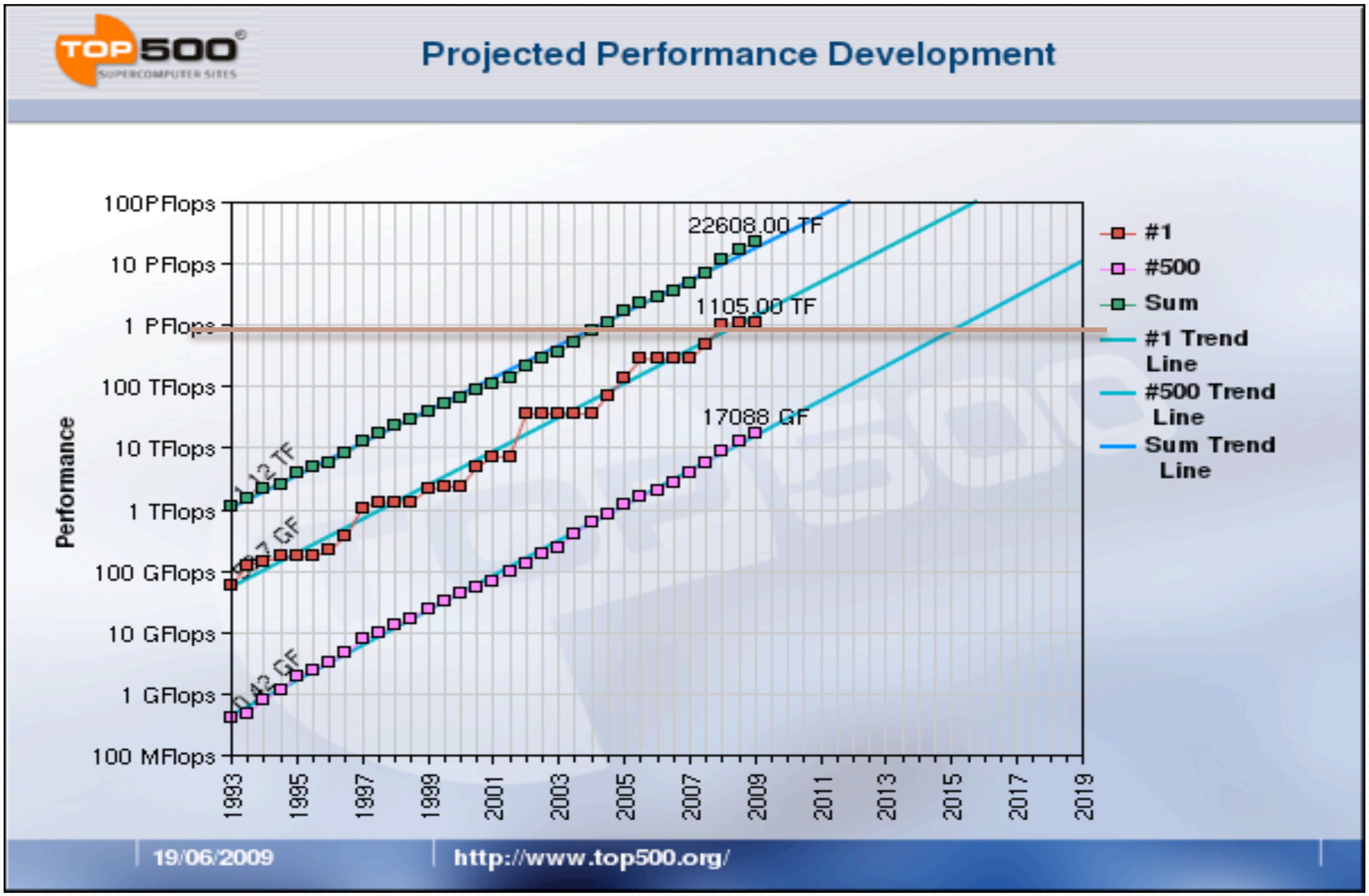
Data Layout Optimization for Petascale File Systems

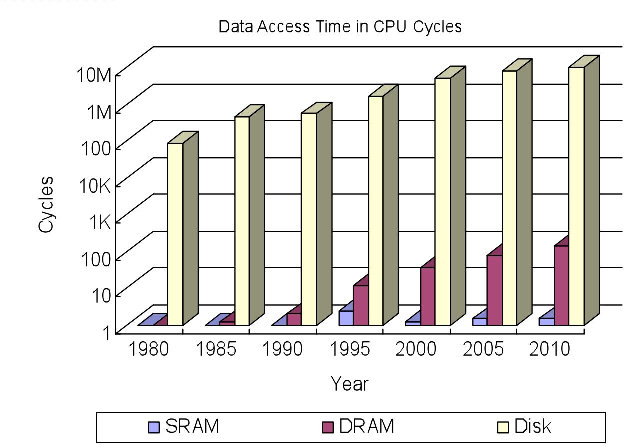
Yong Chen
Xian-He Sun, Yanlong Yin, Huaiming Song, Surendra Byna

