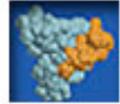
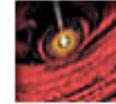
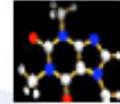
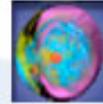




# SciDAC

Scientific Discovery through Advanced Computing



## Update on Petascale Data Storage Institute HEC File System & IO Workshop, Aug 11, 2009

Garth Gibson

Carnegie Mellon University and Panasas Inc.

SciDAC Petascale Data Storage Institute (PDSI)

[www.pdsi-scidac.org](http://www.pdsi-scidac.org)

Special thanks to John Bent, LANL

# SciDAC Petascale Data Storage Institute

---

- Eight organizations on the team
  - Carnegie Mellon University, Garth Gibson, PI
  - U. of California, Santa Cruz, Darrell Long
  - U. of Michigan, Ann Arbor, Peter Honeyman
  - Lawrence Berkeley Nat. Lab, John Shalf
  - Oak Ridge National Lab, Phil Roth
  - Pacific Northwest National Lab, Evan Felix
  - Los Alamos National Lab, Gary Grider
  - Sandia National Lab, Lee Ward



**Sandia  
National  
Laboratories**



**Carnegie Mellon**

[www.pdsi-scidac.org](http://www.pdsi-scidac.org)



center for  
information  
technology  
integration

UNIVERSITY OF MICHIGAN



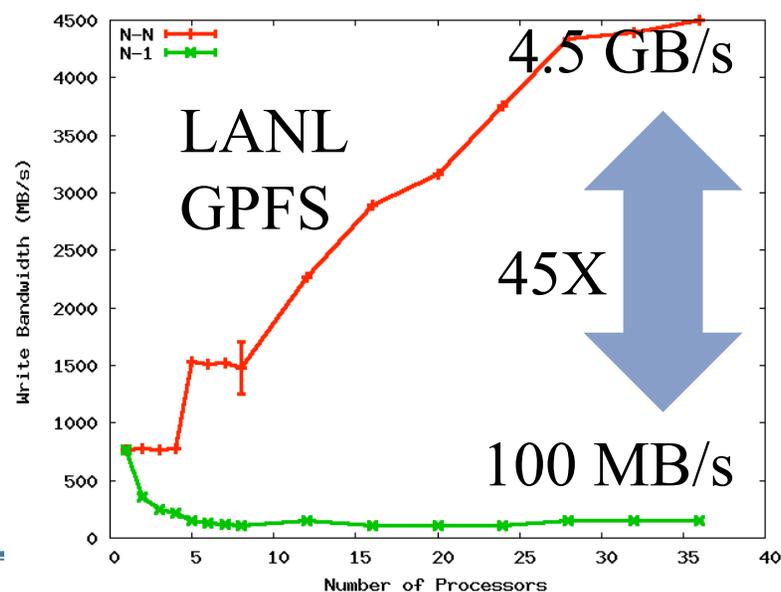
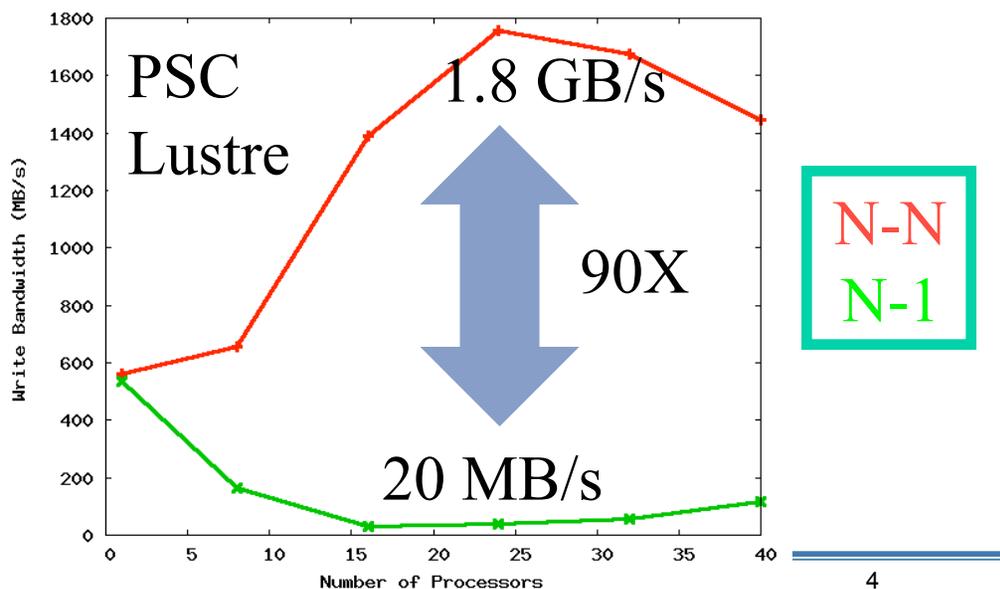
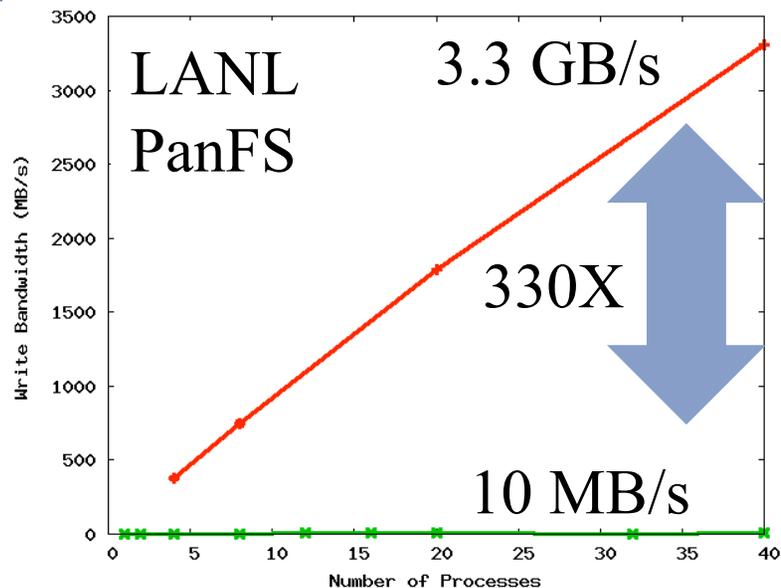
# Current Write Checkpoint Efficiency

