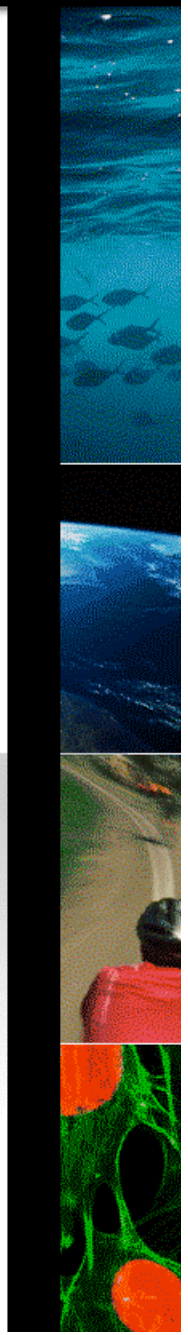
The Supercomputer Company

BlackWidow Software

Don Mason

Cray Inc

CUG 2006



Talking Points

- Cray BlackWidow software objectives
- Hardware capabilities utilized by software
- Cray BlackWidow OS architecture
- Software features
- Application Level Placement Scheduler (ALPS)
- Cray platform software commonality
- Implementation tools and status

BlackWidow Software Objectives

- Deliver the highest possible performance from the the Cray BlackWidow hardware to customer applications
- Provide a natural follow on for Cray X1 and Cray X1E customers
- Significantly move towards common software with other Cray products, especially the Cray XT
- Improve system reliability over previous products

