Continous Integration in HPC

CI Problems

- Normal CI systems assume architecture, OS variants are irrelevant
    - Hosted: single or small number of environments
    - Local: management nightmare
- HPC systems include unorthodox environments
    - ARM, BG/Q, GPU nodes, cross-compiles
    - Multiple different non-portable ways to launch MPI jobs
    - Needs support from HPC admins
- Trust
    - Outside contributions
    - External dependencies
- Parallel testing
- How to share existing work with other groups?
- Visualizing high-dimensional test results and specifying high-dimensional test suites

- Normal CI systems assume architecture, OS variants are irrelevant

- Cloud: single or small number of environments
  - Can cover Linux, OS X, Docker (Travis), Windows (AppVeyor)
  - Can not cover architectures, kernel versions
  - Can not host commercial software easily?
  - Only good for smoke tests on standard environments
- Non-cloud: management nightmare
  - Requires maintenance
  - Label agents for testing dimensions
  - Label agents, jobs, users, build plans for security—may require customization of the CI system
  - Trust of code, users, external authentication
  - Fortify/sandbox local build resources
  - Needs dedicated manpower
  - Can leverage labs/OpenHPC to support other projects when they have resources

- HPC systems include unorthodox environments

- Multiple different non-portable ways to launch MPI jobs

  - Build some common interface for MPI job launch

  - Be sure to work together so interface stays common

Sharing build infrastructure

- Cloud CI: typically a .yml or similar config file, easily sharable
- Non-cloud CI: export scripts or other job descriptors
  - May need to share at a higher level (design)
  - Keep as much as possible in scripts in-repo

Automating parallel build and test of large DAGs

- LLNL has solution based on POSIX flock (SPACK)
- Job dependencies
  - Machine's job system
  - CI job system

- Visualizing high-dimensional test results and specifying high-dimensional test suites

- Hierarchy of test dimensions
  - Common projections onto small number of interesting dimensions

- Test output filtering

- Public dashboards