Deployment of High Energy Physics software with a standard method: HEPrpms

Thomas Hahn[a] and Andrii Verbytskyi[b]

[a]Max-Planck-Institut für Physik, 80805 Munich, Germany. hahn@feynarts.de
[b]Max-Planck-Institut für Physik, 80805 Munich, Germany. andrii.verbytskyi@mpp.mpg.de

For the ACAT2021 Conference, 28 November - 03 December 2021, IBS Science Culture Center, Daejeon, South Korea (remote)

Introduction and goals

The modern studies in theoretical physics, experimental and phenomenology of High Energy physics (HEP) require a significant amount of specialised software. The creation of suitable computing environment and setup of required software requires significant efforts, time and creates a barrier for newcomers. even if the software would be used unmodified.
In the recent years, these problems were attempted to be solved with a creation of different isolated environments, software managing tools or different combinations of those.
However, the available solutions have multiple disadvantages and practically not always result in a sufficient reduction of maintenance efforts, require users to learn each solution from scratch and overwhelmingly lack multiple important features.
Therefore, in our work avoid implementation of custom software deployment system or isolated environments. Instead, we concentrate on the practical application and development of the existing "standard" solutions of the software deployment for the purposes of theoretical, experimental and phenomenology HEP.
In our contribution we show the application of this approach in the Linux/GNU OS distributives of Fedora, RHEL and SUSE families using corresponding standard tools and present a software repository HEPrpms.

Trivia

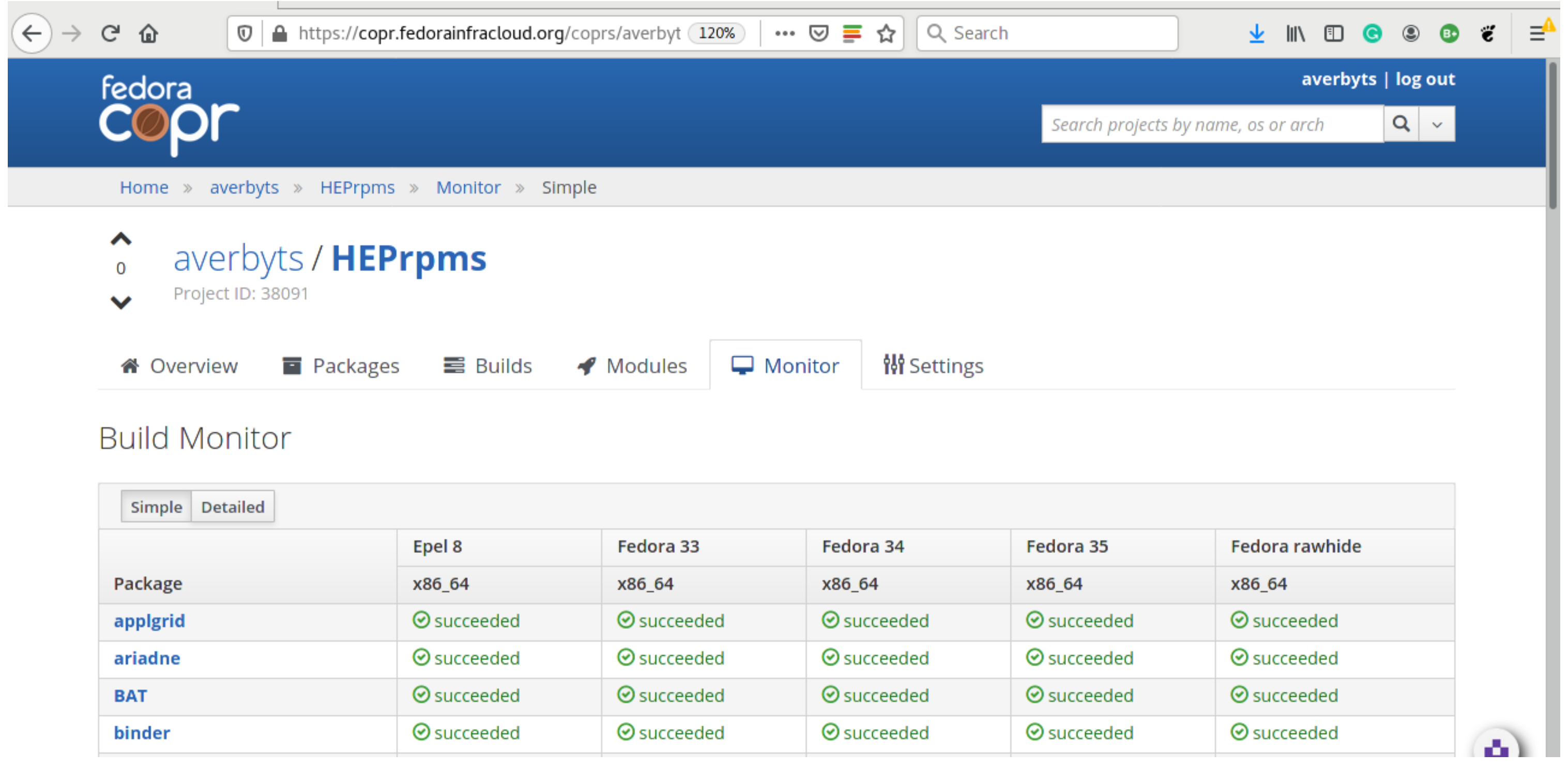
As of 2021, the Red Hat, SUSE and Debian families of Linux are dominant in HEP computing. The first two categories share the same format for the software deployment – the RPM[1]. RPM is a free and open-source package management system existing for over two decades and has

