

# HPCS I/O Scenarios

John Carrier

[carrier@cray.com](mailto:carrier@cray.com)

Lustre Users Group

April 24, 2012



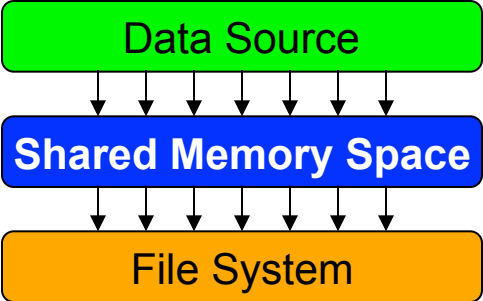
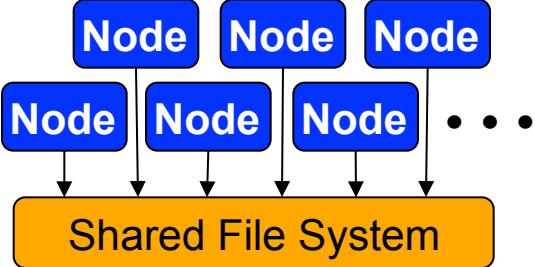
“Comparing system performance tools that measure I/O with I/O benchmarks is like comparing apples with flying pigs.”

Henry Newman  
*personal communication*  
*April 23, 2012*

# What are the HPCS Scenarios?

- As part of its High Productivity Computing Systems (HPCS) program, DARPA's HPCS Mission Partners provided vendors with a set of 14 **scalable** application workloads  
“HPCS Mission Partner File I/O Scenarios”  
<http://sourceforge.net/projects/hpcs-io/files/DARPA.HPCS.IO.Scenarios.2011.pdf>
- The Scenarios are not by themselves benchmarks; instead, they provide well-defined rules for objectively evaluating storage system scalability
- For all Scenarios, scaling performance is more important than absolute throughput for a single configuration
- The Scenarios are meant to show how a storage solution will scale when deployed with **real workloads**

# HPCS I/O Environments

Capture Environment	Parallel Environment
<p>Capture I/O depends on the ability of a single node to create files from streaming data</p> 	<p>Parallel I/O has thousands of nodes accessing the shared file system using file per process (N-to-N) or shared file (N-to-1) access patterns</p> 

HPCS Mission Partners preferred one of these two I/O Environments

# HPCS I/O Scenarios

1. Single stream with large data blocks operating in half duplex mode
  2. Single stream with large data blocks operating in full duplex mode
  3. Multiple streams with large data blocks operating in full duplex mode
  4. Extreme file creation rates **Capture Environment**
5. Checkpoint/restart with large I/O requests **Parallel Environment**
  6. Checkpoint/restart with small I/O requests
  7. Checkpoint/restart Large file count per directory large I/Os
  8. Checkpoint/restart large file count per directory small I/Os
  9. Walking through directory trees
  10. Parallel walking through directory trees
  11. Random `stat()` system call to files in the file system (one process)
  12. Random `stat()` system call to files in the file system (multiple proc's)
  13. Small block random I/O to multiple files
  14. Small block random I/O to a single file

- Lustre operates best in Parallel I/O Environment
- But don't forget there are customers for Capture I/O

# Parallel file system workloads

- Characterizing parallel file systems can generally be summarized by evaluating the impact of the following on scalable I/O performance:
  - File access method
    - shared (N:1)
    - file per process (N:N)
  - I/O size
    - large [1 MB and 10 MB ] (eg, segmented shared files)
    - small [32 KB and 128 KB ](eg, strided shared files)
  - File access pattern
    - sequential
    - random
  - Metadata IOPs

