What is HPCS and How Does Impact I/O
May 19, 2009

Henry Newman

Instrumental Inc/CTO

hsn@instrumental.com

Agenda

- What is HPCS
  - And what is it not
- I/O Survey of Requirements
  - What applications do but also what they need
- HPCS Scenarios
  - Not 12 but 14 Scenarios
- Why this is all important

High Productivity Computing Systems

Critical to National Security

- Develop a **new generation** of **economically viable** high productivity computing systems for national security and industrial user communities (2011)
- **Ensure U.S. lead, dominance, and control** in this critical technology

Phase III Vendors: CRAY IBM

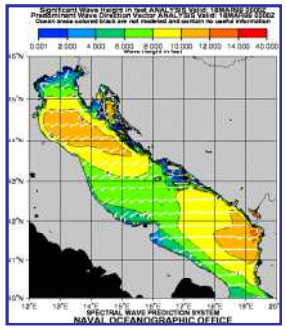
Mission Partners:

Impact:

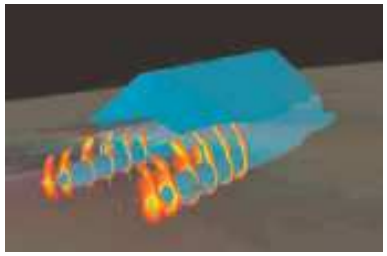
- **Performance** (time-to-solution): speedup by **10X to 40X**
- **Programmability** (idea-to-first-solution): dramatically reduce cost and development time
- **Portability** (transparency): insulate software from system
- **Robustness** (reliability): continue operating in the presence of localized hardware failure, contain the impact of software defects, and minimize likelihood of operator error

Applications:

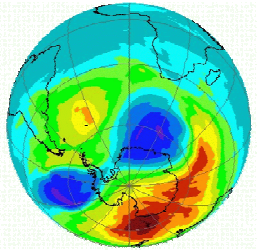
Weather Prediction



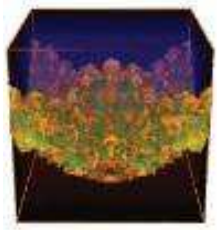
Ocean/wave Forecasting



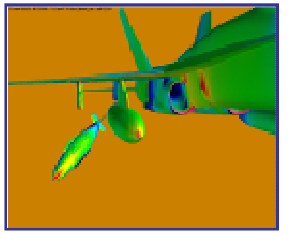
Ship Design



Climate Modeling



Nuclear Stockpile Stewardship



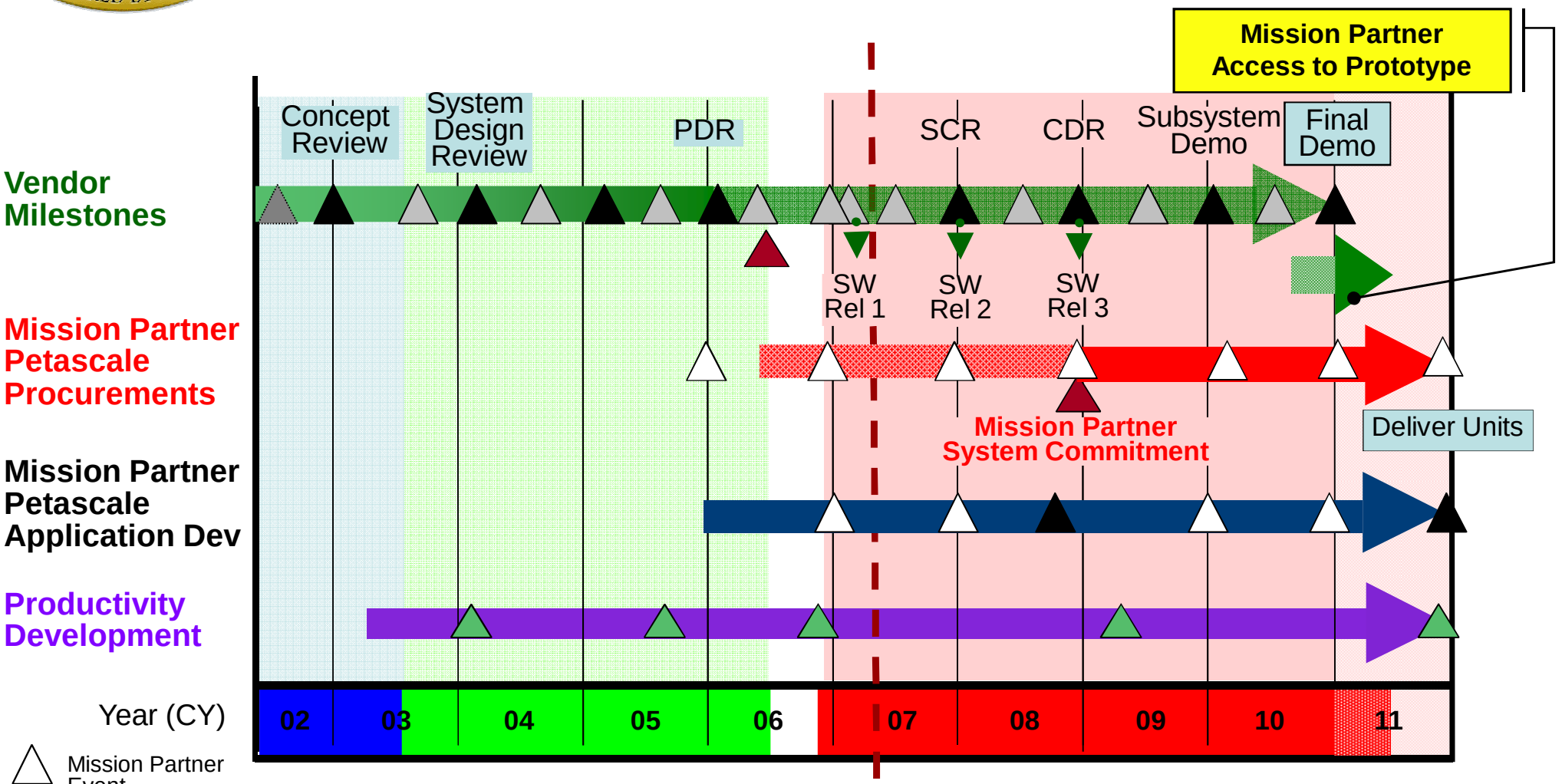
Weapons Integration

Fill the Critical DoD Need for:
Operational weather and ocean forecasting, weapons design and analysis, airborne contaminant modeling, intelligence/surveillance/reconnaissance, cryptanalysis

Phase III

- 2 competing vendors
- Complete HPCS system design and development
- Continue high productivity language work
  - Chapel
  - X10
- Demonstrate prototypes
  - Deliver systems that work and can be tested
- Assess system productivity

HPCS Program Phases I - III



May 19, 2009

Cray's Motivation for Cascade

Why are HPC machines unproductive?

- Difficult to *write* parallel code (e.g.: MPI)
  - Major burden for computational scientists
- Lack of programming tools to *understand* program behavior
  - Conventional models break with scale and complexity
- Time spent trying to modify code to fit *machine's* characteristics
  - For example, cluster machines have relatively low bandwidth between processors, and can't directly access global memory…
  - As a result, programmers try hard to reduce communication, and have to bundle communication up in messages instead of simply accessing shared memory

If the machine doesn't match your code's attributes, it makes the programming job much more difficult.

And codes vary significantly in their requirements…

IBM's Technical Approach

- Unprecedented focus on productivity
  - Hardware/software co-design focused on improving system productivity by more than an order of magnitude and significantly expanding the number of productive users in a petascale environment
  - Application development: programming models, languages, tools
  - Administrative: automation, simplicity, ease of use
- A holistic approach that encompasses all the critical elements of supercomputer system architecture and design (Hardware and Software)
  - Processors, Caches, Memory subsystem, networking, storage, Operating systems, parallel/cluster file systems, programming models, application development environment, compilers, tools for debugging and performance tuning, libraries, schedulers, checkpoint-restart, high availability software, systems management
- Balanced system architecture and design
- Leverage IBM leadership in UNIX systems to provide petascale systems on commercially viable technology
  - POWER, AIX/Linux, ISVs, …
- Focused effort on significantly enhancing the sustained performance experienced by applications at scale
  - Maximize compute time spent on productive computation by minimization of overhead
  - Cluster network optimized for common communication patterns (collective, overlap of communication and computation…)
- Tools to identify and correct load imbalance
- General Purpose, flexible operating environment to address a large class of supercomputing applications
- Significant reductions in complexity and cost for large scale supercomputing infrastructure

HPCS is NOT

- One-off system
  - Development and demonstration program
- Meeting one set of requirements
  - Mission Partners have various requirements
  - Phase III solicitation requested several configurations
- Available only at petascale
  - Single cabinet to very large system configurations
- Using only new languages
  - C
  - Fortran
  - PGAS (UPC, CAF)
  - Other
  - Varies by vendor
- HPCS is not just about large simulations but other problems such as knowledge discover

