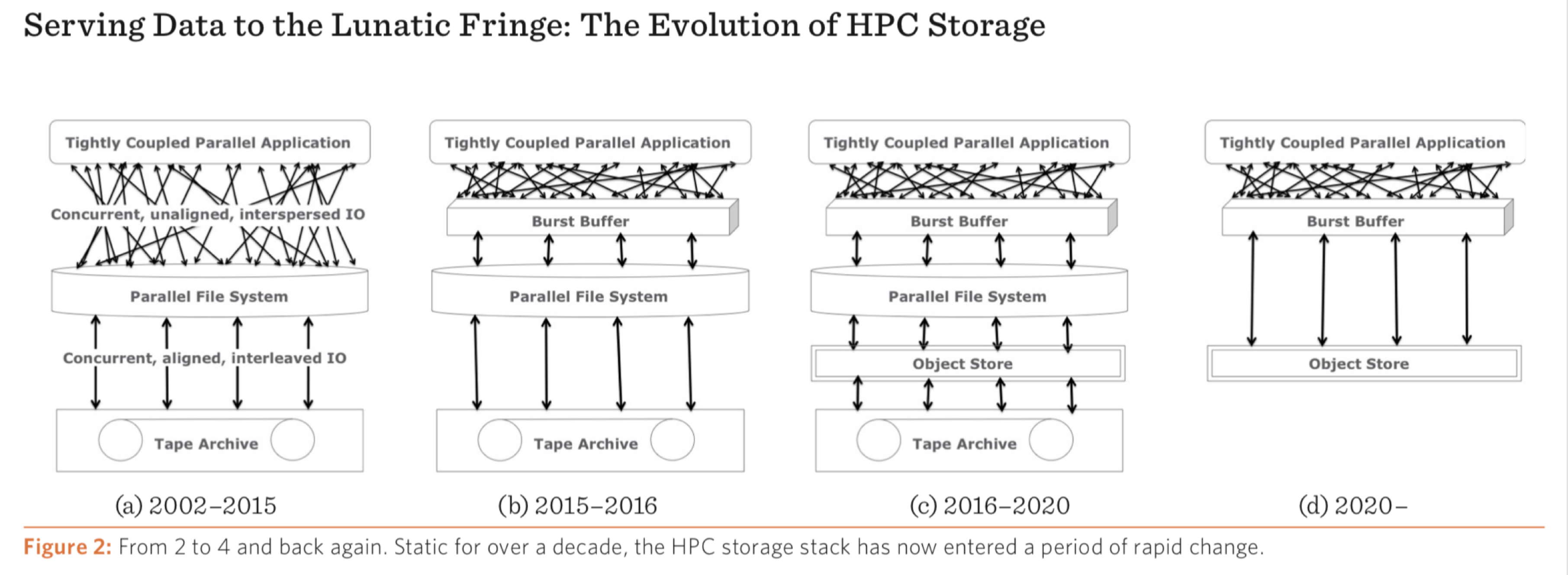
Heterogeneity, Schmeterogeneity.
Object, Schmobject.
Long Live POSIX

John Bent, Global Field CTO

SOS23, March 28, 2019

Predictions From LANL in 2016: Serving Data to the Lunatic Fringe



Serving Data to the Lunatic Fringe: The Evolution of HPC Storage

Tightly Coupled Parallel Application
Concurrent, unaligned, interspersed IO
Parallel File System
Concurrent, aligned, interleaved IO
Tape Archive
Tightly Coupled Parallel Application
Burst Buffer
Parallel File System
Tape Archive
Tightly Coupled Parallel Application
Burst Buffer
Parallel File System
Object Store
Tape Archive
Tightly Coupled Parallel Application
Burst Buffer
Object Store

