HPCBIOS: Getting Your Scientific Software, Users & Documentation in Sync

Eng. Fotis Georgatos <fotis@cern.ch>

Definition of Common Environment for HPC Platforms and Beyond

1st February 2014, FOSDEM

Index

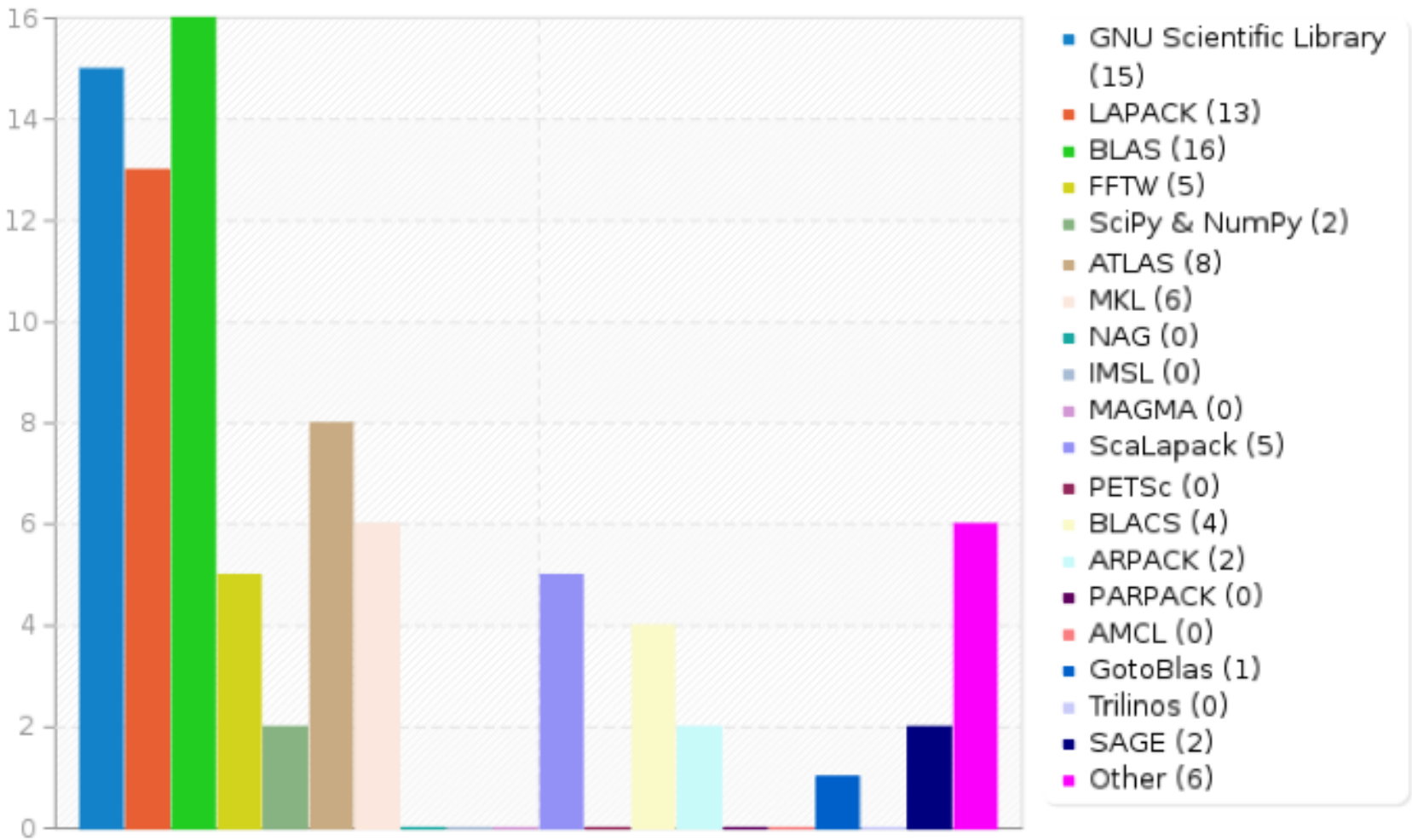
What is HPCBIOS

- HPCBIOS is concerned with:
  - users ability to handle tasks on computational platforms (HPC, Grids, Clouds...),
  - ... in a uniform and painless manner,
  - ... as much as that is technically feasible.
- It is defined at three levels:
