

# Gentoo Prefix as a Physics Software Manager

Benda Xu<sup>1,2,3,5,\*</sup>, Guilherme Amadio<sup>4,5</sup>, Fabian Groffen<sup>5</sup>, and Michael Haubenwallner<sup>5</sup>

<sup>1</sup>Department of Engineering Physics, Tsinghua University, Beijing 100084, China

<sup>2</sup>Key Laboratory of Particle & Radiation Imaging (Tsinghua University), Ministry of Education, Beijing 100084, China

<sup>3</sup>Center for High Energy Physics, Tsinghua University, Beijing 100084, China

<sup>4</sup>CERN, 1211 Geneva 23, Switzerland

<sup>5</sup>Gentoo Linux

**Abstract.** Gentoo Prefix is explored to manage sophisticated physics software stacks. It will be shown that Gentoo Prefix is an advantageous package management solution for big physics experiments, for its reusability on heterogeneous host environments, its vast collection of *ebuild* recipes, its extensibility for the future computing architectures and its deep root in an open diverse community inside and outside science.

## 1 Introduction

In big physics experiments, as simulation, reconstruction and analysis become more sophisticated, scientific reproducibility is not a trivial task[1–3]. Open data alone are not enough to make results replicable. With the advancement of data analysis and detector simulations, the physics software stack is getting deeper and becoming one of the biggest challenges.

Modularity is a common sense practice of software engineering to facilitate quality and reusability of code. However, that often introduces nested dependencies not obvious for physicists to work with. Package managers were invented by GNU/Linux distributions [4] and regarded as the single biggest advancement Linux has brought to the industry. It is the widely practised solution to organize dependencies systematically. Many scientific and high performance computing (HPC) oriented package managers exist with the goal to handle software complexity, such as EasyBuild [5], Spack [6, 7], Conda [8], Nix [9, 10] and GNU Guix [11], etc.

*Gentoo Linux* is a full GNU/Linux distribution since 1999. It is general purpose and used for daily and scientific tasks, and runs on the majority of instruction set architectures (ISA), including x86/amd64, sparc, arm/64, alpha, mips32/64, riscv, etc. It is regarded as a meta-distribution for its ultimate flexibility to be repurposed for specific needs. Development of Gentoo packages is open and distributed, and represents shared pieces of wisdom of the community.

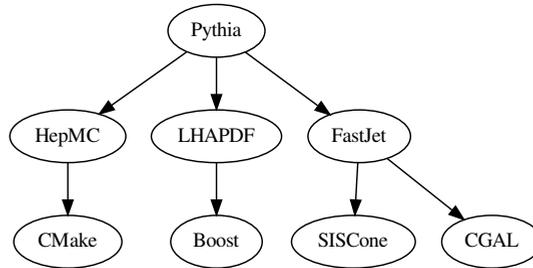
*Portage*, the package manager of Gentoo Linux, is both robust and flexible, and is highly regarded by the free operating system community. In the form of Gentoo Prefix [12–14], portage can be deployed by a normal user into a directory prefix, on a workstation, a cloud instance or a supercomputing node. Software is described by its build recipes *ebuilds* along with dependency relations.

---

\*e-mail: [orv@tsinghua.edu.cn](mailto:orv@tsinghua.edu.cn)

## 2 Dependency Handling

The basic function of a package manager is to automatically satisfy dependency graphs by recursively installing needed software. Figure 1 shows the example of the *Pythia* [15] to be installed by portage.



```
# emerge --tree pythia
[ebuild N ] sci-physics/pythia-8.2.26:8::gentoo
[ebuild N ] sci-physics/hepmc-2.06.09-r1::gentoo
[ebuild N ] sci-physics/lhapdf-6.2.3::gentoo
[ebuild N ] dev-libs/boost-1.71.0:0/1.71.0::gentoo
[ebuild N ] dev-util/boost-build-1.71.0::gentoo
[ebuild N ] sci-physics/fastjet-3.0.6-r1::gentoo
[ebuild N ] sci-physics/siscone-3.0.3::gentoo
[ebuild N ] sci-mathematics/cgal-4.11.3:0/13::gentoo
```

**Figure 1.** Example dependency graph of *Pythia* physics event generator and demonstration of installation by `emerge`, the Gentoo portage user interface command. Recursive dependencies are automatically drawn in.

