independence. Several

issues remain to be addressed, specifically, how to deal with different-sized keys, how to support transactions, how to scale to multiple disks and processes, and how to provide O/S support for cache-obliviousness and memory-mapped massive data.

The proposed work represents a promising new direction for manipulating massive data and overcoming classic I/O bottlenecks. In HEC file systems and databases, this technology will permit rapid streaming of data onto and off of disks for high-throughput processing of data. This work will result in the transfer of recently developed algorithmic techniques to other areas of computer science, engineering, and scientific computing and is intended to transform how scientists and engineers manipulate massive data sets.

- Applicability of Object-Based Storage Devices in Parallel File Systems;
  - Wyckoff, Pete; Ohio State Univ Research Foundation; FSIO
  - HECIWG topics: Metadata, File System and related Communication Protocols, and Next generation IO architectures
  - Keywords
    - Objects trade-offs and attributes
    - Applicability of OSDs in parallel file systems
    - Metadata

While continued improvements in processing speeds and disk densities improve computing over time, the most fundamental advances come from changing the ways in which components interact. Delegating responsibility for some operations from the host processor to intelligent peripherals can improve application performance. Traditional storage technology is based on simple fixed-size accesses with little assistance from disk drives, but an emerging standard for object-based storage devices (OSDs) is being adopted. These devices will offer improvements in performance, scalability and management, and are expected to be available as commodity items soon.

When assembled as a parallel file system, for use in high-performance computing, object-based storage devices offer the potential to improve scalability and throughput by permitting clients to securely and directly access storage. However, while the feature set offered by OSD is richer than that of traditional block-based devices, it does not provide all the functionality needed by a parallel file system.

We will examine multiple aspects of the mismatch between the needs of a parallel file system, in particular PVFS2, and the capabilities of OSD. Topic areas include mapping data to objects, metadata, transport, caching and reliability. Trade-offs arise from the mapping of files to objects, and how to stripe files across multiple objects and disks, in order to obtain good performance. A distributed file system needs to track metadata that describes and connects data. OSDs offer automatic management of some critical metadata components that can be used by the file system. There are transport issues related to flow control and multicast operations that must be solved. Implementing client caching schemes and maintaining data consistency also requires proper application of OSD capabilities.

Our work will examine the feasibility of OSDs for use in parallel file systems, discovering techniques to accommodate this high performance usage model. We will also suggest extensions to the current OSD standard as needed.

- Collaborative Research: SAM<sup>2</sup> Toolkit: Scalable and Adaptive Metadata Management for High-End Computing;
  - Jiang, Hong/Zhu, Yifeng; Univ of Nebraska-Lincoln/Univ of Maine; SDDR
  - HECIWG topics: Metadata
  - Keywords
    - Scalable adaptive Metadata Management (SAM<sup>2</sup>) tools
    - Predictive metadata access patterns
    - Bloom filters for load balance and scalability
    - Adaptive cache coherence protocol for metadata caching
    - Decentralized metadata group schemes

The increasing demand for Exa-byte-scale storage capacity by high end computing applications requires a higher level of scalability and dependability than that provided by current file and storage systems. The proposal deals with file systems research for metadata management of scalable cluster-based parallel and distributed file storage systems in the HEC environment. It aims to develop a scalable and adaptive metadata management (SAM<sup>2</sup>) toolkit to extend features of and fully leverage the peak performance promised by state-of-the-art cluster-based parallel and distributed file storage systems used by the high performance computing community.

The project involves the following components: 1. Develop multi-variable forecasting models to analyze and predict file metadata access patterns. 2. Develop scalable and adaptive file name mapping schemes using the duplicative Bloom filter array technique to enforce load balance and increase scalability 3. Develop decentralized, locality-aware metadata grouping schemes to facilitate the bulk metadata operations such as prefetching. 4. Develop an adaptive cache coherence protocol using a distributed shared object model for client-side and server-side metadata caching. 5. Prototype the SAM<sup>2</sup> components into the state-of-the-art parallel virtual file system PVFS2 and a distributed storage data caching system, set up an experimental framework for a DOE CMS Tier 2 site at University of Nebraska-Lincoln and conduct benchmark, evaluation and validation studies.

- Improving Scalability in Parallel File Systems for High End Computing;
  - Ligon, Walter B; Clemson Univ; FSIO
  - HECIWG topics: Metadata, Management and RAS, and Next generation I/O architectures
  - Keywords
    - Active caching and buffering
    - Server to server and client to client communication
    - Autonomics
    - Scalable metadata
    - Small unaligned data access

- Reliability through redundancy

As high end computing systems (HECs) grow to several tens of thousands of nodes, file I/O is becoming a critical performance issue. Current parallel file systems such as PVFS2 and others, can reasonably stripe data across a hundred nodes and achieve good performance for bulk transfers involving large aligned accesses. Serious performance limits exist, however, for small unaligned accesses, metadata operations, and accesses impacted by the consistency semantics (any time one process writes data that is read by another).

The proposed research would address a few of these most critical issues through a straightforward application of engineering and research. The approach would build heavily on what is already known about similar problems in other distributed systems, especially distributed shared memory systems. These existing techniques would be studied in the new context of a parallel file system, adjusted, adapted, and where prudent, rejected for a novel approach. *The fundamental approach is to build quantitative evidence in support of each technique using analytical and simulation techniques, and to finally develop prototypes for PVFS2.* It is expected that the same techniques could be applied to any other parallel file system as well. A major focus would be on scalability as the key unit of evaluation. It is unclear if we would have the opportunity to test on a very large HEC, but we intend to simulate such machines and use machines we do have access to for validation of those simulations.

The issues we would study are *scalable metadata operations, small, unaligned data accesses, reliability through redundancy, and management of I/O resources.* Techniques we expect to employ include *active caching and buffering, server-to-server and client-to-client communication, and autonomies.* We intend to employ *middleware* whenever possible in order to enhance portability and control complexity. A major theme of the proposal is that file systems that provide everything all of the time are at a disadvantage in terms of scalable performance because features, like strict consistency and parity-based redundancy, are hard to implement with good scalability. *A file system that can configure itself to match the needs of the application can get the best performance possible.* Thus, PVFS2 was developed to allow a large degree of configurability, and the proposed research intends to enhance that file system so that it will scale to very large sizes.

- HECURA: The Server-Push I/O Architecture for High End Computing;
  - Sun, Xian-He; Illinois Inst of Tech
  - HECIWG topics: File systems and related Communication Protocols and Next generation I/O architectures
  - Keywords
    - Server side push
    - Collective I/O aware access patter prediction

Unlike traditional I/O designs where data is stored and retrieved by request, a new I/O architecture for High End Computing (HEC) is proposed based on a novel "Server-Push" model where a data access server proactively pushes data from a file server to the compute node's memory. The objective of this research is two fold: 1) increasing fundamental understanding of data access delay, 2) producing an effective I/O architecture that minimizes I/O latency. The PIs plan to increase the fundamental understanding through the study of data access pattern identification, prefetching algorithms, data replacement strategy, and extensive experimental testing. The PIs will

verify the performance improvement with their file server design for various critical I/O intensive applications by using a combination of simulation and actual implementation in the PVFS2 file system.

- Collaborative research: Scalable I/O Middleware and File System Optimizations for High-Performance Computing;
  - Choudhary, Alok N/Kandemir, Mahmut T; Northwestern Univ/Pennsylvania State Univ University Park
  - File Systems and related Communication Protocols, Next generation I/O architectures, and Measurement and Understanding
    - Middleware cache
    - Small I/O
    - Collective
    - New APIs
    - New benchmarks

This project entails research and development to address several parallel I/O problems in the HECURA initiative. In particular, the main goals of this project are to design and implement novel I/O middleware techniques and optimizations, parallel file system techniques that scale to ultra-scale systems, design and development of techniques that efficiently enable newer APIs and flexible I/O benchmarks that mimic real and dynamic I/O behavior of science and engineering applications. The fundamental premise is that, to achieve extreme scalability, incremental changes or adaptation of traditional techniques for scaling data accesses and I/O will not succeed because they are based on pessimistic and conservative assumptions of parallelism and interactions. We will develop techniques to optimize data accesses that utilize the understanding of high-level access patterns ("intent"), and use that information through middleware and file systems to enable optimizations. Specifically, the objectives are to (1) design and develop middleware I/O optimizations and cache system that are able to capture small, unaligned, irregular I/O accesses from large number of processors and uses access pattern information to optimize for I/O; (2) incorporate these optimizations in MPICH2's MPI-IO implementation to make them available to a large number of users; (3) design and evaluate enhanced APIs for file system scalability, and (4) develop flexible, execution oriented and scalable I/O benchmarks that mimic the I/O behavior of real science, engineering and bioinformatics applications.

- Collaborative Research: Application-adaptive I/O Stack for Data-intensive Scientific Computing;
  - Ma, Xiaosong/Sivasubramaniam, Anand/Zhou, Yuanyuan; North Carolina State Univ/Pennsylvania State Univ University Park/Univ of Illinois at Urbana-Champaign
  - HECIWG topics: Next Generation I/O Architectures and QOS
  - Keywords
    - Parallel Adaptive I/P (PATIO)
    - Multilevel cache/vertical layer caching and pre-fetching

- Access pattern recognition
- Tunable consistency semantics
- Content addressable storage
- Cache partitioning between multiple workloads
- Storage QoS

Advances in computational sciences have been greatly accelerated by the rapid growth of high-end computing (HEC) facilities. However, the continuous speedup of end-to-end scientific discovery cycles relies on the ability to store, share, and analyze the terabytes and petabytes of data generated by today's supercomputers. With the growing performance gap between I/O systems and processor/memory units, data storage and accesses are inevitably becoming more bottleneck-prone.

