

Benchmark Performance of Different Compilers on a Cray XE6

Mike Stewart and Helen He
NERSC User Services Group
May 23-26, CUG 2011



National Energy Research
Scientific Computing Center

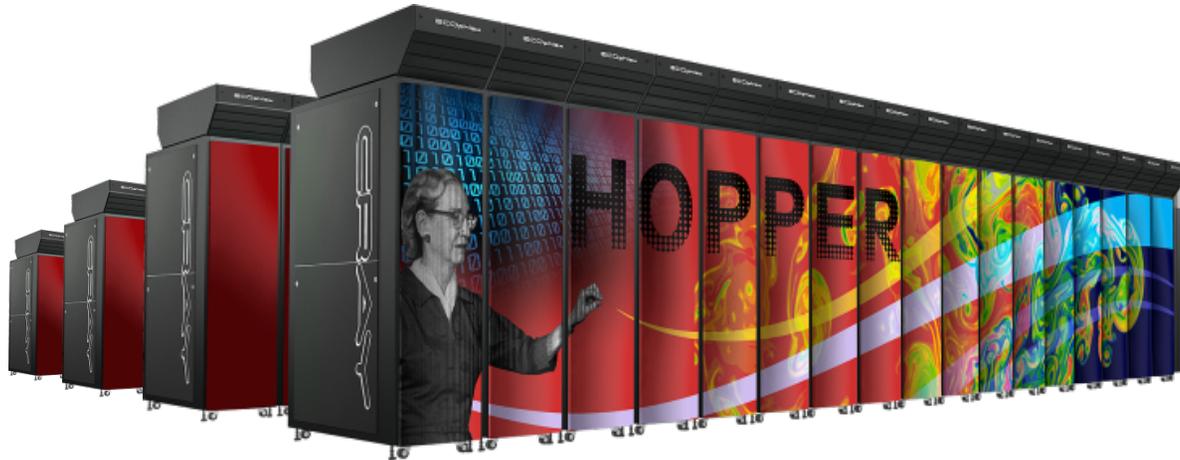


Lawrence Berkeley
National Laboratory



Outline

- **Introduction**
- **Available Compilers on Hopper**
- **Recommended Compiler Options**
- **Benchmarks Used in the study**
- **Performance Results from Each Compiler**
- **Summary and Recommendations**



- Cray XE6, 6,384 nodes, 153,126 cores.
- Each node has 2 twelve-core AMD MagnyCours 2.1 GHz procs.
- 1.28 Pflops/peak, 212 TB memory.



Available Compilers on Hopper

- **Portland Group Compilers**
 - This is the default compiler on Hopper
- **Pathscale Compilers**
 - % module swap PrgEnv-pgi PrgEnv-pathscale
- **Cray Compilers**
 - % module swap PrgEnv-pgi PrgEnv-cray
- **GNU Compilers**
 - % module swap PrgEnv-pgi PrgEnv-gnu



Compile Codes on Hopper

- **Cross compilation from login nodes to build executables to run on the compute nodes.**
- **To use a particular compiler, first swap to the corresponding PrgEnv.**
- **Then use compiler wrappers:**
 - **ftn for Fortran codes**
 - **cc for C codes**
 - **CC for C++ codes**
- **The wrappers can find the proper system and MPI libraries.**



Compiler Flags Comparison

PGI	Pathscale	Cray	GNU	Explanation
-fast	-Ofast	-O3	-O3	High level optimization
-mp=nonuma	-mp	-h omp (default)	-fopenmp	Enable OpenMP
-byteswapio	-byteswapio	-h byteswapio	-fconvert=swap	Read files in big-endian
-Mfixed	-fixedform	-f fixed	-ffixed-form	Fixed form source
-Mfree	-freeform	-f free	-ffree-form	Free form source
-V	-dumpversion	-V	--version	Show version info
not implemented	-zerouv	-e 0	-finit-local-zero	Zero fill uninitialized values



Recommended Options: PGI Compiler

- **NERSC recommends:**
 - **-fast or –fastsse**
- **PGI User Documentation:**
 - **“-fast –Mipa=fast” is a good set of options.**
- **Cray recommends:**
 - **-fast –Mipa=fast**
 - **If can be flexible with precision, also try –Mfprelaxed.**



