SDC
STORAGE DEVELOPER CONFERENCE
SNIA SANTA CLARA, 2016

Data Integrity Support for Silent Data Corruption in Gfarm File System

Osamu Tatebe
University of Tsukuba

Silent Data Corruption

- Data may be corrupted silently
  - Transient soft error by cosmic ray, ...
  - RAID firmware bug, storage software bug
  - RH6.2 & 6.3 – XFS regularly truncating files after crash/reboot [RH Bug 845233]

Data-Intensive Science

- Big Data Science
  - High energy physics experiment - LHC, Belle (PB/year)
  - Wide-field imaging - Subaru HSC survey, LSST, SDSS (100TB/year)
  - Next generation DNA sequencer
  - Data assimilation in climate science

Not only FLOPS but Byte/sec (IO bandwidth) is critical

Scalable performance requirement for Parallel File System

Year	FLOPS	#cores	IO BW	IOPS	Systems
2008	1P	100K	100GB/s	O(1K)	Jaguar, BG/P
2011	10P	1M	1TB/s	O(10K)	K, BG/Q
2017	100P	10M	10TB/s	O(100K)	
2022	1E	100M	100TB/s	O(1M)	**Performance target**

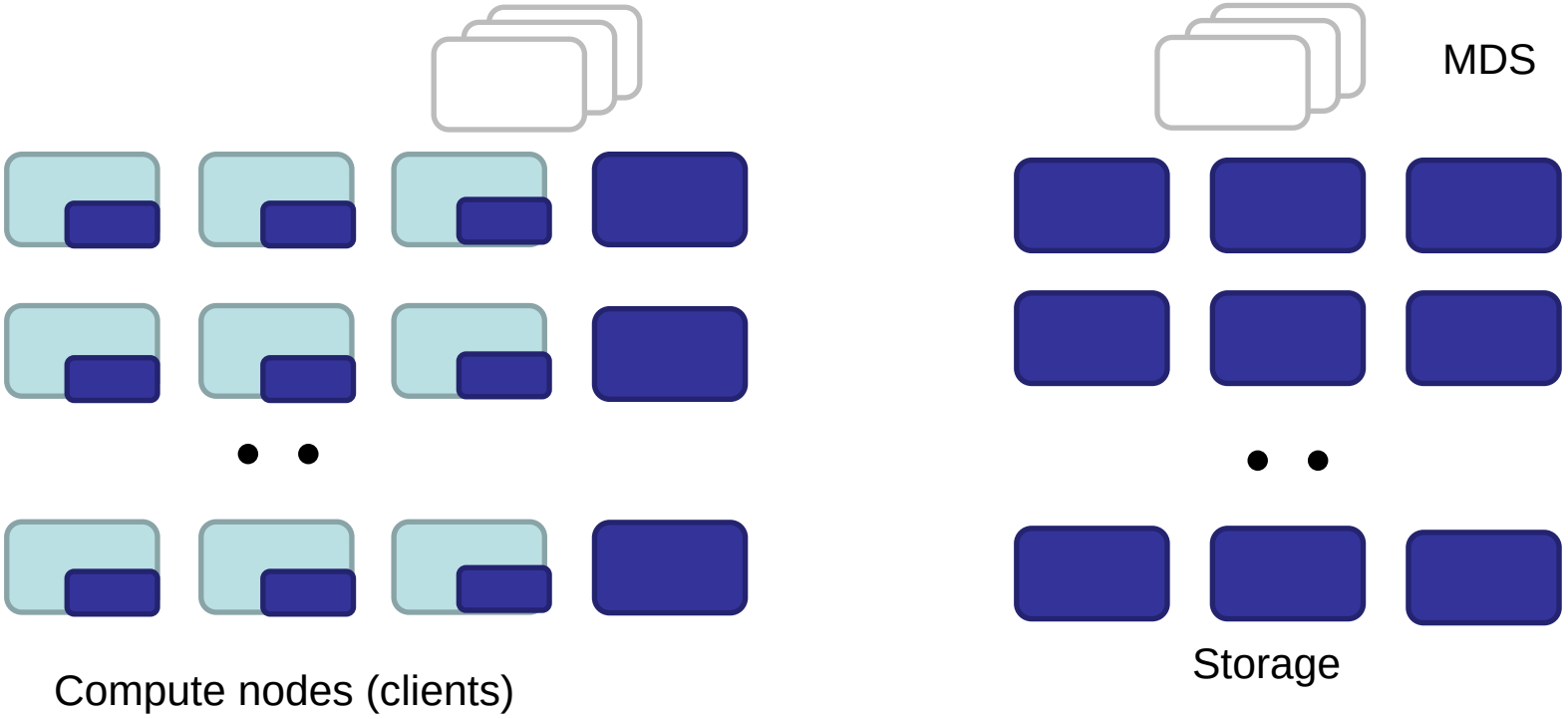
IO BW and IOPS are expected to be scaled-out in terms of # cores or # nodes

Convergence of HPC and Big Data Computing

- Scale-out system R&D such as MapReduce in Data Center
  - Exploit local storage due to poor network bandwidth
  - Tolerate fault since it is norm
- OTOH, MPI and MPI-IO system R&D in HPC using high-performance and high bisection network
- Complex data analysis requirement in data-intensive science imposes the convergence

Extreme Big Data

Converged Architecture



R&D for scale-out system software required

Storage System of Converged Architecture

- Three-tier and more
  - O(10,000) Local storage
    - More than staging
  - O(1,000) File cache system
    - More than burst buffer
  - O(100) Parallel file system
- Data locality should be considered for local storage

SNIA-J Extreme Storage Society of Science Study

- Established in May, 2016
- Discuss about next-generation high-performance storage (extreme storage) beyond HPC, Big Data, and Cloud technologies
- Monthly meeting
- Chair: Osamu Tatebe (University of Tsukuba)
- Members: Fujitsu, Hitachi, NEC, Toshiba, HGST Japan, TÜV Rheinland Japan, CTCSP, SCSK, DDN, KEL, TIS Solution Link, ...

