LA-UR-12-23164

Title: Hard Disk/Solid State Drive Synergy in Support of Data-Intensive Computing

Author(s):
Liu,Ke
Jiang, Song
Davis, Kei

Intended for: LANL student symposium 2012

Disclaimer:
Los Alamos National Laboratory, an affirmative action/equal opportunity employer,is operated by the Los Alamos National Security, LLC for the National NuclearSecurity Administration of the U.S. Department of Energy under contract DE-AC52-06NA25396. By approving this article, the publisher recognizes that the U.S. Government retains nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or to allow others to do so, for U.S. Government purposes. Los Alamos National Laboratory requests that the publisher identify this article as work performed under the auspices of the U.S. Departmentof Energy. Los Alamos National Laboratory strongly supports academic freedom and a researcher's right to publish; as an institution, however, the Laboratory does not endorse the viewpoint of a publication or guarantee its technical correctness.

Abstract

Data-intensive applications are becoming increasingly common in high-performance computing. Examples include combustion simulation, human genome analysis, and satellite image processing. Efficient access of data sets is critical to the performance of these applications. Because of the size of the data today's economically feasible approach is to store the data files on an array of hard disks or data servers equipped with hard disks and managed by a parallel file system such as PVFS or Lustre wherein the data is striped over a (large) number of disks for high aggregate I/O throughout.

With file striping, a request for a segment of logically contiguous file space is decomposed into multiple sub-requests, each to a different server. While the data unit for this striping is usually reasonably large to benefit disk efficiency, the first and/or last sub-requests can be much smaller than the striping unit if the request does not align with the striping pattern, severely compromising hard disk efficiency and thus application performance.

We propose to exploit solid state drives (SSD), whose efficiency is much less sensitive to small random accesses, to enable the alignment of requests to disk with the data striping pattern. In this scheme hard disks mainly serve large, aligned, sequential requests, with SSDs serving small or unaligned requests, thus respecting the relative cost, performance, and durability characteristics of the two media, and thereby achieving synergy in performance/cost. We will describe the design of the proposed scheme, its implementation on CCS-7's Darwin cluster, and performance results.

Hard Disk/Solid State Drive Synergy in Support of Data-intensive Computing

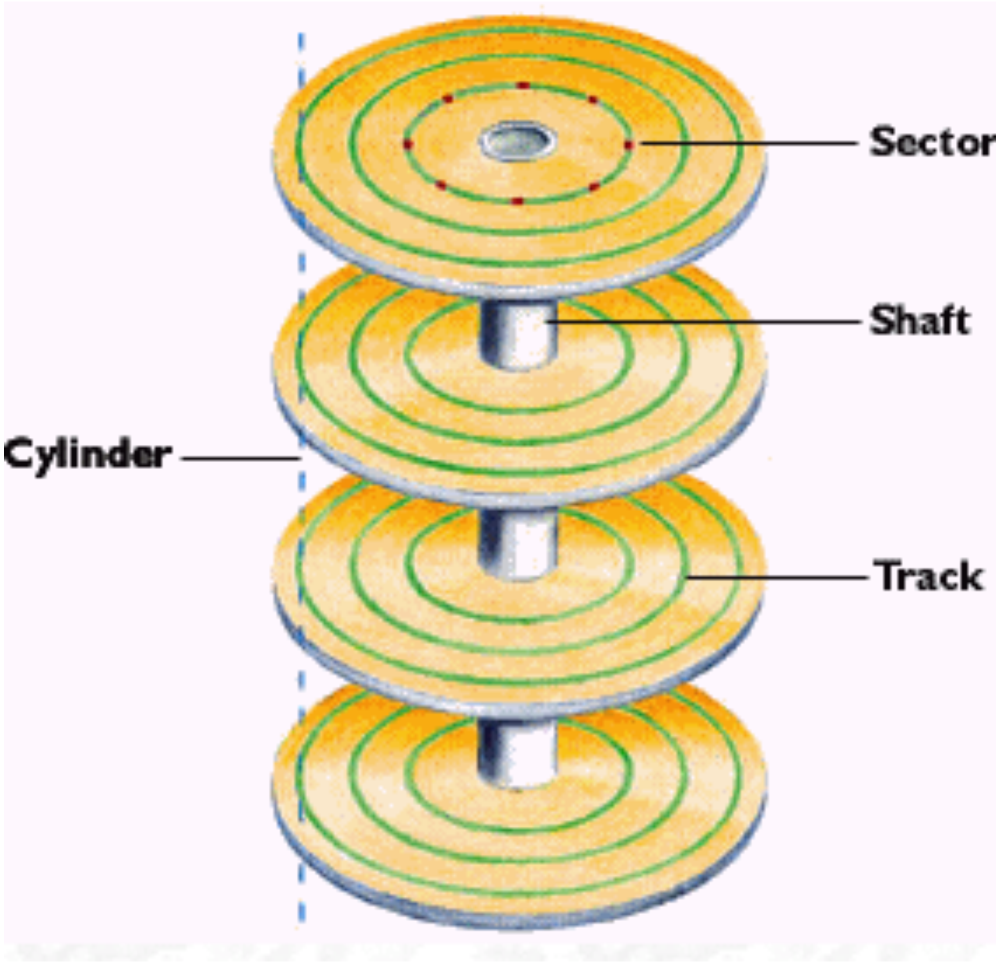
Ke Liu
Song Jiang
Wayne State University