Recommended Options: Pathscale Compiler

- **NERSC recommends:**
 - **-Ofast**
- **Pathscale User Documentation:**
 - **Start with `-O2`, then `-O3`,**
 - **then `-O3 -OPT:Ofast`, then `-Ofast`.**
- **Cray recommends:**
 - **-Ofast**



Recommended Options: Cray Compiler

- **NERSC recommends:**
 - **-O3**
- **Cray recommends:**
 - **Use default `-O2`, which is equivalent to `-O3` or `-fast` in other compilers.**
 - **Use `-O3,fp3` (or `-O3 -hfp3`)**
 - **`-O3` only slightly better than `-O2`**
 - **`-hfp3` gives maximum freedom in floating point optimization, may not conform to IEEE standard.**



Recommended Options: GNU Compiler

- **NERSC recommends:**
 - **-O3**
- **Cray recommends:**
 - **-O3 –ffast-math –funroll-loops**
 - **-ffast-math: may not conform IEEE standard**



NERSC6 Application Benchmarks

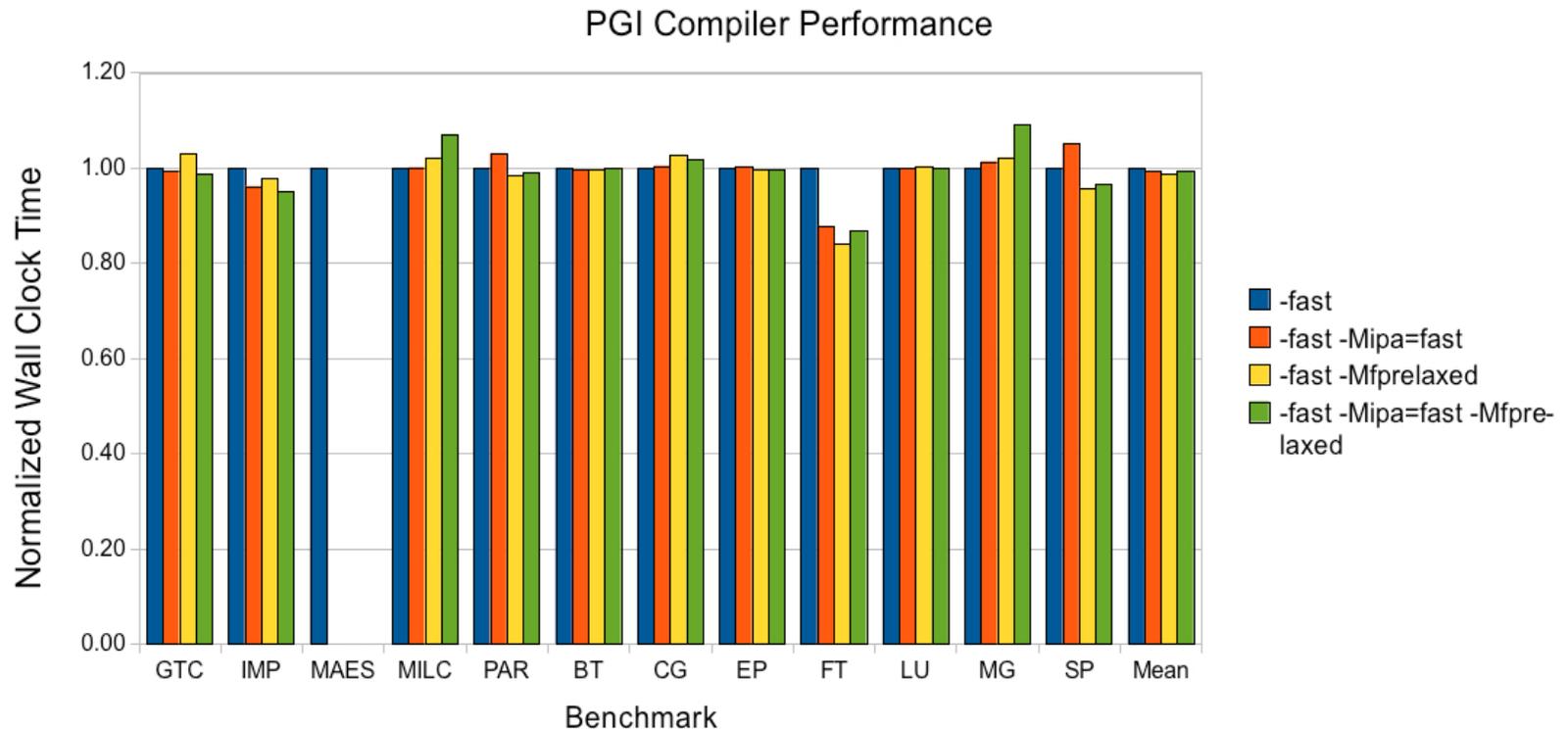
Benchmark	Science	Algorithm	Concurrency	Language
GTC	Fusion	PIC, finite difference	2048 (weak scaling)	F90
IMPACT-T	Accelerator Physics	PIC, FFT	1024 (strong scaling)	F90
MAESTRO	Astrophysics	Block structured-grid multiphysics	2048 (weak scaling)	F90
MILC	Lattice Gauge Physics (QCD)	Conjugate gradient, sparse matrix, FFT	1024 (weak scaling)	C, Assembly
PARATEC	Material Science	DFT, FFT, BLAS	1024 (strong scaling)	F90