  - structured documentation, aimed at scientific software
  - policies automation (bundles, handled via EasyBuild)
  - wrapping code to satisfy the above and provide cheap buildsets
- At present, HPCBIOS activity is apparently more of CC-BY-SA Open Source Documentation, since the synergy with EasyBuild implies most coded part ends up in the later! (the success of this is, we need deal w. Python, not shell code)

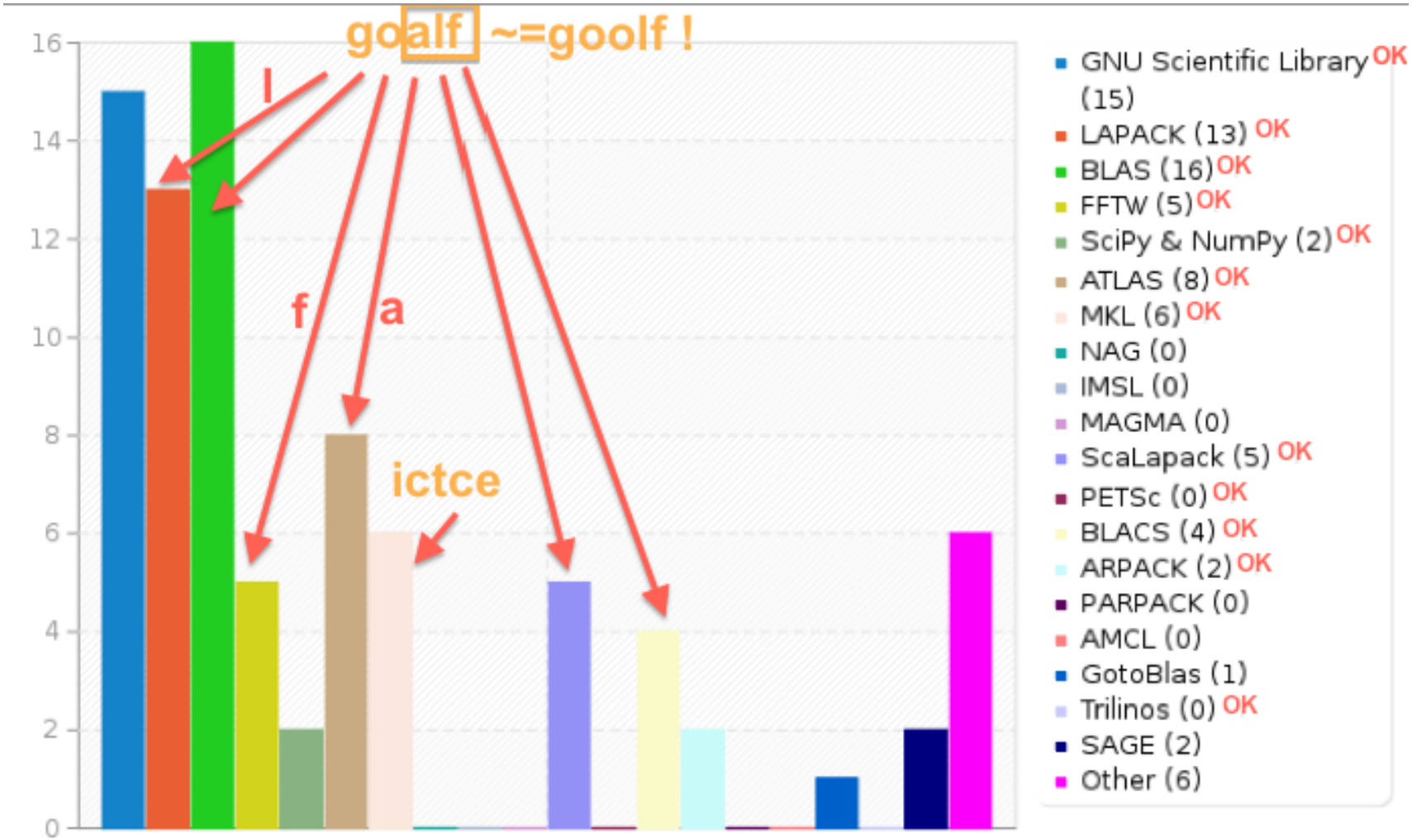
Who may care about HPCBIOS?

