Using the Cray Compiler at NERSC
- Usability and Performance

Zhengji Zhao, Megan Bowling and Jack Deslippe
NERSC User Services

Cray Quarterly, July 25, 2012

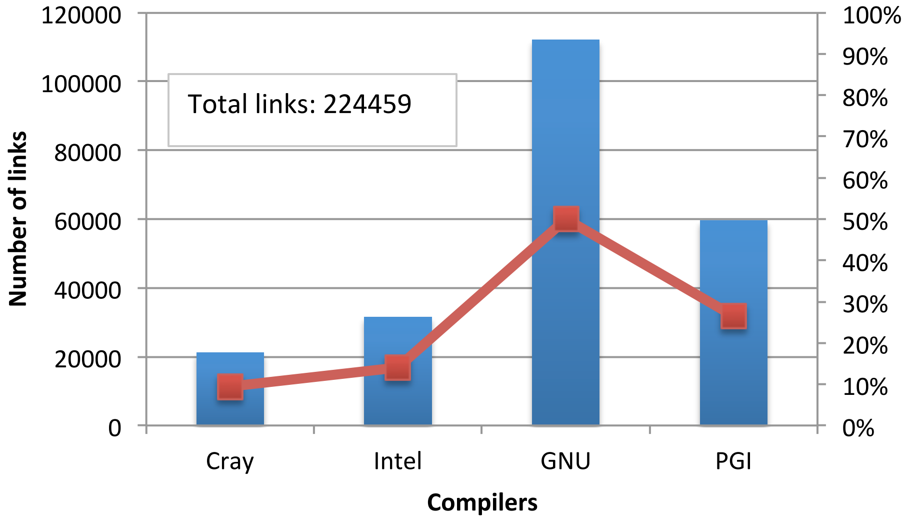
U.S. DEPARTMENT OF ENERGY | Office of Science
NERSC National Energy Research Scientific Computing Center
BERKELEY LAB Lawrence Berkeley National Laboratory

Motivation and outline

- Provide feedback to Cray about how Cray compiler is used at NERSC, focusing on its usability and performance

- Report issues encountered with compilation, execution, validity check and performance, using a set of materials and chemistry application codes Instead of using the standard N6 application benchmark codes

Compiler usage on Hopper (2012-05-15 - 2012-07-20)

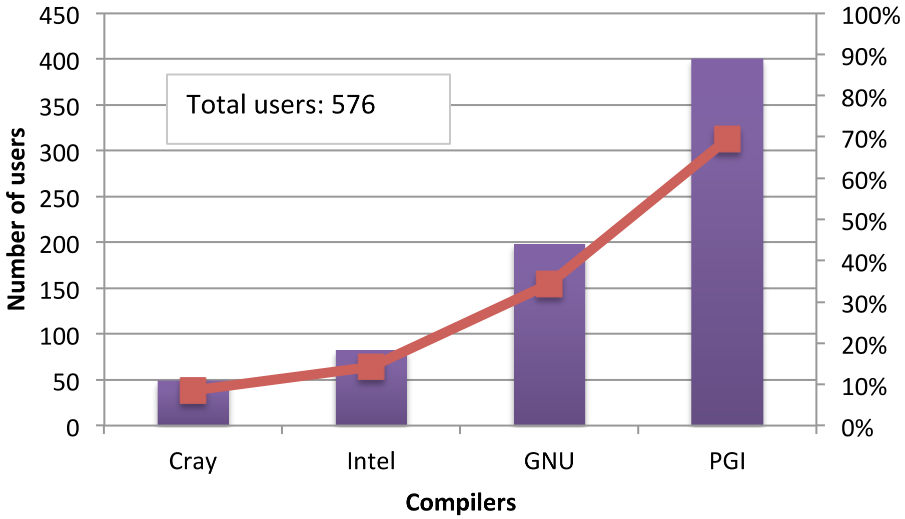


1. Automatic Library Tracking Database (ALTD, developed by NICS) tracks the library usage both at compile and run time by intercepting the "ld" and "aprun" commands, respectively.