Gfarm file system

Most Innovative Use of Storage
In Support of Science Award in SC05

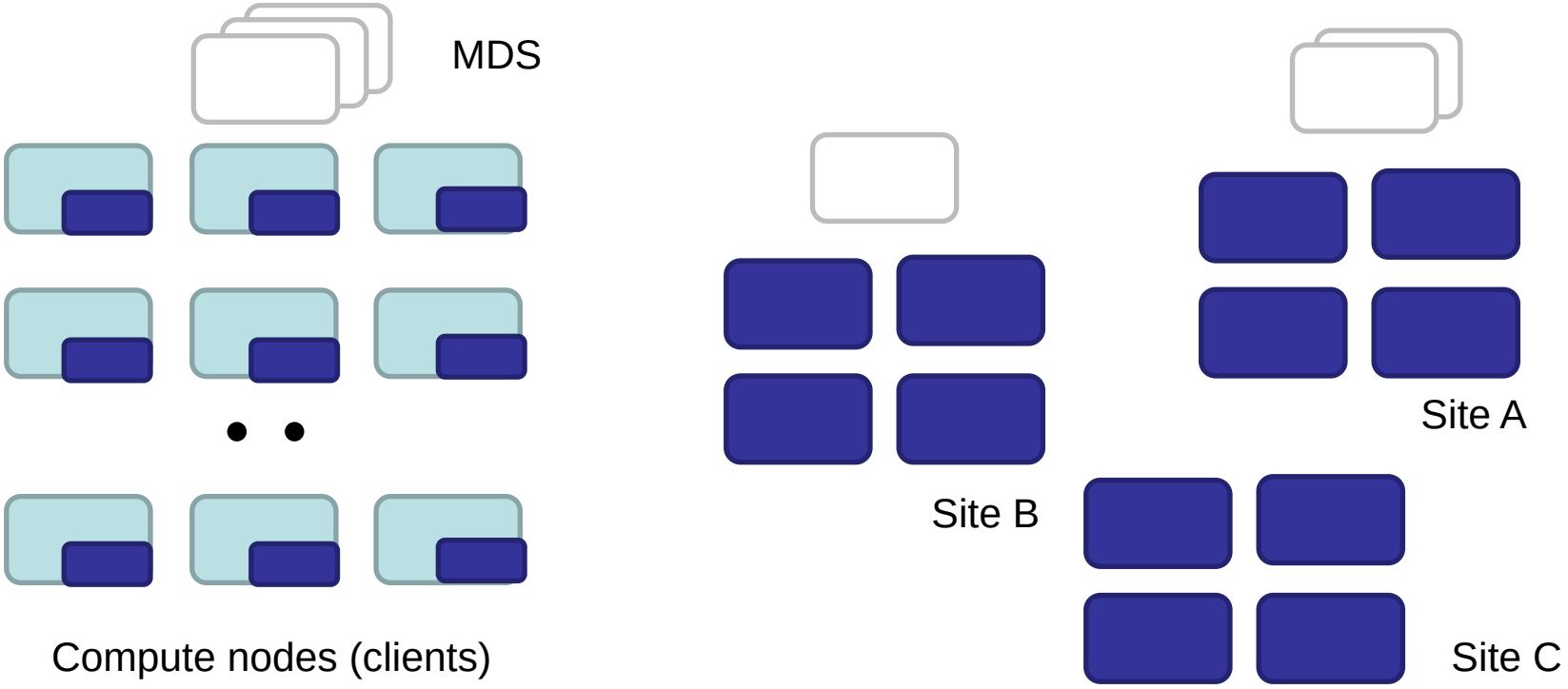
Winner – Large Systems
in HPC Storage Challenge in SC06

- Open Source Software
  - http://sf.net/projects/gfarm
  - 18,000 downloads since March 2007
- Major installations
  - JLDG (7.8 PB, 7 sites, 39 file servers)
  - HPCI Storage (21.9 PB, 3 sites, 97 file servers)
- Research for much more scalability and Non-volatile Storage

Gfarm file system

- Runtime system to exploit data locality
  - Pwrake workflow system
  - MapReduce, MPI-IO, batch queuing system
- NPO Tsukuba OSS Technology Support Center
  - http://oss-tsukuba.org/
  - Support for Gfarm file system
  - Gfarm symposium, Gfarm workshop

Two Extremes



O(1,000) local storages for data-intensive applications

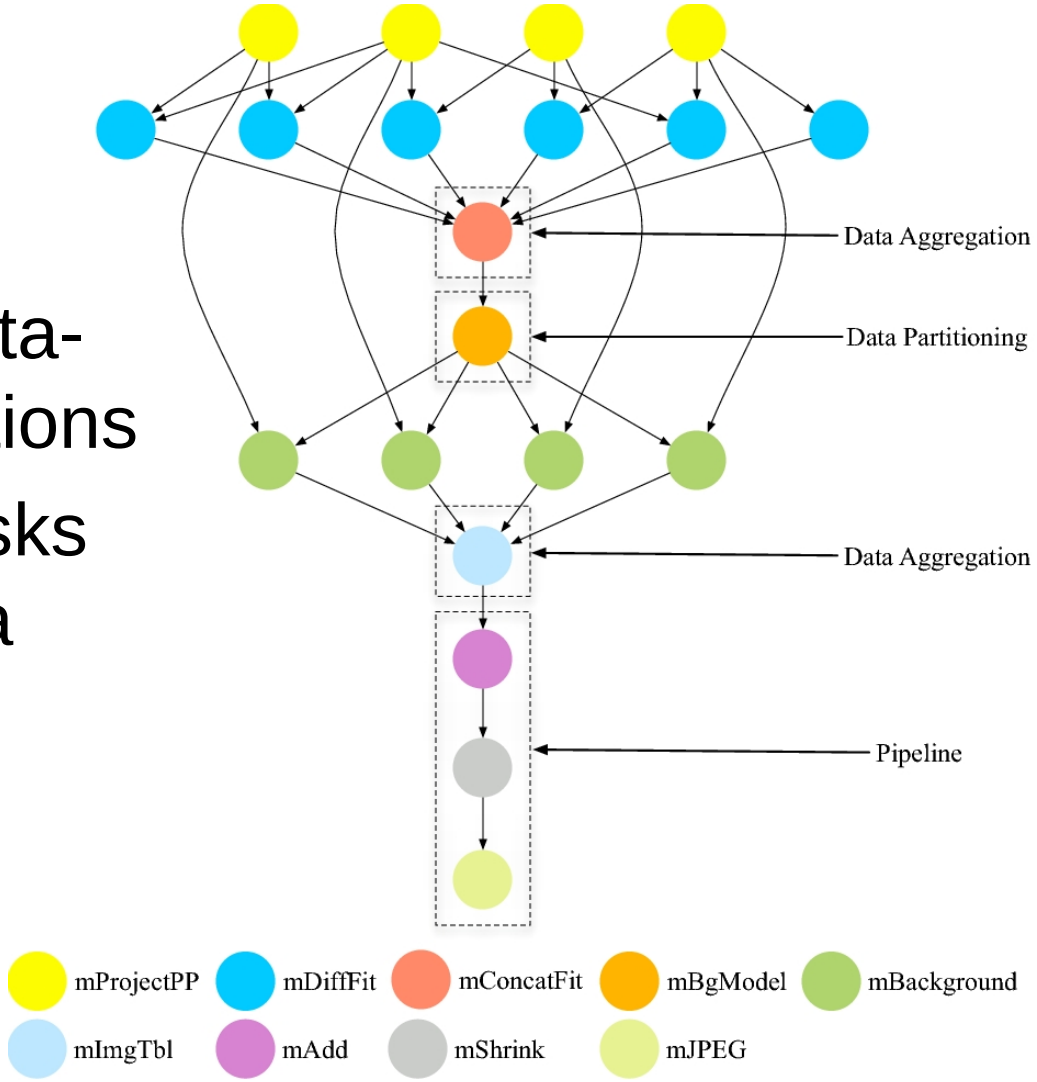
O(1,000) file servers in distant sites for global data sharing and archives

