Spack Package Manager

Proseminar Python im Hochleistungsrechnen

Hauke Sommerfeld

Arbeitsbereich Wissenschaftliches Rechnen
Fachbereich Informatik
Fakultät für Mathematik, Informatik und Naturwissenschaften
Universität Hamburg

2019-05-13

Gliederung (Agenda)

Was ist Spack?

Spack is a Package Manager for supercomputers, Linux and macOS *(spack.io)*

- Ausgelegt für die Benutzung auf wissenschaftlichen Systemen / Clustern
- Alternative zu herkömmlichen Package-Managern, deren Features in wissenchaftlichen Umgebungen an ihre Grenzen stoßen oder unpraktisch sind

Wichtige Abgrenzung

Python Package $\neq$ Spack Package

- PIP installiert Python Pakete / Bibliotheken
- Spack installiert reguläre Software auf Betriebssystemebene

Wer benutzt Spack?

Beispiel:

Github: ~960 Stars / ~400 Entwickler

Wie wird Software auf herkömmlichem Wege installiert?

- macOS/Windows: App Store, manuell oder third-party Paketmanager
- Linux/Unix: Paketmanager (Standard) oder "from source"

Generelles Problem:

- Wissenschaftliche Anwendungen erfordern oft spezielle Versionen und sind hardwarenah programmiert
- Herkömmliche Paketmanager stellen Binaries bereit
- Paketmanager arbeiten destruktiv

Vorteile gegenüber anderen Paketmanagern

- ✓ Spack ist ein Paketmanager :-)
- ✓ Installation via Package, aber standardmäßig aus dem Quellcode ⇒ hardwarespezifisch
- ✓ Koexistenz verschiedener Programmversionen/Konfigurationen möglich und fester Bestandteil ⇒ nicht-destruktiv

Vorteile gegenüber anderen Paketmanagern (Forts.)

- ✓ Spack selber und alle Paketinformationen/Installationsskripte sind in Python geschrieben ⇒ erweiterbar/anpassbar
- ✓ Einfache Einbindung bestimmter Compiler, Programmversionen, etc. (mehr dazu später)

Installation

- Spack ist auf GitHub verfügbar
- Installation per git clone
- Einrichten der Umgebungsvariablen durch mitgeliefertes Skript

```
1 ~$ git clone https://github.com/spack/spack.git
2 ~$ source spack/share/spack/setup-env.sh
```

- Anschließende Benutzung durch den Befehl

  `spack`

Aufbau der Spack-Installation

- bin — Spack-Binaries
- etc — Konfigurationsdateien, etc.
- lib — Benötigte Bibliotheken, Dokumentation, etc.
- opt — Installationsverzeichnis für Pakete
- share — Hilfreiche Skripte und Utilities
- var — Build- und Package-Dateien

Die Ordnerstruktur von Spack ist auf die Einbindung in eine Unix-/Linux-Installation ausgelegt.

Anzeigen aller verfügbaren Pakete

- Befehl:

  `spack list`

```
-bash-4.2$ spack list
==> 3121 packages.
abinit              multitail           r-blob
abyss               multitime           r-blockmodeling
accfft              multiverso          r-bookdown
ack                 mummer              r-boot
[...]               [...]               [...]
```

Informationen zu einem Paket

- Befehl:

  `spack info <package>`

```
-bash-4.2$ spack info zlib
Package:   zlib

Description:
    A free, general-purpose, legally unencumbered
    lossless data-compression library.

Homepage: http://zlib.net

Tags:
    None

Preferred version:
    1.2.11     http://zlib.net/fossils/zlib-1.2.11.tgz
```

Abhängigkeiten eines Paketes

- Befehl:

  `spack spec <package>`

```
-bash-4.2$ spack spec gdbm
Input spec
--------------------------------
gdbm

Concretized
--------------------------------
gdbm@1.18.1%gcc@4.8.5 arch=linux-centos7-x86_64
    ^readline@7.0%gcc@4.8.5 arch=linux-centos7-x86_64
        ^ncurses@6.1%gcc@4.8.5~symlinks~termlib [...]
            ^pkgconf@1.5.4%gcc@4.8.5 arch=linux-[...]
```

Installieren eines Paketes

- Befehl:

  `spack install <package>`

  (Codebeispiel auf der nächsten Folie)

Installieren eines Paketes (Forts.)

```
1  -bash-4.2$ spack install zlib
2  ==> Installing zlib
3  ==> Searching for binary cache of zlib
4  ==> No binary for zlib found: installing from
       ↪ source    /zlib-1.2.11.tar.gz
5  ==> Staging archive:
       ↪ [...]/var/spack/stage/zlib-1.2.11-<hash>.tar.gz
6  ==> Created stage in
       ↪ [...]/var/spack/stage/zlib-1.2.11-<hash>
7  ==> No patches needed for zlib
8  ==> Building zlib [Package]
9  ==> Executing phase: 'install'
10 ==> Successfully installed zlib
11   Fetch: 0.03s.  Build: 3.23s.  Total: 3.26s.
12 [+] [...]/spack/opt/spack/<os>/<compiler>/zlib-[...]
```

