Parallel IO and Fault Tolerance

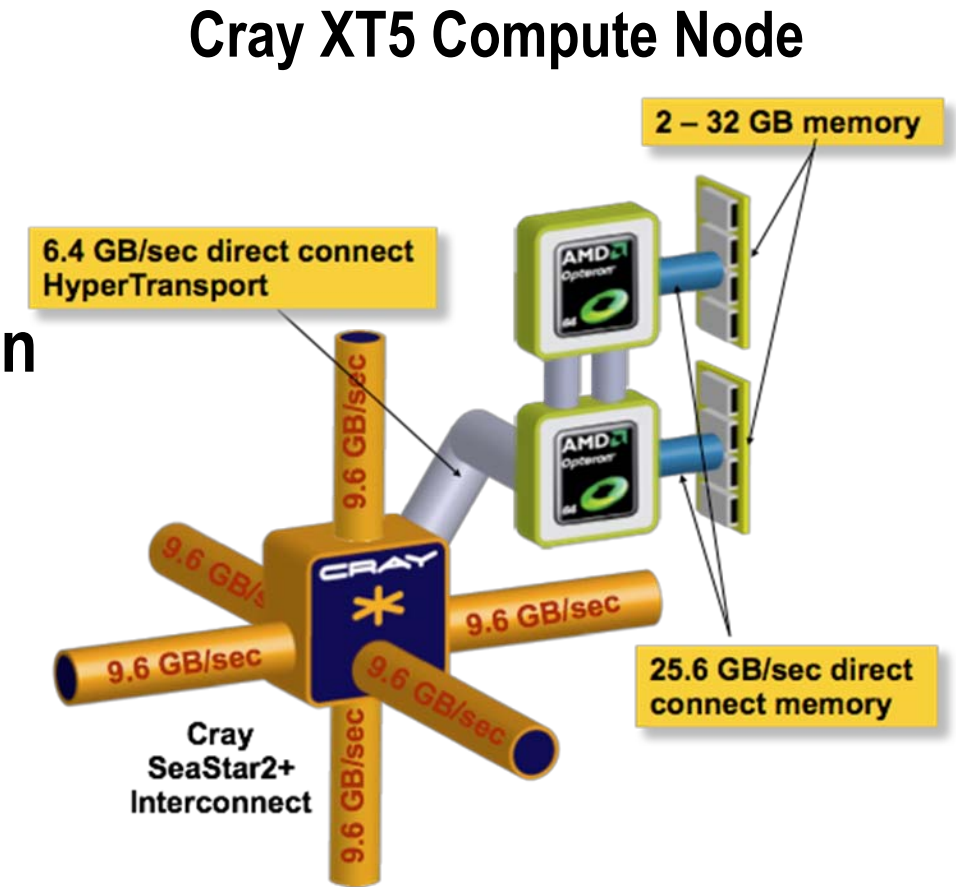
Lonnie D. Crosby

"Summer School 2009: Scaling to Petascale"

August 5, 2009

Application Performance

- Computation (FLOPs)
  - Processor
- Inter-process Communication
  - Interconnect
- Memory
  - Capacity and Speed
- I/O
  - File System



Factors which affect I/O.

- I/O is simply data migration.
  - Memory ⟷ Disk
    - Cache (L1, L2, L3)
    - RAM
    - Disk
- Size of write/read operations
  - Bandwidth vs. Latency
- Data continuity and locality on disk
  - Bandwidth vs. Latency
- Number of processes performing I/O
- Characteristics of the file system
  - Distributed or Shared

Application I/O Patterns

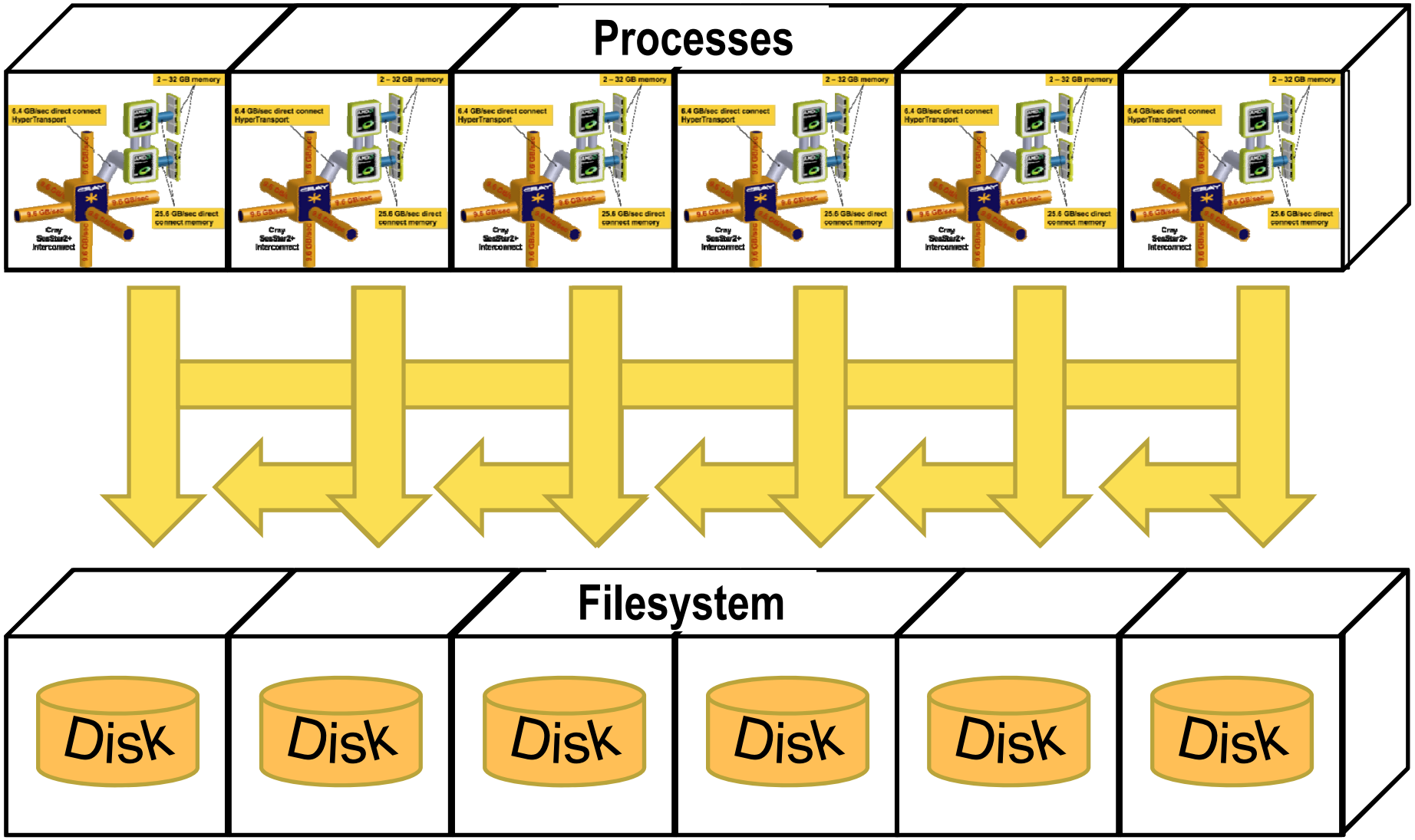
Serial I/O

- Spokesperson
  - One process performs I/O.