- users involved in scientific computing
- experts and sysadmins involved in support of scientific codes
- developers & deployers of EasyBuild ; and *that* is by now the vehicle of choice, for the objectives covering HPC software!

LS2 survey of HPC User needs in foundation software



LS2 survey of HPC User needs in foundation software

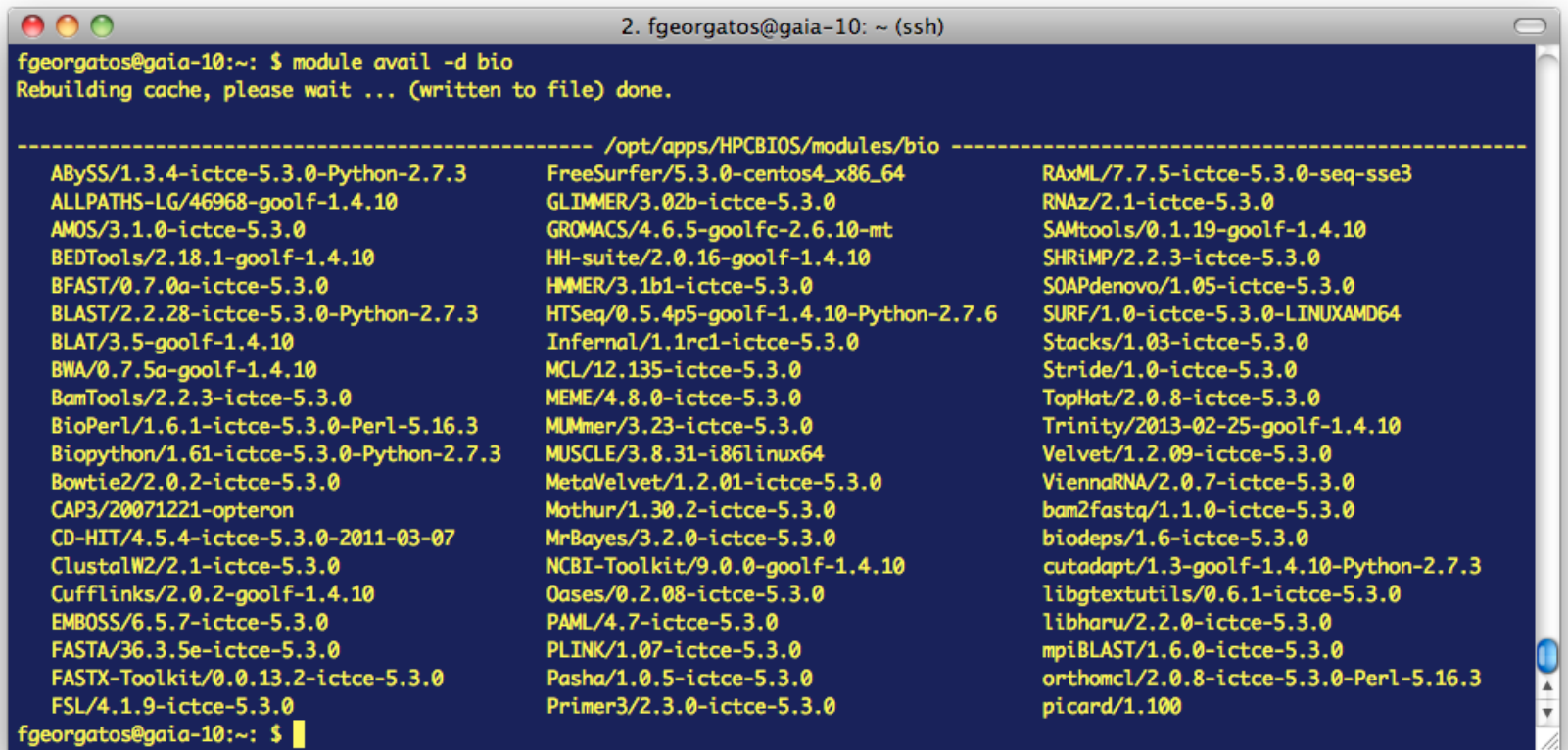


Common stumbling blocks

Ever-present aspects of computational infrastructures that puzzle or delay scientific users are:

- Need of common tools for handling software (tar, gzip/bzip2, autoconf/automake, bison, CMake, ...)
- Need of libraries/software for common Math operations (Linear_Algebra, FFT, GSL, ...)
- Need of software popular with scientific communities (R, Python/numpy/scipy, ...)
- Diversified software versions; this always proves itself to be very essential for differential debugging.

Bioinformatics packages, how to handle?



Objectives

- Freedom to define soft environment at user/group/system levels?
  - This could be feasible by employing modules
- Total reproducibility of software builds?
  - This could be feasible by using EasyBuild
- Handle apparent conflict between agility/stability?
  - Continuity of default APIs when not in maintenance window
  - Perfect rollback or, might even ... roll-forward

HPCBIOS: Effort to improve the soft environment

- HPCBIOS strives to minimize the time people spend with individual sites' or systems' configuration
- Math, Bioinfo & Life Sciences are taken highly into account, but not exclusively - many science domains are considered
- It is about standardization & consistent user experience across systems/sites (even if the site definition can be reduced to ... your laptop)

HPCBIOS: Implementation layers outline

- modules (environment-modules-C/Tcl, Lmod, ...)
- EasyBuild (v1.10 and later versions)