Software Component of Gfarm File System

- Master-slave MDS
  - Synchronous replication for fault tolerance, and asynchronous replication for disaster recovery
- IO server for each local storage
- Client software
  - Gfarm2fs to mount Gfarm file system
  - Gfarm command for replica management and parallel file copy, and POSIX equivalent
  - Gfarm library API in C, C++, Java

Workflow

- Often used by data-intensive applications
- A collection of tasks that includes data dependency



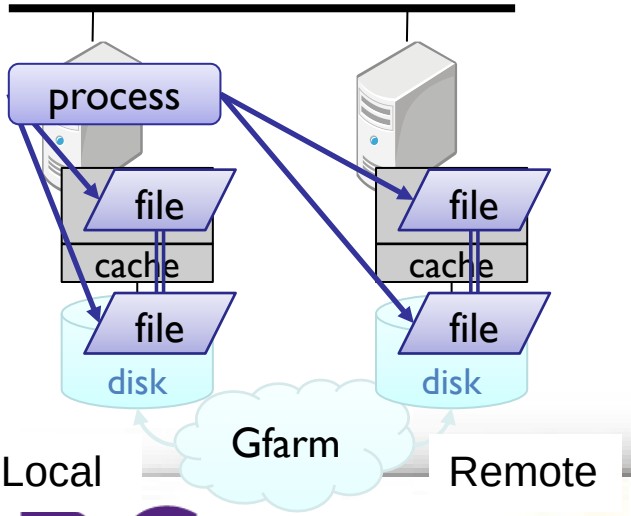
https://confluence.pegasus.isi.edu/display/pegasus/WorkflowGenerator

Pwrake: Data-Intensive Workflow System [Tanaka]

Workflow System based on Rake (Ruby make)
http://github.com/masa16/Pwrake/

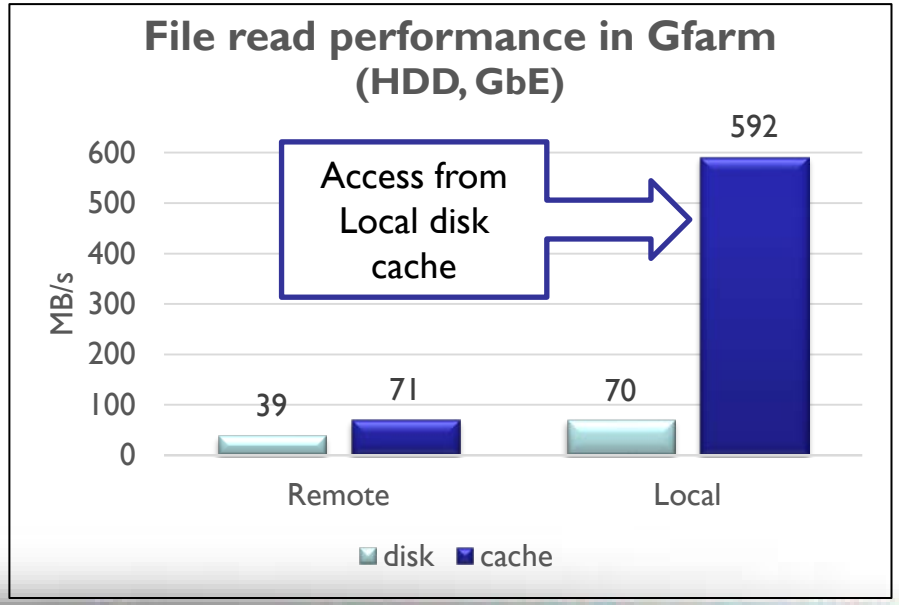
Rakefile as a workflow language

- Dynamic workflow like Montage astronomical data analysis possible



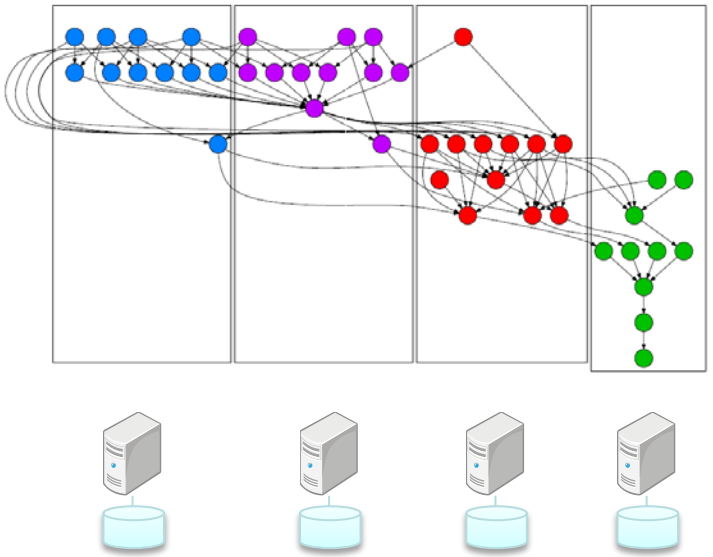
IO-aware task scheduling

- Locality-aware scheduling (CCGrid2012)
  - Minimize data transfer by multi constraint graph partitioning
- Disk Cache aware (Cluster2014)
  - Maximize disk cache hit ratio and avoid trailing task problem