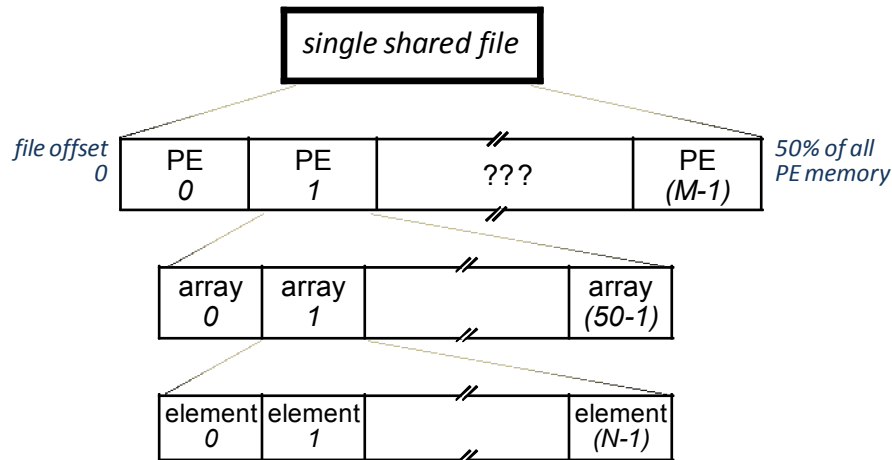
# HPCS Checkpoint/Restart Scenarios

- Scenarios 5-8 vary by I/O record size and file access pattern

Scenario	I/O Record Size	File Access Pattern
5	Large 1% of node memory	Shared (N-1) Segmented access
6	Small 10,000 * (Real*8)	Shared (N-1) Strided access
7	Large 1% of node memory	File per Process (N-N)
8	Small 10,000 * (Real*8)	File per Process (N-N)

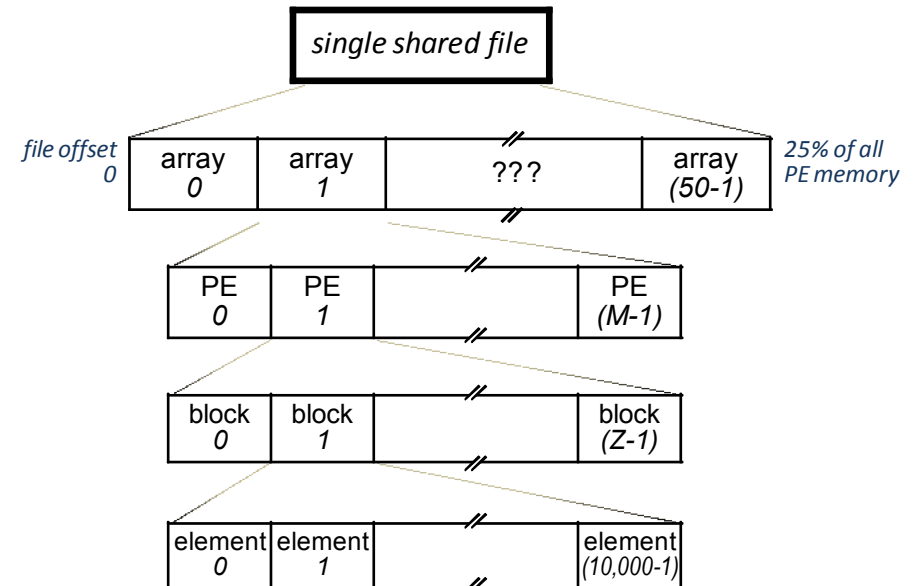
- Each Scenario runs on at least three (3) different hardware configurations to demonstrate system scalability using hardware bottleneck analysis

# Shared File Scenarios 5 & 6



## Scenario 5 : Shared, Segmented I/O

- $M$  processes (PEs) share one file
- File organized by PE
- Each PE creates 50 arrays
- Each array has  $N$  Real\*8 elements
- All 50 arrays on each PE use 50% of node memory
- Each PE seeks to its location in the file based on its PE index
- Each PE writes each array of  $N$  elements sequentially from this offset



