HEC FSIO Session 4: Quality of Service Talks & Roadmap

Gary Grider, Los Alamos National Lab

August 2009

Current R&D Gaps

- End-to-end QoS in HEC
- Standard API for QoS

2006 HECURA/CPA Projects

- End-to-End Performance Management for Large Distributed Storage; Scott Brandt, University of CA, Santa Cruz
  - 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.

2006 HECURA/CPA Projects

- QoS Guarantee for Scalable Parallel Storage Systems; Tzi-Cker Chiueh, SUNY at Stony Brook
  - 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 system

2006 HECURA/CPA Projects

- Performance Insulation and Predictability for Shared Cluster Storage; Greg Ganger, Carnegie-Mellon University
  - Accomplishing the desired insulation and predictability requires cache management, disk layout, disk scheduling, and storage-node selection policies that explicitly avoid interference. This research combines and builds on techniques from database systems (e.g., access pattern shaping and query-specific cache management) and storage/file systems (e.g., disk scheduling and storage-node selection). Two specific techniques are: (1) Using prefetching and write-back that is aware of the applications associated with data and requests, efficiency-reducing interleaving can be avoided; (2) Partitioning the cache space based on per-workload benefits, determined by recognizing each workload's access pattern, one application's data cannot get an unbounded footprint in the storage server cache.

2009 HECURA Projects and Presentations

- QoS-driven Storage Management for High-End Computing Systems; Ming Zhao, Florida International
  - This NSF HECURA project tackles the challenges in quality of service (QoS) driven HEC storage management, aiming to support I/O bandwidth guarantees in PFSs by addressing the following four research aspects: 1. Per-application I/O bandwidth allocation based on PFS virtualization, where each application gets its specific I/O bandwidth share through its dynamically created virtual PFS. 2. PFS management services that control the lifecycle and configuration of per-application virtual PFSs as well as support application I/O monitoring and storage resource reservation. 3. Efficient I/O bandwidth allocation through autonomic, fine-grained resource scheduling across applications that incorporate coordinated scheduling and optimizations based on profiling and prediction. 4. Scalable application checkpointing based on performance isolation and optimization on virtual PFSs customized for checkpointing I/Os.

2009 HECURA Projects and Presentations

- Adaptive Techniques for Achieving End-to-End QoS in the I/O Stack on Petascale Multiprocessors; Mahmut Kandemir, Penn State
  - This project investigates a revolutionary approach to the QoS-aware management of the I/O stack using feedback control theory, machine learning, and optimization. The goal is to maximize I/O performance and thus improve overall performance of large scale applications of national interest. The project uses (1) machine learning and optimization to determine the best decomposition of application-level QoS to sub-QoSs targeting individual resources, and (2) feedback control theory to allocate shared resources managed by the I/O stack such that the specified QoSs are satisfied throughout the execution. The project tests the developed I/O stack enhancements using the workloads at NCAR, LBNL and ANL systems. It also involves two efforts in broadening participation: CISE Visit in Engineering Weekends (VIEW) and NASA-Aerospace Education Services Project (NASA-AESP) at the Center for Science and the Schools (CSATS).

2009 HECURA Projects and Presentations

- Interleaving Workloads with Performance Guarantees on Storage Cluster; Alma Riska, College of William and Mary
  - This research focuses on the design and implementation of a lightweight, yet, versatile middleware framework that provides effective and scalable solutions to the problem of interleaving storage workloads with a wide spectrum of demands. The framework uses simple and non-intrusive collection of workload statistics such as workload histograms and measures of temporal dependence to provide accurate forecasting of system workload characteristics and their impact on system metrics. The framework maps accurately and swiftly complex processes that exist and interact in storage clusters into robust allocation decisions. Central to the framework is its ability to estimate beforehand the effect of resource allocation policies on system metrics, which enables navigating through multiple possible allocations of system resources and selecting the on that best meets system targets.