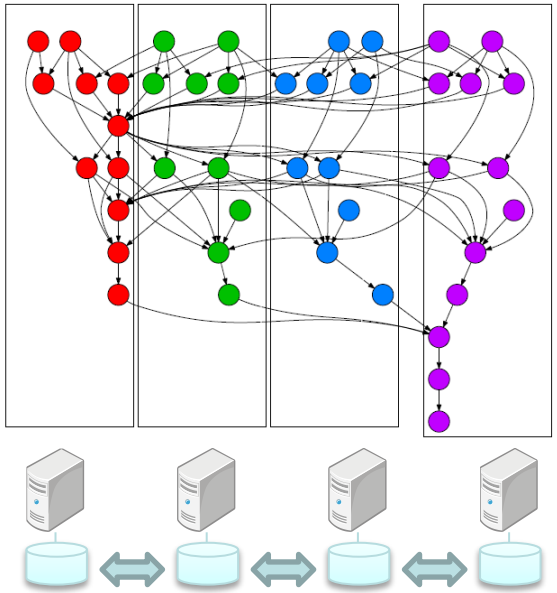
Maximize Locality using Multi-Constraint Graph Partitioning [Tanaka, IEEE CCGrid 2012]

Simple Graph Partitioning



Parallel tasks are unbalanced among nodes

Multi-Constraint Graph Partitioning

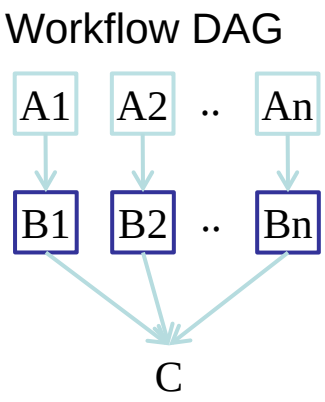


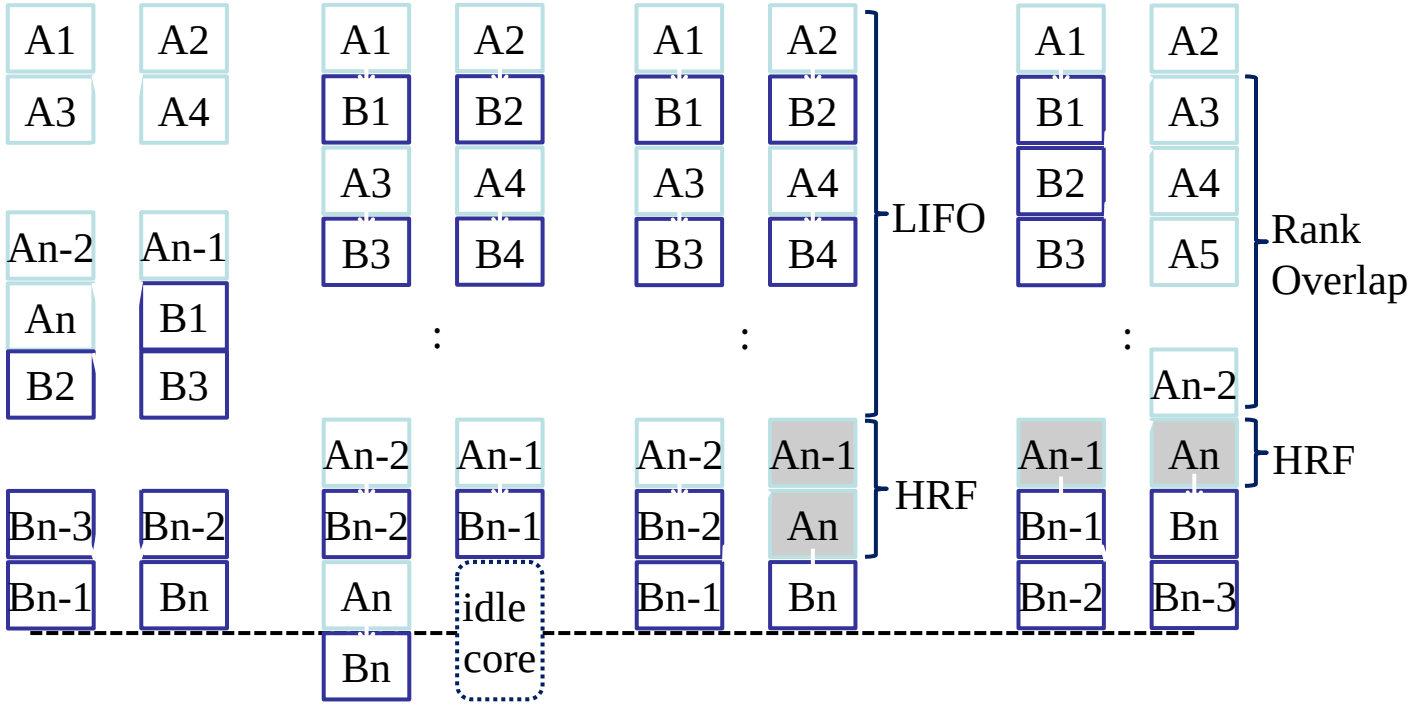
Data movement reduced by 86%
Execution time improved by 31%

Disk Cache Aware Scheduling

[Tanaka IEEE Cluster 2014]

switch LIFO and HRF depending on # tasks



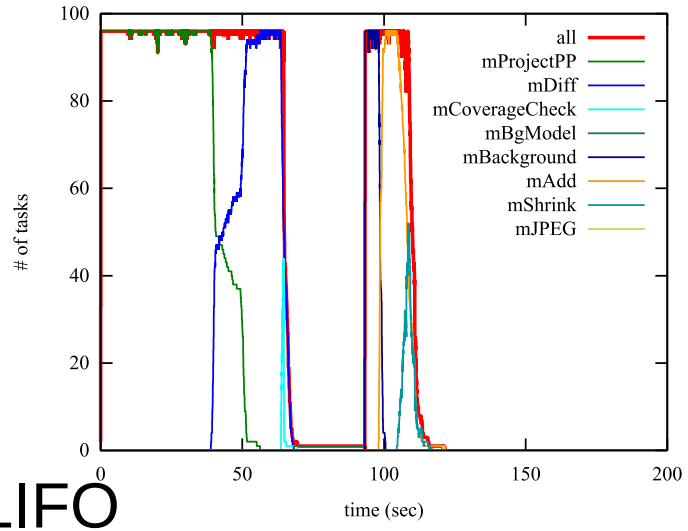


	FIFO (HRF)	LIFO	LIFO+HRF	Rank Equalization+HRF
Disk Cache	×	◎	◎	○
Trailing Task	○	×	○	○
Task Overlap	×	×	×	○