- Excellent support by major Linux distributives and industry.
- Excellent packaging guides from RPM and major distros.
- Support for binary, source and debug packages.
- Standardised and simple packaging prescription.
- Build/runtime resolution of dependencies.
- Cryptography signatures.
- Multiple repositories and compatibility for **decades**.
- Plug-n-play cloud build services.
- Familiar to many users.

We present a repository of HEP packages build for Red Hat and SUSE families of Linux **https://copr.fedorainfracloud.org/coprs/averbyts/HEPrpms**

Packages

applgrid ariadne BAT binder blackhat cascade cernlib CGAL chaplin clhep collier cuba DD4hep Delphes EvtGen fastjet fastnlo FeynHiggs fjcontrib form geant4 ginac golem95 gosam gosam-contrib HepPDT Herwig HJets hoppet hztool iminuit JetVHeto LCIO LoopTools MC-TESTER MCFM MG5aMC njet nlojet++ noweb openloops PHOTOS Professor PTL pythia6 python-awkward python-uproot4 qcdloop qcdnum rapgap recola recola2 Rivet SHERPA-MC TAUOLA TheP8I ThePEG tmdlib topdrawer ugs VBFNLO whizard YODA
≈ 60 **packages, latest versions, build with all dependencies**.



	Epel 8	Fedora 33	Fedora 34	Fedora 35	Fedora rawhide
Package	x86_64	x86_64	x86_64	x86_64	x86_64
applgrid	✓ succeeded	✓ succeeded	✓ succeeded	✓ succeeded	✓ succeeded
ariadne	✓ succeeded	✓ succeeded	✓ succeeded	✓ succeeded	✓ succeeded
BAT	✓ succeeded	✓ succeeded	✓ succeeded	✓ succeeded	✓ succeeded
binder	✓ succeeded	✓ succeeded	✓ succeeded	✓ succeeded	✓ succeeded

References and acknowledgements

[1] RPM Developers, RPM Package Manager, `https://rpm.org/`
[2] Red Hat Inc. et al., COPR,`https://pagure.io/copr/copr`
[3] openSUSE Project, Community Build Service,`https://cbs.centos.org`
[4] CentOS Community, Open Build Service,`https://build.openSUSE.org/`
[5] HEPRrpms, spec repository, `https://github.com/andriish/HEPrpms`
[6] HEPRrpms, packages, `https://copr.fedorainfracloud.org/coprs/averbyts/HEPrpms`, `https://copr.fedorainfracloud.org/copts/averbyts/HEPrpmsSUSE`
[7] Red Hat Inc. et al., Extra Packages for Enterprise Linux, `https://docs.fedoraproject.org/en-US/epel/`
[8] openSUSE contributors et al., openSUSE:Science Repositories, `https://en.openSUSE.org/openSUSE:Science_Repositories`
[9] NorduGRID, Advanced Resource Connector, `http://www.nordugrid.org/`

- Thanks to Matthias Ellert for a great example of HEP software packaging.