Kei Davis
CCS-7
LANL

Outline

- Introduction
- Design and Implementation
- Performance Evaluation
- Conclusions

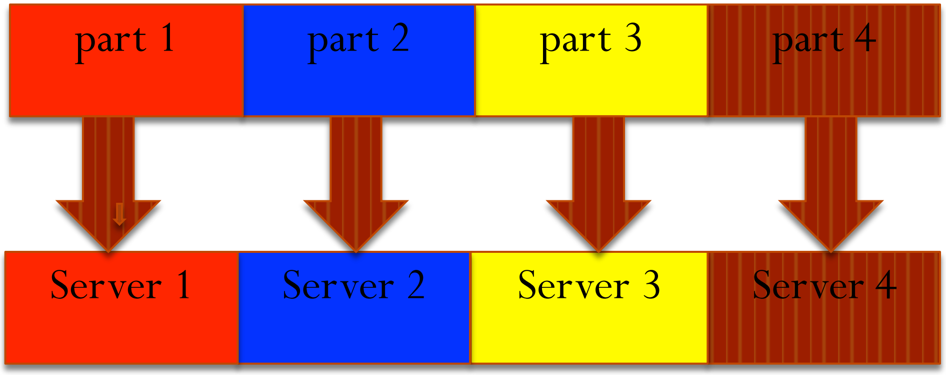
How hard disk works



Data-intensive applications are common in HPC

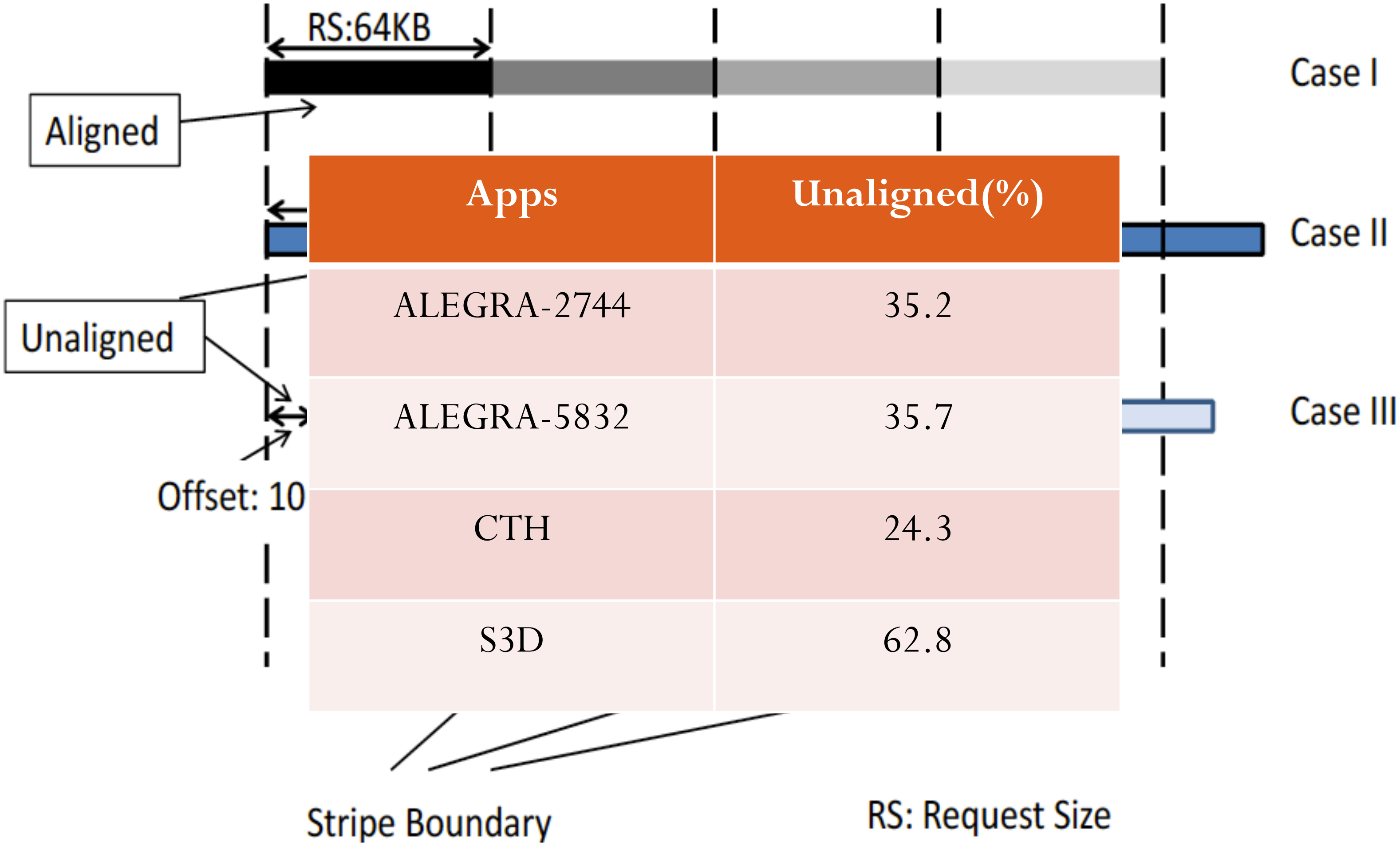
- Data-intensive applications are common in high-performance computing environments
