Infinite Memory Engine (IME)

Partial Non-Deterministic I/O System for Exascale

Jean-Thomas Acquaviva

June, 2015

Teratec Forum Palaiseau, France

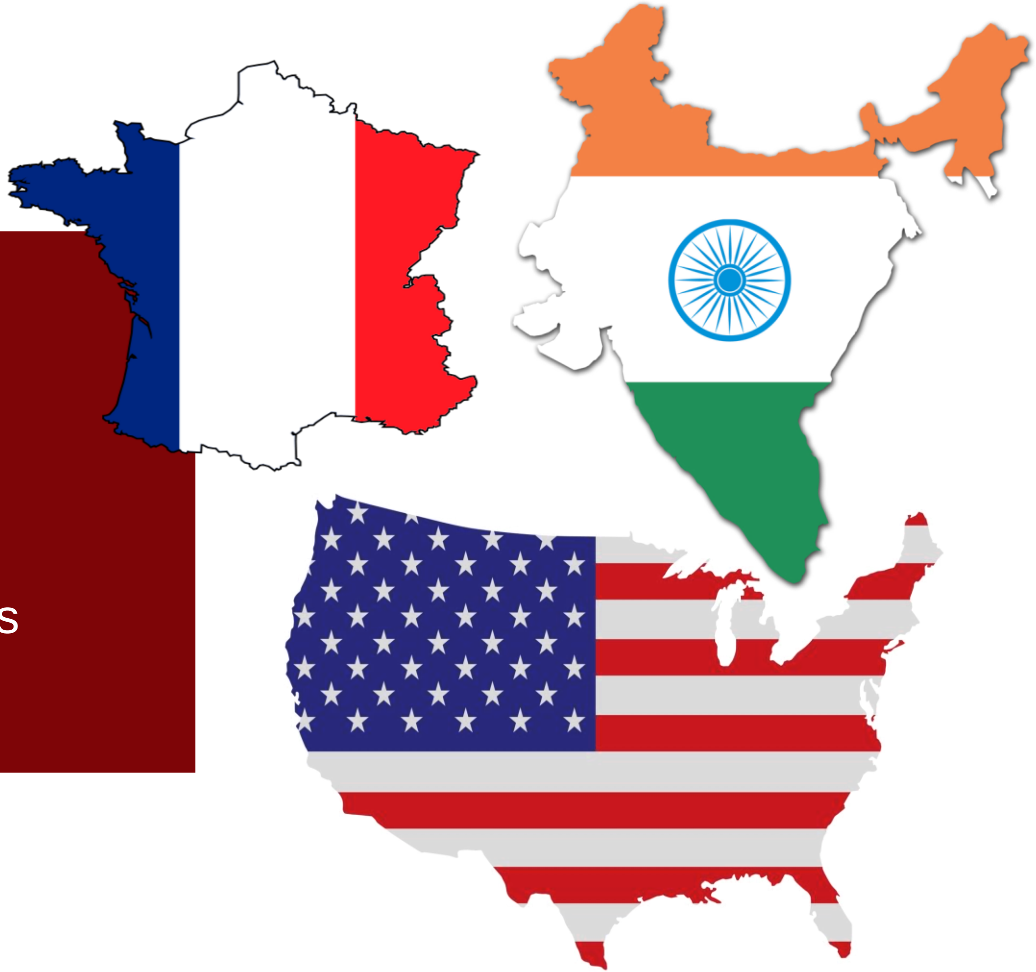
Who's IME?