Parallel I/O

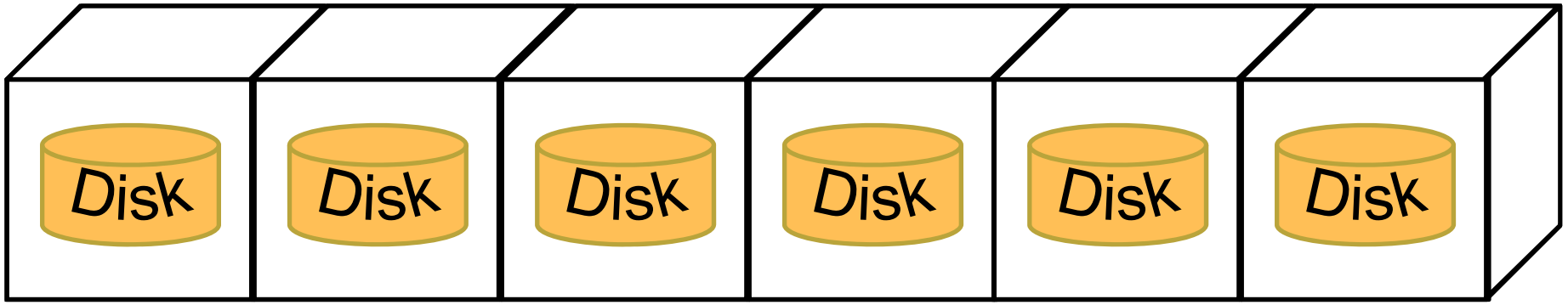
- File per Process
  - Each process performs I/O to a single file.
- Single Shared File
  - Each process collectively performs I/O to a single shared file.
- Multiple Shared Files
  - Groups of processes perform I/O to a single shared file.

Parallelism



Parallelism

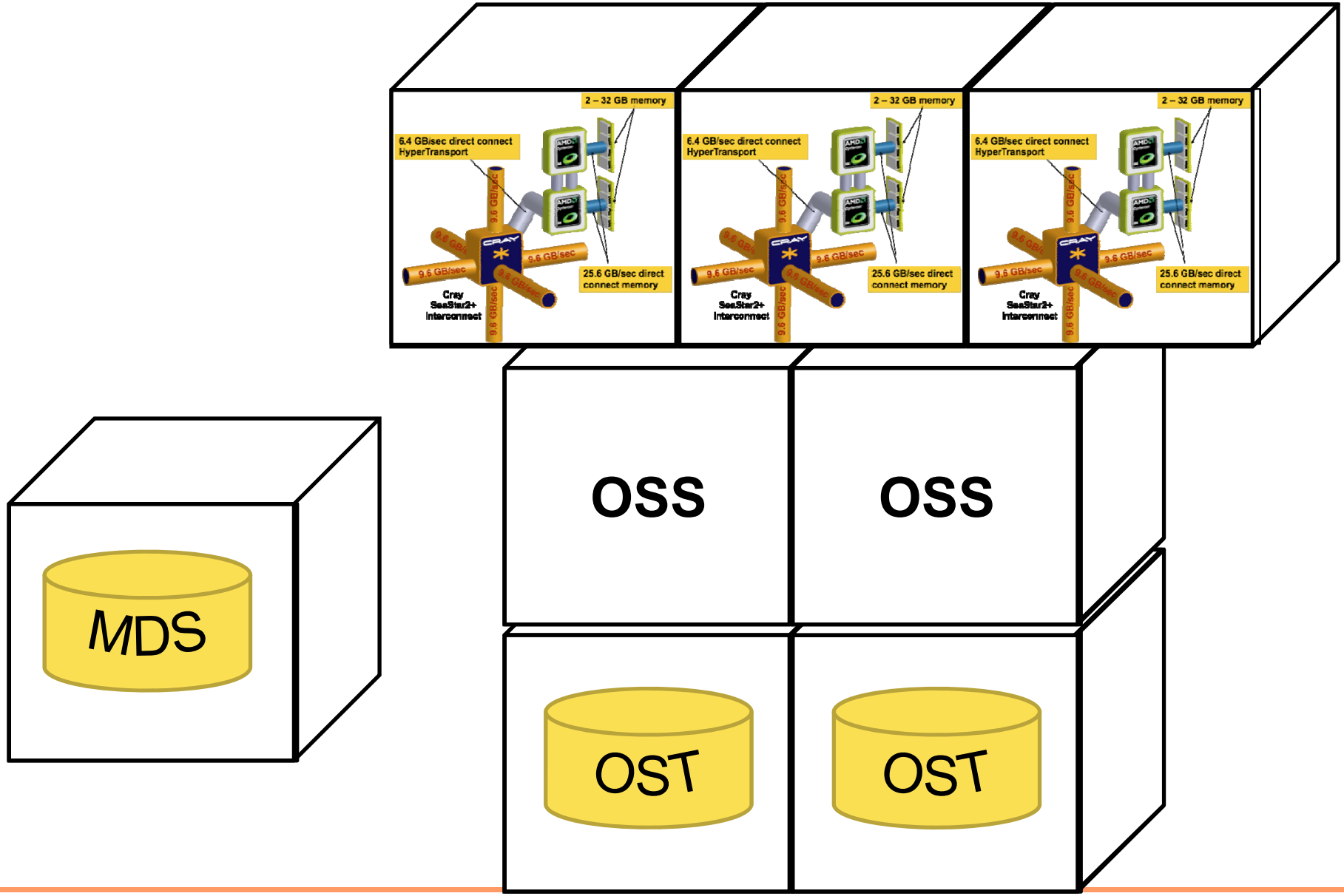
- Process level parallelism
  - MPI
  - IO Libraries (HDF5, MPI-IO, p-netCDF)
- File System parallelism
  - Distributed File System
  - Shared Parallel File System (GPFS, Lustre)