---

- File system capture slows with concurrency
  - N nodes writing to 1 file, often strided, small writes (e.g. AMR)
- Written data is organized by app, middleware, file system, usually optimized for later read
  - Middleware structures logical data in files
    - GaTech/ORNL ADIOS is new attack on this
  - File system concurrent write sharing is serialized
    - LANL/CMU PLFS is new attack on this
  - File system seeks disks for layout of streaming objects
    - PSC Zest is new attack on this
- Basic approach: optimize for write, not read
  - History: database logs, log-structured filesystem

# How bad can it be?

- John Bent, LANL
- N-1 small strided writing hurts
  - ~ 100X reduction vs N-N
- Applies generally
  - ie., PanFS, GPFS, Lustre
  - Cross graph comparisons not meaningful



# N-1 is prominent

---

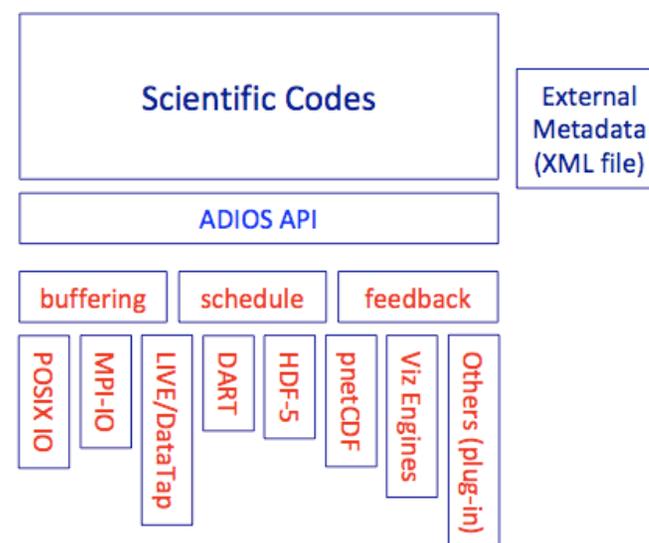
- Several old LANL codes use N-1 (over 50% of cycles)
- Newly written codes still choosing N-1
  - 2 of 8 open science applications on Roadrunner
  - NetCDF and HDF5 formatting libraries
- N-1 also prominent elsewhere
  - At least 10 of 23 on PIO benchmarks page are N-1
  - BTIO, FLASH IO, Chombo IO, QCD, etc. (GTC?)