## Scenario 6 : Shared, Strided I/O

- $M$  processes (PEs) share one file
- File organized by Array
- Each PE creates 50 arrays
- Each array has  $Z$  blocks of 10,000 Real\*8 elements
- All 50 arrays on each PE use 25% of node memory
- For each array, each PE seeks a new location in the file based on the Array index
- Each PE writes the  $Z$  blocks of 10,000 elements sequentially from this offset



# HPCS I/O Scenarios Tests

- DARPA provided the Scenarios, not the actual tests
  - A number of proprietary implementations of the Scenarios exist, but none have been available as open source
- Cray implemented the Scenarios for its HPCS program and published the source on SourceForge under Cray's BSD-compliant Open Source License
  - <http://hpcs-io.cray.com/>
  - The Scenarios and Cray's implementation are file system agnostic
  - The repository includes scripts with example command line parameters for running each test

# Testing on ORNL's Cray XT4

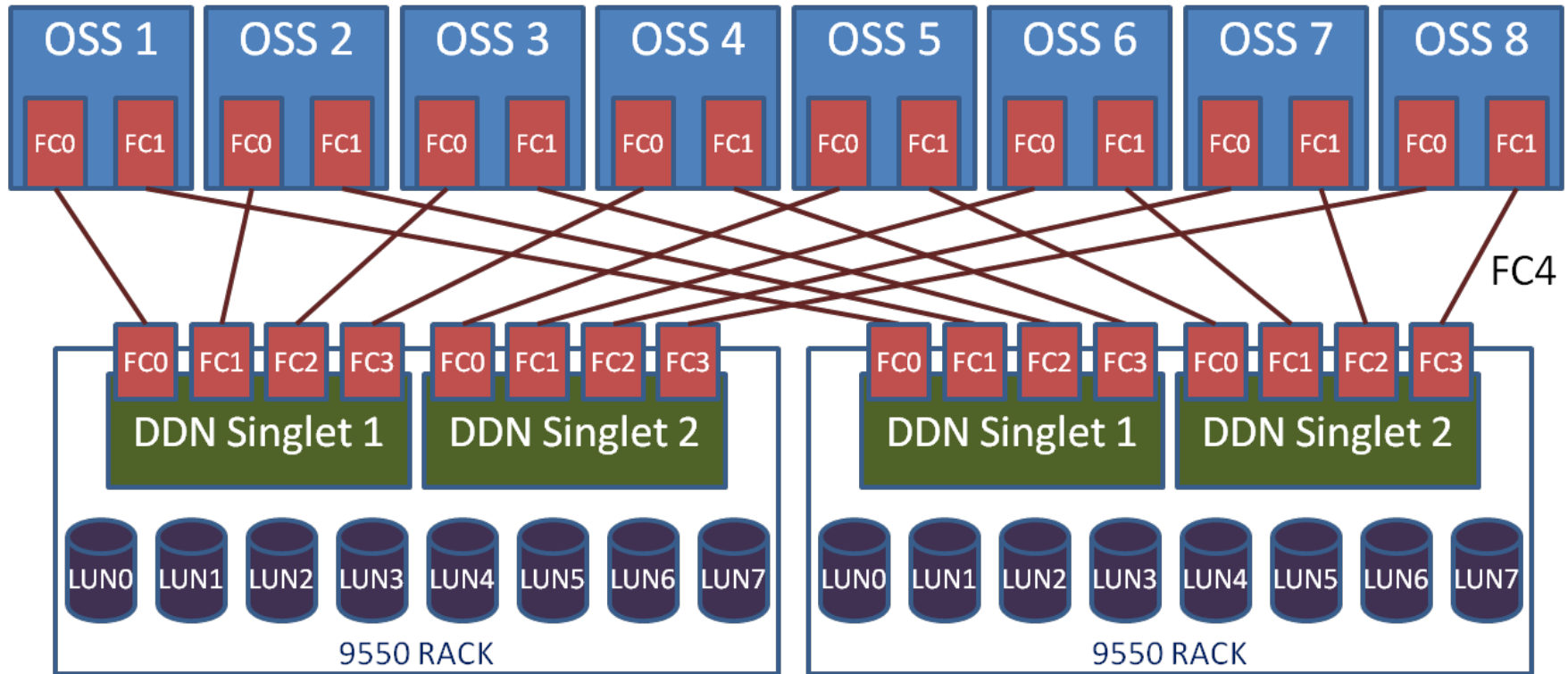
- In March 2011, after decommissioning their Cray XT4, Oak Ridge National Laboratory (ORNL) provided Cray dedicated access to the supercomputer and its file system to validate our initial implementation of the HPCS Scenarios tests