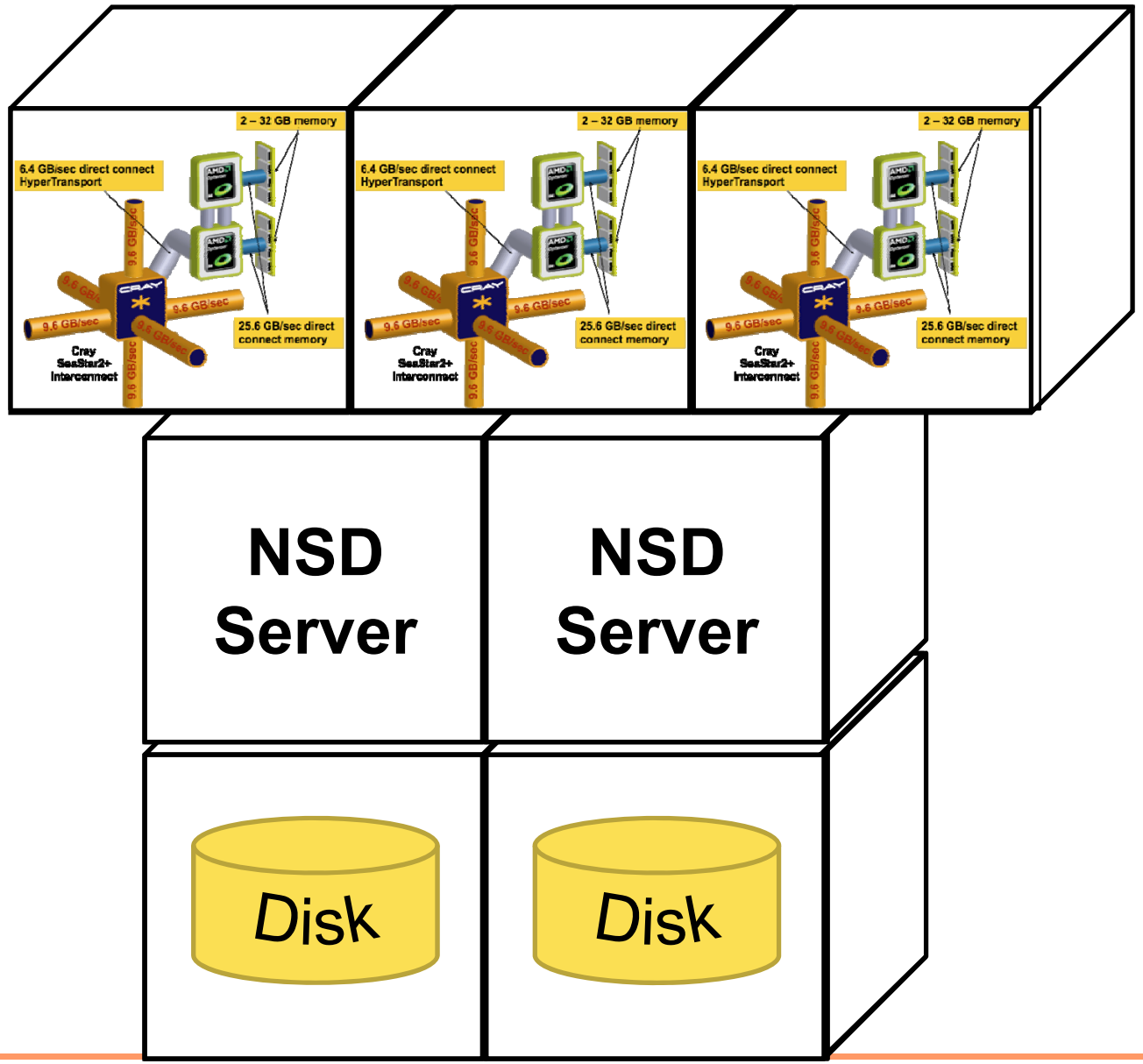
Limits of I/O

- Serial I/O
  - Is limited by the single process which performs I/O.
- Many Process I/O
  - Is limited by the number of disks which are concurrently utilized.
- Distributed File System
  - Files are localized on a single disk.
- Parallel File System
  - Files are localized on a single disk.
  - Files are striped across multiple disks.

A Bigger Picture: Lustre File System



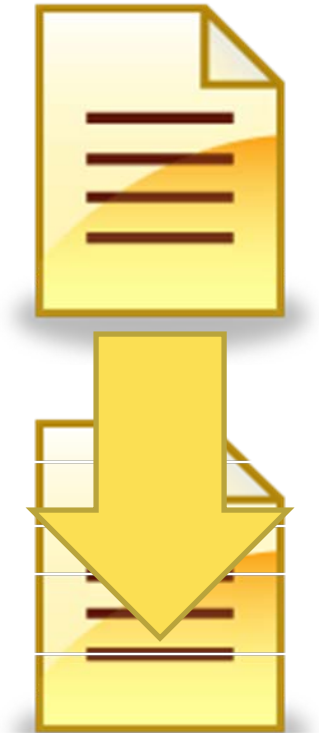
A Bigger Picture: GPFS



Lustre Striping: File Parallelism

- lfs setstripe
  - Stripe size -s (default: 1M)
  - Stripe count -c 5 (default 4, -1 All)
  - Stripe index -i 0 (default: -1 round robin)

