Towards a full Spack / Spackdev – based build ecosystem for art-based experiments

Chris Green, FNAL, 2 May 2018
Current Efforts

– Produce an “MVP” – Minimum Viable Product:
  • Software stack with one OS / compiler / C++ standard / optimization level to allow experiments to, well, experiment.
  • Keep track of issues along the way but achieve the narrow goal first and go back for the others later.
  • “Pinnacle” of the software stack: Root.
  • Everything built “our way” to maximize realism & compatibility for experiments.
  • Minimize use of system-available packages via `packages.yaml` where possible.
  • First demonstration of `cetmods`.
  • NOT “release”-oriented.
  • NOT a solution to every problem.
  • NOT a guarantee that every remaining problem can be solved.
  • NOT a collection of every piece of software every experiment is likely to need.
Progress so far

• Packages:
  – **boost**, required Spack enhancement (PR in queue).
  – **mariadb, postgresql** with **–client** bases and virtual packages (PRs 1, 2 in queue).
  – **XRooT**D (PR merged) – no C++ std specification yet.
  – **TBB** (PR under discussion) – no C++ std specification yet.
  – **Root** in progress, filling in other dependencies to minimize internally—built packages.

• Common threads:
  – Even when specs already exist, dependency connections often missing, other inadequacies (e.g. Python).
  – Existing specs not even close to flexible enough, in general.
Problems

• Adding a variant changes the hash, even if the default setting is off.
• Significant problems with concretization, really obvious as variants are added / specified. Change in the works, but timescale and ability to solve our particular issues unknown at this time.
• No version specs: with current iteration of Spack, this will result in permanently-forked, very local specs and issues with concurrent release visibility / sharing.
• C++ standard a problem, especially with concretization: cxxflags a possibility, except for the fact that many packages (e.g. boost, tbb, root) need to be told explicitly.
• GPL encumbrance (readline, gettext, gsl).
• Not even thought about optimization levels.
• Not worrying about relocate-ability / buildcache yet.