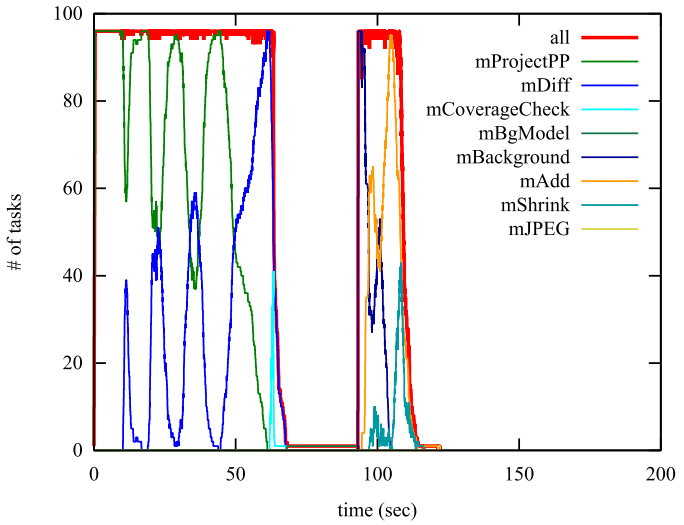
HRF: Highest Rank First

Parallel Execution Tasks over Time

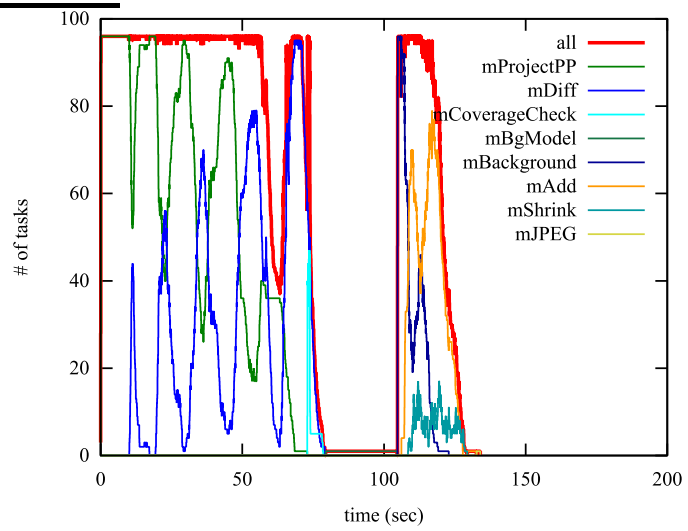
FIFO



LIFO+HRF



LIFO



- LIFO utilizes disk cache but it has a trailing task problem
- LIFO+HRF utilizes disk cache and solves the trailing task problem

HPCI Storage

- HPCI – High Performance Computing Infrastructure
  - RIKEN AICS (“K”), NII, Hokkaido, Tohoku, Tsukuba, Tokyo, Titech, Nagoya, Kyoto, Osaka, Kyushu, JAMSTEC, ISM, AIST
- A 20PB single distributed file system consisting East and West sites
- Single Sign-on by Grid Security Infrastructure (GSI) and user identification by Subject DN
- Parallel file replication among sites
- Parallel file staging to/from each Supercomputer center

Picture courtesy by Hiroshi Harada (U Tokyo)

How to Use HPCI Storage

% **mount.hpci** **# mount**

Update proxy certificate for gfarm2fs

timeleft : 167:50:40 (7.0 days)

Mount GfarmFS on /gfarm/hp120273/tatebe

% **cd /gfarm/hp120273/tatebe**

% **gfpcopy –P /work/CSI/tatebe/data . #parallel copy**

….