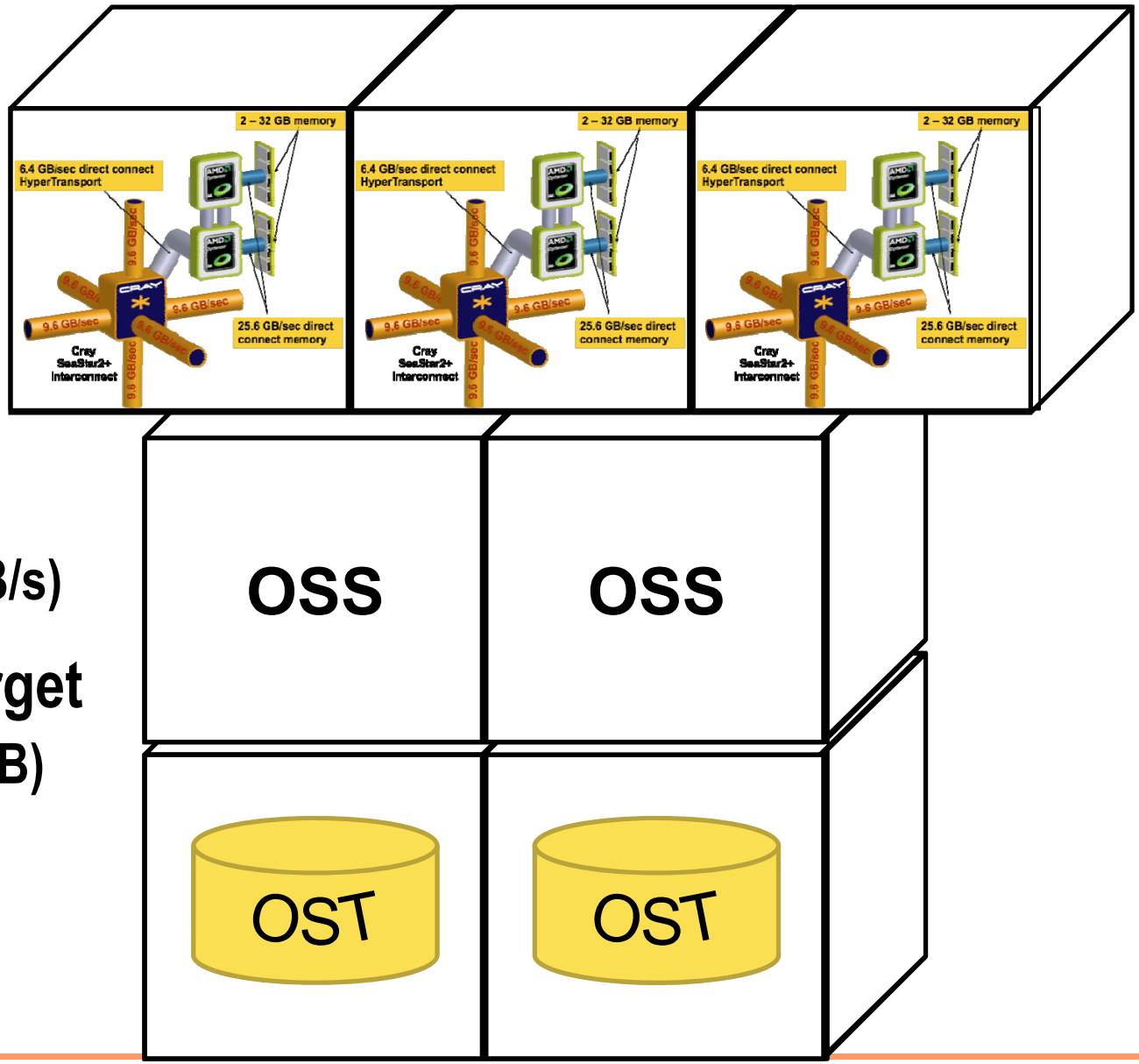
  < file | directory >



NICS

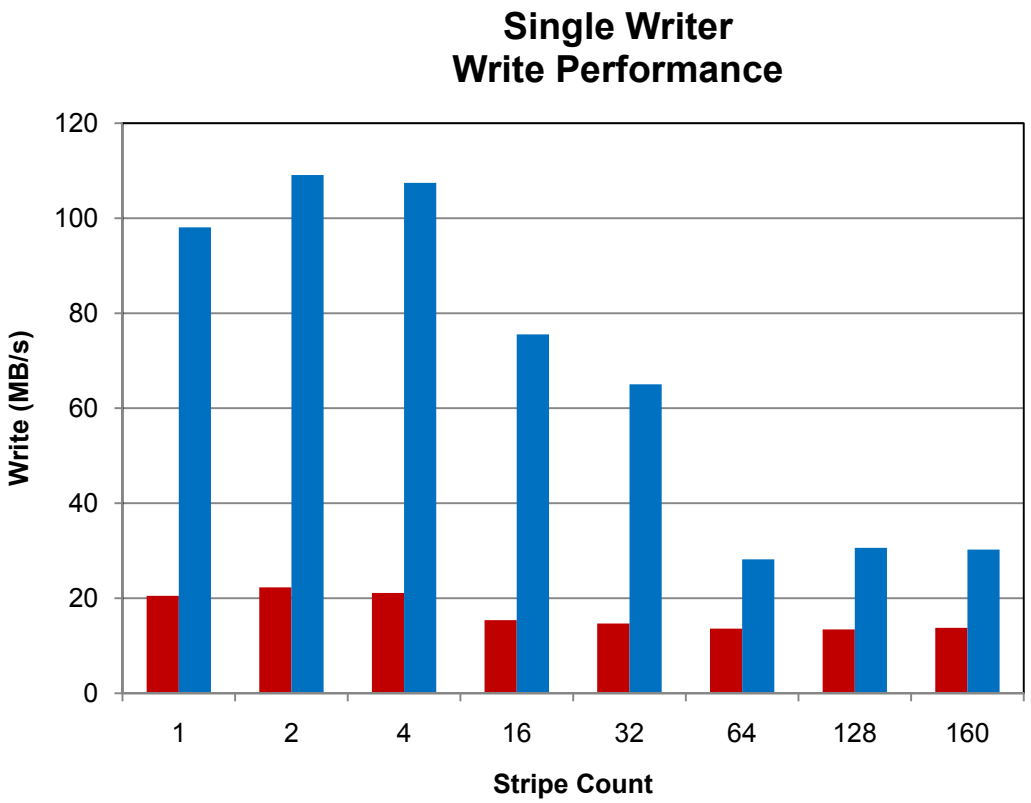
A Bigger Picture

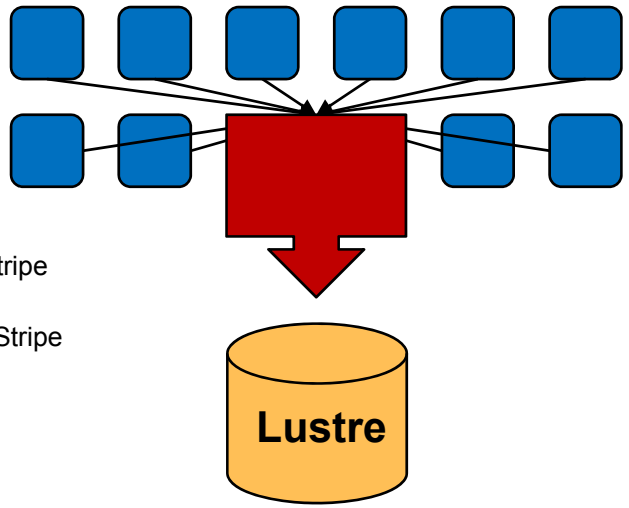
- Computational Nodes
  - Kraken: 8253
- Object Storage Server Nodes
  - Kraken: 48 (30 GB/s)
- Object Storage Target
  - Kraken: 336 (2.4 PB) [7.2 TB Disk]



Spokesperson – Serial I/O Importance of data locality

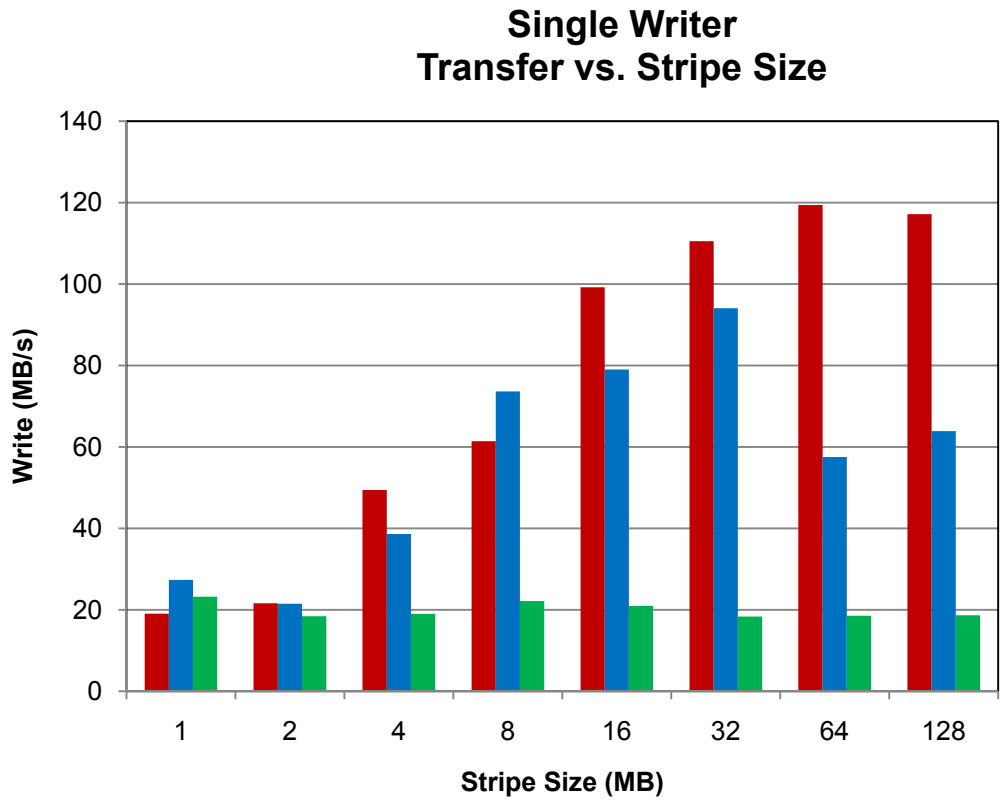
- 32 MB per OST (32 MB – 5 GB) and 32 MB Transfer Size

