Cray RS Programming Environment

Gail Alverson

Cray Inc.

Red Storm

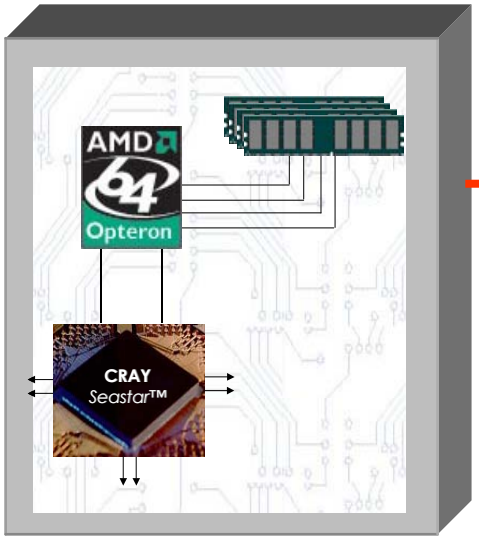
Red Storm is a supercomputer system leveraging over 10,000 AMD Opteron™ processors connected by an innovative high speed, high bandwidth 3D mesh interconnect designed by Cray

CUG 2004

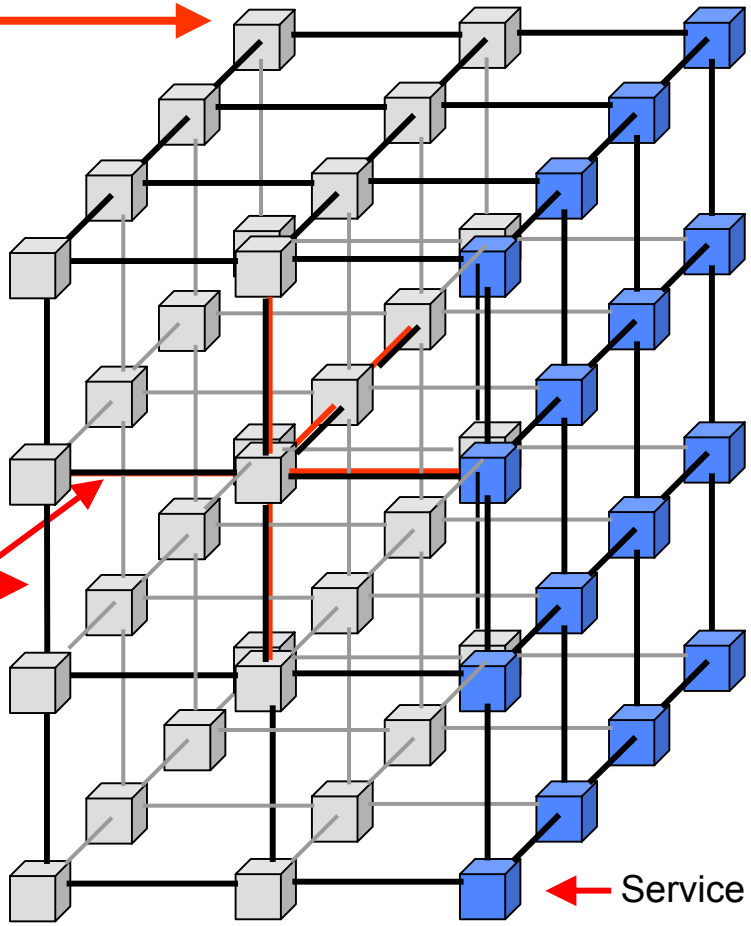
Cray RS

Cray RS is the production version of Red Storm, based on the same software and hardware

Cray RS architecture



3D Mesh (Torus)