2008 QoS Gap Area

Area	Researchers	FY 07	FY 08	FY 09	FY 10	FY 11	FY 12	Rankings
End to End QoS in HEC	Brandt							Good research, but much work needed to get a standards based solution. Scale and dynamic environments have to be addressed at some point in time.
	Chiueh							
	Ganger							
Standard Interfaces for QoS	SciDAC - PDSI							Very partially addressed by proposed HEC POSIX Extensions. Will be driven by above "End to End QoS in HEC".
	POSIX HPC Extensions							

Very Important
Medium Importance
Low Importance
Greatly Needs Research
Needs Research
Does Not Need Research
Greatly Needs Commercialization
Ready and Needs Commercialization
Not Ready for Commercialization
Full Calendar Year Funding
Partial Calendar Year Funding
On-Going Work

New R&D Gaps

- End-to-end QoS in HEC
  - Local/server side (progress made here)
  - Network
  - Distributed
  - Analysis of where the slowdown is
  - Learning vs theory (control theory)
  - Thrashing identification and action
- Useful API's for QoS
  - User specification (look at schedulers/enterprise/SLA (what is different about HEC (multiple parallel!)
  - Metrics that can be specified that matter to the system
  - Integration with job scheduler information
- Measurement and understanding for QoS (may be overlap with M&U area)

QoS Breakout Discussion

- We have spent a lot of $
  - 3 2006 HECURA/CPA (12% of total)
  - 3 new HECURA (15% of total)
- There are only two gaps, end to end QoS and QoS APIs – have we made any progress
  - We need to break these gaps down into subgaps to show the progress that has been made
    - Progress on single server side QoS,
    - Some limited progress on network and client
    - Not much progress in distributed/parallel QoS
    - Not much progress on API
    - There has been no demonstration of two parallel jobs sharing file storage with QoS (this would be an excellent goal for the near future)

QoS Breakout Discussion (Cont)

- We should consider both learning based QoS and theory based QoS
  - We are funding a new QoS based on control theory
- Discussion on how to specify QoS
  - What are the metrics
    - No consensus direction on this
  - Users probably wont know what to specify (other than fast), admins don't know until a user yells, admins need ways of turning knobs until things are better
  - We really need to add a measurement and understanding for QoS Gap because we need to understand what is contributing to slow (stragglers), which might overlap with the measurement and understanding HECFSIO area. Is QoS measurement and understanding that different from other M&U

QoS Breakout Discussion (Cont)

- How do we get input, other than knobs
  - From users, from admins, from schedulers (maybe 3 api's)
  - Should leverage other communities work (like SLA)
  - Schedulers can help a LOT
  - Maybe this should be another goal show scheduler providing node/space share/time share info and inform simple QoS
    - » We do know that nodes are related, we do know which jobs are important by job queue information
    - » Can we tell the difference between interactive and batch, how bad can interactive performance be for the sake of not bothering the big batch job, 10 minutes for an (LS) may get the system admin yelled at
- Policy vs thrashing, trade off space
  - If you build the machine so there is no bottlenecks, then you minimize the need for QoS
  - It is all trade off space, spend more on hardware/software, go slower/faster, add QoS features
  - Some things are policy enforcement, others detecting thrashing

QoS Breakout Results

- End-to-end QoS in HEC 3 2 3 2 2 2 3 2 2 3 3 3 = 30
  - Local/server side (lots progress made here but hard to do)
  - Network (some progress)
  - Distributed
  - Analysis of where the slowdown is
  - Learning vs theory (control theory)
  - Thrashing identification and action
- Useful API's for QoS 3 1 1 1 1 3 3 1 2 1 3 2 3 3 2 = 30
  - User specification (look at schedulers/enterprise/SLA (what is different about HEC (multiple parallel!)
  - Metrics that can be specified that matter to the system
  - Integration with job scheduler information
- (added potential gap) Measurement and understanding for QoS (may be overlap with M&U area) 2 2 3 2 1 3 1 3 1 1 2 1 = 22