Performance of The Galley Parallel File System

Nils Nieuwejaar
Dartmouth College

Joint work with David Kotz

Overview

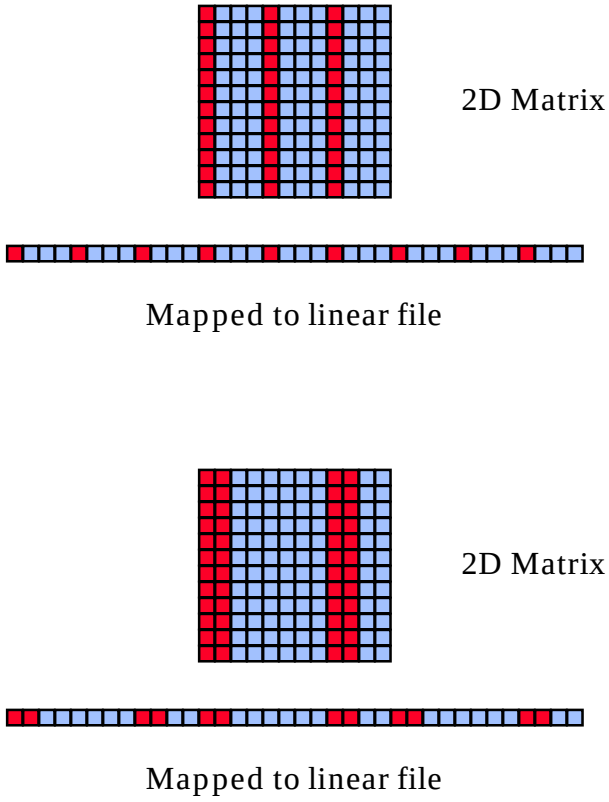
- Background
- Galley structure and interface
- Access patterns
- Performance
- Conclusion

Workload Characterization

- Most requests were small
  - Most requests were less than 300 bytes
- Requests were frequently non-contiguous
- Access patterns within an application were very regular
  - Repeated request sizes
  - Repeated interval between requests

Strided Access Patterns

- A 2D matrix is stored on disk in row-major order, and the columns of the matrix are distributed across the 4 nodes of an application.

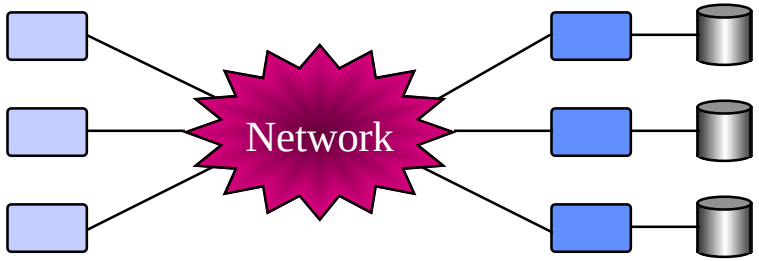


Design Goals

- Provide high performance for common access patterns
- Allow applications to explicitly control parallelism
- Allow easy implementation of high-level libraries
- Scale

System Structure

- Compute Processors
  - User applications
  - Galley run-time library
- I/O Processors
  - Control disks
  - Run Galley's system code

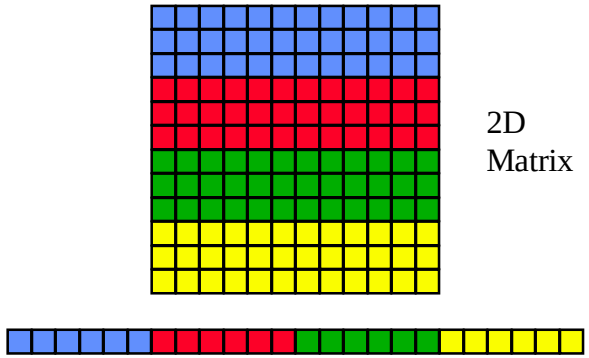


Data Transfer Operations

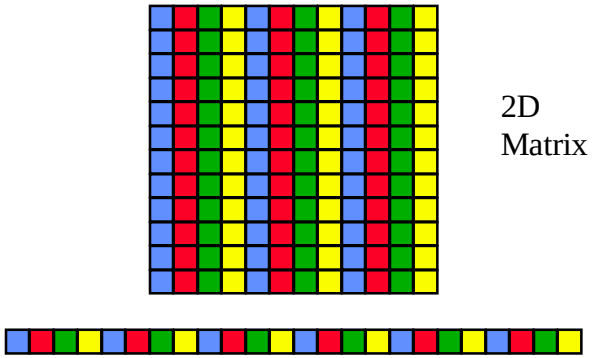
- Traditional interface
  - `gfs_read()`, `gfs_write()`
- Batched interfaces
  - Strided: `gfs_read_strided()`
  - Nested-strided: `gfs_read_nested()`
  - Nested-batched: `gfs_read_batched()`
  - List I/O: `gfs_read_listio()`
- Blocking and non-blocking I/O

Access patterns

- Broadcast
  - All nodes read the whole file
- Partitioned
  - Each node reads an independent, contiguous region of the file
- Interleaved
  - Each node reads a non-contiguous, but regularly spaced, series of records from the file