Lead Architect: Paul Nowoczynski

R&D team on 3 continents

- Maryland, US
- Paris, France
- Pune, India

First project developed in the new Paris Advanced Technical Center



DataDirect™ NETWORKS

ddn.com

Why IME?
Bandwidth coarse estimation

Bandwidth needs next-gen pre-Exascale systems

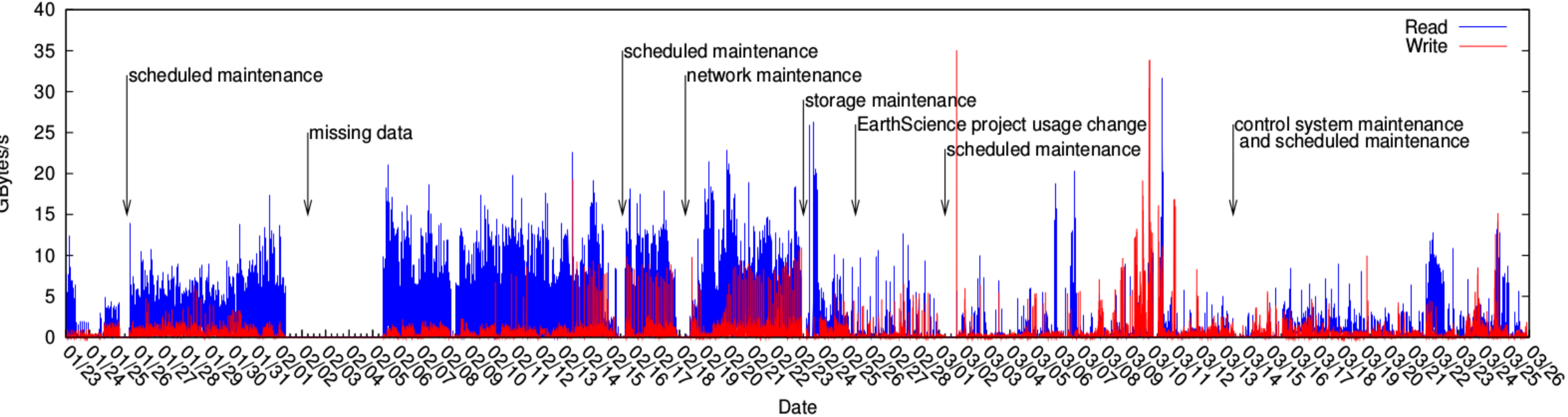
Rules of thumb

1/ Checkpointing less than 6 minutes per hours
2/ Checkpointing means draining half of system memory

Pre-Exascale system:
4 Petabyte → bandwidth requirement 5.6 TB/s

Oakridge,
Teng Wang, Weikuan Yu et al. " An Efficient Distributed Burst Buffer for Linux"

Why IME?
Bandwidth detailed view



99% of the time the IO sub-system is stressed bellow 30% of its bandwidth
70% of the time the system is stress under 5% of its peak bandwidth

Argone lab.

P. Carns, K. Harms et al., *Understanding and Improving Computational Science Storage Access through Continuous Characterization*