- HPCBIOS Buildsets & HPCBIOS Bundles (aka policies)

HPCBIOS: Buildsets and Bundles

```
fgeorgatos@gaia-10:~: $ module avail HPCBIOS_ HPCBIOS/2013
Rebuilding cache, please wait ... (written to file) done.
                                                          bundles (policies)
---------------------------------- /opt/apps/HPCBIOS/modules/toolchain ----------------------------------
   HPCBIOS_Bioinfo/20130829-goolf-1.4.10              HPCBIOS_Math/20130829-goalf-1.1.0-no-OFED
   HPCBIOS_Bioinfo/20130829-ictce-5.3.0        (D)    HPCBIOS_Math/20130829-goolf-1.4.10
   HPCBIOS_Debuggers/20130829-goolf-1.4.10            HPCBIOS_Math/20130829-ictce-5.3.0          (D)
   HPCBIOS_LifeSciences/20130829-goolf-1.4.10         HPCBIOS_Profilers/20130829-goolf-1.4.10
   HPCBIOS_LifeSciences/20130829-ictce-5.3.0    (D)
                                                                                  buildsets
------------------------------------ /opt/apps/default/modules/all ------------------------------------
   HPCBIOS/20130226    HPCBIOS/20130715-bycategory    HPCBIOS/20131117-bycategory
   HPCBIOS/20130301    HPCBIOS/20130715               HPCBIOS/20131117
   HPCBIOS/20130302    HPCBIOS/20130902-bycategory    HPCBIOS/20131224-bycategory
   HPCBIOS/20130401    HPCBIOS/20130902               HPCBIOS/20131224             (D)
   HPCBIOS/20130501    HPCBIOS/20131004-bycategory
   HPCBIOS/20130601    HPCBIOS/20131004

  Where:
   (D):  Default Module
```

HPCBIOS: Examples

Example common Applications or Tools that are needed and, categories thereof, are:

- HPCBIOS_06-01: Open Source Math Libraries (ScaLAPACK/BLAS APIs, FFTW, GSL)
- HPCBIOS_06-04: Editors and Scripting Tools (Tcl/Tk, Perl, ... etc)
- HPCBIOS_07-02: Performance, Testing and Profiling tools (gprof, Valgrind, PAPI...)
- more: Life Sciences, Bioinformatics, MD, DFT, Climate et al

Are we alone in this Universe?

