Eos Overview

Eos Composition

Eos Access and Billing

Eos Queue Policy

Eos Compilers, PrgEnv, and Software

Documentation
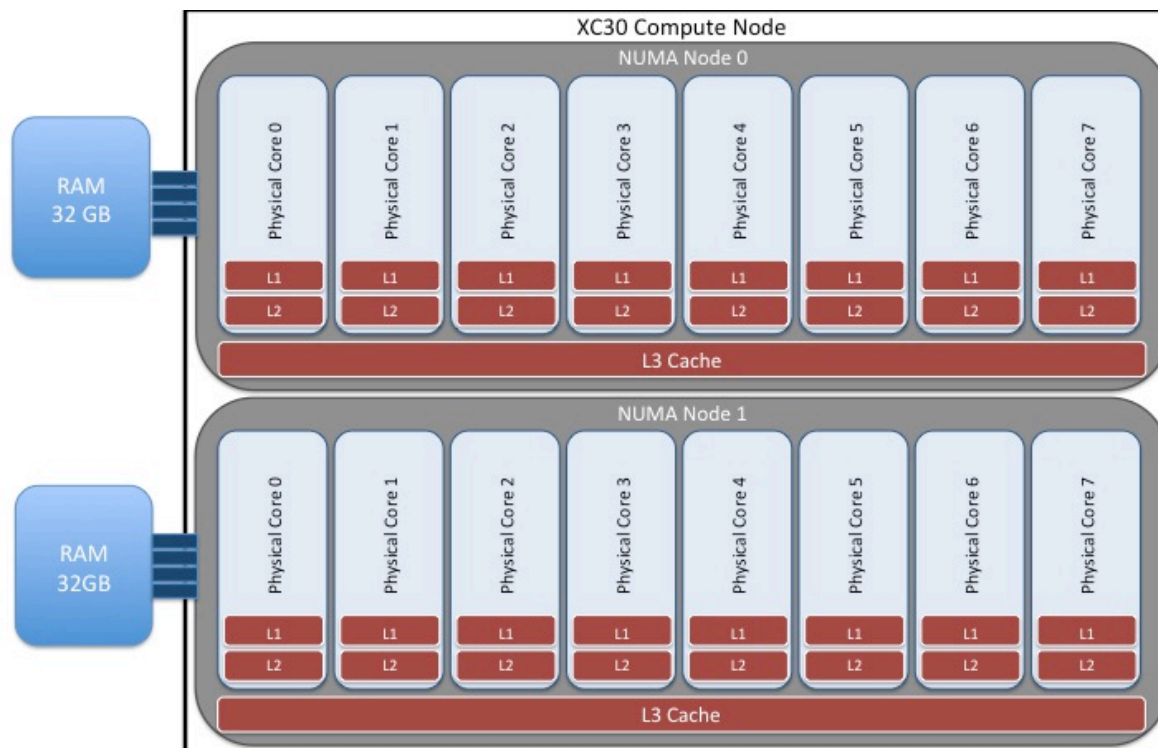
Know Issues

File System

Suzanne Parete-Koon

U.S. DEPARTMENT OF **ENERGY**

OAK RIDGE NATIONAL LABORATORY
MANAGED BY UT-BATTELLE FOR THE DEPARTMENT OF ENERGY

Eos Nodes

- 744 compute nodes, 16 physical cores/node, 64 GB/node.
  - Two NUMA Domains L3 cache = 20 MB
  - No Accelerators
  - Hyper threading (32 logical core/node)

- 2 External Logins

- Total of 11,904 cores and 47.6 TB DDR3 SDRAM

Eos Access and Billing

For the remainder of this year:

- Eos is prioritized as an extra resource for INCITE users.

- All INCITE projects will automatically be given an account on Eos.

- 30 Eos core hours are charged per node.
  - Hours count for Usage
  - Hours are not subtracted from INCITE Allocation

OLCF ●●●●

OAK RIDGE
National Laboratory

Eos Queue Policy

- Queue Policy
  - Unlimited running jobs
  - Limit of (2) *eligible-to-run* jobs per user.
  - Jobs in excess of the per user limit above will be placed into a *held* state, but will change to eligible-to-run at the appropriate time.