Illinois Institute of Technology

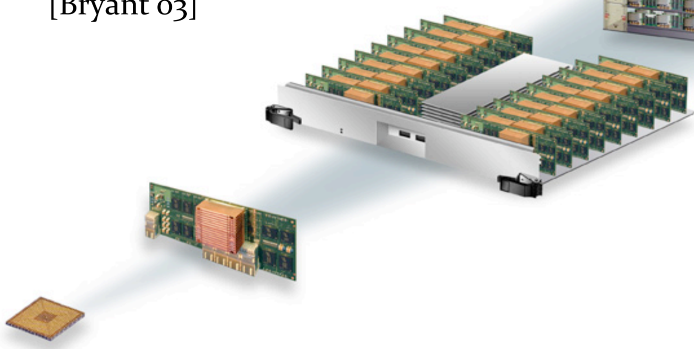
High-Performance Computing System Trend



Problem: I/O Bottleneck



Comparison of data access latency
[Bryant'03]

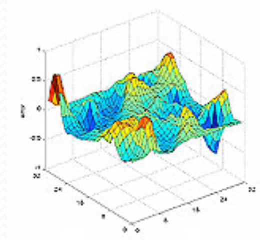


Source: ANL-ALCF
Intrepid Supercomputer

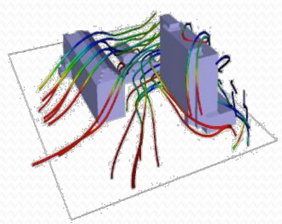
- Significant gapbetween processing capacity and data-access performance
- Long I/O access latency leads to a severe overall performance degradation
- Bottleneck accentuatedby the expanding performance gap

Applications Trend

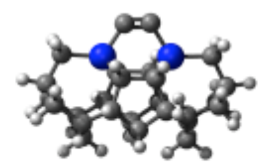
- Applications tend to be data intensive[Reed'03][May'02]
  - Scientificsimulation
  - Data mining,large-scale data processing
  - Visualization applications
  - Geographic information system, etc.
- Apps vary widely in their I/O characteristics[Kotz'98] [Reed'03]
  - Various access patterns
- Applications features should be well considered to deal with I/O bottleneck



Source: Multi-grid solver

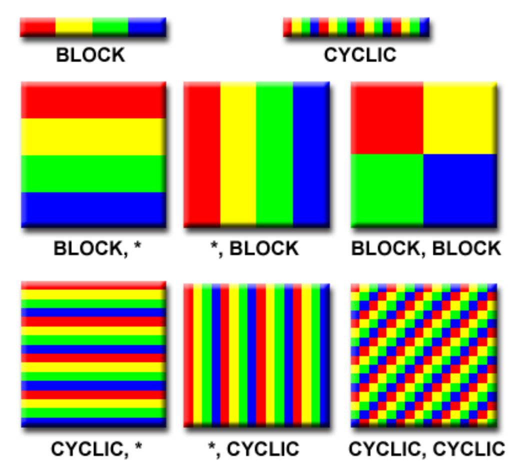


Source: NaSt3DGP



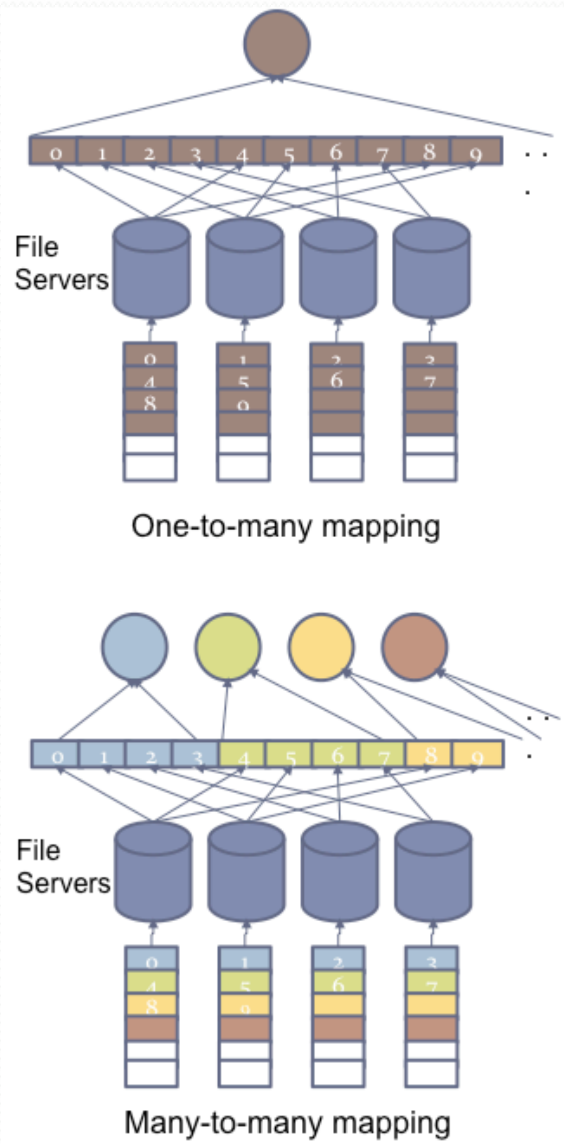
Source: MPQC

Application access patterns



Data Layout and Data Accesses

- Data layout mechanism decides how data are distributed among multiple file servers
- A crucial factorthat decides the data access latency and the I/O subsystem performance for HPC
- Significance and performance improvement demonstrated by arranging data properly in recent studies
  - Log-like reordering
  - Parallel Log-structured File System (PLFS): virtual interposition layer
  - Adaptable IO System (ADIOS)



