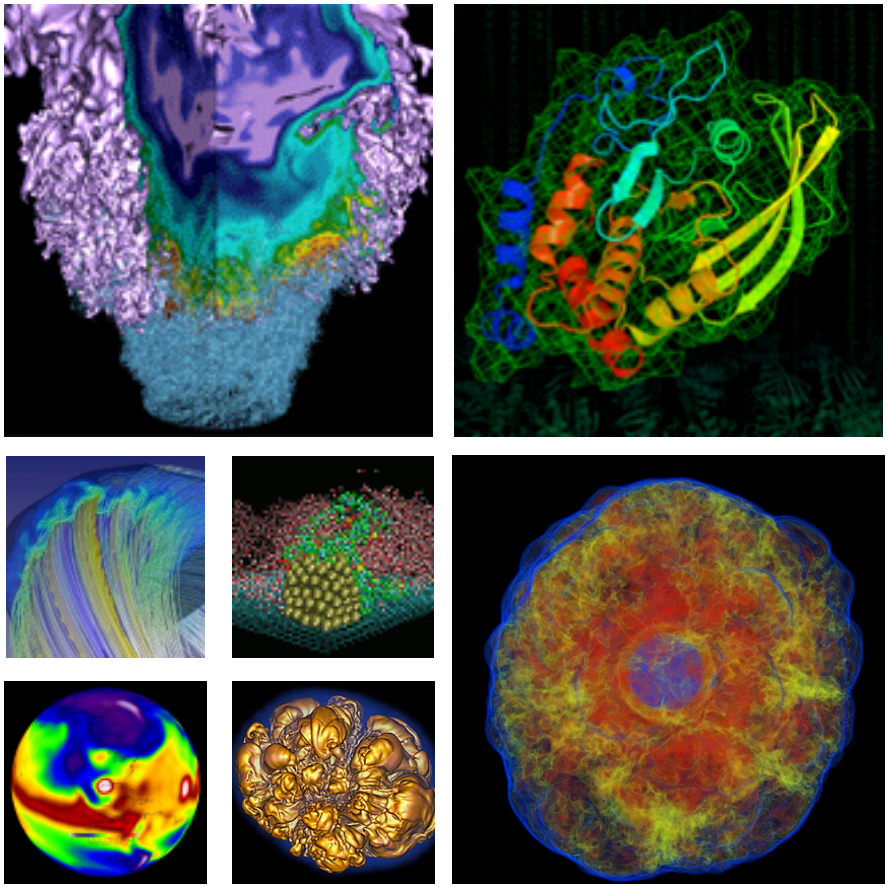
Process and Thread Affinity with OpenMP



Zhengji Zhao

NERSC User Services Group

October 7, 2013

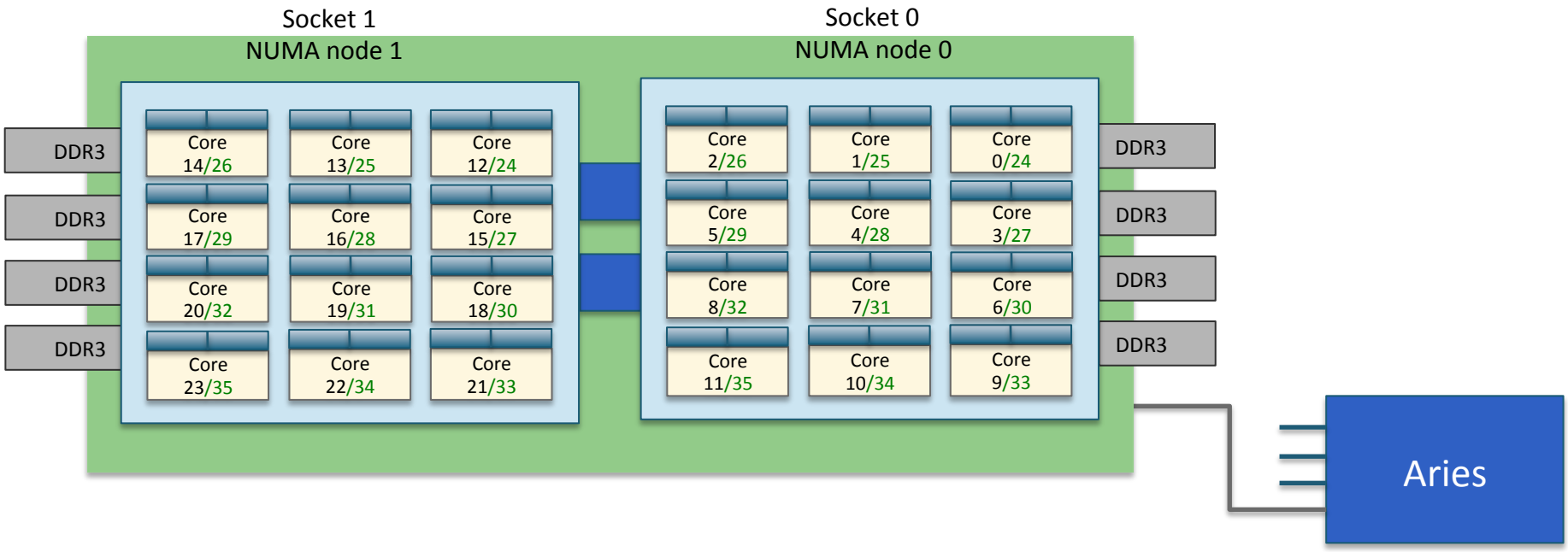
Outline

- Motivation
- How to compile and run
- Default process/thread affinity
- Aprun options to manipulate affinity
- Recommendations to users
- Hands on