Agenda

- What is HPCS
  - And what is it not
- I/O Survey of Requirements
  - What applications do but also what they need
- HPCS Scenarios
  - Not 12 but 14 Scenarios
- Why this is all important

Survey Goals

- Characterize the requirements for I/O for petascale systems based on Mission Partner (MP) input
  - Bandwidth
    - Write and read
  - IOPS
    - Metadata for file systems
    - Knowledge formation
- Look at 2010 requirements based on 2004/5 applications
  - Some of the guesses were fair some where not good
- The real requirements today are far different than were provided and documented in 2004/2005
  - Everyone makes mistakes
- Additional requirements were provided to Phase 3 Vendors in 2008

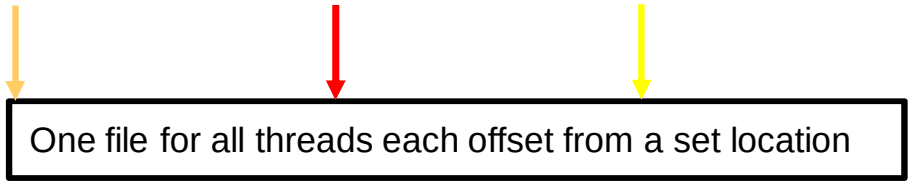
Phase II Mission Partner Requirements

- I/O from nodes to the file system 30,000
  - Preferred a single file
- 32,000 open creates per second
  - Serious challenge at the time for shared file systems
    - Still a significant challenge for the file system and hardware configuration
- 60 GB/sec full duplex from a single node
  - This has been the source of confusion
    - The real issue is for data capture into a single shared memory destination
- 1 Trillion files in a file system
  - Significant challenge given that 1 Billion had not been reached
- Remember that the P in HPCS stands for Productivity not performance

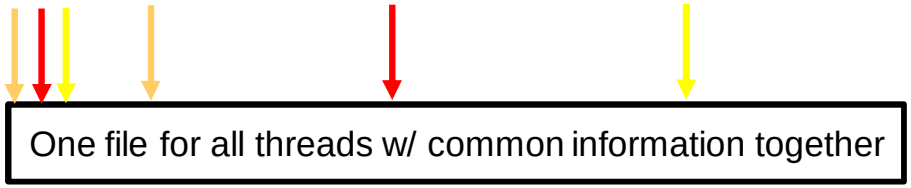
Application I/O Types

This is what most applications do

Writing I/O to a single file can be much slower on some file systems than using 1 file per This method is not typically used in applications given its poor performance, but is a far simpler programming method and it is much easier to deal with one file. Alignment for each file system, operating system and hardware allocations are critical for performance for the whole path



In this case, data from each node is placed together in a single file with offsets such that the data can accessed by category by another application (e.g., visualization software). Then offsets are repeated for each node for the rest of the data within the single file. This is a very difficult problem for file systems because in the beginning you will do small block I/O to the file from each node and then large block I/O to the file from each node. Applications programmers would like to be able to do this but it is too slow currently on all file systems. One vendor is trying to address this



2008 MP Requirements Update

- Higher Bandwidth
  - Both Streaming and IOPs
- Large file names
  - 255 characters or greater
- More files created per/sec
  - 40,000 not 32,000
- Billion files in a directory
  - More than 1 Trillion files in a file system
- PB size files
- Up time 99.99% and 99.97% no degradation of performance
- T10 or equivalent for reliability of data
  - 10E28 bits for detection and 10E21 for correction

Agenda

- What is HPCS
  - And what is it not
- I/O Survey of Requirements
  - What applications do but also what they need
- HPCS Scenarios
  - Not 12 but 14 Scenarios
- Why this is all important

Are the HPCS Scenarios Benchmarks?