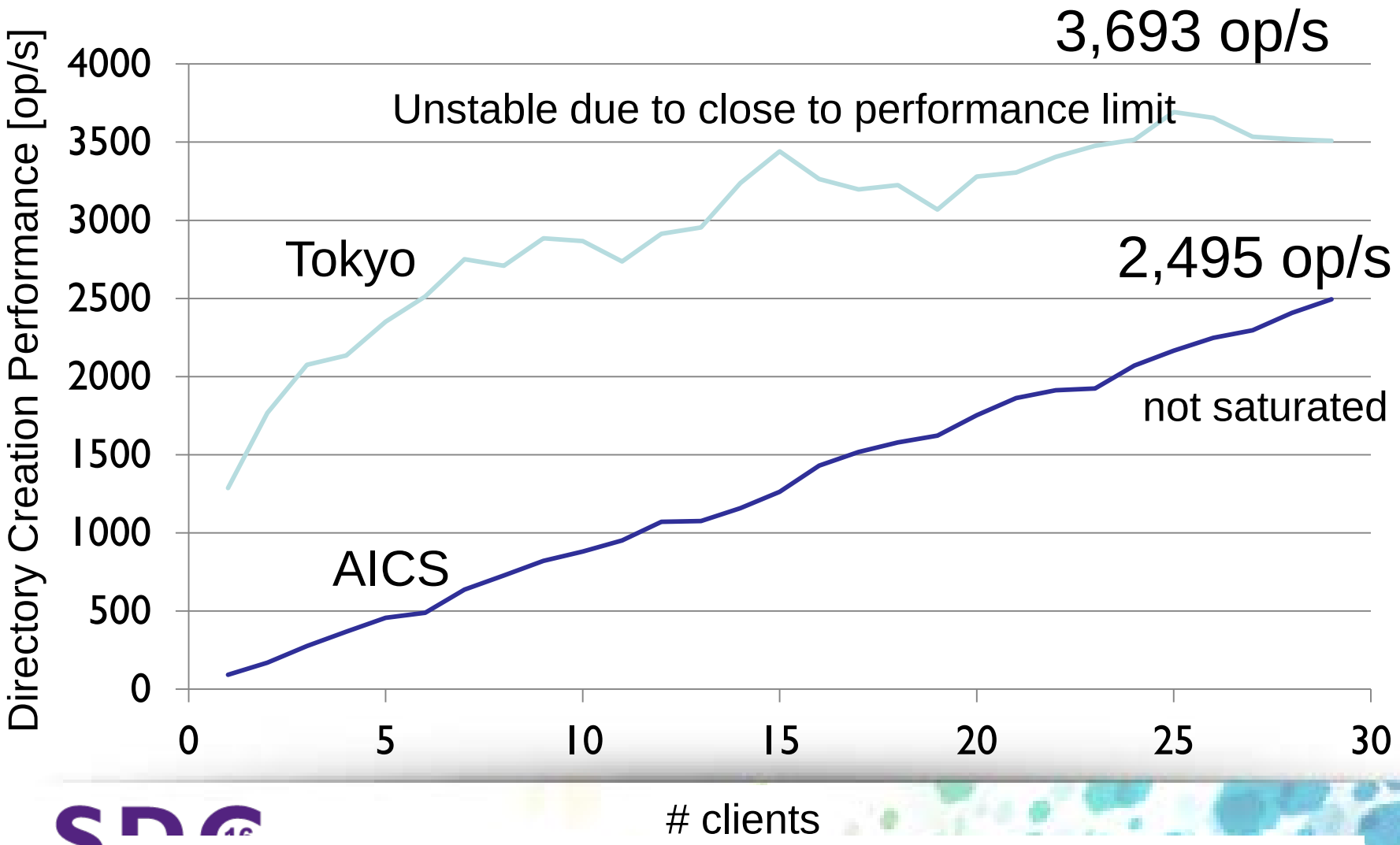
total_throughput: 70.233735 MB/s

total_time: 93.311284 sec.

% **gfncopy –s 2 data** **#specify # of file replicas**

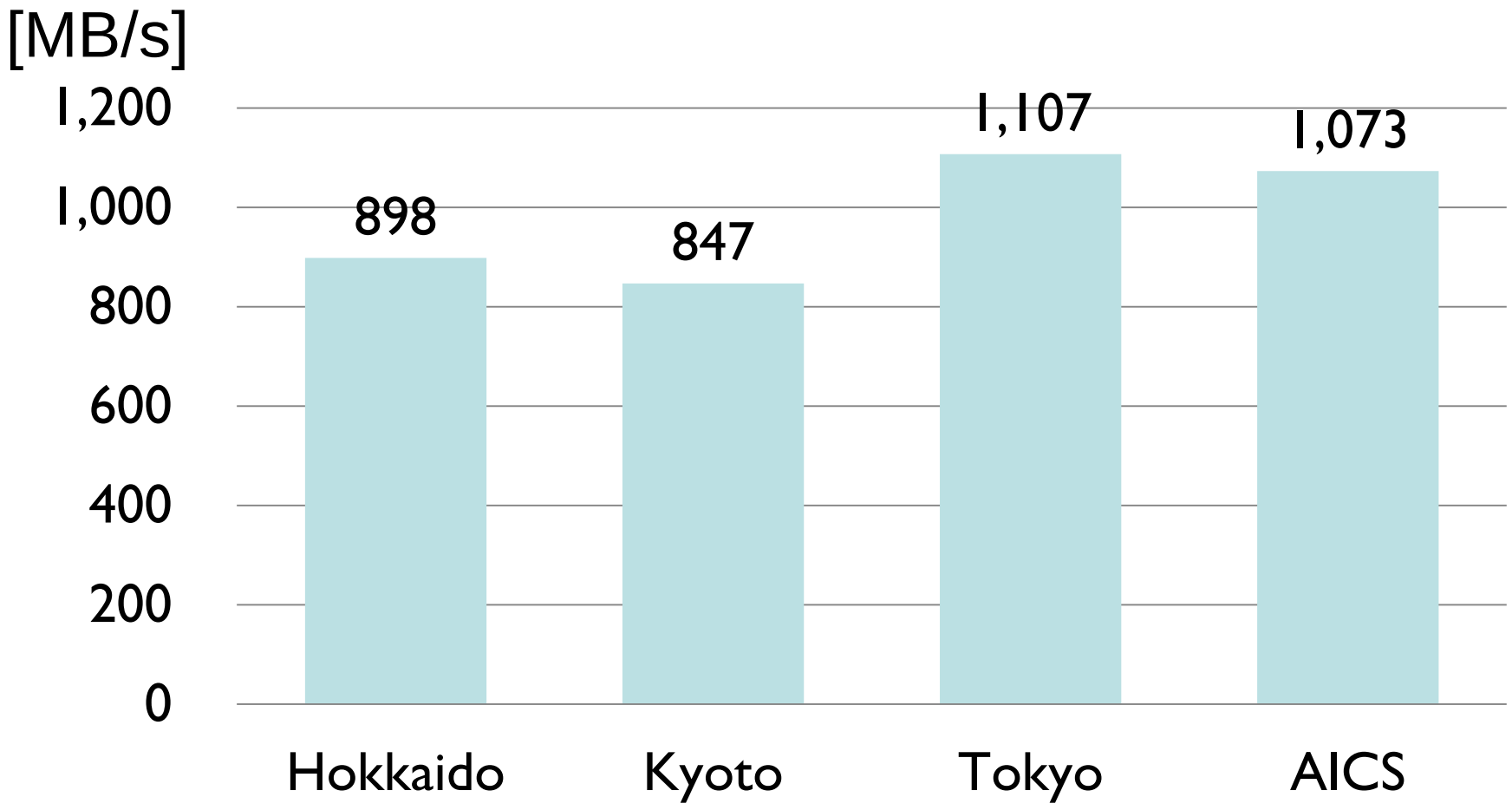
(file replication starts in the background)