(a) 2002–2015

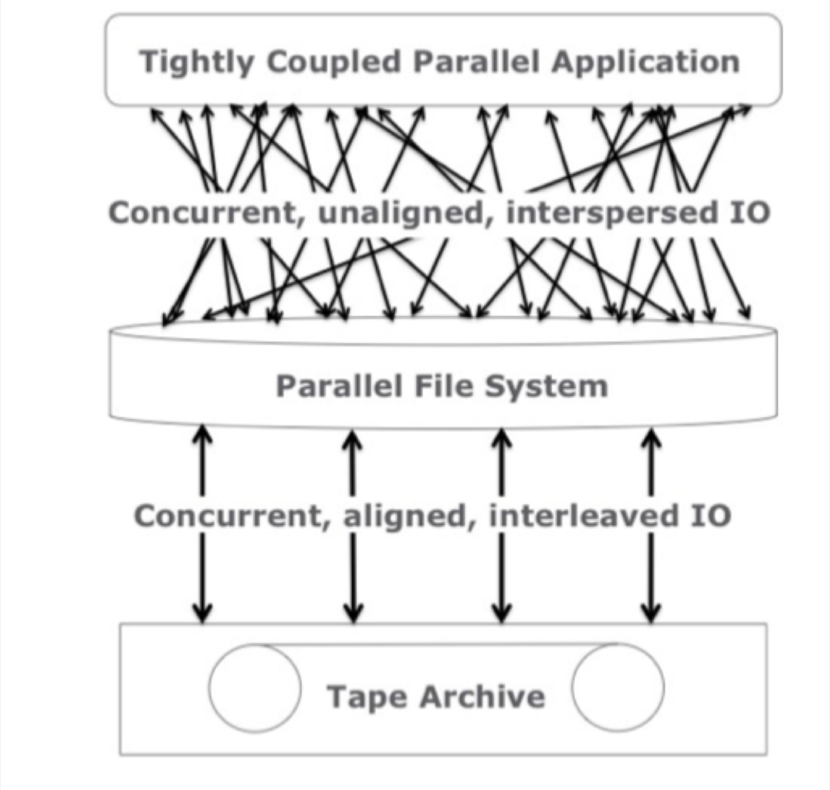
(b) 2015–2016

(c) 2016–2020

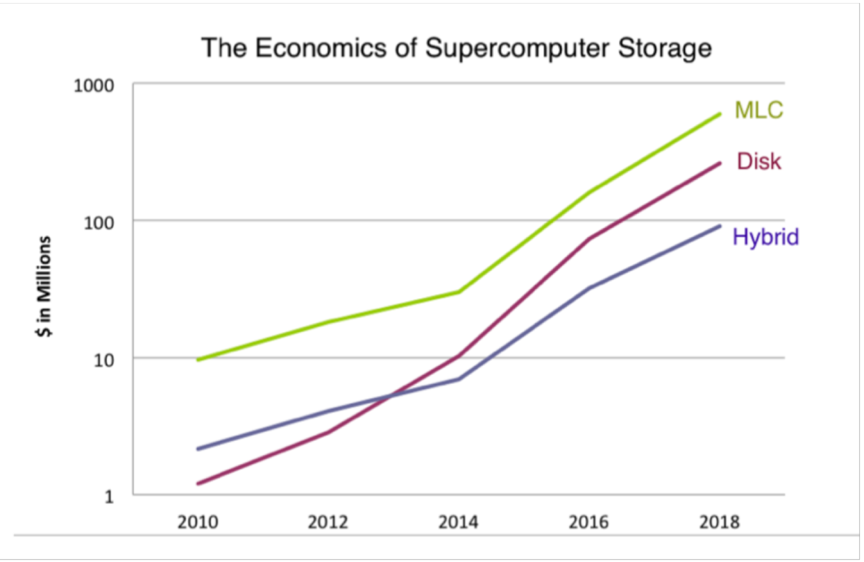
(d) 2020–

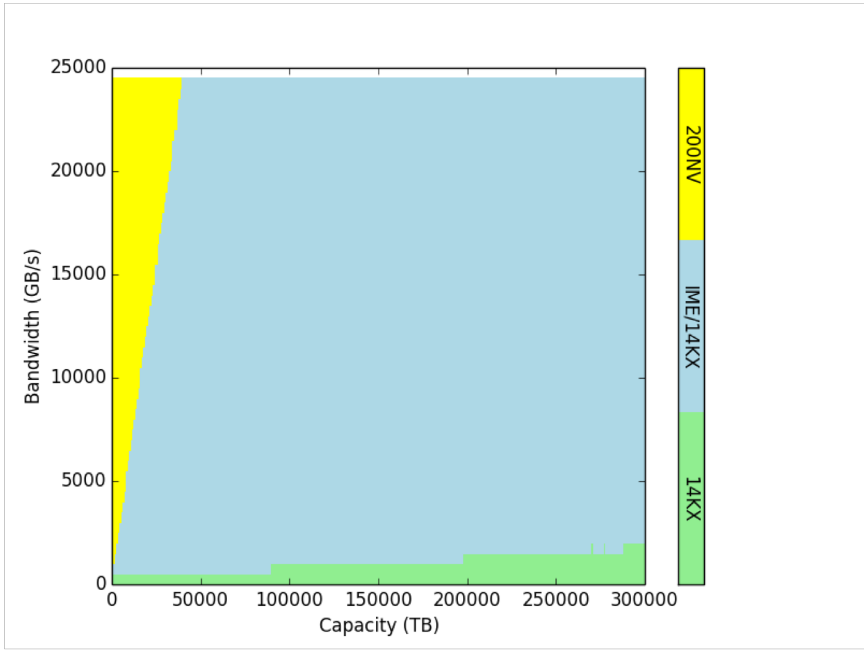
Figure 2: From 2 to 4 and back again. Static for over a decade, the HPC storage stack has now entered a period of rapid change.