DataDirect NETWORKS

ddn.com

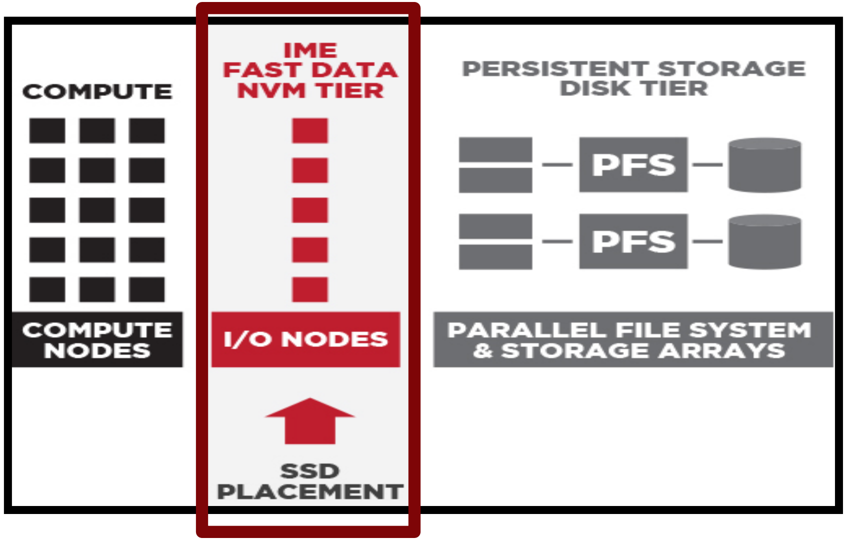
Why IME?
economics

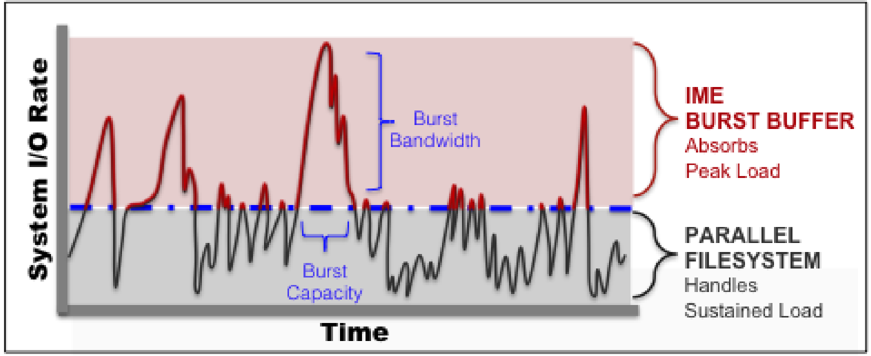
SSD reshuffle the parameters
Latency / 40 :   4ms → 0,1 ms
Bandwidth x 3: 150 → 450 MB/s
Capacity / 8 :   8 → 1TB.

Cost x 10 $ 0,05/Gbit → $0.04

+ SSD deprecation rate !!

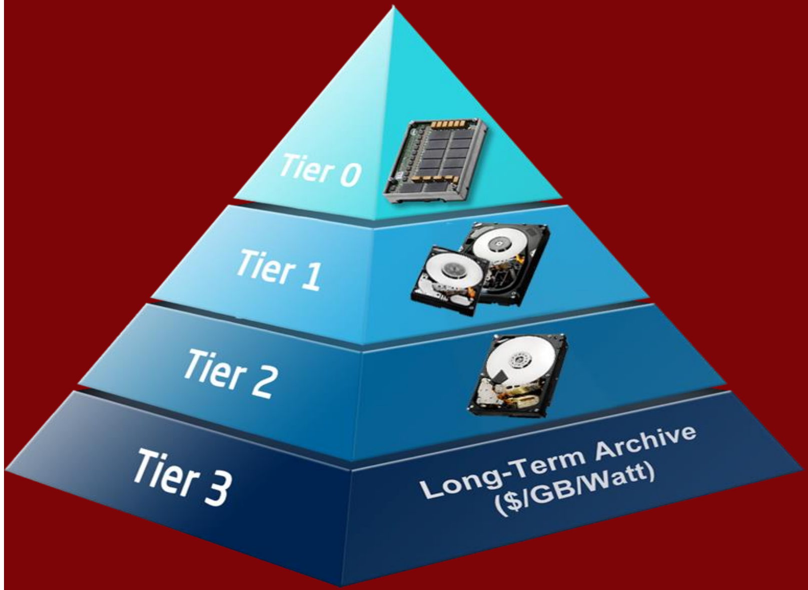
What can we do with a costly high bandwidth low latency technology ?