  - Combustion simulation, complete human genome, satellite images
- How to store these huge data sets?
  - PVFS
  - GPFS
  - Lustre

How data is stored on a parallel file system



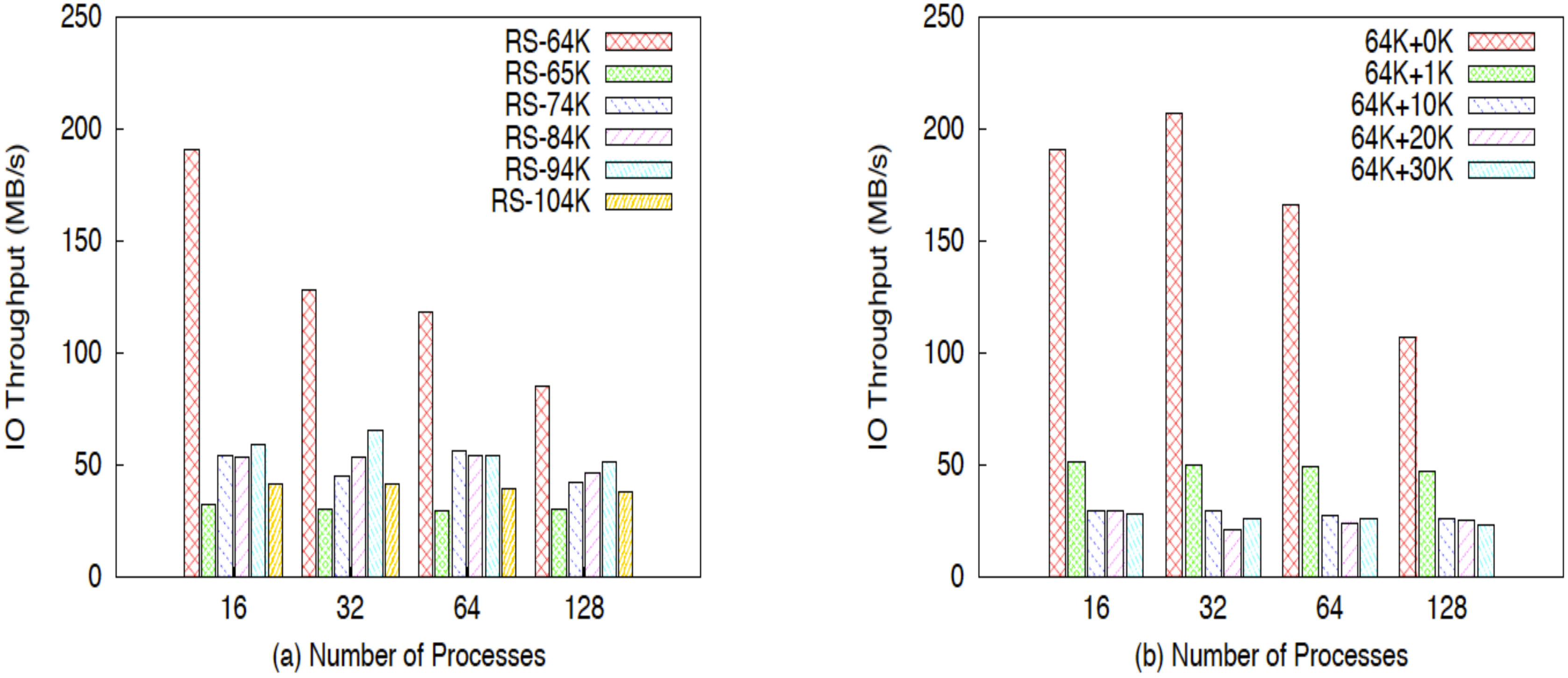
- Striping unit size is preset (64KB by default in PVFS2)