The system had

- 18 DDN 9550 storage controllers
- 72 Lustre object storage servers (OSS), running on XT4 IO nodes

# Scalable Storage Unit (SSU) Definition



- Each SSU had 8 OSS nodes, 2 DDN 9550 racks, 16 OSTs
- Each OSS had two OSTs, one on each DDN rack
- Theoretical performance per SSU was ~5 GB/s

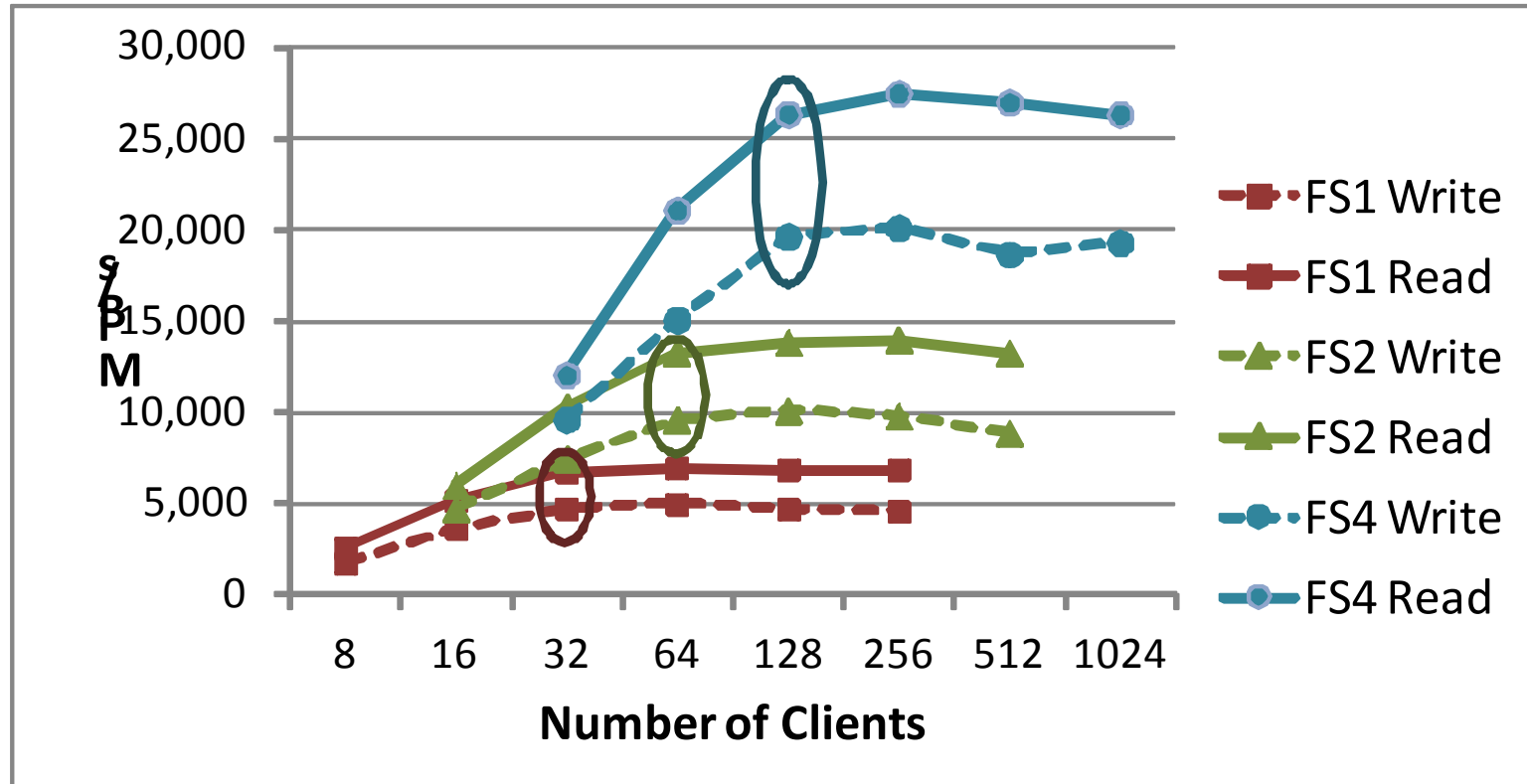
# File System Definitions

- Cray reconfigured the storage to create three file systems, each with its own metadata server, using one, two and four SSUs