## 3 Gentoo Prefix as an Universal Environment

*Gentoo Prefix* installs a complete Gentoo userspace into a directory, represented by the shell variable `EPREFIX`. Prefix extends the host operating system with almost the full collections of software from Gentoo. A bootstrap helper script is provided to compile out Prefix from scratch on many of the existing POSIX environments, including macOS, Solaris, Windows/-Cygwin and most variants of GNU/Linux. Restricting our discussion to GNU/Linux, the script installs Prefix in 3 stages: compile python and portage in `EPREFIX/tmp`; install gcc in `EPREFIX/tmp` by portage, referred as Stage 2; build Gentoo in `EPREFIX` by portage and Stage 2 gcc. The flow is schematically shown in Figure 2.

From Figure 2, it is evident that Prefix depends only on the kernel on the host, providing a full GNU userspace. Dynamic linking is performed within the Prefix. Neither `LD_LIBRARY_PATH` nor `ELF DT_RPATH` is needed. A Gentoo Prefix is independent of the host userspace, but unlike containers[16], it shares the file system view of the host and does not rely on any special privilege or configuration assumptions.

This flexibility makes Gentoo Prefix redistributable to another host provided the kernel APIs are compatible, which is straightforwardly achieved if Prefix is tuned to the lowest



Thanks to the adaptable design of `portage`, it is interoperable with other software package managers. One good example is the GNU R [19] ecosystem offered by the *R Overlay*, with *overlay* being a Gentoo term for a package repository other than the main one [20]. 18993 packages are automatically generated from R packages as of November 2019. Other ebuild generators, such as PyPI, Emacs ELPA, Octave Forge, Java Maven, Rust Cargo exist, making Gentoo extensible by the available packaging systems.

## 5 Case Studies

This section discusses some typical use cases of Gentoo Prefix.

### 5.1 Salvage of a System More than 10 Years Old

As of 2020, there still may be some HPC systems running end-of-life products like RHEL 5 initially released in 2007, for various reasons, most notably needing to support legacy scientific software and lack of funding for upgrade. Disregarding security consequences, the biggest drawback is that the old tools usually lack useful features.

Gentoo Prefix is independent of the host userspace. It selects the newest `glibc` possible to match the old running Linux kernel. The rest of Prefix is smoothly established with few patches. Taking RHEL 5 for example in November 2019, without intervening the host, a normal user can upgrade Python from 2.4 to 3.6, GCC from 4.1 to 8.2, `ld` from 2.17 to 2.30, `glibc` from 2.5 to 2.19. That makes the outdated system as usable as its cutting-edge counterparts.

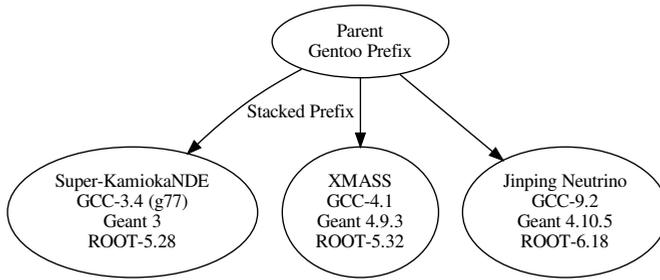
### 5.2 Coexistence of Multiple Versions by Stacked Prefix

Conventional GNU/Linux distributions including Gentoo values deduplication of code in a single system. They hold well tested versions of software in stable branches or newest versions in development branches, but not both. This by default does not align well with physics computing work loads, because scientific tasks value reproducibility the most, which sometimes means a specific version of software to be used throughout the entire lifespan of an experiment for decades. Gentoo `portage` offers *SLOTS* to allow coexistence of multiple versions of a single piece of software[20], but those are only enabled in carefully selected packages.