Mapped onto linear file

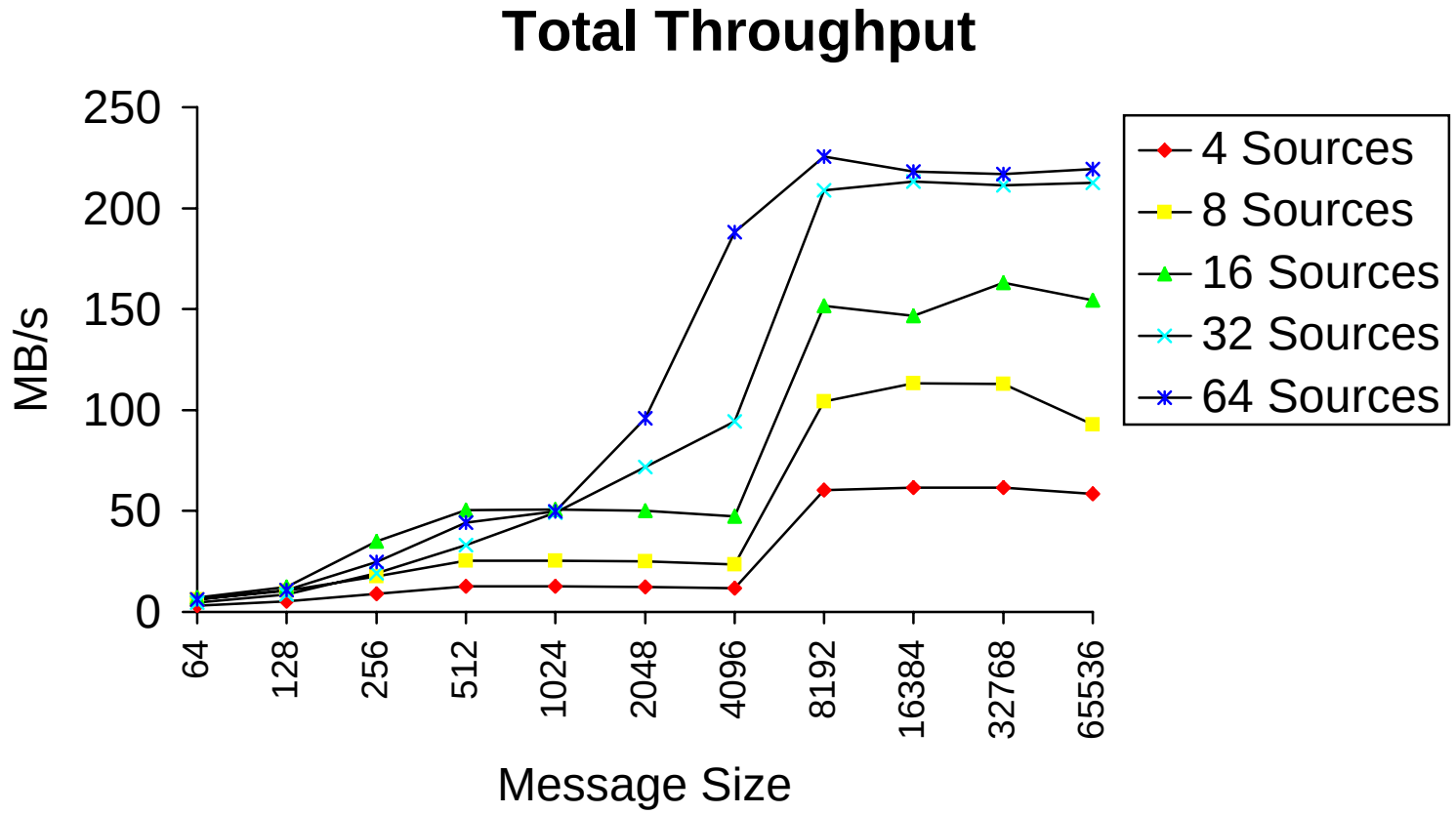


Mapped onto linear file

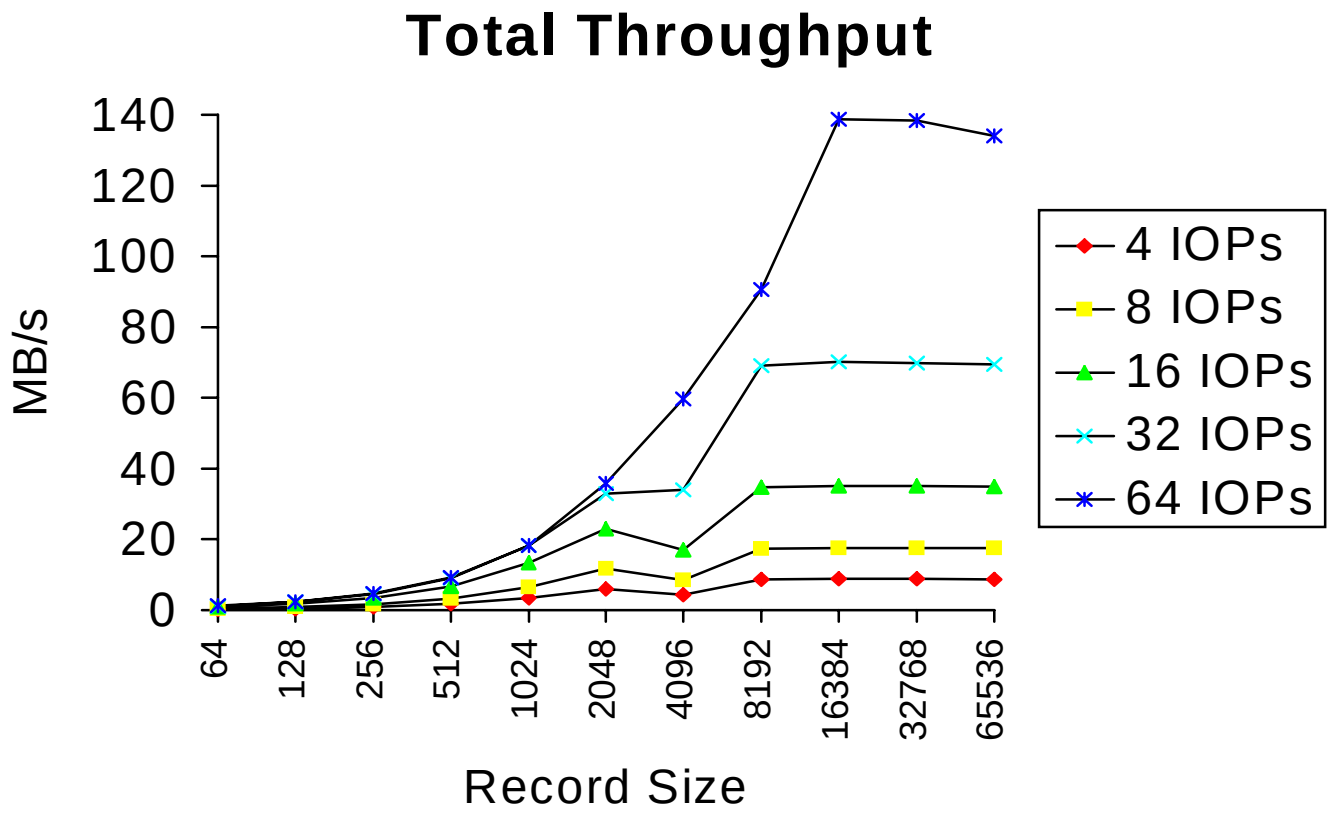
Platform

- IBM SP-2 at NASA Ames
- 160 nodes (140 available)
  - RS/6000s running AIX 4.1.3
  - at least 128MB on each node
  - One simulated disk on each IOP
    - 2.2 MB/s sustainable throughput
- Proprietary interconnection network
  - 34 MB/s between two nodes using MPI/MPL
  - 17 MB/s between two nodes using TCP/IP