FS name	# of SSUs	# of OSSs	# of Racks	# of LUNs
FS1	1	8	2	16
FS2	2	16	4	32
FS4	4	32	8	64

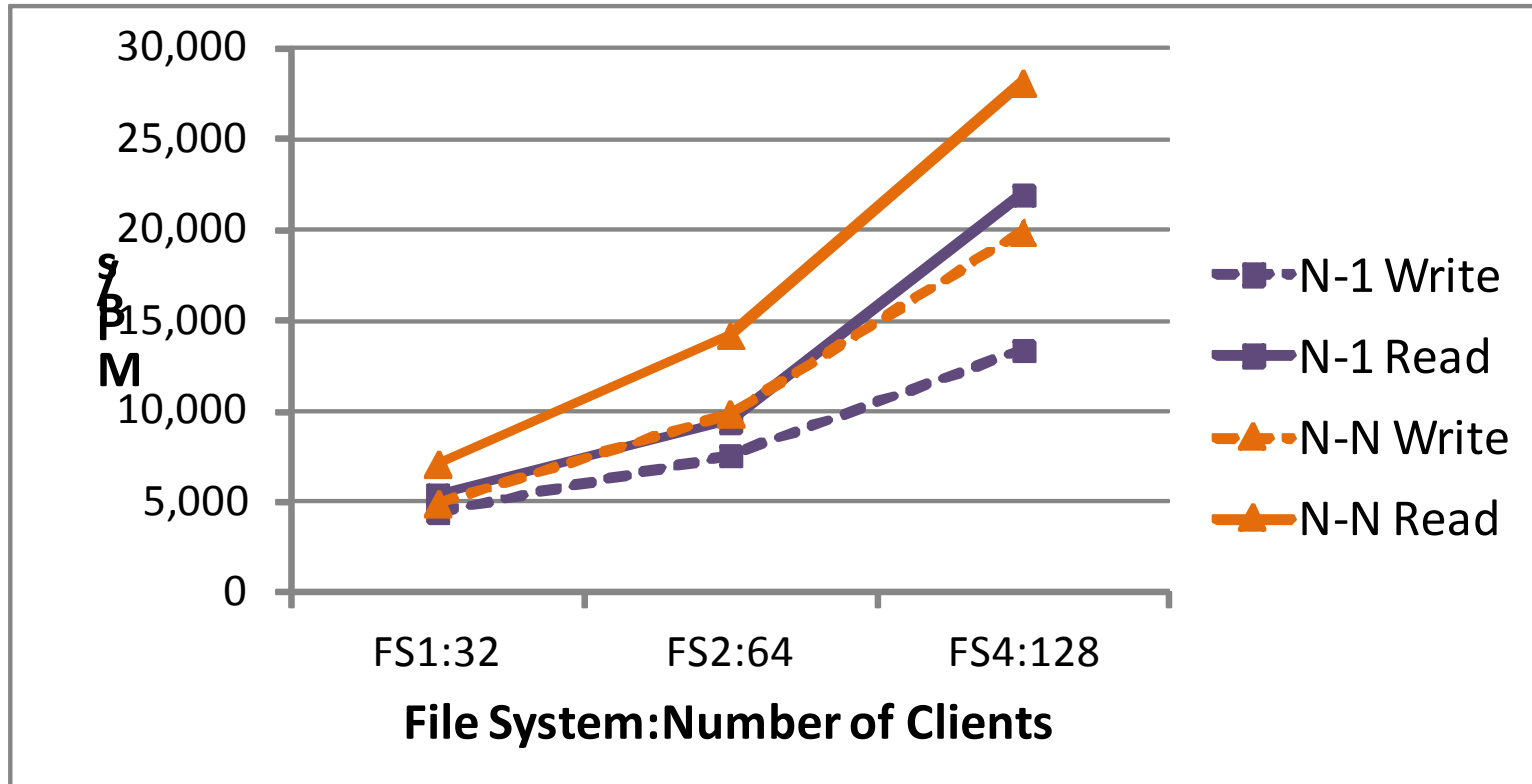
- If the SSU was a bottleneck in a Scenario test, then repeating the test on a file system with twice the number of SSUs should double the performance of the Scenario
- The XT4 had enough Lustre client nodes and sufficient network bandwidth that we could run tests against all three file systems simultaneously