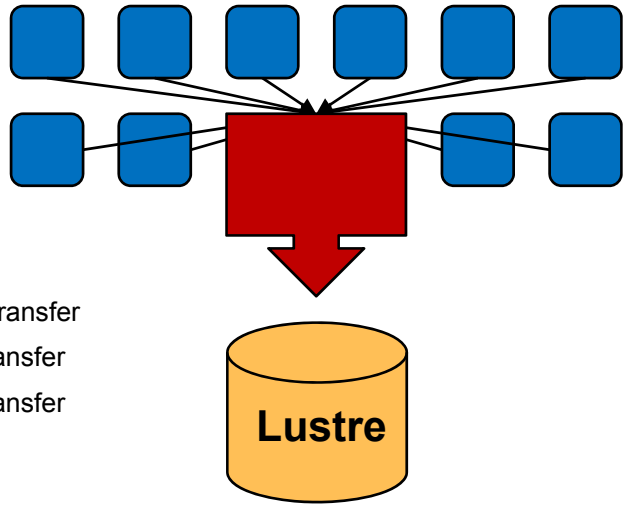




Spokesperson – Serial I/O Importance of data continuity

- Single OST, 256 MB File Size





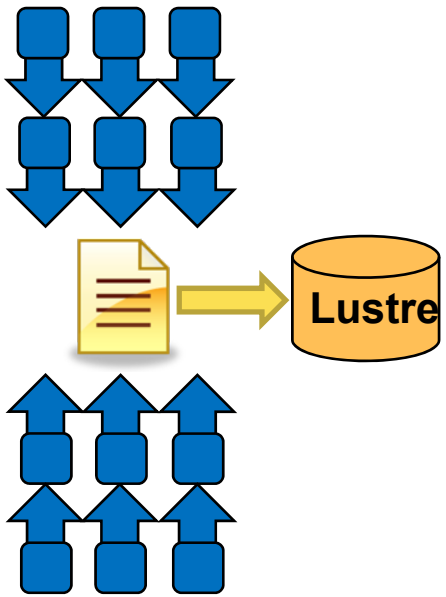
Data Locality and Continuity

- Data Locality
  - Performance is decreased when a single process accesses multiple disks.
  - Is limited by the single process which performs I/O.

- Data Continuity
  - Larger read/write operations improve performance.
  - Larger stripe sizes improve performance (places data contiguously on disk).
  - Either may become a limiting factor.

Single Shared File

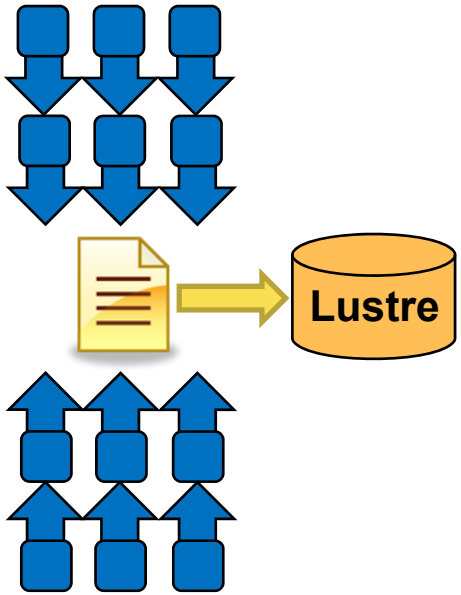
- Important Considerations
  - Data locality
  - Data continuity
- Parallel file Structure

