SciDAC
Scientific Discovery through Advanced Computing

Office of Science

SciDAC PDSI Update (part 2)

CS/VIS PI Meeting, October 23, Germantown, MD

Garth Gibson

Carnegie Mellon University and Panasas Inc.

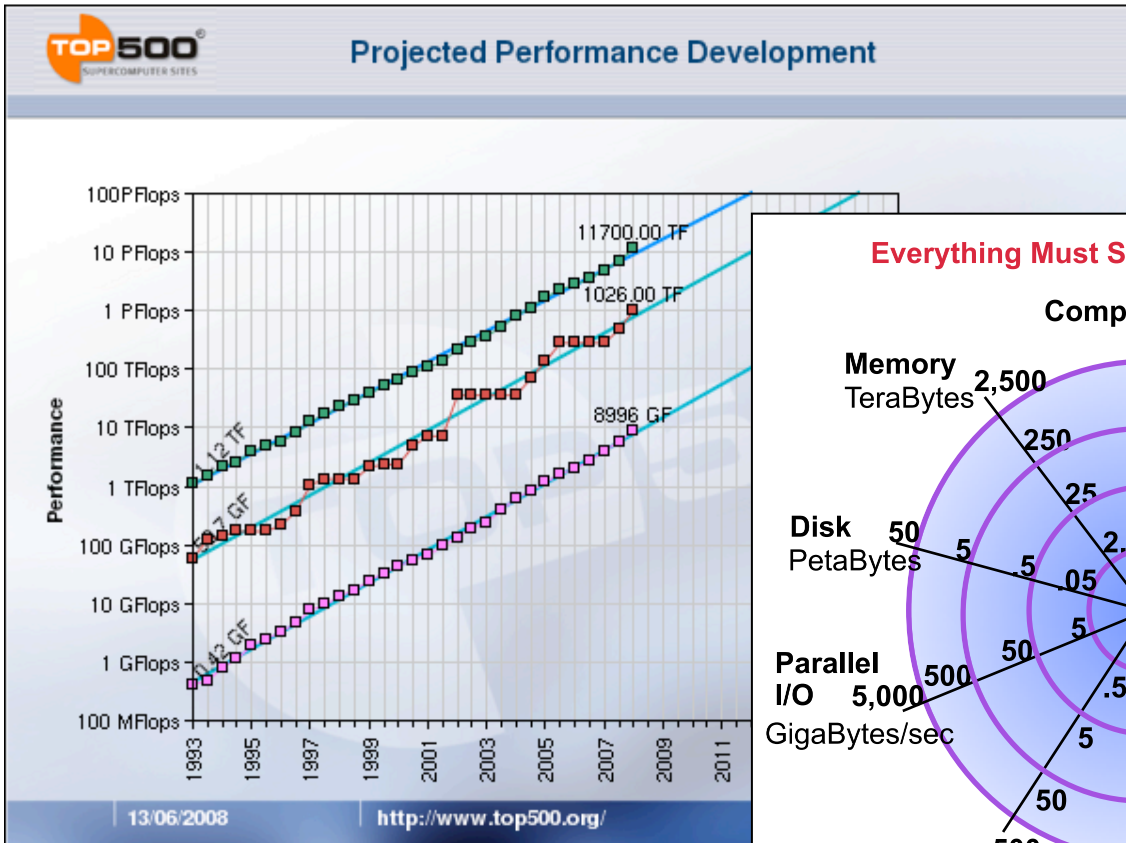
SciDAC Petascale Data Storage Institute (PDSI)

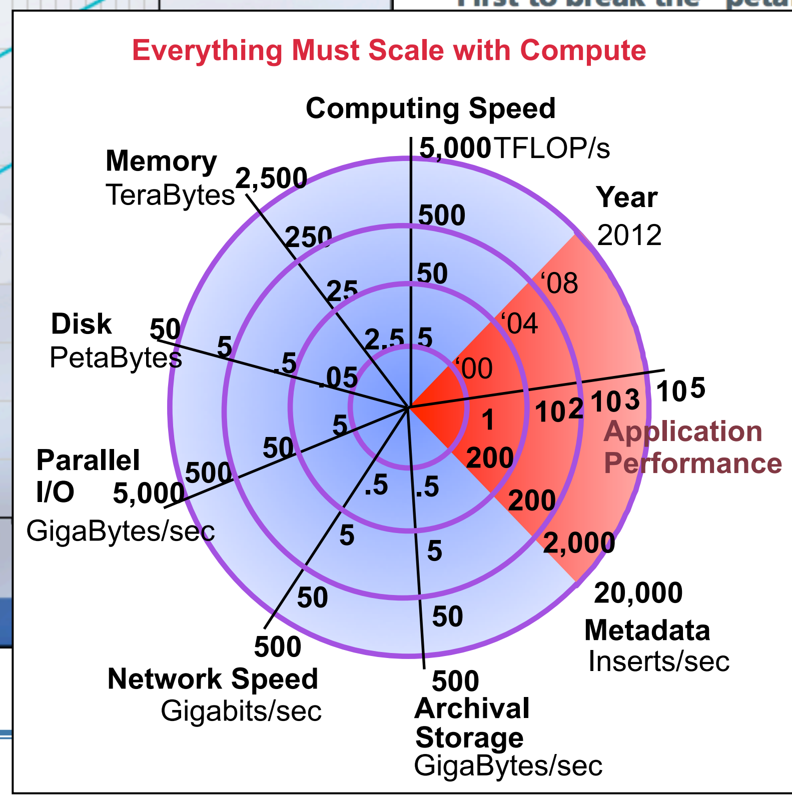
www.pdsi-scidac.org

w/ LANL (G. Grider), LBNL (W. Kramer), SNL (L. Ward),
ORNL (P. Roth), PNNL (E. Felix),
UCSC (D. Long), U.Mich (P. Honeyman)

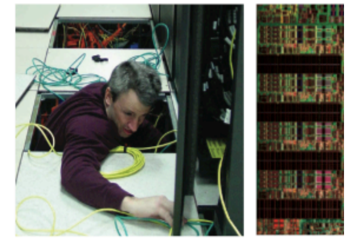
Clearing Path thru Petascale to Exascale

- Scaling 100%/yr given disk realities is hard
  - Disk BW @ 20%/yr, IO/s @ 5%/yr
  - Storage problem renews itself each year





Los Alamos NATIONAL LABORATORY EST. 1943



Roadrunner

First to break the "petaflop" barrier

...emorial Day, the ...eeded a sustained speed ...n calculations per second. ... Roadrunner more than ...r 1 system on the TOP500 ...nce to date is 74.5% ef-

- PETASCALE DATA STORAGE INSTITUTE 06-11
  - 3 universities, 5 labs, G. Gibson, CMU, PI
  - Enabling HEC storage to meet SciDAC needs
- SciDAC @ Petascale storage issues
  - Community building: ie. PDSW @ SCxy
  - APIs & standards: ie., Parallel NFS, POSIX
  - Failure data collection, analysis: ie., cfdr.usenix.org
  - Performance trace collection & benchmark publication
  - IT automation applied to HEC systems & problems
  - Novel mechanisms for core (esp. metadata, wide area)

Annual PDSI Sponsored Workshops

institutes.lanl.gov/hec-fsio/workshops/2007/

HEC FSIO '07

August

HEC FSIO R&D Workshop/HECURA FSIO PI Meeting '07 AGENDA

Workshop Location: National S

Session

Monday 8/6/2007

Welcome Review of HEC FSIO 06 outcomes, F 2007 Workshop Overview

Welcome from NSF

NSF Vision

Research Session 1 QoS

Quality of Service Guarantee for Scalable For scalable Parallel Storage Systems

End-to-End Performance Management for Large Distributed Storage

Open review of gaps/progress

LANL ISSDM and IRPIT

LANL New Data Available

Research Session 2 Measurement, Understadning, Cache Mgmt

File System Tracing, Replaying, Profiling, and Analysis on HEC Systems

Memory caching and prefetching

Open review of gaps/progress

Research Session 3 Metadata

Petascale I/O for High End Computing

Techniques for Streaming File Systems And Databases

Microdata Storage Systems for High-End Computing

SAM^2 Toolkit: Scalable and Adaptive Metadata Management for High End Computing

Open review of gaps/progress

Research Session 4 Security and Archive

Asymmetry in Performance and Security Requirements for I/O in HEC

Integrated Infrastructure for Secure and Efficient Long-Term Data Management

Open review of gaps/progress

Posters for all Day 1 talks

Tuesday 8/7/2007

Use of Xen for Testing File Systems At Scale

Research Session 5 Next Generation I/O Architectures

Deconstructing Clusters for High End Biometrics

www.pdsi-scidac.org/SC07/

Supercomputing '07

SC07

Petascale Data Storage Workshop
Session Chair: Garth Gibson, CMU

Sunday, November 11, 2007
Reno, Nevada

November

WORKSHOP ABSTRACT

Petascale computing infrastructures make petascale demands on informati and manageability. The last decade has shown that parallel file systems ca dimensions; this poses a critical challenge when near-future petascale requ the data storage problems and emerging solutions found in petascale scien community collaboration can be crucial, problem identification, workload ca shared tools.

Petascale Data Storage Workshop Introduction
Garth Gibson

SESSION I: Scalable Systems

E. Krevat (presenter), V. Vasudevan, A. Phanishayee, D. Andersen, G. Ganger, G. Gibson, S. Seshan, Carnegie Mellon University
On Application-level Approaches to Avoiding TCP Throughput Collapse in Cluster-Based Storage Systems
Paper / Slides / Poster

Lei Chai, Xiangyong Ouyang, Ranjit Noronha (presenter) and Dhabaleswar K. Panda,
Ohio State University
pNFS/PVFS2 over InfiniBand: Early Experiences
Paper / Slides

Brent Welch (presenter), Panasas, Inc.
Integrated System Models for Reliable Petascale Storage Systems
Paper / Slides

Peter Braam, Byron Neitzel (presenter), Sun/Cluster File Systems
Scalable Locking and Recovery for Network File Systems
Paper / Slides

POSTER SESSION 1 - see info below

SESSION II: Scalable Services

Jonathan Koren (presenter), Yi Zhang, Univ. of California, Santa Cruz
Searching and Navigating Petabyte Scale File Systems Based on Facets
Paper / Slides

Swapnil V. Patil (presenter), Garth A. Gibson, Sam Lang, Milo Polte, Carnegie Mellon University
GIGA+: Scalable Directories for Shared File Systems
Paper / Slides / Poster

D. Bigelow, S. Iyer, T. Kaldewey, R. Pineiro, A. Povzner, S. Brandt, R. Golding (presenter), T. Wong,C. Maltzahn, Univ. of California, Santa Cruz, IBM-Almaden
End-to-end Performance Management for Scalable Distributed Storage
Paper / Slides

Sage A. Weil (presenter), Andrew W. Leung, Scott A. Brandt, Carlos Maltzahn,
Univ. of California, Santa Cruz
RADOS: A Fast, Scalable, and Reliable Storage Service for Petabyte-scale Storage Clusters
Paper / Slides

www.pdsi-scidac.org/events/FAST08BOF/

FAST '08

February

Wednesday, February 27, 2008
Petascale Data Storage BoF Session at FAST '08

FAST '08 FEBRUARY 26–29, 2008 SAN JOSE, CALIFORNIA Sponsored by USENIX in cooperation with ACM SIGOPS, IEEE MSSTC, and IEEE TCOS USENIX

Organizer: Garth Gibson, Carnegie Mellon University and Panasas
Co-organizers: Peter Honeyman, U. Michigan/CITI; Darrell Long, U.C. Santa Cruz; Gary Grider, Los Alamos NL; Lee Ward, Sandia NL; Evan Felix, Pacific Northwestern NL; Phil Roth, Oak Ridge NL; Bill Kramer, Lawrence Berkeley NL

The Petascale Data Storage Institute is a DOE-funded collaboration of three universities and five national labs with the objective of anticipating the challenges of data storage for computing systems operating in the peta-operations per second to exa-operations per second and working toward the resolution of these challenges in the community as a whole. An important part of our agenda is outreach to other researchers and practitioners to share our resources and gather better understanding of the petascale issues ahead from all.

In this BOF we will:
1) Introduce the Petascale Data Storage Institute (PDSI),
2) Advertise PDSI gathered and released sources of useful data, including
 - data sets of node and storage failures in large scale computing
 - file access traces of non-trivial petascale computing applications
 - collections of file systems statistics gathered from petascale computing systems and other systems,
3) Discuss requirements for one or more petascale data storage systems and applications, and
4) Lead an open discussion of these and other issues for large scale data storage systems.

PRESENTATIONS

PDSI FAST 2008 BOF Introduction - Garth Gibson, CMU

The Computer Failure Data Repository (CFDR) - Bianca Schroeder, University of Toronto

File System Statistics - Shobhit Dayal, CMU, Garth Gibson, CMU, Marc Unangst, Panasas

PNNL – Petascale Data Storage Institute Data release Update - Evan Felix, PNNL

NERSC Reliability Data - Bill Kramer, Jason Hick, Akbar Mokhtarani, NERSC

LANL SciDAC Petascale Data Storage Institute Operational Data Releases - James Nunez, Gary Grider, John Bent, HB Chen, Meghan Quist, Alfred Torrez, Los Alamos National Lab

Ceph: An Open-Source Petabyte-Scale File System - Ethan Miller, Storage Systems Research Center, UCSanta Cruz

Special Presentation on HPC User Requirements:
I/O Requirements for HPC Applications: A User Perspective
John Shalf, National Energy Research Scientific Computing Center (NERSC), LBNL

PDSI POSTER AT THE FAST '08 POSTER SESSION

PDSI Data Releases and Repositories

PDSW07 papers published & online

THE ACM DIGITAL LIBRARY　Feedback

Conference on High Performance Networking and Computing archive

Proceedings of the 2nd international workshop on Petascale d
2007, Reno, Nevada　*November 11 - 11, 2007*

Additional Information: full citation

Conference Chair Garth A. Gibson Carnegie Mellon University and Panasas Inc.

