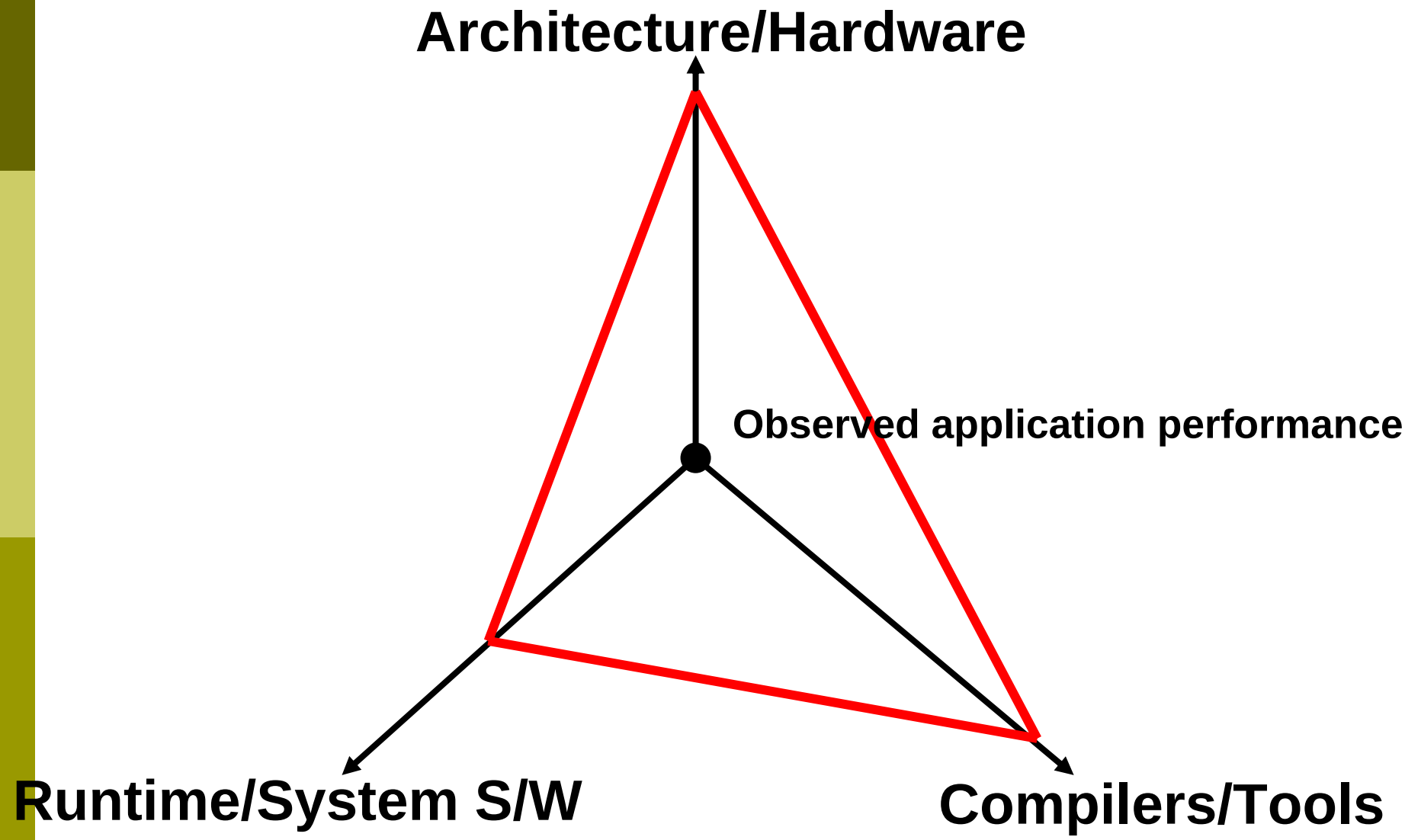
System Software Issues for the Future

Alok Choudhary, Professor
Director: Center for Ultra-Scale Computing and Information Security
Dept. of Electrical & Computer Engineering
And Kellogg School of Management
Northwestern University
choudhar@ece.northwestern.edu