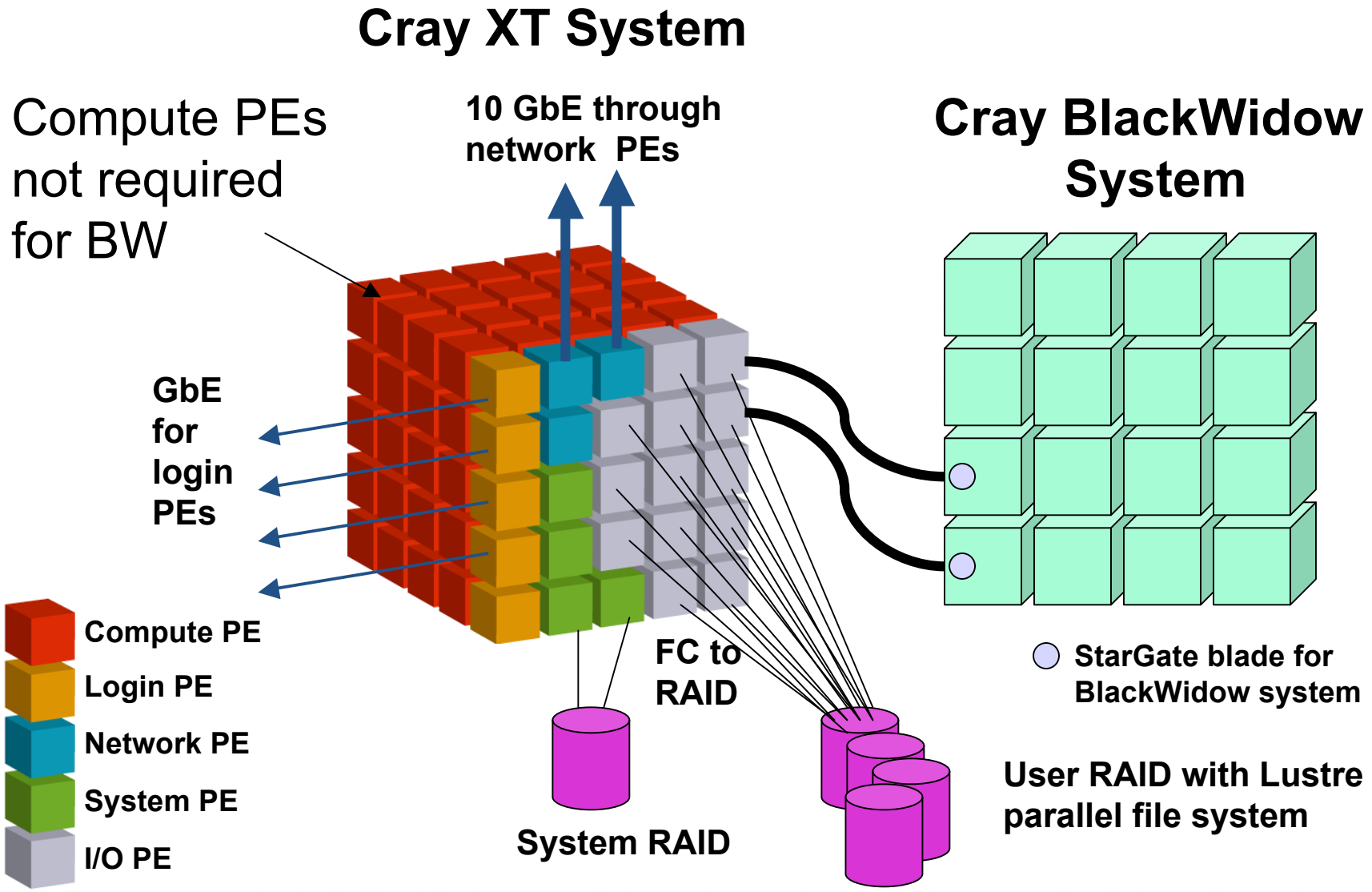
New Hardware Capabilities

- A globally addressable memory using a fat tree topology
  - Large pages linked together to provide a parallel application with one address space (similar to the Cray X1/X1E, but more flexible – nodes do not have to be contiguous)
- A single node consists of 4 8-pipe vector processors. A Cray BlackWidow cabinet has 32 nodes on 16 boards for a total of 128 processors
  - No “MSP” (Multi-streaming processor) as on the Cray X1/X1E
  - OpenMP supported across a 4 processor node
  - New processor features such as vector atomic operations that improve the compilers ability to generate high performance code
  - Each node treated as an SMP by the OS

Hardware Capabilities (cont.)

- Error protection available to software
  - Memory protection to keep a rogue kernel from writing into the space of another kernel
  - More information available to OS on hardware faults
  - Graduation timeout provides way to detect errors before impacting entire system
  - Better multi-bit error handling
- High bandwidth Cray XT I/O
  - Cray XT service nodes provide the I/O for BlackWidow.
  - A StarGate bridge blade provides the interface between BlackWidow and a Cray XT system

I/O and Networking



OS Architecture

- **Linux 2.6 Kernel**
- **Each node is an independent instance; a Linux SMP**
- **Root file system for commands (limited) and libraries**
- **Few changes in hardware independent part of Linux kernel**

Blackwidow node

Cray XT ⟺ BW Bridge

ALPS launch to BW

- Standard SUSE distribution and environment.
- BlackWidow compilation environment

Login node (XT)

Software Features

System Software

- Linux Kernel
- ALPS (Application Level Placement Scheduler)
  - Batch interfaces to PBS and other schedulers
- Lustre parallel file system
- *Mazama* System Administration
- Process, Project and Job Accounting
- XT Networking and I/O

Programming Environment

- Compilers
  - C/C++ and UPC
  - gcc
  - Fortran and Co-Array
- MPICH2 MPI and shmem
- OpenMP support
- Scientific and math libraries
- TotalView Debugger
- Performance Analysis
  - Cray PAT
  - Cray Apprentice^2