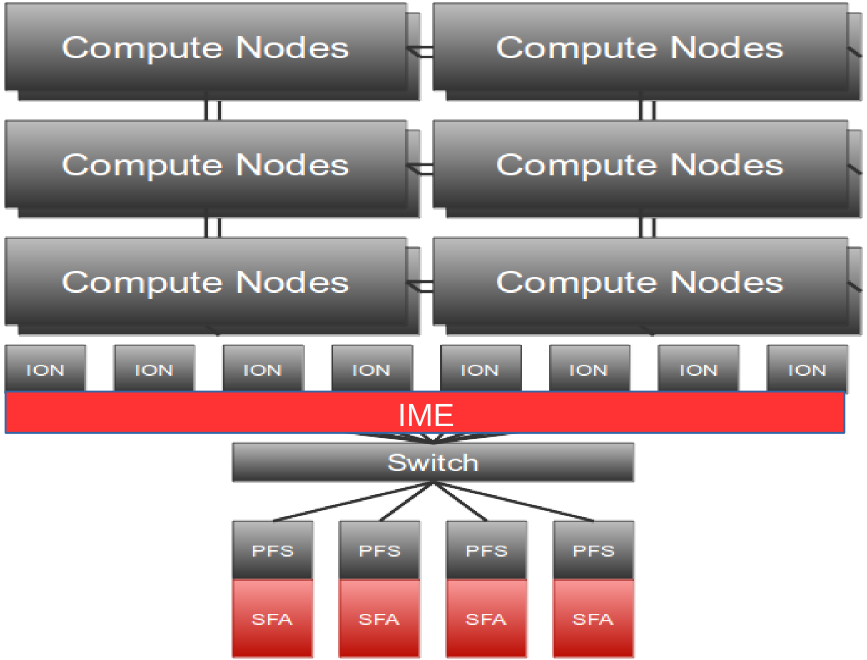


What is IME?
Distributed virtually shared coherent array of SSDs

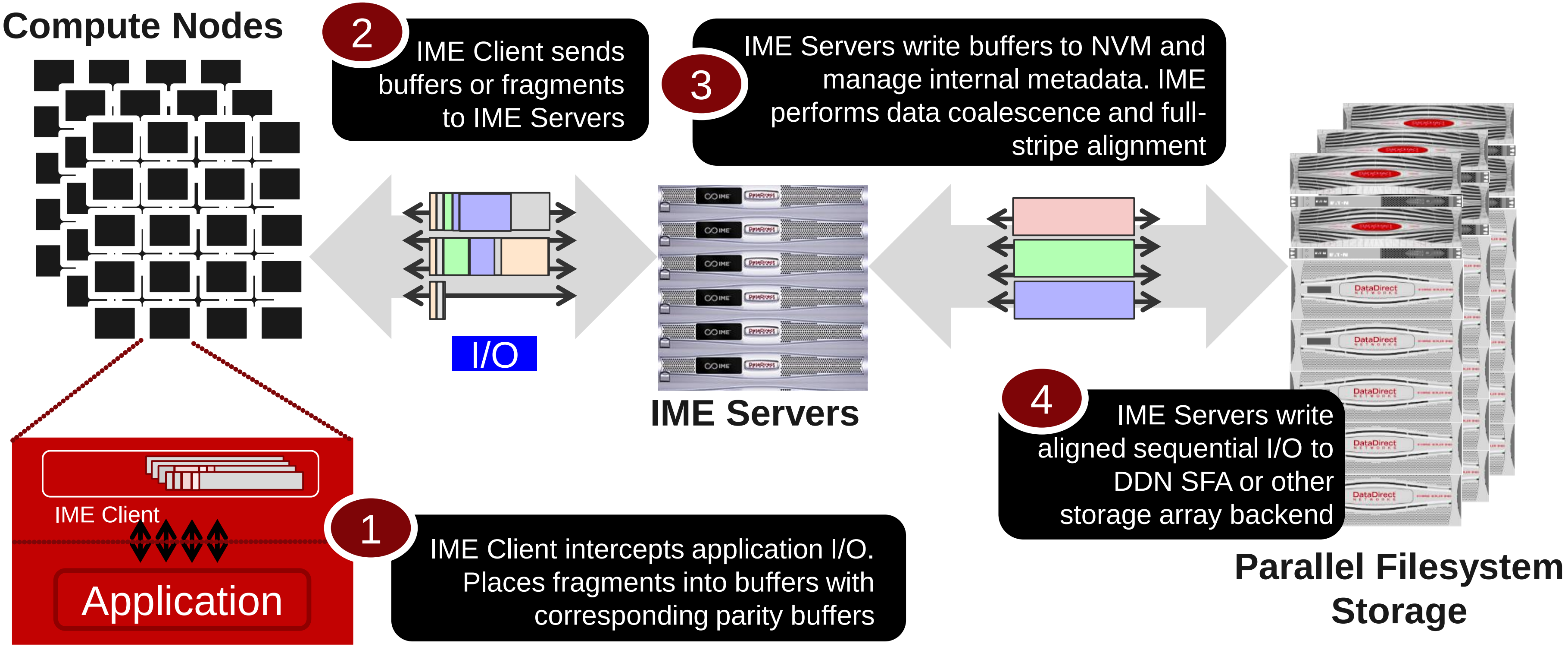
Hardware tiering



System architecture



IME New Data Flow, New software stack



IME means Scalability and Reliability

Alleviate lock pressure: byte addressable, no page lock

Every IO is recorded as a frag:

File ID

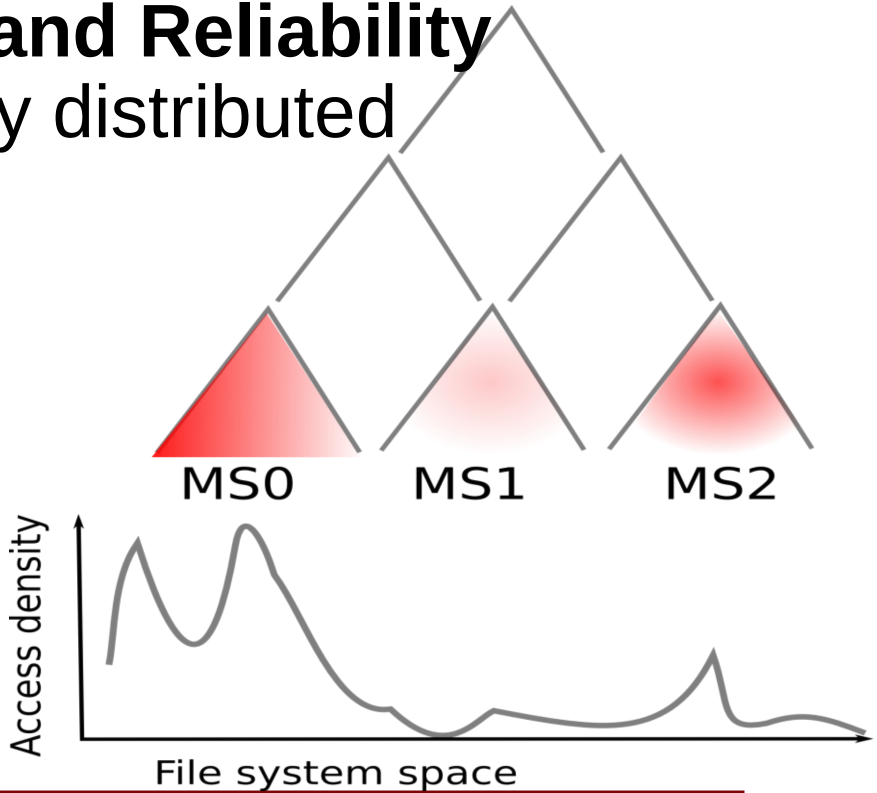
Address

Length

Frags are aggregated in large network buffer

Frags are weakly ordered, weak Posix compliance

IME means Scalability and Reliability
Metadata directory is fully distributed

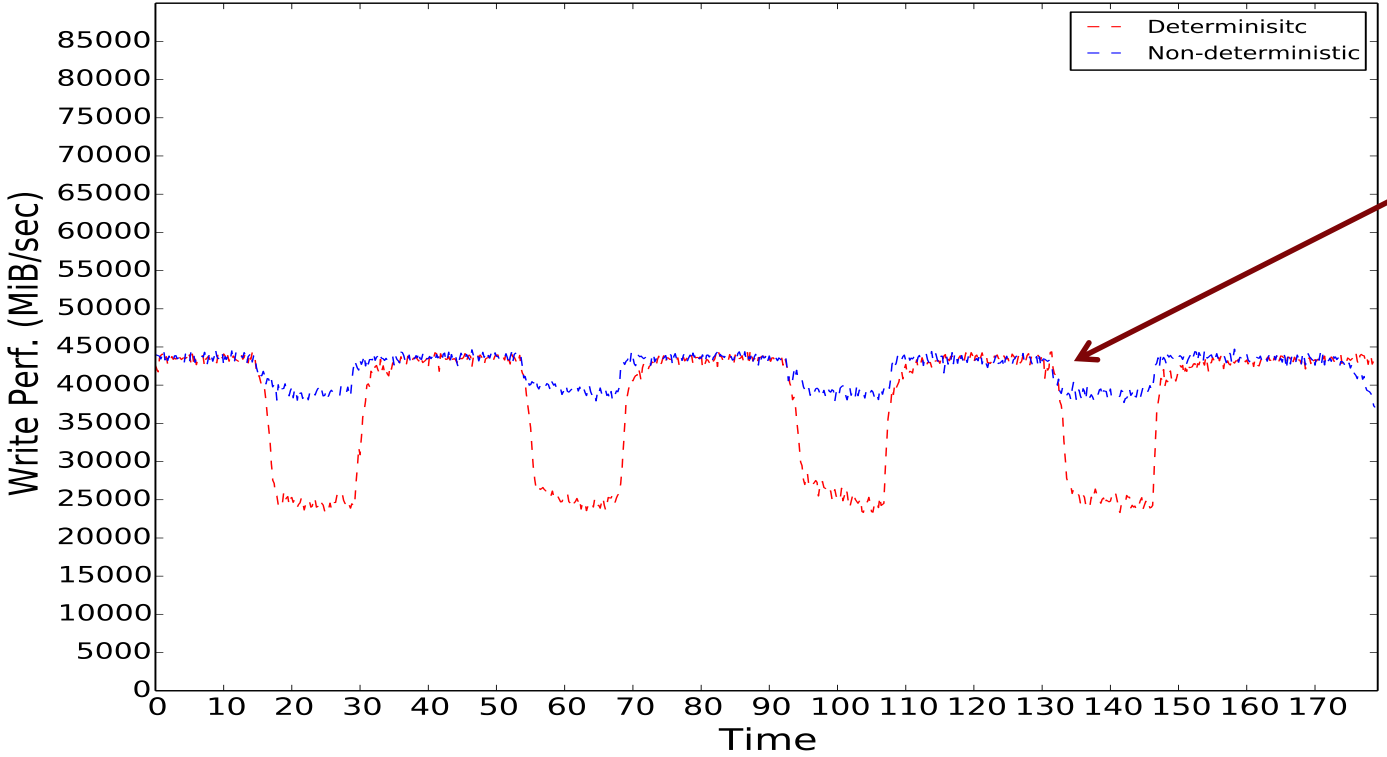


No centralized directory
Distribution independent from structure
→ no hot directory issue
Common knowledge ensured through hash
→ CRUSH like hash function

Santa Cruz U., Supercomputing 2006
Sage Weil, Scott Brandt et al. *CRUSH: Controlled, Scalable, Decentralized Placement of Replicated Data*

From fault tolerance to resilience

Leverage server fault tolerance for load balancing