# Output delayed & grouped before write

- GaTech
  - Lofstead
  - Zheng
  - Schwan
- ORNL
  - Klasky
- Chimera:
  - 1000x better BW
- GTC
  - 60% raw BW
- PDSW08

## Architecture (ADIOS)

- Change IO method by changing XML file
- Switch between synchronous and asynchronous
- Hook into other systems like visualization and workflow



# No filesystem used in capture

---

- PSC: Nowoczynski, Yanovich, Stone, Sommerfield
- Dedicated capture to checkpoint device, offline copy into slower file system
- PDSW08

## **Zest: Methods for optimized writes.**

*Pittsburgh Supercomputing Center*

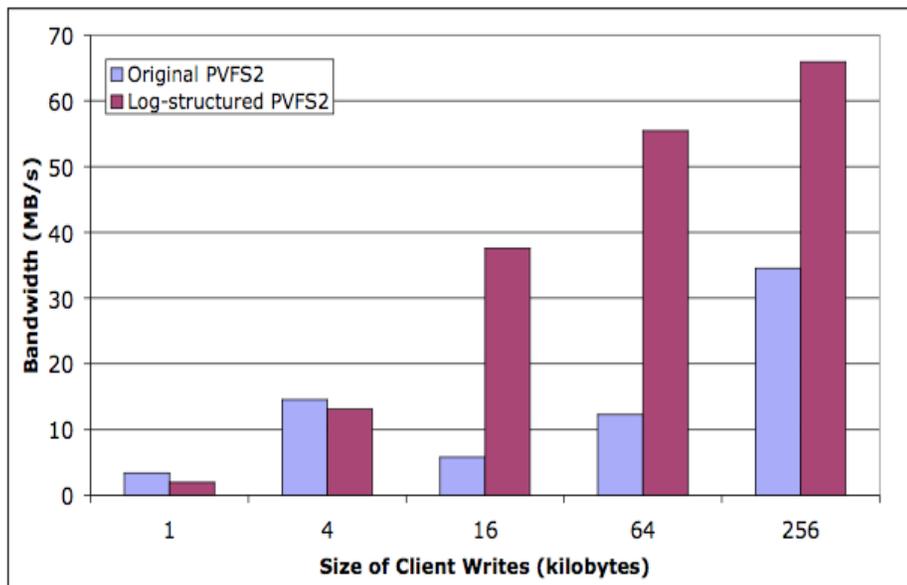
Zest uses several methods to minimize seeking and optimize write performance.

- Each disk is controlled by single I/O thread.
- Non-deterministic data placement. (*NDDP*)
- Client generated parity.
- No Leased locks

# Decouple concurrent interactions

- Log-structure writes into files for fewer seeks
- Try 1: CMU class project (Polte et al PDSW08)
  - log-structure storage inside PVFS
  - Promising results in PDSW08

## Log-Structured Writing



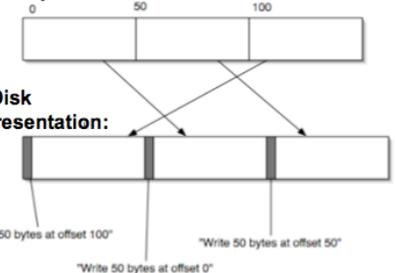
- A log can allocate random writes sequentially
  - Reduces number of seeks
- Proposed in LFS (Rosenblum, Ousterhout, 1992)
  - Implemented in WAFL and PanFS
- Use for checkpoints inspired by Zest