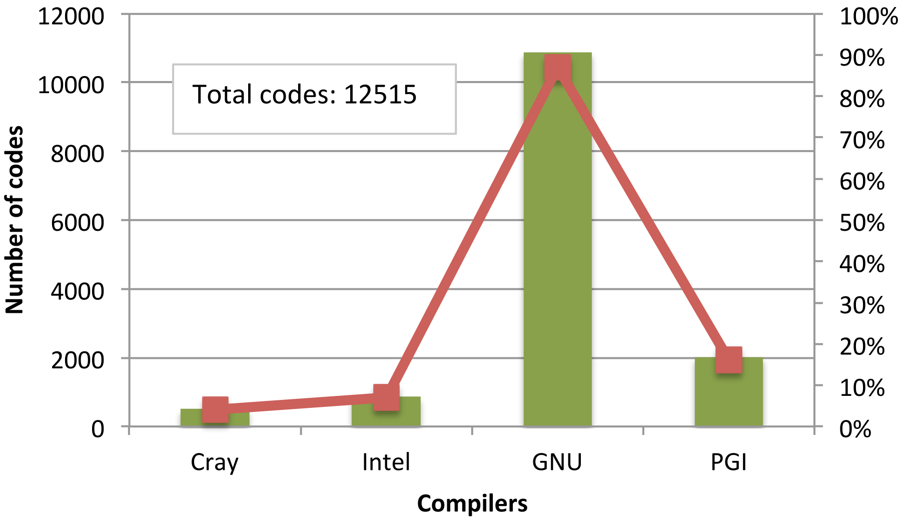
2. 6/21/2012 in production on Hopper

Among the 224459 successful links, only 9% of them used the Cray compiler

Compiler usage on Hopper (2012-05-15 – 2012-07-21)--continued



9% of unique users who compiled codes on Hopper used Cray compiler.



4% of unique binaries were compiled with Cray compiler.

Difficulties encountered when compiling codes using Cray compiler

- Many application codes do not support the Cray compiler in their configure script – configure fails
- Cray compiler often fails to compile the codes that all other compilers, PGI, GNU, Intel, compile fine.
  - Pros: good for new code development, less buggy codes; can help finding bugs in the codes .
  - Cons: difficult to use with existing codes
- Atomics operation is not supported in Cray compiler

Compilation example: VASP

```
diff -r vasp.5.2/aedens.F ../orig/vasp.5.2/aedens.F
<       TYPE (grid_3d),TARGET ::  GRID_SOFT,GRIDC_,GRIDUS
---
>      TYPE (grid_3d)     GRID_SOFT,GRIDC_,GRIDUS
```

Failed with Cray compiler – works with all others on Hopper

```
diff -r vasp.5.2/dfast.F ../orig/vasp.5.2/dfast.F
<       USE dfast,only : NBLK
---
USE dfast
```

Failed with Cray compiler – works with all others on Hopper

Compilation example: VASP

```
diff -r vasp.5.2/hamil.F ../orig/vasp.5.2/hamil.F
<     SUBROUTINE PW_CHARGE_TRACE(WDES1, CHARGE, CR1, CR2)
---
>     SUBROUTINE PW_CHARGE_TRACE(WDES1,  CHARGE, NDIM, CR1, CR2)
```

