

Building Software Environments for Research Computing Clusters

Mark Howison, Aaron Shen, Andrew Loomis

Brown University Center for Computation & Visualization

Challenges [HPC] sysadmins face

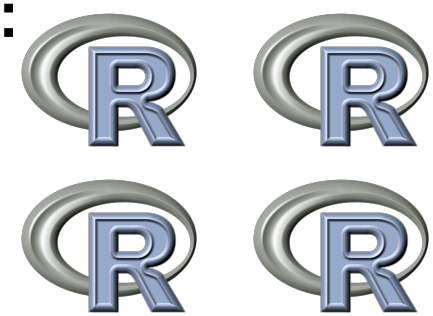
BROWN UNIVERSITY CENTER FOR COMPUTATION & VISUALIZATION

Maintain multiple software installations on a shared filesystem supporting hundreds of users

-Typical system:



HPC:

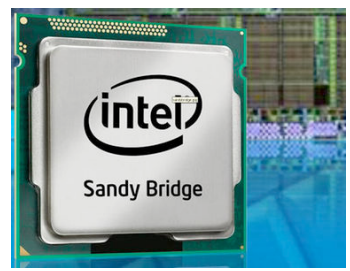


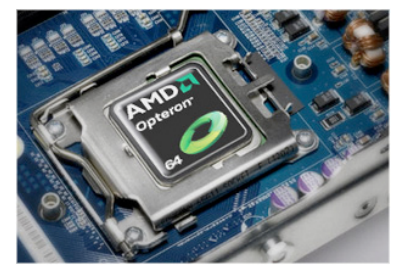
Building/optimizing software

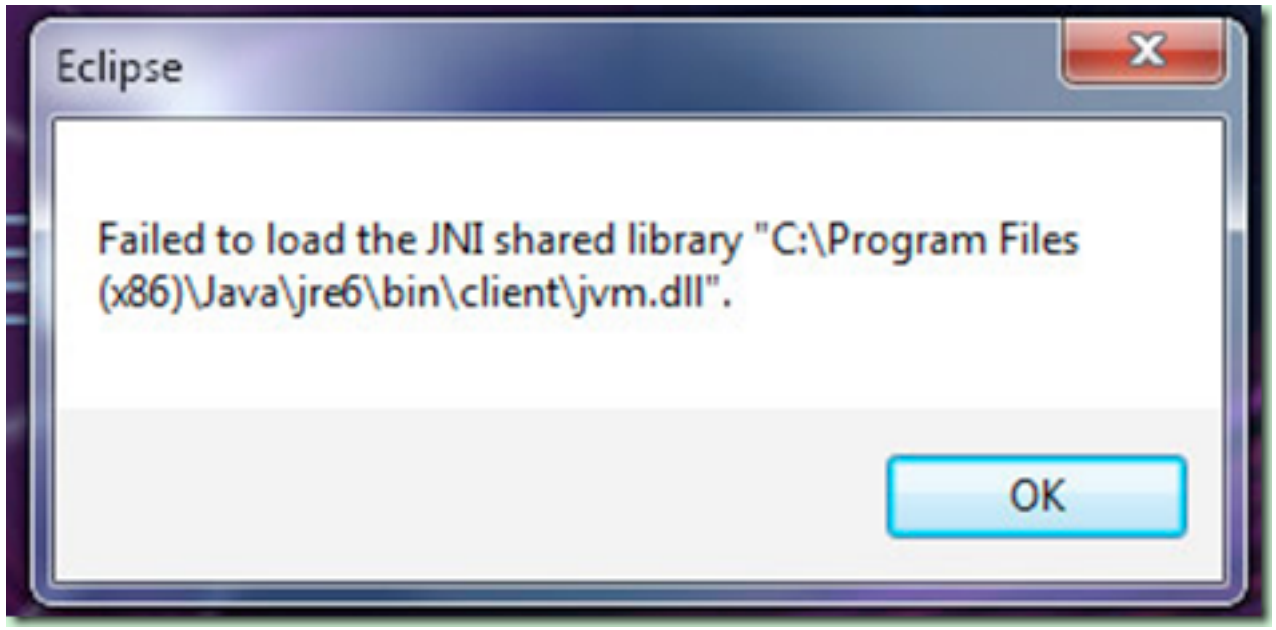
-Many different dependencies

```
./configure --disable-threads --with-petsc=/gpfs/runtime/opt/petsc/3.0.0-p12 --with-petsc-arch=linux-gnu-
cxx-opt --with-umfpack --with-trilinos=/gpfs/runtime/opt/trilinos/10.2.2 --with-metis=/gpfs/runtime/opt/
metis/4.0.1 --with-blas=goto2 --with-lapack=goto2 --with-p4est=/gpfs/runtime/opt/dealii/7.0.0/p4est --with-
mumps=/gpfs/runtime/opt/mumps/4.9.2 --with-scalapack=/gpfs/runtime/opt/gotoblas2/1.13/lib --with-blacs=/
gpfs/runtime/opt/gotoblas2/1.13/src/BLACS --enable-mpi CC=mpicc CXX=mpiCC LDFLAGS=-L/gpfs/runtime/opt/
gotoblas2/1.13/lib
```