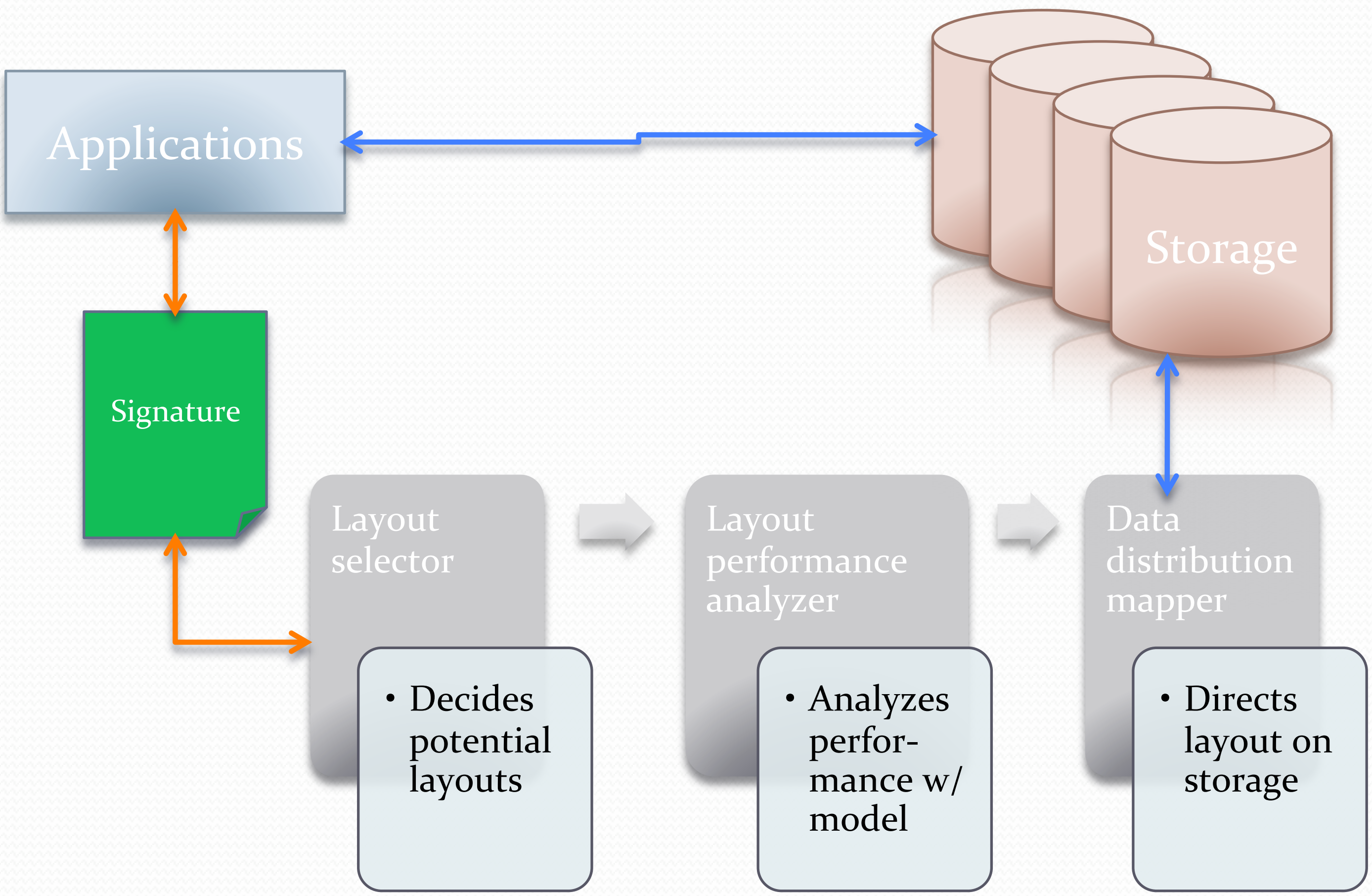
Limitations of Current Parallel I/O System

- Information gapbetween these two sub-systems
  - Parallel file system decides data layout on storage
  - Parallel I/O middleware optimizes, groups and rearranges accesses from applications
- Existing parallel file systems provide high bandwidth for simple, well-formed, and generic I/O access characteristics, but performance varies from application to application
- Tune data layout according to specific I/O access patterns for a parallel I/O system is a necessity
  - A challenging and tedious task for users
  - Manual configuration and hint mechanism are limited
  - Not scalable for petascale systems

Our Solution: Customized Data Layout

- A System-level Application-Specific Data Layout Strategy
  - System-level: integrated into the file system and transparent to programmers and users
  - Application-specific: adapt to specific data access patterns for a proper data layout
- Contributions
  - Demonstrate the data layout strategy has aclear impact on data storage performance
  - Present a framework to pass some of the application-specific I/O request information to file systems and to foster a better integration of parallel I/O and parallel file systems
  - Illustrate with a simple performance model that layout strategy can be modeled and application-specific optimization could be beneficial