Front matter
pdf
Front matter (Title page, TOC, Committee, Author index)

Table of Contents

http://www.pdsi-scidac.org/events/SC07/index.html

PDSI: Petascale Data Sto… | ft:pio:start [FT Wiki] | Petascale Data Storage I… | TOC

9:00am - 10:20am　**SESSION I: Scalable Systems**

E. Krevat (presenter), V. Vasudevan, A. Phanishayee, D. Andersen, G. Ganger, G. Gibson, S. Seshan, Carnegie Mellon University
On Application-level Approaches to Avoiding TCP Throughput Collapse in Cluster-Based Storage Systems
Paper / Slides / Poster

Lei Chai, Xiangyong Ouyang, Ranjit Noronha (presenter) and Dhabaleswar K. Panda,
Ohio State University
pNFS/PVFS2 over InfiniBand: Early Experiences
Paper / Slides

Brent Welch (presenter), Panasas, Inc.
Integrated System Models for Reliable Petascale Storage Systems
Paper / Slides

Peter Braam, Byron Neitzel (presenter), Sun/Cluster File Systems
Scalable Locking and Recovery for Network File Systems
Paper / Slides

10:30am - 11:00am　**POSTER SESSION 1 - see info below**

11:00am - 12:20pm　**SESSION II: Scalable Services**

Jonathan Koren (presenter), Yi Zhang, Sasha Ames, Andrew Leung, Carlos Maltzahn, Ethan Miller, Univ. of California, Santa Cruz
Searching and Navigating Petabyte Scale File Systems Based on Facets
Paper / Slides

Swapnil V. Patil (presenter), Garth A. Gibson, Sam Lang, Milo Polte, Carnegie Mellon University
GIGA+: Scalable Directories for Shared File Systems
Paper / Slides / Poster

D. Bigelow, S. Iyer, T. Kaldewey, R. Pineiro, A. Povzner, S. Brandt, R. Golding (presenter), T. Wong, C. Maltzahn, Univ. of California, Santa Cruz, IBM-Almaden
End-to-end Performance Management for Scalable Distributed Storage
Paper / Slides

Petascale Data Storage Workshop 08

- Monday Nov 17, 8:30-5, room 14, SC08
- www.pdsi-scidac.org/events/PDSW08
- IEEE Digital Library publication
- Tentative program
  - Input/Output APIs and Data Organization for High Performance Scientific Computing
  - Fast log-based concurrent writing of checkpoints
  - Scalable Full-Text Search for Petascale File Systems
  - Zest: Reliable Terabytes Per Second Storage for Petascale Systems
  - Performance of RDMA-capable Storage Protocols on Wide-Area Network
  - Comparing performance of solid state devices and mechanical disks
  - Arbitrary Dimension Reed-Solomon Coding & Decoding for Extended RAID on GPUs
  - Pianola: A script-based I/O benchmark
  - Introducing Map-Reduce to High End Computing
  - Logan: Automatic Management for Evolvable, Large-Scale, Archival Storage
  - Just-in-time Staging of Large Input Data for Supercomputing Jobs
  - Revisiting the Metadata Architecture of Parallel File Systems

pNFS: Scalable NFS Standard & Code Soon

From: Tigran Mkrtchyan <tigran.mkrtchyan@desy.de>
Date: July 16, 2008 4:18:13 AM PDT
To: pnfs@linux-nfs.org
Subject: [pnfs] pnfs becomes real!

today we ran the first real physics analysis job using dCache-pnfs server and linux pnfs client:

tigran@nairi:~/work/linux-pnfs> git show | head -5
commit 6ae52464ba2c77f1bf2365e415305dfd9b51dd20
Author: Benny Halevy <bhalevy@panasas.com>
Date: Tue Jul 15 20:22:51 2008 +0300

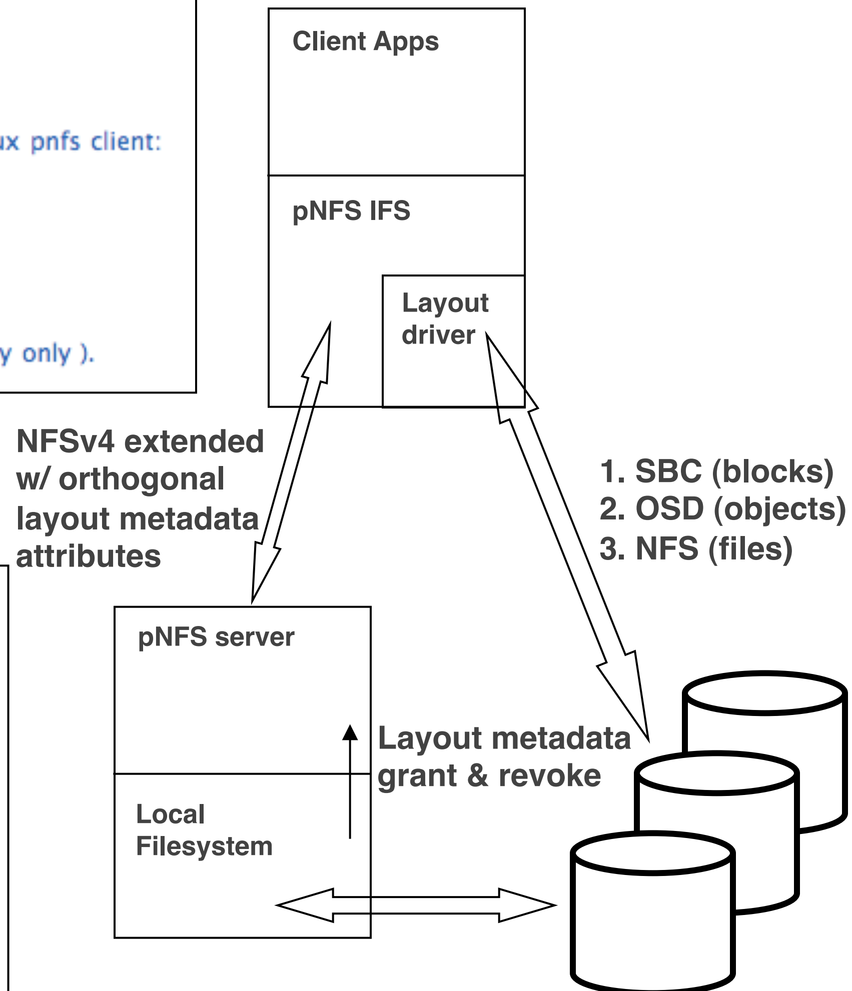
Anyway, fist time we can show that NFSv4.1 is something real (and not my hobby only).

- Open source & competitive offerings!
- NetApp, Sun, IBM, EMC, Panasas ….

From: Spencer Shepler <Spencer.Shepler@Sun.COM>
Date: August 1, 2008 4:34:46 PM GMT-04:00

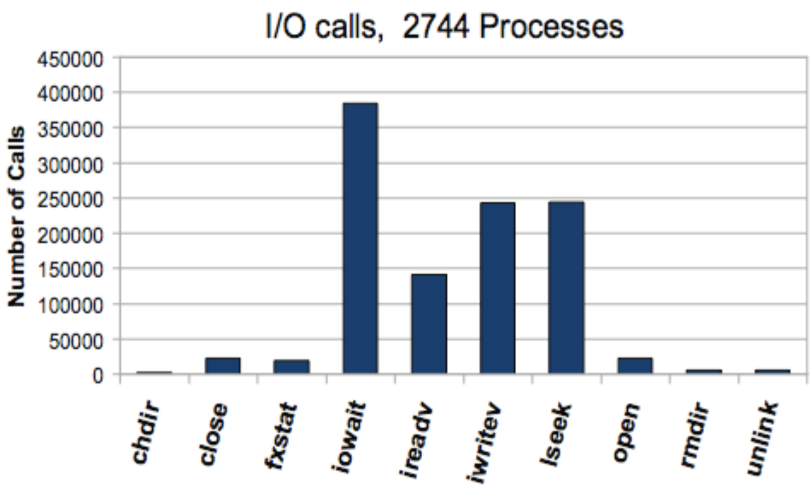
2. IETF status

All of the current working group internet drafts are moving forward for publication. This means that they have submitted to the area director and will start their way through the process (IETF last call and IESG review).



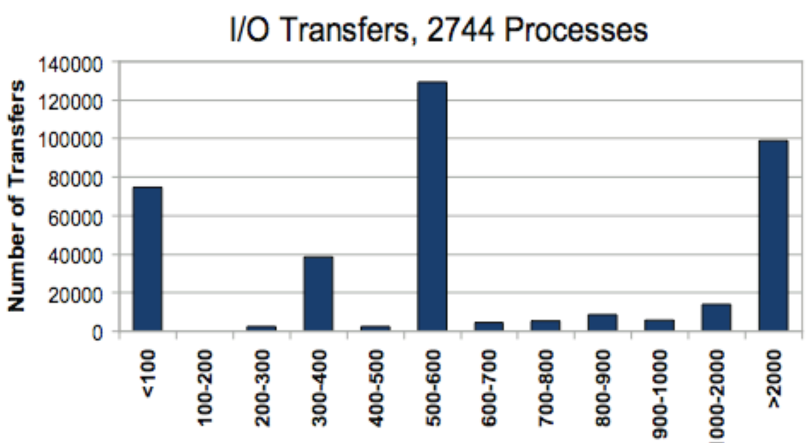
- SC-08 BOF Wed Nov 19 5:30pm

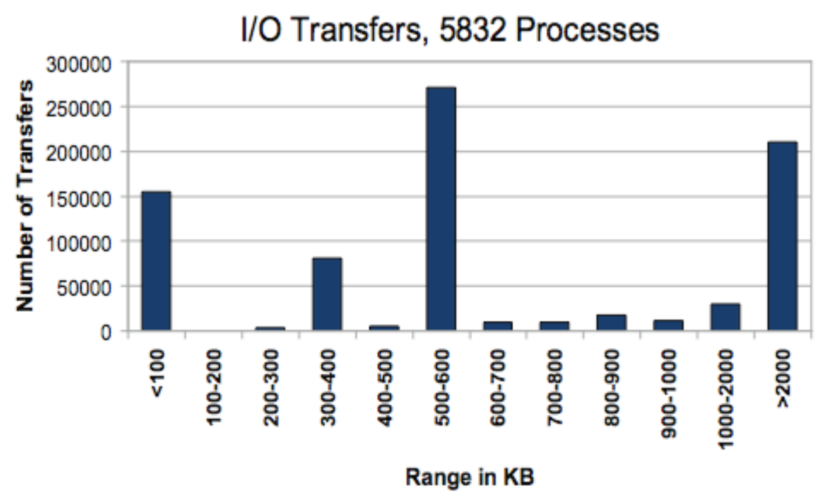
Tools for Understanding IO in Apps



NEWEST TRACE DATA, REDSTORM, SANDIA NAT'L LAB

- A physics simulation problem for a common Sandia application, Alegra
  - Runs were performed alongside regular user runs
- Each run generated 4 restart dumps, and ran for 20 simulation cycles
- Both single core per node, and 2 core (virtual node mode) per node
  - Repeated with and without tracing enabled
- The single core per node jobs ran at a client size of 2744 processes
  - Non-tracing elapsed run time 10:42 minutes
  - Tracing elapsed run time 11:07 minutes
- The 2 core per node jobs ran at 2916 nodes, 5832 processes.
  - Non-tracing elapsed run time 15:52 minutes
  - Tracing elapsed run time 16:37 minutes
- Raw trace file sizes 30K-50K per MPI rank, except rank zero (600KB-700KB)
  - Rank 0 I/O to terminal records progress in the job.



