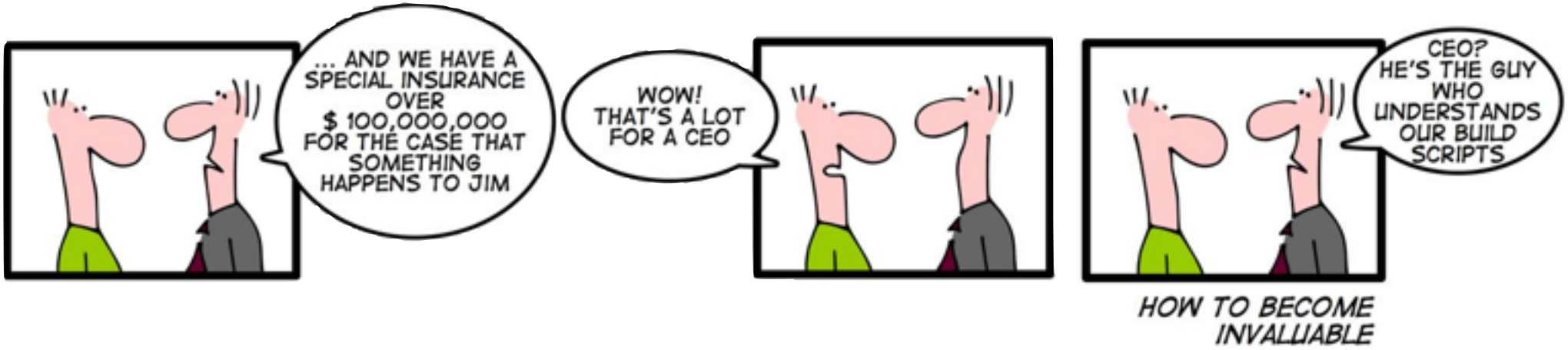
Getting Scientific Software Installed
Tools & Best Practices



Supercomputing'13 Birds-of-a-Feather
November 19th 2013

kenneth.hoste@ugent.be

andy.georges@ugent.be

http://geekandpoke.typepad.com/geekandpoke/2010/05/how-to-become-invaluable.html

- part of central IT department (DICT) of Ghent University, Belgium
- central contact for HPC at Ghent University, Belgium
- member of Flemish supercomputer centre (VSC)
  - collaboration between Flemish university associations

- six Tier2 systems, one Tier1 system (~1.3K servers)
  - Tier1: #163 in Top500 (Nov. 2012), now #306
- team currently consists of 8 FTEs, for ~500 users
- tasks include system administration of HPC infrastructure, user training, user support, ...

Getting Scientific Software Installed
Tools & Best Practices
Outline

- lightning talks
  - **Lmod** (Robert McLay, TACC)
  - **HashDist** (Andy Terrel, TACC)
  - **EasyBuild** (Andy Georges, UGent)
- show-of-hands and a couple of key topics
- open discussion
  - what are the major issues (for you)?
  - which tools are you using, and would you recommend?
- let's join forces...

Show of hands (setup)

- Go to **socrative.com**

  (use your laptop, smartphone, tablet, ...)
- Click ‘Student Login’
- Enter the room number: **570181**
- Participate!

Who are you?

- scientific software developer
- researcher / end-user of scientific software
- system administrator
- member of user support team
- manager
- other?

socrative.com
570181

Which modules tool do you use?

(tip: if you're not sure, check the output of "type module" or "which module")

- C environment modules
  - 'modulecmd' command
- Tcl environment modules
  - 'modulecmd.tcl' script
- Lmod
- no modules tool
- something else?

socrative.com
570181

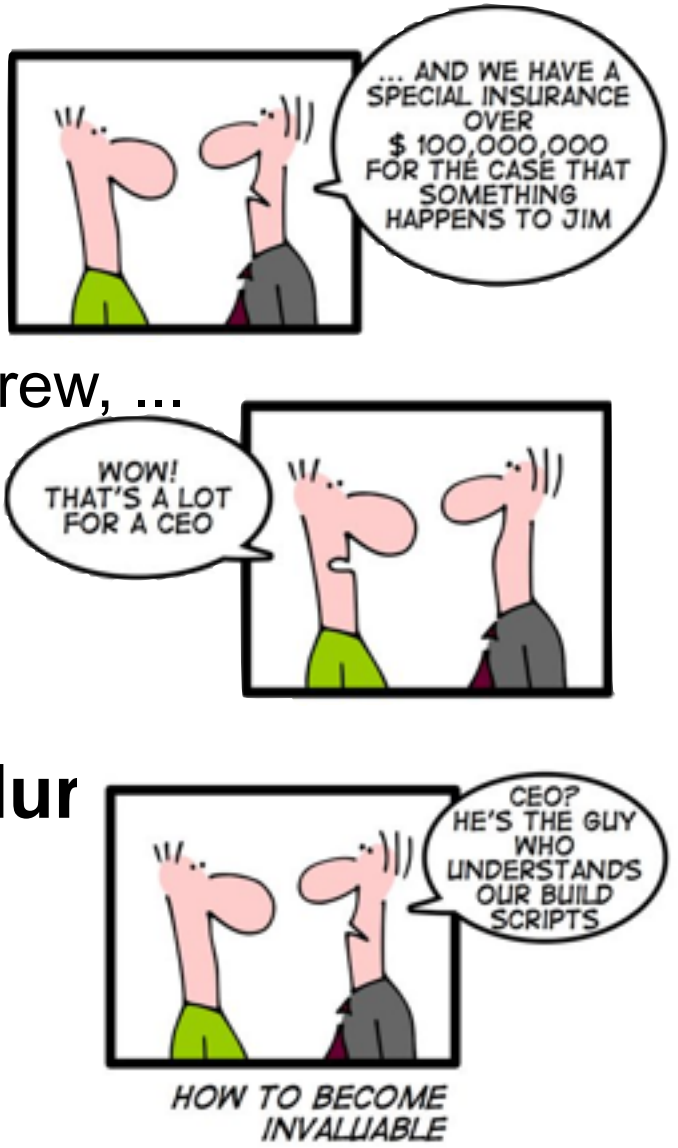
Which module naming scheme do you use?

- **flat scheme**
  - ‘module list’ show *all* the available modules
- **hierarchical / tree scheme**
  - ‘module list’ only shows compilers
  - ‘module load <compiler>’ first
  - then ‘module load <MPI>’, ‘module load <software>’
- **something else?**

socrative.com
570181

Tools for building/installing scientific software

- **(bash) scripts, Makefiles**
- **wrappers**
  - Portage, Ports, HomeBrew, linuxbrew, ...
- **packages**
  - RPMs, .deb, ...
- **well-documented build procedur**
- **‘that guy’ (Jim)**
- **other?**



http://geekandpoke.typepad.com/geekandpoke/2010/05/how-to-become-invaluable.html

socrative.com
570181

Best practices, which ones do you use?

- **collaboration with other HPC sites** (w.r.t. installing scientific software)
- **automation of builds**
  - in some way or another (not Jim)
- **auto-generated module files**
- **providing multiple builds of the same software**
  - different versions, building with different compilers/MPI libraries
- **testing of the software installation**
  - simple: make sure everything is there (binaries, libraries, header files, ...)
  - thoroughly: using well-defined tests, verification of results, ...
- **performance evaluation** (post-build)**, performance monitoring** (over time)
- **keeping track of build 'metadata'**
  - build procedure, build log, patch files, build time, built by, dependencies, ...
- **keeping repository of sources**
  - to remedy disappearing upstream sources

socrative.com
570181