Predictions From LANL in 2016: Serving Data to the Lunatic Fringe


Tightly Coupled Parallel Application
Concurrent, unaligned, interspersed IO
Parallel File System
Concurrent, aligned, interleaved IO
Tape Archive

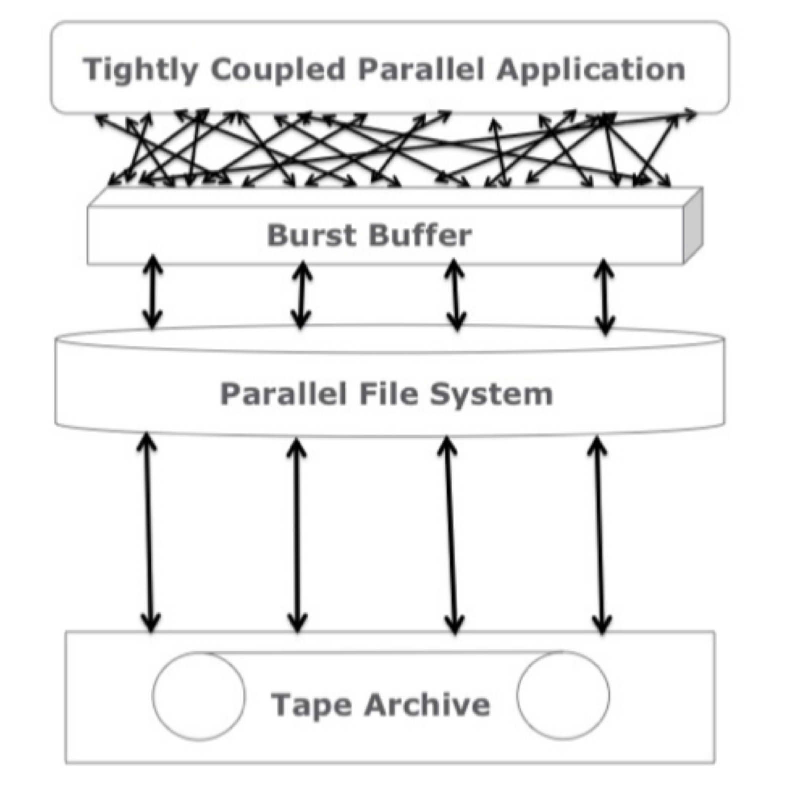
(a) 2002–2015



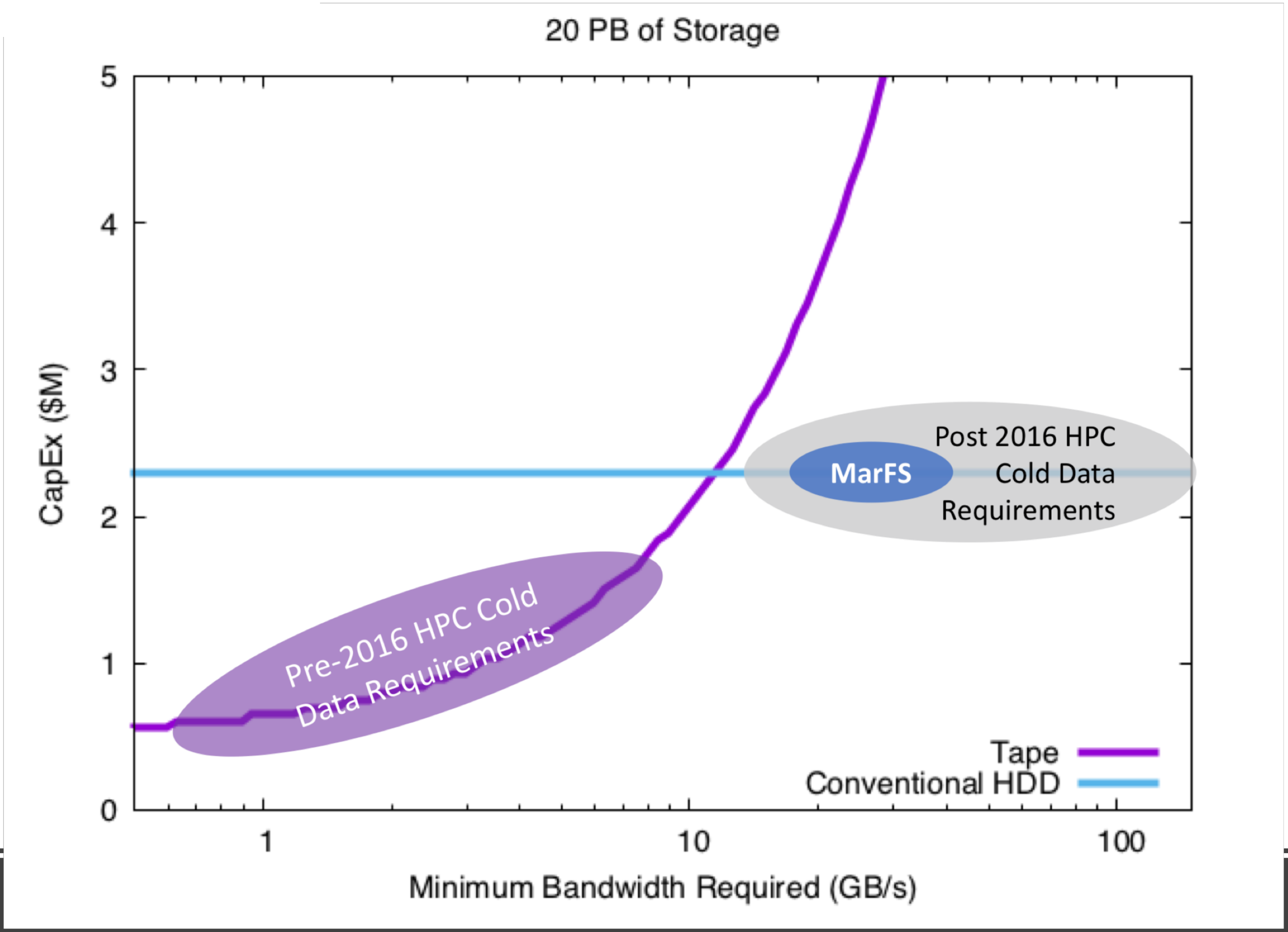


This architecture was imperiled by SSD economics

Predictions From LANL in 2016: Serving Data to the Lunatic Fringe

