

# Introducing the Intel Compiler on Edison

Using the Intel Compiler on Edison and Porting PGI codes from Hopper to Intel on Edison

Michael Stewart  
NERSC User Services Group  
February 14, 2013

# Current Edison Programming Environment Status

- The Edison/XC30 Programming Environment is still a work in progress.
- There may be major changes to it by the time Edison is accepted.
- Monitor the [www.nersc.gov](http://www.nersc.gov) Edison pages to keep track of the changes.

# Introduction

- Similarities and differences between the Edison and the Hopper compiling environments.
- The Edison Intel programming environment.
- Porting from PGI on Hopper to Intel on Edison.
- Preliminary report on Edison performance.
- Your feedback.

# Difference Between the Edison and Hopper Compiling Environments

- Edison supports 3 compilers: Intel(default), Cray, and Gnu. PGI and Pathscale are not installed.
- Gnu and Cray use Cray libsci for math library routines, but Intel uses Intel's MKL math library (add "-mkl=cluster" as an LDFLAG).
- Several Cray provided libraries, e.g. netcdf and hdf5, have different names, prefixed with cray-.
- The Intel OpenMP and hybrid MPI/OpenMP run time environments do not work by default at this time. See below for workarounds.

# Edison Math Libraries

- Edison Gnu and Cray math library is the same as on Hopper.
  - Cray libsci
  - no special flags needed at link time.
- Intel uses Intel's MKL math library
  - Add "-mkl=cluster" as a flag at link time to load the MKL library.
  - The libsci library is currently not available for the Intel compiler.

# Cray Library Module Name Changes on Edison

- New names for Hopper Cray modules, prefixed with **cray-**
  - **cray-hdf5**
  - **cray-hdf5-parallel**
  - **cray-netcdf**
  - **cray-netcdf-hdf5parallel**
  - **cray-petsc** (not yet available with Intel)
  - **cray-trilinos** (not yet available with Intel)

The modules also exist currently on Hopper with the new names.

# Converting from PGI to Intel

PGI	Intel	Description
-fast	-fast -no-ipo	Produce well optimized code.
-mp= nonuma	-openmp	Implement OpenMP directives.
-Mfixed	-fixed	Fortran fixed source form.
-Mfree	-free	Fortran free source form.
-byteswapio	-convert big_endian	Read and write Fortran unformatted data files as big endian.
default	-mkl=cluster	Use math library routines.
-V	--version	Show Fortran compiler version.

# The -fast option to the Intel Compiler

- The Intel and PGI compiler -fast options have different effects.
- PGI
  - "A generally optimal set of options is chosen for targets that support SSE capability."
  - Same as -fastsse.
- Intel
  - Includes interprocedural optimization which can increase compile time by an order of magnitude or cause it to fail.
  - Always turn it off with -no-ipo when using -fast.
  - -fast -no-ipo generally improves performance over the default (no optimization arguments) on Edison.
  - No significant improvement to performance over default on Hopper.

# The Intel Hybrid/OpenMP run time environment

- The Cray thread affinity settings and Intel's run time OpenMP environment conflict.
- Intel has an extra thread at run time, so 2 threads are scheduled on the same core and the job takes twice as long as it should.
- Current workaround:
  - `OMP_NUM_THREADS <=8`
    - set `KMP_AFFINITY compact`
    - `aprun -cc numa_node ....`
  - `OMP_NUM_THREADS >8 and <=16`
    - set `KMP_AFFINITY scatter`
    - `aprun -cc none ....`

# Compiler performance on Edison

- Based on a set of NERSC 6 and NPB 3.1.1 benchmarks.
- All compilers produce significantly faster code on Edison compared to Hopper.
- Cray and Intel compilers have comparable performance, Gnu compiled codes run 10% slower on average.
- Recommended optimization arguments:
  - Intel: -fast -no-ipo (different from Hopper).
  - Cray: default, no explicit arguments (same as Hopper).
  - Gnu: -O3 -ffast-math (same as Hopper).

# Questions and Comments