Multiple hardware setups






Eclipse
Failed to load the JNI shared library "C:\Program Files (x86)\Java\jre6\bin\client\jvm.dll".
OK

Challenges [HPC] sysadmins face

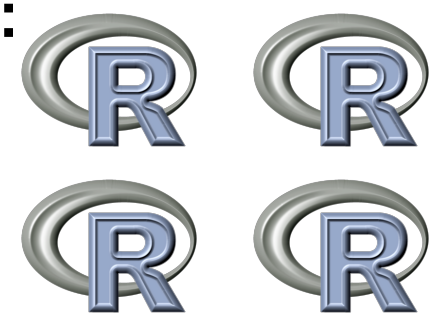
BROWN UNIVERSITY CENTER FOR COMPUTATION & VISUALIZATION

Maintain multiple software installations on a shared filesystem supporting hundreds of users

-Typical system:



HPC:

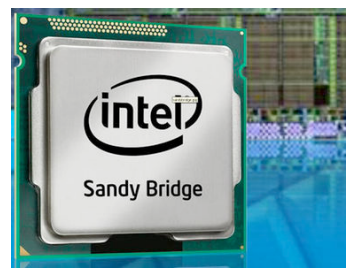


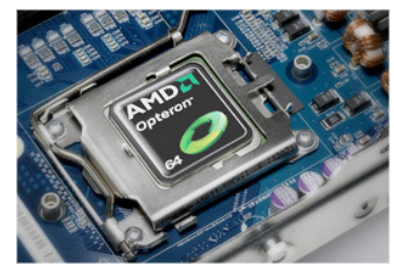
Building/optimizing software

-Many different dependencies

```
./configure --disable-threads --with-petsc=/gpfs/runtime/opt/petsc/3.0.0-p12 --with-petsc-arch=linux-gnu-
cxx-opt --with-umfpack --with-trilinos=/gpfs/runtime/opt/trilinos/10.2.2 --with-metis=/gpfs/runtime/opt/
metis/4.0.1 --with-blas=goto2 --with-lapack=goto2 --with-p4est=/gpfs/runtime/opt/dealii/7.0.0/p4est --with-
mumps=/gpfs/runtime/opt/mumps/4.9.2 --with-scalapack=/gpfs/runtime/opt/gotoblas2/1.13/lib --with-blacs=/
gpfs/runtime/opt/gotoblas2/1.13/src/BLACS --enable-mpi CC=mpicc CXX=mpiCC LDFLAGS=-L/gpfs/runtime/opt/
gotoblas2/1.13/lib
```