High-Level View



I/O Signature: Pattern Classification

Spatial Patterns

- ❑Contiguous
- ❑Non-contiguous
  - ▪Fixed strided
  - ▪2d-strided
  - ▪Negative strided
  - ▪Random strided
  - ▪*k*d-strided
- ❑Combination of contiguous and non-contiguous patterns

Request size

- ❑ Fixed
- ❑ Variable

- ❑ Small
- ❑ Medium
- ❑ Large

Temporal Intervals

- ❑Fixed
- ❑Random

Repetition

- ❑Single occurrence
- ❑Repeating

I/O Operation

- ❑Read only
- ❑Write only
- ❑Read/write

I/O Signature Notation

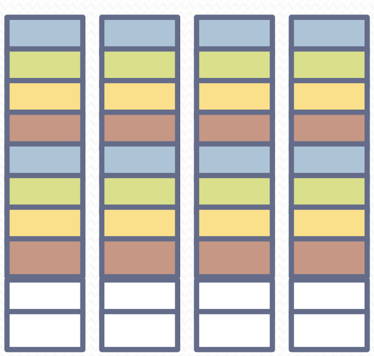
- A combination of two notations
  - Trace Signature
  - Pattern Signature
- Trace signature
  - Description of a sequence of I/O accesses in a pattern
  - Form: *{I/O operation, init position, dimension, ([{offset pattern}, {request size pattern}, {pattern of number of repetitions}], […]), # of repetitions}*
  - Provides a way to reconstruct the sequence of I/O accesses
- Pattern signature
  - Provides a simple description that explains the nature of a pattern
  - An abstraction of a trace pattern
  - Stores information consisting of all five factors of our classification
  - Form: {I/O operation, <Spatial pattern, Dimension>, <Repetitive behavior>, <Request size>, <Temporal Intervals>}

Application-Specific Data Layout Modeling

- Assumptions for a simple model
  - # of computing nodes: *p;* # of I/O servers: *n*
  - I/O server performance model: $\alpha+s\beta$
    - α : latency, e.g. seek time, rotation time; β: transmission time
    - s: size of a contiguous request
- 1-dhorizontal (1-DH) model / simple round-robin model
  - One client process accessing data takes time: $\alpha + \frac{s}{n}\beta$
  - If *p* processes accessing data simultaneously, take time: $p\alpha + \frac{ps}{n}\beta$
- 1-d vertical(1-DV) model
  - Data to be accessed by each process stored on one given server
  - If *p* processes accessing data simultaneously, take time: $\left\lceil \frac{p}{n} \right\rceil (\alpha + s\beta)$
- 2-d layout (2-D) model
  - If *p* processes accessing data simultaneously, take time: $\alpha + \frac{s}{\left\lfloor \frac{n}{p} \right\rfloor}\beta$

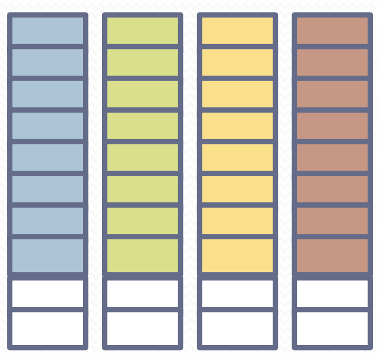
Data Layout Matters

$$p(\alpha + \frac{s}{n}\beta) = p\alpha + \frac{ps}{n}\beta$$



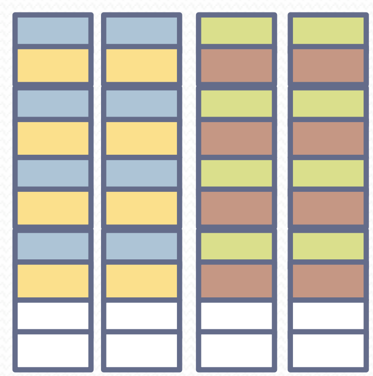
1-d horizontal

$$\alpha + s\beta \quad \text{or} \quad \left\lceil \frac{p}{n} \right\rceil (\alpha + s\beta)$$



1-d vertical

$$\alpha + \frac{s}{\left\lfloor \frac{n}{p} \right\rfloor}\beta$$



2-d striping

Application-Specific Data Layout Optimization

- Implications of simple layout model
  - If $p >= n$, the 1-d vertical striping data layout is better
  - If $p < n$, then data can be stored either on n servers using 1-d horizontal striping data layout or use 2-d striping data layout, where each process gets n/p file servers for data storage