# Client scalability of the three file systems



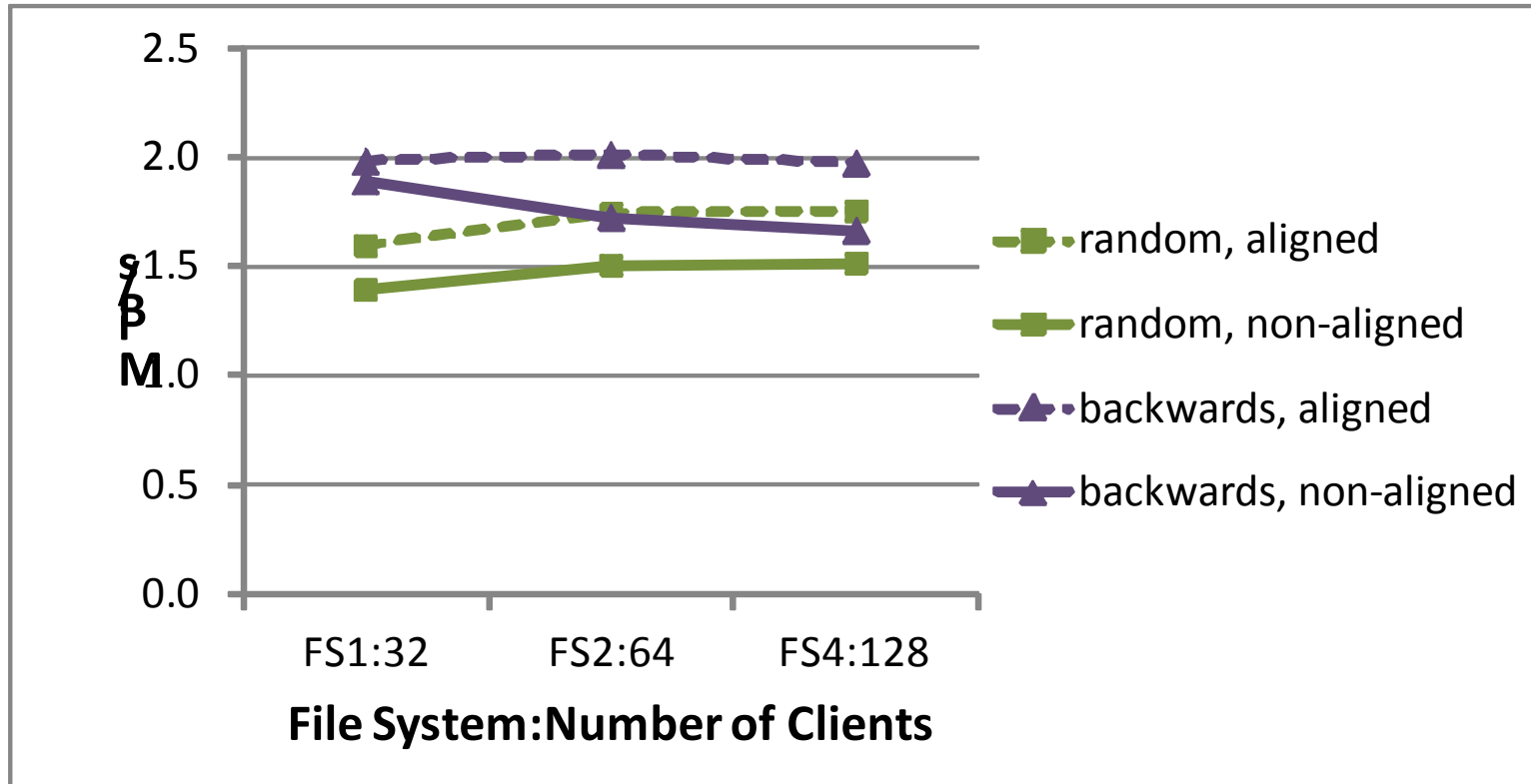
- Used Scenario 7 (N-N, sequential large I/Os) to find minimum client count to exercise each file system
- 4 clients per OST gave best performance