Motivation

- What is process/thread affinity?
- In proper process/thread affinity could slow down code performance significantly
- Learn about default process/thread affinity on Edison and how to manipulate process/thread affinity to avoid performance penalty

Edison compute node diagram



Note: This diagram is for illustration purpose only, the actual core distribution on the node may be different from this diagram.

How to compile hybrid MPI + OpenMP codes

- **Intel compiler:**
  - PrgEnv-intel is the default
  - cc -openmp -o xthi.intel xthi.c
- **GNU compiler:**
  - module swap PrgEnv-intel PrgEnv-gnu
  - cc -fopenmp -o xthi.gnu xthi.c
- **Cray compiler:**
  - module swap PrgEnv-intel PrgEnv-cray
  - cc -o xthi.cray xthi.c

How to run hybrid MPI/OpenMP jobs


```
#!/bin/bash -l
#PBS -q regular
#PBS -l mppwidth=240
#PBS -l walltime=12:00:00
#PBS -N my_job
#PBS -j oe

cd $PBS_O_WORKDIR

export OMP_NUM_THREADS=12

# for executables compiled with Intel compiler
aprun –n20 -N 2 -S 1 -d 12 -cc numa_node ./a.out

# for codes compiled with Cray or GNU compilers
# aprun –n20 -N2 -S1 -d12 ./a.out
```

Default process/thread affinity

http://portal.nersc.gov/project/training/EdisonPerformance2013/affinity/

Edison default is to pin each process/thread to a specific core (in a compact way). Most of the cases, the default process/thread affinity works fine, but not always.

Default process/thread affinity

- **Intel compiler**
  - Pure MPI codes, the process affinity works fine if running on fully packed nodes
  - There is issues with thread affinity - all threads from an MPI task are pined to a single core where the MPI task is placed.
- **GNU compiler**
  - Works fine
- **Cray compiler**
  - Works fine

Manipulate process/thread affinity

- **-S, -sn, -sl, -cc, and -ss options control how your application uses the NUMA nodes.**
  - -n Number of MPI tasks.
  - -N (Optional) Number of MPI tasks per Edison Node. Default is 24.
  - -S (Optional) Number of tasks per NUMA node. Values can be 1-12; default 12
  - -sn (Optional) Number of NUMA nodes to use per Edison node. Values can be 1-2; default 2
  - -ss (Optional) Demands strict memory containment per NUMA node. The default is the opposite - to allow remote NUMA node memory access.
  - -cc (Optional) Controls how tasks are bound to cores and NUMA nodes. The default setting on Edison is -cc cpu which restricts each task to run on a specific core.
- **These options are important on Edison if you use OpenMP or if you don't fully populate the Edison nodes.**

http://portal.nersc.gov/project/training/EdisonPerformance2013/affinity

Recommended process/thread affinity

- **Running on unpacked nodes**

```
#PBS –l mppwidth=48    #2 nodes
aprun –n 24 –N 12 –S 6 ./a.out
```

- **Running with OpenMP threads**

```
#for threads per task <= 12
    setenv OMP_NUM_THREADS 12
    #for binaries compiled with Intel compilers
    aprun –n 4 –N 2 -S 1 –d 12 –cc numa_node ./a.out
    # for binaries compiled with gnu or cray compilers.
    aprun -n 4 -N 2 -S1 -d 12 ./a.out

#for threads per task <= 24
    export OMP_NUM_THREADS=24
    #for binaries compiled with Intel compilers
    aprun –n 2 –N 1 –d 24 –cc none ./a.out
    # for binaries compiled with gnu or cray compilers.
    aprun –n 2 –N 1 -d 24 ./a.out
```

Default OMP_NUM_THREADS

- **Intel compiler:**
  - Default to the number of cpu slots available
- **GNU compiler:**
  - Default to the number of cpu slots available
- **Cray compiler:**
  - Default: OMP_NUM_THREADS=1

Hands on

cp –pr /project/projectdirs/training/EdisonPerformance2013/affinity/affinity.tar $HOME

- Compile and run
  - Compile.sh
  - qsub –I –l mppwidth=48 –q debug

- Set OMP_NUM_THREADS
  - aprun -n4 -N4 -S2 -d6 -cc numa_node xthi.intel|sort +3 +5 +11 –n
- Try with OMP_NUM_THREADS not set
  - aprun -n4 -N4 -S2 -d6 -cc numa_node xthi.intel|sort +3 +5 +11 –n

Thank you for your attention!