Packaging with RPMs

The packaging of some software, e.g. 'TheP8I-2.0.1' consist of

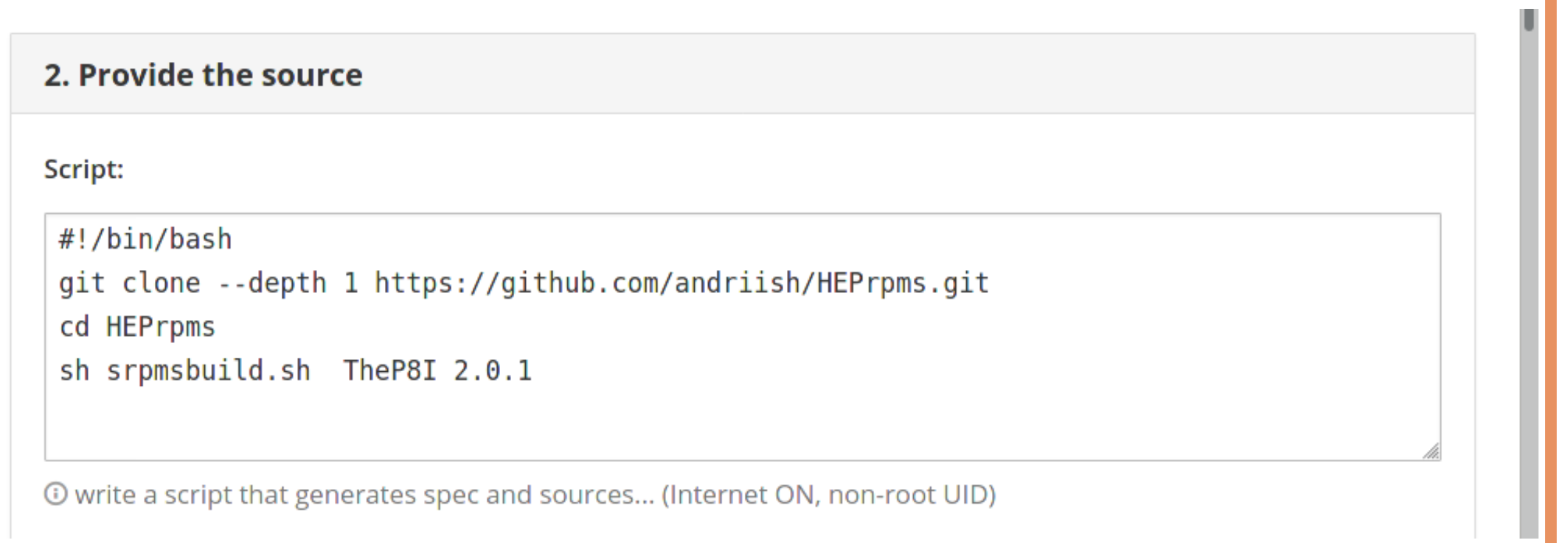
- Writing a 'TheP8I.spec' (see right) build receipt which contains
  - List of sources (URLs).
  - List of runtime/build time dependencies.
  - Prescription for builds/installation.
- Rebuilding the packages locally, or
- Use '.spec' files in the cloud build system COPR [2] (or CBS [3], or OBS [4]) (see right). The build services typically create a 'yum/dnf' repository.

```
Name:           TheP8I
Version:        2.0.1
Release:        3%{?dist}
License:        GPL
Url:            https://gitlab.cern.ch/TheP8I/TheP8I
Source0:        https://gitlab.cern.ch/TheP8I/TheP8I/-/archive/2.0.1/%{name}-2.0.1.tar.gz
Prefix:         %{_prefix}
Summary:        Lund hadronisation for Herwig
%if 0%{?rhel} || 0%{?fedora}
BuildRequires:  pythia8-devel pythia8 gsl lhapdf lhapdf-devel
Requires:       pythia8 gsl ThePEG
%endif
%if 0%{?suse_version}
BuildRequires:  pythia-devel libpythia8 gsl libLHAPDF LHAPDF-devel
Requires:       libpythia8 gsl ThePEG
%endif
BuildRequires: gcc-c++ gcc ThePEG-devel ThePEG autoconf automake libtool gsl-devel
%description
Lund hadronisation for Herwig. Part of earlier ThePEG codes.

%prep
%setup -qn TheP8I-2.0.1

%build
%configure --disable-rpath
make %{?_smp_mflags}
%install
%make_install

%files
%{_libdir}/ThePEG/*
%_datadir/*

%changelog
* Sun Aug 01 2021 Andrii Verbytskyi<andrii.verbytskyi@mpp.mpg.de> 2.0.1
  - RPATH
* Thu Mar 12 2020 Andrii Verbytskyi <andrii.verbytskyi@mpp.mpg.de> 2.0.0
+ Update to 2.0.0
```