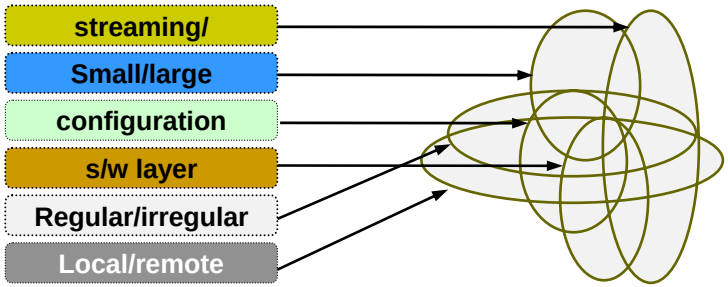
HPC Foundation



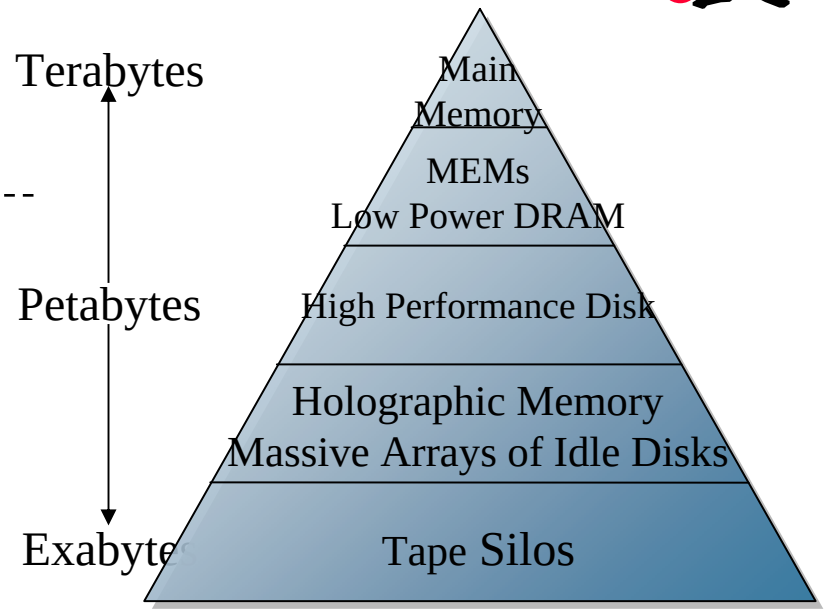
Runtime Systems – Challenges

User specifies how

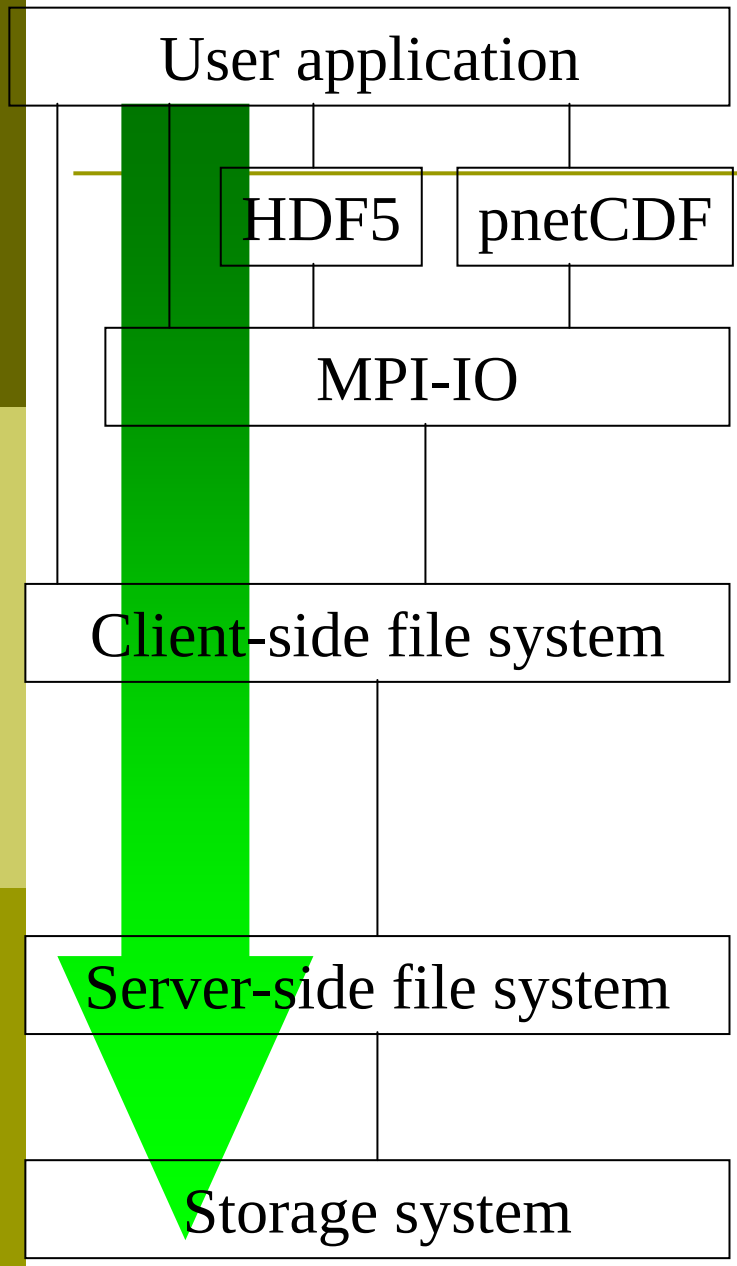
Complex non-portable optimization space



- **user burdened**
- **Ineffective interfaces**
- **Non-communicating layers**



Emerging Storage Hierarchy



Access patterns: shared files, individual files, data partitioning, check-pointing, data structures, inter-data relationship

Data types (byte-alignment), data structures (flexible dimensionality), hierarchical data model

Collectives, independents
I/O hints: access style (read_once, write_mostly, sequential, random, …), collective buffering, chunking, striping

Caching, fault tolerance, read-ahead, write-behind, I/O load balance, wide-area, heterogeneous FS support, thread-safe