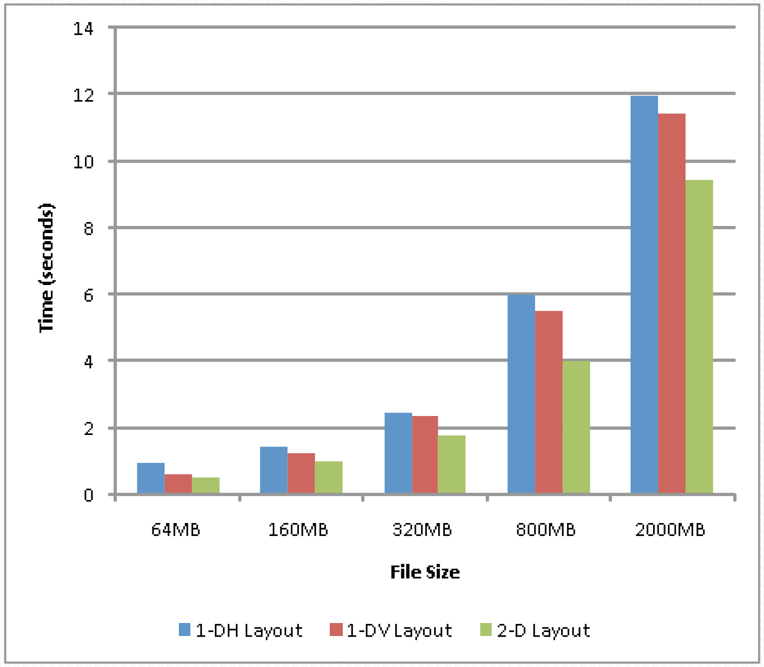
Heuristics for Choosing Layouts

Global Access Pattern Feature	Layout
Random	Default Round-robin
High degree of I/O concurrency	1-d vertical striping
Low degree of I/O concurrency	Simple striping or 2-d striping
Too many I/O servers on TCP/IP	2-d striping

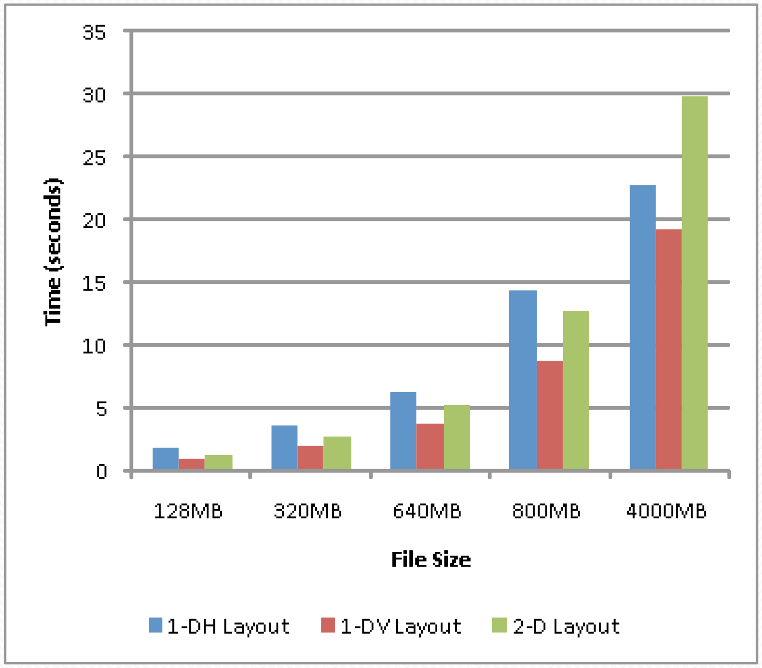
Experimental Setup

- Experimental environment
  - 17-node Dell PowerEdge Linux-based cluster
    - 2x73GB U320 10K-RPM SCSI drive on head node
    - 40 GB 7.2K-RPM SATA drive on each compute node
    - PVFS2, 1 metadata server node, 8 I/O server nodes
  - 65-node Sun Fire Linux-based cluster
    - 12x500GB 7.2K-RPM SATA-II drives configured as a RAID-5
    - 250GB 7.2K-RPM SATA hard drive
    - PVFS2, 32 metadata/IO server nodes, 32 client nodes
- Benchmarks
  - Synthetic benchmark
  - IOR benchmark

A Simple Evaluation – Synthetic Benchmark



4 processes



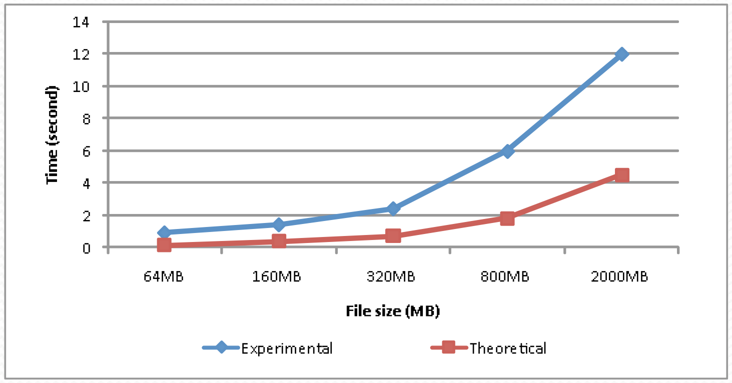
16 processes

- 2-D layout achieved the best perf.
- 1-DH < 1-DV/2-D, variation up to 48.8%

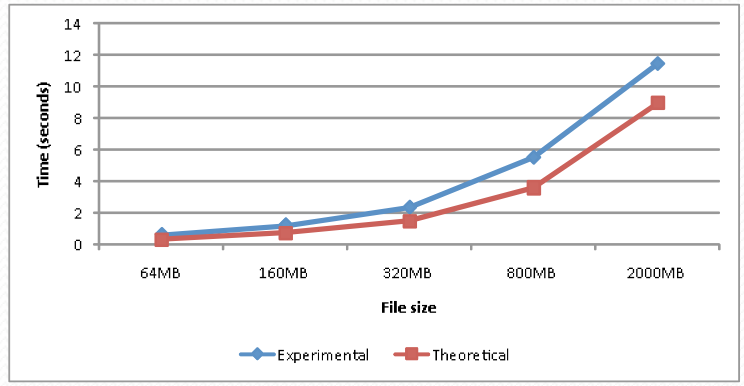
- 1-DV layout achieved the best perf.
- Variation was up to 55.3%

