Python Usage Metrics on Blue Waters

Colin MacLean (NCSA/University of Illinois)
Email: cmaclean@illinois.edu
Python on Blue Waters is maintained via a patched version of Gentoo Prefix

- Over 650 packages installed (230 Python)
- Includes non-Python dependencies
- Uses Cray MPICH and libsci
- Patches to Gentoo Portage enable chaining of prefixes
- Eselect module for selecting Environment Modules
- Gentoo Portage made aware of Environment Modules provided dependencies
- Most ebuilds work out of the box
Use of Gentoo Prefix on Blue Waters

Blue Waters Python prefix hierarchy:

- **Portage**
  - Invisible to users; Python 2.7/3.4; Package management/maintenance utilities

- **BWPY**
  - User visible; CPython 2.7/3.{3,4,5}; Non-mpi config; Most packages installed here

- **BWPY-MPI**
  - MPI functionality enabled; Only on MOM/Compute nodes

- **Tensorflow-0.12.1**

- **Tensorflow-1.0.0**

  Quickly developing software updated more frequently in sub-modules
Gentoo Prefix Environment

- Gentoo Prefix management runs bash with $EPREFIX/etc/profile
- No LD_LIBRARY_PATH to prevent breaking other software
- CPATH/LIBRARY_PATH for correct library location ordering
- Uses RUNPATH
  - May use custom ld-linux.so in future
Module Usage Logging

- Added exit hook in sitecustomize.py
- Errors in sitecustomize.py aren’t fatal
- Logs timestamp, user, host, job, number of nodes, list of modules loaded at exit
- Only sees modules loaded at exit. Exec’d code missed.
- Compute node jobs counted once per aprun
- Collection started January 13th, 2017
Python Versions

- BWPY supports CPython 2.7, 3.3, 3.4, 3.5, Pypy and Pypy3
- Python 2.7 dominates,
- Python 3.3, Pypy, and Pypy3 not used
- ~270,000 invocations
Module Usage: By User

- 131 total users
- Unique module names imported: 663
Module Usage: Most Imported

- 2,391,323 total imports
- Some user provided modules overrode BWPY modules, some weren’t provided by BWPY
Compute Node Job Walltimes

- 10893 jobs < 15 minutes
- 7004 jobs 15-30 minutes
- 6288 jobs 1/2-1 hour
- Only 162 jobs >16 hours
MOM Node Job Walltimes

- 424 jobs < 15 minutes
- 152 jobs 15-30 minutes
- 351 jobs 1/2-1 hour
- 719 jobs 1-4 hours
- 214 jobs >16 hours
Executions per Job

- Some jobs execute Python many times
- Detect and notify of this behavior?
Compute Job Sizes

<table>
<thead>
<tr>
<th># of nodes</th>
<th>Invocations</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>170488</td>
</tr>
<tr>
<td>2-32</td>
<td>8555</td>
</tr>
<tr>
<td>33-64</td>
<td>53</td>
</tr>
<tr>
<td>65-128</td>
<td>222</td>
</tr>
<tr>
<td>129-256</td>
<td>600</td>
</tr>
<tr>
<td>257-512</td>
<td>94</td>
</tr>
<tr>
<td>513-1024</td>
<td>71</td>
</tr>
<tr>
<td>1025-2048</td>
<td>25</td>
</tr>
</tbody>
</table>

- The vast majority of Python jobs were single node.
- How does slow Python startup time affect job width?
Python Usage

• Python used in many ways
  − Login node data analysis
  − MOM node job launching
  − For short compute jobs
  − Some HPC workloads

• Support for different situations
  − Separate MPI-enabled site-packages for non-Compute use
Conclusions

- Python used in many ways
  - Login node data analysis
  - MOM node job launching
  - For short compute jobs
  - Some HPC workloads

- Support for different situations
  - Separate MPI-enabled site-packages for non-Compute use
• Adding new BLAS options due to extensive use of numpy

• May help work around CPython Global Interpreter Lock
Thank You!