In this proposal, we address the I/O stack performance problem with adaptive optimizations at multiple layers of the HEC I/O stack (from high-level scientific data libraries to secondary storage devices and archiving systems), and propose effective communication schemes to integrate such optimizations across layers. In particular, our proposed PATIO (Parallel Adaptive I/O) framework explores multi-layer caching/prefetching that coordinates storage resources ranging from processors to tape archiving systems. This novel approach will bridge existing disjoint optimization efforts at each individual layer and responds to the critical call of improving the overall I/O system performance with increasingly deep HEC I/O stacks.

- Active Storage Networks for High End Computing;
  - Chandy, John A; Univ of Connecticut
  - HECIWG topics: Next Generation I/O Architectures
  - Keywords
    - Active storage networks-computation at the networks such as reductions and transformations

Recent developments in object-based storage systems and other parallel I/O systems with separate data and control paths have demonstrated an ability to scale aggregate throughput very well for large data transfers. However, there are I/O patterns that do not exhibit strictly parallel characteristics. For example, HPC applications typically use reduction operations that funnel multiple data streams from many storage nodes to a single compute node. In addition, many applications, particularly non-scientific applications, use small data transfers that can not take advantage of existing parallel I/O systems. In this project, we suggest a new approach called active storage networks (ASN) - namely putting intelligence in the network along with smart storage devices to enhance storage network performance. These active storage networks can potentially improve not only storage capabilities but also computational performance for certain classes of operations. The main goals of this project will include investigation of ASN topologies and architectures, creation of ASN switch from reconfigurable components, studying HEC applications for ASNs, protocols to support programmable active storage network functions, and storage system optimizations for ASNs.

- Active Data Systems;
- Reddy, A.L. Narasimha; Texas Engineering Experiment Station

- HECIWG topics: Next generation I/O architectures and QoS
- Keywords
  - Broadening active disk applicability by examining running multiple applications at disk concurrently
  - Scheduling
  - Security
  - Sharing

This project plans to address several issues related to broadening the practicality of active storage. More specifically, this project plans to study and investigate:

(1) The impact of mixed workloads (both active and normal requests) at the active devices. (2) The impact of multiple active applications at the active devices. (3) The resource scheduling and QOS policies for a diverse set of workloads. (4) The impact of intelligent allocation in active storage systems.

In order to address these issues, the project plans to develop (a) an "active data" model to allow flexible processing of data, either at devices or at the requester. (b) QOS algorithms and security mechanisms for mixed workloads. (c) Algorithms and prototypes for exploiting the nature of data to develop content-based active storage.

- Quality of Service Guarantee for Scalable Parallel Storage Systems;
  - Chiueh, Tzi-Cker; SUNY at Stony Brook; SSSDR
  - HECIWG topics: Next generation I/O architectures, Measurement and Understanding, and QoS
  - Keywords
    - Platypus – storage system
    - QoS trace replay
    - Bandwidth guarantees
    - Prefetching using decoupled architecture by extracting a prefetch thread from the computation thread

The Platypus project will develop a parallel I/O system that supports guaranteed storage QoS for concurrently running parallel applications while maximizing the parallel storage system's utilization efficiency. In addition, it will implement a timing-accurate parallel trace play-back tool to evaluate the effectiveness and efficiency of the proposed parallel I/O system

- Concurrent I/O Management for Cluster-based Parallel Storages;
  - Shen, Kai; Univ of Rochester
  - HECIWG topics: Next generation I/O architectures
  - Keywords
    - Concurrent I/O workload
    - Disk seek/spin reduction by prefetching and anticipatory I/O scheduling
    - Server level coscheduling
    - Load adaptive parallel data aggregation

High-end parallel applications that store and analyze large scientific datasets demand scalable I/O capacity. One recent trend is to support high-performance parallel I/O using

clusters of commodity servers, storage devices, and communication networks. When many processes in a parallel program initiate I/O operations simultaneously, the resulted concurrent I/O workloads present challenges to the storage system. At each individual storage server, concurrent I/O may induce frequent disk seek/rotation and thus degrade the I/O efficiency. Across the whole storage cluster, concurrent I/O may incur synchronization delay across multiple server-level actions that belong to one parallel I/O operation.

This project investigates system-level techniques to efficiently support concurrent I/O workloads on cluster-based parallel storages. Our research will study the effectiveness of I/O prefetching and scheduling techniques at the server operating system level. We will also investigate storage cluster level techniques (particularly co-scheduling techniques) to support better synchronization of parallel I/O operations. In parallel to developing new techniques, we plan to develop an understanding on the performance behavior of complex parallel I/O systems and explore automatic ways to help identify causes of performance anomalies in these systems.

- Performance Models and Systems Optimization for Disk-Bound Applications;
  - Thottethodi, Mithuna S; Purdue Univ
  - HECIWG topics: Next generation I/O architectures, Measurement and Understanding, and File System and related Communications Protocols
  - Keywords
    - Disk array modeling/algorithms
    - Network aware placement and migration
    - Power and thermal optimization via entropy-aware disk caching

Despite many recent breakthroughs in the understanding and optimization of data-intensive applications and disk-array-based systems, significant challenges remain in system modeling, algorithm design, and performance optimization. Existing analytical models do not incorporate application characteristics, internal disk behavior, and I/O interconnection network contention; these shortcomings cause two key problems. First, optimization opportunities are lost since designers are compelled to design for the worst case rather than for specific application characteristics that may be significantly more benign. We propose an application characterization-driven approach wherein the behavior of the application (e.g., entropy, locality) shapes the optimization decisions. Second, inaccurate models may lead to wasted design effort because of differences between model-predicted performance and actual disk-array performance. We propose a unified and flexible disk-array access model that improves accuracy by accounting for (a) the contention on the interconnection network between disks and memory and (b) internal disk behavior. We propose to develop and distribute an integrated execution-driven simulation environment that incorporates all the individual components described above. We envision that the insights from our models and simulator will lead to a range of optimizations such as network-contention-aware data placement and migration policies, improved caching and pre-fetching policies and techniques to ameliorate power and thermal problems in large disk arrays.

- Exploiting Asymmetry in Performance and Security Requirements for I/O in High-end Computing;
  - Sivasubramaniam, Anand; Pennsylvania State Univ University Park
  - HECIWG topics: Security
  - Keywords
    - Data Vault – security
    - Tunable tradeoff between security and performance for site-specific policies
    - Visualization dashboard

Application sciences are more collaborative, with sharing of data sets becoming prevalent not just between users/applications of a single organization, but across organizations as well placing even higher performance requirements on the storage system. Given the sensitive nature of many of these applications, in addition to the performance demands, there is an impending need to secure such data from adversarial attacks. The consequences of security breaches can have far reaching consequences, over and beyond the costs of detecting and investigating such breaches. At the same time, one cannot fully confine the data physically since these need to be shared by collaborative applications from different administrative domains. Regulations are also mandating the maintenance of audit records and provenance of data.

The motivation for our DataVault project is driven by the need to secure storage systems which cater to the demands of high-end applications, while meeting their stringent performance requirements. Rather than have a one-solution-fits-all approach, we propose to investigate the rich design space - threats, storage architecture, enforcement mechanism, performance – to offer insightful choices that can be useful when deploying/customizing storage systems. DataVault will also include a usable objective-driven policy interface to configure the system for a given set of security and performance needs, while offering a convenient visualization dashboard for security management.

- Integrated Infrastructure for Secure and Efficient Long-Term Data Management;
- Odlyzko, Andrew; Univ of Minnesota-Twin Cities
  - HECIWG topics: Security, Archive
  - Keywords
    - Security
    - Hierarchical cluster-based archive
    - Long-term key management

To achieve the level of security and privacy for enterprise data that is increasingly required by laws or industry standards, data should be encrypted both at rest and in transit. Yet, numerous recent privacy breaches through loss or theft of archival tapes or notebook computers show that today most data, even of extremely sensitive nature, is not encrypted. The main reason is that we do not have a flexible system for key management. Loss of the encryption key (through lapses of memory, death of staff members, or destruction of stored copies) would mean that the owner of the data would effectively lose it completely, with potentially catastrophic consequences.

This project will develop a high-performance long-term data management system that will ensure the necessary levels of security throughout the lifecycle of a data set. The goal is a hierarchical cluster-based archival storage solution that will provide: (i) transparent backup, restore, and data access operations that will allow individual application programs and business entities to securely and efficiently archive data for decades; (ii) high-performance data access in a cluster computing environment; and (iii) innovative techniques for efficiently insuring long-term data security and accessibility, including long-term key management. The solution will be suitable for heterogeneous computing environments, including the extremely high-throughput ones of the high-performance computing (HPC) community.

- Formal Failure Analysis for Storage Systems;
  - Arpaci-Dusseau, Remzi H; Univ of Wisconsin-Madison
  - HECIWG topics: Management and RAS and Measurement and Understanding
  - Keywords
    - Formal analysis of failures with Wisconsin's Program Analysis of Storage Systems (PASS) program

Building scalable storage systems requires robust tolerance of the many faults that can arise from modern devices and software systems. Unfortunately, many important storage systems handle failure in a laissez-faire manner. In this proposal, we describe the Wisconsin Program Analysis of Storage Systems project (PASS), wherein we seek to develop the techniques needed to build the high-end, scalable, robust storage systems of tomorrow. Our focus in PASS is to bring a more formal approach to the problem, utilizing programming language tools to build, analyze, test, and monitor these storage systems. By applying these techniques, we will raise the level of trust in the failure-handling capabilities of high-end storage systems by an order of magnitude.

The PASS project will change the landscape of storage systems in three fundamental ways. First, by developing more formal failure analysis techniques, we will be able to uncover a much broader range of storage system failure-handling problems. Second, within PASS we will develop more robust and scalable testing infrastructure; such a framework will be of general use to the development of any future storage system. Finally, through run-time instrumentation of a large Condor cluster, we plan to gather information as to what types of faults occur in practice as well as how they manifest themselves as failures. Such data will be invaluable to future designs and implementations of robust, scalable storage systems.