IOPS for Directory Creation



10,000 op/s and more achieved without synchronous slave MDS

I/O bandwidth of HPCI Storage



File copy performance of 300 x 1GB files to HPCI Storage

End-to-end Data Integrity

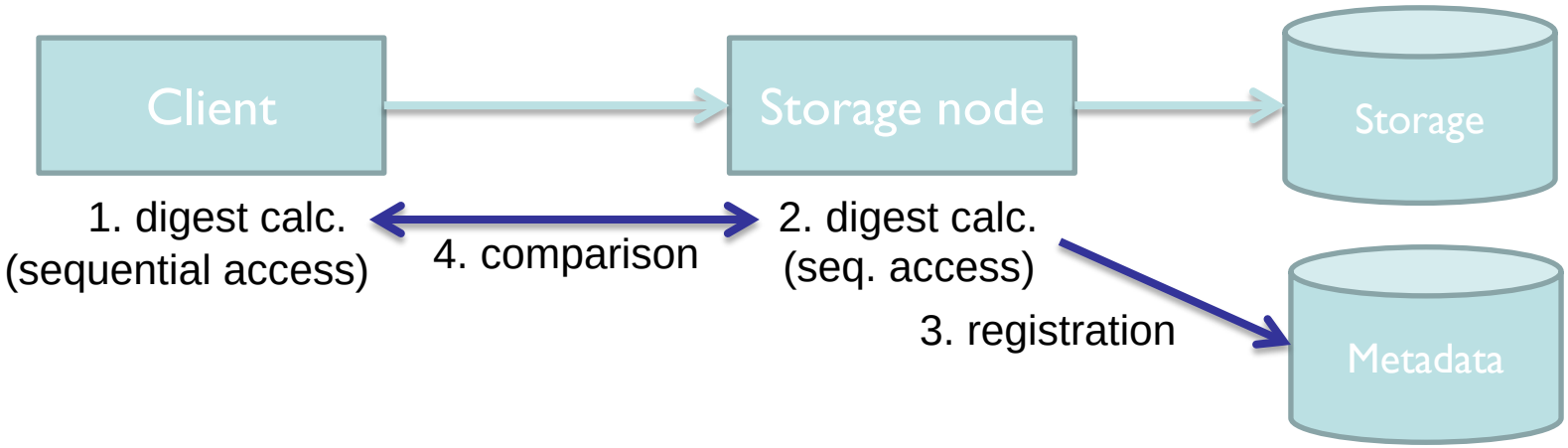
- Silent data corruption in large-scale storage
  - No error happens, but damaged

→ End-to-end data integrity by Gfarm file system

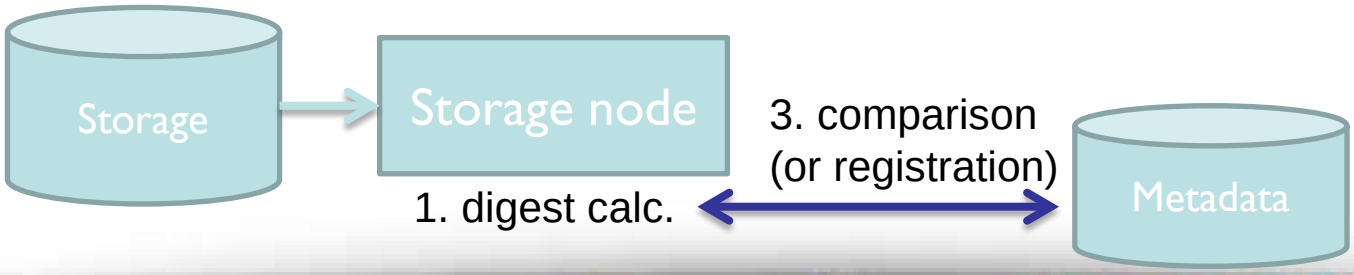
  - Checksumming by client and storage node
  - Checksumming when reading and replicating data
  - Write verify
  - Corrupted files moved to /lost+found for automatic recovery

Data Integrity in Gfarm (1)

- Writing data

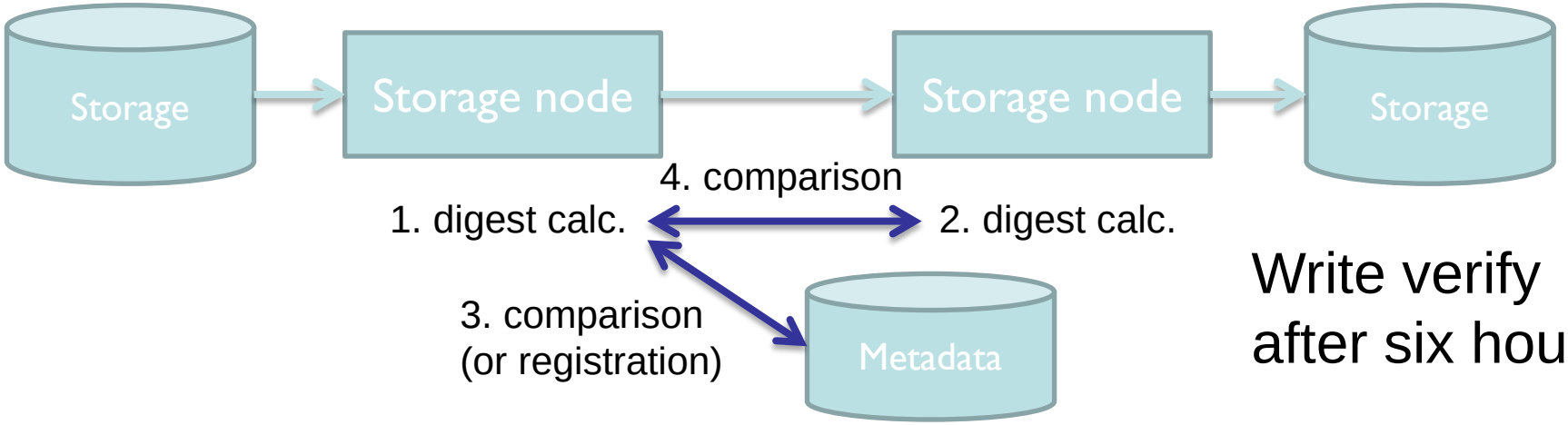


- Write verify after six hours by default



Data Integrity in Gfarm (2)

- Replicating data (just after writing)