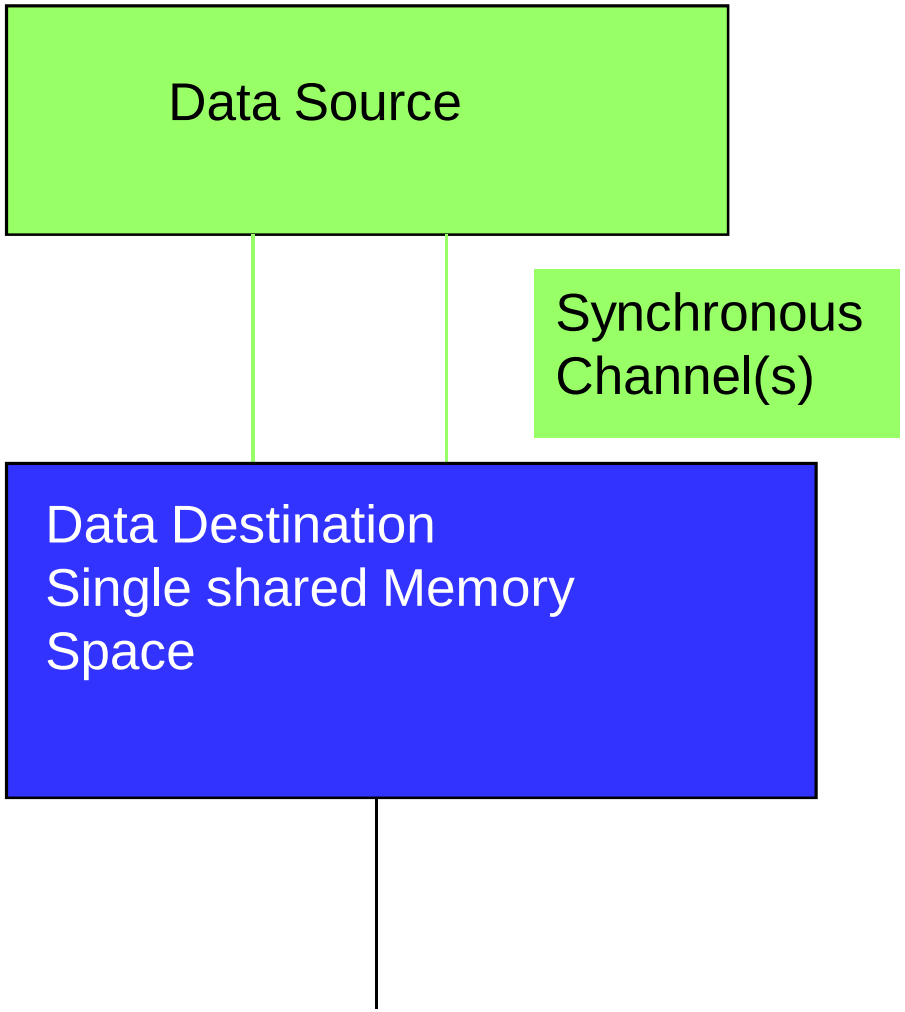
- No and they were never designed to be benchmarks or used as benchmarks
  - As written they could not be used as benchmarks
- The goal was to give examples of types of problems that MP face and give the HPCS vendors some examples to allow them to show scaling
  - Remember HPCS is about scalability not just performance
  - And potentially new techniques for I/O
- Could they be used as benchmarks?
  - Maybe, but not as written as they are designed to show I/O datapath scaling
  - It would require major rewrite on the part of the organization doing the procurement