Series of writes:



Logical Representation:

```
write(file, buffer[100], 50, 100);  
write(file, buffer, 50, 0);  
write(file, buffer[50], 50, 50);
```



Carnegie Mellon  
Parallel Data Laboratory  
<http://www.pdl.cmu.edu/>

6

Milo Polte © November 08

Carnegie Mellon  
Parallel Data Laboratory

[www.pdsi-scidac.org](http://www.pdsi-scidac.org)

 pdsi

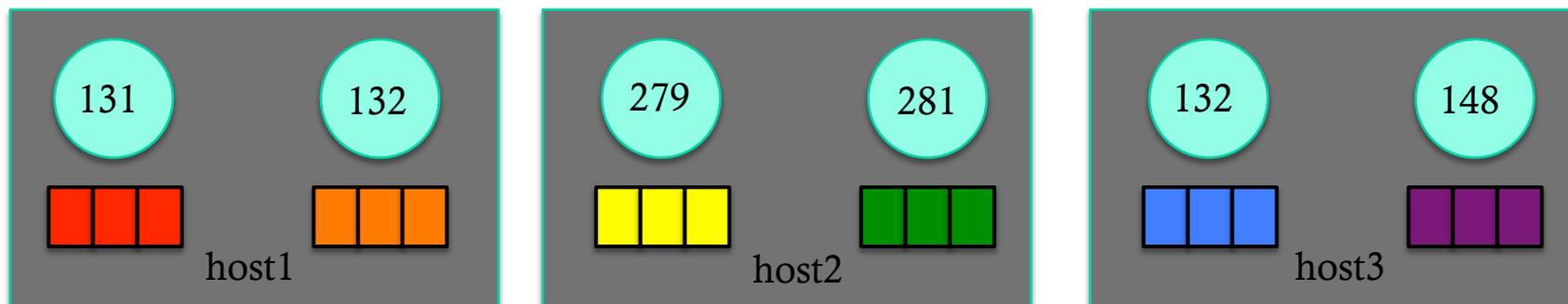
Garth Gibson, 11/21/2008

# So, Convert N-1 into N-N

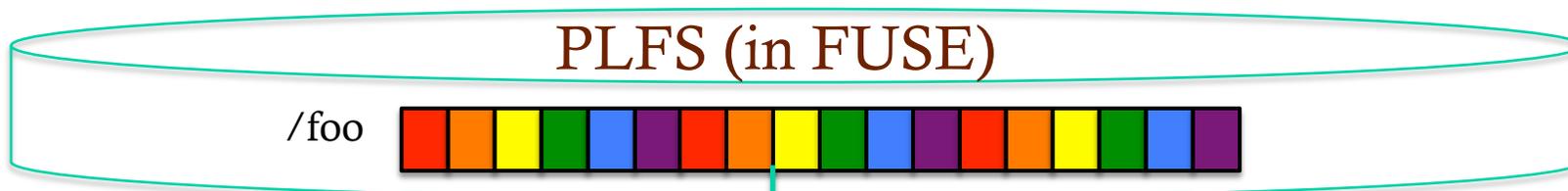
---

- But many applications won't do it
  - Archiving, mgmt, visualization, non-uniform restart
  - Developers are aware of the N-1 problems
    - But are loathe to change to N-N
    - One app wrote 10K lines of code, bulkio, to try to improve N-1
- If the apps won't do it, interposition can
  - Desirable characteristics
    - Low overhead (performance and resource)
    - User transparency (i.e. NO CODE REWRITING)
    - Portable and maintainable