- Toward Automated Problem Analysis of Large Scale Storage Systems;
  - Narasimhan, Priya; Carnegie-Mellon Univ
  - HECIWG topics: Measurement and Understanding and Management and RAS
  - Keywords
    - Continuous performance and anomaly tracing
    - Auto blame assignment and performance diagnosis
    - Automated analysis of failure and performance degradation

This research explores methodologies and algorithms for automating analysis of failures and performance degradations in large-scale storage systems. Problem analysis includes such crucial tasks as identifying which component(s) misbehaved, likely root causes, and supporting evidence for any conclusions. Automating problem analysis is crucial to achieving cost-effective storage at the scales needed for tomorrow's high-end computing systems, whose scale will make problems common rather than anomalous. Moreover, the distributed software complexity of such systems make by-hand analysis increasingly untenable.

Combining statistical tools with appropriate instrumentation, the investigators hope to significantly reduce the difficulty of analyzing performance and reliability problems in deployed storage systems. Such tools, integrated with automated reaction logic, also provide an essential building block for the longer-term goal of self-healing. The research involves understanding which statistical tools work and how well in this context for the problems of problem detection/prediction, identifying which components need attention, finding root causes, and diagnosing performance problems. It will also involve quantifying the impact of instrumentation detail on the effectiveness of those tools so as to guide justification for associated instrumentation costs. Explorations will be done primarily in the context of the Ursa Minor/Major cluster-based storage systems via fault injection and analysis of case studies observed in its deployment.

- File System Tracing, Replaying, Profiling, and Analysis on HEC Systems;
  - Zadok, Erez; SUNY at Stony Brook
  - HECIWG topics: Measurement and Understanding, Next generation I/O architectures, and File System and related Communication Protocols
  - Keywords
    - Visualization
    - Tracing and replaying file system activity

File systems are difficult to analyze, as they are affected by OS internals, hardware used, device drivers, disk firmware, networking, and applications. Traditional profiling systems have focused on CPU usage, not on I/O latencies. Worse, existing tools for profiling, analysis, and visualization are too simplistic, cannot cope with massive and complex data streams, and do not scale to large clusters. We have expertise in single-host file system tracing, replaying, profiling, and benchmarking---as well as having developed over 20 file systems; large data analysis and visualization; and designing and implementing petabyte-size storage clusters.

In this project we are developing tools and techniques that will work on large clusters and scale well. We are conducting large scale tracing and replaying, collecting vital information useful to analyze the cluster's performance given a specific application. We use automated and user-driven feedback to raise or lower the level of tracing on individual cluster nodes to (1) ``zoom in" on hot-spots and (2) trade off information accuracy vs. overheads. We use advanced data analysis techniques to identify performance bottlenecks, and we will visualize them for cluster users for ease of analysis. The end goal is to help identify I/O bottlenecks in running distributed

applications, so as to improve their performance significantly---resulting in more effective use of these expensive clustering resources by scientists worldwide.

- End-to-End Performance Management for Large Distributed Storage;
  - Brandt, Scott A; UC-Santa Cruz
  - HECIWG topics: QoS
  - Keywords
    - QoS server side
    - Server I/O scheduling
    - Server and client cache management
    - Client-to-server network flow control
    - Client-to-server connection management

End-to-end Performance Management for Large Distributed Storage Scott Brandt, Darrell Long, and Carlos Maltzahn, UC Santa Cruz Richard Golding and Theodore Wong, IBM Almaden Research Center.

Storage systems for large and distributed clusters of compute servers are themselves large and distributed. Their complexity and scale make it hard to manage these systems and, in particular, to ensure that applications using them get good, predictable performance. At the same time, shared access to the system from multiple applications, users, and internal system activities leads to a need for predictable performance.

This project investigates mechanisms for improving storage system performance in large distributed storage systems through mechanisms that integrate the performance aspects of the path that I/O operations take through the system, from the application interface on the compute server, through the network, to the storage servers. We focus on five parts of the I/O path in a distributed storage system: I/O scheduling at the storage server, storage server cache management, client-to-server network flow control, client-to-server connection management, and client cache management.

## SciDAC2 I/O Projects

### PetaScale Data Storage Institute

Garth Gibson (Lead PI) - Carnegie Mellon University  
Evan Felix - Pacific Northwest National Laboratory  
Gary Grider – Los Alamos National Lab  
Peter Honeyman - University of Michigan at Ann Arbor  
William Kramer - Lawrence Berkeley National Laboratory/NERSC  
Darrell Long - University of California at Santa Cruz  
Philip Roth - Oak Ridge National Laboratory  
Lee Ward – Sandia National Lab

- Leveraging experience in applications and diverse file and storage systems expertise of its members, the institute will enable a group of researchers to collaborate extensively on developing requirements, standards, algorithms, and development and performance tools.

Petascale computing infrastructures for scientific discovery make petascale demands on information storage capacity, performance, concurrency, reliability, availability, and manageability. The last decade has shown that parallel file systems can barely keep pace with high performance computing along these dimensions; this poses a critical challenge when petascale requirements are considered. This proposal describes a Petascale Data Storage Institute that focuses on the data storage problems found in petascale scientific computing environments, with special attention to community issues such as interoperability, community buy-in, and shared tools. Leveraging experience in applications and diverse file and storage systems expertise of its members, the institute allows a group of researchers to collaborate extensively on developing requirements, standards, algorithms, and development and performance tools. Mechanisms for petascale storage and results are made available to the petascale computing community. The institute holds periodic workshops and develops educational materials on petascale data storage for science.

The Petascale Data Storage Institute is a collaboration between researchers at Carnegie Mellon University, National Energy Research Scientific Computing Center, Pacific Northwest National Laboratory, Oak Ridge National Laboratory, Sandia National Laboratory, Los Alamos National Laboratory, University of Michigan, and the University of California at Santa Cruz.

The Institute's work will be organized into six projects:

- Petascale Data Storage Outreach: (Type: Dissemination) Development and deployment of training materials, both tutorials for scientists and course materials for graduate students; support and advise other SciDAC projects and institutes; and development of frequent workshops drawing together experts in the field and petascale science users.

- Protocol/API Extensions for Petascale Science Requirements: (Type: Dissemination) Drive deployment of best practices for petascale data storage systems through development and standardization of application programmer interfaces and protocols, with specific emphasis on Linux APIs. Validate and demonstrate these APIs in large scale scientific computing systems.

- Petascale Storage Application Performance Characterization: (Type: Data Collection) Capture, characterize, model and distribute workload, access trace, benchmark and usage data on terascale and projected petascale scientific applications, and develop and distribute related tools.

- Petascale Storage System Dependability Characterization: (Type: Data Collection) Capture, characterize, model and distribute failure, error log and usage data on terascale and projected petascale scientific systems, and develop and distribute related tools.
- Exploration of Novel Mechanisms for Emerging Petascale Science Requirements: (Type: Exploration) In anticipation of petascale challenges for data storage, explore novel mechanisms such as global/ WAN high performance file systems based on NFS; security aspects for federated systems, collective operations, and ever higher performance systems; predictable sharing of high performance storage by heavy storage load applications; new namespace/search and attribute definition mechanisms for ever large namespaces; and integration and specialization of storage systems for server virtualization systems.
- Exploration of Automation for Petascale Storage System Administration: (Type: Exploration) In anticipation of petascale challenges for data storage, explore and develop more powerful instrumentation, visualization and diagnosis methodologies; data layout planning and access scheduling algorithms; and automation for tuning and healing configurations.

## **Scientific Data Management Center for Enabling Technologies**

Arie Shoshani (PI), Doron Rotem, Lawrence Berkeley National Laboratory  
 Rob Ross, Bill Gropp, Rajeev Thakur, Argonne National Laboratory  
 Terence Critchlow, Chandrika Kamath, Lawrence Livermore National Laboratory  
 Nagiza Samatova, Jeff Vetter, Oak Ridge National Laboratory  
 Jarek Nieplocha, Pacific Northwest National Laboratory  
 Alok Choudhary, Wei-Keng Liao, Northwestern University  
 Mladen Vouk, North Carolina State University  
 Steve Parker, University of Utah  
 Bertram Ludaescher, University of California at Davis  
 Ilkay Altinas, San Diego Supercomputer Center

Managing scientific data has been identified as one of the most important emerging needs by the scientific community because of the sheer volume and increasing complexity of data being collected. Effectively generating, managing, and analyzing this information requires a comprehensive, end-to-end approach to data management that encompasses all of the stages from the initial data acquisition to the final analysis of the data.

Based on the community input, we have identified three significant requirements. First, more efficient interactions with disks and the resulting files are needed. In particular, parallel file system improvements are needed to write and read large volumes of data without slowing a simulation, analysis, or visualization engine. Second, scientists require improved access to their data, in particular the ability to effectively perform complex data

analysis and searches over large data sets. Specialized feature discovery and statistical analysis techniques are needed before the data can be understood or visualized. Finally, generating the data, collecting and storing the results, data post-processing, and analysis of results is a tedious, fragmented process. Tools for automation of these workflows this process in a robust, tractable, and recoverable fashion are required to enhance scientific exploration.

We have organized our activities in three layers abstracting the end-to-end data flow described above: the Storage Efficient Access (SEA), Data Mining and Analysis (DMA), and Scientific Process Automation (SPA) layers. The SEA layer is immediately on top of hardware, operating systems, file systems, and mass storage systems, and provides parallel data access technology. On top of the SEA layer exists the DMA layer, consisting of indexing, feature selection, and parallel statistical analysis. The SPA layer, which is on top of the DMA layer, provides the ability to compose scientific workflows from the components in the DMA layer as well as application specific modules. Together these layers provide an integrated system for data management in computational science.