High Speed Communication Links

Compute

Catamount LWK

Service & IO

Linux

Aggregate Sustained Bandwidth

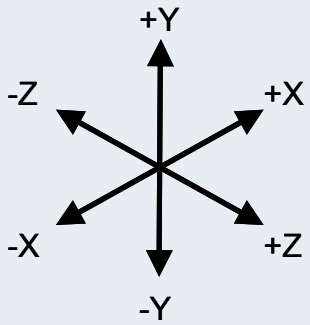
- 120 TB/s

Bisection Bandwidth

- 1.5 TB/s bi-directional

MPI Latency

- 2 us Nearest neighbor
- 5 us Furthest point



CUG 2004

Software

- **Operating Systems**
  - LINUX on service and I/O nodes
  - Catamount LWK on compute nodes
  - Portals ipc
- **Run-Time System**
  - Job launch: yod
  - Node allocator
  - Logarithmic loader
  - Batch system – PBS Pro
- **Programming Environment**
  - **PGI Compilers - Fortran, C, C++**
  - **Libraries - MPI, I/O, Math, MPI-2, SHMEM**
  - **Showmesh**
  - **Debugger - TotalView**
  - **Performance Monitor**
- **High Performance File Systems**
  - Lustre
- **System Mgmt and Admin**
  - Accounting
  - Red Storm Management System
  - RSMS Graphical User Interface

Sandia National Laboratories

ETNUS TOTALVIEW

The Portland Group Compiler Technology

SuSE

Pioneering Science and Technology

ARGONNE NATIONAL LABORATORY

A U.S. Department of Energy laboratory operated by The University of Chicago

CUG 2004

Focus on an integrated PE

- **A major goal of Red Storm PE is to provide an integrated set of tools.**
- **Compiler example:**
  - RS compile drivers use PGI compilers, pull in correct libraries and set correct flags for the RS environment
  - Programs can be a mix of C, C++ and Fortran90
  - RS libraries correctly link and run with PGI programs
  - TotalView can debug PGI compiled programs
- **Cray provides first line support for the these tools, and has an agreement with the vendors for greater support and updates**

CUG 2004

A typical user session

- **Initialize environment**

```
% module load PrgEnv
```

- **Compile**

```
% rsf90 pingpong.f -o pingpong
```

- **Run**

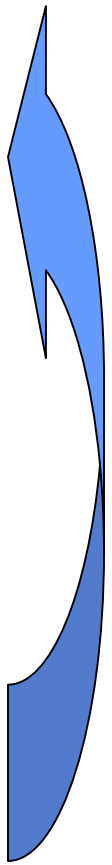
```
% yod -np 2  pingpong
% showmesh
```

- **Debug**

```
% totalview yod -a pingpong
```

- **Evaluate performance**

```
% perftool pingpong.trace
```



Edit

CUG 2004

First, load the PE environment

- As with other Cray products, Cray RS uses the *Modules* package to control the software versions used by commands

```
% module load PrgEnv
```

  Loads the default versions of the compiler, libraries, and tools

```
% module switch mpich2/current mpich2/new
```

  Switches in the new mpi2 library so that rsf90 picks it up automatically

- Modules eases updates to new software as it allows old and new to be installed concurrently

CUG 2004

Then start compiling …

```
% rsf90 -v pingpong.f -o pingpong

mpif90 for

pgf901 pingpong.f -static -lpgf90 -
lpgf90_rpm1 -lpgf902 -lpgf90rtl -lpgftnrtl -
lc -lnspgc -lpgc -lmpich -lshmem -lsysio -
llus -lcatamount -lportals -lc -lm -lacml [-
lscalapack -lsuperlu]

PGF90 5.0-2: compilation successful
```

Compilers from PGI

STMicroelectronics

The Portland Group
Compiler Technology

- PGI is now part of STMicroelectronics, one of the largest semiconductor companies in the nation
- Performance improvements flow to/from the HPC products and the embedded products
- Provided first optimized F90 x86-64 compiler and performance increases with each release
- Support F90, C, and C++
- Full F95 scheduled for release 6.0, June 2004
- C99 features are migrating from the C compilers for embedded systems