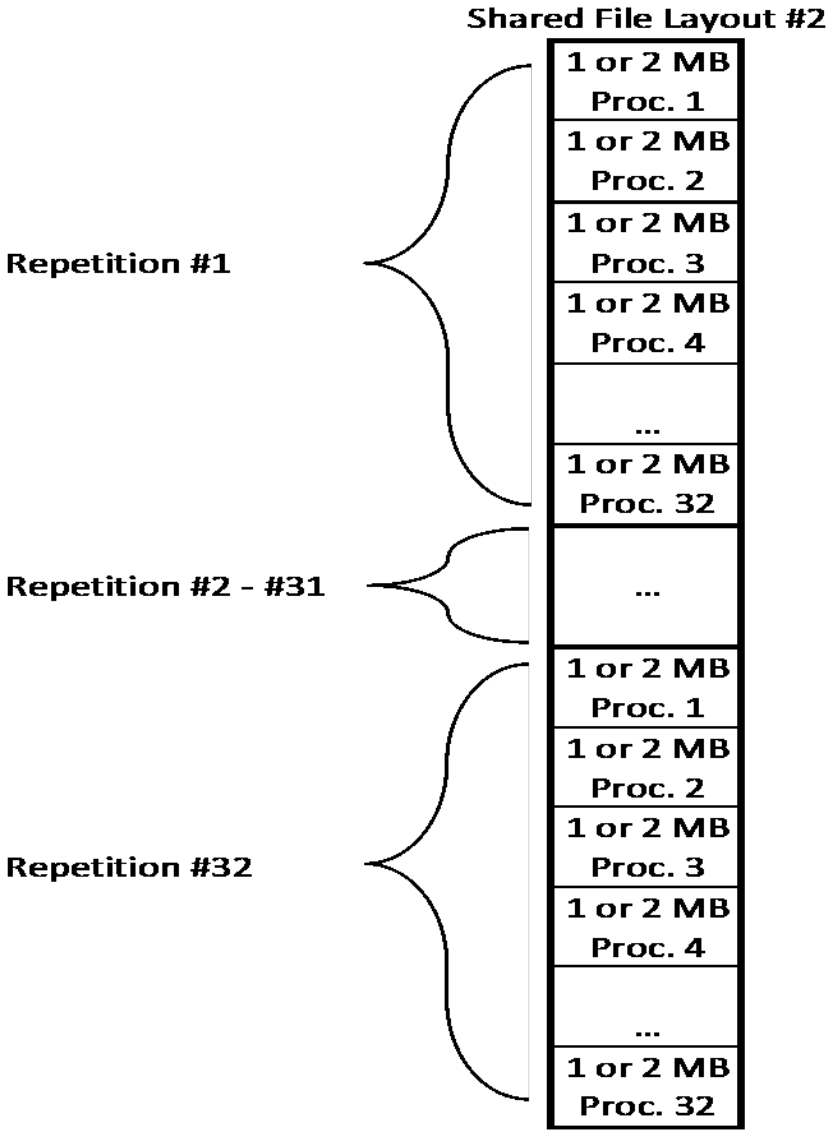


Shared File Layout #1

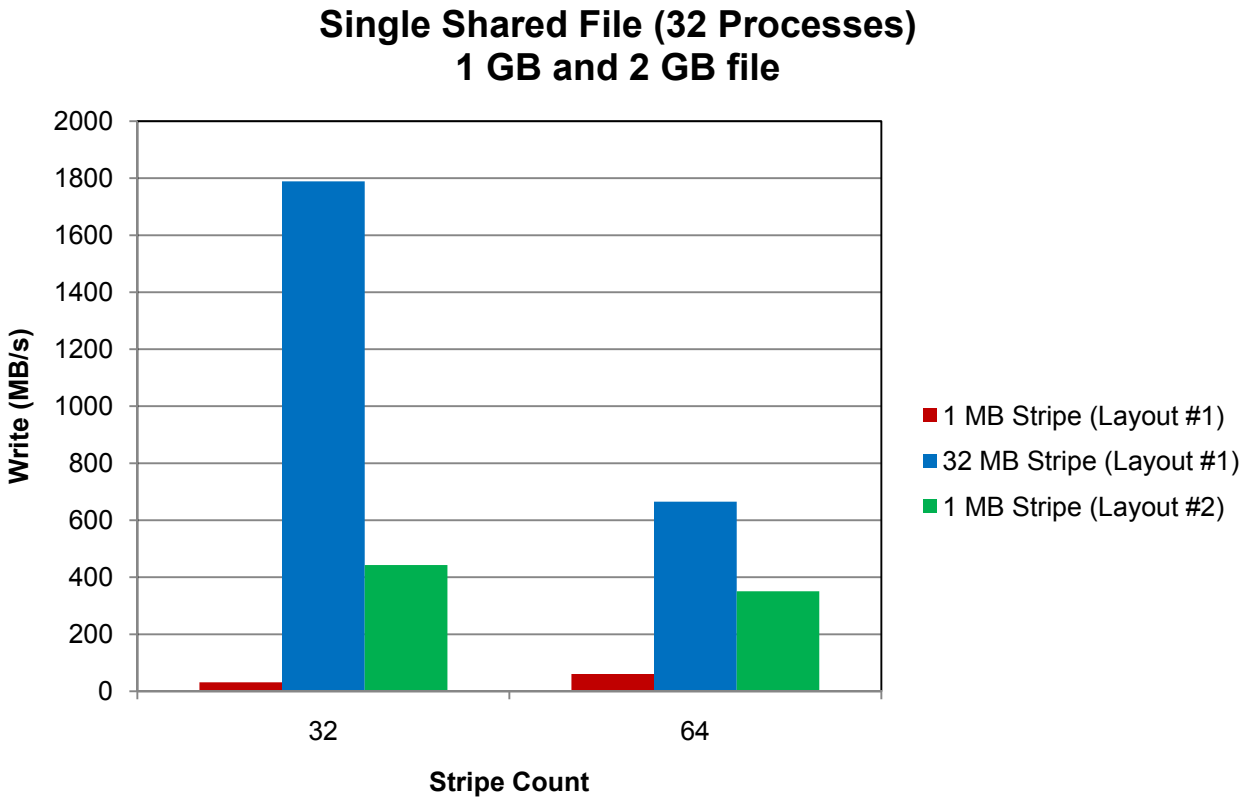
32 or 64 MB Proc. 1
32 or 64 MB Proc. 2
32 or 64 MB Proc. 3
32 or 64 MB Proc. 4
...
32 or 64 MB Proc. 32

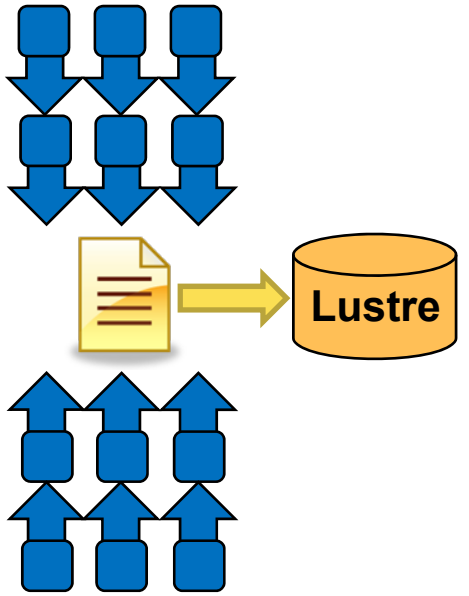
Single Shared File





Single Shared File





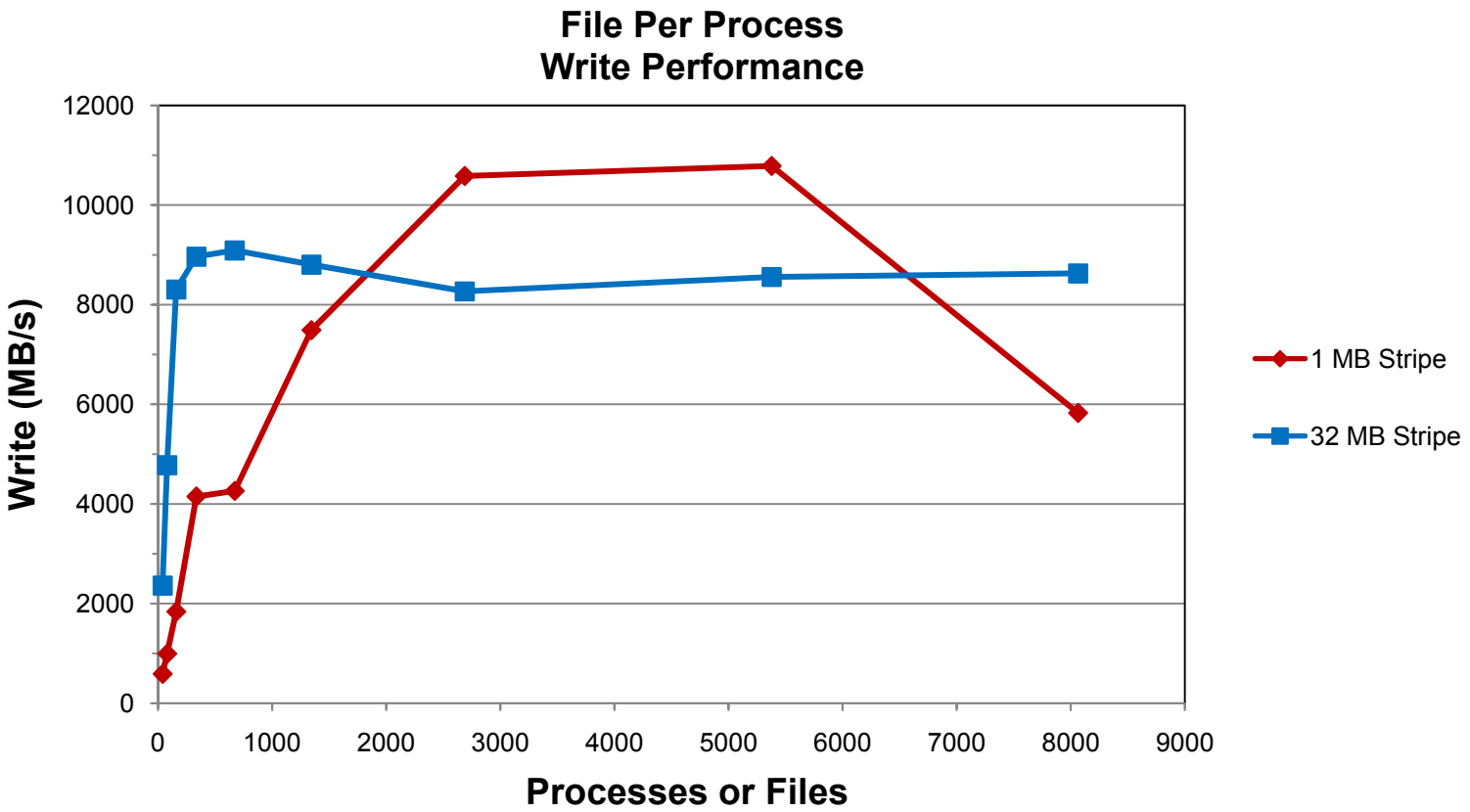
Data Locality and Continuity

- Data Locality
  - Performance is increased when portions of a shared file are localized on a single drive.