## **APPENDIX B (HECIWG FSIO 2006 attendees)**

Philip Andrews  
UC San Diego

John A. Chandy  
University of Connecticut

Remzi Arpaci-Dusseau  
Univ of Wisconsin

Albert Cheng  
The HDF Group

Dave Aune  
Seagate

Alok N. Choudhary  
Northwestern University

Almadena Chtchelkanova  
National Science Foundation

Lee Ward  
Sandia National Laboratories

Michah Beck  
University of Tennessee

Tzicker Chuiueh  
Stony Brook University

Michael A. Bender  
Stony Brook University

Peter Corbett  
Network Appliance

John Bent  
Los Alamos National Laboratory

Harriet G. Coverston  
Sun Microsystems

Scott Brandt  
University of California, Santa Cruz

Carolyn M. Davenport  
Los Alamos National Laboratory

Roger D. Chamberlain  
Washington University

Cory Devor  
University of Minnesota

David Du  
National Science Foundation

William Miner  
ITRD

David Ellis  
LSI Logic

Gary Grider  
Los Alamos National Laboratory

Bryan W. Embry  
Dept. of Defense

Robert J. Grzybowski  
National Reconnaissance

Rob Farber  
Battelle/PNNL

Johnathan Hall  
Sherwood Information

Evan Felix  
Battelle/PNNL

Arthur B. McCabe  
UNM

Mike J. Folk  
The HDF Group

Thuc Hoang  
DOE/NNSA

Greg Ganger  
Carnegie Mellon University

Wynona Holton  
Los Alamos National Laboratory

Garth A. Gibson  
Carnegie Mellon University

Peter Honeyman  
University of Michigan (CITI)

Richard Golding  
IBM Almaden

Hank Jakiela  
Hewlett-Packard

Kim Goodwin  
Army Research

Hong Jiang  
University of Nebraska

Fred Johnson  
DOE Office of Science

Tina Macaluso  
SAIC

Karsten Schwan  
Georgia Tech University

Arthur B. Maccabe  
University of New Mexico

Stephen Kenmurian  
The MITRE Corporation

Kandemir Mahmut  
Penn State University

Jeremy V. Kepner  
MIT Lincoln Laboratory

Carlos Maltzahn  
University of California Santa Cruz

Yongdae Kim  
University of Minnesota

Brent B. Welch  
Panasas

Ben Kobler  
NASA Goddard Space Flight

Randal H. Melen  
Stanford Linear Accelerator

Bradley C. Kuszmaul  
MIT CSAIL

Mike Mesnier  
Intel/CMU

Thomas W. Lanzatella  
Symantec Corporation

Ethan L. Miller  
University of California

Walter Ligon  
Clemson University

Henry Scott Newman  
Instrumental Inc.

William Edward Loewe  
Lawrence Livermore

Paul Nowoczynski  
Pittsburgh Supercomputing

James A. Nunez  
Los Alamos National Laboratory

Ellen M. Salmon  
NASA

Dhabaleswar K. Panda  
Ohio State University

Julian Satran  
IBM Research Laboratory

Narasimha Reddy  
Texas A&M University

Bianca Schroeder  
Carnegie Mellon University

Erik Riedel  
Seagate Technology

Jun Wang  
University of Central Florida

Rob Ross  
Argonne National Laboratory

Anand Sivasbramianam  
Penn State University

Phillip Roth  
Oak Ridge National Laboratory

T. Basil Smith  
IBM Watson Research

Andy Rudoff  
Cluster Filesystems, Inc.

Xian-He Sun  
Illinois Institute of Technology

Daniel E. Russ  
National Institutes of Health

Judith E. Terrill  
NIST

Thomas M. Ruwart  
I/O Performance, Inc.

Rajeev S. Thakur  
Argonne National Laboratory

Subhash Saini  
NASA Ames Research

John Thorp  
Los Alamos National Lab

Mithuna Thottethodi  
Purdue University

Yuanyuan Zhou  
University of Illinois

Tai-Ching Tuan  
US Army Human Resources

Yifeng Zhu  
University of Maine

Jeffrey S. Vetter  
Oakridge National Laboratory

Ronald J. Vetter  
University of North Carolina

Peggy Vigil  
Los Alamos National Laboratory

Pete Wyckoff  
Ohio Super Computer Center

Murali N. Vilayannur  
Argonne National Laboratory

Erez Zadok  
Stony Brook University

**APPENDIX C (Inter-Agency HPC FSIO R&D Needs Document)**

**HPC File Systems and Scalable I/O: Suggested Research and Development Topics for the fiscal 2005-2009 time frame**

**DOE Office of Science**

**Rob Ross ANL**

**Evan Felix PNL**

**DOE NNSA**

**Bill Loewe LLNL**

**Lee Ward SNL**

**Gary Grider LANL**

**DOD**

**Rob Hill, NSA**

Executive Summary .....	74
Background .....	76
Purpose.....	76
Frequently used terms .....	77
Historical perspective.....	78
The early 1990's .....	78
The mid 1990's .....	78
The late 1990's to present.....	79
Key areas of possible future research, development, and standards work.....	83
Conclusion .....	93

## Executive Summary

The need for immense and rapidly increasing scale in scientific computation drives the need for rapidly increasing scale in storage for scientific processing. Individual storage devices are rapidly getting denser while bandwidth is not growing at the same pace. In the past several years, Research and Development (R&D) into highly scalable file systems, high level I/O libraries, and I/O middleware was done to provide some solutions to the problems that arise from massively parallel storage. This document primarily concentrates on file systems and I/O middleware since high level I/O libraries have been addressed in many Data Management discussions and calls for R&D. The purpose of this document is to present areas of needed research and development in the HPC file systems and scalable I/O area which should be pursued by the government.

In the last five years, supercomputers with thousands of nodes and over ten thousand processors have been deployed. Additionally, thousands of small clusters have been deployed worldwide. File systems and I/O has come a long way since the simple cross mounted NFS that was used with early clusters. File systems and I/O middleware have been developed and deployed for these supercomputers and clusters systems which have enabled bandwidths beyond ten gigabytes/sec and metadata performance beyond one thousand file inserts/sec. It is now possible to scale bandwidth, and there is competition in the scalable global parallel file systems market space through products that span the gamut from completely proprietary to open source. I/O Middleware is maturing which is enabling many applications to get good performance from the underlying file systems.

Recently several new I/O R&D efforts have begun to try to address future needs in HPC file systems and I/O middleware. Efforts in the areas of:

- Relaxation of POSIX Semantics for parallelism
- Scalable metadata operations in a single directory
- NFSv4 security and the pNFS effort to allow NFS to get more native file system performance through separation of data and control which enables parallelism
- I/O Middleware enhancements to enable dealing with small, overlapped, and unaligned I/O
- Tighter integration between high level I/O libraries and I/O middleware
- Initial autonomic storage management
- Sharing of global parallel file systems between multiple clusters
- Initial active storage research

In the near future, sites will deploy supercomputers with tens of thousands or even hundreds of thousands of processors. Immense bandwidth, metadata, security, and management needs are emerging. Work flows for efficient complex science will begin to approach the exabyte range, and the ability to handle a more varied I/O workload including small to extremely large I/O operations, extremely high metadata activities, and multiple simultaneous workloads will be required. File systems will be so large that complete checks or rebuilds, entire tree walks, and other large operations will not be able to be contemplated. Management of these large storage systems will become increasingly difficult.

To meet the demands posed by the future HPC environments, investment in R&D and standards work need to be undertaken. The following are key areas to consider investment in:

- Scaling of metadata , security, reliability, availability, and management to deal with the enormity of future systems
- Enhancements in the POSIX I/O API and I/O middleware in the areas of ordering, coherence, alternate metadata operations, shared file descriptors, locking schemes, and portability of hinting for layouts and other information.
- Support for machine architectures which do not have full operating systems on each node and exploitation of hierarchies of node types
- Additional work in active storage concepts, including use in the LAN/SAN, WAN, application integration, and object archives
- Continued support for development and standards work on NFSv4 and pNFS
- Tracing, Simulation, and benchmarking, including application realistic benchmarking and simulation
- New metadata layouts other than tree based directories

More focused and complete government investment needs to be made in the file systems and I/O middleware area of HPC, given its importance and its lack of sufficient funding levels in the past, compared to other elements of HPC. Scalable I/O is perhaps the most overlooked area of HPC R&D, and given the information generating capabilities being installed and contemplated, it is a mistake to continue to neglect this area of HPC. Many areas in need of new and continued investment in R&D and standardization in this crucial HPC I/O area have been summarized in this document.

## **Background**

The need for immense and rapidly increasing scale in scientific computation is well understood and documented. To keep up with the need for immense scaling, Moore's Law, which maps how individual processors get faster, and the idea of ever increasing parallelism are combined. In providing storage for scientific processing, similar and well documented concepts apply. It is well known that individual storage devices are getting denser at an amazing rate, keeping up with the ever faster speeds of processors. It is also well known that bandwidth to/from individual storage devices is getting faster but at an alarmingly slower rate than density of the devices. To deal with the ever increasing need for bandwidth to storage, massive parallelism is required.

In the past several years, Research and Development (R&D) into highly scalable file systems and I/O, was done to provide some solutions to the problems that arise from massively parallel storage. The I/O software stack is primarily composed of 4 basic layers: higher level I/O libraries, I/O Middleware, file systems, and storage devices. Higher level I/O libraries, for the most part, provide the needed high level abstraction for application programmers who are familiar with the science involved in the application. I/O Middleware, for the most part, provide abstractions that are computer science based, and deal with distributed/parallel nature of the memories involved and the distributed/parallel access to the storage devices. File systems, of course, provide the standard interfaces for organization, storage and retrieval of data and deal with coordination of the storage devices. The goal of the I/O software stack is to maintain performance and correctness from the high level I/O interface all the way to the storage devices and to fully leverage innovations in storage devices and the connectivity to those devices.