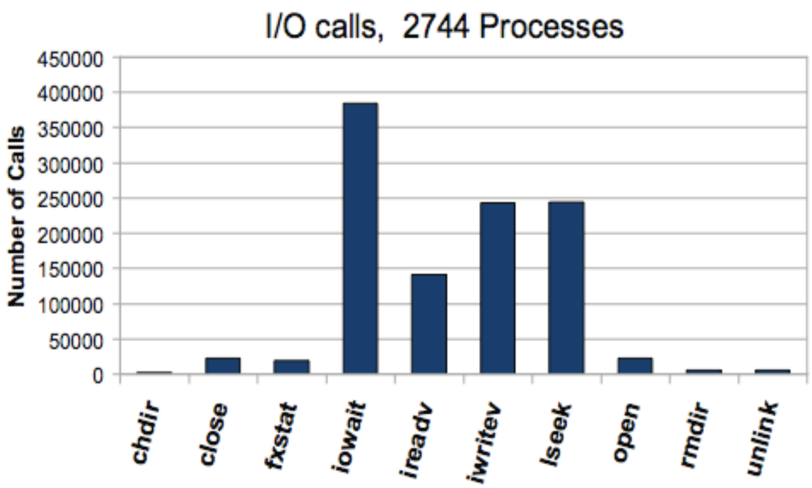


Sandia National Laboratories

sourceforge.net/projects/libsysio

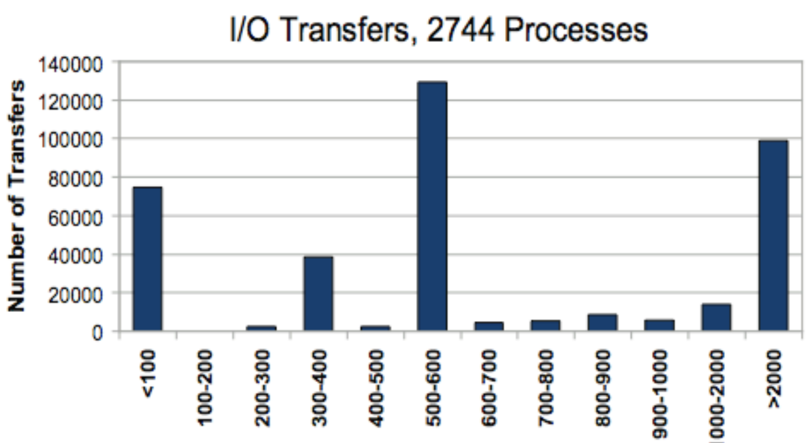
pdsi

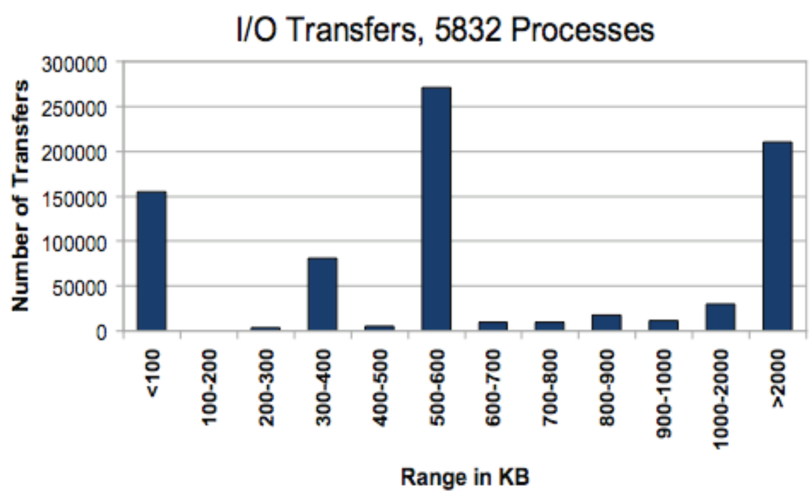
PDSI distributes parallel workloads



NEWEST TRACE DATA, REDSTORM, SANDIA NAT'L LAB

- A physics simulation problem for a common Sandia application, Alegra
  - Runs were performed alongside regular user runs
- Each run generated 4 restart dumps, and ran for 20 simulation cycles
- Both single core per node, and 2 core (virtual node mode) per node
  - Repeated with and without tracing enabled
- The single core per node jobs ran at a client size of 2744 processes
  - Non-tracing elapsed run time 10:42 minutes
  - Tracing elapsed run time 11:07 minutes
- The 2 core per node jobs ran at 2916 nodes, 5832 processes.
  - Non-tracing elapsed run time 15:52 minutes
  - Tracing elapsed run time 16:37 minutes
- Raw trace file sizes 30K-50K per MPI rank, except rank zero (600KB-700KB)
  - Rank 0 I/O to terminal records progress in the job.





Sandia National Laboratories

sourceforge.net/projects/libsysio

pdsi

PDSI distributes parallel workloads

MPI-IO Test

Although there are a host of existing file system and I/O test programs
most are not designed with parallel I/O in mind and are not useful at the
the clusters at Los Alamos National Lab (LANL). LANL's MPI-IO Test was
with parallel I/O and scale in mind. The MPI-IO test is built on top of MPI
and is used to gather timing information for reading from and writing to
using a variety of I/O profiles; N processes writing to N files, N processe
to one file, N processes sending data to M processes writing to M files, o
processes sending data to M processes to one file. These diagrams illust
various I/O access patterns. A data aggregation capability is available ar
can pass down MPI-IO, ROMIO and file system specific hints. The MPI-IO
be used for performance benchmarking and, in some cases, to diagnose
with file systems or I/O networks.

The MPI-IO Test is open sourced under LA-CC-05-013.

Release	Date	Source	Docume
1.000.21	8 July 2008	mpi_io_test_21.tgz	READM
1.000.20	13 November 2007	mpi_io_test_20.tgz	READM
1.000.09	15 December 2006	mpi_io_test_09.tgz	READM
1.000.08	2 March 2006	mpi_io_test_08.tgz	READM

MPI_IO_TEST traces

These traces were collected using LANL-Trace (V 1.0.0) on the LANL MPI_IO test (V 1.00.020) application. These traces are all from system data machine number 25 on this computer systems table. Here is the README and FAQ that explains how LANL-Trace works and what the output files look like:
TRACE README,
TRACE FAQ.

N-to-N

	64 KB	256 KB	448 KB	512 KB	1024 KB	4096 KB	8192 KB	16386 KB	32772 KB	65544 KB
32 Procs		TGZ	TGZ	TGZ	TGZ	TGZ	TGZ	TGZ	TGZ	TGZ
96 Procs		TGZ	TGZ	TGZ	TGZ	TGZ		TGZ	TGZ	TGZ

N-to-1 nonstrided

	64 KB	256 KB	448 KB	512 KB	1024 KB	4096 KB	8192 KB	16386 KB	32772 KB	65544 KB
32 Procs	TGZ		TGZ	TGZ	TGZ	TGZ	TGZ	TGZ	TGZ	TGZ
96 Procs	TGZ	TGZ		TGZ	TGZ	TGZ		TGZ	TGZ	TGZ

N-to-1 strided

	64 KB	256 KB	448 KB	512 KB	1024 KB	4096 KB	8192 KB	16386 KB	32772 KB	65544 KB
32 Procs	TGZ	TGZ	TGZ	TGZ	TGZ	TGZ	TGZ	TGZ	TGZ	TGZ
96 Procs	TGZ	TGZ	TGZ	TGZ	TGZ	TGZ		TGZ	TGZ	TGZ

PDSI distributes parallel workloads

MADBench: Microwave Anisotropy Dataset Computational Analysis Package Benchmark
The benchmark code MADBench is a "stripped-down" version of MADCAP, a Microwave Anisotropy Dataset Computational Analysis Package ...more>>>

IPM benchmarks: Medium, Large and X-large datasets.

MILC: MIMD Lattice Computation
The benchmark code MILC represents part of a set of codes written by the MIMD Lattice Computation (MILC) collaboratoration used to study quantum chromodynamics (QCD), the theory of the strong interactions of subatomic physics ...more>>>

IPM benchmarks: Medium and Large datasets.

PMEMD: Particle Mesh Ewald Molecular Dynamics
The benchmark code PMEMD (Particle Mesh Ewald Molecular I
Dynamics (MD), NMR Refinement and minimizations ...more>

IPM benchmarks: Medium and Large datasets

IO Benchmarks with IPM*

The new version of IPM integrates the standard POSIX IO cal
runs are made with this new feature on Jacquard (courtesy o

MADBench:

- 256 tasks, POSIX one file per task [plots] [stats]
- 64 tasks, POSIX one file per task [plots] [stats]
- 16 tasks, POSIX shared file [plots] [stats]

Chombo:

- 256 tasks, 2 components [plots] [stats]
- 32 tasks, 2 components [plots] [stats]
- 32 tasks, 10 components [plots] [stats]

AMRScalingXfer: 128 tasks, small run [plots] [stats]

*Note: This is development software, and the runs/plots arer
profiling in IPM.

I/O Benchmark and Characterization Links:

I/O Performance for HPC Platform using IOR PDF ppt
This study analyzes the I/O practices and requirements of current HPC applications and use them as criteria to select a subset of microbenchmarks that reflect workload requirements.

FLASH I/O Benchmarck PDF
This code from 'The Center for Astrophysical Thermonuclear Flashes' can test either HDF5, Parallel NetCDF, or a direct Fortran write. The I/O bencmarks are compared for Seaborg and Bassi systems

Performance Effect of Multi-core on Scientific Applicationa (PDF) paper slides
Presents performance measurements of several complete scientific applications on single and dual core Cray XT3 and XT4 systems.

MADBench - IPM of a Cosmology Application on Leading HEC Platforms PDF
Presents MADBench, a lightweight version of MADCAP CMB power spectrum estimation code, and uses the Integrated Performance Monitoring (IPM) package to extract MPI message-passing overheads

MADBench2 PDF
Presents I/O analyses of modern parallel filesystems and examines a broad range of system architectures and configurations. It also describes use of Luster striping to improve concurrent file access performance.

Effective I/O Bandwidth Benchmark PDF
This paper describes the design and implementation of a parallel I/O benchmark useful for comparing filesystem performance on a variety of architectures, including, but not limited to cluster systems.

Effcient Parallel I/O on thee Cray XT3/XT4 PDF
Provides an overview of I/O methods for three different applications

Trace Data

Here are files containing trace data for some of the applications. These traces are generated by invoking the "strace" utility on every task and piping the data for each task to a separate file. Process ID is used to create unique file names. All applications where run on Jacquard . The files are compressed tar files of the trace data

PMEMD 16 tasks small dataset run

MADbench 64 tasks medium dataset run

MILC 16 tasks medium dataset run

pdsi @ NERSC
NATIONAL ENERGY RESEARCH SCIENTIFIC COMPUTING CENTER

PDSI distributes convenient packages

http://www.pdsi-scidac.org/repository/ubuntu/index.html

PDSI: P... | USENI... | PDSI: P... | Static ... | PDSI: P... | Los Al... | PDSI a... | ft:pio:s...

How do I get pvfs installed on my system?

Follow the instructions above for installing the pdsi ubuntu repository on your ubuntu system. Then there are several packages to choose from to install pvfs depending on what you want to do with the box. Currently pvfs isn't built to use infiniband networking, however, you can run ip over infiniband and use these packages.

- karma - gtk gui for looking at a pvfs filesystem graphically
- libpvfs - library for pvfs
- libpvfs-dev - libraries and headers for developing with pvfs
- pvfs-fs-utils - pvfs file system utils for talking directly to pvfs
- pvfs-server - pvfs server binary
- pvfs-server-utils - pvfs helper utils for generating configs, pinging the servers, etc.
- pvfs-source - source for the pvfs client driver
- pvfs-modules-2.6.21-1-686 - pvfs modules and client for Linux (kernel 2.6.21-1-686)

Then simply:

```
apt-get install pvfs-server pvfs-modules-2.6.21-1-686
```

NOTE: The architecture used in this example is a 686 compatable box. So if you are using amd64 or ia64 please replace 686 with the appropriate architecture.

How do I get lustre installed on my system?

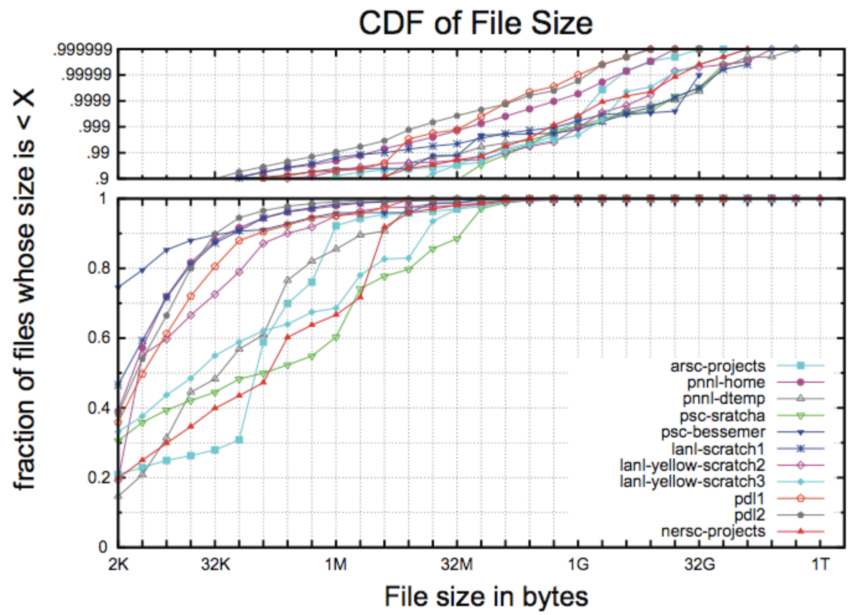
Follow the instructions above for installing or for unstable for installing the pdsi ubuntu repository on your ubuntu system. Then there are several packages to choose from to install lustre depending on what you want to do with the box. Currently lustre isn't built to use infiniband or quadrics, however you can run IP over both quadrics and infiniband and still use these packages.

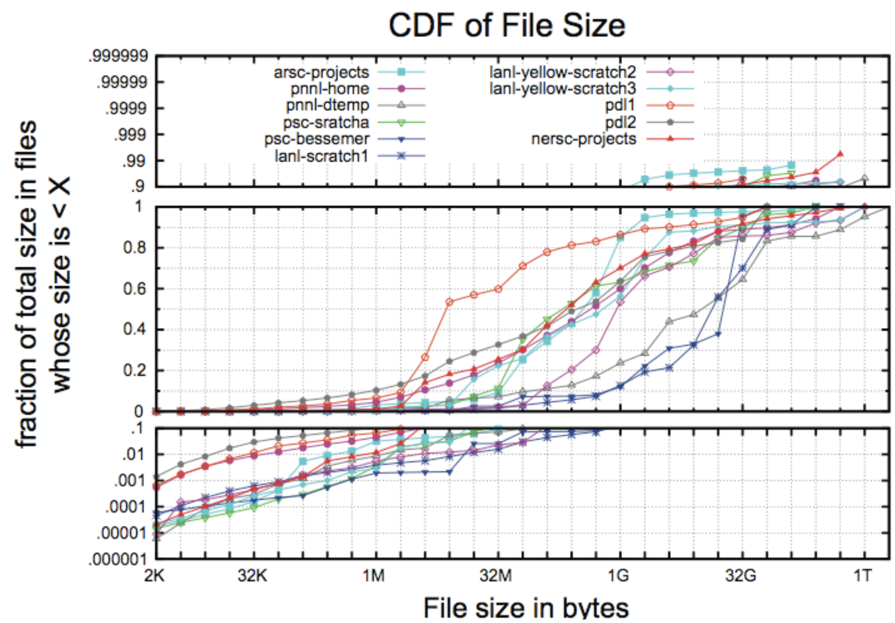
- linux-doc-2.6.18-4-lustre-686 - linux kernel specific documentation for version 2.6.18-4-lustre-686
- linux-headers-2.6.18-4-lustre-686 - common header files for linux 2.6.18-4-lustre on 686
- linux-image-2.6.18-4-lustre-686 - linux 2.6.18-4-lustre image on 686
- linux-manual-2.6.18-4-lustre-686 - linux kernel section 9 manual pages for version 2.6.18-4-lustre
- linux-patch-lustre - linux kernel patch for the lustre filesystem
- linux-source-2.6.18-4-lustre-686 - linux kernel source for version 2.6.18 with debian and lustre patches
- lustre-dev - development files for the Lustre filesystem
- lustre-modules-2.6.18-4-lustre-686 - lustre filesystem driver modules for linux 2.6.18 on 686
- lustre-source - source for lustre filesystem client kernel modules
- lustre-utils - userspace utilities for the lustre filesystem
- liblustre - liblustre library for the lustre filesystem