Message passing with MPICH-2

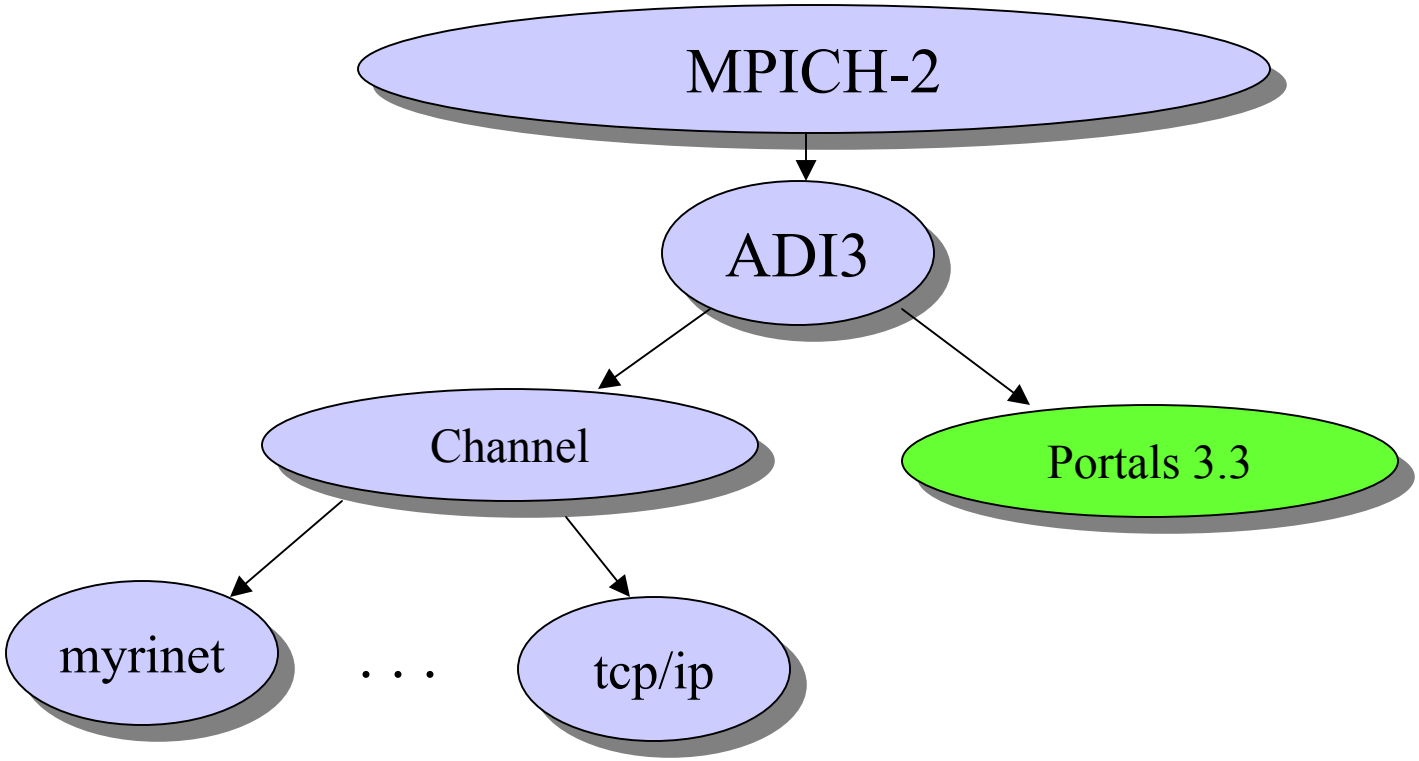
Pioneering Science and Technology

ARGONNE NATIONAL LABORATORY
A U.S. Department of Energy laboratory operated by The University of Chicago

- Cray RS provides an MPI-2 message passing library based on MPICH-2 from ANL
- First release MPICH-2 was November 2002
- Current release 0.96p2 has all MPI-2 features except passive one-sided communication and dynamic process creation (which is not supported in Cray RS)
- Argonne researchers are committed to support Cray (Rusty Lusk, Bill Gropp)