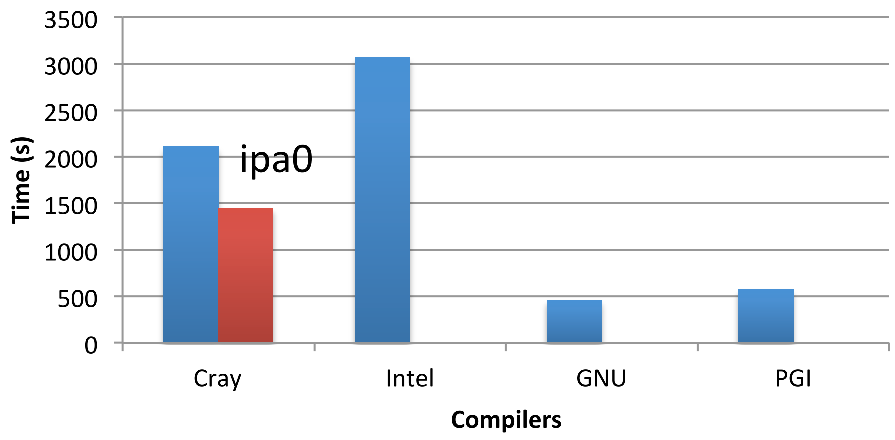
Failed with Cray compiler

```
diff -r vasp.5.2/subrot_lr.F ../orig/vasp.5.2/subrot_lr.F
<         W0%CW(:,:,NK,ISP), W0%CPROJ(:,:,NK,ISP), DEG_CLUSTER(NK,ISP)
   %DEG_CLUSTER, .FALSE., .FALSE.)
---
>         W0%CW(1,1,NK,ISP), W0%CPROJ(1,1,NK,ISP), DEG_CLUSTER(NK,ISP)
   %DEG_CLUSTER, .FALSE., .FALSE.)
```

Failed with Cray compiler – works with all others on Hopper

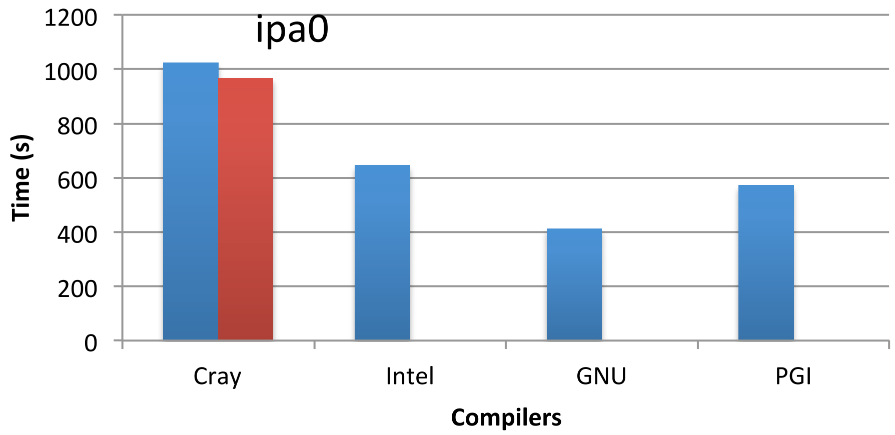
Cray compiler takes longer time to compile

VASP (Fortran) Compalation Time



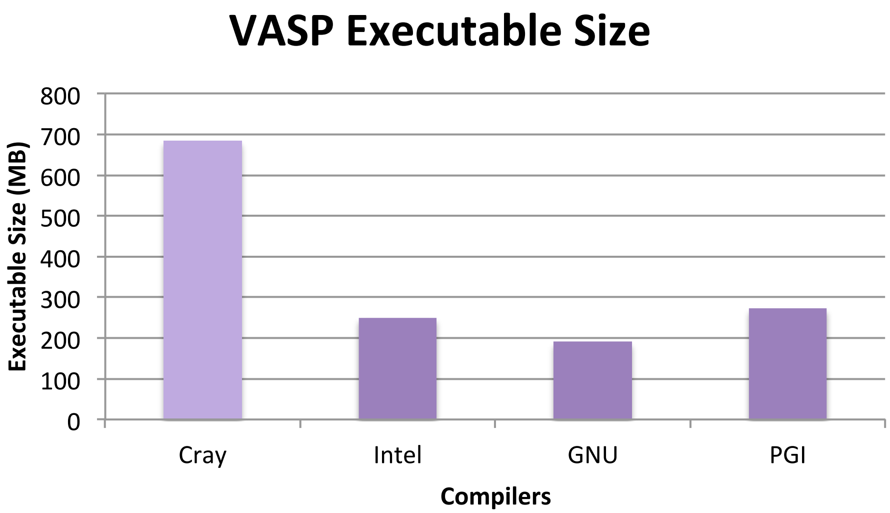
Compiler	Compile options
Cray	Default; -O –ipa0
Intel	-O3, -fast
GNU	-O3, -ffast-math
PGI	-fastsse, -O3, -Mvect