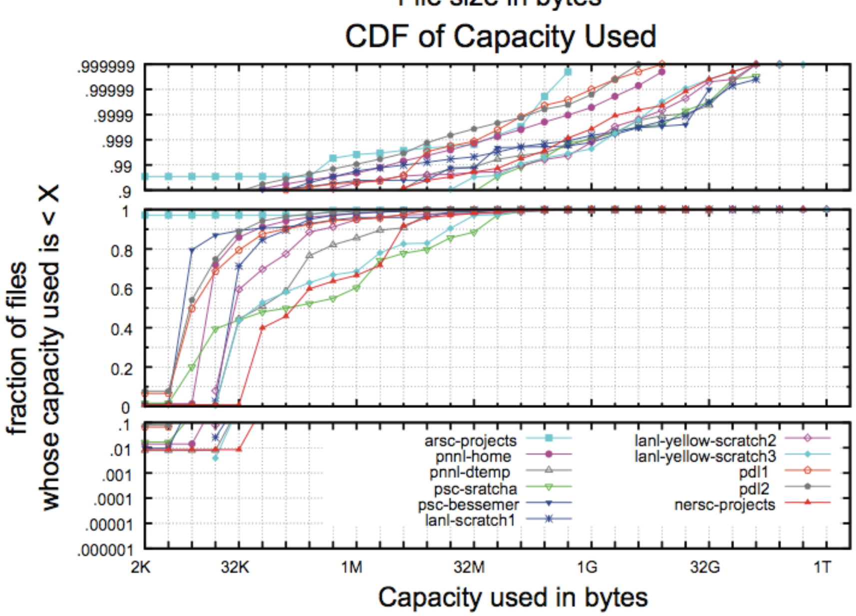
Then simply:

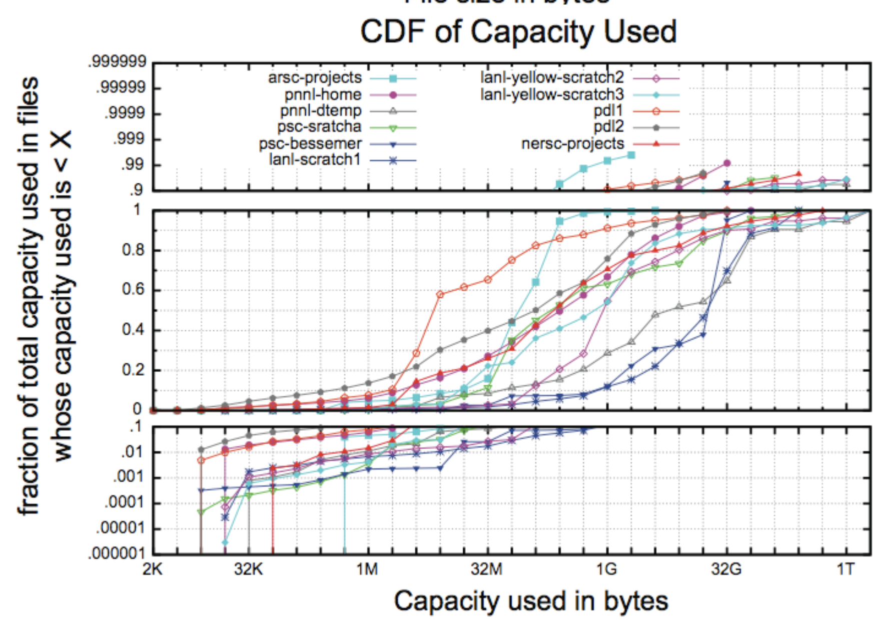
```
apt-get install linux-image-2.6.18-4-lustre-686 lustre-utils lustre-modules-2.6.18-4-lustre-686
```

Newest: File Statistics









www.pdsi-scidac.org/fsstats

Results

Uploaded File in CSV format	Organization	Date	Data Size	System Name	Form Questions	Formatted Result	Graphs	Graphs	fsstats Version
BradHavel_nanu1.csv	ARSC	Mar122008	69TB	SAMQFS	Form	Histograms	PS	PDF	1.4
BradHavel_seau2.csv	ARSC	Mar132008	115TB	SAMQFS	Form	Histograms	PS	PDF	1.4
BradHavel_seau1.csv	ARSC	Mar132008	305TB	SAMQFS	Form	Histograms	PS	PDF	1.4
BradHavel_nanprojects.csv	ARSC	Mar132008	32TB	SAMQFS	Form	Histograms	PS	PDF	1.4
JamesNunez_panscratch1.csv	LANL	Apr012008	9TB	Panasas	Form	Histograms	PS	PDF	1.4
JamesNunez_yellowscratch2.csv	LANL	Apr042008	25TB	Panasas	Form	Histograms	PS	PDF	1.4
JamesNunez_yellowscratch3.csv	LANL	Apr042008	26TB	Panasas	Form	Histograms	PS	PDF	1.4
AkbarMokhtarani_NGFfsstats.csv	NERSC	Apr082008	107TB	GPFS	Form	Histograms	PS	PDF	1.4
PhilRoth_fsstats.csv	ORNL	Oct102007	305GB	Panasas	Form	Histograms	PS	PDF	1.4
MichaelStroucken_pdl2.csv	PDL	Apr092008	1TB	WAFL	Form	Histograms	PS	PDF	1.4
MichaelStroucken_pdl1.csv	PDL	Apr092008	4TB	WAFL	Form	Histograms	PS	PDF	1.4
EvanFelix_dtemp.csv	PNNL	Mar172008	23TB	Lustre	Form	Histograms	PS	PDF	1.4
EvanFelix_nwfs.csv	PNNL	Mar172008	265TB	Lustre	Form	Histograms	PS	PDF	1.4
EvanFelix_home.csv	PNNL	Mar172008	5TB	ADVFS	Form	Histograms	PS	PDF	1.4
EvanFelix_mpp2dtemp.csv	PNNL	Oct102007	12TB	ext3	Form	Histograms	PS	PDF	1.4
EvanFelix_nwfs.csv	PNNL	Oct102007	233TB	ext3	Form	Histograms	PS	PDF	1.4
EvanFelix_mpp2home.csv	PNNL	Oct102007	4TB	advfs	Form	Histograms	PS	PDF	1.4
Katie_scratch1.csv	PSC	Mar272008	32TB	Lustre	Form	Histograms	PS	PDF	1.4
Katie_bessemer1.csv	PSC	Mar272008	4TB	Lustre	Form	Histograms	PS	PDF	1.4

A Comparative graph of some of the above results EPS
A Comparative graph of some of the above results PDF
A Comparative graph of archival file system EPS
A Comparative graph of archival file systems PDF

PDSI Targeted Apps

- P. Roth (ORNL) led
  - Parallel Ocean Program (POP) with PERI
  - Turbulent Combustion (S3D) with PERI
  - IO characterization and modeling
- L. Ward (SNL) led
  - Climate Change (CCSM) with M. Taylor
  - Trace-based performance debugging
- G. Gibson (CMU) starting
  - Astrophysics (Flash), reaching out to P. Hovland

Carnegie Mellon
Parallel Data Laboratory

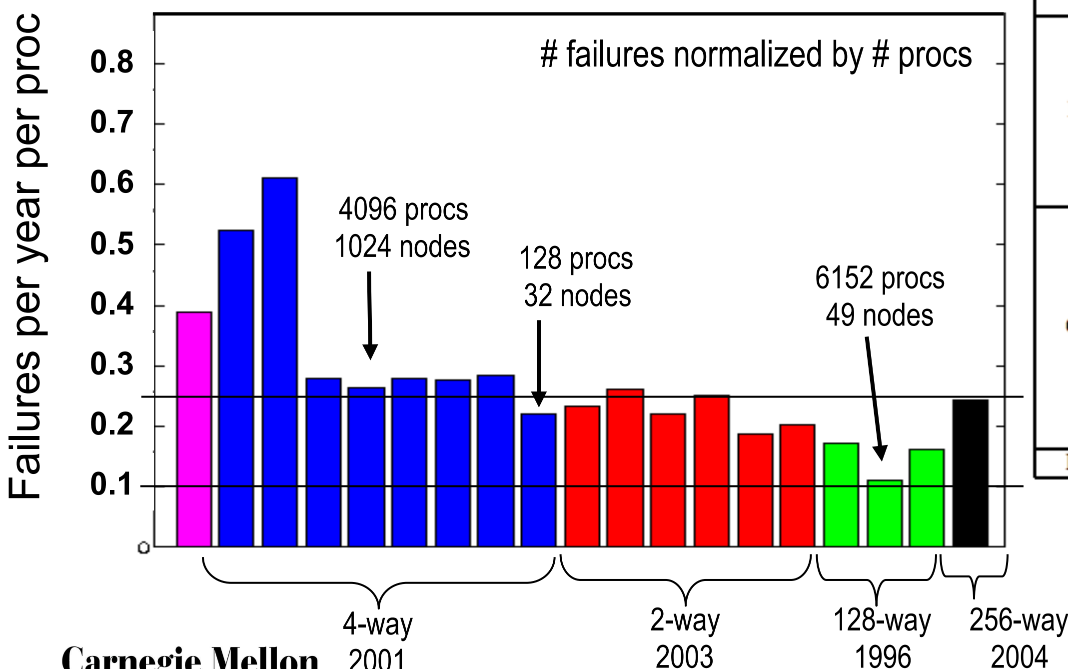
Sandia National Laboratories

Oak Ridge National Laboratory

pdsi

Failure Data Collection

- Los Alamos root cause logs
  - 22 clusters & 5,000 nodes
  - covers 9 years & continues
  - cfdr.usenix.org publishes this and many other failure datasets



(I) High-level system information				(II) Information per node category			
HW	ID	Nodes	Procs	Procs /node	Production Time	Mem (GB)	NICs
A	1	1	8	8	N/A – 12/99	16	0
B	2	1	32	32	N/A – 12/03	8	1
C	3	1	4	4	N/A – 04/03	1	0
D	4	164	328	2 2	04/01 – now 12/02 – now	1 1	1 1
E	5	256	1024	4	12/01 – now	16	2
	6	128	512	4	09/01 – 01/02	16	2
	7	1024	4096	4 4 4 4	05/02 – now 05/02 – now 05/02 – now 05/02 – now	8 16 32 352	2 2 2 2
	8	1024	4096	4 4 4	10/02 – now 10/02 – now 10/02 – now	8 16 32	2 2 2
	9	128	512	4	09/03 – now	4	1
	10	128	512	4	09/03 – now	4	1
	11	128	512	4	09/03 – now	4	1
	12	32	128	4 4	09/03 – now 09/03 – now	4 16	1 1
F	13	128	256	2	09/03 – now	4	1
	14	256	512	2	09/03 – now	4	1
	15	256	512	2	09/03 – now	4	1
	16	256	512	2	09/03 – now	4	1
	17	256	512	2	09/03 – now	4	1
	18	512	1024	2 2	09/03 – now 03/05 – 06/05	4 4	1 1
G	19	16	2048	128 128	12/96 – 09/02 12/96 – 09/02	32 64	4 4
	20	49	6152	128 128 80	01/97 – now 01/97 – 11/05 06/05 – now	128 32 80	12 12 0
	21	5	544	128 32 128 128	10/98 – 12/04 01/98 – 12/04 11/02 – now 11/05 – 12/04	128 16 64 32	4 4 4 4
H	22	1	256	256	11/04 – now	1024	0

Table 1. *Overview oj SMP-based, and systen*

Los Alamos NATIONAL LABORATORY EST. 1943

Revisiting checkpoint: Log representation

- Fastest checkpoint just a series of “variable=value” (ie. PSC Zest)
  - Instead of seeking to serialized location, just append operation to log
  - Each thread writes strictly sequential log of operations
  - “Meaning” of set of logs is applying log to (possibly null) initial database
- Prior: Gatech/ORNL ADIOS, ANL summer project
  - Discussing with SciDAC SDM on application to pHDF5/netCDF
- Embed in storage software as general, transparent service
  - Optimize writing and reading representations separately
  - Defer serializing by just storing changelogs for later application
  - Some checkpoints never read, so never serialized
  - If read before serialized, trigger serialization (or something smarter)
- Opens up embedded indices (FastBit, B-trees, database)

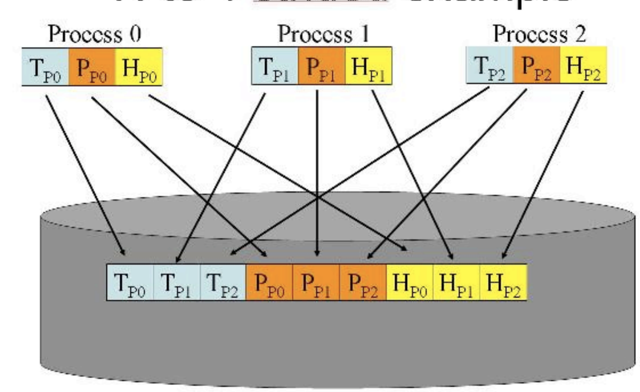
CMU class project: log-structured PVFS files