The Scenarios

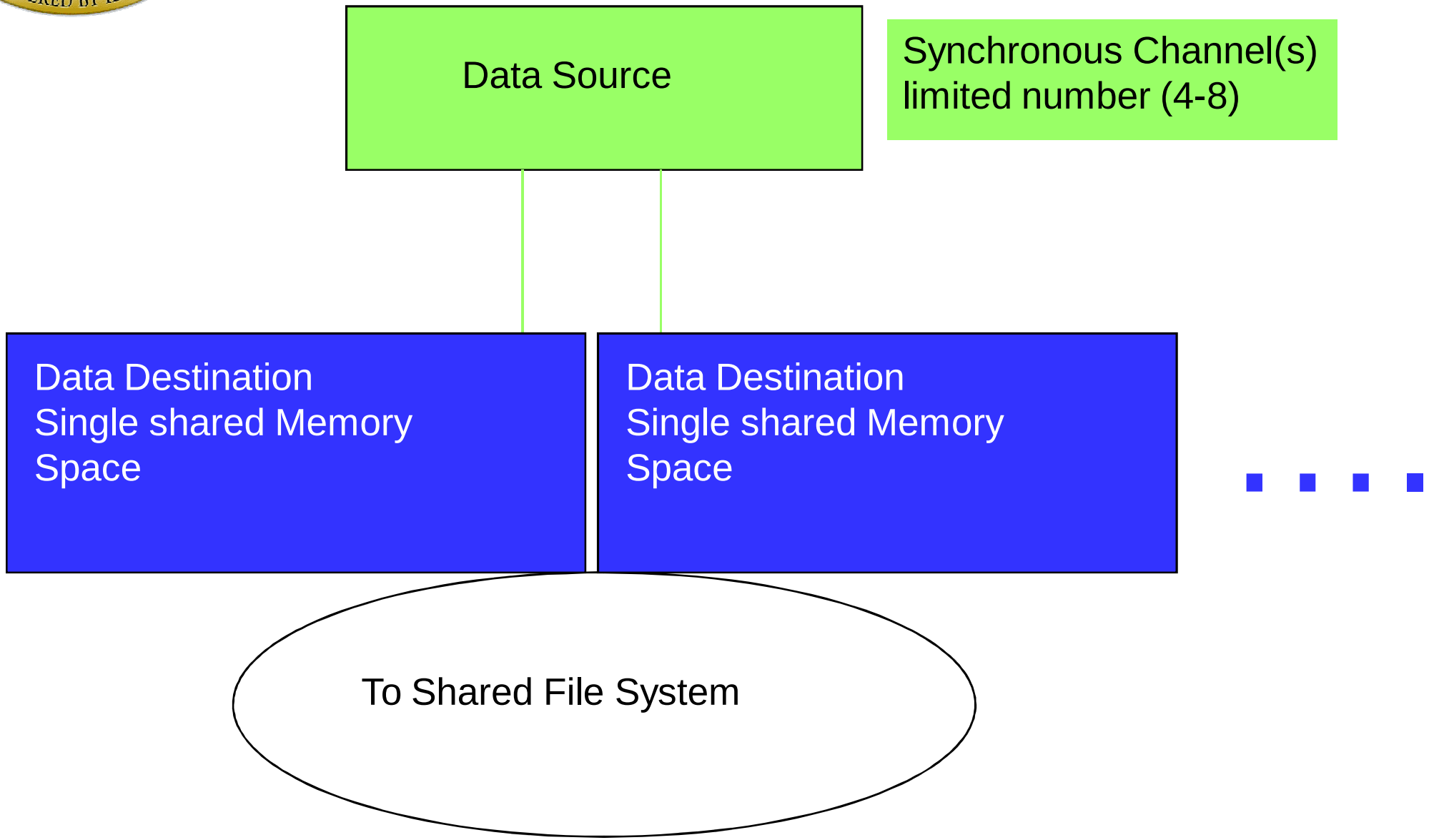
1. Single stream with large data blocks operating in half duplex mode
2. Single stream with large data blocks operating in full duplex mode
3. Multiple streams with large data blocks operating in full duplex mode
4. Extreme file creation rates
5. Checkpoint/restart with large I/O requests
6. Checkpoint/restart 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 (1) process
12. Random stat() system call to files in the file system - multiple processes
13. Small block random I/O to multiple files
14. Small block random I/O to a single file

Scenarios 3, 4, 5 and 14 are considered the most important given the state of current technology and the degree of difficulty.