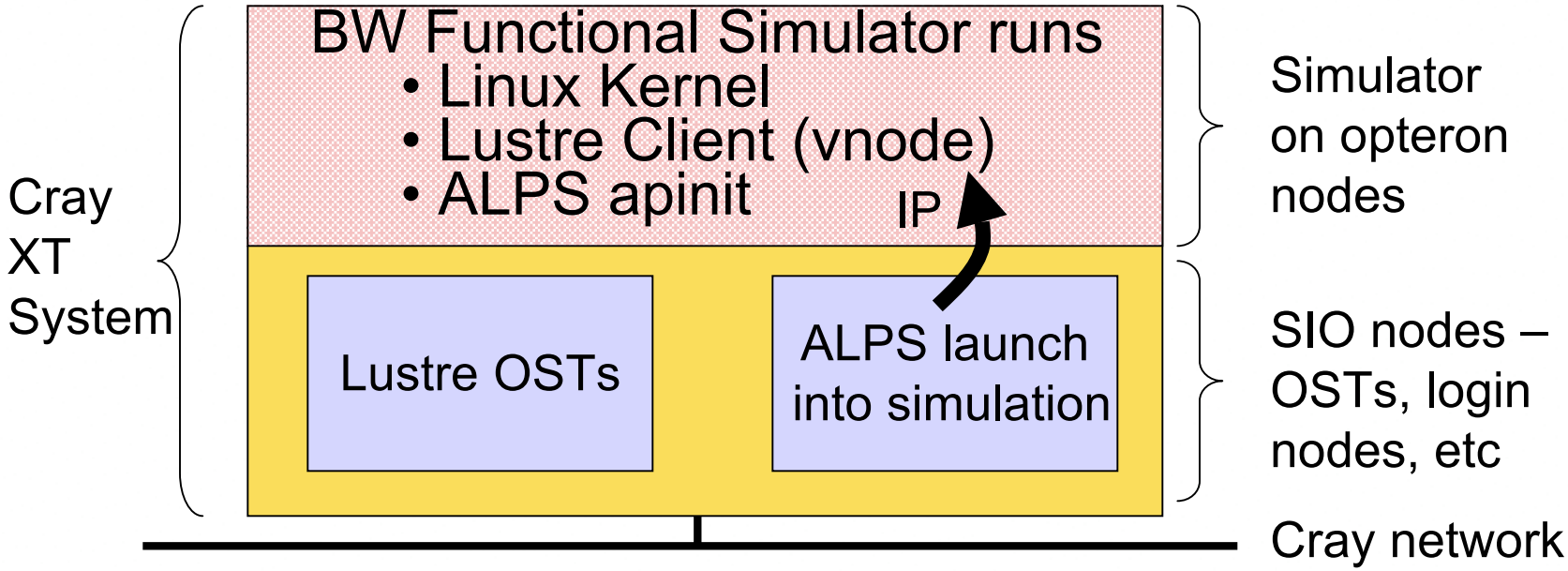
Application Level Placement Scheduler (ALPS)

- Encapsulates “job” functionality for system scalability
  - Placement and launch of applications
  - Management of stdin and stdout
  - Handles signals for application
  - Interacts with a batch scheduler via an interface (basil) that allows different schedulers, such as PBSPro to work with ALPS
- Early measurements of launch and exit times on an XT (running a test version of BW Linux on the compute nodes) are significantly faster than on the Cray X1

Software Commonality - 2007

- Moving quickly toward high commonality between Cray XT and BlackWidow systems
  - At first BlackWidow release
    - Common I/O subsystem with XT, Eldorado and BlackWidow
    - Lustre file system with file sharing
    - ALPS placement (XT Linux systems only)
    - Mazama System Administration for BlackWidow and part of XT administration functionality
    - Common performance analysis tools and debuggers
  - In the future
    - Mazama supports all platforms
    - Common Hardware Supervisory System (HSS)

Implementation Tools



- Two powerful simulators for software – functional simulator and application simulator
  - Functional simulator supports boot of OS, execution of applications with BlackWidow instruction set
  - Application simulator performance accurate for validation of BlackWidow compiler optimizations

Implementation Status

- Following plan to have all functionality complete when hardware first powers up
  - All programming environment functionality complete (C, C++, UPC, Fortran, CAF, MPI, shmem, libraries) and extensively tested
  - SUSE 9 Linux kernel running in simulation for over two years
  - ALPS functionality complete
  - Hundreds of test suites run with high pass rates
  - StarGate Blade driver simulated; portals running over driver
- Starting hardware checkout

Questions?