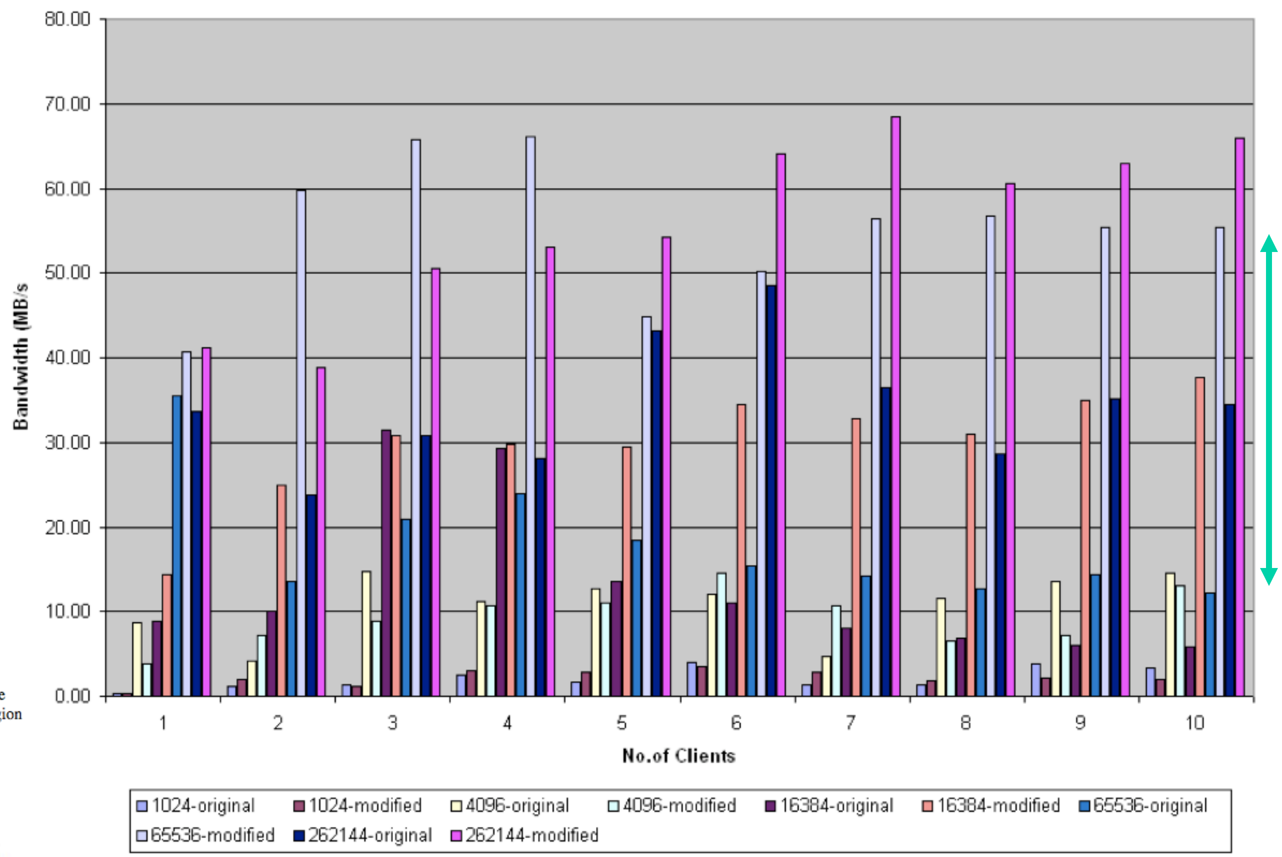
- HPC checkpoints
  - AMR apps are non-sequential concurrent writers
    - – Lousy BW
  - Store file as log of writes
    - – Good BW

N to 1 strided

Each process writes each element in a single shared "stride" within a single shared file. The file consists of one region per element (not one region per *process* as in N-1 non strided). Each region contains "strided" data from each process.

N-to-1 strided example





(c) Write Bandwidth of modified and unmodified PVFS with various numbers of clients and block sizes

- Group 8 mpi-io write test (from LANL)
  - S. Dayal, M. Chainani, D.K. Uppugandi, W. Tantosiriroj

pdsi

pdsi

Storage suffers failures too

		Type of drive	Count	Duration
Pittsburgh Supercomputing Center	HPC1	18GB 10K RPM SCSI 36GB 10K RPM SCSI	3,400	5 yrs
Los Alamos National Laboratory Est. 1943	HPC2	36GB 10K RPM SCSI	520	2.5 yrs
 Supercomputing X	HPC3	15K RPM SCSI 15K RPM SCSI 7.2K RPM SATA	14,208	1 yr
 Various HPCs	HPC4	250GB SATA 500GB SATA 400GB SATA	13,634	3 yrs
 Internet services Y	COM1	10K RPM SCSI	26,734	1 month
	COM2	15K RPM SCSI	39,039	1.5 yrs
	COM3	10K RPM FC-AL 10K RPM FC-AL 10K RPM FC-AL 10K RPM FC-AL	3,700	1 yr

Storage failure especially painful

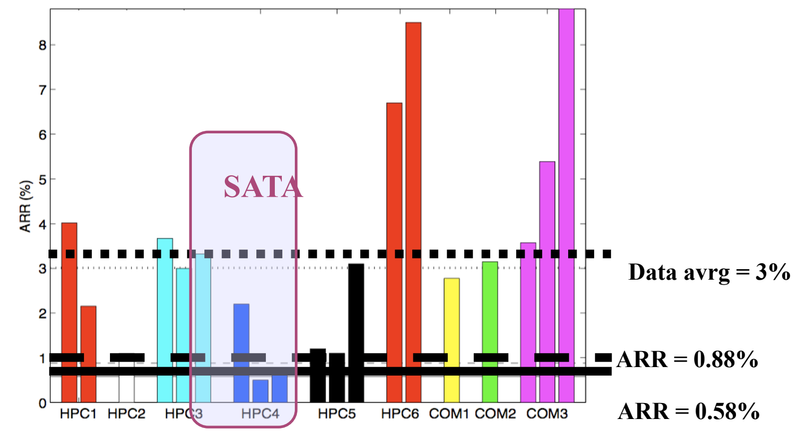
- Scalable performance = more disks
- But disks are getting bigger
- Recovery per failure increasing
- Hours to days on disk arrays
- Consider # concurrent disk recoveries

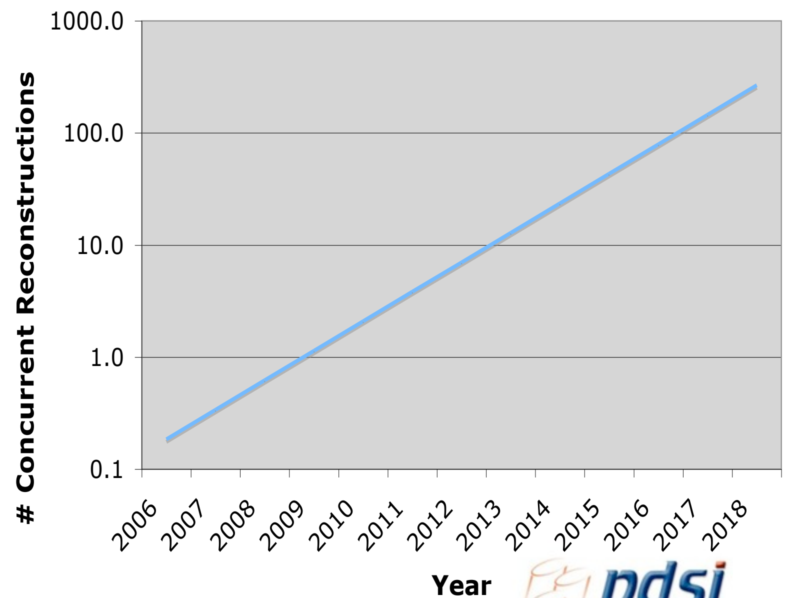
  e.g. 10,000 disks

  3%/year replacement rate

  1+ day recovery each

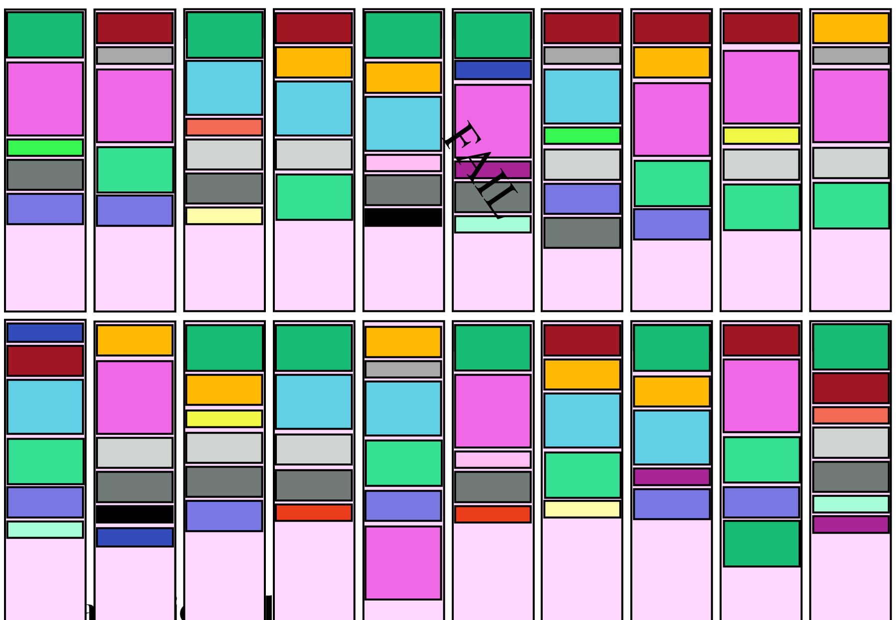
  Constant state of recovery?
- Maybe soon 100s of concurrent recoveries (at all times?)
- Design normal case for many failures (huge change!)

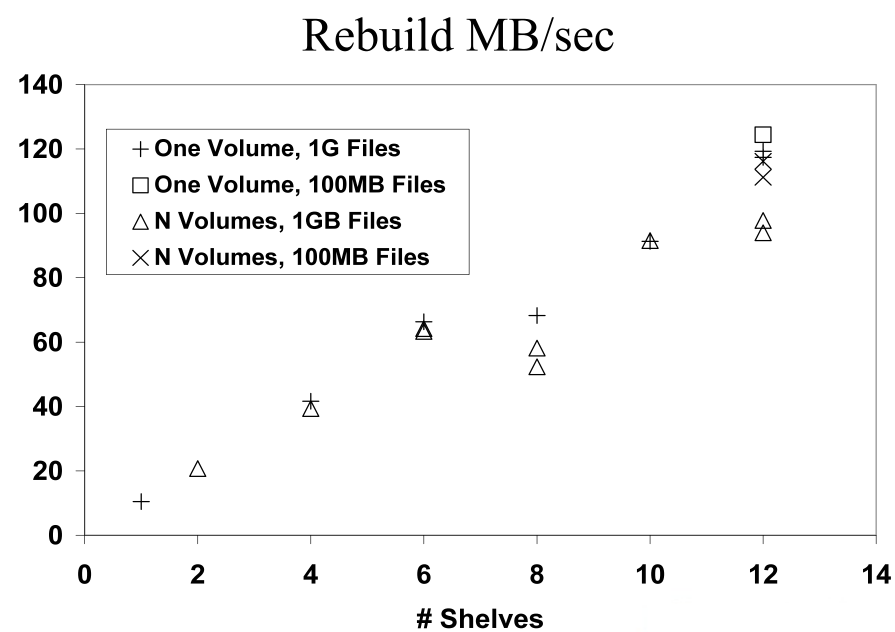




Object storage & scalable repair

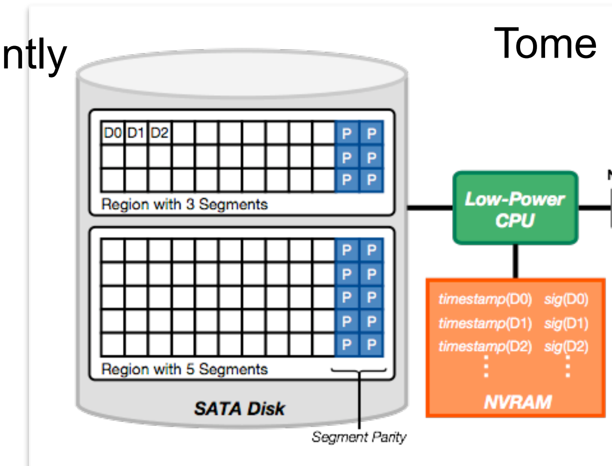
- Defer the problem with parallel scalable repair
- File replication and, more recently, object RAID can scale repair
  - "decluster" redundancy groups over all disks (mirror or RAID)
  - use all disks for every repair, faster is less vulnerable
- Object (chunk of a file) storage architecture dominating at scale
  GFS, HDFS, …. PanFS, Lustre, PVFS, … Centera, …

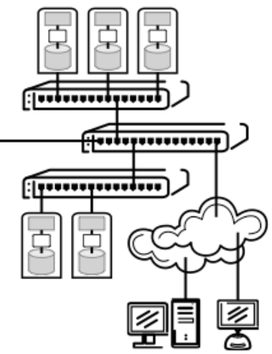


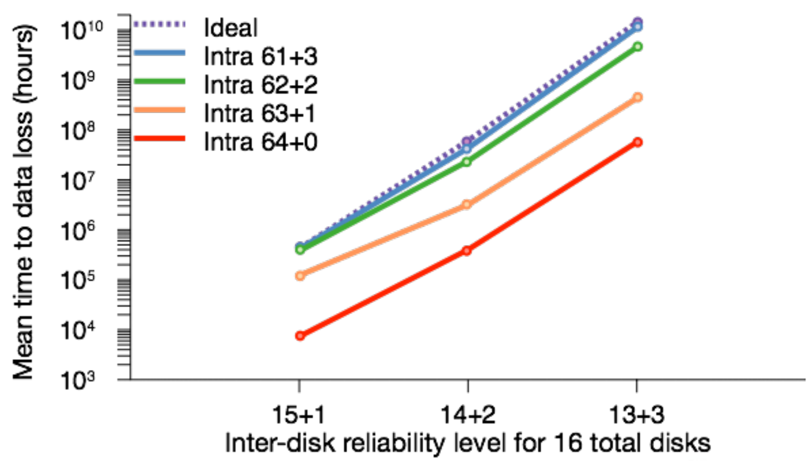


Developing reliable, evolvable Archives

- Evolvable, distributed network of intelligent, disk-based *tomes*
  - Smart enough to function independently
  - Provide inter-disk redundancy
  - Building blocks for more complex systems
  - Evolve over time: integrate new technologies
- Handle errors at multiple levels
  - Scale response to size of problem
  - Very high reliability!
- Control costs
  - Commodity low-power hardware
    - Keep disks spun down
  - Standardized interfaces