Adaptive heuristic learns "quickly"

4x Performance Lost with Non-adaptive

IME means Scalability and Reliability

Parity and data protection is not going to scale

IME software only
Erasure coding not scalable on server side
Done on the IME client side
- **Vector instructions**
- **Heavily multithreaded**

Pittsburgh Supercomputing Center., Supercomputing 2006
Paul Nowoczynski et al. Zest Checkpoint Storage System for Large Supercomputers

IME Fault Recovery

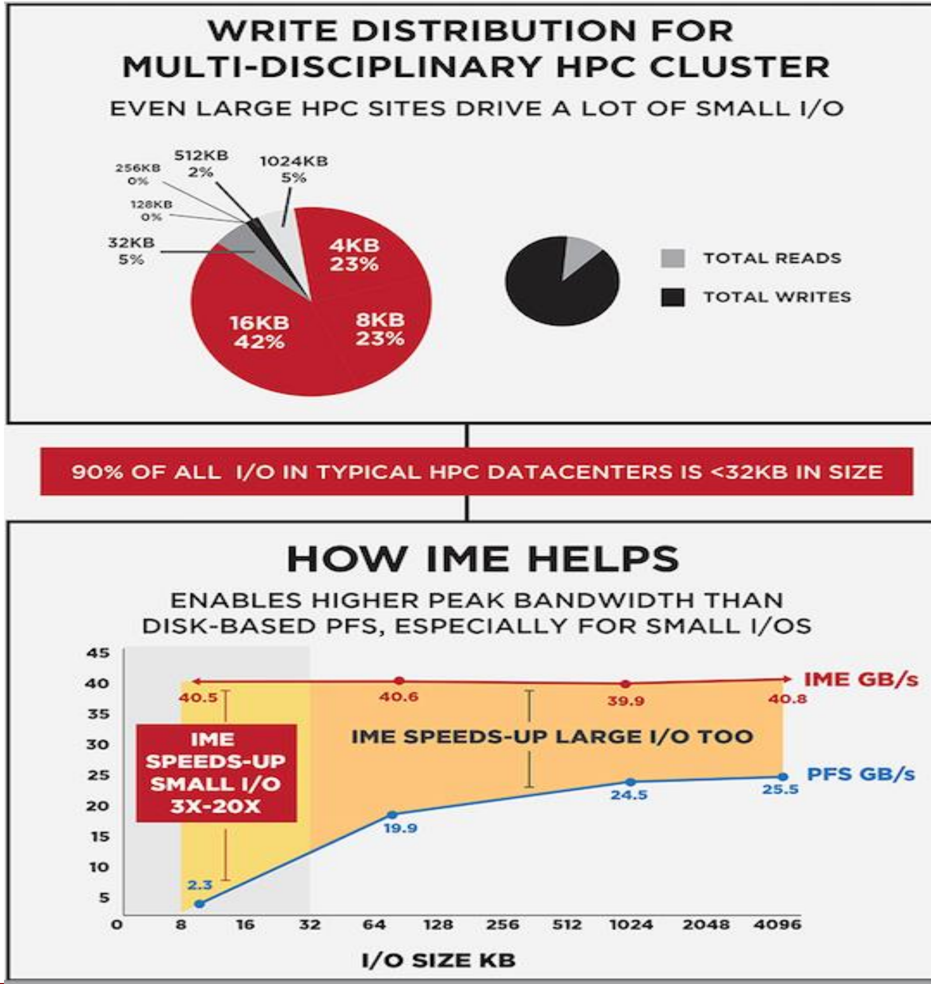
- SSD Failure
  - Data recovery responsibility belongs to IME server with failed device
  - Recovered data are rewritten to remaining SSDs