- Different layout strategies clearly have impact on performance

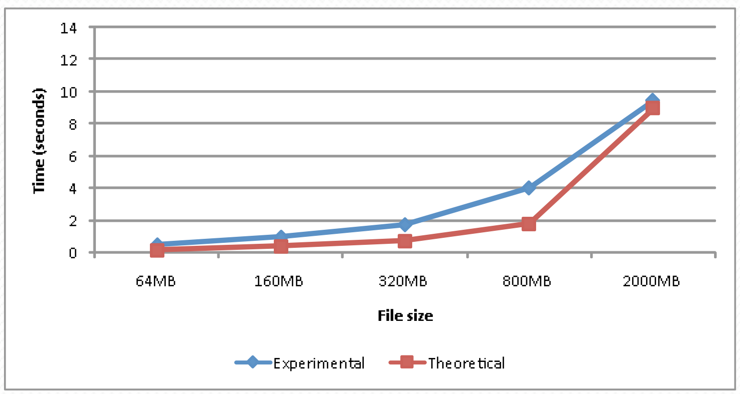
Experimental and Theoretical Results



1-DH Layout



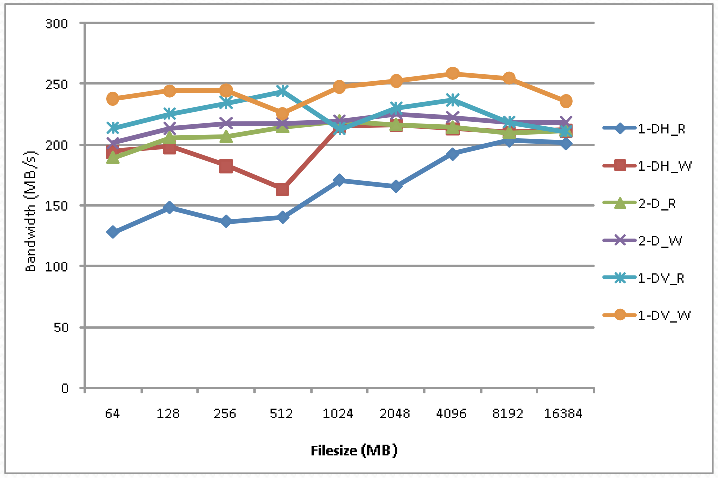
1-DV Layout



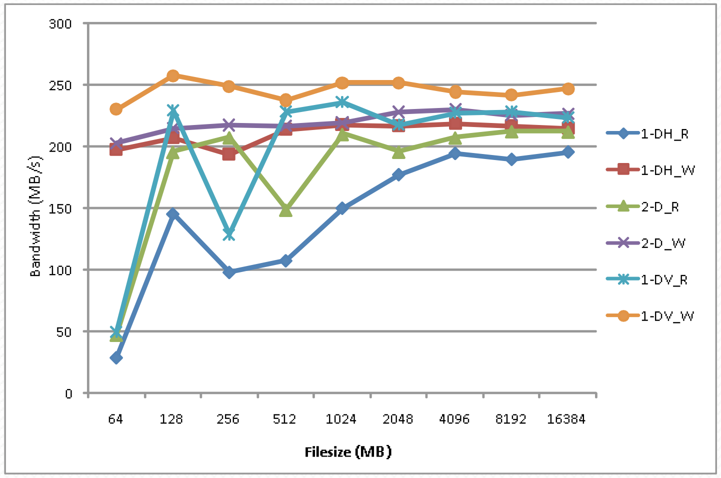
2-D Layout

- Compute the theoretical value
- A fairly close match

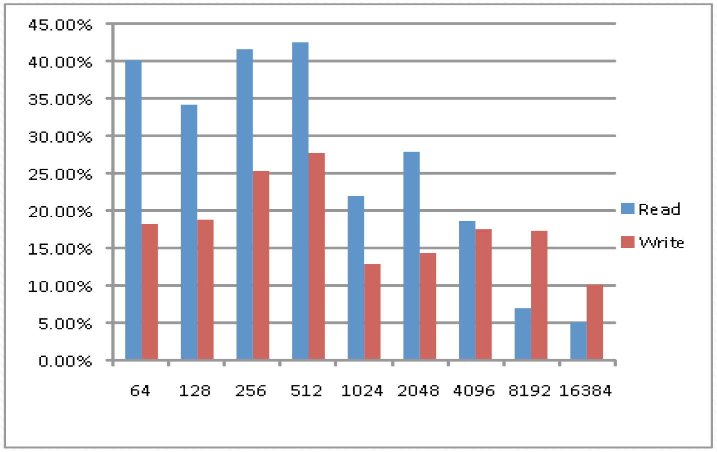
IOR Benchmark

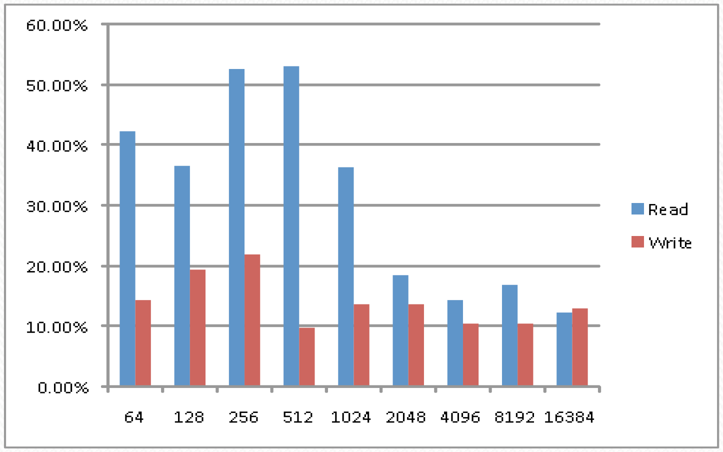


64p, random, 64KB strip, X4KB

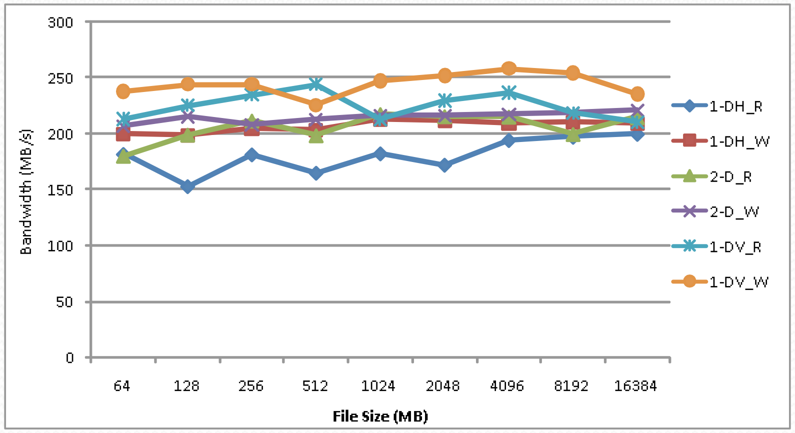


64p, sequential, 64KB strip, X4KB

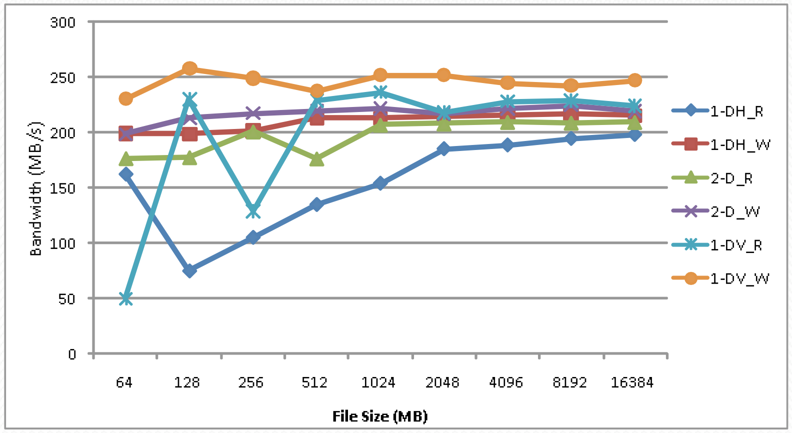




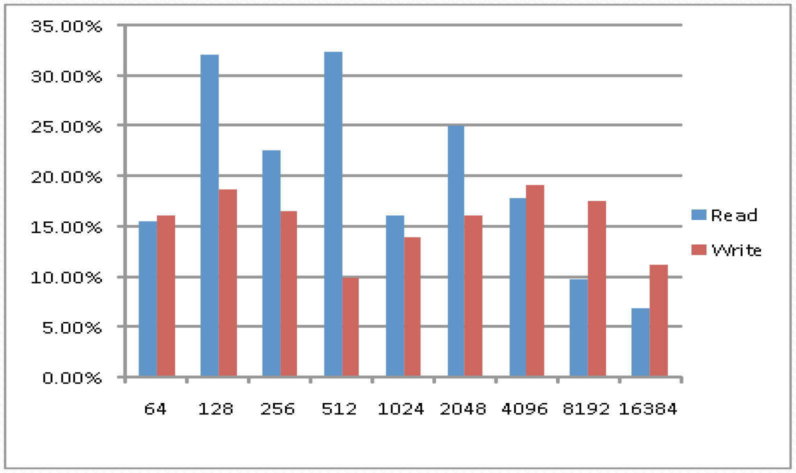