Open mode (O_RDONLY, O_WRONLY, O_SYNC), file status, locking, flushing, cache invalidation
Machine dependent: data shipping, sparse access, double buffering

application-aware caching, pre-fetching, file grouping, "vector of bytes", flexible caching control, object-based data alignment, memory-file layout mapping, more control over hardware, Shared file descriptors,

Read-ahead, write-behind, metadata management, file striping, security, redundancy

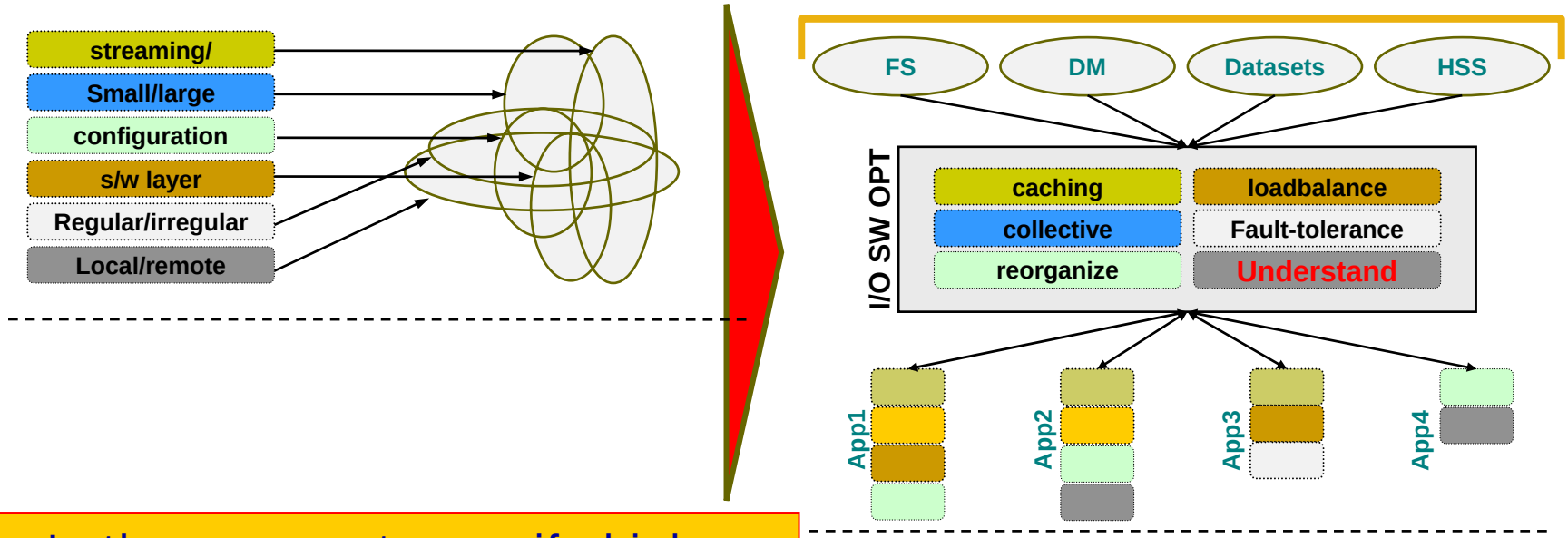
Group locks, flexible locking control, scalable metadata management, zero-copying, QoS, Shared file descriptors,

Access base on : file blocks, objects Scheduling, aggregation

Active storage: data filtering,object-based/hierarchical storage management, indexing, mining, power-management

Decouple "What" from "How"