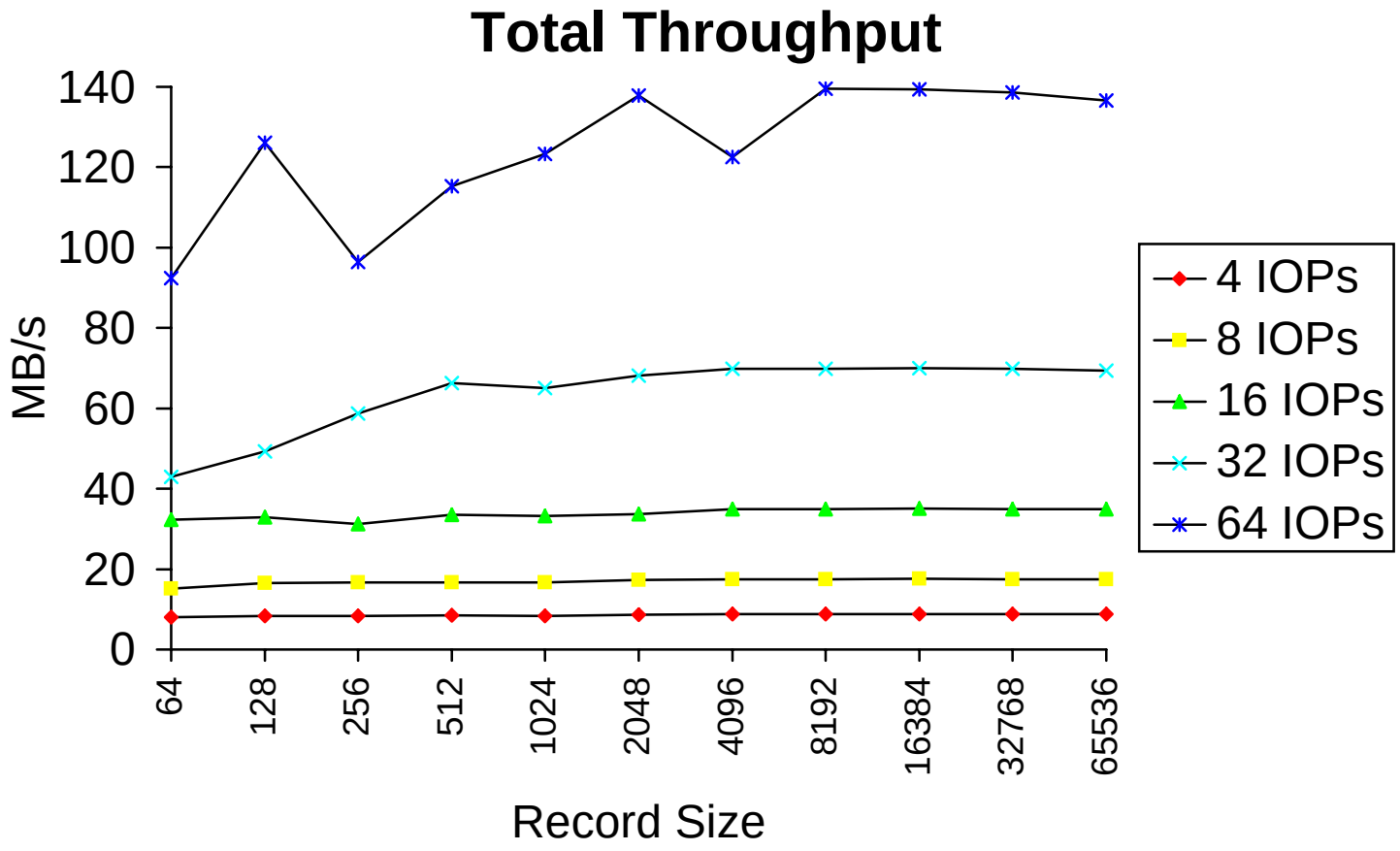
TCP/IP Performance



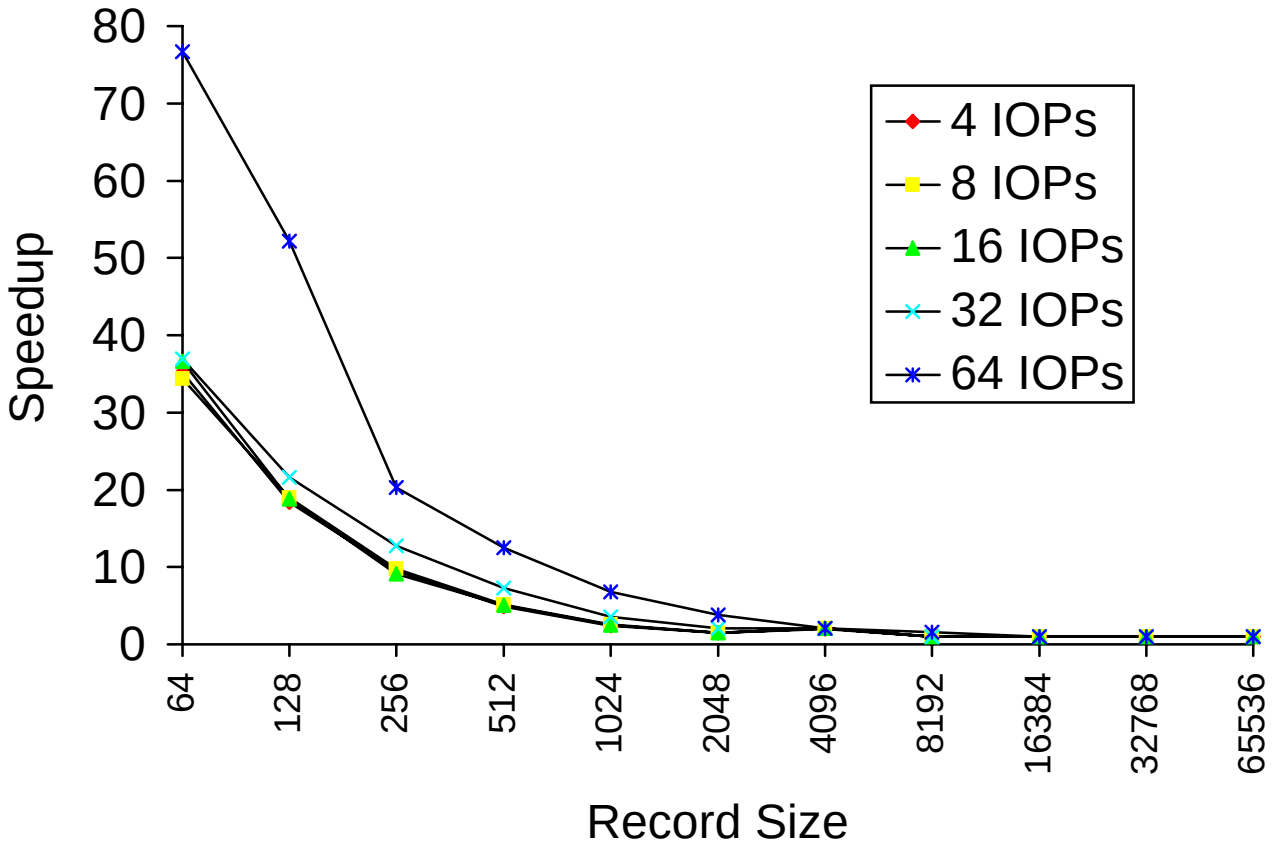
Traditional Interleaved Read



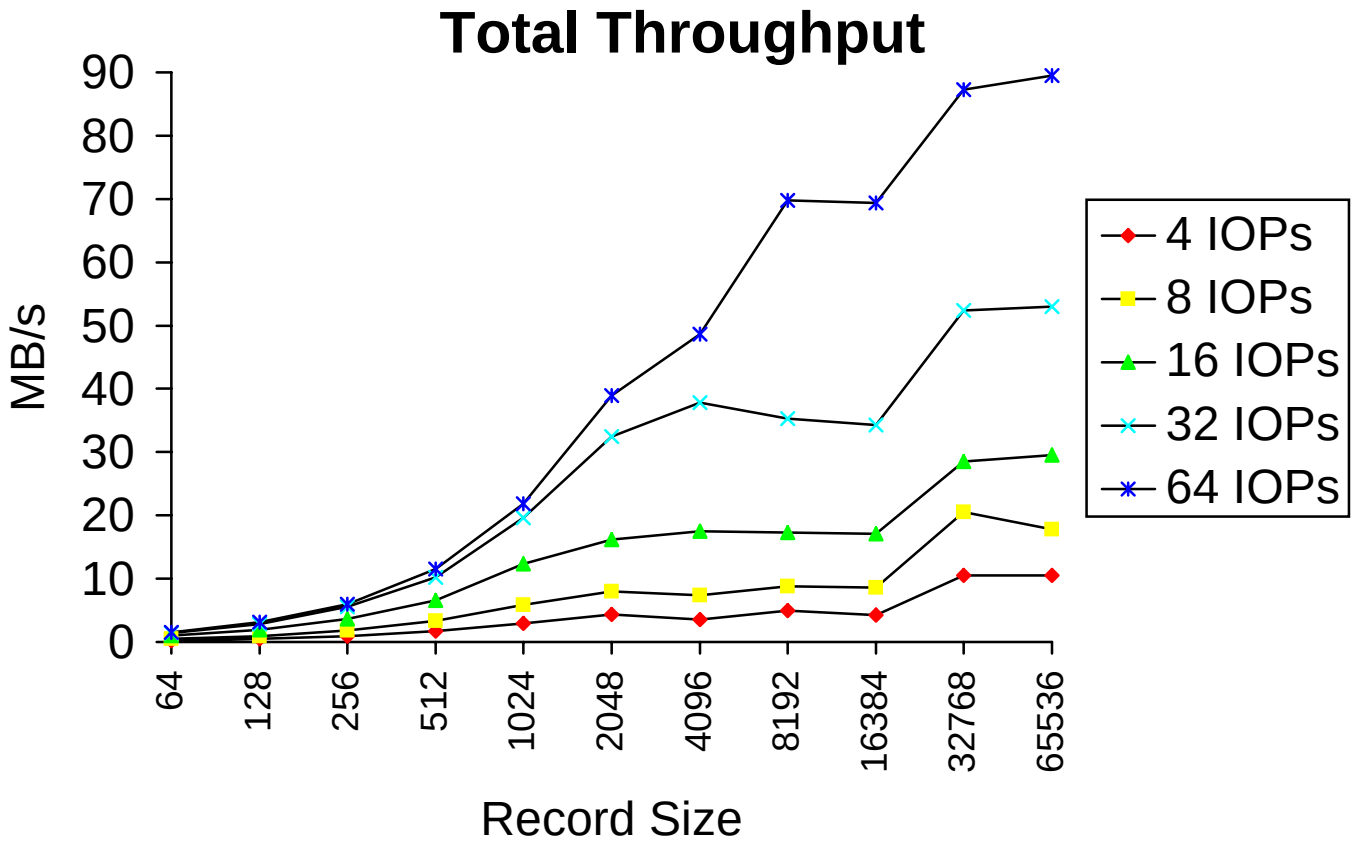
Strided Interleaved Read