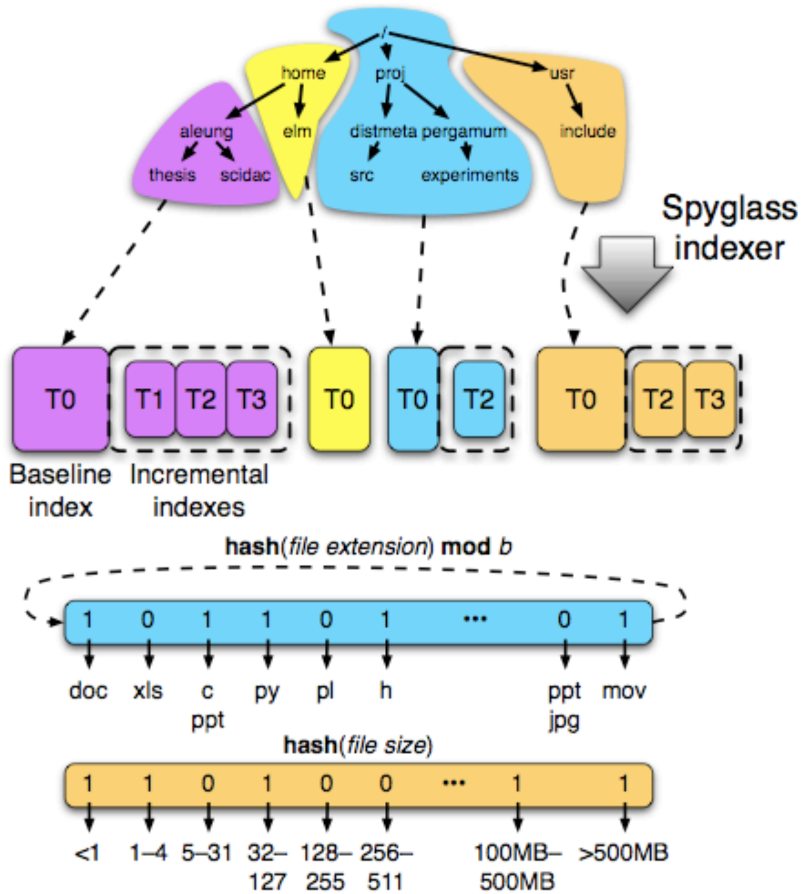


pdsi

Metadata: Subindexing by Name

Spyglass design

- Partition file system hierarchy by subtree
  - Each subtree is an independent subindex
- Summarize contents of each subindex
  - Quickly rule out entire subindexes that can't satisfy the query
- Log incremental changes
  - Rebuild index when there are "enough" changes
- Integrity is much easier
  - Rebuild subindex, not entire index



Use cases for huge directories

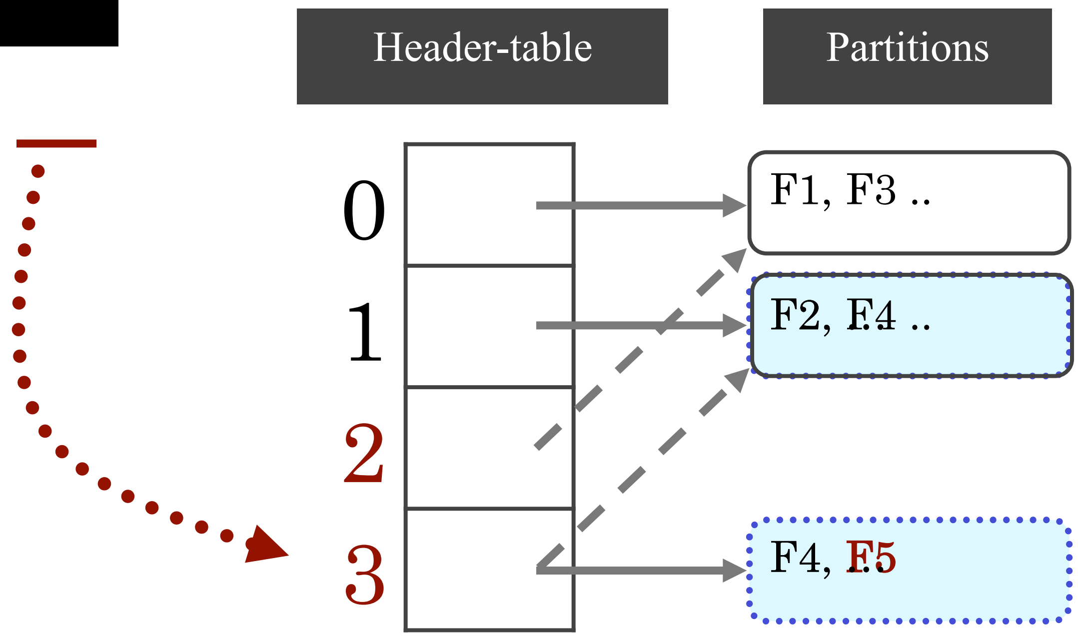
- Apps use FS as fast, lightweight database
  - Use case: All clients inserting millions of small files in a single directory as fast as possible
  - Retain VFS API: create(), lookup(), readdir(), etc.
- Creating many small files in a "burst"
  - E.g., per-process checkpoint on large clusters
  - E.g., science experimental capture
- Creating many small files "steadily"
  - E.g., "log" files from long-running apps for later post-processing (history, bio device runs,…)
- Most interested in pushing the boundaries

Extendible Hashing [Fagin79]

Hash keys for load-balancing

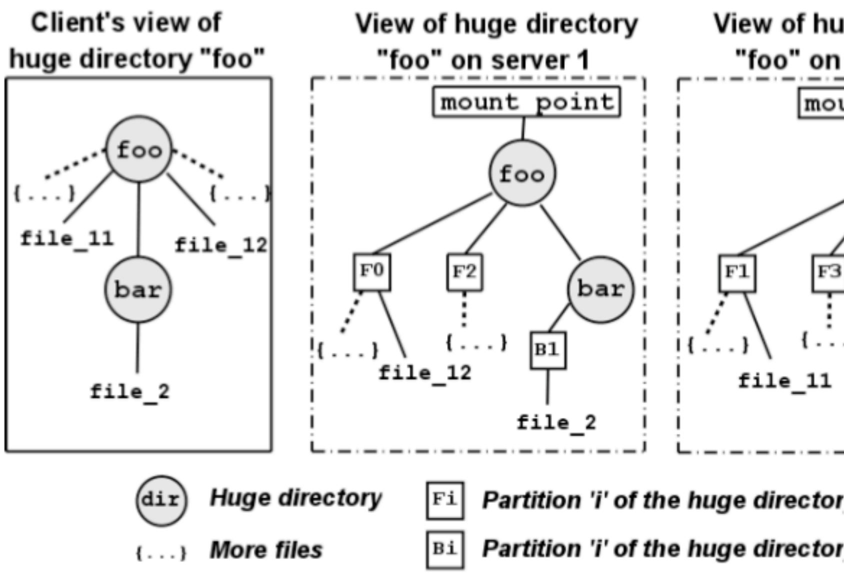
hash(“F5”) = 1001…011

RADIX increases, that uses the growing table (R = 2 bits)



- Header-table doubles, if necessary
  - On splitting, the new partitions distribute their keys
- Mechanism designed for single server impln.

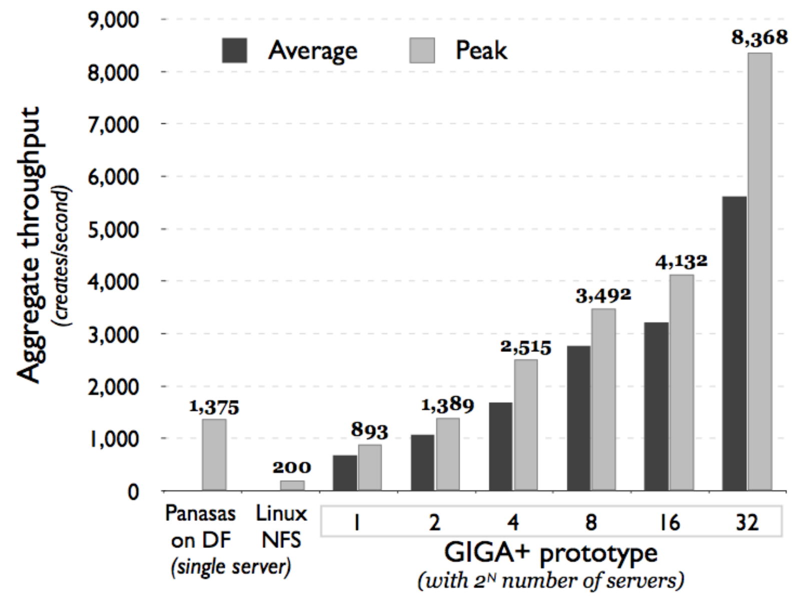
GIGA+ Directories (PVFS, FUSE)



Local representation of huge directory in Giga

- Eliminate serialization
  - All servers grow directory independently, in parallel, without any co-ordinator

- No synchronization & consistency bottlenecks
  - Servers only keep local "view", no shared state

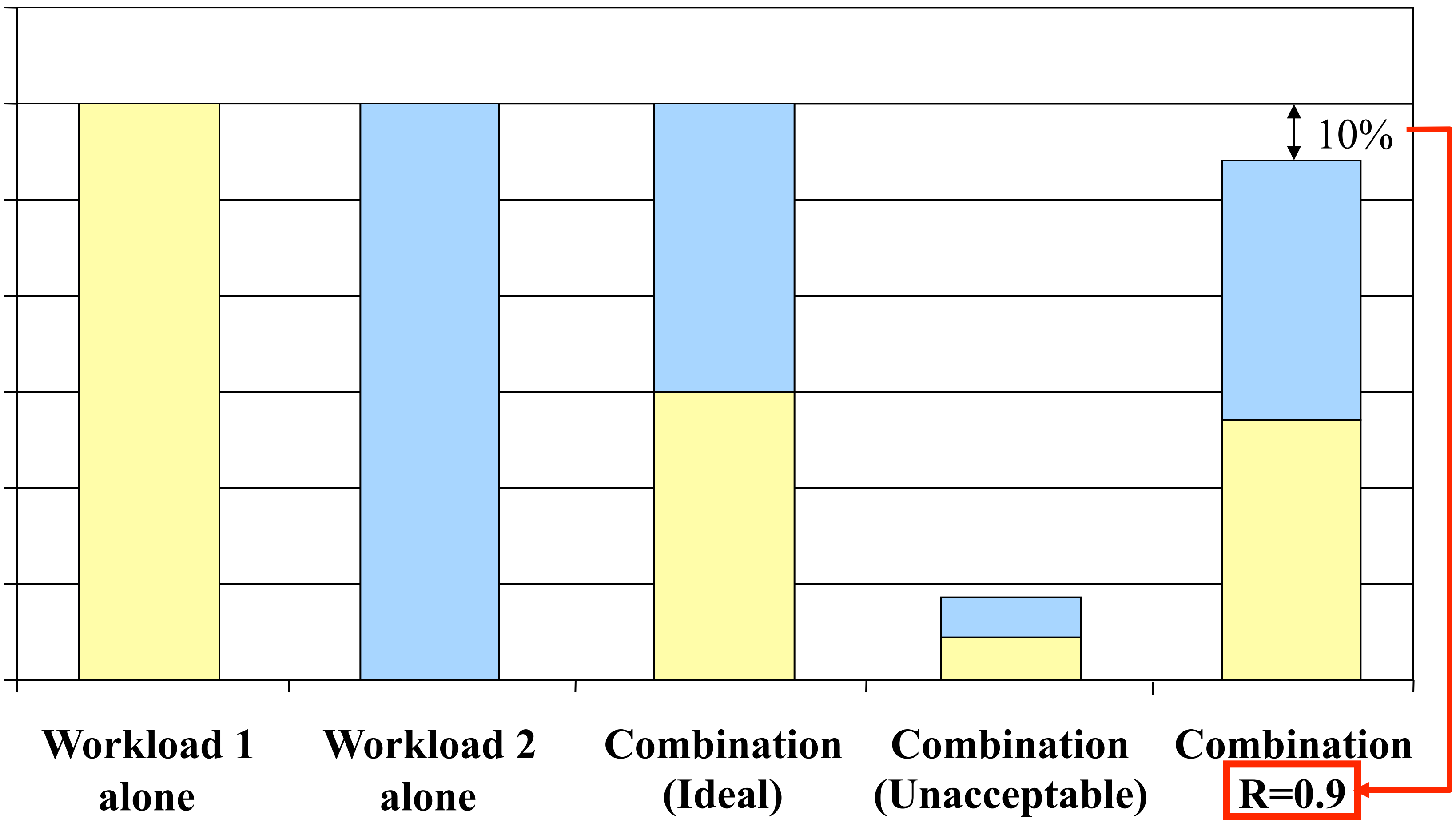


Scale and performance of Giga+ using UCAR Metarates benchmark.

Whither shared storage clusters ?