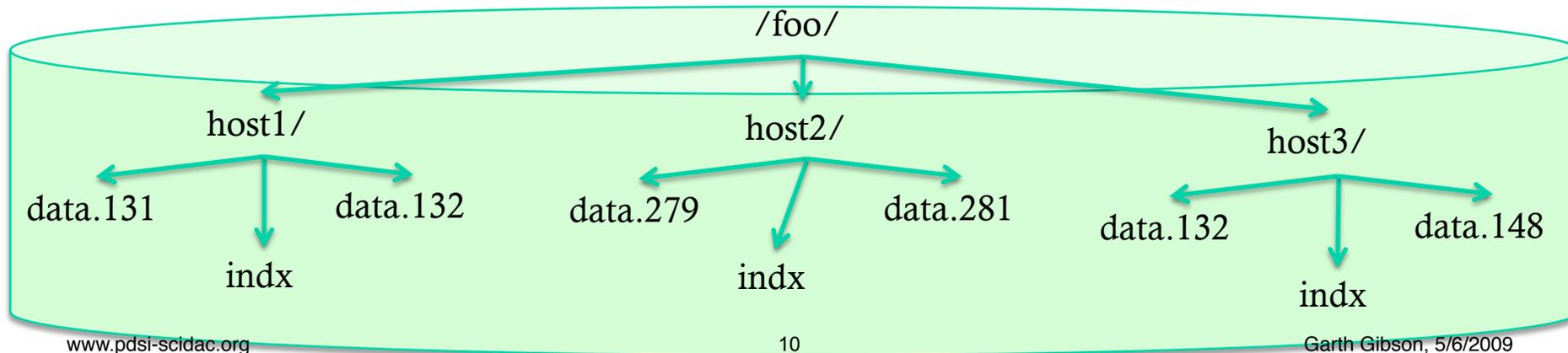
# Decoupling: PLFS at LANL



Application logical view matches PLFS virtual view



Physical storage is per-thread log of written data



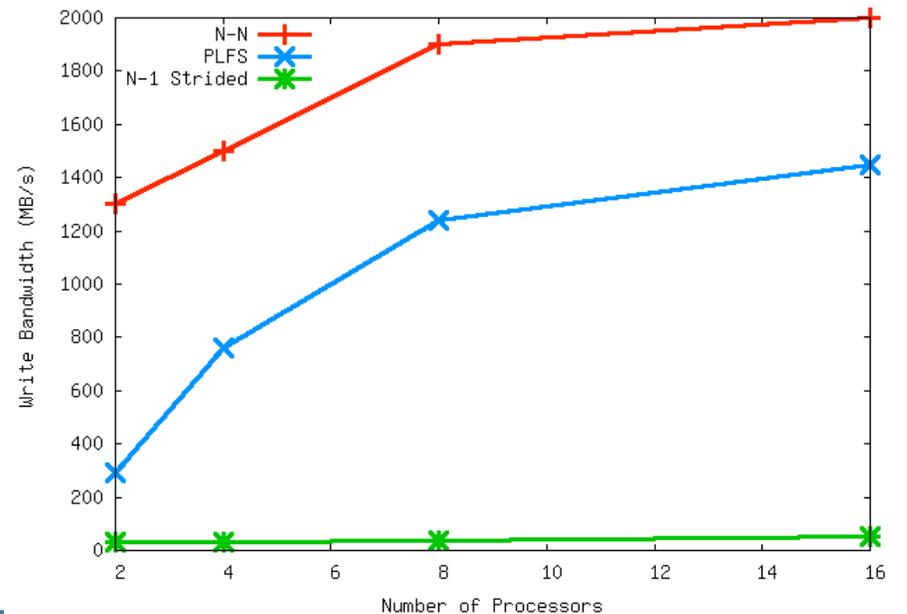
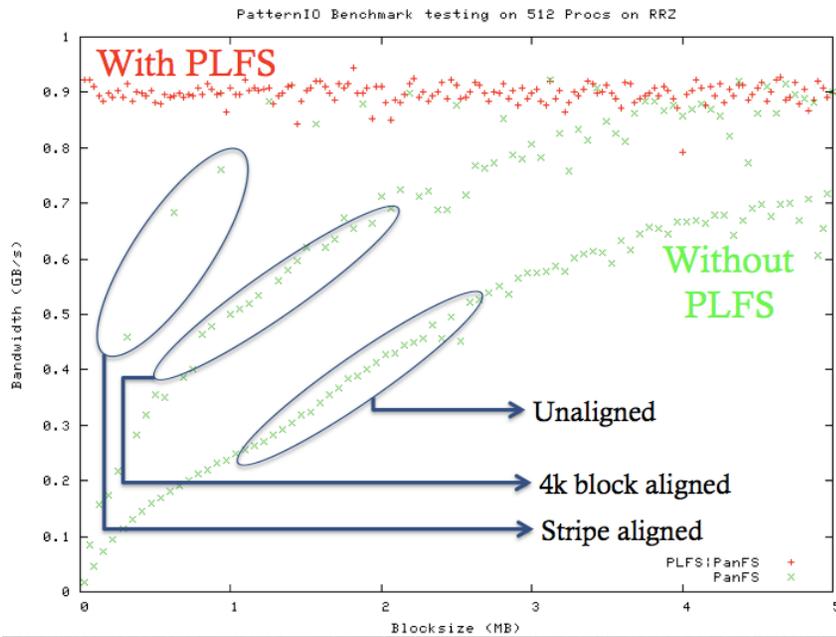
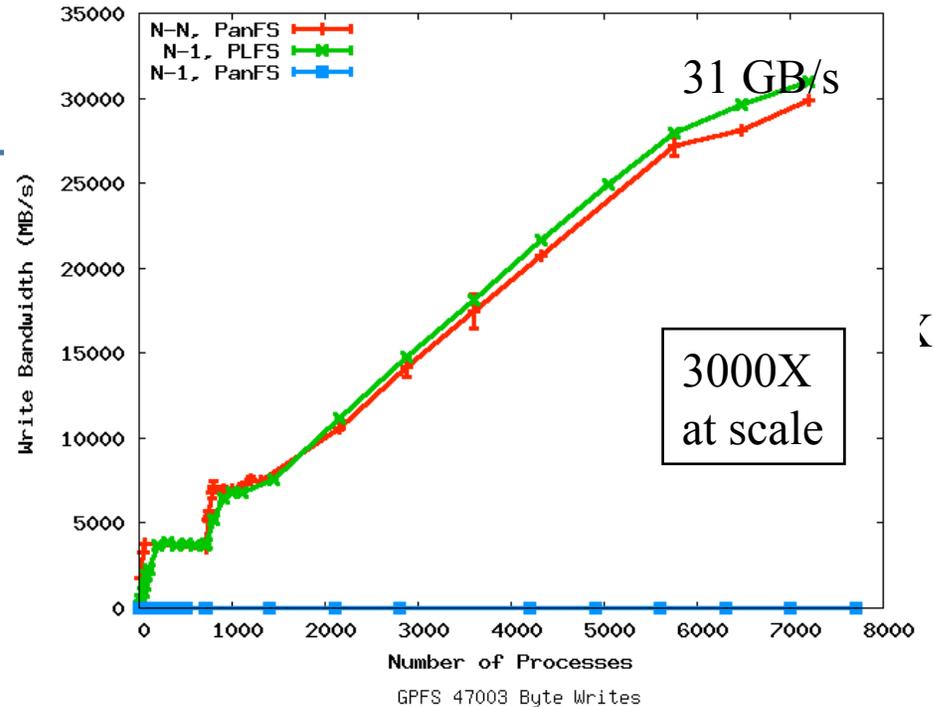
# Parallel Log-structured File System

---

- LANL:, Bent, Grider, Nunez (with CMU help)
- FUSE “interposition” transparent to app & FS
  - Apps see single file with no changes in API
  - File systems see VFS operations on many files in a directory
- Reorganizes data of one logical file into many log files, one per writing process
- Checkpoint is non-concurrent append-only writing by each process (converts N-1 writing to N-N writing)
- Index of logical/physical map in metadata files
- 3000 lines of C++
- Runs as normal user with that user’s permission

# Preliminary results

- Huge improvement transparently on PanFS & GPFS
  - Scales to 30 GB/s !
- Avoids alignment pitfalls too!