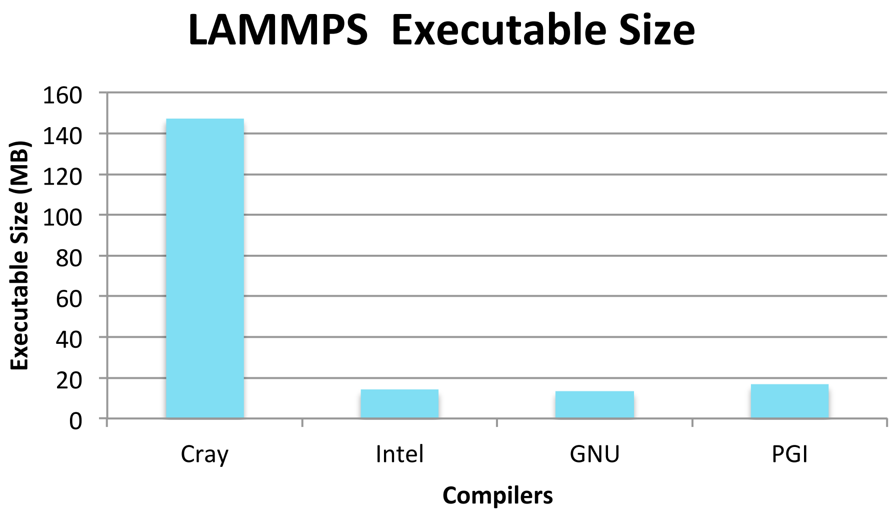
LAMMPS (C++) Compilation Time



Intel compiler takes longest time to compile

Cray compiler generates larger binaries in size





Run time and validity issues

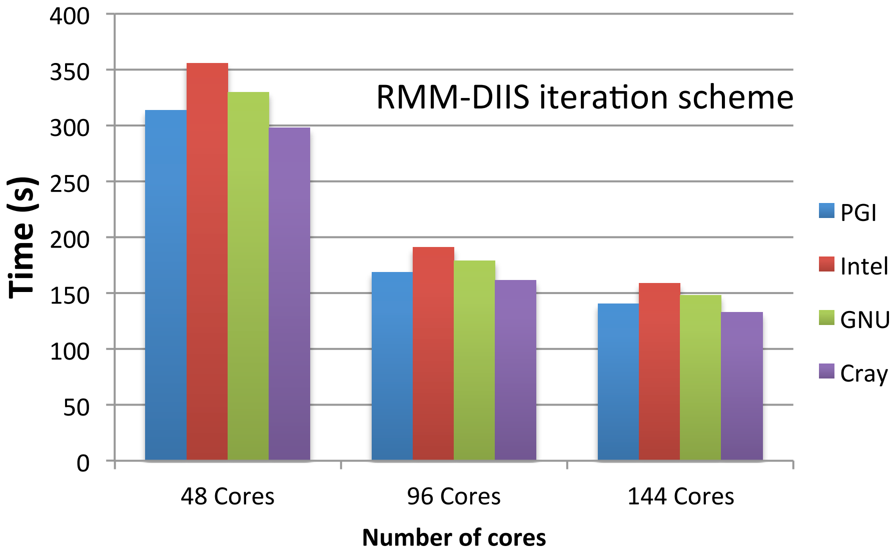
- VASP failed validity check
  - Failed to run for 2 of the test cases (out of 3)
  - If remove all compiler optimizations, then code ran fine
  - "Randomly" lowered the optimization levels for the "relevant" routines, and then the code passed the other two test cases.
- Quantum Espresso generated wrong results similarly
  - Had to lower a specific routine's compiler optimization levels.
- NWChem failed to run
  - Error