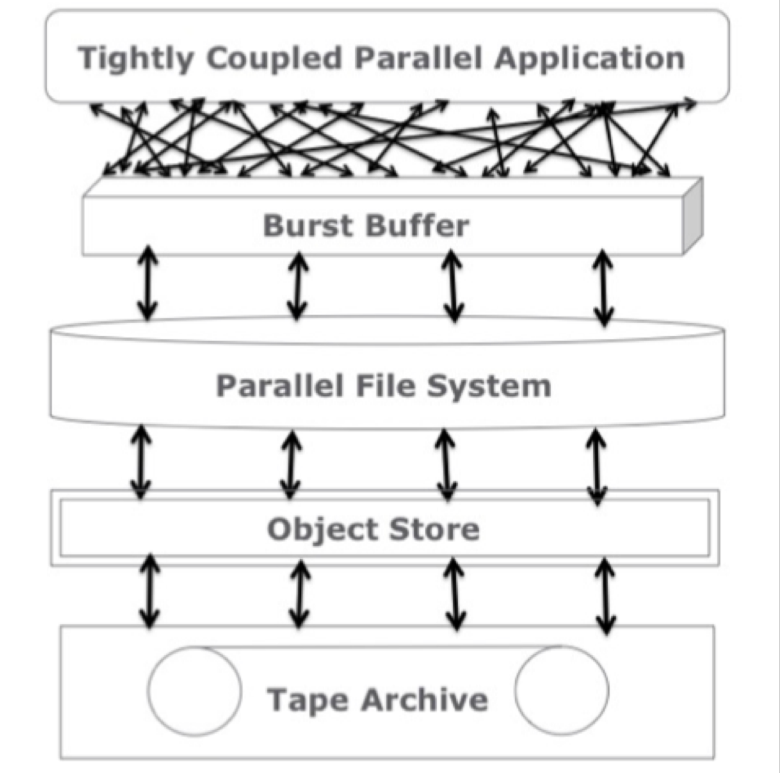


(b) 2015–2016



This architecture becomes imperiled by tape economics

Predictions From LANL in 2016: Serving Data to the Lunatic Fringe



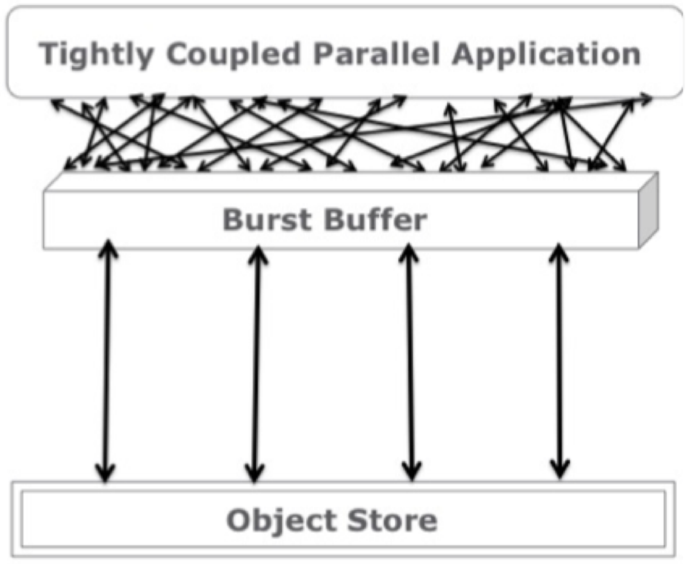
(c) 2016–2020

"DOE doesn't want tiers. Tiers are an unfortunate accident of economics. DOE wants infinite memory and a system without unplanned interrupts.

Just remember this:
The fewer tiers, the fewer tears."

This architecture becomes imperiled by Lang's Law