We define HPC file systems and scalable I/O as a subset of the overall scientific data management function, that subset of the I/O stack dealing with the storage devices, networks to reach those storage devices, data movement and metadata protocols, file systems, and I/O Middleware functions.

## **Purpose**

The purpose of this document is to present areas of needed research and development in the HPC file systems and scalable I/O area which should be pursued. This document does not heavily delve into R&D needs in the high level I/O libraries area, because that topic is better addressed by Scientific Data Management (SDM) experts and scientific application designers and developers, although it is understood that coordination with high level I/O libraries is necessary and good. Nor does this document address R&D needs for advancement of physical storage devices. The R&D at the storage device level is largely driven by the needs of the multi-billion dollar storage industry and is done at a scale that is well beyond the proposed R&D investments in this document. The focus area of this document is the I/O Middleware and file system layers of the I/O software stack, primarily, with some extension upwards into high level I/O libraries and down into storage devices.

Additionally, this document does not address breakthrough storage technologies that would fundamentally change I/O paradigms. This document assumes an evolution of storage devices with no extremely disruptive technologies. While there are new storage technologies nearly ready to enter the market place like Micro-Electro-Mechanical Systems (MEMS) (<http://www.memsnet.org/mems/what-is.html>) and Magnetic Random Access Memory (MRAM) (<http://computer.howstuffworks.com/mram.htm>), given the market drivers behind these technologies, for the HPC market, it is very unlikely these technologies will be disruptive enough, in the next several years, to cause radical or wholesale changes in the I/O stack. While it is important to remain vigilant in exploiting new evolutionary technologies and watching for disruptive technologies, this document only addresses evolutionary technologies and ideas.

This document is also based on the idea that high end supercomputing will continue to be based on physically distributed memories. Much of the I/O software stack assumes and deals with this current distributed memory reality.

## Frequently used terms

*I/O* – input/output

*File system* – A combination of hardware and software that provides applications access to persistent storage through an application programming interface (API), normally the Portable Operating System Interface (POSIX) for I/O.

*POSIX* - Portable Operating System Interface (POSIX), the standard user interfaces in the UNIX based and other operating systems

(<http://web.access.net.au/felixadv/files/output/book/x1164.html>)

*Global* – refers to accessible globally (by all), often implies all who access see the same view

*Parallel* – multiple coordinated instances, such as streams of data, computational elements

*Scalable* – decomposition of a set of work into an arbitrary number of elements, the ability to subdivide work into any number of parts from 1 to infinity

*Metadata* – information that describes stored data, examples are location, creation/access dates/times, sizes, security information, etc.

*Higher level I/O library* – software libraries that provide applications with high level abstractions of storage systems, higher level abstractions than parallelism, examples are the Hierarchical Data Formats version 5 library (HDF5) ([http://www-rcd.cc.purdue.edu/~aai/HDF5/html/RM\\_H5Front.html](http://www-rcd.cc.purdue.edu/~aai/HDF5/html/RM_H5Front.html)) and parallel Network Common Data Formats library (PnetCDF) ([http://www-unix.mcs.anl.gov/parallel-netcdf/sc03\\_present.pdf](http://www-unix.mcs.anl.gov/parallel-netcdf/sc03_present.pdf))

*I/O Middleware* – software that provide applications with higher level abstractions than simple strings of bytes, an example is the Message Passing Interface – I/O library (MPI-IO) (<http://www-unix.mcs.anl.gov/romio>)

*WAN* – Wide Area Network, refers to connection over a great distance, tens to thousands of miles

*SAN* – Storage Area Network, network for connecting computers to storage devices

## Historical perspective

To put into context the rationale for the proposed set of research and development topics, a high level summary of more than the last half decade of HPC file systems and Scalable I/O related research and development is presented here.

### ***The early 1990's***

In the early 1990's, most cluster based supercomputers were relatively small by today's standards, typically with tens of nodes, and for the most part used Network File System (NFS) servers for both home (program/source code/etc.) and scratch space. Sometimes only one NFS server was used, but frequently multiple cross mounted NFS servers were used. Given the small nature of these clusters, this approach provided sufficient performance and acceptable access. Given the primitive and limited scale of the applications in this time frame, parallel access methods to I/O typically were not needed.

### ***The mid 1990's***

In the mid 1990's, large scale cluster based supercomputers, hundreds to a few thousand nodes, began to show up. In order to serve the storage bandwidth needs of these clusters, hundreds of redundant array of independent disks (RAID) controllers with over a thousand individual disks used in a coordinated manner were required. Given that the Linux cluster phenomenon had not occurred yet, these clusters were for the most part proprietary operating system based clusters. NFS or cross mounted NFS was simply not an option on these machines due to their need for scalable bandwidth and manageability at scale. The file systems used on these clusters were first generation client/server based proprietary cluster file systems. Examples include IBM's General Purpose File System (GPFS) ([http://www.almaden.ibm.com/StorageSystems/file\\_systems/GPFS/](http://www.almaden.ibm.com/StorageSystems/file_systems/GPFS/)), Meiko's Parallel File System (Meiko PFS), and Intel's PFS (Intel PFS). Additionally, due to the large scale parallelism in the clusters of this era, middleware libraries like the MPI-IO library to enhance parallel access were begun. Projects to develop Storage Area Network (SAN) based file systems, which relied, at the time, on every file system client having direct access to the storage devices, typically via a fibre channel SAN, were also begun. Examples of SAN approaches at that time are the University of Minnesota/Sistina Global File System (GFS) (<http://www.redhat.com/software/rha/gfs/>), the Silicon Graphic Incorporated (SGI) Clustered-XFS (C-XFS) (<http://www.sgi.com/products/storage/cxfs.html>), the Advanced Digital Information Corporation (ADIC) Clustered Virtual File System (CVFS) (<http://www2.adic.com/stornext/>). These SAN file systems were primarily intended for use in smaller node count clusters (tens to a few hundred) due to the difficulty and expense in building large secure Fibre Channel SAN's. Fibre Channel SAN's lack the ability to share data at a granularity smaller than a volume (disk or virtual disk) and all nodes/machines that have direct access to the SAN that are sharing a volume have root style access to that entire volume, so there is no concept of sharing at the file or object level with authentication based security. Another concept that took shape in this timeframe was the use of an old idea, the separation of metadata activity from data

movement activity. Some file systems deployed this concept through the use of separate metadata servers and others shared the metadata responsibility among the file systems clients through the use of lock mechanisms.

### ***The late 1990's to present***

In the late 1990's to the present, cluster supercomputers are now routinely in the thousands of nodes with tens of thousands being contemplated. Given the slow improvement of individual disk device bandwidth improvement compared to processor speed increases, the number of individual disk devices needed to be coordinated to achieve the needed bandwidth for these super clusters is currently several thousand. Open source Linux clusters are common place, implying that other, more open and Linux based, solutions for global parallel file systems are needed and past proprietary solutions for global parallel file systems are diminished in value.

Both SAN based and client server file systems have grown in popularity as clustered computing has become more prevalent in thousands of computing centers worldwide. Pure SAN file systems, again, where all file system clients have access to the disk devices, still suffer from scalability for very large clusters due to the difficulty in building scalable Fibre Channel SANs, also have security issues due to clients having direct access to storage. Most SAN based file systems have begun to offer ways to extend access to beyond clients that have direct access the disk devices through a variety of mechanisms like SCSI protocol over IP (iSCSI), gateway functions, and others. In part, due to government sponsored R&D, new and less proprietary client server file systems have become popular. Examples include: the DOE Office of Science/Argonne National Laboratory/Clemson Parallel Virtual File System (PVFS) (<http://www.parl.clemson.edu/pvfs/desc.html>), the DOE/NNSA/tri-labs/HP/CFS/Intel Lustre file system (<http://www.lustre.org>), and the Panasas PanFS (<http://www.panasas.com>) file system. These new and less proprietary client server file systems leverage heavily separation of data and control, abstracting the storage device functions like allocation and data-write operations for less than full block updates away from the file system proper.

Recently the ANSI T10/1355-D Object based Storage Devices (OSD) (<http://www.t10.org/ftp/t10/drafts/osd/osd-r10.pdf>) standard was introduced, putting the industry on a course to standardize the interface to storage devices that provide allocation, read-update-write on partial block writes, and transactional security for the storage devices. Additionally, using separate metadata servers for metadata operations has become even more popular. The MPI-IO library and other parallel access methods became popular in parallel applications during this time.

***Additionally, in this time frame several important HPC file systems and I/O related R&D efforts were begun including:***

**Relaxation of Posix Semantics**

A realization that the POSIX semantics had to be broken to enable scaling occurred. Ordering semantics to ensure last writer wins on overlapped I/O operations and lazy updates of last update times/dates and file sizes for files being written to by many clients concurrently are both examples of where relaxations have been made. File systems like PVFS, Lustre, and Panasas have all implemented options for relaxed POSIX semantics where absolutely necessary for scaling. These three file systems and others have implemented special (non POSIX standard) I/O controls (IOCTL's) which allow applications to control widths, depths, and other striping oriented layout information for files. The PVFS is probably the best example of extending POSIX semantics for scaling. This is accomplished by providing a user space library interface to the file system which is well integrated with the MPI-IO parallel I/O library. To assist with situations where overlapped I/O is necessary and control over last writer wins semantics is important for the application, Northwestern University has done important work in enabling detecting and handling these overlapped I/O operations within the MPI-IO library, where multiple clients can coordinate these activities in an intelligent way.