- Not at all: conglomerations of other HPC sites have also developed similar efforts, to align their environments
- In fact, HPCBIOS is intentionally backwards-compatible with the Baseline Configuration definition, already deployed across 6 HPC sites affiliated to U.S. Department of Defense. Ref. http://centers.hpc.mil/consolidated/bc Compliance Matrix
- The concept is quite common in 3rd-party collaborating projects (fi. PRACE/DEISA Common Production Environment) but it is not always documented all that much, on per site basis; can we improve on this gap?

Contributing back

Which type of software code helps in the described objectives and where/how to contribute to it?

- Use modules - any technology variety that suites you is OK
  - Organize even your personal customized software via modules!
- Use EasyBuild - the best and only alternative for its task ;-)
  - Try to provide easyconfigs/easyblocks and issue PRs on github!
- Contribute documentation and/or scripts back to HPCBIOS, if/when solutions do not fit above
  - Solution is relevant when it's applicable across 2 sites/systems
  - Issue PR on github as soon as you get something ready to share!

Current challenges & ongoing work; audience feedback?

- Unknown territory: which other "policy" objectives matter?
- We are ab/using EasyBuild's toolchains at present for bundles; that constraints functionality to homogeneous targets (fi. only Intel builds); this approach is more of a hack and may need be revisited.
- Delivering complex buildsets, with "properties" (bigmem, AVX, GPUs, MIC...); how do you schedule that, starting from point-zero, optimally?
- Certain components do auto-tuning and may require node-exclusive runs upon build phase: ATLAS, FFTW
- Python's LooseVersion and modules/Lmod's equivalent do not always align; fi. Lmod assumes order: 2.4dev1<2.4a1 <2.4rc2<2.4<2.4-1<2.4.1 . This can lead to surprises on the user-side, because "default" can diverge from expected one.

References

- Homepage: http://hpcbios.readthedocs.org/en/latest/
  - Look at HPC_Baseline_Configuration.html
- github location: https://github.com/fgeorgatos/HPCBIOS
- How to use it, where to go from here:
  - eb HPCBIOS_Math-20130829-goolf-1.4.10.eb -r yes, it is that simple!
  - Read the HPCBIOS objectives, embrace and extend :)

Thank you

What HPCBIOS covers, now and later

- Multiple-Version Software
- Open Source Math Libraries (GSL, FFTW, Linear Algebra)
- Multiple-Version Software Access via Modules
- Open Source Performance and Profiling Tools
- Common Open Source High Productivity Languages
- Domain specific policies (Bioinformatics, MD/DFT, Climate...)

Many more are/can be available, the above is just a a shortlist!

Presentation objectives

- present ongoing efforts and concepts tried in centers located in the EU & US, streamlining the user experience in scientific computing
- promote the concept of buildsets, for delivering scientific software and, bundles to keep sets of software manageable
- probe the interest of the community for current needs and future work

Problems & Solutions

Common Areas for Improvement which may be encountered on scientific platforms:

- missing tools common in a Unix environment (shells, version control, archivers etc)
  - Solution: look at HPCBIOS_2012-90, it defines a few dozens of the relevant bits
- monolithic design of software environment (ie. hard dependencies)
  - Solution: use modules to Boost your flexibility; rant intended :)
- Required -ancient- compilers which have now lost compatibility with modern software
  - Solution: compile an array of GCC versions (and more compilers if you can)