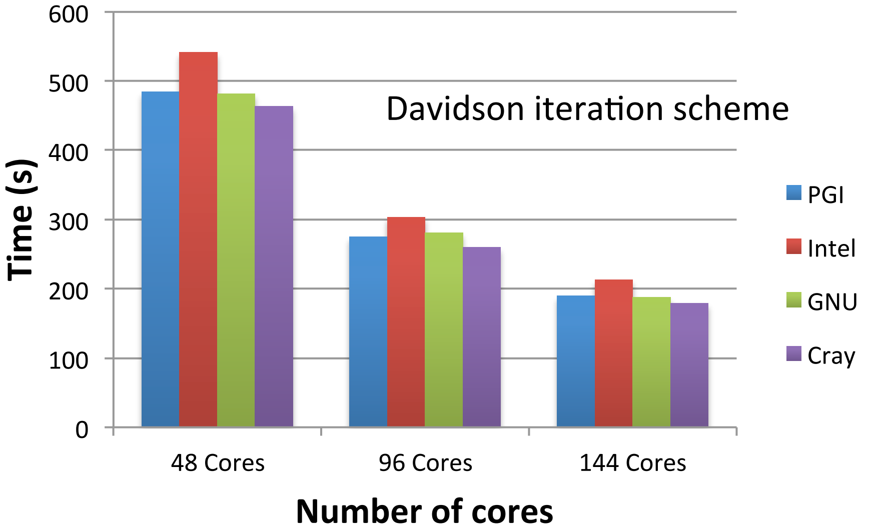
```
MA internal error: MAi_inform_base: invalid datatype: 307307478419244017
MA internal fatal error: MA_sizeof: unable to set sizes of FORTRAN datatypes
```

VASP (5.2.12)

- Program Description
  - VASP is a Fortran code that performs atomic scale materials modeling.
- Options explored
  - Compilers and optimization flags used
    - PGI: -fastsse, -O3, -Mvect
    - Intel: -O3, -fast
    - GNU: -O3, -ffast-math
    - Cray: -O –ipa0
- Tested with 3 test cases
  - Algorithms: DIIS-RMM, Davidson, Hybrid
  - Concurrencies: 48, 96, 144; 384,768; 48,72

Cray compiler outperforms other compilers with medium sized VASP runs





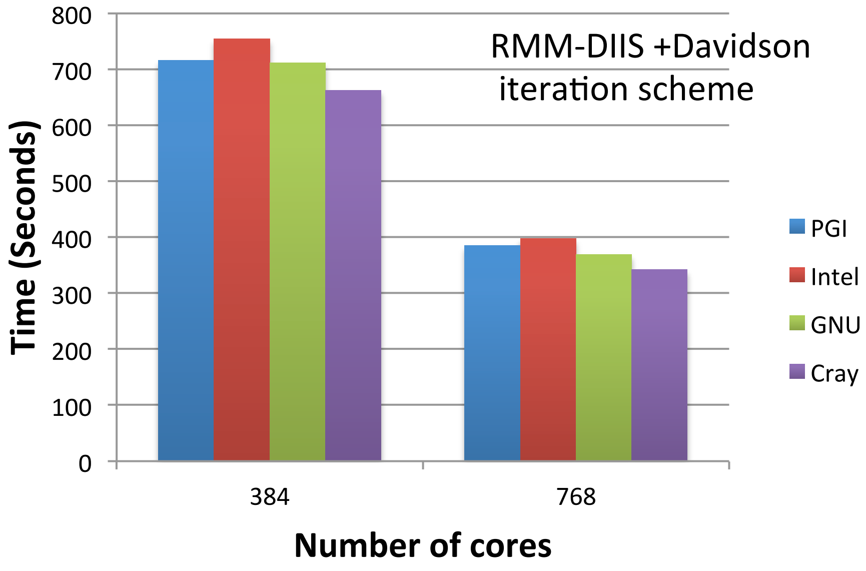
Test case 1:

- NERSC user provided test case:
- A 155 atom system
- The time to complete first 20 electronic steps were measured

Compiler	Performance gain relative to PGI compiler (%)
Intel	-12%
GNU	-6% ~ +1%
PGI	default
Cray	5.8%

VASP runs faster by 5.8% when switching to Cray compiler.