Size in Nodes	Wall Clock Limit
< 372	12 hours
>= 372	24 hours

- Feedback?

OLCF ●●●●

Compilers and PrgEnv

- Our programming environment modules will load the correct pairing of compiler version, message passing libraries, and other items required to build and run.

- Programming environments available PrgEnv-Intel, PrgEnv-Cray, PrgEnv-PGI and PrgEnv-GNU.

- Intel is the default compiler and programming environment.

    Module swap PrgEnv-Intel PrgEnv-Cray

- Compiler wrappers ftn, cc,CC still recommended.

- Linking is static by default.

OLCF

OAK RIDGE
National Laboratory

Profiling and Debugging

- Saleable debugger DDT

- ATP

- Profiling Craypat, perftools

- Papi Hhardware counters

- More to come . . .
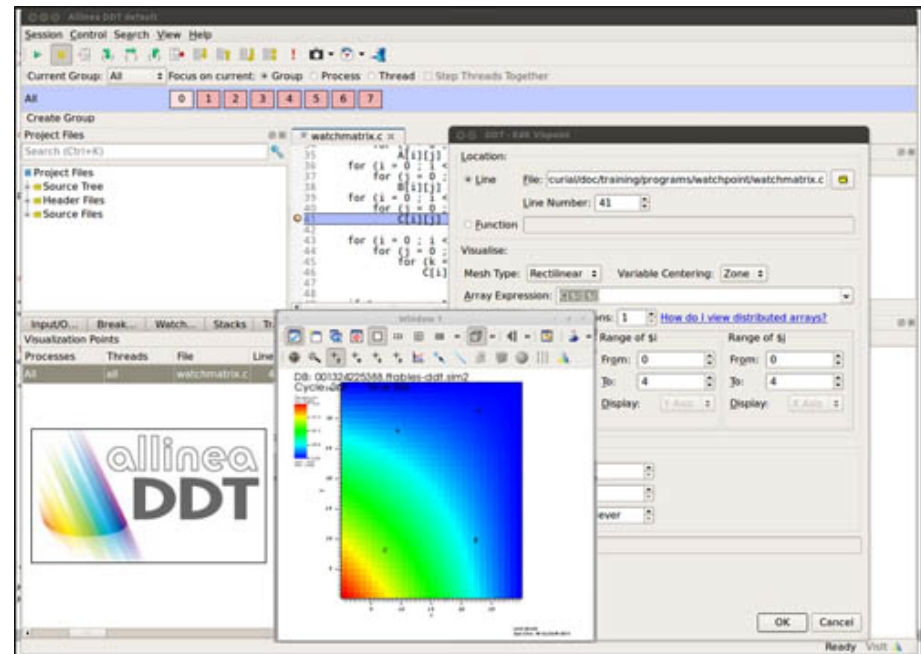
Software

- Early times so we are still updating documentation. In the mean time . . .

% module avail (everything)   %Module list (just what's loaded)

```
[eos-ext2] [08:37:50] [~]$module avail

-------------------- /opt/cray/craype/default/modulefiles --------------------
craype-accel-host              craype-hugepages512M
craype-accel-nvidia20          craype-hugepages64M
craype-accel-nvidia35          craype-hugepages8M
craype-haswell                 craype-ivybridge
```

. . .

```
xc-sysroot/5.0.41

------------------------- /sw/xc30/modulefiles -------------------------
DefApps               git/1.8.3.4          python/2.7.3
adios/1.5.0           gsl/1.16             python/2.7.5
altd/1.0              lammps/19Sep13       python/3.3.2
autoconf/2.69         lustredu/1.3(default) ruby/1.9.3-p448
automake/1.14         mercurial/2.6.3      subversion/1.7.8
boost/1.54.0          mxml/2.6             subversion/1.8.3
cmake/2.8.11.2        ncl/6.1.0            vim/7.4
ddt/4.1-32834         otf/1.12.4salmon
ddt-memdebug/1.0      pdtoolkit/3.19
[eos-ext2] [08:37:55] [~]$G
```