The most time consuming part is to create the '.spec' and prepare patches. For the maintenance and rebuilds it is sufficient to have just a web-browser.

Maintainer perspective:

- The number of packages that can be easily maintained ≈ 100/person, is limited by the needed domain-specific knowledge for those packages [9]. The rebuilds require < 1h/month with some extra hours used for bugreports to upstream.


2. Provide the source

Script:

```
#!/bin/bash
git clone --depth 1 https://github.com/andriish/HEPrpms.git
cd HEPrpms
sh srpmsbuild.sh  TheP8I 2.0.1
```

ⓘ write a script that generates spec and sources... (Internet ON, non-root UID)

- Almost all packages are scientific packages, **no need to maintain general purpose packages**. Problems with general-purpose package are bugreported.
- The build tools such as RPM, yum, dnf, zypper & documentation are maintained by the upstream.
- Many dependency packages are already available in EPEL [7] and openSUSE [8] science reposotories.

Deployment and usage with RPMs

Administrator/user perspective: no extra scripts, manuals, no maintenance time.

- Add the repository URL from cloud to the list of your system repositories, e.g.

```
#Fedora
[root@host ~]# dnf copr enable averbyts/HEPrpms
#CentOS8
[root@host ~]# yum copr enable averbyts/HEPrpms
#openSUSE Tumbleweed
[root@host ~]# wget https://copr.fedorainfracloud.org/coprs/averbyts/HEPrpmsSUSE/repo/opensuse
-tumbleweed/averbyts-HEPrpmsSUSE-opensuse-tumbleweed.repo
[root@host ~]# sed -i 's/\$basearch/x86_64/g' averbyts-HEPrpmsSUSE-opensuse-tumbleweed.repo
[root@host ~]# zypper addrepo averbyts-HEPrpmsSUSE-opensuse-tumbleweed.repo
```

- Use standard commands to install the software. Updates are automatic.

```
#Fedora
[root@host ~]# dnf install Herwig
#CentOS8
[root@host ~]# yum install Herwig
#openSUSE Tumbleweed
[root@host ~]# zypper install Herwig
```

Actual usage includes, but not limited to machines in MPP with openSUSE and couple with CentOS and CI of some HEP projects in github.cern.ch/gitlab.cern.ch/github.com.

User perspective: no extra scripts, manuals, no maintenance time, no ugly startup scripts.

- **Just run your software**.

```
[user@host ~]# Herwig read mycard.dat  #Fedora, CentOS8, openSUSE
```

- Or do a full installation in container. **Just some minutes and 10LOC of config are needed**.

Conclusions

The usage of standard approach to packaging and deployment of HEP software has significant benefits in comparison to other approaches.

- Requires **zero preparations** for users and administrators of target systems.
- Requires **zero additional maintenance efforts** for packaging tools or documentation.
- Provides **reproducible** source, binary and debug **packages**.
- **Saves** the CPU and working **time** preventing endless rebuilds.
- Let the maintenance effort to be **concentrated on the scientific packages** only.
- Provides tests for the scientific packages in the most modern environments.
- Provides tests for the general-purpose packages in their native upstream environment.

The used approach can be easily scaled to accommodate some hundreds of packages, as that was proven in the past in the ultimately successful ARC Middleware project [9].