NPB 3.3 Benchmarks

Benchmark	Full Name	Level	Concurrency
BT	Block Tridiagonal	D	256
CG	Conjugate Gradient	E	256
EP	Embarassingly Parallel	E	256
FT	Fast Fourier Transform	D	256
LU	Lower-Upper Symmetric Gauss-Siedel	E	256
MG	MultiGrid	E	256
SP	Scalar Pentadiagonal	D	256

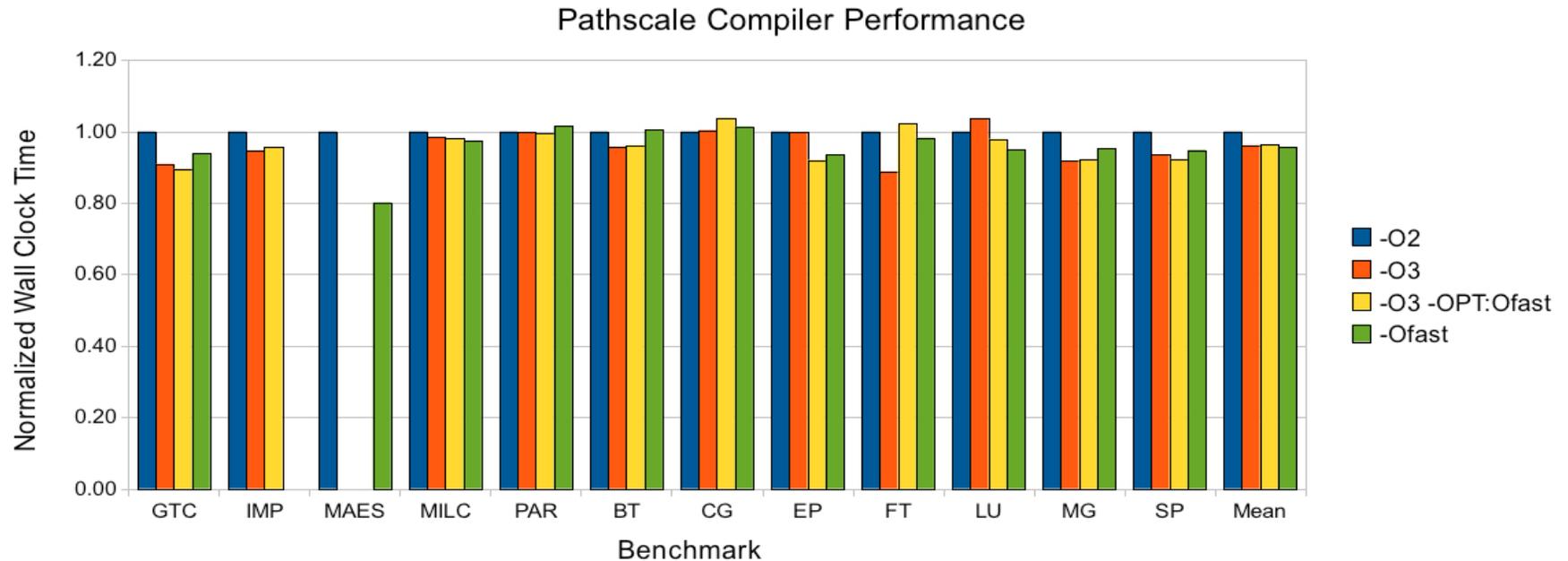
PGI Compiler Results



- Other 3 options do not significantly improve performance over “-fast”.
- The NPB FT case D is an exception.