# Large I/O with Scenarios 5 (N-1) and 7 (N-N)



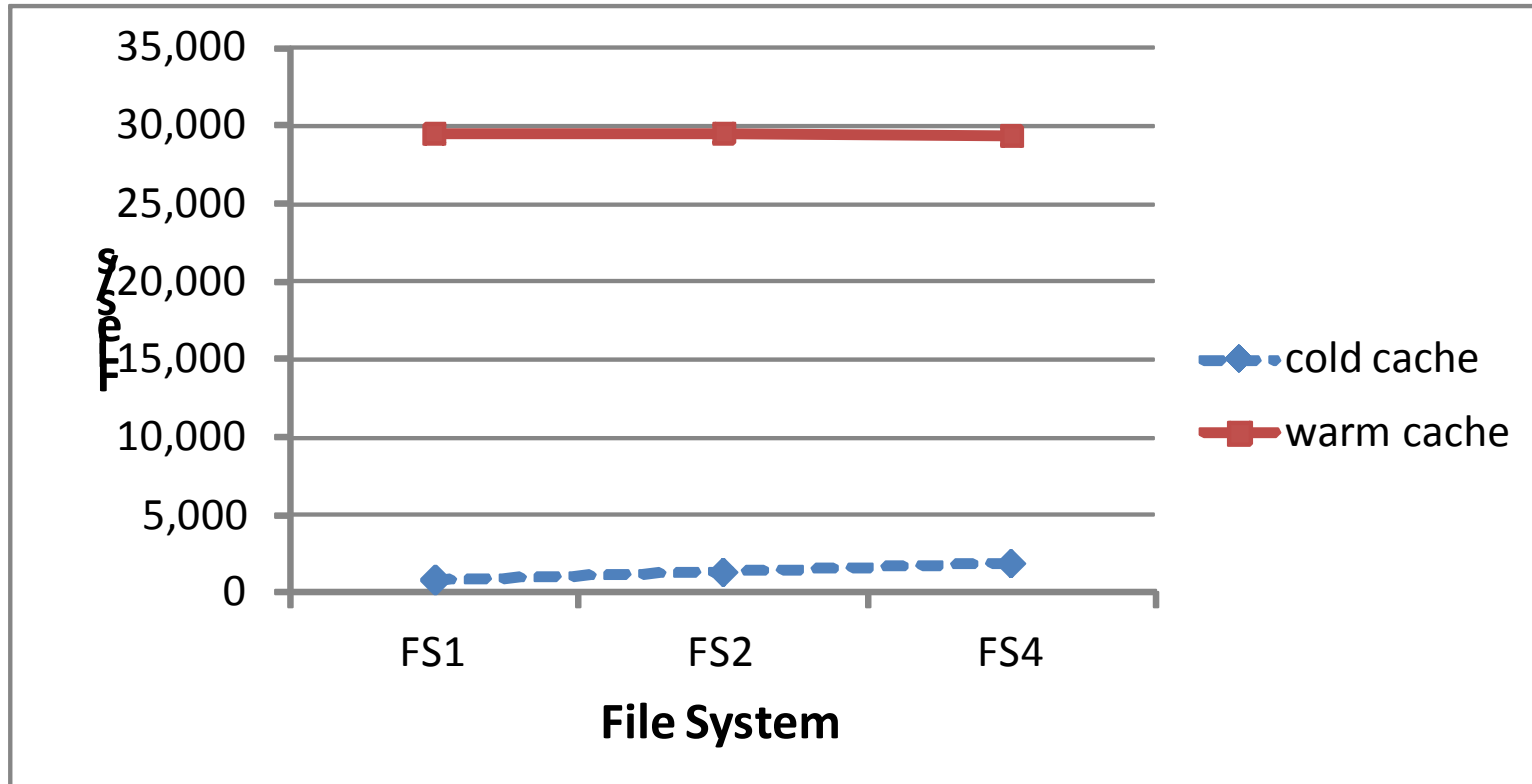
- N-N (scenario 7) performance was greater than N-1 (Scenario 5: shared file, segmented access)
- For both of these sequential workloads performance increases with file system size, though N-1 does not scale as well as N-N