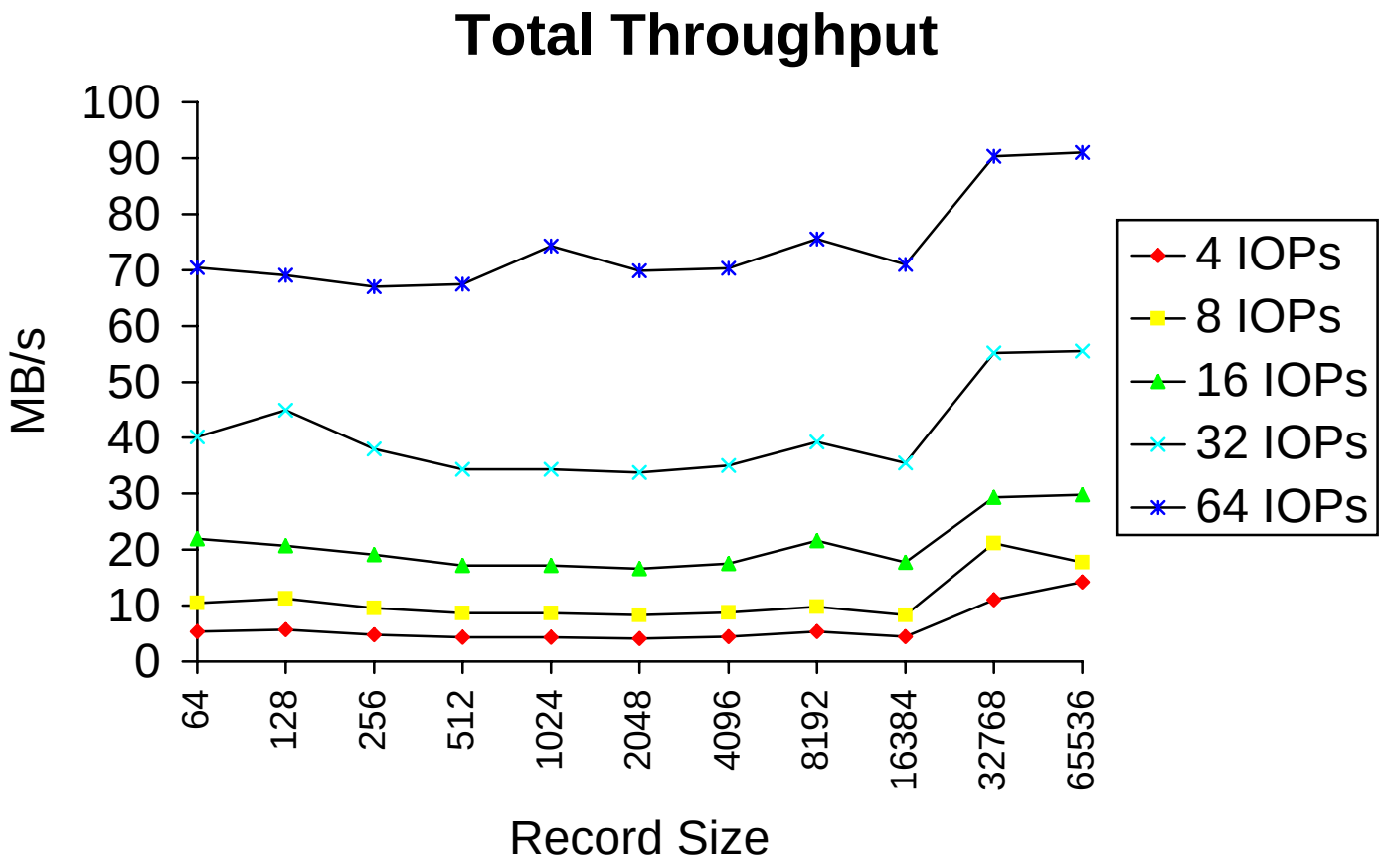
Interleaved Read Speedup



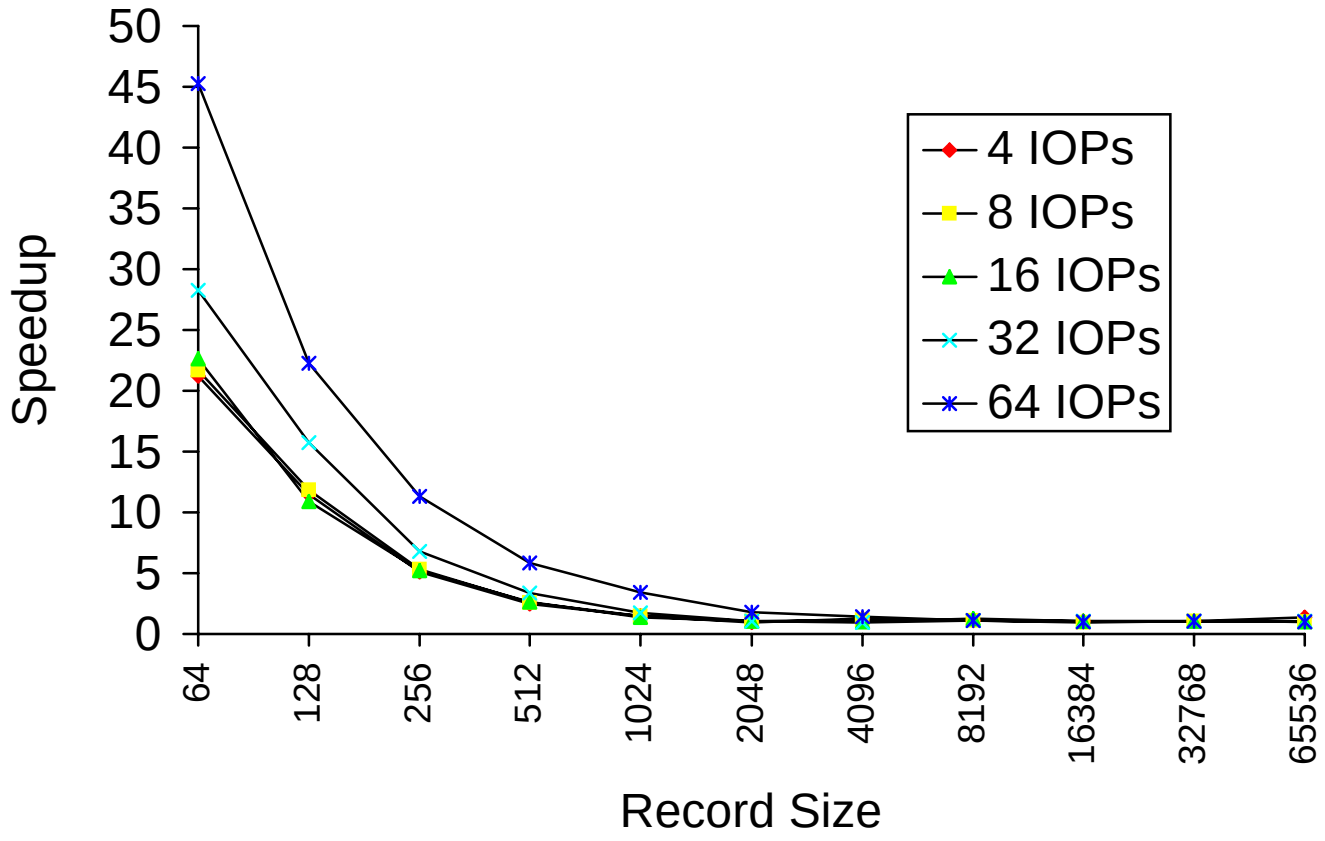
Traditional Interleaved Write



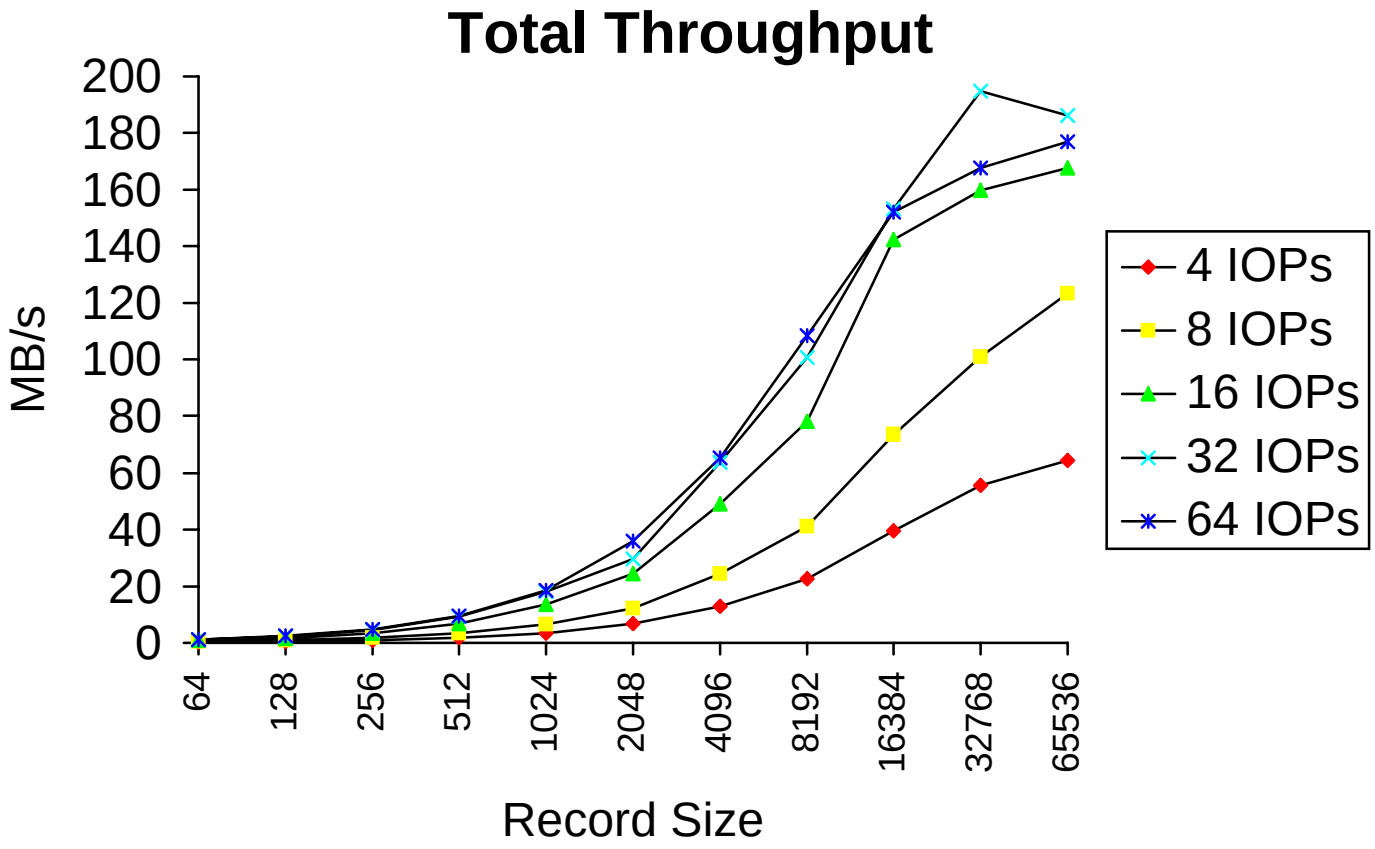