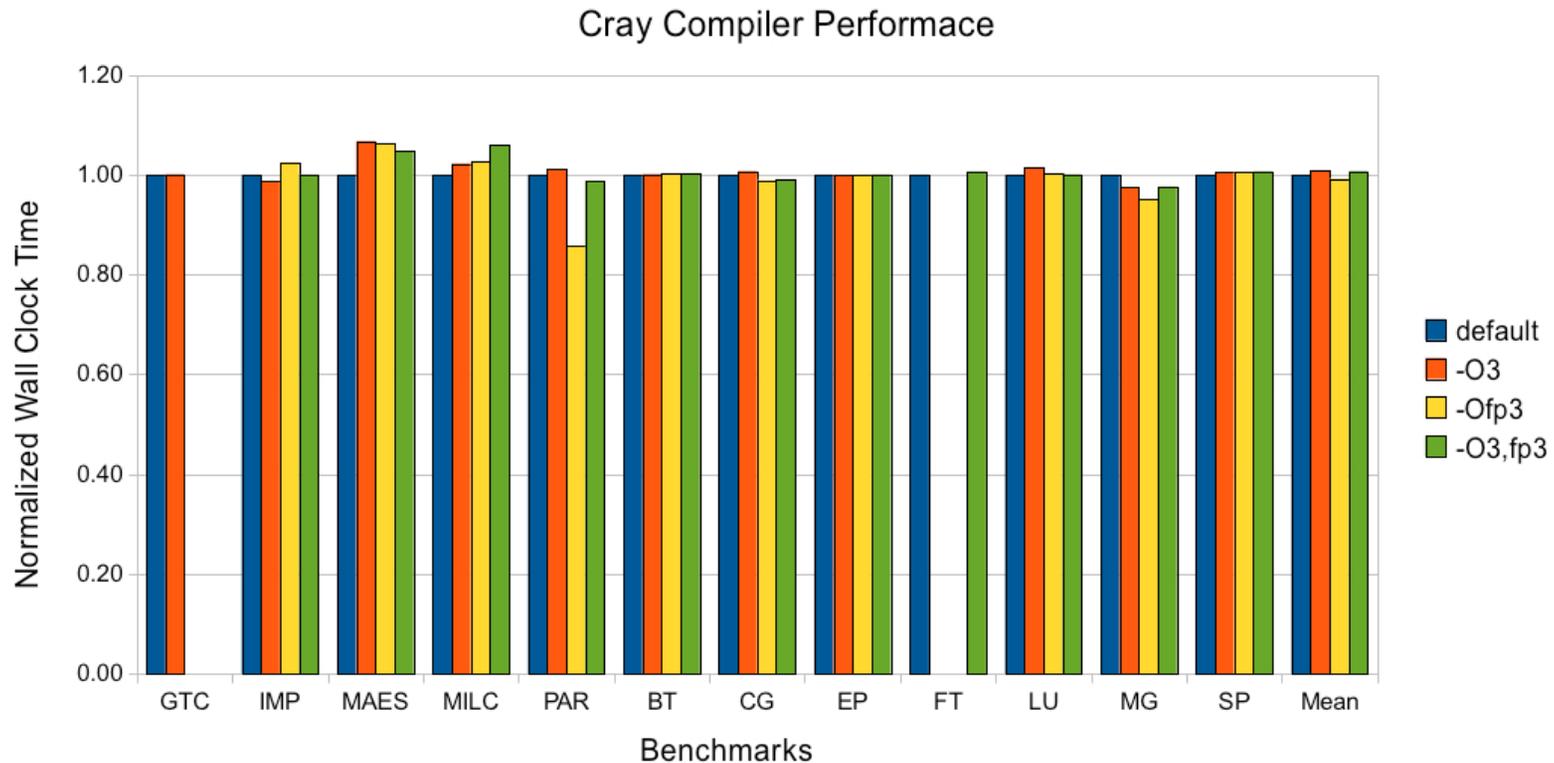
Pathscale Compiler Results



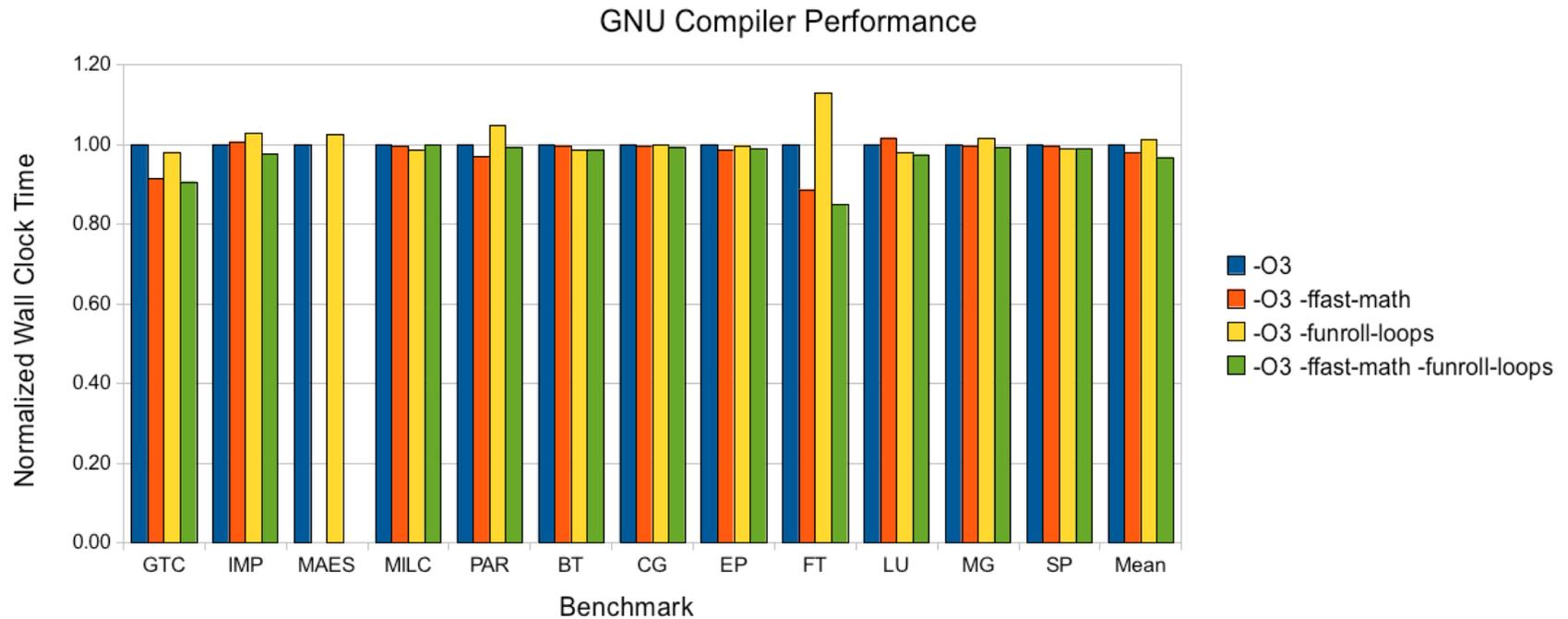
- **-O2 performs worse than other 3 options.**
- **-O3 optimizes almost all benchmarks well.**
- **Extra options on top of -O3 do not improve significantly.**