Different access pattern on parallel file system



Apps	Unaligned(%)
ALEGRA-2744	35.2
ALEGRA-5832	35.7
CTH	24.3
S3D	62.8

Unaligned accesses degrade performance



Characteristics of solid state drive

How to use SSD to handle unaligned access

- Use SSD as a buffer to store all data
  - Too much data to write to a limited size SSD
  - SSD would be worn out quickly in HPC
  - Cost-ineffective
- Why keep the existing hard disk idle?
- Use SSD only for small random and unaligned access

Outline

- Introduction
- **Design and Implementation**
- Performance Evaluation
- Conclusions

Ibridge: Improving unaligned parallel file access with SSD

- Objective:
  - Use SSDs to serve fragments produced by unaligned data accesses
- Challenges
  - Distinguishing ordinary random requests and fragments
  - Establishing appropriate criteria for either a regular random request or a fragment to be admitted to the SSD.

Architecture

- Client-side
  - Identifying fragments, passing the information to the data servers
- Server-side
  - Composed of one hard disk and one SSD
  - Check if a subrequest is a fragment or not
  - SSD is treated as local cache for hard disk

Management of SSDs for Fast and Balanced Access of Disks

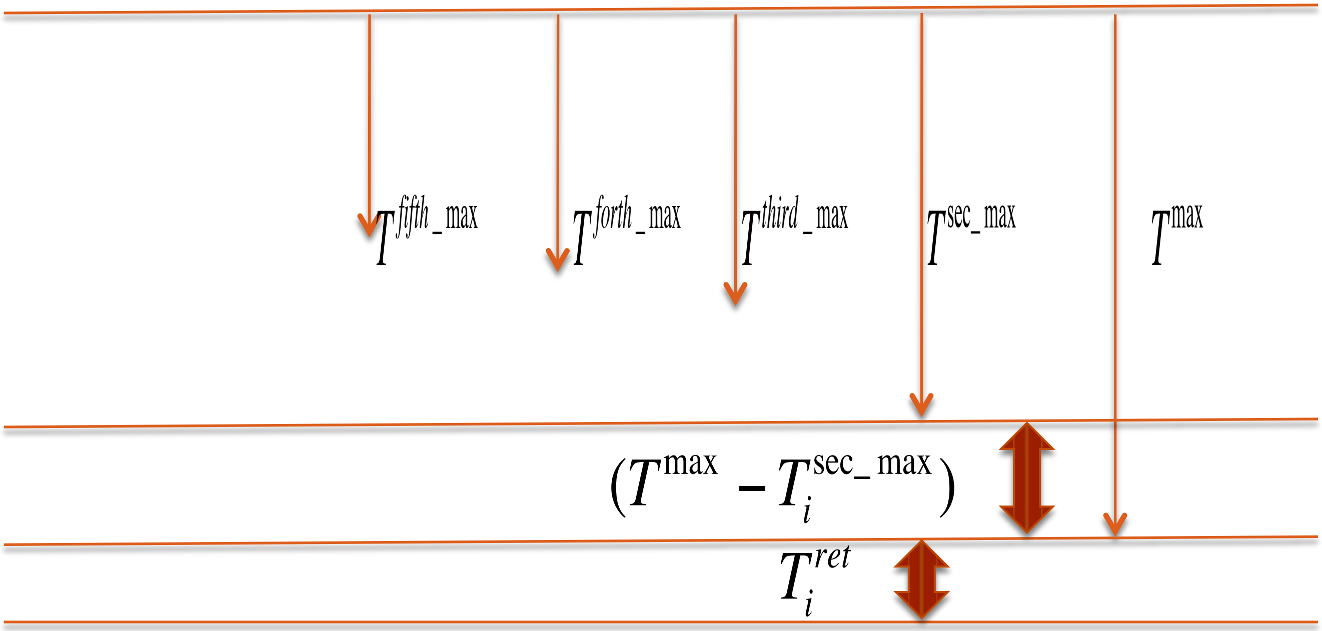
- Average request service time T$_i$ for the i-th request arriving at and served by a disk:

$$T_i = \frac{T_{i-1}}{8} + \frac{(D\_to\_T(\Delta_i - \Delta_{i-1}) + R + Size_i / B) * 7}{8}$$

- The i-th request is served at the disk T_i^{disk}
- The i-th request is served at the SSD T_i^{ssd}
- $T_i^{ret} = T_i^{disk} - T_i^{ssd}$

Management of SSDs for Fast and Balanced Access of Disks (Cont.)

- The T value of the fragment who holds the largest T value denotes as $T^{\max}$
- The T value of the subrequest who holds the second largest T value denotes as $T^{\sec\_\max}$
- Benefits: $T_i^{ret\_frag} = T_i^{ret} + (T^{\max} - T_i^{\sec\_\max}) * n$



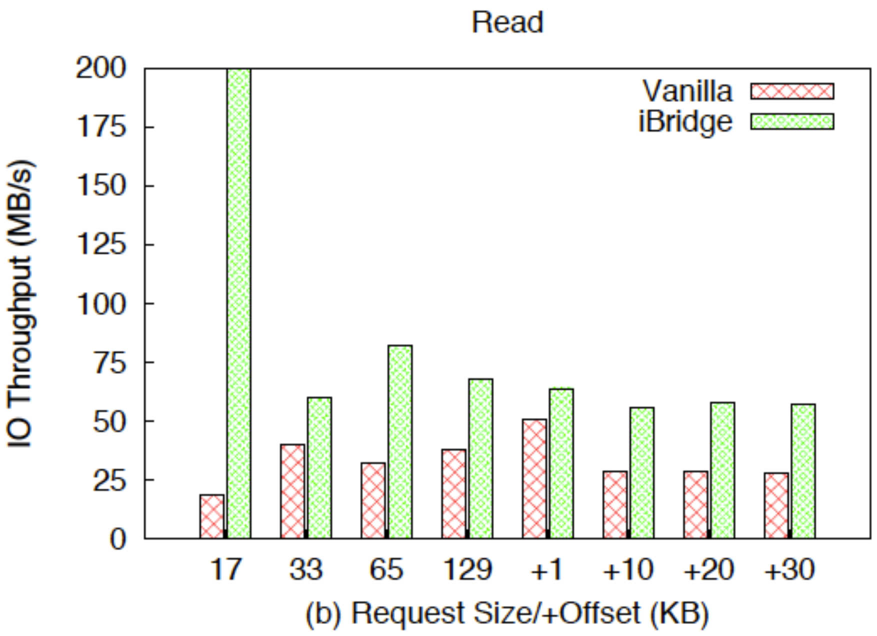
Outline

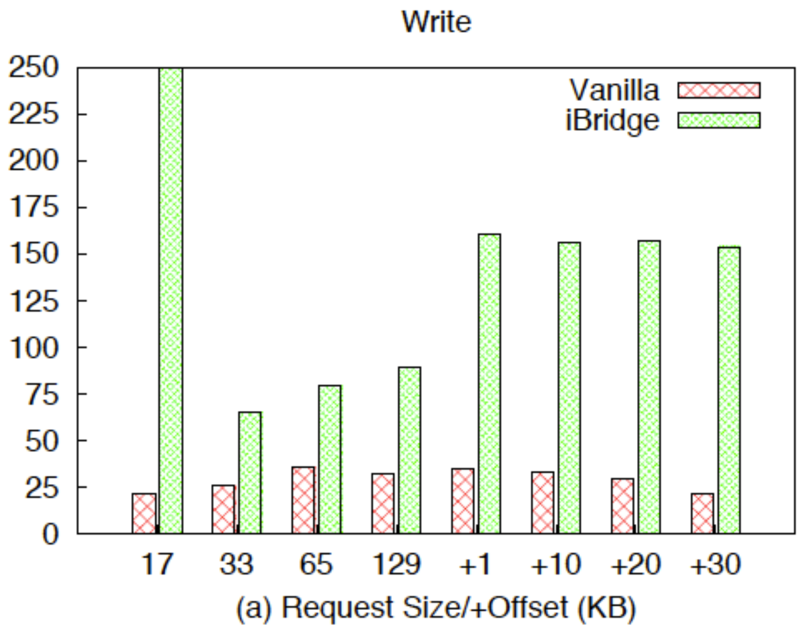
- Introduction
- Design and Implementation
- **Performance Evaluation**
- Conclusions

Experimental setting

- Darwin cluster at Los Alamos National Laboratory
  - 120 nodes
  - 48-core 2GHz AMD, 64GB memory
  - 120GB SSD, RAID0 consisting of two disks
- Software setting.
  - Linux kernel 2.6.35.10
  - PVFS2 parallel file system
  - MPICH2-1.4 compiled with ROMIO
  - CFQ for RAID, NOOP for SSD