### **Scaling**

Extremely scalable parallel data movement bandwidth has been achieved by client server based file systems. Both Lustre and Panasas have shown coordinated bandwidths in excess of 10 GigaBytes/sec and both have plans of exceeding 30-50 GigaBytes/sec in fy05. For the most part, the data movement bandwidth scaling problem has been solved at least for non-overlapped, large buffer, parallel I/O operations to/from file systems. Some initial research has begun at the University of California Santa Cruz (UCSC) in the area of scalable metadata. The problem here is handling tens to hundreds of thousands of inserts, deletes, and queries per second in file systems that will manage billions of files. Some file systems like the IBM SAN file system and others have introduced scalable metadata systems, but only for scalability across multiple directories. Scalability within a single directory is a very hard problem especially when posed with the possibility that mass operations like tens of thousands of inserts could be requested into the same directory or subdirectory all within a few milliseconds. UCSC has come up with some novel ideas in trying to address this problem. Additionally, the Lustre and Panasas file systems are building first generation engineering approaches to the scalable metadata problem for scalability across multiple directories as well as within one directory.

### **NFS version 4**

The Internet Engineering Task Force (IETF) NFS version 4 (NFSv4) (<http://nfsv4.org>) effort has seen progress in the last two years. The NFS has been riddled with security problems since its inception. The NFSv4 effort is addressing this issue through the use of the General Security Services (GSS) infrastructure. Many government sites see this as a very useful move for the NFS. Additionally, the NFSv4 effort has new compound operations capability which allows for multiple operations to be coalesced to allow for efficiencies never before possible with the NFS. Additionally, there is a new parallel NFS effort (pNFS) (<http://www.ietf.org/proceedings/04mar/slides/nfsv4-1.pdf>) which is a part of the overall IETF NFSv4 project. This pNFS effort promises to allow NFSv4 clients to perform metadata operations on file systems via the NFSv4 server and then bypass the NFSv4 server for data movement operations. There have been many non-

standard modifications to the NFS over the years to bypass the NFS server for data movement operations, going directly to the storage devices. This pNFS effort legitimizes these approaches via the IETF standardization process. This pNFS effort promises to make NFS clients first class clients to parallel file systems for performance. This allows for a large variety of heterogeneous clients to have high performance clients to parallel file systems. Important work by the University of Michigan is enabling a Linux implementation of these NFSv4 features. Many vendors are participating in building NFSv4 releases as well as working on the NFSv4 and pNFS IETF standards effort. Garth Gibson of Panasas and Carnegie Mellon University (CMU) has been instrumental in working on the pNFS standards effort.

### **Higher level I/O libraries and integration with I/O Middleware**

The I/O software has to be able to extract the available performance from the underlying storage devices. The high level I/O library must be well integrated into the overall I/O stack so that performance can be attained. It is vital to recognize important performance work done by assisting with tighter integration between I/O Middleware and higher level I/O libraries like the Hierarchical Data Formats version 5 (HDF5) from the National Center for Supercomputer Applications (NCSA) and parallel Network Common Data Format (PnetCDF) Argonne National Laboratory (ANL) and Northwestern University. It is important to tune these higher level libraries and tune how applications utilize these libraries. Work by ANL to utilize the PnetCDF library integrated with the MPI-IO library and the PVFS achieved promising performance results. The joint work by Los Alamos National Laboratory and NCSA using the Unified Data Models (UDM) library in conjunction with the HDF5 library which uses the MPI-IO I/O Middleware has also achieved promising performance for applications.

### **Enterprise Class Global Parallel File System (GPFS – not to be confused with the IBM GPFS – General Purpose File system)**

Supercomputer sites have been deploying more than one computing cluster for many years. Sometimes sites will have a very large cluster for simulation with smaller clusters for pre/post processing of data, reduction, analysis, and visualization. In the last few years this has become common for larger supercomputer sites to have more than one very large cluster for simulation. This often arises from sites installing a new large cluster every year or two but keeping the past generation or two before decommissioning. Sites with more than one cluster increasingly want to share access to the same data between clusters. In the past it was very common to have a separate parallel or high bandwidth file system on each cluster with some means to move the data between clusters for sharing, either through direct data movement or via a common archive capability. Many supercomputer sites are now expressing the desire to have and deploy parallel and scalable file systems that are shared between all the supercomputers at the site. We refer to these deployments as Enterprise Class Global Parallel File Systems (GPFS). Often sites desiring this Enterprise Class GPFS capability even want to provide access not only to just the clusters in the enterprise, but for workstations and other servers in the enterprise. Giving access to multiple clusters and workstations to a common file system gives rise to new issues like Quality of Service (QoS) guarantees to more important

clients, differing security between different kinds of clients, and heterogeneous access. Space allocation on a filesystem according to various policies, which relate to which system, project, or user is creating the data, will also become important, as multiple funding sources will be dictating how the resources are used. Relevant work in this area includes the NFSv4 and pNFS work already mentioned, as well as some early design work for QoS in the Lustre file system and in the ANSI T10/1355-D Object based Storage Device standardization effort.

### **Utilize processing power near the storage devices – Active Storage**

With the success of client server oriented file systems, the opportunity to utilize server side processing power near the disk storage device was recognized. Many organizations have contemplated using this power near the disk. Early research at the Intelligent Storage Consortia (ISC) at the University of Minnesota has looked at ways to utilize the power near the storage device to provide functions like hierarchical storage management (HSM), indexing, and mining. Other universities have also done preliminary research into how the power near the storage device could be used. Proof of Active Storage concepts has been done at PNNL(<http://www.emsl.pnl.gov/>), by augmenting the Lustre filesystem. Additionally, industry has begun to deploy storage systems with processing capabilities, probably the most noteworthy are the Content Addressable Storage released recently by the EMC<sup>2</sup> Corporation (<http://www.emc.com/products/systems/centera/>) which allows access to data objects by their content, and the Data Appliance also released recently by the Nettezza Company (<http://www.netezza.com>) provides database query operations. This area of R&D represents great promise, but only initial work has been done. The advent of the ANSI T10/1355-D OSD standard which provides a standard and secure way to request actions from an intelligent storage device is an important step to being able to utilize processing power near the storage device.

### ***Summary of current state***

It would be appropriate to call the late 1990's to the present the era spent enabling extremely scalable data movement bandwidth. Multiple client/server based file systems have achieved greater than 10 GigaBytes/sec and have plans to exceed 30-50 GigaBytes/sec in fy05. Supercomputing sites have deployed scalable global parallel file systems for individual extremely large clusters as well as for multiple clusters in an enterprise. It is now possible to scale bandwidth, and there is competition in the scalable global parallel file systems market space through products that span the gamut from completely proprietary to open source. Standards are emerging, like the ISCSI standard, the ANSI T10/1355-D OSD standard, and others, which will lead to even more competition. Some work has been done to deal with POSIX limitations; the NFSv4/pNFS efforts appear to be headed in a good direction for security, heterogeneous access, and WAN access; tighter integration of the I/O software stack has yielded good results; and some promising initial work on metadata scaling and how to utilize the power near the disk has been started. Much progress has been made in the HPC file systems and scalable I/O area in the last decade.

### ***Demands of the next 5 years***

In the near future, sites will deploy supercomputers with tens of thousands of processors routinely, perhaps even hundreds of thousands. Bandwidth needs to storage will go from tens of GigaBytes/sec to TeraBytes/sec. Online storage needs to support work flows for efficient complex science will begin to approach the exabyte range. The ability to handle a more varied I/O workload including small to extremely large I/O operations, extremely high metadata activities, and multiple simultaneous workloads will be required. Additionally, new access methods, such as access methods to files/objects in arrangements other than tree based directories, will be required. Global or virtual enterprise and wide sharing of data with flexible and effective security will be required. Current extreme scale file system deployments already suffer from reliability and availability issues including recovery times from corruption issues and rebuild times. As these extreme scale deployments grow larger, these issues will only get worse. It will possibly be unthinkable for a site to run a file system check utility, yet it is almost a given that corruption issues will arise. Recovery times need to be reduced by orders of magnitude and these types of tools need to be reliable, even though they may rarely be used. The number of storage devices needed in a single coordinated operation could be in the tens to hundreds of thousands, making the need for integrity and reliability schemes to be far more scalable than available today. Management for enterprise class global parallel file/storage systems will become increasingly difficult due to the number of elements involved and the extreme varied workloads. The challenges of the future are formidable.

## **Key areas of possible future research, development, and standards work**

As indicated by the challenges for the future, the needed R&D activities need to shift from scaling of data movement bandwidth for large I/O operations to scaling performance for high volumes of small I/O operations, high volumes of metadata operations, and extreme mixing of a variety of workloads. Improving reliability, integrity, availability, and manageability by orders of magnitude must be addressed. Additionally, R&D into new access methods, issues for enterprise wide sharing, WAN and heterogeneous access, security, as well as extreme scaling issues for metadata operations, and security will be required. Novel approaches to these issues such as leveraging the power near the storage devices, tighter integration of the I/O software stack, as well as other approaches should be studied.

It is important that the natural evolution from ideas to prototypes, from prototypes to user level library implementations, from user level library implementations to standards, standard implementations and even products, be supported and managed. Thus, all efforts throughout this evolution should be undertaken where promising results are indicated.

Below, categories of future R&D and standards work are discussed.

### **The POSIX I/O API**

The POSIX API is "unnatural" for high-end computing applications. Opportunity abounds to make the POSIX I/O API more friendly to HPC and parallelism. The entire

set of operations should be combed over and carefully, consistently, altered for high-end computing needs. Then, the result must be re-integrated in such a way that it is enabled by all applications in a positive fashion. The resulting semantics, after all, are less than useful for legacy applications as well as single platform applications.