# Random I/O with Scenario 13 (N-N)



- All three file systems showed extremely poor performance (<2 MiB/s) with the random I/O workloads
- There is no indication of performance scaling with increasing file system size

# Metadata performance with Scenario 9



- Lustre currently has a single metadata server, regardless of the size of the number of OSSs in the file system
- As a result, metadata performance shows no scaling as the capacity of the file systems increased



# Lessons Learned at ORNL

- Reducing test execution time
  - Test completions based on memory transfer size are indeterminate
  - Most tests reached a steady-state bandwidth or IOP rate long before the test completed the specified data transfer
  - As a result, Cray restructured its code to exit the test and report results after a user-specified time
- Creating a cold metadata cache
  - Scenarios 9-12 are to be run with both a cold and warm cache
  - Restarting the storage system to create a cold cache is too intrusive and too time consuming
  - Cray found that creating a dirty cache by running the metadata tests against a different copy of the directory tree was as effective as and gave similar results
- DARPA updated the Scenarios document with these notes and cleaned up confusing sections of the Scenario descriptions based on Cray's implementation

# Conclusions

- The important idea behind the HPCS Scenarios is the definition of scalable workloads that create a level playing field for comparing the capabilities and scalability of different storage systems
- Cray has demonstrated the scalability of its implementation and the effectiveness of the tests to expose the strengths and weaknesses of a storage subsystem
- Cray's Scenarios tests are not just for Lustre, but can be used to evaluate any other scalable file system
- Download the Scenarios document and tests from SourceForge : <http://hpcs-io.cray.com/>

# Thank You

- This material is based upon work supported by the Defense Advanced Research Projects Agency under its Agreement No. HR0011-07-9-0001. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the Defense Advanced Research Projects Agency.
- Our tests were performed at the Oak Ridge National Laboratory, which is managed by UT-Battelle, LLC. The author gratefully acknowledges the assistance of Buddy Bland, Don Maxwell, Galen Shipman and Sarp Oral of ORNL's National Center for Computational Science (NCCS); John Dawson, Mike Booth, and Ed Giesen of Routing Dynamics; Jeff Garlough of Cray's testing group; Tom Griffith and Dick Sandness of Cray's benchmarking group; and Jeff Becklehimer, Kim Kafka, and John Lewis from Cray's ORNL Support team.