Strided Interleaved Write



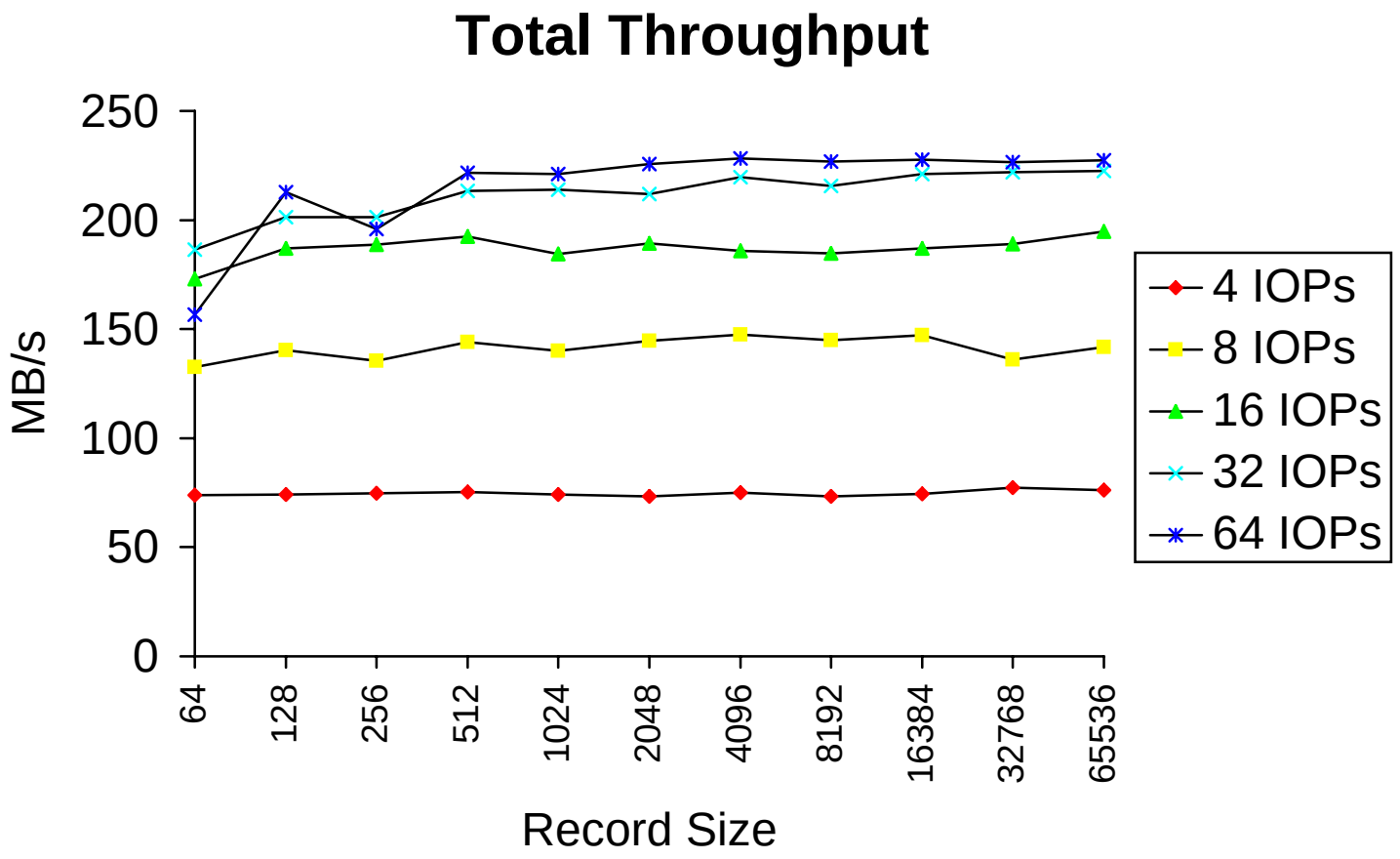
Interleaved Write Speedup



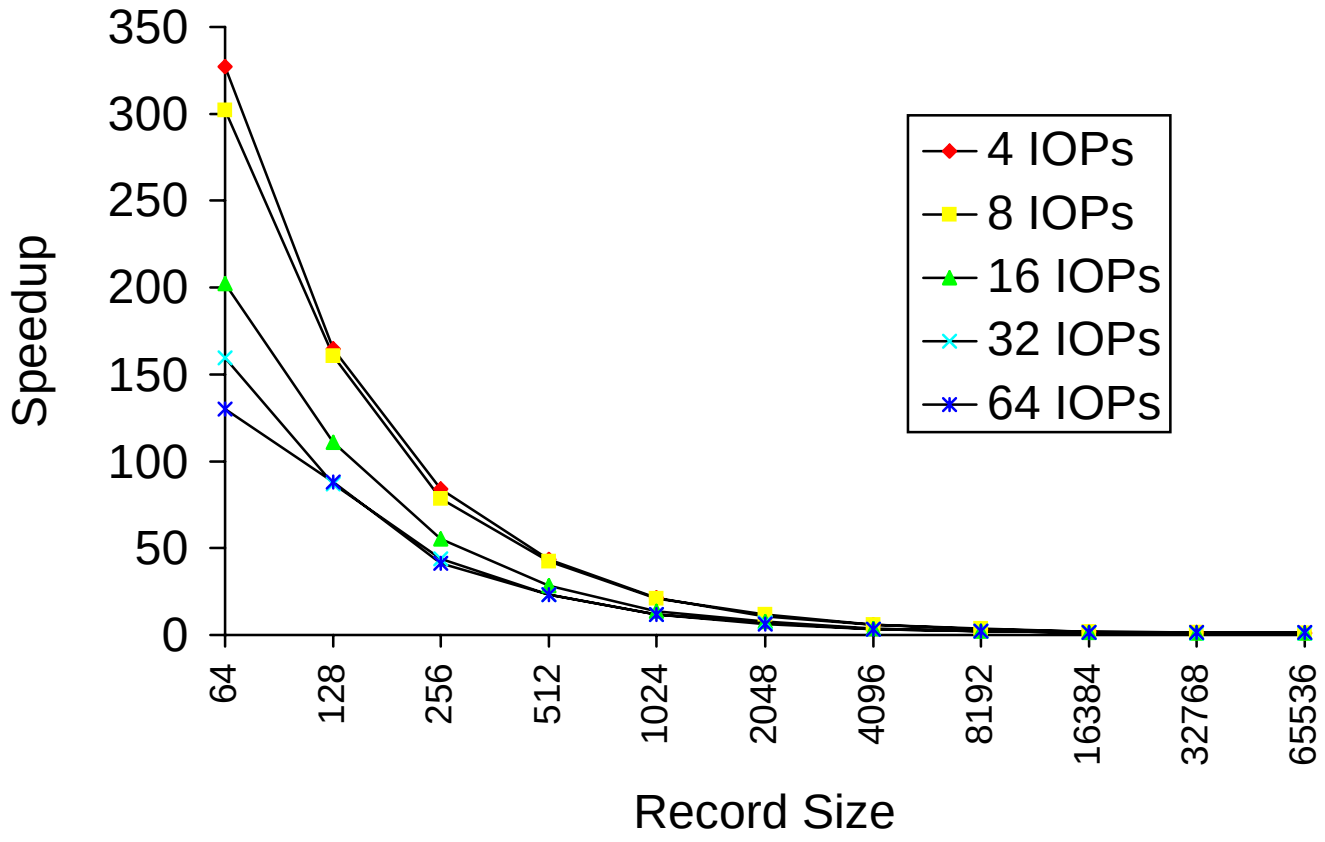
Traditional Broadcast Read



Strided Broadcast Read



Broadcast Read Speedup



Summary

- Based on analyses of production workloads, we have designed a new parallel file system
- Designed to meet needs of parallel scientific applications
- Exposes the full parallelism of the system to the application
- Performance is excellent

Future Work

- Porting benchmarks, applications, libraries, and compilers to Galley.
- Examine how to support multi-application workloads fairly and efficiently.
- Long-term project: examine possibility of moving application code to I/O nodes.

WWW

- More information and source (soon) at:
  - `http://www.cs.dartmouth.edu/~nils/galley`