### *Ordering*

One of the prime reasons the POSIX I/O API is awkward for HPC stems from the "stream of bytes" paradigm in which POSIX I/O is based. POSIX I/O was developed to provide an interface from a single machine with a single memory space to a streaming device with some simple random access capabilities. HPC/parallel I/O applications are based on distributed memories being mapped to many storage devices. A re-interpretation towards the concept of a "vector of bytes" would be more appropriate for HPC applications versus a "stream of bytes" model. This would entail a careful reexamination of the POSIX I/O-related interfaces to eliminate or relax stream-oriented semantics.

### *Coherence*

Probably the worst offense for really high bandwidth I/O is the fundamental read and write calls. They have two glaring problems. The first is coherency. The last-writer-wins semantic of the write call without cache-snooping is difficult, perhaps impossible, to achieve in a scalable fashion. Similarly, again without cache-snooping, the overhead of cache invalidation is enormous for reads. Especially if attempted for regions for which an application might never have interest. Additionally, block boundaries can present coherence issues for the application. A standard way for applications to assume all responsibility for coherency is needed, which implies that application control of cache flushing/invalidation at some level is also needed. Additionally dealing with block boundaries and alignment needs to be dealt with in the API in a consistent manner which takes alignment/block boundary issues away from the application. This problem is particularly bad in the implementation of `O_Direct` today.

### *Extension issues*

The POSIX I/O API is organized as a set of mandatory functions and sets of extensions to that set of mandatory functions. Current extensions of the API are awkward for use in HPC. For instance, the real-time extensions for list-based I/O (`listio`) are useful, however they are awkward in that they have the restriction that the memory regions of interest must coincide exactly with a region in the file. For many applications, relationship of memory regions to regions in the file is often not possible. Instead, two separate vectors, one of memory regions and one of file regions, could be passed and the two lists reconciled by the implementation. Such a concept allows scatter/gather-gather/scatter semantics.

### *Missing capabilities*

The POSIX I/O API is also not as complete as one would like. For instance, there is a call to position and write a buffer of data (pwrite) but no call to position and write a vector of described memory regions, like a pwritev.

### *Metadata*

The classic "wish" in this area is for the support of "lazy" attributes. These are results to the stat call, where some values may not be maintained at the same granularity normally expected. The most obvious fields are those that record timestamps for last update and modify. Many file systems implement these but no two in the same way. A standard, portable set of semantics would be useful. Explorations into a more descriptive API for metadata management and query to allow applications to deal with the needed information could be helpful in this area. For years, the backup/archive industry has needed a portable bulk metadata interface to the metadata of file systems. There are many opportunities for R&D in the area of an overhaul of how metadata is handled and the API's with which it is accessed given the extremely limited implementations in today's file systems.

### *Locking schemes*

Locking schemes that support cooperating groups of processes is also necessary. The current POSIX semantics assume only a single process would ever want exclusive write access. In the HPC/parallel world, groups of processes may want to gain exclusive access to a file. Perhaps a mandatory version of the fcntl and flock is needed. It is necessary that more options for how locks can be requested and revoked be provided. Legacy codes must continue to work as expected, so current locking semantics must be maintained.

### *Shared file descriptors*

Shared file descriptors between nodes in a cluster would be of great value, not just between processes on a node. The component count for supercomputers is going up in the next generation of supercomputers. Full lookups in the name space are going to have to become a thing of the past for parallel codes. Some mechanism decreasing the need for mass name-space traversal is desperately needed, even if it requires new semantics to accomplish it. It is possible to implement this via higher level function in the I/O stack. Implementation of shared file descriptors at the file system level might be difficult but none the less would be quite useful, if achievable with a reasonable amount of R&D investment. As mentioned in the metadata section above, an alternate API for name space traversal as well as alternate file organization (something other than a tree) might also be a way to assist in this area.

### *Portability of hinting for layouts and other information*

There is a need for proper hinting calls. Things like stripe, stride, depth, raid layout options, etc. need to be accomplished in some portable way. Additionally, there needs to be mechanisms for adding standard hinting without major new standardization efforts. Perhaps the MPI-IO Info approach for hinting can serve as a prototype, particularly in terms of the semantics, like ignoring of unknown hints and the mechanism for getting the

values to use. For users to understand and use these hints effectively, they need to be as easy to use as things like umasks, shell variables, or file permissions.

### **Necessary determinism**

Additionally, all operations done on the basis of time are awkward to deal with on supercomputers with light weight operating systems due to the inability to respond via asynchronous signaling to call back mechanisms. Supercomputing applications need more deterministic behavior and more control over the hardware throughout the entire computation and I/O hardware stacks. Operations with the ability to be driven from clients that can't listen for call backs is vital. It is quite likely that some variants of supercomputers with hundreds of thousands of processors simply won't be able to be bothered with call back mechanisms at all.

### **Active storage concepts**

Active storage concepts are those ideas where CPU power near the storage device is utilized for better overall application performance or machine throughput. The work on active storage and associated concepts is interesting, although it is important that the value proposition be examined closely. Just because processing power near the storage device can be used to participate in the problem solution does not mean there is value in doing the processing near the disk as opposed to on other hardware. Many examples of particular applications and classes of applications enjoying significant benefits from this technology are available. However, to date, no pursuit of a generally useful, secure interface has been made. Research into a library, or hardware/firmware, interface supporting "sandboxes" and rich programming support could go a long way toward motivating applications to make use of the scheme. Without proper interfaces, this proposed new function will always be a "one-off" for any application, generally useful and portable for that application, but extremely hard to reuse, especially in an environment where more than one application might like to leverage active storage paradigms simultaneously. This R&D area is in its infancy, still mostly in prototypes. More work needs to be done to understand better the value proposition this idea brings and how it might be used in a more standard way.

#### *Application of active storage concepts – Network Topology*

In the local, machine, or system area network setting, the processing power near the storage device has no real advantage in bandwidth or latency to the storage device over any other processor, it is unclear that applications actually benefit from the processor near the disk versus just adding another general purpose processor to the application pool or a co-processor near the disk used only for application specific code. There is risk, availability/reliability risk, in putting application oriented function directly in the path of a highly shared item like a storage device. The processing power near the storage device does enjoy one advantage in this setting, that being location. All accesses to the storage

device go through this processor. It is possible that this advantage could be exploited in some manner.

In the wide area network setting, the processing power near the storage device offers at least a latency advantage and possible also a bandwidth advantage over a general purpose processor. In this setting, there are many advantages to exploit, including smart batching of requests to hide latency, retrieving only the data needed for transmission over the WAN, etc.

#### *Application of active storage concepts – enhancing the I/O stack function*

Another important aspect is the ability to utilize the processing power near the storage device to simplify the higher layers in the I/O stack. Pushing more function nearer the storage device could have the benefit of allowing more innovation to occur for file systems, I/O Middle Ware, and high level I/O libraries. Exposing data layout information to the processor near the storage device could help that processor better map I/O operations to the geometry of the underlying storage and open up new possibilities for I/O stack exploitation of this concept. Database systems live in this world, so it is likely that many ideas can be formulated by studying database technology. R&D in this area could pay big dividends for some applications.

#### *Leveraging active storage based file system technology, Archive/HSM, Other approaches*

One example of utilizing active storage to allow for enhancement of the I/O stack is the possibility of integration of Archive/HSM function. Disk-based file systems for clusters are increasingly using multiple software "mover" components to accomplish parallel data transfers. These movers, most often, function by exporting access to unique, independent data stores.

Classic hierarchical storage management (HSM) methodologies also employ multiple movers, but curiously, usually to support more connections, not parallel connections. Lessons learned from recent file systems work could be used to simplify the back-end data path in HSMs by using a metadata service to maintain tape and location layout information. Perhaps a realistic core set of requirements for archive products for science use might be a stepping-off point to an acceptable interface to HSM software with a usable lifetime greater than a decade. The marriage of modern file system designs with a subset of classic HSM software could yield a seamless infinite global parallel file system solution which could eliminate the need for a separate parallel or serial archive capability.

Further, there are doubtless additional approaches not yet considered for using active storage. There needs to be an effort to pursue other ways one might design a scalable file system with computational power near the storage devices to contribute to providing solutions for data intensive related problems?

#### *Application integration with Active Storage*

As has been mentioned earlier, tighter integration in the I/O stack is becoming important for applications to effectively tap the performance of the I/O Middleware, file systems, and storage devices. Extending this integration from the application all the way to the storage device through the use of processing power near the storage device, which has been shown in the past to have promising performance, warrants a closer look. As mentioned above, if the processing power near the storage device had information about data layout, much could be done to exploit that fact higher in the I/O stack. A generic capability for applications to securely and effectively utilize processing power near the disk should be explored, prototypes of this environment need to be developed and tested to determine if the performance value proposition is worth the risks of destabilizing a highly shared device like a storage device. More work in understanding the performance payoff for applications, the API(s) needed to accomplish this capability, and the risks in providing this capability needs to be done.

To date, though, all research in this area has only been applicable to a single application at any given moment. To be really useful, sandboxes or other technology must be applied to allow independent applications access simultaneously.

#### **NFSv4**

As has been mentioned earlier in this document, the NFSv4 effort has yielded a much more secure NFS capability. There is still good work in the pipeline from the NFSv4 effort which must continue to be supported. Additions of directory leasing capabilities for WAN access performance, load balanced NFS serving, and the important pNFS effort which promises to allow heterogeneous NFSv4 clients to access file systems more natively by bypassing the NFS server for data movement operations, are all vital parts of the NFSv4 effort that need to be accomplished. In order for these efforts to be successful, development and standards work need to continue. The IETF requires two interoperating implementations, so this requires persistence in funding and oversight to see these projects through to completion and insertion officially into the IETF.