IOR Benchmark

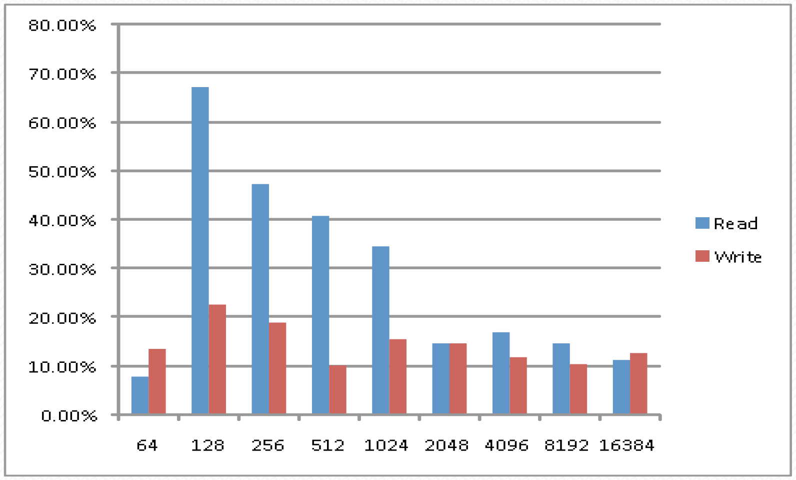


64p, random, 1MB strip, X4KB

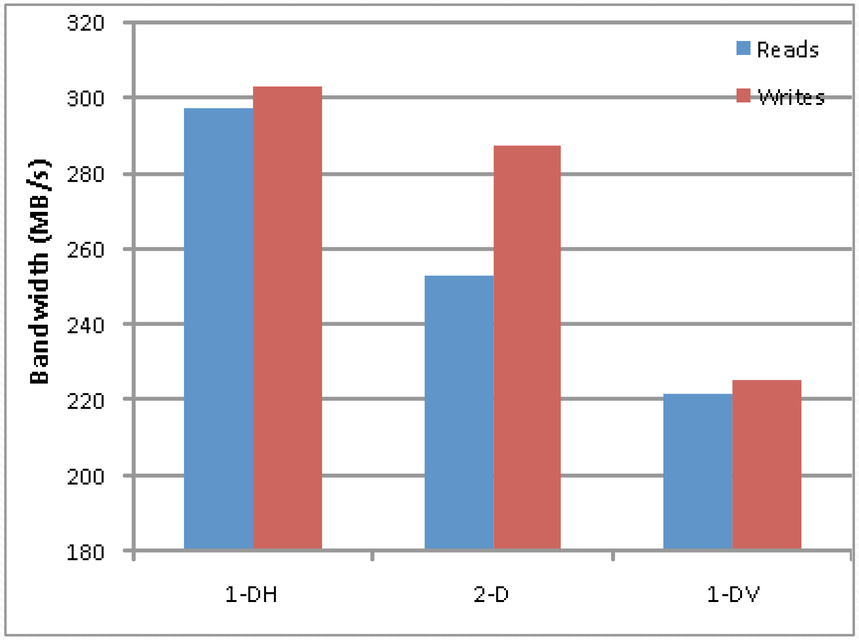


64p, sequential, 1MB strip, X4KB

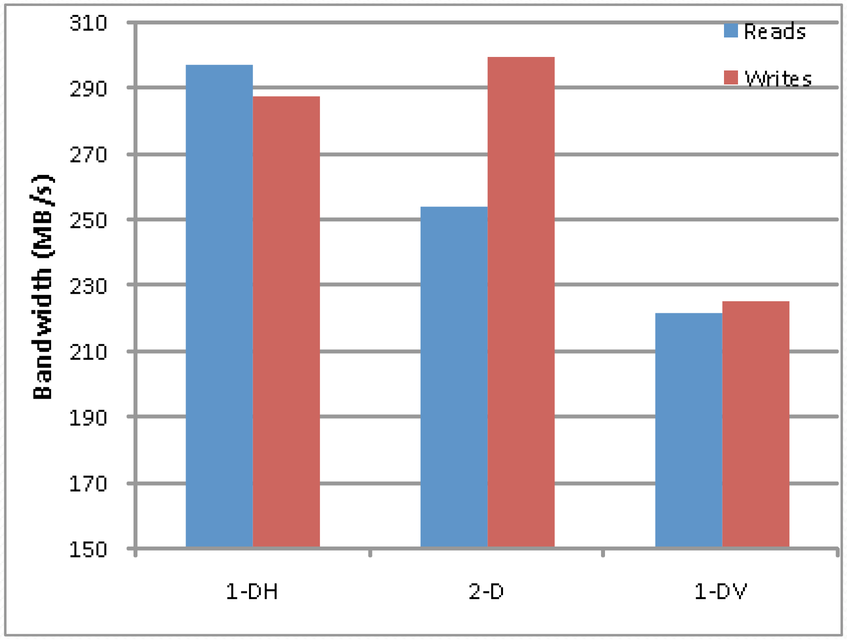




IOR Benchmark



8p, random, 4KB strip, X1MB



8p, random, 64KB strip, X1MB

- 1-DH generally better
- Data layout strategies have a clear impact

Conclusion

- Parallel I/O middleware and parallel file systems are fundamental and critical components for petascale storages
- Little has been done to application-specific data layout
  - Simple round-robin strategy does not always work well
- We propose a System-level Application-specificData Layoutstrategy
  - Optimize accesses according to distinct application features
  - Integrate into file system and benefit users transparently
- Preliminary results have demonstrated the potential
- More research needed for next-generation I/O architectures to support access awareness, intelligence, and application-specific adaptive data distribution and redistribution

Thank You!

- Acknowledgement
  - National Science Foundation
  - Dr. Rajeev Thakur, Dr. Rob Ross and Sam Lang of Argonne National Laboratory
  - Anonymous reviewers

- Welcome to visit http://www.cs.iit.edu/~scs