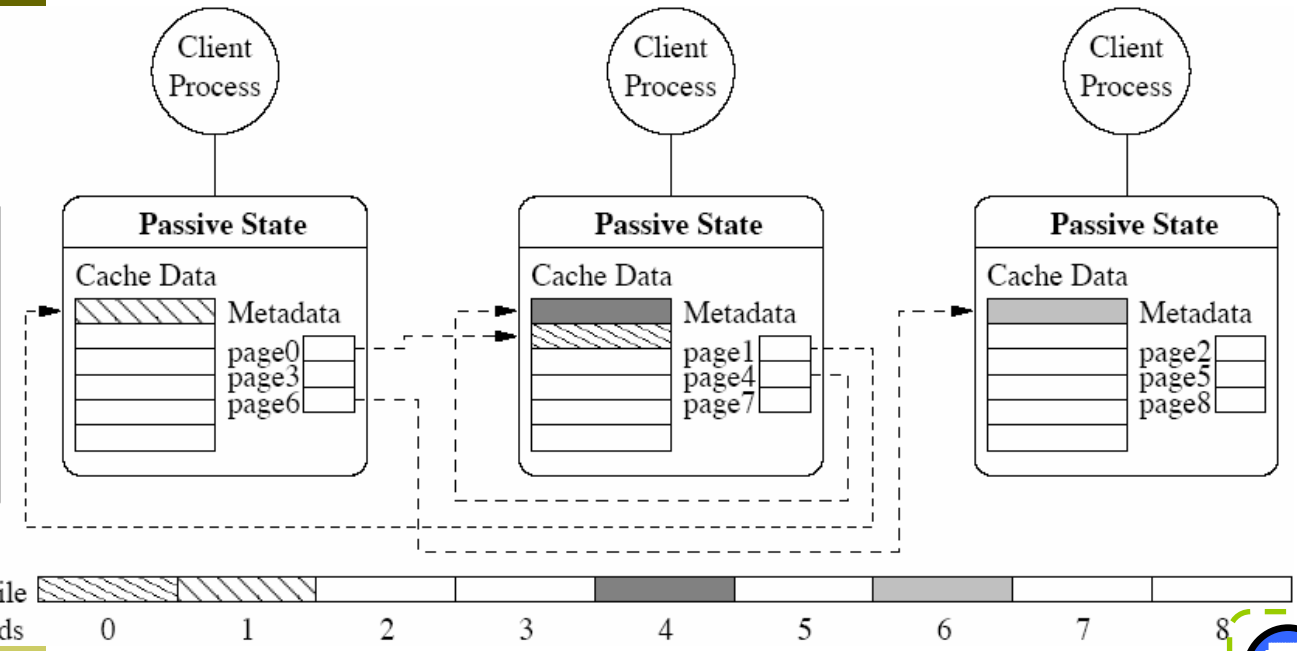
Current **Goal**



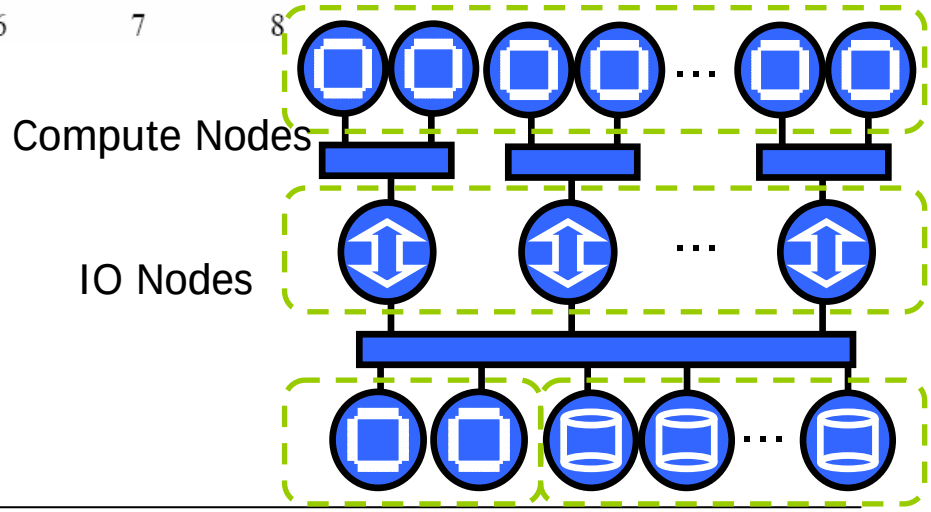
- Is there a way to specify high-level information?