Predictions From LANL in 2016: Serving Data to the Lunatic Fringe



(d) 2020–

Predictions from NERSC: Storage 2020

Four Tiers *Three Tiers* *Two Tiers*

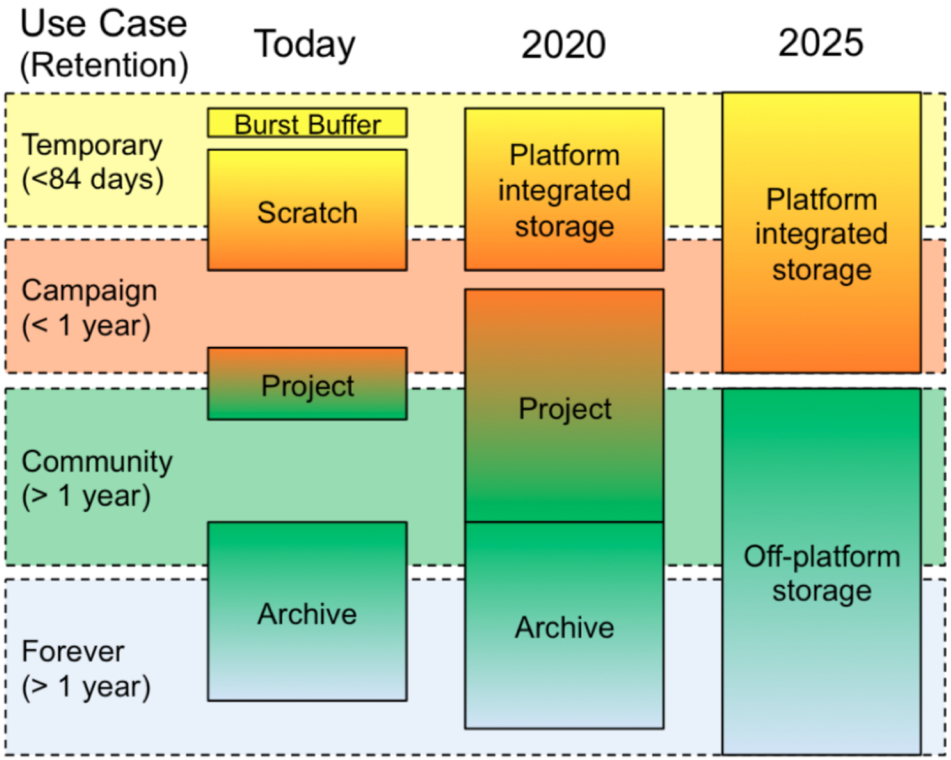


FIGURE 11. EVOLUTION OF THE NERSC STORAGE HIERARCHY BETWEEN TODAY AND 2025.

Re-examining our predictions

- Object?
  - My thinking has evolved
- Economics?
  - Yep

“What?”

“Do you have any water?”

“Build an object store.”