A more flexible solution exists for Gentoo Prefix. Stacked Prefix makes thin instances based on a common parent. Build-time dependencies are reused to avoid code duplication. In the child Prefix, any version of needed software can be installed, masking those in the parent Prefix. With overlays, out-of-tree ebuilds can be added and maintained specifically for a certain experiment. Figure 4 shows coexistence of software stacks required by 3 underground experiments, Super-Kamiokande [21], XMASS [22] and Jinping Neutrino Experiment [23].

### 5.3 Portability to Future ISA

In the field of HPC, new ISAs are being developed for better power efficiency and scalability, for example the Sunway TaihuLight [24] with extended alpha ISA, the Post-K [25] with arm64 ISA. The portability of Gentoo allows a smooth migration path to a different ISA, without much extra efforts. Since 2013, Gentoo Prefix runs on Android-based mobile devices in the form of *Gentoo on Android*, giving native GNU userspace environment to handheld users. The same technology applies to arm64-based supercomputing facilities. It is remarkable that the most portable and the most powerful devices are being powered by the same ISA, while enhanced by Gentoo Prefix.



**Figure 4.** Legacy software stacks of experimental physics are managed by stacked Prefix, to give maximal flexibility while avoiding code duplication using the parent Prefix.

## 5.4 Institutional Facilities

Large institutions host many physics experiments. The computer infrastructures are often shared by people from different groups. In this scenario, Gentoo Prefix offers complementary functions to the existing software managers. Authors of this article have deployed Gentoo Prefix at CERN (European Organization for Nuclear Research) at `/cvmfs/sft.cern.ch/lcg/contrib/gentoo/startprefix` and IHEP (Institute of High Energy Physics) at `/cvmfs/juno.ihep.ac.cn/ci/gentoo/startprefix`.

## 5.5 Ebuild Example: ROOT

Gentoo ebuilds are written in carefully organized bash script, dictated by the package manager specification [20]. A code snippet of an ebuild for CERN ROOT [26, 27] is reproduced below.

```
# Copyright 1999–2019 Gentoo Authors
# Distributed under the terms of the GNU General Public License v2

EAPI=6

# ninja does not work due to fortran
CMAKE_MAKEFILE_GENERATOR=emake
FORTRAN_NEEDED="fortran"
PYTHON_COMPAT=( python2_7 python3_{5,6} )

inherit cmake-utils eapi7-ver elisp-common eutils fortran-2 \
        prefix python-single-r1 toolchain-funcs

DESCRIPTION="C++ data analysis framework and interpreter from CERN"
HOMEPAGE="https://root.cern"
SRC_URI="https://root.cern/download/${PN}_v${PV}.source.tar.gz"
```

**Figure 5.** Example ebuild for CERN ROOT. The script starts with a declarative header to specify properties of the piece of software.

## 6 Conclusion

Ever since its birth in 1999, Gentoo has been a paradise for geek users and developers. Its package manager portage is a comprehensive solution meeting all the needs of software management in big physics experiments, and performs especially well with CernVM-FS [17]. The physics use case of Gentoo is a natural consequence of its flexibility and expertise in building operating systems. It is delightful to discover that systems developed by the community are so useful for science. By examining real world use cases of Gentoo Prefix, it has been shown that physicists could benefit from existing tools of proven superiority to guarantee reproducibility in simulation, reconstruction and analysis of big physics experiments.