- Proactive
  - Performance
  - Power
  - Portability

Caching Example: Direct Access Caching at Compute Nodes (BG)

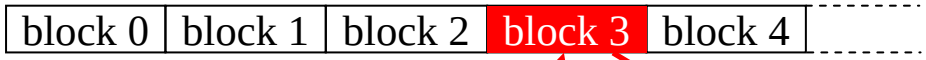


- Exploits larger network
- Exploits larger memory
- I/O nodes' SW does not change
  - (-)Use of compute node resources
  - (-)Dependent on RMA capability



Coherence Control at I/O Nodes

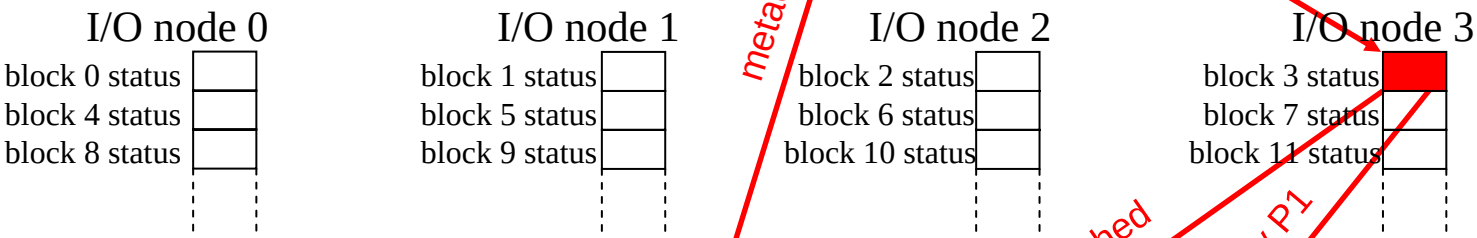
Logical partitioning view of a file



Metadata communication

Cache data communication

Distributed metadata

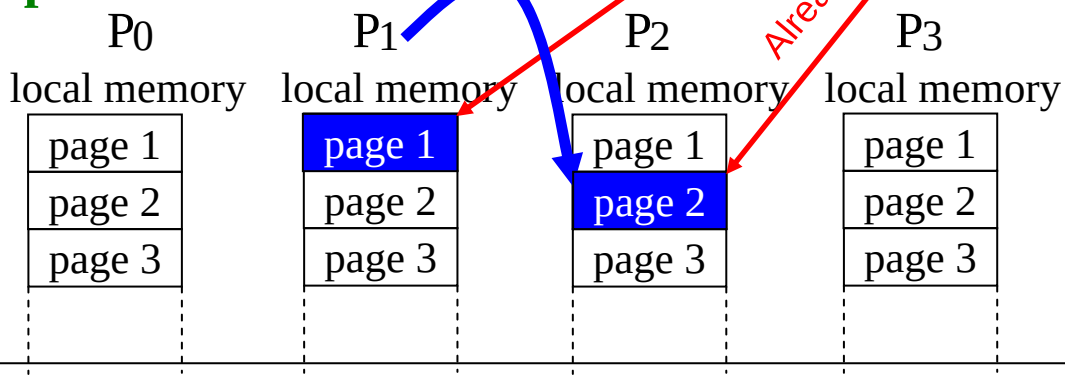


metadata lookup

If not yet cached

Already cached by P1

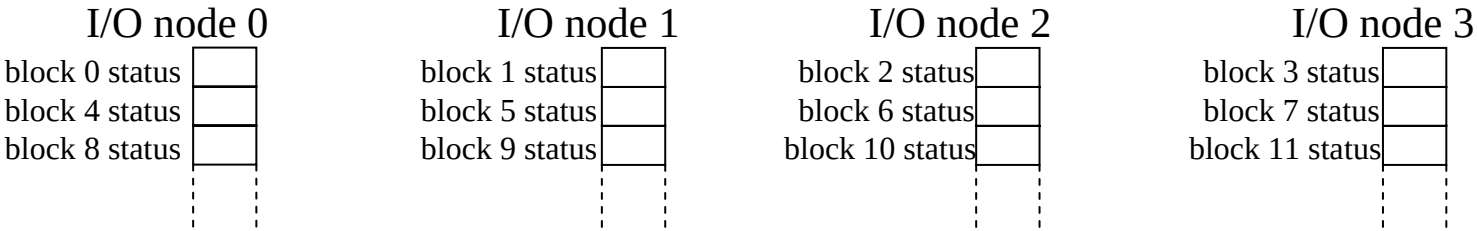
Cache pages at compute nodes



Data at I/O Nodes

Logical partitioning view of a file

Distributed metadata



Cache pages at I/O nodes