Cray compiler outperforms other compilers for larger test cases



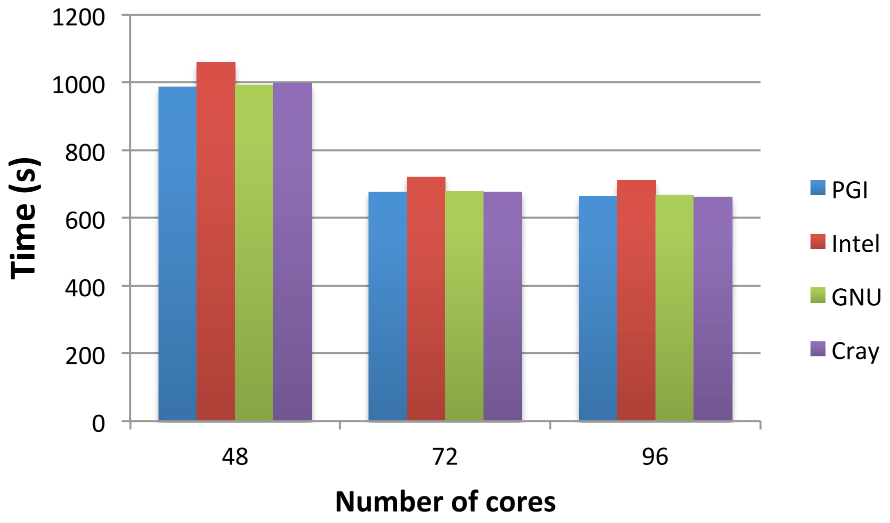
Test case 2

- NERSC user provided
- A 660 atom system
- Time for first 4 electronic steps

Compiler	Faster than the default compiler by (%)
Intel	-5%
PGI	default
GNU	4%
Cray	11%

VASP with Cray compiler runs faster by up to 11% for the larger test case.

Compiler performance varies depending on job types



Test case 3

- Provided by NERSC users
- Hybrid calculation for a 105 atom system

Compiler	Faster than PGI compiler by (%)
Intel	-6%
Cray, GNU	0%
PGI	default

VASP with Cray compiler runs at the same speed as PGI compiler for the hybrid jobs

Performance increase compared to PGI

Program	PGI	Intel	GNU	Cray	Best compiler
VASP	0%	-12% ~ -5%	-6% ~ 4%	0% ~ 11%	Cray
QE	0%	2%	-1%	-7%	Intel
NAMD	0%	14%	18%	Failed	GNU
LAMMPS*	0%	5% ~ 17%	-5% ~ 9%	-6% ~ 4%	Intel
BerkeleyGW	0%	0%	-13%	-8%	PGI/Intel
NWChem	0%	12% ~ 34%	-9% ~ 28%	Failed	Intel

Compiler versions	
PGI	11.9.0
GNU	4.6.2
Intel	12.1.2.273
Cray	cce/8.0.1

Blue: max performance increase
Red: max performance decrease

*) LAMMPS data updated with newer versions of compilers, pgi/12.4.0, intel/12.1.4.319, cce/8.0.5, gcc/4.6.3

Summary

- Cray compiler is in low usage on Hopper
- Cray compiler is proven to be difficult to use for existing third party application codes.
- The performance varies, a good performance is observed with VASP (Fortran code), but not for other codes.
- We do not recommend changing the compiler default on Hopper to Cray compiler at any time soon until the usability issues are resolved or reduced to some extent.

```
zz217@hopper12:~> qsub -I -l mppwidth=24 -q debug -V
qsub: waiting for job 1975244.sdb to start
qsub: job 1975244.sdb ready

ModuleCmd_Switch.c(172):ERROR:152: Module 'PrgEnv-cray' is currently not loaded
zz217@nid04755:~>
```