Typical Object Requirements ("Object Schmobject; long live POSIX")

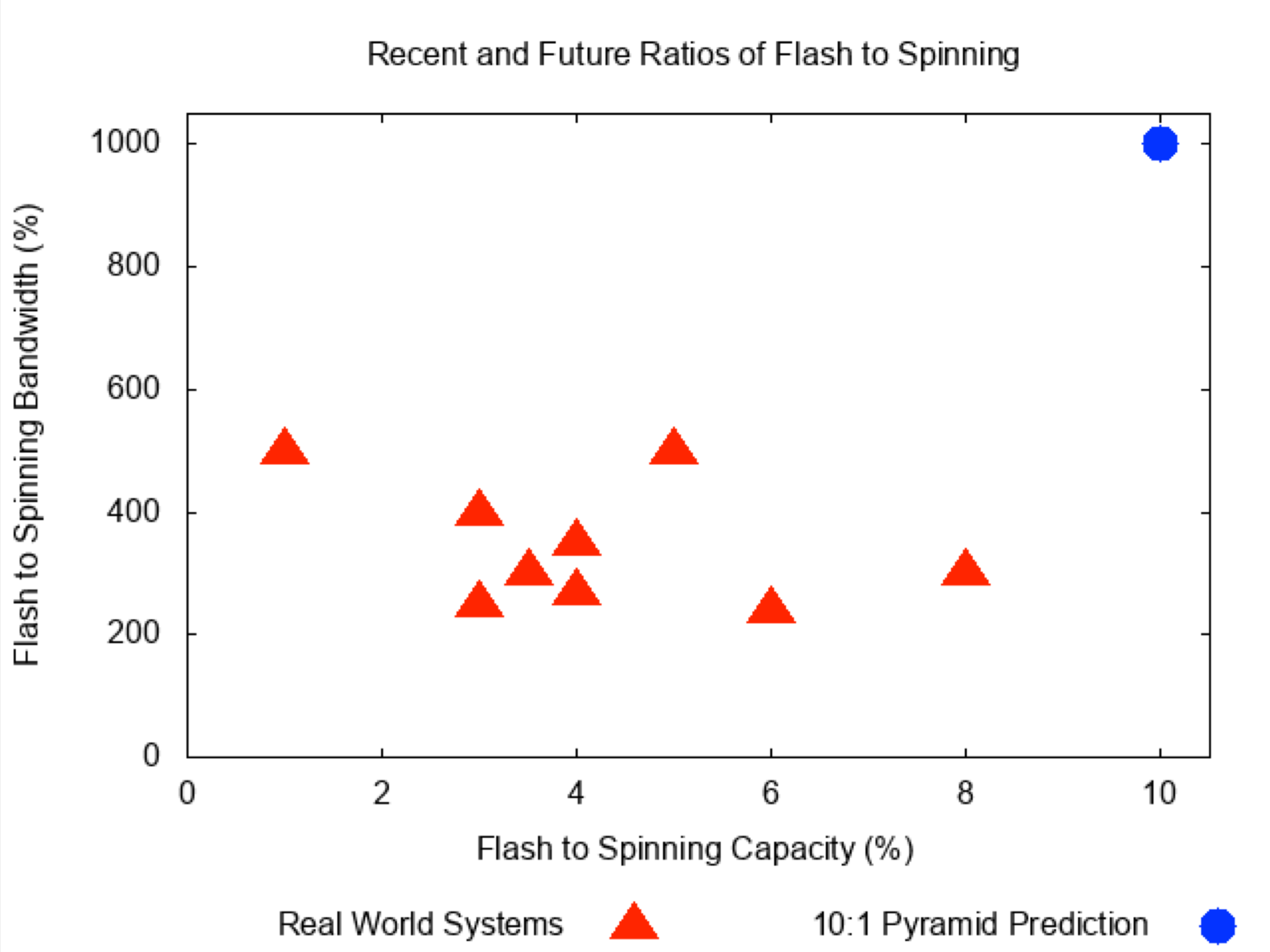
- Immutable, transactional get/put, trillions of objects
- . . . .
- Named objects
- Group objects into logical collections
- Nest logical collections within each other
- Have the same object appear within multiple collections
- Tag objects

- Object is a subset of file
  - There is no application which uniquely requires object semantics
  - O_TMPFILE and rename are useful primitives
- Object requirements grow as humans use them
  - Eventually they become file requirements
- We do not live on a deserted desert island
  - We have two decades experience building parallel file systems