Multiple hardware setups

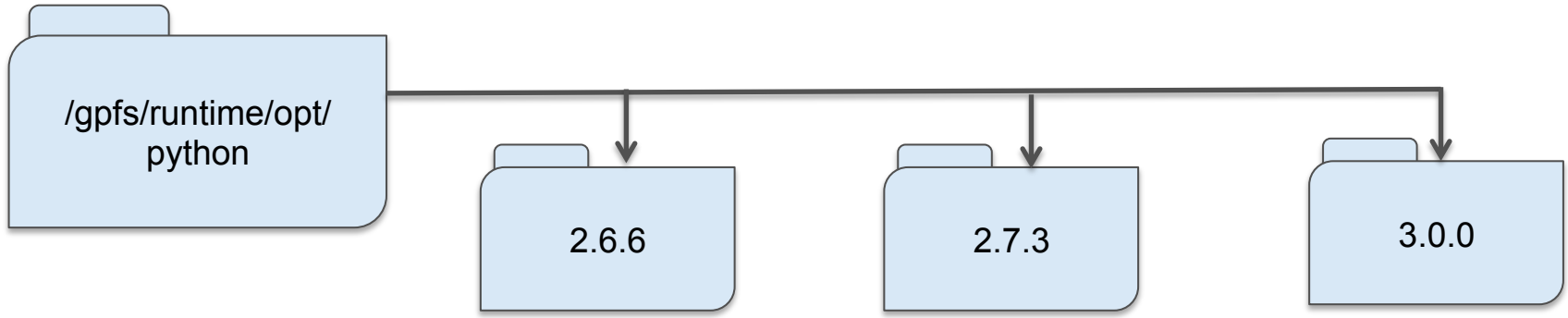




Pymodules

BROWN UNIVERSITY CENTER FOR COMPUTATION & VISUALIZATION

Pymodules: Our enhanced Environment Modules* system designed to mitigate the above issues.



Without modules	With modules

`PYTHON_ROOT=/gpfs/runtime/opt/python/2.7.3`
`export PATH=$PYTHON_ROOT/bin:$PATH`
`export LD_LIBRARY_PATH=$PYTHON_ROOT/lib:$LD_LIBRARY_PATH`
`export MANPATH=$PYTHON_ROOT/share/man:$PYTHON_ROOT/man:$MANPATH`
`export PKG_CONFIG_PATH=$PYTHON_ROOT/lib/pkgconfig:$PKG_CONFIG_PATH`
…
	`module load python/2.7.3`

Pymodules is…

User-friendly **Flexible** **Fast**

Pymodules relies on modulefiles

BROWN UNIVERSITY CENTER FOR COMPUTATION & VISUALIZATION

Modulefiles: small, INI-style text files for each application specifying changes to be made to the environment, for each version (example: Boost C++ modulefile)

```
[DEFAULT]

brief = Boost - The Boost C++ Libraries
url = http://www.boost.org/
category = libraries

prepend CPATH = %(rootdir)s/include
prepend LIBRARY_PATH = %(rootdir)s/lib
prepend LD_RUN_PATH = %(rootdir)s/lib

set BOOST = -L%(rootdir)s/lib
set BOOST_DIR = %(rootdir)s
set BOOST_ROOT = %(rootdir)s

[1.49.0]
default = true

[1.52.0]
prepend CPATH= %(rootdir)s/src/boost_1_52_0

[1.40.0]
```



metadata

common environment settings

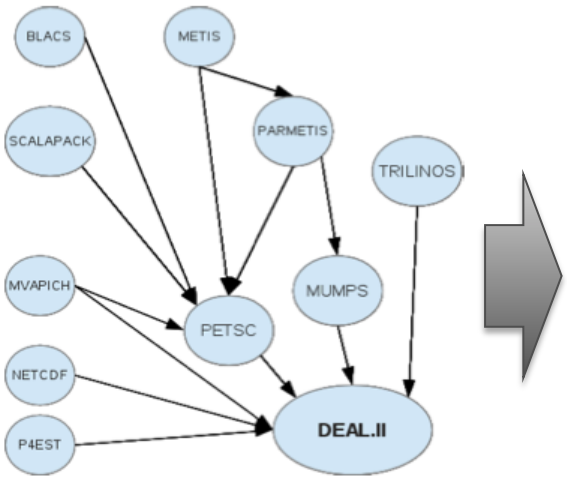
all versions present on system

Using modules in the software installation process

BROWN UNIVERSITY CENTER FOR COMPUTATION & VISUALIZATION

Build time issues

Software can depend on (many) other softwares in the build process, leading to very complicated builds.




```
./configure --disable-threads --with-petsc=/gpfs/runtime/opt/petsc/
3.0.0-p12 --with-petsc-arch=linux-gnu-cxx-opt --with-umfpack --with-
trilinos=/gpfs/runtime/opt/trilinos/10.2.2 --with-metis=/gpfs/runtime/
opt/metis/4.0.1 --with-blas=goto2 --with-lapack=goto2 --with-p4est=/
gpfs/runtime/opt/dealii/7.0.0/p4est --with-mumps=/gpfs/runtime/opt/
mumps/4.9.2 --with-scalapack=/gpfs/runtime/opt/gotoblas2/1.13/lib --
with-blacs=/gpfs/runtime/opt/gotoblas2/1.13/src/BLACS --enable-mpi
CC=mpicc CXX=mpiCC LDFLAGS=-L/gpfs/runtime/opt/gotoblas2/1.13/lib
```