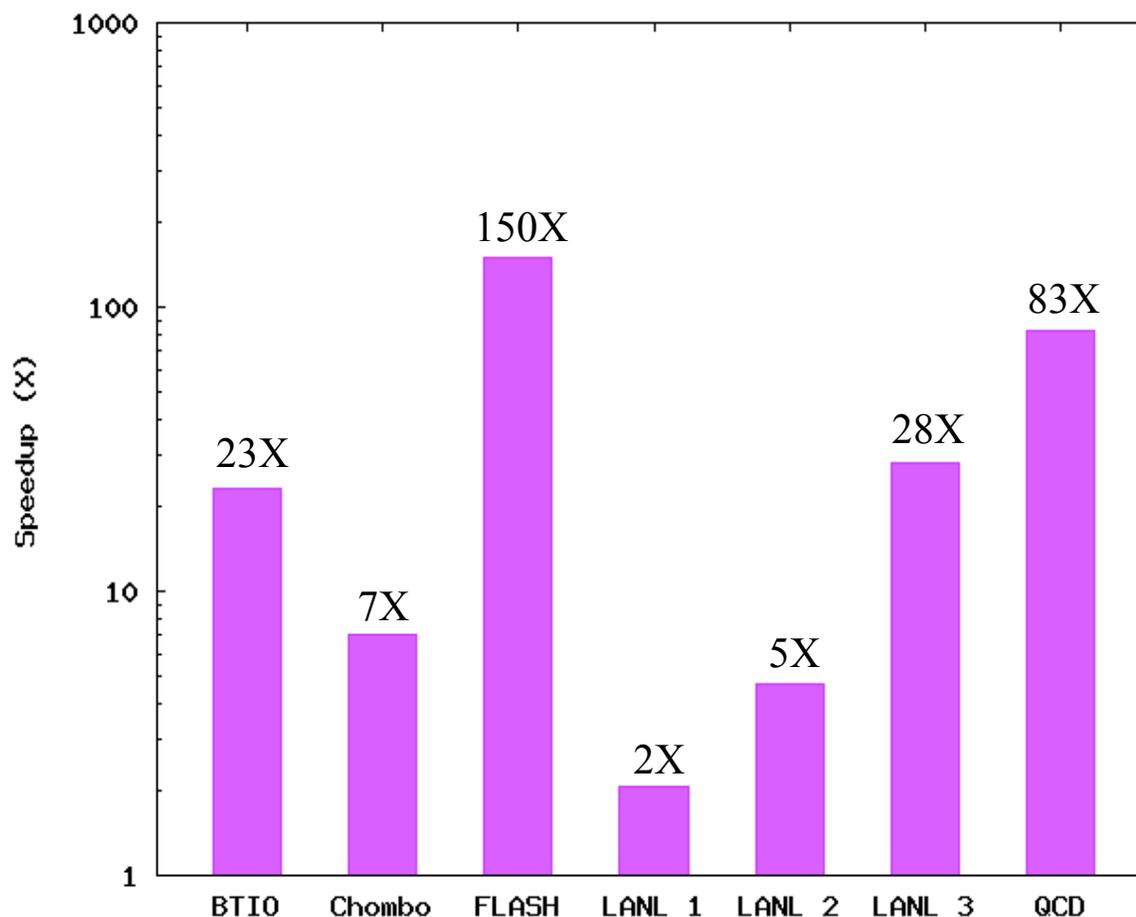
# PLFS Optimizations

---

- Reads
  - When possible (i.e. O\_RDONLY), construct global index on the open, reuse for each read call
- Stats
  - On close, create a container/metadata/host.B.L.T
    - B = blocks of capacity
    - L = last offset (i.e. file size)
    - T = timestamp of last write
  - Stat can be implemented with a readdir

# PLFS Checkpoint BW Speedups so far

- PLFS promising!
  - Transparent (FUSE)
  - Tested at scale (#1 Roadrunner)
  - 3 LANL production applications
  - ChomboIO, FlashIO, QCDIO, BTIO
    - IO skeletons of science apps
  - Working with PDSI partners to enhance and prove out
  - SC09 paper (posting asap)



# Q&A

---