LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

From makefile to spack

for the Belle II externals

Alexander Lory

Outline

- Situation
- Goal
- Implementation
- Personal experience

Belle 2 “externals”

- We need 67 external software packages :
  gcc, root, geant4, ... and 23 python packages.
- Compiled using a makefile.
- Makefile creates relocatable binaries

Makefile

- One makefile builds all the “externals”
- Build in the correct order to respect dependencies
- We recompile everything when we update

Goal

Goal:

- Have a tool to manage the externals
- Smarter updates
- Keep relocatability

Implementation

How spack is used

- Split the makefile into spack packages
- One “dummy” package to install everything and create a filesystem view :

  “spack install externals”
  “spack view symlink externals-view externals”

```
depends_on("gcc@6.3.0")
depends_on("binutils")
depends_on("neurobayes")
depends_on("millipede2")
depends_on("belle-legacy")
depends_on("scons")
depends_on("nsm2")
depends_on("evtgen")
depends_on("valgrind@3.11.0")
depends_on("libxml2@2.9.4")
depends_on("libxslt@1.1.29")
depends_on("zlib@1.2.8")
depends_on("bzip2@1.0.6")
depends_on("freetype@2.6.3")
depends_on("sqlite@3.1.3")
depends_on("gdb@7.11")
depends_on("cmake@3.5.2")
depends_on("googletest@1.8.0")
depends_on("boost@1.63.0")
depends_on("cppcheck@1.75")
depends_on("doxygen@1.8.11")
depends_on("clhep@2.2.0.4")
depends_on("geant4@10.01.p02")
depends_on("postgresql@9.2.4")
depends_on("xrootd@4.3.0")
depends_on("root@6.08.00")
depends_on("fastbdt@3.2")
depends_on("vgm@4.3")
depends_on("rave@0.6.25")
depends_on("hepmc@2.06.09")
depends_on("pythia@8215")
depends_on("photos@3.56")
depends_on("tauola@1.1.4")
depends_on("mg5-amc@2.2.2")
depends_on("exrootanalysis@1.1.2")
depends_on("cry@1.7")
depends_on("belle-flc")
depends_on("eigen@3.2.8")
depends_on("curl@7.49.1")
depends_on("git@2.9.0")
depends_on("fann@0")
depends_on("astyle@2.05.1")
depends_on("python@3.5.2")
depends_on("py-autopep8")
depends_on("py-cherry")
depends_on("py-cycler")
depends_on("py-decorator")
depends_on("py-ipython@4.0.0")
depends_on("py-ipython-genutils")
depends_on("py-lxml")
depends_on("py-matplotlib")
depends_on("py-numpy")
depends_on("py-path")
depends_on("py-pep8")
depends_on("py-pexpect")
depends_on("py-pickleshare")
depends_on("py-pip")
depends_on("py-parsing")
depends_on("py-dateutil")
depends_on("py-tz")
depends_on("py-requests")
depends_on("py-setuptools")
depends_on("py-setuptools-scm")
depends_on("py-simplegeneric")
depends_on("py-six")
depends_on("py-traitlets")
```

Relocatability in Conda

- Rewrite rpaths to be relative (patchelf, ...)
- Any file containing the build prefix is registered.
  Use this information when installing the binary.

→ Use the same method for spack

Personal Experience

- 6 months of work at 8-15 h/week include:
  - learning about software in general
  - learn how to use spack
  - look into conda
  - write the packages

- For me, the big part of the work to write packages is to look at other people's packages

  I used a lot of packages from the HEP github repository :

  `https://github.com/HEP-SF/hep-spack`

Wanted improvements

- **Updating spack** caused trouble
- **Patches** are applied before you can manipulate files. Would be nice to apply them whenever we want

Future of the project

- Martin Ritter will take over the project
- Public repository soon