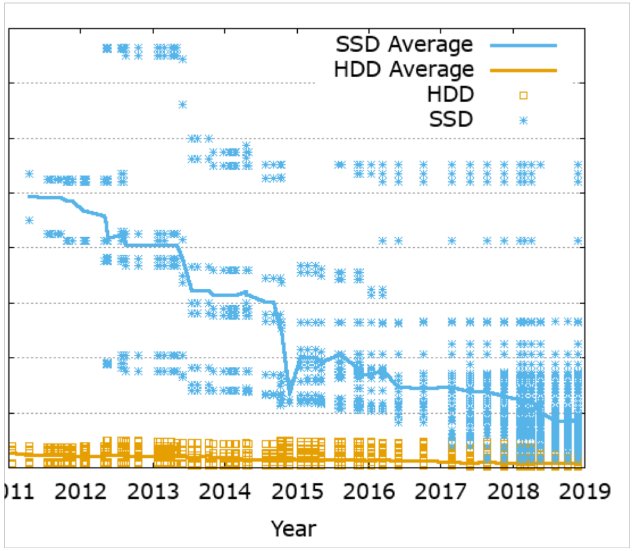
- RELEVANT LESSON FROM OBJECT STORES?
  - POSIX relaxation is useful

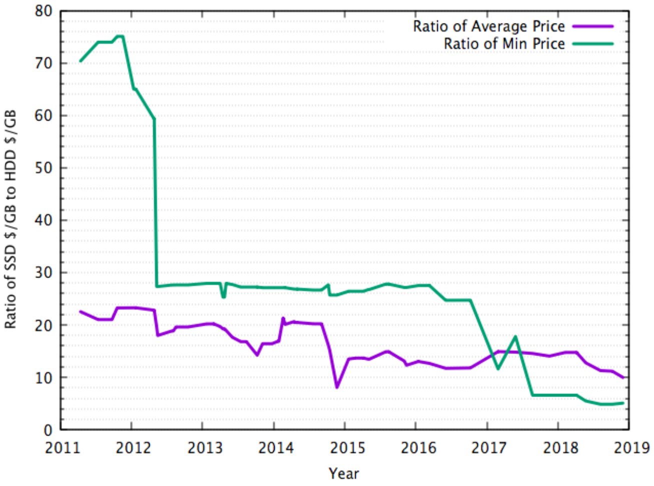
Economics?
Burst Buffers [“Flash Acceleration Layers”] Have Arrived

	Capacity Filesystem	Capacity IME	Performance Filesystem	Performance IME
IME @ TACC	50 PB	±2.5 PB (±5%)	300 GB/s	1500 GB/s (5x)
IME @ KISTI	20 PB	0.8 PB (±4%)	300 GB/s	800 GB/s (2.7x)
IME @ Large EU HPC	25 PB	±750 Tbyte (±3%)	100 GB/s	250 GB/s (2.5x)
IME @ JCAHPC	26 PB	0.9 PB (±3.5%)	500 GB/s	1500 GB/s (3x)
IME @ EPFL BBP	3 PB	80 Tbyte (±2.7%)	20 GB/s	80 GB/s (4x)
CORAL2 LLNL	400 PB	16 PB (4%)	4.8 TB/s	17 TB/s (3.5x)
CORAL2 Oak Ridge	800 PB	8 PB (1%)	2.8 TB/s	15 TB/s (5.3x)
KAUST ShaheenII	17 PB	1.5 PB (8%)	500 GB/s	1.5 TB/s (3x)
NERSC CORI	30 PB	1.8 PB (6%)	750 GB/s	1.7 TB/s (2.3x)
ORNL Summit	250 PB	7.3 PB (3%)	2.5 TB/s	10 TB/s (4x)


Recent and Future Ratios of Flash to Spinning
Flash to Spinning Bandwidth (%)
1000
800
600
400
200
0
0
2
4
6
8
10
Flash to Spinning Capacity (%)
Real World Systems ▲
10:1 Pyramid Prediction ●

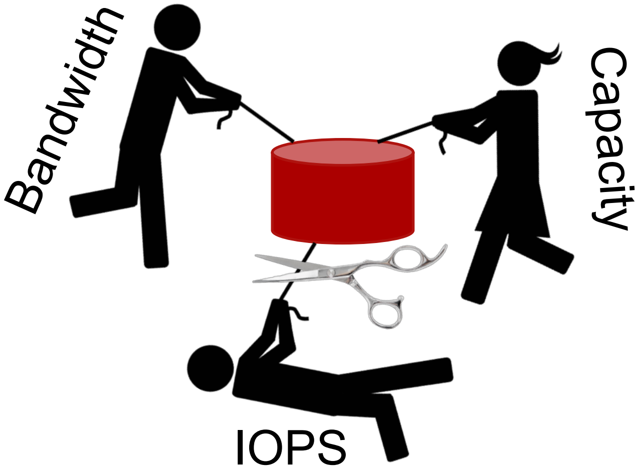
The Foreseeable Future Remains Tiered





Data Center

Consumer Market



Tiering Schmiering

All due respect to Lang’s Law (‘fewer tiers, fewer tears’), tiering is a (mostly) solved problem.