Solution:

Dependency modules set LIBRARY_PATH and CPATH.

```
./configure --disable-threads --with-petsc-arch=linux-gnu-cxx-opt --with-umfpack --enable-mpi
CC=mpicc CXX=mpiCC
```

Using modules in the software installation process, cont.

BROWN UNIVERSITY CENTER FOR COMPUTATION & VISUALIZATION

Runtime issues

Modules can depend on other modules' libraries @ runtime

Goal: Make things simple as possible for the user

	User (@ Runtime)	**Admin (@ build time)**

Make user load dependency modules (LD_LIBRARY_PATH)	`module load deal.ii`
	`module load openmpi`
	`module load petsc`
	…
		`module load deal.ii`
		`module load openmpi`
		`module load petsc`
		…

Dependencies to module X set LD_RUN_PATH	`module load deal.ii`	`module load deal.ii`
		`module load openmpi`
		`module load petsc`
		…

Using modules to organize software inventory

BROWN UNIVERSITY CENTER FOR COMPUTATION & VISUALIZATION

Usability issues

Finding the application you want out of 500+ options is a challenge.

Use a database, not files, to keep track of all installed software.

Runtime of "module avail" command:

Files	Scales up w/# of software packages installed
Database	Constant

Using modules to organize software inventory

BROWN UNIVERSITY CENTER FOR COMPUTATION & VISUALIZATION

Modules offer the following features:

-Search by package name/version (can also search by regex)

-Tab completion

-Store metadata: category of software…date last loaded, etc

```
[user@ccv ~]$ module avail
~~~~~~~~~~~~category: Math ~~~~~~~~~~
atlas/3.10.1  sage/5.11
~~~~~~~~~~~~category: bio ~~~~~~~~~~~
abyss/1.3.4                    macs/1.4.2
agalma/0.3.0                   macs/2.0.10
…
```

Using modules for performance optimization

BROWN UNIVERSITY CENTER FOR COMPUTATION & VISUALIZATION

Performance issues

-Heterogeneous hardware

- -AMD vs. Intel
- -SIMD instruction set
- -Processor Model

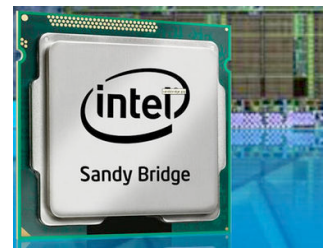
Solution:

-Optimized modules for different architectures

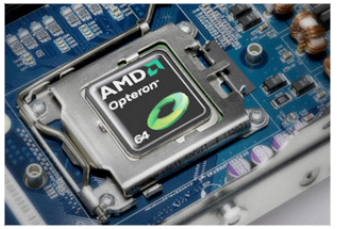
Example:

LLCBench benchmark linked against Intel and AMD linear algebra routines…





Comparison of BLAS performance under ACML and MKL. AMD machine, 4 GB memory, 1 CPU core



