HPCS I/O

ORNL LCE Scalability Summit

19 May 2009

John Carrier
HPCS I/O Lead
Cray, Inc.
carrier@cray.com

DARPA HPCS Program Award

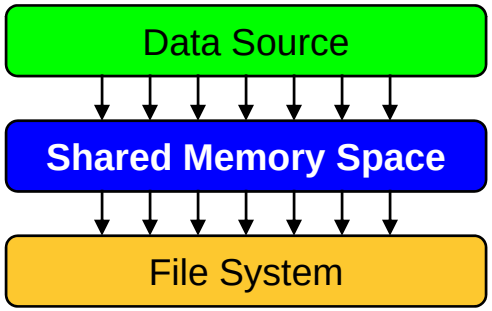
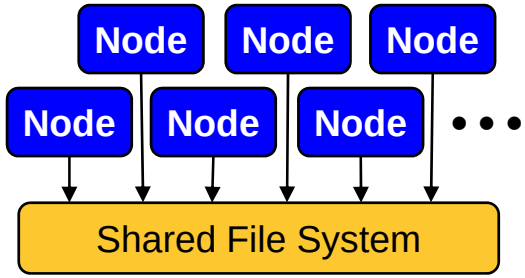
In November 2006, DARPA awarded Cray and IBM separate $250 million development contracts under its High Productivity Computing Systems (HPCS) program.

HPCS Goals:
Provide a new generation of economically viable high productivity computing systems for the national security and industrial user community in the 2010 timeframe

- **Performance** (time-to-solution):
  speed up critical applications by factors of 10 to 40
- **Programmability** (idea-to-first solution):
  reduce cost and time for developing application solutions
- **Portability**:
  insulate application software from system specifics
- **Robustness**:
  protect applications from hardware faults and system software errors

The result will be greater productivity (not just faster machines)

HPCS I/O Environments

Capture Environment	Parallel Environment
System dedicated to moving external data to a large shared memory space where an application will analyze the data streams, create files and then write data to disk	System with many client nodes connected to a single shared file system or name space. Both the data and metadata operations are important for efficient use.
	
Requirements ▪ Streaming I/O at 30 GB/sec ▪ 32K file creates per second	Requirements ▪ 30K nodes ▪ One trillion files in a single file system ▪ 10K metadata operations per second

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

Parallel Environment

5. Checkpoint/restart with large I/O requests
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

- DARPA requires that Cray demonstrate the scalability of its I/O solution using tests that *execute these I/O scenarios*
- *Scaling performance, rather than absolute throughput, is important to all scenarios*

Cray's HPCS I/O Goals

HPCS Goal	I/O Targets
Capacity	• 1 trillion files per file system • 10 billion files per directory • 100 PB system capacity • 1 PB single file size • >30k client nodes • 100,000 open files
Reliability	• End-to-end data integrity • No performance impact during rebuild
Performance	• 40,000 file creates/sec from a single client node • 10,000 directory listings/sec aggregate • 30GB/sec streaming data capture from a single client • 1.5 TB/sec aggregate I/O – file per process and shared

And, demonstrate file and storage system scalability using the HPCS I/O Scenarios!

Lustre and HPCS

- Cray partnered with Sun to meet our HPCS I/O Goals
  - Founded on the close working relationship Cray had with Cluster File Systems deploying Lustre on our XT platforms
  - HPCS extended the relationship for development of advanced features that are already impacting the Lustre roadmap

HPCS Goal	Lustre Solution
Capacity	• Lustre ZFS integration • Clustered Metadata (CMD)
Reliability	• End-to-end check-summing integrated with ZFS • ZFS rebuild performance improvements
Performance	• Clustered Metadata (CMD) • IO Channel Bonding

HPCS & Lustre Scalability

- Our goal today is to present an overview of the Lustre HPCS File System Design
  - A draft of the overview document will be available soon
  - Final detailed designs of the new features will be available later this summer after Sun completes its design SOW with Cray

- Next Talks
  - *"HPCS I/O Scenarios"*
    Henry Newman, *Instrumental*, DARPA HPCS

  - *"Lustre HPCS File System Design"*
    Andreas Dilger, *Sun*, Lustre Principal Engineer

THANK YOU