Porting MPICH-2 to portals

- **Main development effort for using MPICH-2 was to provide a Portals Abstract Device Layer**



Preparing for SHMEM

- The Cray single sided communication library, ***SHMEM***, is being developed for Cray RS and is targeted for release Q2/05

- Our focus is first on those calls that can be implemented efficiently on this system, for example:
  - Symmetric data objects (not asymmetric)
  - Standard put, get, barrier (not strided, indexed operations

- Implementing some low level operations in the firmware can lead to performance payoffs

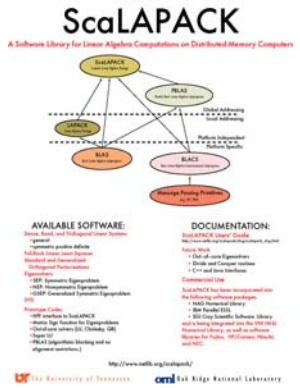
CUG 2004

Using a glibc-based libc

- Catamount libc is based on gnu libc
- Many functions execute locally on the compute node, however Catamount is MICROKERNEL built for scalability and inasmuch:
  - Offloads some OS calls, in particular, I/O, to service nodes
  - Does not support some OS calls, such as threads, new process creation (fork, exec), sockets, system call

Supporting scientific computing

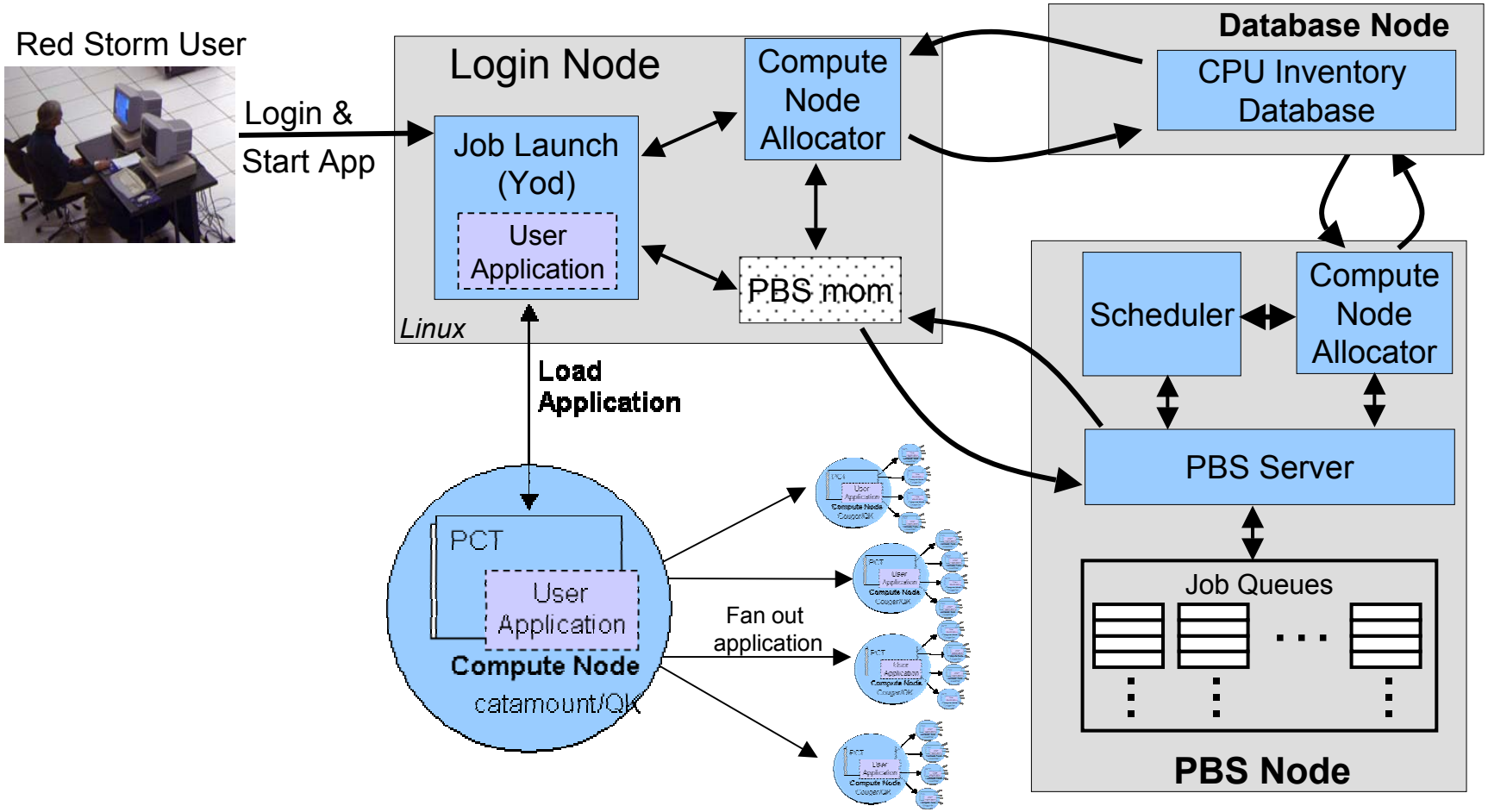
- **A suite of scientific libraries are available with Cray RS:**
  - **ACML: AMD Core Math Library**
    - **Levels 1, 2, and 3 BLAS**
    - **LAPACK**
    - **FFT's in single-, double-, single-complex and double-complex data types**
  - **ScaLAPACK: Distributed memory LAPACK, based on BLACS**
  - **SuperLU: Distributed memory sparse solver**