- Data Continuity
  - Larger read/write operations improve performance.
  - Larger stripe sizes improve performance (places data contiguously on disk).
  - Either may become a limiting factor.

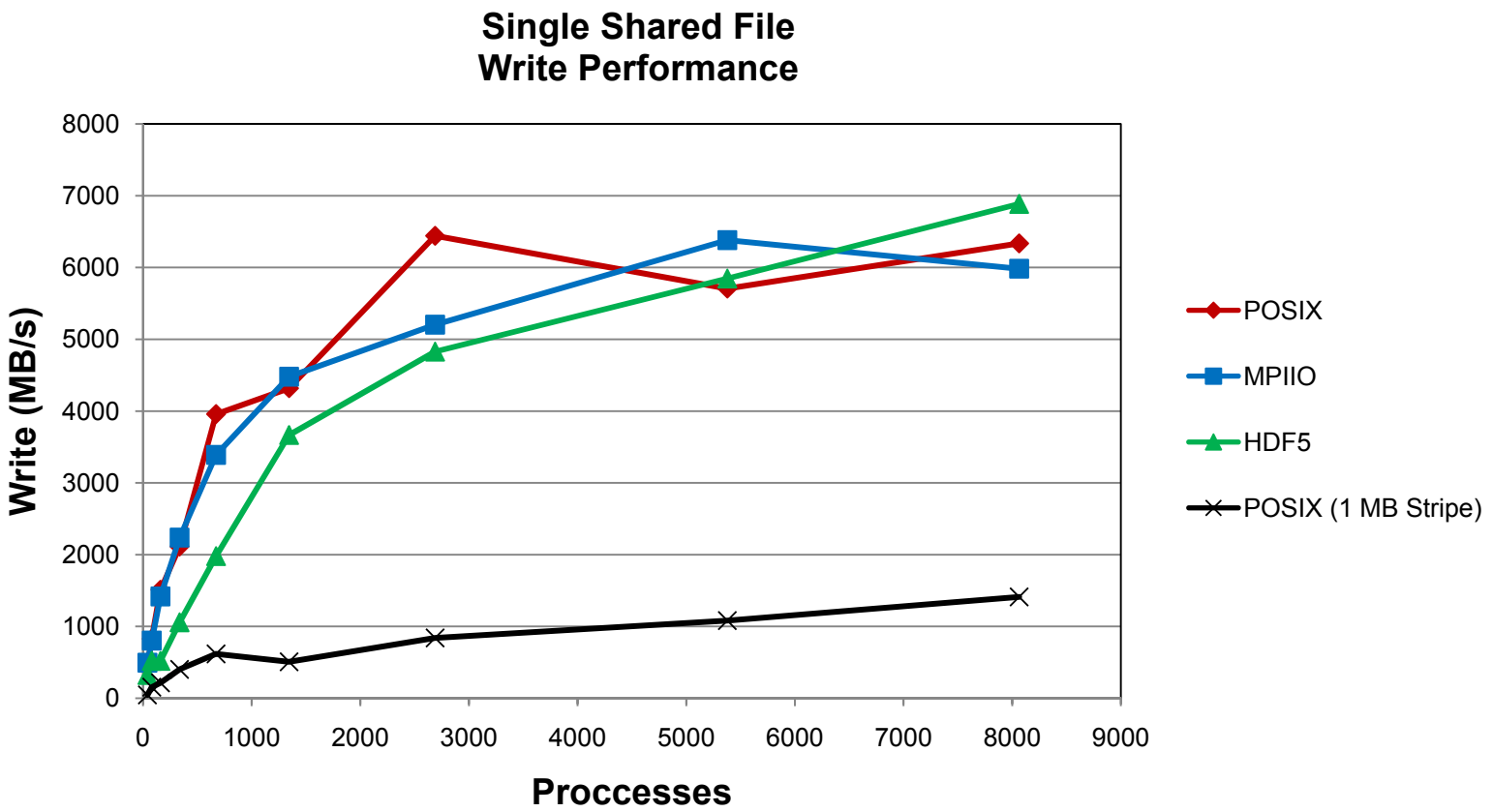
Scalability: File Per Process

- 128 MB per file and a 32 MB Transfer size



Scalability: Single Shared File

- 32 MB per process, 32 MB Transfer size and Stripe size

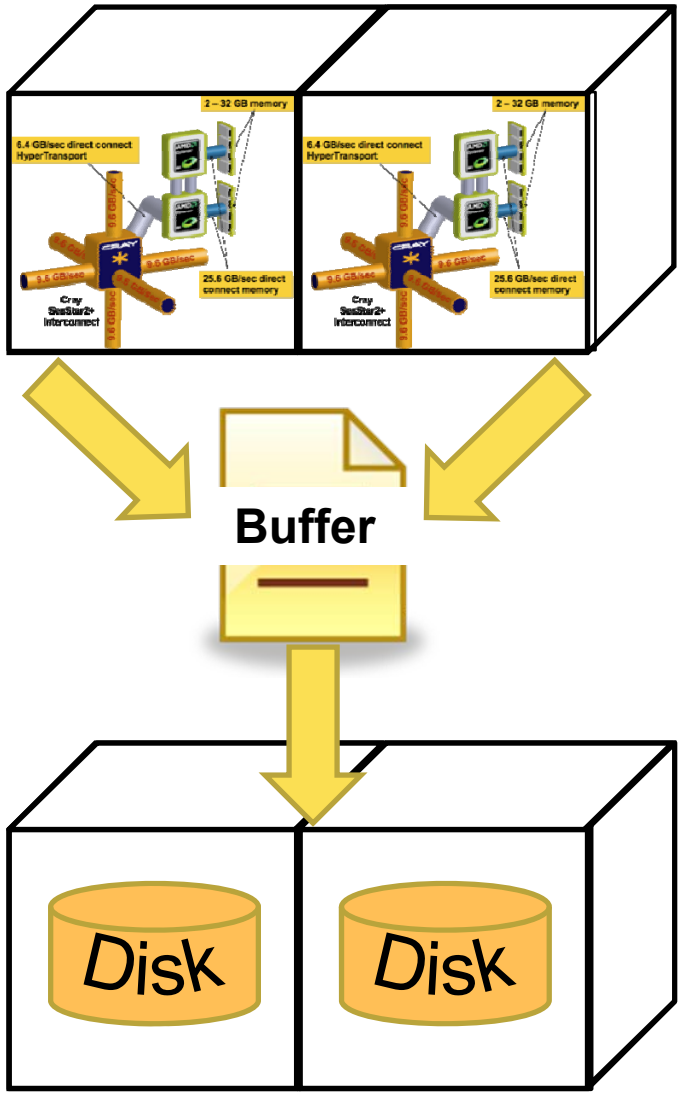


Scalability

- Serial I/O
  - Is not scalable. Limited by single process which performs I/O.
- File per Process
  - Limited at large process/file counts by:
    - Metadata Operations
    - Contention on a single drive
- Single Shared File
  - Limited at large process counts by contention on a single drive.
  - File striping limitation of 160 OSTs in Lustre