OLCF ●●●●

OAK RIDGE National Laboratory

Documentation

Documentation From a Search
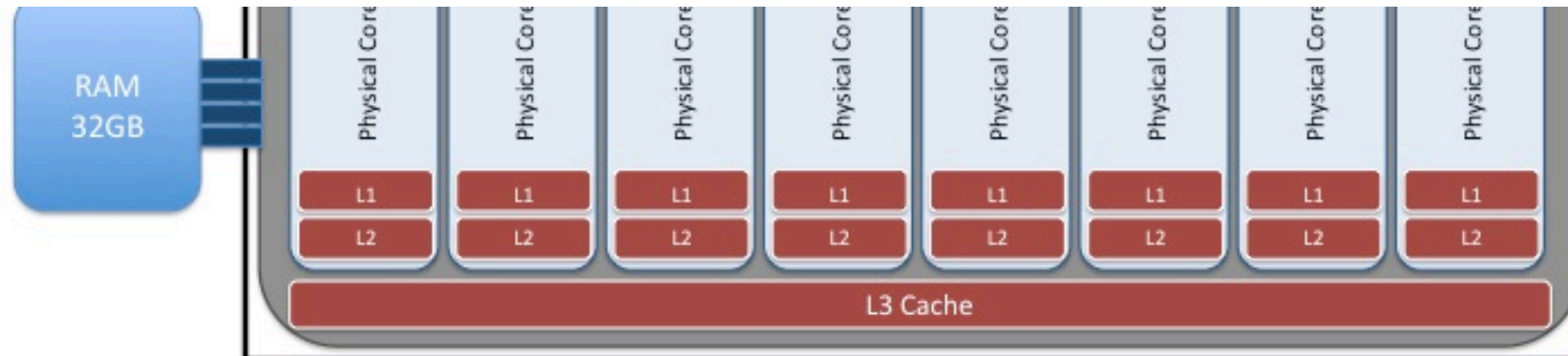
Hyper Threading Overview

Eos includes Intel processors with Intel's Hyper-Threading technology. When Hyper-Threading is enabled, the operating system recognizes each physical core as two logical cores. Two independent processes or threads can run simultaneously on the same physical core, but because the two logical cores are sharing the same execution resources, the two streams may run at roughly half the speed of a single stream. If a process in a stream running on one of the logical cores stalls, the second stream on that core can use the stalled stream's execution resources and possibly recoup cycles that would have been idle if the streams has been run with only one per physical core. Hyper Threading is supported by all the compilers on Eos — Intel, PGI, Cray and GNU.

> **Note:** The `-j2` option to `aprun` enables Hyper Threading on Eos.

Hyper Threading for MPI Applications

For MPI applications, Hyper Threading can be utilized in a few different ways. One way is by running on half the nodes that you would typically need to allocate without Hyper Threading. The example below shows the code to do so and the resulting task layout on a node.

OLCF ● ● ● ●

OAK RIDGE
National Laboratory

Documentation from a search

8.12.3. Hyper Threading

• (Back to Top) • (Edit Article)

Hyper Threading Overview

Eos includes Intel processors with Intel's Hyper-Threading technology. When Hyper-Threading is enabled, the operating system recognizes each physical core as two logical cores. Two independent processes or threads can run simultaneously on the same physical core, but because the two logical cores are sharing the same execution resources, the two streams may run at roughly half the speed of a single stream. If a process in a stream running on one of the logical cores stalls, the second stream on that core can use the stalled stream's execution resources and possibly recoup cycles that would have been idle if the streams has been run with only one per physical core. Hyper Threading is supported by all the compilers on Eos — Intel, PGI, Cray and GNU.

Known Issues

- cmake with the Intel compiler
  - Check for working C compiler fails
  - Adds the –rdynaimc flag and tries unsuccessfully to cross compile.

  This has been mended in the Module file

- However if you still have trouble:
  - explicitly specify the compiler and then force cmake not to check if that compiler works. This can be done with the -D options when you issue cmake:

$ cmake -D CMAKE_CXX_COMPILER=icc

-D CMAKE_C_COMPILER_WORKS=TRUE .

- cmake works fine with PGI, GNU and Cray compilers.

OAK RIDGE
National Laboratory

New Filesystem

- New Spider II Filesystem – only mounted on EOS

- Data transfer is required for files not in Home or HPSS

- Chris Fuson will tell you more