- IME server failure
  - Recovery performed by remaining servers
    - Aka “Node Ejection”
  - Fully declustered
    - Remaining servers share rebuild processing

IME Boost

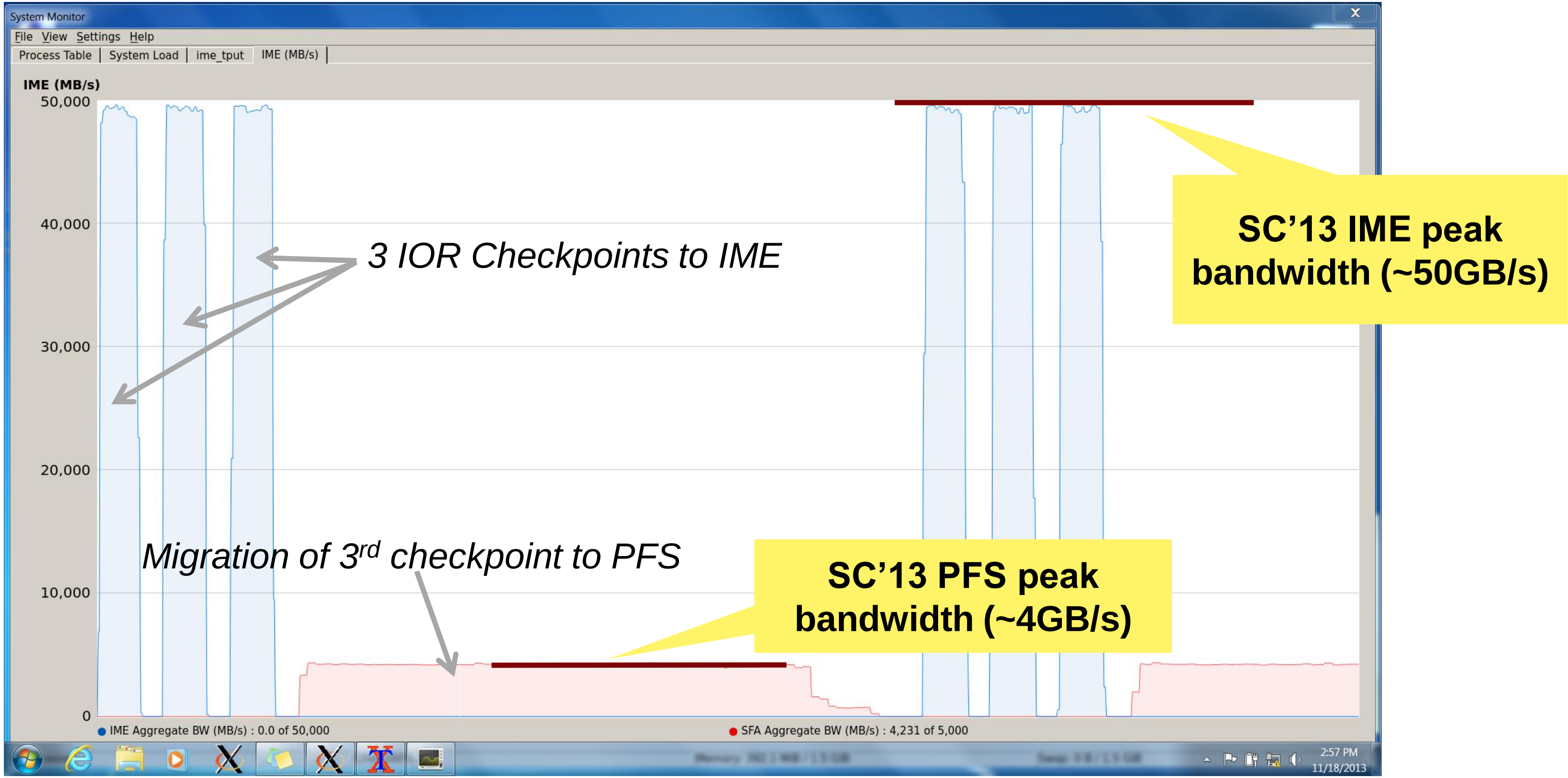
Accelerates applications
→ especially small or mal-aligned I/O