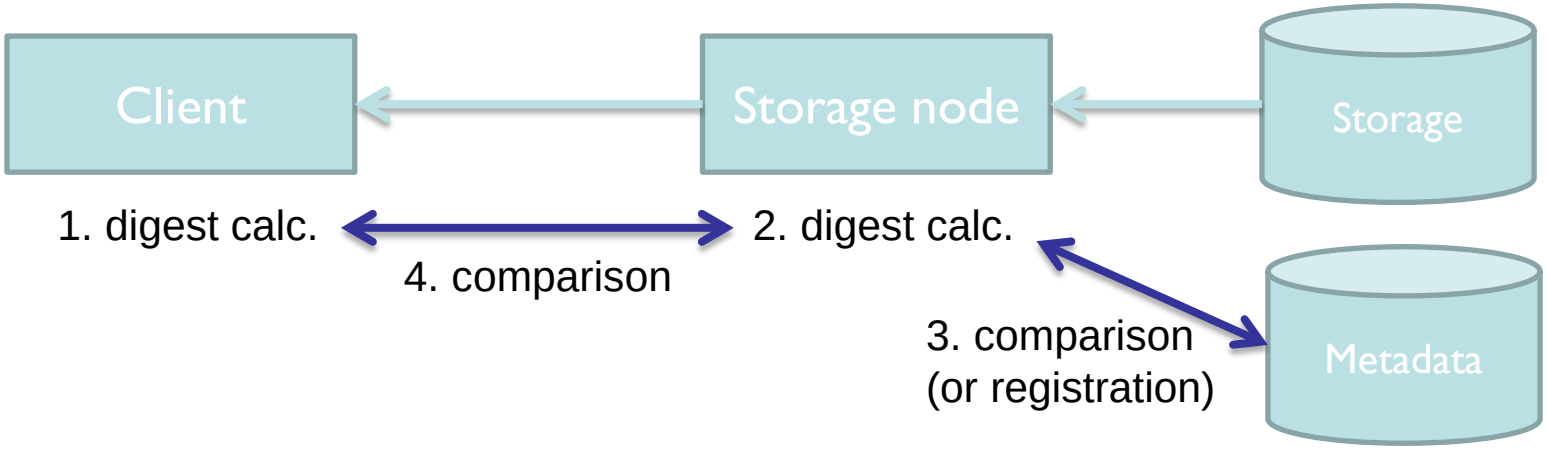


Write verify
after six hours

When checksum mismatch happens, moves to /lost+found

Data Integrity in Gfarm (3)

- Reading data



When checksum mismatch happens, read returns I/O error, and moves to /lost+found

⇨ Prevent from reading corrupted data

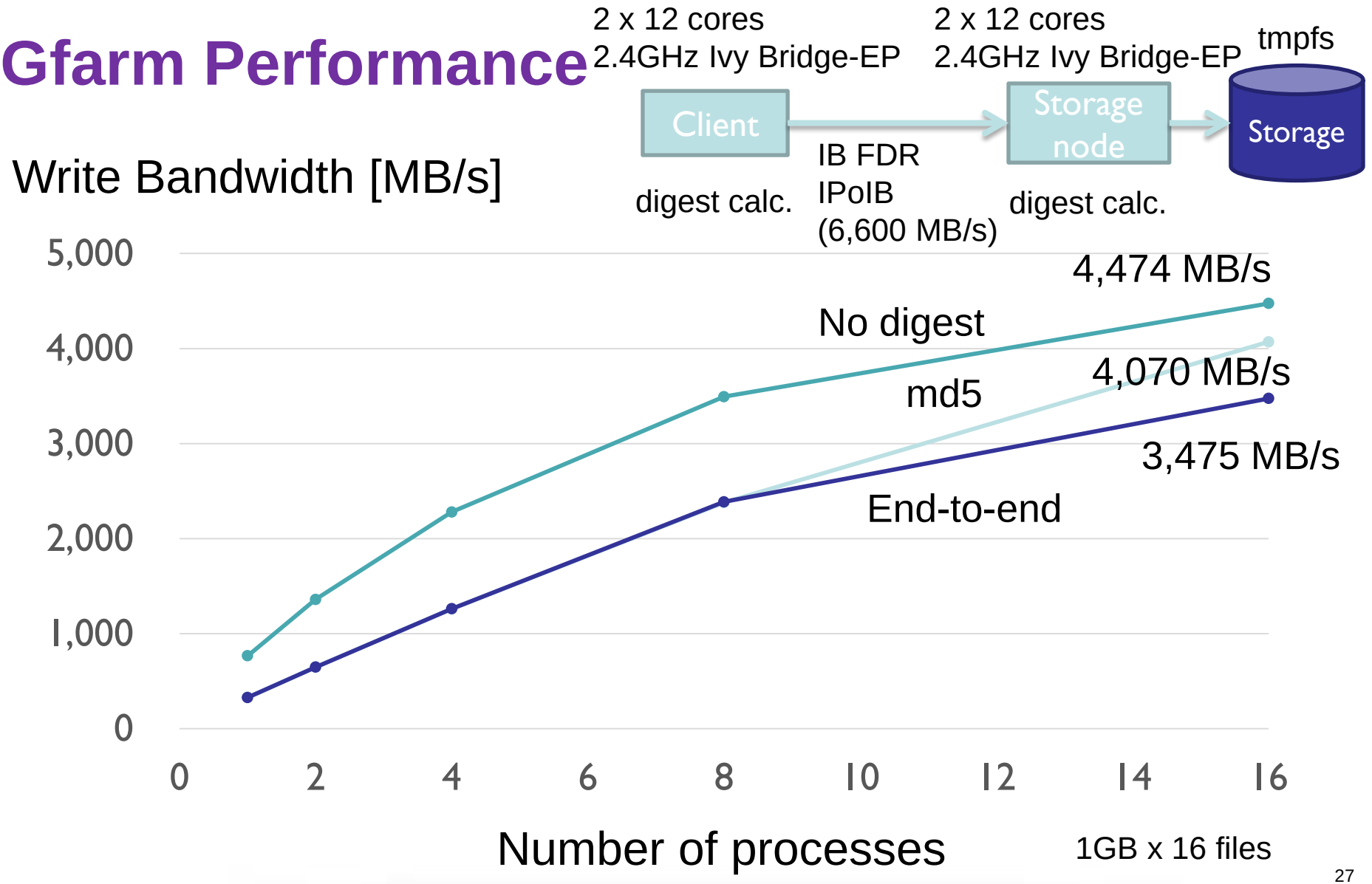
Automatic repair by file replicas

Openssl digest evaluation

	md5	sha1	sha256	sha512
2.4GHz Xeon E5-2695 v2 (Ivy Bridge-EP)	541	584	218	337
2.4GHz Xeon E5-2665 (Sandy Bridge-EP)	564	585	176	274
2.4GHz Xeon E5620 (Westmere-EP)	483	417	150	238

8KB block size, MB/s

Gfarm Performance



Case Study in JLDG

- 7.8 PB, 7 sites, 39 file servers
  - Nation-wide storage in Physics community
  - 7.2 PB used, 109 M files
- End-to-end data integrity by md5, and write verify enabled
- Aug 19~22, 2016
  - Six damaged files found by digest mismatch during write verify and replica creation
  - Still, there is no I/O error

Related work

- ZFS
  - Checksumming in each block
  - RAID-Z, not only replication, to recover data

Conclusion

- Silent data corruption is not rare, but often, in petascale storage
  - Checksumming is promising
- Gfarm file system detects SDC by write verify, replica creation, file read and (partial) scrubbing
- Native and required feature of file replicas in distant sites can correct it without any waste storage capacity
- SNIA-J Extreme Storage Society of Science Study

Contact

- Osamu Tatebe
  University of Tsukuba
  tatebe@cs.tsukuba.ac.jp