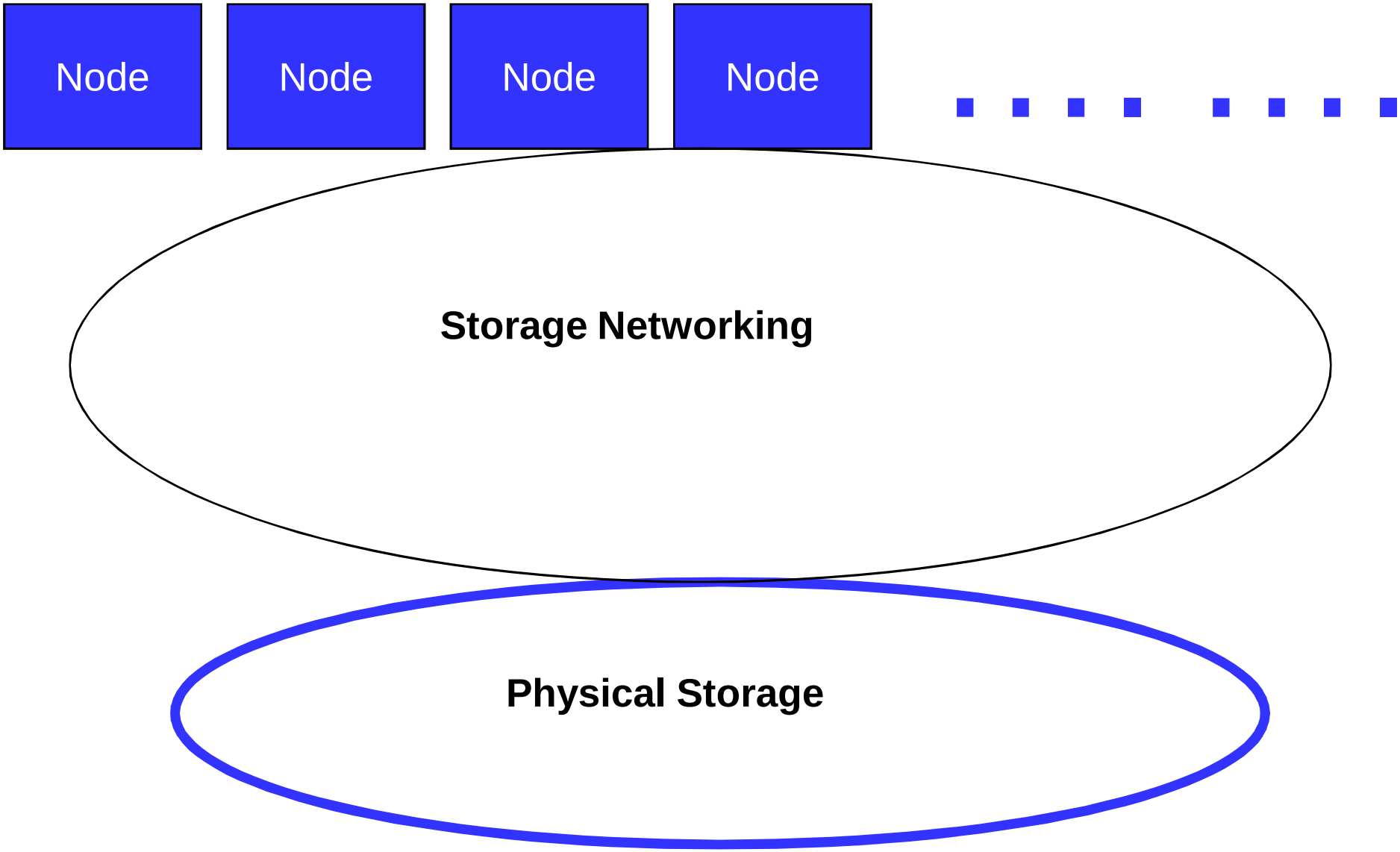
Capture Environment Scenarios 1-2,4



Capture Environment Scenario 3



Parallel Environment Scenarios 5-14



Agenda

- What is HPCS
  - And what is it not
- I/O Survey of Requirements
  - What applications do but also what they need
- HPCS Scenarios
  - Not 12 but 14 Scenarios
- Why this is all important

Importance of I/O

- Scaling for I/O is abysmal
  - No standards for shared file systems
  - Limits imposed by POSIX with no changes planned
  - Limited improvement in channel performance
    - FW-SCSI was 20 MB/sec in 1994, FC today is 800 MB/sec-40x
  - Disk density improvements are poor but access performance is horrible
    - Only 8x seek and latency change in 30+ years
- Defensive I/O is needed given poor reliability of systems
  - Reliability is not improving, but systems are getting larger
  - Poor I/O performance impacts system efficiency given the number of checkpoints needed
- Knowledge discovery requires at least today small block random
  - No current file system, or storage infrastructure is designed to address this
    - Current RAID controllers and HBA/HCAs do not have the command queue to address this

Real Applications' I/O

- One of the problems is few application benchmarks require real I/O
  - It just takes to long
  - Vendors cannot afford the machine time
  - Often times running on a pristine system is not useful
    - Fragmentation of data and metadata for some file systems in dramatically impact performance
- I/O tests such as IOR and others often are not run in the same way that applications are run
  - Applications use memory and memory bandwidth
  - Some application do asynchronous I/O
    - Weather, climate and IC community especially
  - Is IOR run the way current systems are used or the way that future systems need/should be used?
    - Are planned application modifications for single files, alignment taken into account?

Final Thoughts

- A file per thread is dramatically impacting scaling and performance
  - Historically shared file systems did better with a file per thread rather than 1 file for all threads
    - Most shared file systems do not have this issue, but the applications have not changed
  - HPCMP archive growth rate last year for file count was 70%, but 40% for data
    - This directly translates to file counts on HPC systems
- I/O is often forgotten part of HPC and other applications such as knowledge formation
  - Remember that defensive I/O is only part of the requirement for just some application domains
    - I/O is more than large block sequential writes hopefully read never
  - Small block random dominate some application domains
    - That is why these were added to the HPCS Scenarios

