

Spack is an open-source package manager for building and customizing HPC software stacks across laptops, supercomputers, and clouds. Maintained by a large international community, Spack contributes to HPC software sustainability efforts through initiatives like PESO and the High Performance Software Foundation. Spack is the foundation for DOE's E4S software stack.

Spack's 1.2 release adds an interactive parallel installer, concretization groups, cached concretization, and SBOM generation.

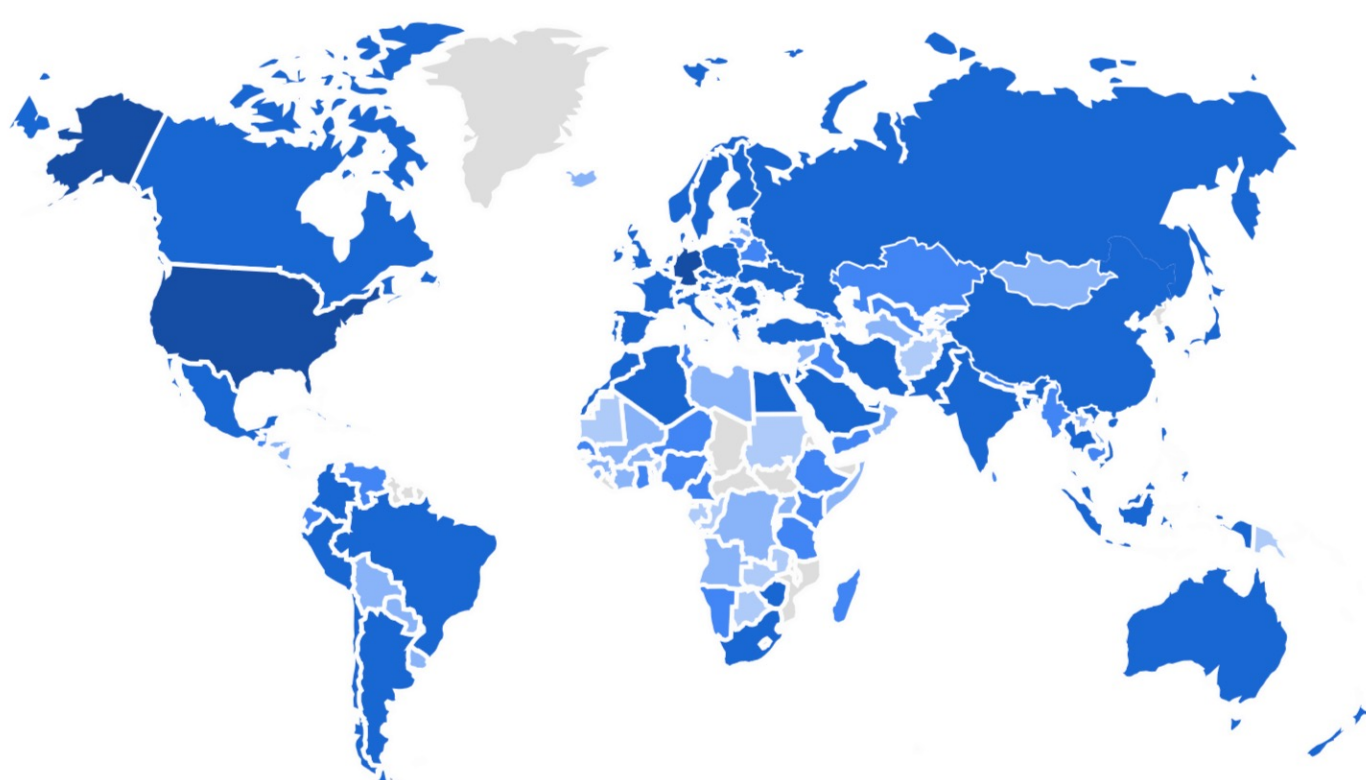
What is Spack?

- Spack automates builds of scientific software and its dependencies
- Packages are *parameterized*; users can easily tune configuration

Simple syntax enables complex installs

```
$ spack install hdf5@1.10.5
$ spack install hdf5@1.10.5 %clang@6.0
$ spack install hdf5@1.10.5 +threadssafe
$ spack install hdf5@1.10.5 cppflags="-O3 -g3"
$ spack install hdf5@1.10.5 target=haswell
$ spack install hdf5@1.10.5 +mpi ^mpich@3.2
```

Spack is supported by a large international community and has over 1,600 contributors from academia, industry, and government labs



Visitors to Spack's documentation site

What's happened in the past two years?