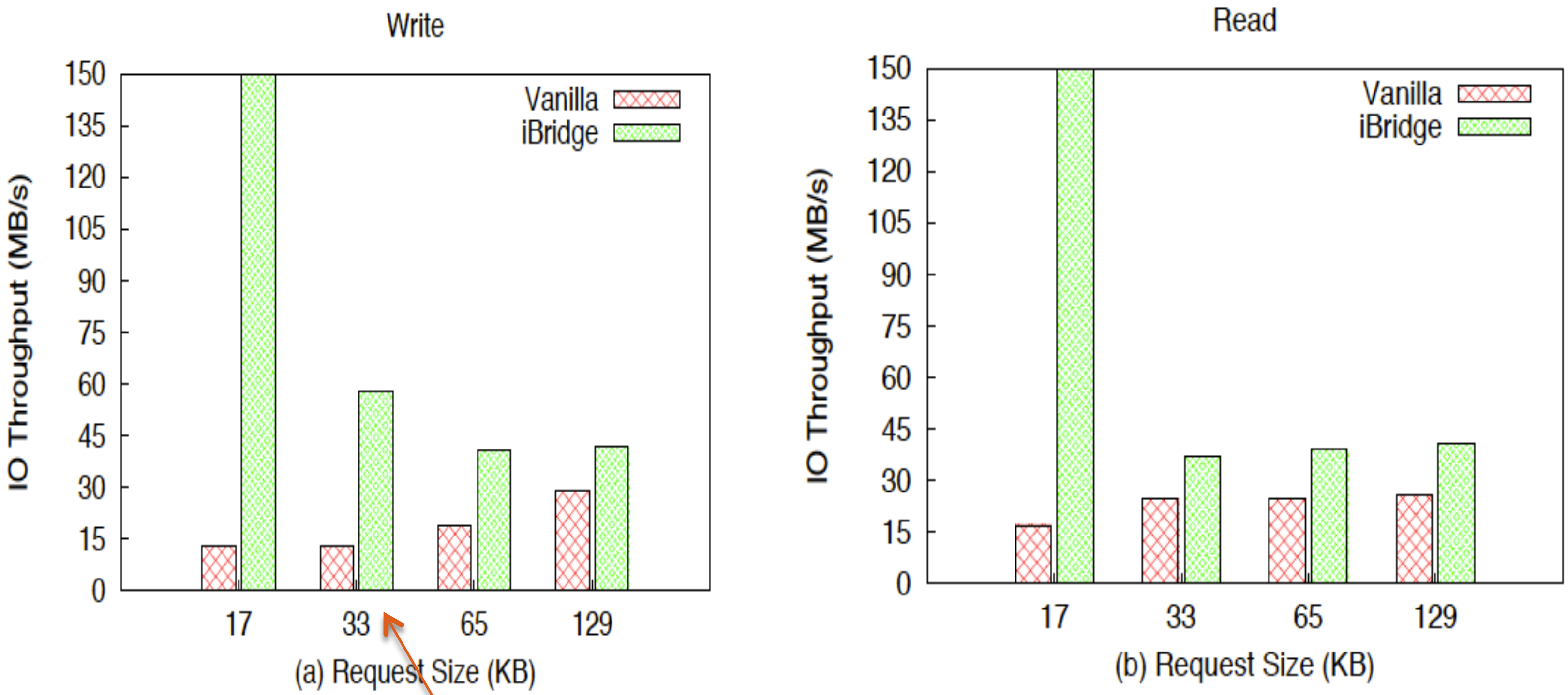
mpi-io-test benchmark





IBridge achieves up to 1100% improvement

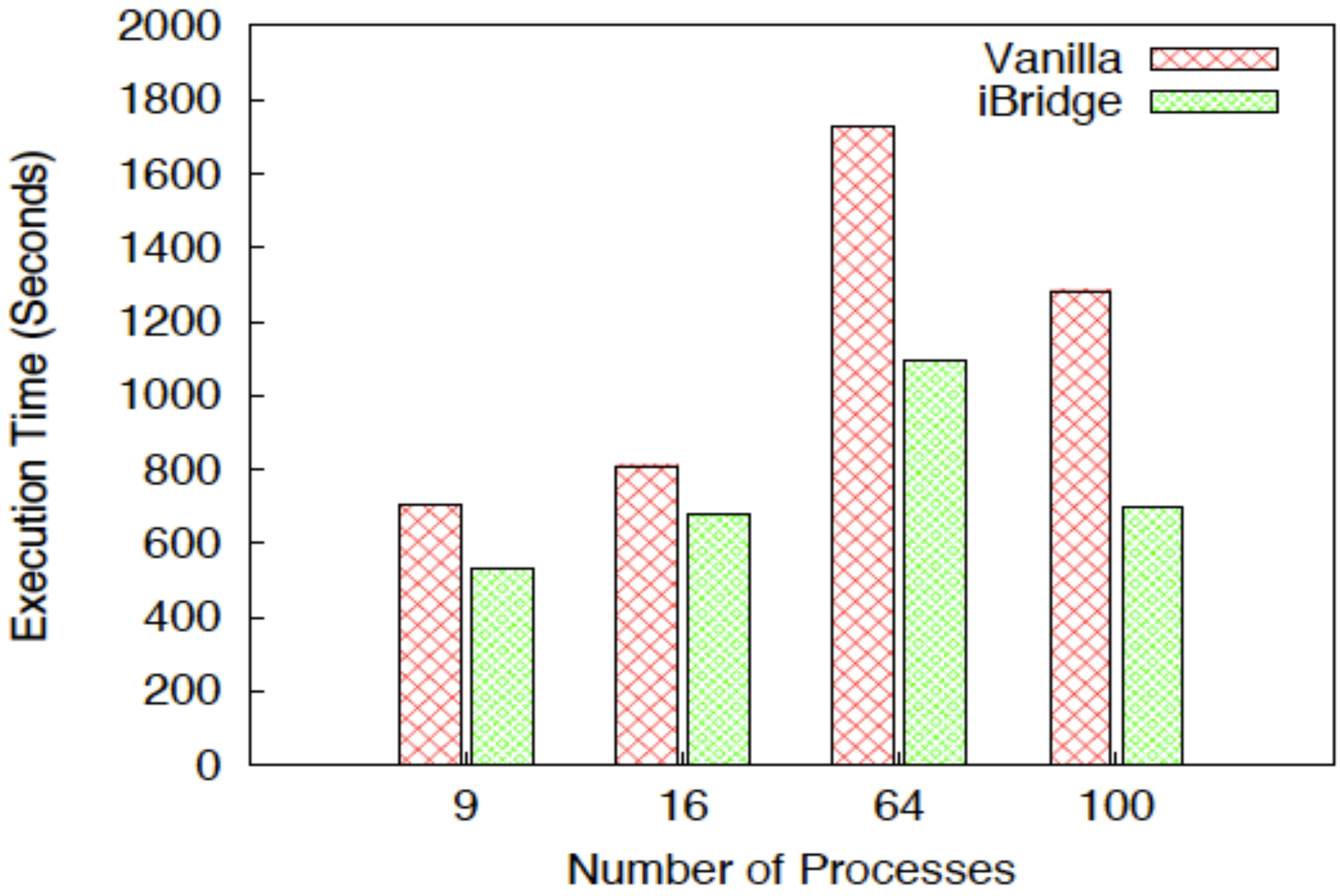
ior-mpi-io benchmark



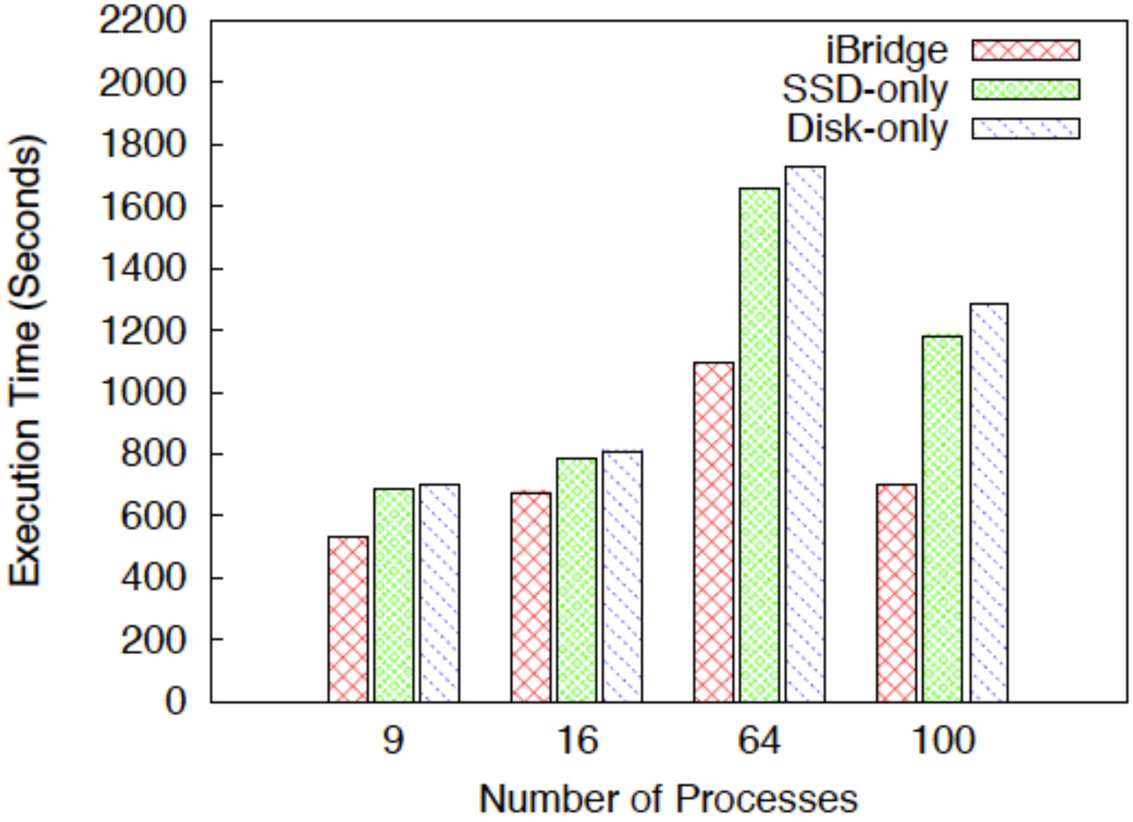
We have up to 4.5 times improvement when data is stored on both SSD and hard disk

BTIO benchmark

- Program's execution times are reduced by up to 45%



BTIO benchmark (cont.)



IBridge is more efficient than using SSD alone let alone using disk alone

Outline

- Introduction
- Design and Implementation
- Performance Evaluation
- Conclusions

Conclusions

- We designed and implemented a hybrid storage scheme for parallel file systems to handle the fragmentation resulting from unaligned parallel data access, and evaluated it on a large HPC cluster.
- We achieved a synergistic coupling of SSD and hard disk, exceeding the performance of SSD or hard disk alone, by exploiting the relative strengths, and respecting the relative weaknesses, of both.