CUG 2004

Now you've built your program…

- **Launch it with "Yod" (interactively or via PBS)**



And monitor it with showmesh

- Showmesh displays the status of each node in the system, and presents it in a 2-D text display
- Widely used on ASCI Red to give a status snapshot

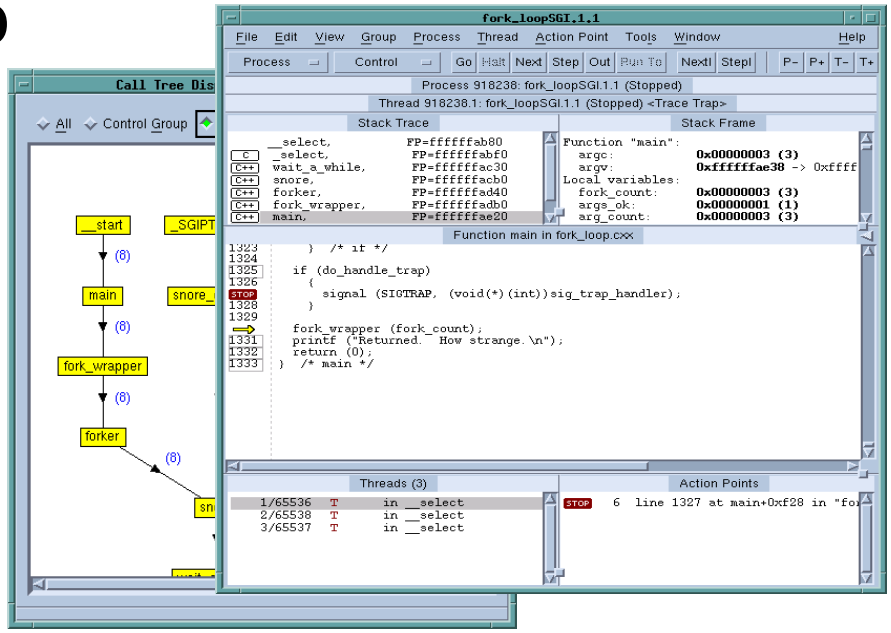
```
Cabinet 0
Node-> 01234567
Row 0   aaaabb||
Row 1   aaaa||||


Available compute nodes: 0 interactive, 6 batch
Job ID User       Size   Start          yod command line
--- -- ----       ----   ------         ----------------
a    8  khubert     8     Oct  8 17:26:33   yod -np 8 app
b    6  u3186       2     Oct  6 11:41:17   yod -np 2 app
```

What, a bug?