#### **Enterprise Class Global Parallel File System (GPFS – not to be confused with the IBM GPFS – General Purpose File system)**

The use of a global parallel file system by multiple clusters within an enterprise and extending access to the desktop workstations of an enterprise causes a set of issues to arise. These issues have mostly to do with treating one set of clients differently than others. There may be a need for security or other services to behave differently based on file system client or sets of clients. Additionally, this idea applied to performance implies a QoS solution is needed to enable one set of applications/clients to be treated differently than others applications/clients. R&D and standards work need to occur to enable these capabilities to support this enterprise class sharing concept in a portable way. Further, when connecting multiple clusters of different technologies and workstations to an Enterprise Class GPFS, scalable backbone technologies that allow heterogeneous interconnection at extremely scalable bandwidths with high reliability and availability are needed. Normally, single cluster interconnects are designed to scale to very high cross sectional bandwidth, but intra-cluster networks are not designed to scale that broadly. This multi-heterogeneous-cluster to common GPFS scalable network is a new

development and needs to be studied. It is possible that Internet Protocol version 6 (IPv6) may be of assistance with this issue.

### **Scaling**

As mentioned before, clusters of unprecedented scale are on the drawing boards with tens to hundreds of thousands of processors. Given that data movement scaling has been accomplished, R&D to address scaling other attributes of file systems and I/O is desperately needed.

#### *Metadata*

Clustered file systems seem to be converging on an architecture that employs a centralized metadata service to maintain layout and allocation information among multiple, distinct movers. While this has had significant, positive impact on the scalability in the data path, it has been at the expense of the metadata service. Due to scale up, the transaction rates against the metadata service have increased. As well, the amount of information communicated between the metadata service component and the clients has increased. There is some belief that such a file system design is problematic. The reason for this is seek-latency in disk media. Additionally, alternate metadata access methods, like bulk metadata access and perhaps alternative to tree shaped access of the metadata might provide both new needed function and relieve some of the metadata scaling issues. It is vital that continued R&D investments be made in this vital scaling of metadata performance.

#### *Data movement bandwidth with small and unaligned I/O operations*

With the incredible success in scaling data movement bandwidth using large I/O operations, it is now time to concentrate on dealing well with the scaling of small I/O operations. Many applications have not been able to take advantage of the enormous improvements in file system scalability in the last several years due to the small I/O operation sizes used by these applications. It is vital that all applications be able to have scalable I/O available to them. Often, it is inconvenient, or impossible, for the applications to be altered. They are expensive, proven codes and, in some cases, the host machines do not have the memory it would require to efficiently rearrange working sets for efficient data transfer. For non dusty deck applications, R&D in areas of more aggressive caching in high level I/O libraries, I/O middleware, and alternate file system consistency close-to-open semantics could pay off, particularly in applications with lots of small I/O operations to independent files. It is also possible that active storage pursuits could assist in this area significantly by providing data layout information to the processor near the storage devices to allow for better mapping of the workload to the underlying storage geometry. For the more dusty deck oriented applications, this is a very difficult and perhaps nearly intractable problem, however, if R&D could assist these applications in their ability to use global parallel file systems more effectively and efficiently, there is a win for users.

#### *High Level library exploitation of scalability enhancements*

As has been mentioned several times in this document, integration up and down the entire I/O software stack has yielded good performance benefits. If file systems become better at scaling of metadata and small I/O operations as mentioned above, further re-integration of higher level I/O libraries to take advantage of these file system improvements may be possible and should be explored. As an example, it is possible that formatting libraries, which currently put all data and metadata for a single application run into a single file, might leverage scalable metadata operations by keeping a family of related files associated with a single application run. It is important to not resort to thousands of files per application or one file per process in this endeavor, but some modest number of files based on access patterns could be exploited. In the storage management field, the term “collections” is used to describe this concept. All advancements in the file system layer and below should be exploited if possible by higher layers in the I/O software stack.

### *Security*

Another dimension for file systems that must be addressed in a scalable fashion is security. Allowing file system clients to access storage devices in a scalable way may require transactional oriented security so storage devices can trust that clients are authorized to perform the request. Additionally, with metadata services becoming more scalable, security related workload for authentication and authorization, which the metadata server must do, is increasing. The necessity for security even as scaling increases means security services must be scalable too. For this reason, R&D investments in security scaling must be undertaken.

### *Reliability and availability*

As has been discussed, clusters of unprecedented scale are being planned. To provide the needed file system bandwidth to these clusters, unprecedented numbers of storage devices will need to be used in a coordinated way. Striping data from a single application over enormous numbers of disks will eventually lead to difficulties in protecting against data unavailability or loss. Current RAID protection and availability technologies are not designed to provide sufficient protection for such immense scale. One concept that could be pursued is lazy redundancy, producing redundant data at specific points in time, perhaps associated with checkpoints or snapshots. This concept allows for variable redundancy on a per file basis which allows for trading off reliability for performance based on expected usage. Another concept that could be pursued is the ideas related to raiding memories in compute nodes. This concept works quite well for pure defensive I/O and dovetails nicely with MPI-2 features of dynamic process/communicator growth. There are no doubt other redundancy concepts that could be pursued as well.

### *Management*

Yet another area affected by immense scale is management. The number of devices needed to provide the needed scalable file system service in a demanding and mixed workload environment of the future will be extremely difficult to manage given current technology. Advances must be made in massive scale storage management to enable management survival with future file system deployments.

### *Autonomic Storage Management concepts*

The storage industry is currently working on management solutions that are fully automated. Ideas like, storage that self configures, self heals, self migrates, and self tunes are all being pursued. These ideas are all good ideas and the related projects need to be at least followed by the HPC I/O community. Additionally if these features are to be useful in the HPC environment, where things like determinism in parallel are important, it would be very useful for the HPC community to be involved in this Autonomic Storage Management R&D to ensure that these good ideas are implemented in a way that the HPC community can benefit. As an addition to this thinking, automated mining of data is also being pursued. These features also could be useful in the HPC environment but must be developed with consideration for the HPC I/O environment.

#### *Hierarchical I/O architectures*

Many supercomputers are arranged with compute nodes, I/O/routing nodes, and storage. I/O. All I/O operations on behalf of the compute nodes are routed in some manner through the I/O/routing nodes. Some of the newer architectures emerging, like the Red Storm and Blue Gene/Light architectures, require this type of I/O arrangement. An excellent research questions is: “Can part of the I/O function be placed at these I/O/router nodes to assist in performance for the user, especially in the areas of caching and aggregation”?

#### **Tools for scalable I/O tracing and file system simulation**

In parallel application building and tuning, there are a multitude of correctness and performance tools available to applications. In the area of scalable I/O and file systems, there are few generally applicable tools available. Tools and benchmarks for use by application programmers, library developers, and file system managers would be of enormous use for the future.

#### *Tracing*

Tools to quickly get tracing information from parallel applications, analyzing these traces to characterize the applications I/O footprint, and even being able to replay the traces in parallel against real or simulated parallel file systems would be of great use.

#### *Simulation*

Tools for simulation of portions or entire global parallel file systems would also be of great use to assist in understanding design trade-offs for I/O performance for applications, libraries, and file systems. Most other areas of the computer industry rely heavily on simulation tools. While asking for a parallel file system simulator is a tall order, the value it would have could be enormous.

#### *Benchmarking*

As in other areas of computing, benchmarking is a vital part of the I/O professional’s toolkit. Benchmarks in the areas of interactive benchmarks (simulating user experience items like ls, cp, rm, tar, etc.), throughput benchmarks (including both peak performance

and I/O kernels from real applications), and metadata benchmarks (collective opens, creates, resizes, etc) are needed. Some of these benchmarks exist and others do not. R&D in the benchmarking area is needed to collect and index current benchmarks and design and build new benchmarks to fill any gaps. At the very least, a clearing house for all I/O benchmarks and their usage could be of benefit. It currently is quite difficult to determine if a benchmark exists for a particular function or workload.

### **New metadata layouts other than tree based directories**

In order to assist applications with managing enormous amounts of data, application programmers and data management specialists are calling for the ability to store and retrieve data in organizations other than the age old file system tree based directory structure. The data formats libraries currently provide some of this function, but it is not at all well mated to the underlying file system capabilities. Databases are often called upon to provide this capability but they are not designed for petabyte or exabyte scale stores with immense numbers of clients. Exploratory work in providing new metadata layouts is vital to address this identified need.

### **Kernel/user space data movement and control passing**

There are many benefits to providing I/O related services completely in user space. Due to the kernel based file system layer paradigm on which most all operating system function rests, it is necessary to continue providing file system services through the Unix kernel. If a zero-copy data path from user space to kernel and back were standardized in Unix/Linux, it would be possible to implement more file system services in user space without a penalty in bandwidth or latency performance. This development could open up new and innovative I/O and file system services never before possible due to the abundance of user space developers. As well, user-space implementations tend to be more highly portable and cheaper to implement.

### **IPv6**

If machines continue to grow in power at the same rate – Moore’s law again – then the number of components in the I/O system must increase dramatically. Most of these components are uniquely addressable. While the internet protocol (IP) is ubiquitous, its address space is partitioned into only very small remaining chunks. The advent of IPv6 presents an opportunity to cleanly craft addressable sub-units in the I/O system. Unfortunately, little attention has been paid to this relevant streamlining.

### **Support of Storage Centers of Excellence**

As part of the overall investment strategy, consideration should be given to supporting the approximately 3 university storage centers of excellence within the US at some base level. These centers provide ongoing file systems, storage, and scalable I/O research funded by industry partners. Supporting and being involved in these centers leads to more access to industry planners, more leverage over research topics, and more access to students and faculty. Additionally, supporting these centers produces the next generation of researchers in the field.

## **Conclusion**

More focused and complete government investment needs to be made in this area of HPC, given its importance and its lack of sufficient funding levels in the past, compared to other elements of HPC. Scalable I/O is perhaps the most overlooked area of HPC R&D, and given the information generating capabilities being installed and contemplated, it is a mistake to continue to neglect this area of HPC. Many areas in need of new and continued investment in R&D and standardization in this crucial HPC I/O area have been summarized in this document.