Four feature releases: v0.23.0, v1.0.0, v1.1.0, v1.2.0

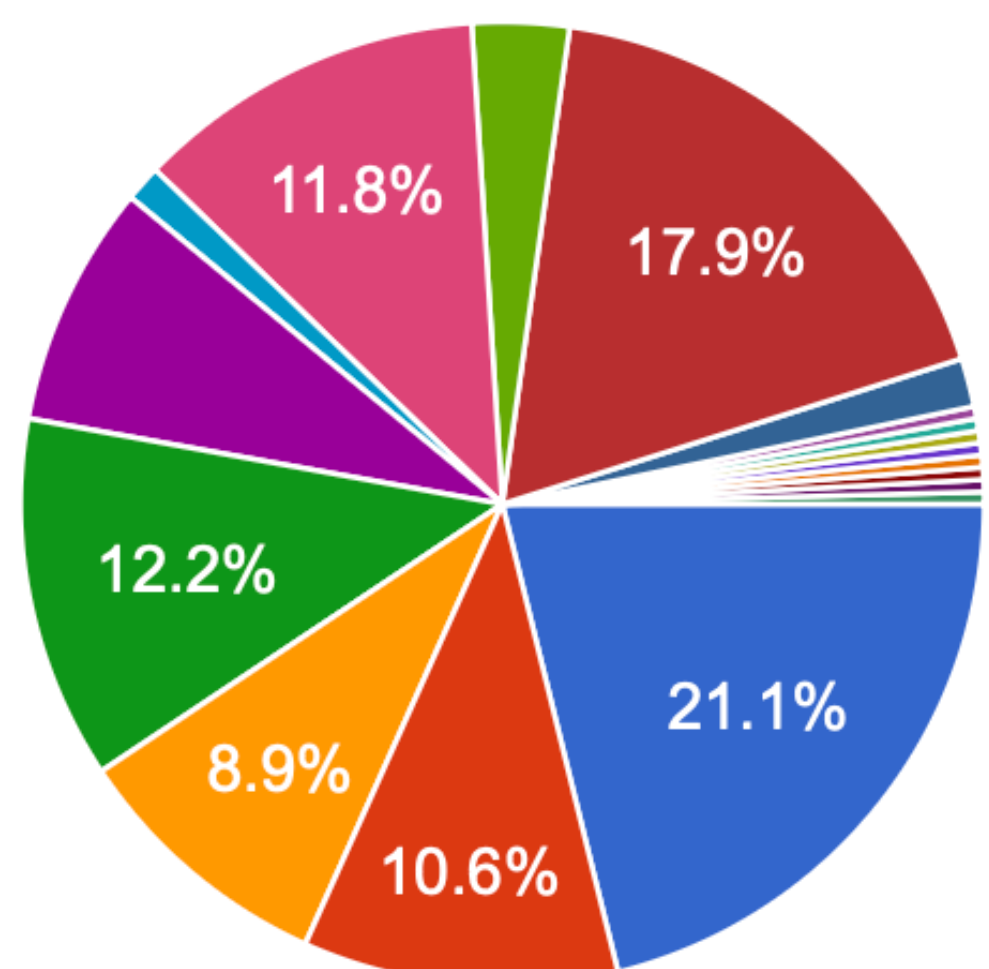
Growth:

- Contributors: from 1,400 to over 1,600 (15-20 per month)
- Packages: from 8,200 to 8,800
- Joined Linux Foundation, set up neutral governance

New user meetings foster collaboration: HPSFCon 2025-2026

The user base remains quite diverse:

- Research Software Engineer (RSE)
- Software Developer
- DevOps/SRE
- System Administrator
- User Support Staff
- Data Scientist
- Computational Scientist
- HPC User / Analyst



Spack's new installer

spack install exploits parallelism by scheduling independent package builds concurrently

Builds share a jobserver to dynamically manage work

The new TUI shows all concurrently running builds

Supports intra- and inter-node build parallelism with little overhead

```
Spack: 7/10 - 14 jobs

Progress: 7/10 +/-: 14 jobs /: filter v: logs (n)ext/(p)rev
[+] kpx337 ca-certificates-mozilla@2025-08-12
[+] thy5vqo glibc@2.39
[+] fhe8mve gcc@13.3.0
[+] mfeo10a legion@develop
[+] laoej29 realm@develop
[+] 78dsk1k gasnet@master
[+] 8r92018 llvm-amdgpu@6.4.3
[-] rome7em rust-bootstrap@nightly staging (2s)
[-] 2ktvbgz kokkos@5.0.0 cmake (30s)
[\] zv7glmj autoconf@2.69 build (10s)
```

Concretization Caching

We now:

- Canonicalize concretizer inputs
- Store result by input hash
- Retrieve by hash later

Useful for sequences of commands

```
> time spack spec hdf5
6.21 real
> time spack spec hdf5
1.79 real
```

Performance improvements

```
> spack spec hdf5
> spack stage hdf5
> spack install hdf5
```

Before: 3 solves
Now: 1 solve

Concretization Groups

Groups are named and can have dependencies

Groups are concretized independently

If a group depends on another:

- It's concretized after dependencies
- Specs from dependencies are *always* reused

```
spack:
  specs:
    - group: compiler
      specs:
        - gcc@15.2
    - group: apps
      needs: [compiler]
      specs:
        - hdf5 %gcc1@5.2
        - libtree %gcc@15.2
```

SBOMs

Standard industry format, allowing us to integrate security tools:

- Identify and uninstall/update compromised software

Include NTIA minimum elements:

- Supplier
- Component name
- Version
- Unique identifiers
- Dependency relationships

Future work:

- More data, more SBOM formats
- SARIF format
- Scanning with Spack CI

```
{
  "spdxVersion": "SPDX-2.3",
  "creators": [
    "Organization: Spack Project",
  ],
  "name": "zlib-ng-2.2.5-3pdcgzlZ5sscd1fwq7",
  "packages": [
    {
      "name": "zlib-ng",
      "versionInfo": "2.2.5",
      "supplier": "Organization: zlib-ng",
      "downloadLocation": "https://github.com/zlib-ng",
    },
    {
      "name": "apple-clang",
      "versionInfo": "17.0.0"
    }
  ]
}
```

Abbreviated SPDX 2.3 SBOM generated by Spack

Other new features

Spack 1.2 also includes experimental build sandboxing and features to isolate independent configuration scopes.

Future directions

- Work with E4S team to automate CI and harden Frank cluster
- Work with HPSF binaries working group to increase range of machines where optimized binary deployment makes sense
- Full-source bootstrapping, increased reproducibility

Connect

spack.io

github.com/spack/spack