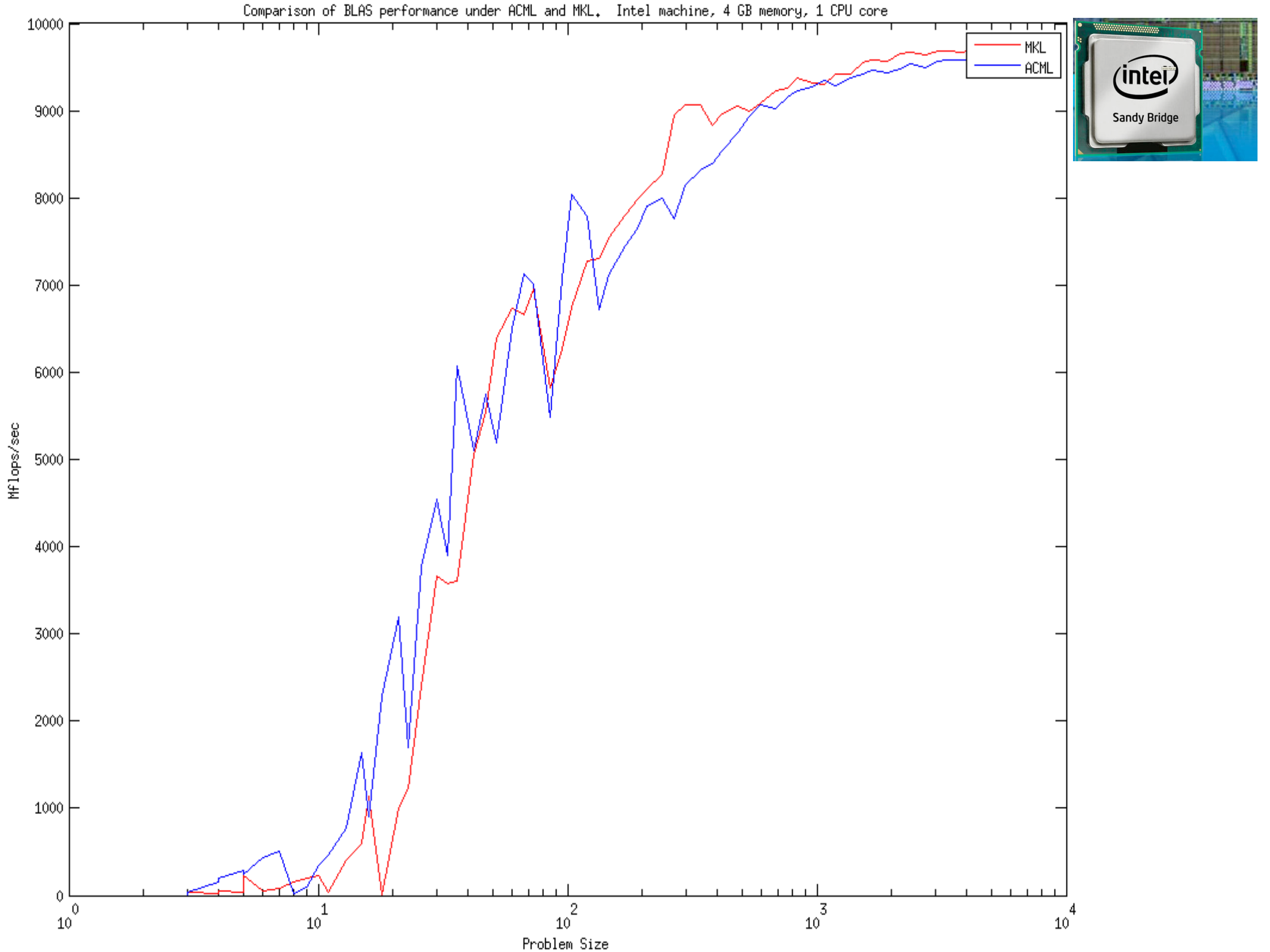
MKL
ACML

Mflops/sec

5000
4500
4000
3500
3000
2500
2000
1500
1000
500
0

10^0 10^1 10^2 10^3 10^4

Problem Size


Comparison of BLAS performance under ACML and MKL. Intel machine, 4 GB memory, 1 CPU core
MKL
ACML
Mflops/sec
Problem Size
10000
9000
8000
7000
6000
5000
4000
3000
2000
1000
0
10^0
10^1
10^2
10^3
10^4
intel
Sandy Bridge

Pymodules vs other tools

BROWN UNIVERSITY CENTER FOR COMPUTATION & VISUALIZATION

Environment Modules (original):
-Tcl syntax
-Uses files to store module information

CDE (Guo):
-Build once, run everywhere
-Not optimized for specific hardware or libraries

Traditional package managers
-Not good at managing multiple versions of same software.

yum
yellowdog updater modified

apt-get



Conclusion

BROWN UNIVERSITY CENTER FOR COMPUTATION & VISUALIZATION

-Software modules can maintain multiple software installations on shared system.

-Can improve ease of software installation and use, performance gains

-Pymodules

Further work: Integrating the install process

-Pymodules is available at http://www.bitbucket.org/mhowison/pymodules