- Cray is working with Etnus to provide TotalView on Red Storm and Cray RS
- TotalView provides source level debugging of mpi programs; it can debug up to 1024 RS processes, compiled with PGI C, C++, F90

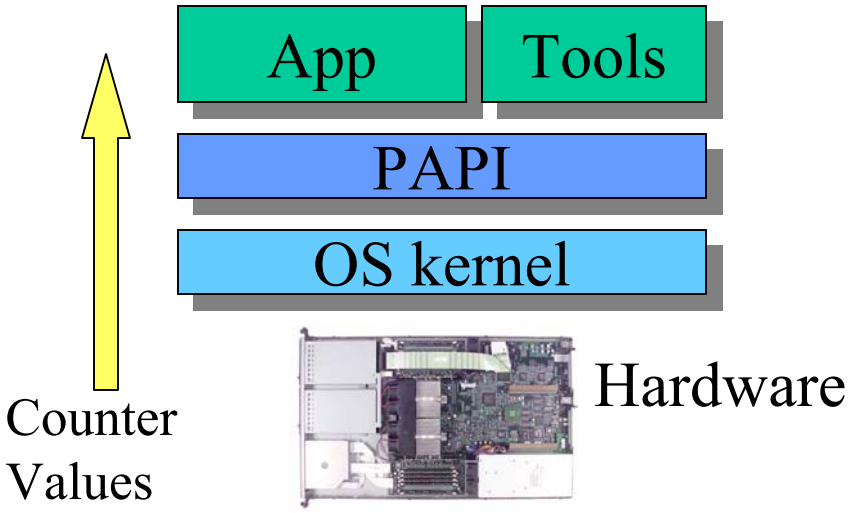
Cool MPI features of TotalView

- Automatically acquires processes as they start
- Offers barrier breakpoints for synchronizing parallel tasks
- Work with a single process, all processes or a user-defined group
- Laminated variables



Finally, the fun stuff: Tuning!

- One approach is to look at data from hardware counters
- Red Storm provides the PAPI – Performance Application Programming Interface – developed at UTennessee (Jack Dongarra ICS group)



Using PAPI

- **You can use PAPI directly:**

```
PAPI_create_eventset(&EventSet);
PAPI_add_event(&EventSet, PAPI_FP_INS); // flops
PAPI_start(EventSet); …
PAPI_read(EventSet, v1);
```

- **Or you can download and run tools that are based on PAPI:**
  - **TAU, Tuning and analysis utilities (U of Oregon)**
  - **Pablo, Performance visualizer (U of Illinois)**
  - **vProf, Visual profiler (Curtis Jansen, Sandia)**
  - **PSRun (U of Illinois), HPC Suite (Rice)**
  - **many others, both public and commercial**

CUG 2004

Performance tuning MPI

- Another tuning angle is to look at MPI communication patterns and potential bottlenecks
- Intel is working on an Extended Memory 64 Technology version of VT and providing a prototype version for Sandia Red Storm
- Cray is considering several options for RS, including a new MPI GUI for PAT that is under development and could be connected to an MPICH2 trace package
- Stay tuned…!

CUG 2004

Summary of Cray RS PE

Compiler	PGI cc, c++, f90
MPI	MPICH-2 over portals
Libc	glibc-based
Scientific libs	ACML, ScaLAPACK, BLACS, SuperLU
Job launch	Yod
Job monitoring	Showmesh
Debugger	TotalView
HW counter access	PAPI library
MPI performance	TBD