No page lock
Client coalescing:
→ Bulk RPC transfer
→ Save SSD write cycle
Server flush scheme
→ tune for PFS parameters



IME Boost

IOR SSF 4k interleaved writes (SC'13)



IME Boost

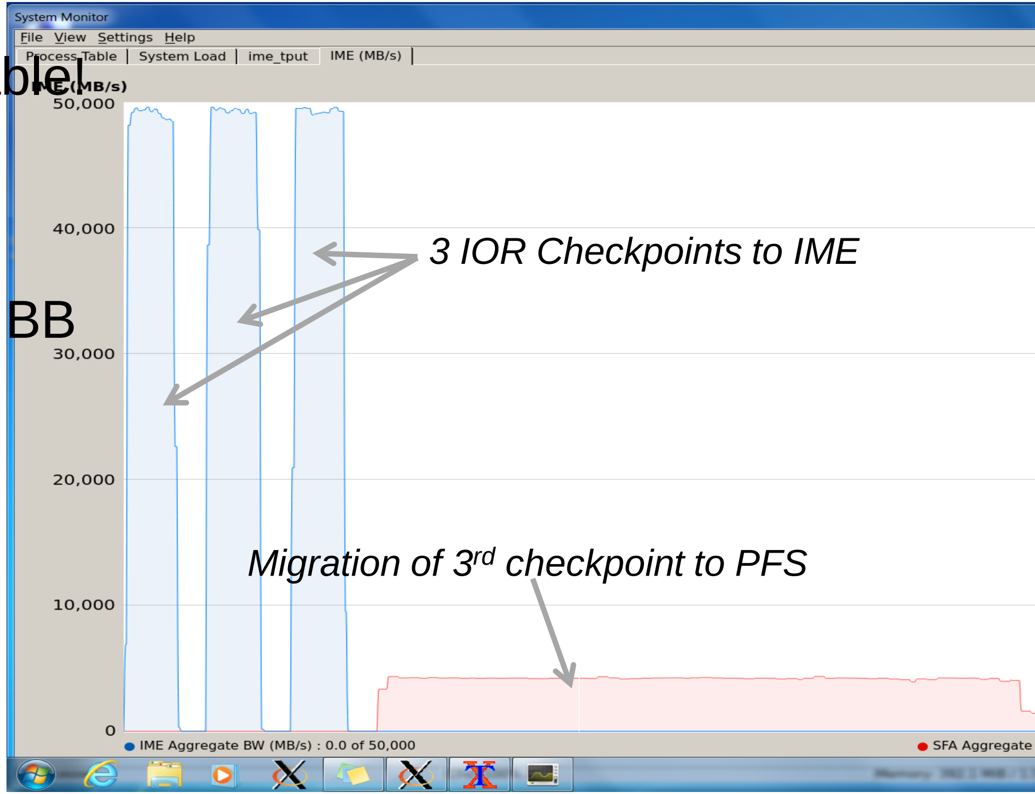
Reduce data movement

Copying all data to PFS is not desirable!

- Reduce data movement

- 3rd checkpoint should be freed from BB following copy to PFS

- ***Explicit management is required***



Future: Node Local NVM

Node local provides a new set of challenges.

- Affiliation of file components to specific compute nodes must be expressed

- Will a job actually run on the compute nodes where data has been staged-in?
  - At large issue in HPC deployments

- Blur the distinction between client and server.
  - Performance instructiveness in compute node

IME take away

- Extra tier of SSD in storage hierarchy
  - → Re-design the software stack to address real issues
  - → Resilience and scalability
- Keep data as close as possible from compute node
- Pluggable with other IO storage technology
  - → IO libraries
  - → Object storage
- Pave the way for future evolutions
  - → A component of the Exascale storage stack

Thanks !

IME Approach

Distributed Hash Table + Log Structuring