## References

- [1] M.R. Munafò, B.A. Nosek, D.V.M. Bishop, K.S. Button, C.D. Chambers, N.P.d. Sert, U. Simonsohn, E.J. Wagenmakers, J.J. Ware, J.P.A. Ioannidis, *Nat Hum Behav* **1**, 1 (2017)
- [2] M. Baker, *Nature News*, **533**, 452 (2016)
- [3] J. Conrad, *Nature News*, **523**, 27 (2015)
- [4] D. Spinellis, *IEEE Software* **29**, 84 (2012)
- [5] D. Alvarez, A. O’Cais, M. Geimer, K. Hoste, in *2016 Third International Workshop on HPC User Support Tools (HUST)*, pp. 31–40 (2016)
- [6] T. Gamblin, M. LeGendre, M.R. Collette, G.L. Lee, A. Moody, B.R. de Supinski, S. Futral, in *SC ’15: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, pp. 1–12 (2015)
- [7] G. Becker, P. Scheibel, M. LeGendre, T. Gamblin, in *2016 Third International Workshop on HPC User Support Tools (HUST)*, pp. 14–23 (2016)
- [8] B. Grüning, R. Dale, A. Sjödin, B.A. Chapman, J. Rowe, C.H. Tomkins-Tinch, R. Valieris, J. Köster, *Nat Methods* **15**, 475, number: 7, (2018)
- [9] A. Devresse, F. Delalondre, F. Schürmann, in *Proceedings of the 3rd International Workshop on Software Engineering for High Performance Computing in Computational Science and Engineering*, SE-HPCCSE ’15, pp. 25–31 (2015)
- [10] B. Bzeznik, O. Henriot, V. Reis, O. Richard, L. Tavard, in *Proceedings of the Fourth International Workshop on HPC User Support Tools*, HUST’17, pp. 1–6 (2017)
- [11] L. Courtès, R. Wurmus, in *Euro-Par 2015: Parallel Processing Workshops*, pp. 579–591 (2015)
- [12] B. D. Xu, G. Amadio, F. Groffen, M. Haubenwallner, <https://wiki.gentoo.org/wiki/Project:Prefix> (2020)
- [13] G. Amadio, B. D. Xu, arXiv:1610.02742 [cs] (2016)
- [14] H.I. Ioanas, B. Saab, M. Rudin, *Research Ideas and Outcomes* **3**, e12095 (2017)
- [15] T. Sjöstrand, S. Ask, J.R. Christiansen, R. Corke, N. Desai, P. Ilten, S. Mrenna, S. Prestel, C.O. Rasmussen, P.Z. Skands, *Computer Physics Communications* **191**, 159 (2015)
- [16] G.M. Kurtzer, V. Sochat, M.W. Bauer, *PLOS ONE* **12**, e0177459 (2017)
- [17] J. Blomer, P. Buncic, I. Charalampidis, A. Harutyunyan, D. Larsen, R. Meusel, *J. Phys.: Conf. Ser.* **396**, 052013 (2012)
- [18] LSB Workgroup, The Linux Foundation, [https://refspecs.linuxfoundation.org/FHS\\_3.0/fhs/index.html](https://refspecs.linuxfoundation.org/FHS_3.0/fhs/index.html) (2015)
- [19] W.N. Venables, D.M. Smith et al., *An introduction to R* (2020)
- [20] S.P. Bennett, C. McCreesh, F. Christian, U. Müller, *Package Manager Specification* (2018)

- [21] S. Fukuda, Y. Fukuda, T. Hayakawa, E. Ichihara, M. Ishitsuka, Y. Itow, T. Kajita, J. Kameda, K. Kaneyuki, S. Kasuga et al. (Super-Kamiokande Collaboration), Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment **501**, 418 (2003)
- [22] K. Abe, K. Hieda, K. Hiraide, S. Hirano, Y. Kishimoto, K. Kobayashi, S. Moriyama, K. Nakagawa, M. Nakahata, H. Nishiie et al. (XMASS Collaboration), Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment **716**, 78, 00013 (2013)
- [23] J.F. Beacom, S. Chen, J. Cheng, S.N. Doustimotlagh, Y. Gao, G. Gong, Hui Gong, L. Guo, R. Han, H.J. He et al., Chinese Phys. C **41**, 023002 , 00003 (2017)
- [24] H. Fu, J. Liao, J. Yang, L. Wang, Z. Song, X. Huang, C. Yang, W. Xue, F. Liu, F. Qiao et al., Sci. China Inf. Sci. **59**, 072001 (2016)
- [25] B. Sorensen, IEEE Annals of the History of Computing **21**, 48 (2019)
- [26] I. Antcheva, M. Ballintijn, B. Bellenot, M. Biskup, R. Brun, N. Buncic, P. Canal, D. Casadei, O. Couet, V. Fine et al., Computer Physics Communications **180**, 2499 (2009)
- [27] R. Brun, F. Rademakers, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment **389**, 81 (1997)