- Contrasted to per-application/per-machine
  - sharing allows common namespace
  - sharing allows common provision+use of spare
    - including bursty usage
- But, interference can kill storage performance
  - Disk: “context switch” = mechanical seek (slow!)
  - Cache: what does time-sharing mean?
  - Cluster: coordinating timing across nodes

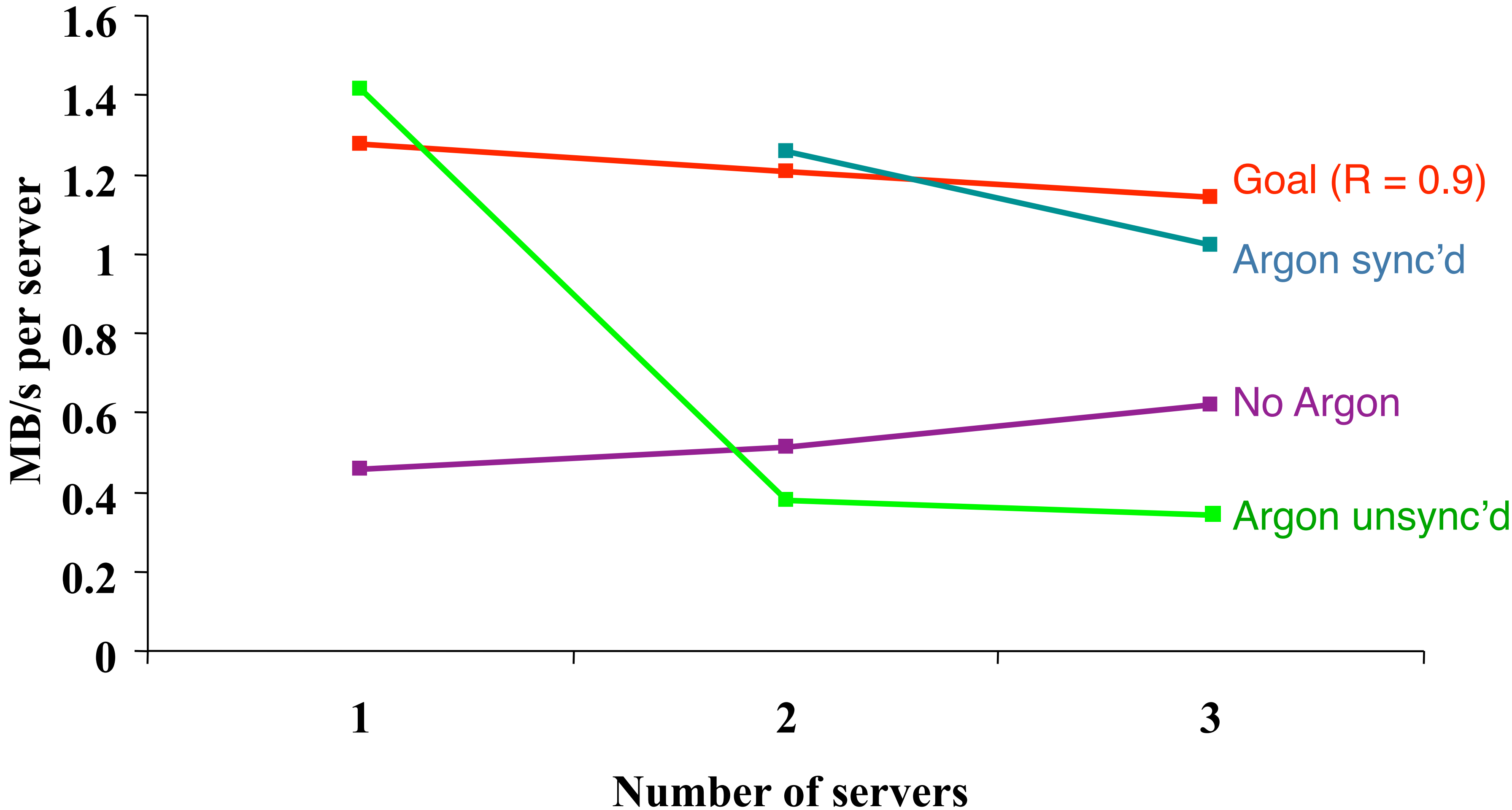
R-value quantizing of disk arm (2 apps)



But, data will be striped over servers

- Data striped for performance (esp. bandwidth)
  - each client req. translates to multiple server accesses
  - client req. is “done” when all accesses are done
    - so, overall req. waits for the slowest one
  - unsynchronized quanta can lead to significant delays
    - so, need to coordinate quanta (a la synch spindles)

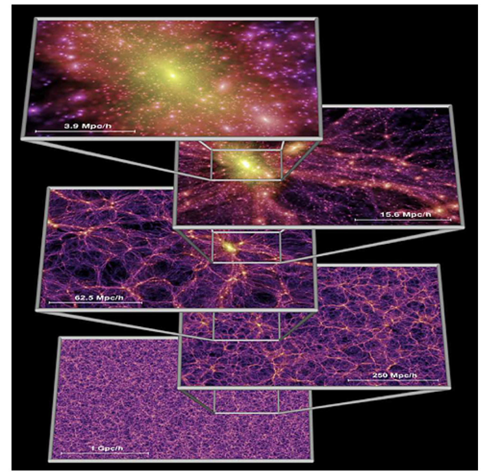
Promising initial results (seek intensive)



Cosmology Simulations (A. Szalay, JHU)

Cosmological simulations have 10^9 particles and produce over 30TB of data (Millennium, Aquarius, …)

- Build up dark matter halos
- Track merging history of halos
- Use it to assign star formation history
- Combination with spectral synthesis
- Realistic distribution of galaxy types



- Too few realizations (IO and storage limited)
- Hard to analyze the data afterwards ->need DB (Lemson)
- What is the best way to compare to real data?

Looking to another form of object storage

- HPC & Web search operate at similar scale
  - 10s of thousands of nodes & growing
  - Want to co-opt web effort/excitement for HPC

The Google File System

By Sanjay Ghemawat, Howard Gobioff,
Shun-Tak Leung
(Presented at SOSP 2003)

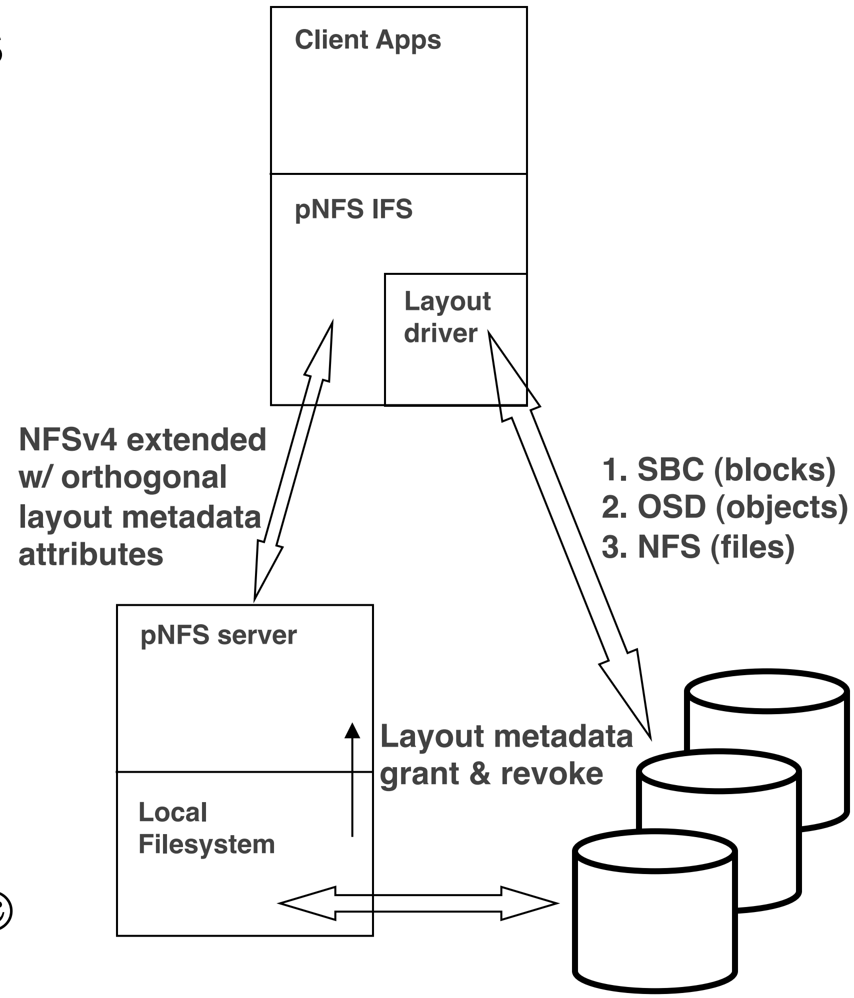
Motivational Facts

- More than 15,000 commodity-class PC's.
- Multiple clusters distributed worldwide.
- Thousands of queries served per second.
- One query reads 100's of MB of data.
- One query consumes 10's of billions of CPU cycles.
- Google stores dozens of copies of the entire Web!

Conclusion: Need large, distributed, highly fault-tolerant file system.

Recall pNFS: scalable NFS very soon

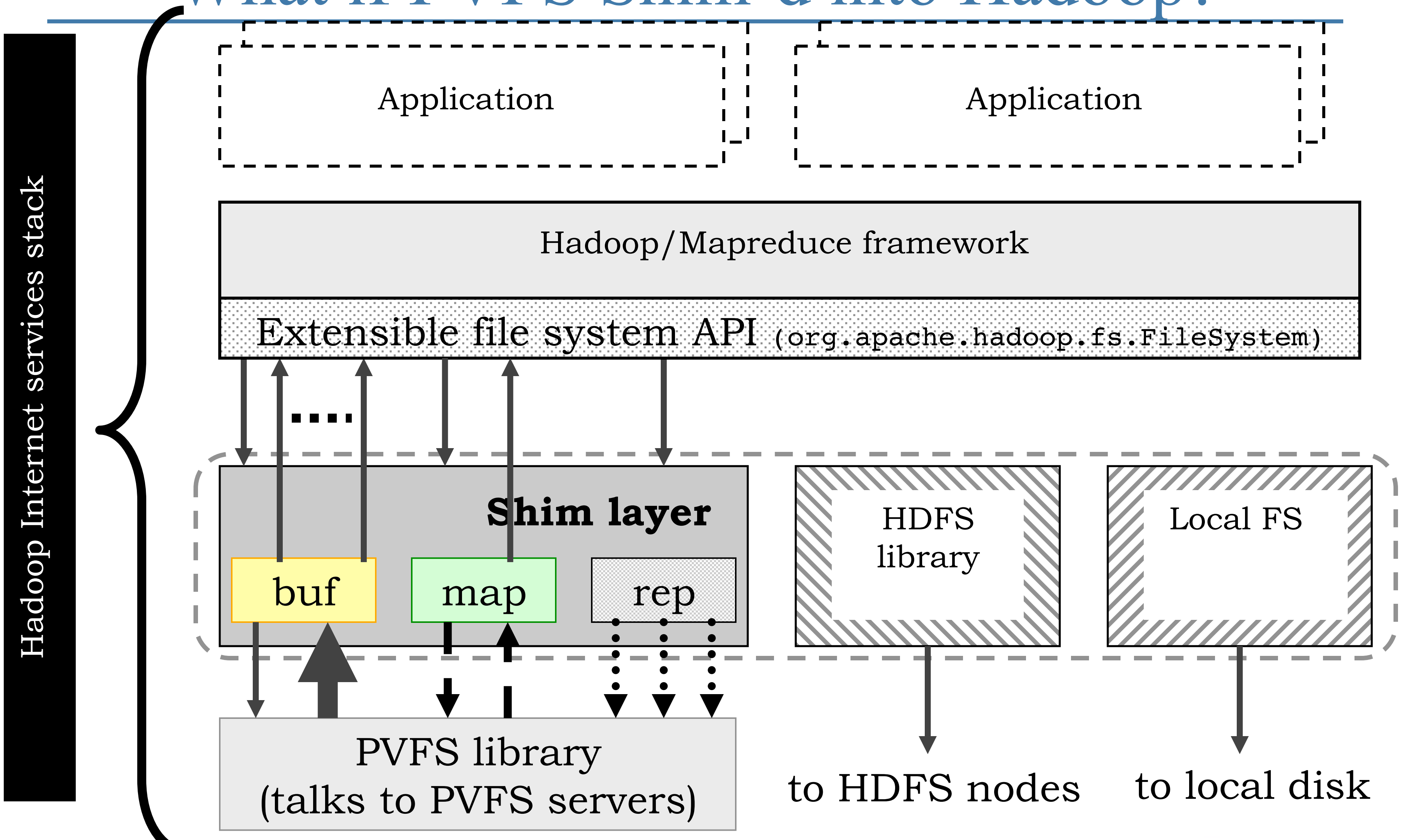
- Teach NFS to delegate file maps
  - Client directs parallel transfer
  - Scales bandwidth up
  - Scales metadata load down
- IETF standard near complete
  - Sun, NetApp, EMC,
    IBM, Panasas, BlueArc, etc
  - Open source Linux essential
    - Linux core team active
- Data servers can be clients too
  - Maps expose placement
  - Full, friendly, familiar, file systems ☺