Cray Compiler Results



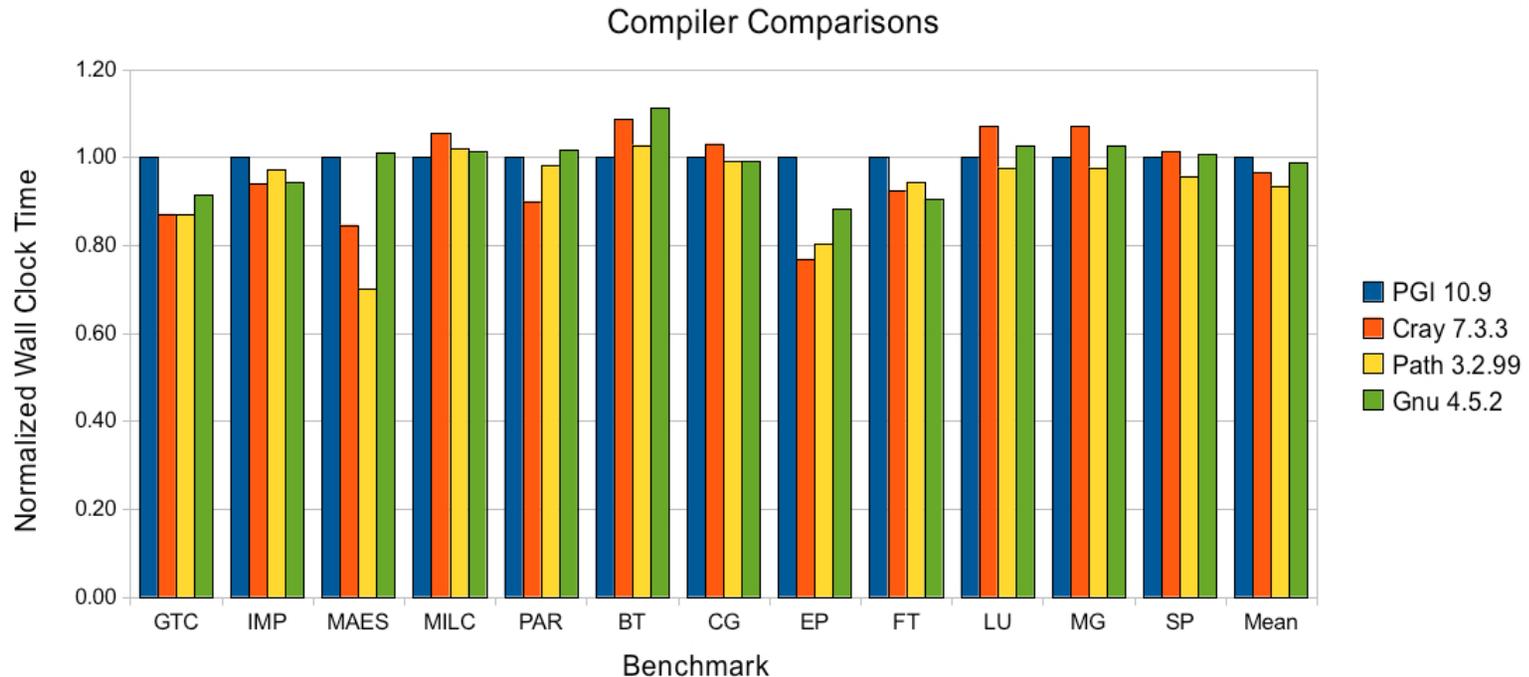
- Only one benchmark with `-Ofp3` shows significant improvement over default `-O2`.



- **-O3 generally gives a good level of optimization.**
- **Worth to try `-ffast-math` option. Improves performance significantly in some cases.**



Overall Compilers Comparison



- **Pathscale fastest: 6 out of 12.**
- **Cray fastest: 3 out of 12.**
- **PGI fastest: 2 out of 12.**
- **GNU fastest: 1 out of 12.**
- **Mean against PGI: Cray 0.96, Pathscale 0.94, GNU 0.99**



Summary and Recommendations

- Users should experiment with different compilers and compiler options to tune their application performance on Hopper.
- On the average the Pathscale and Cray compilers produce somewhat faster code on Hopper (or another Cray system), since they are specifically designed for these processors. In addition the Cray compilers make use of the Cray math libraries at compile time to further optimize codes.
- PGI compilers are available on a wide variety of platforms other than Cray machines. Many existing codes have PGI targeted Makefiles, could generate very good performance.
- Using the gnu compilers allows you to compile on virtually every Unix and Linux system. Although the performance on Hopper for some codes with GNU compilers is quite good, there is no guarantee for optimal performance on other platforms.