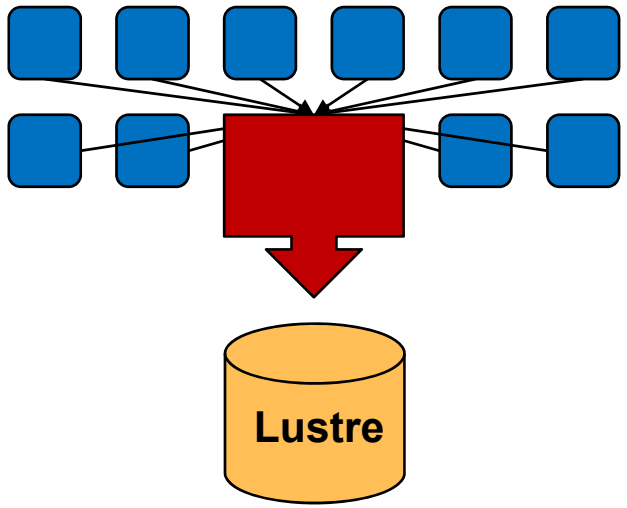
Buffered I/O

- Advantages
  - Aggregates smaller read/write operations into larger operations.
  - Examples: OS Kernel Buffer, MPI-IO Collective Buffering
- Disadvantages
  - Requires additional memory for the buffer.
  - Can tend to serialize I/O.
- Caution
  - Frequent buffer flushes can adversely affect performance.



Standard Output and Error

- Standard Ouput and Error streams are effectively serial I/O.
- Generally, the MPI launcher will aggregate these requests. (Example: mpirun, mpiexec, aprun, ibrun, etc..)
- Disable debugging messages when running in production mode.
  - "Hello, I'm task 32000!"
  - "Task 64000, made it through loop."



Binary Files and Endianess

- **Writing a big-endian binary file with compiler flag byteswapio**

```
File "XXXXXX"
                  Calls        Megabytes       Avg Size
  Open                1
  Write         5918150      23071.28062           4088
  Close               1
  Total         5918152      23071.28062           4088
```

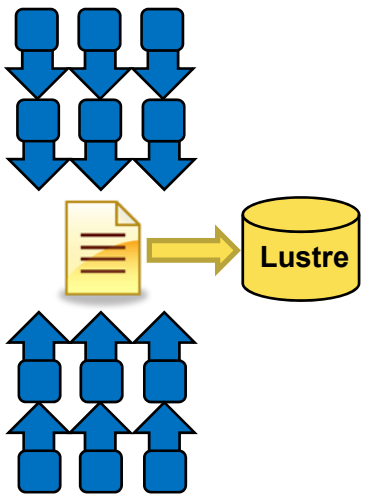
- **Writing a little-endian binary**

```
File "XXXXXX"
                  Calls        Megabytes       Avg Size
  Open                1
  Write             350      23071.28062       69120000
  Close               1
  Total             352      23071.28062       69120000
```

- **Can use more portable file formats such as HDF5, NetCDF, or MPI-IO.**

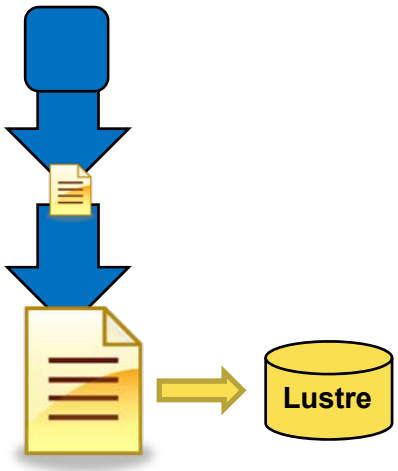
Case Study: Parallel I/O

- A particular code both reads and writes a 377 GB file. Runs on 6000 cores.
  - Total I/O volume (reads and writes) is 850 GB.
  - Utilizes parallel HDF5
- Default Stripe settings: count 4, size 1M, index -1.
  - 1800 s run time (~ 30 minutes)
- Stripe settings: count -1, size 1M, index -1.
  - 625 s run time (~ 10 minutes)
- Results
  - 66% decrease in run time.



Case Study: Buffered I/O

- A post processing application writes a 1GB file.
- This occurs from one writer, but occurs in many small write operations.
  - Takes 1080 s (~ 18 minutes) to complete.
- IOBUF was utilized to intercept these writes with 64 MB buffers.
  - Takes 4.5 s to complete. A 99.6% reduction in time.




```
File "ssef_cn_2008052600f000"
                  Calls          Seconds       Megabytes   Megabytes/sec   Avg Size
  Open                1         0.001119
  Read              217         0.247026        0.105957        0.428931        512
  Write         2083634         1.453222     1017.398927      700.098632        512
  Close               1         0.220755
  Total         2083853         1.922122     1017.504884      529.365466        512
  Sys Read            6         0.655251      384.000000      586.035160   67108864
  Sys Write          17         3.848807     1081.145508      280.904052   66686072
  Buffers used              4 (256 MB)
  Prefetches                6
  Preflushes               15
```


NICS

Fault Tolerance: Faults

- MTBF Mean Time Between Failure
- MTBI Mean Time Between Interrupt
  - (Includes scheduled maintenance)

	MTBF	MTBI	Period
Kraken XT5	141.7 hours (5.9 days)	89.6 hours (3.7 days)	Feb 09 - July 09
Kraken XT5	139.5 hours (5.8 days)	91.0 hours (3.8 days)	April 09 - June 09

	Total Jobs	Jobs Failed*	Period
Kraken XT5	70,016	1409 (2.0%)	April 09 - June 09

* Due to System Failure

Fault Tolerance: Tolerance

- First, allow application to generate checkpoint files.
  - Should be minimal in size.
  - Should not be written too often.
- Keeping checkpoint files minimal
  - Only incorporate unique information. Allow application to calculate or derive appropriate information.
- Keeping the checkpoint generation low.
  - The goal isn't to keep all information at all times. (checkpointing after every iteration.)
  - Pick a write frequency which allows for a reasonable loss of computation time.

References

- Lustre File System – White Paper October 2008
  - http://www.sun.com/software/products/lustre/docs/lustrefilesystem_wp.pdf
- GPFS: Concepts, Plannin, and Installation Guide
  - http://www.publib.boulder.ibm.com/epubs/pdf/a7604132.pbf
- Introduction to HDF5
  - http://www.hdfgroup.org/HDF5/doc/H5.intro.html
- The NetCDF Tutorial
  - http://www.unidata.ucar.edu/software/netcdf/docs/netcdf-tutorial.pdf