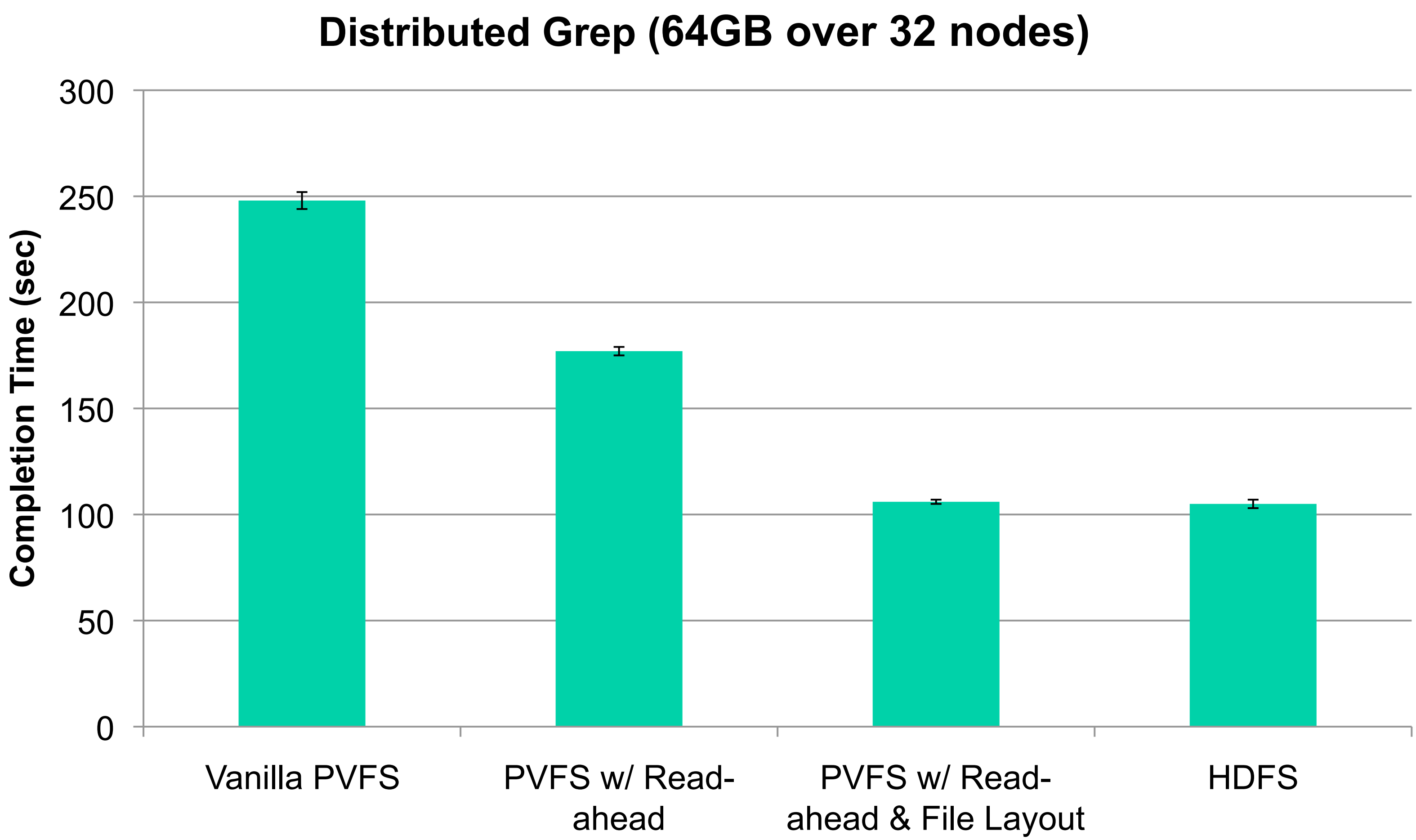
Carnegie Mellon
Parallel Data Laboratory

pdsi

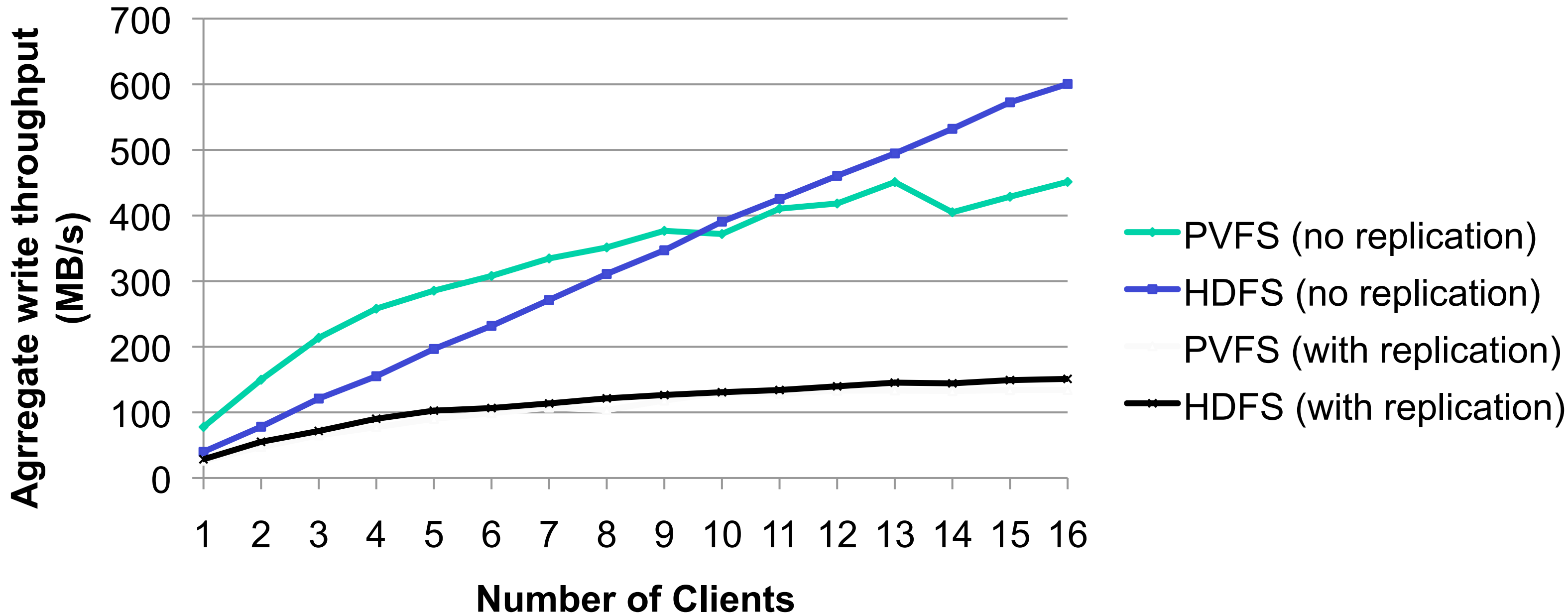
What if PVFS Shim'd into Hadoop?



Modified PVFS



N clients writes to n distinct files



- Multiple copies requires HDFS and PVFS to perform more write operation
  - HPC file systems need to go this route for scalable rebuild
- HDFS writes the first copy locally (bad distribution)
  - A good trick only if real work is already subdivided

Concurrent writes to a single file

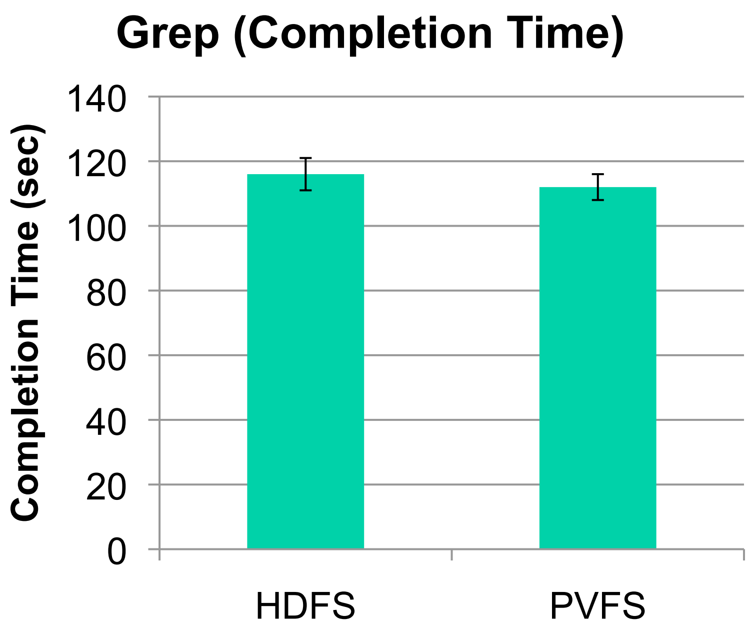
	HDFS	PVFS
Throughput (MB/s)	24.6	105.5
Network Traffic In (GB)	49.7	59.0
Network Traffic Out (GB)	48.1	59.2
Completion Time (min:sec)	10:50	2:31

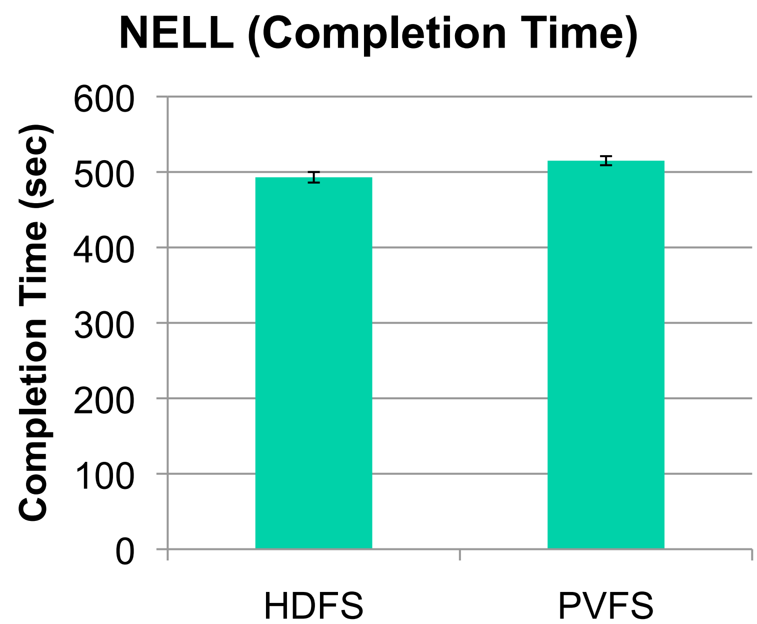
- PVFS enables concurrent writes to non-overlapping regions, so N clients, each can copy 1/N each.
- Without multiple writers to a file, HDFS can only go as fast as a single client can
- Real issue is Internet service users have to play with data to store in lots of right-sized sub-files (ugh)

Test with Analytics benchmarks

- **Grep**: Search for a rare pattern in two million 100-bytes records
- **Sort**: Sort two million 100-bytes records
- **Never-Ending Language Learning (NELL)**: (from J. Betteridge) Count the numbers of selected phases in 37GB data-set
- **Page-Rank Application**: (from L. Zhao) Rank webpage by their reading difficulty level (aka. easy to read)

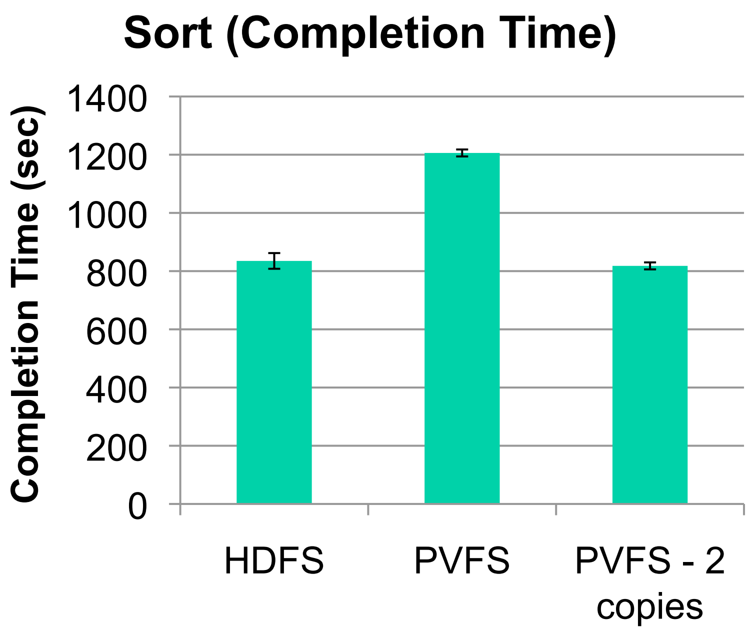
Read-Intensive Benchmark

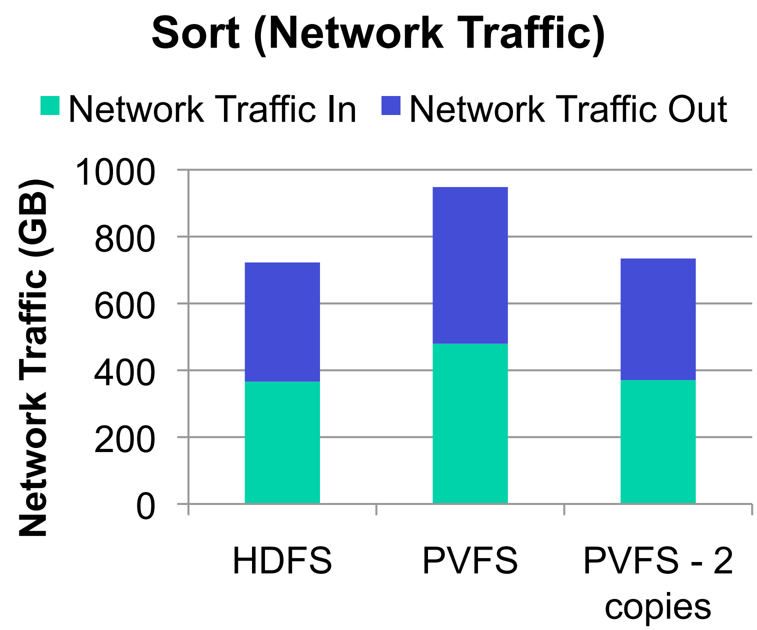




- PVFS's performance is similar to HDFS in read-intensive applications

Write-Intensive Benchmark





- In write-intensive application, HDFS performs better because it writes the first copy locally.

Its All About Data, Scale & Failure (not cycles)

- Continual gathering of data on data storage
  - Failures, distributions, traces, workloads
- Nurturing of file systems to HPC scale, requirements
  - pNFS standards, benchmarks, testing clusters, academic codes
- Checkpoint specializations
  - App compressed state, special devices, special representations
- Failure as the normal case?
  - Risking 100s of concurrent disk rebuilds (need faster rebuild)
  - Quality of service (performance) during rebuild in design
- HPC vs Cloud Storage Architecture
  - Where is the storage? What traffic patterns? Common code?
- Correctness at increasing scale?
  - Testing using virtual machines to simulate larger machines
  - Formal verification of correctness (performance?) at scale

SciDAC
Scientific Discovery through Advanced Computing

Office of Science

SciDAC PDSI Update (part 2)

CS/VIS PI Meeting, October 23, Germantown, MD

Garth Gibson

Carnegie Mellon University and Panasas Inc.

SciDAC Petascale Data Storage Institute (PDSI)

www.pdsi-scidac.org

w/ LANL (G. Grider), LBNL (W. Kramer), SNL (L. Ward),
ORNL (P. Roth), PNNL (E. Felix),
UCSC (D. Long), U.Mich (P. Honeyman)