Legende: [...] = Auslassung; <...> = Element

Auflisten aller installierter Pakete

- Befehl:

  `spack find`

```
1 -bash-4.2$ spack find
2 ==> 4 installed packages
3 -- linux-centos7-x86_64 / gcc@4.8.5 ------------------
4 libsigsegv@2.11  m4@1.4.18  zlib@1.2.11
5
6 -- linux-centos7-x86_64 / ncc@2.1.0 ------------------
7 zlib@1.2.11
```

Deinstallieren eines Paketes

- Befehl:

  `spack uninstall <package>`

```
1 -bash-4.2$ spack uninstall m4
2 ==> The following packages will be uninstalled:
3
4         -- linux-centos7-x86_64 / gcc@4.8.5 ----------
5         hhsvnbb m4@1.4.18%gcc patches=[...]
6 ==> Do you want to proceed? [y/N]
```

Versionen und Konfigurationen anpassen

Wir möchten möglicherweise...

- eine bestimmte Paketversion installieren
- einen bestimmten Compiler verwenden
- Abhängigkeiten hinzufügen
- Compile-Time Konstanten setzen

Möglichkeiten, dies zu tun:

- Befehle mit weiteren Optionen aufrufen
- Spack Package per Hand anpassen

Versionen und Konfigurationen anpassen (Forts.)

In den vorher besprochenen Befehlen können die Paketnamen um bestimmte Suffixe erweitert werden.

```
glib@2.60.1, zlib%gcc+debug,
openfoam@1812%ncc^openblas
```

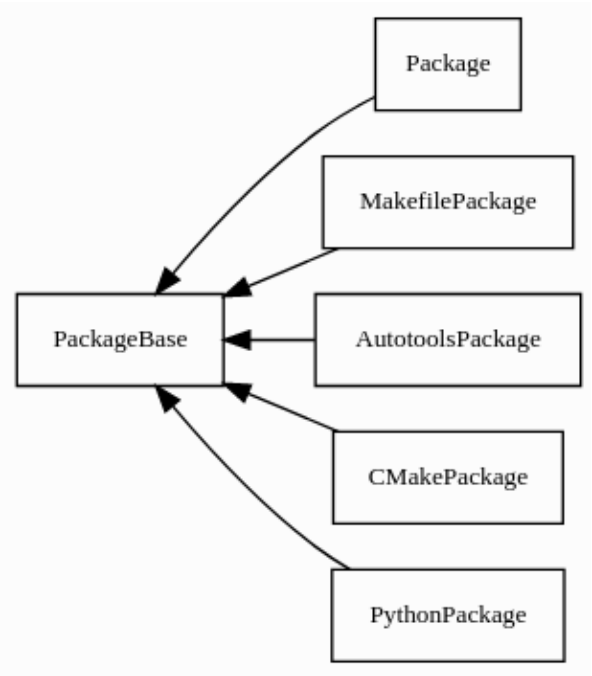
- `@` - Version
- `%` - Compiler
- `^` - Paketabhängigkeit
- `+` - Boolean Compile-Time Konstante (Flag)
  - `<name>=...` - Sonstige Konstanten

Einblick in Packages

- Spack-Package realisiert als eigenes Python-Skript:
  `var/spack/repos/builtin/packages/<name>/package.py`
  - `package.py` enthält die Grundfunktionalität und erbt von der Python-Klasse Package
  - Kann beliebig komplexen Python-Code ausführen
- Erstellen per

  `spack create <package>`

Einblick in Packages (Forts.)



- Alle Packages erweitern die Klasse Package
- Benutzerdefinierte Packages erweitern i.d.R. eine der Subklassen

Einblick in Packages (Forts.)

```
1  from spack import *
2
3  class Helloworld(Package):
4      """Package description"""
5
6      homepage = "http://www.example.com"
7      url      = "helloworld"
8
9      version('1.2.3', '0123456789abcdef01234[...]')
10
11     depends_on('foo')
12
13     def install(self, spec, prefix):
14         make()
15         make('install')
```

Zusammenfassung

- Spack ist ein alternativer Paketmanager für Supercomputer
- Spack vereinfacht die oft sehr komplexe und mühsame Installation wissenschaftlicher Software
- Spack ist durch seine offene, in Python umgesetzte Implemmentation sehr leicht anzupassen

Zusammenfassung (Forts.)

Noch einmal die grundlegenden Befehle im Überblick:

- `spack list` — Auflisten aller verfügbaren Packages
- `spack info` — Informationen zu einem Package
- `spack spec` — Abhängikeitsbaum eines Packages
- `spack install` — Installieren eines Packages
- `spack find` — Auflisten aller installierten Packages
- `spack uninstall` — Deinstallieren eines Packages

Literatur

- Spack Projektseite,
  URL: https://spack.io/,
  Zugriff am 12. Mai 2019
- Spack 0.12.1 Dokumentation,
  URL: https://spack.readthedocs.io/en/latest/index.html,
  Zugriff am 12. Mai 2019
- Todd Gamblin auf Twitter,
  URL:
  https://twitter.com/tgamblin/status/1085938737924788229,
  Zugriff am 12. Mai 2019