Buffer-caching is a (mostly) solved problem!

Russel Kirsch developed it for the SEAC in 1952.

Flash Acceleration Layer Usage and Tiering Workflows in Traditional HPC

Lee Ward, Use Cases or BB Roles, Informal Burst Buffer Presentation via Sandia National Laboratories, 2015.

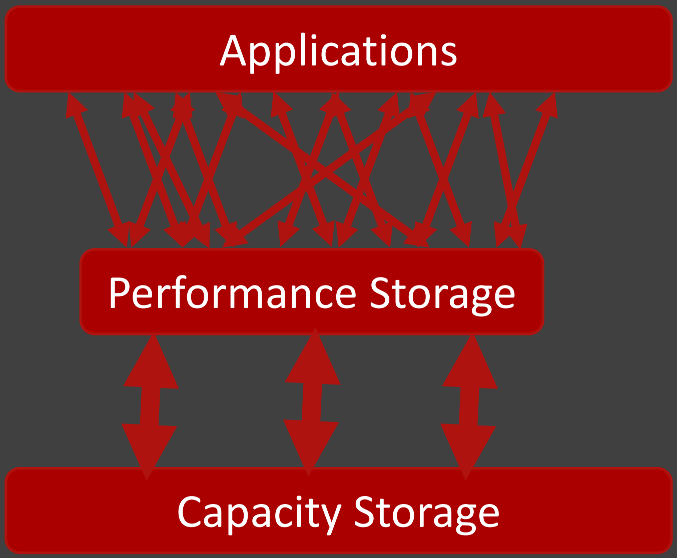
Development of a Burst Buffer System for Data-Intensive Applications, Teng Wang, Sarp Oral, Michael Pritchard, Kevin Vasko, Weikuan Yu, 2015.

An Operational Perspective on a Hybrid and Heterogeneous Cray XC50 System. Sadaf Alam, Nicola Bianchi, Nicholas Cardo, Matteo Chesi, Miguel Gila, Stefano Gorini, Mark Klein, Colin McMurtrie, Marco Passerini, Carmelo Ponti, Fabio Verzelloni, 2017.

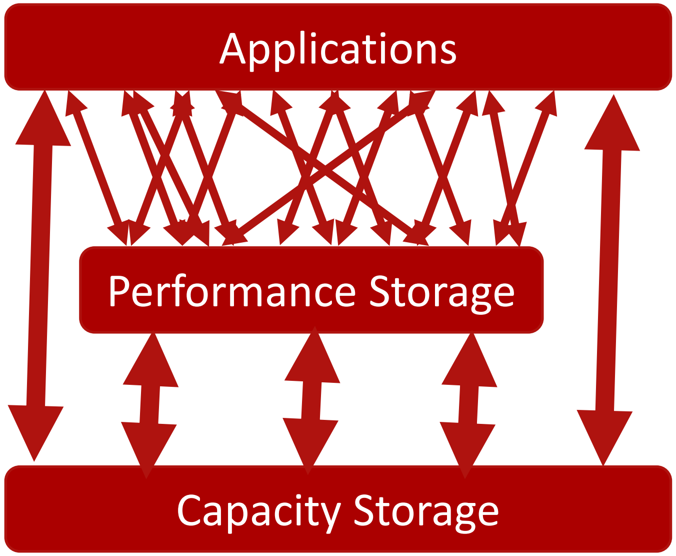
Challenges and Considerations for Utilizing Burst Buffers in High-Performance Computing, Melissa Romanus, Robert Ross, Manish Parashar, 2018.

1. Checkpoint-Restart
2. In-situ/transit viz/analysis
3. Accelerated reads (pre-stage)
4. Out-of-core

A Subtle Shift in Perception



Unnecessarily Strict Tiering



Relaxed Tiering

Thanks to Nic Dube and Jeff Kuenh

The future is bright and mostly as we predicted it.

Don’t be scared of POSIX; embrace relaxations.

Don’t be scared of tiering; embrace relaxations.

Thanks!

jbent@ddn.com