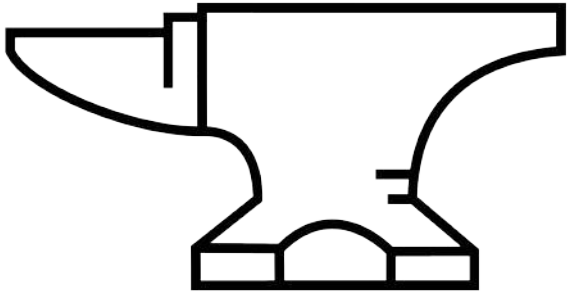


conda-forge

supporting the growth of the volunteer-driven, community-based packaging project

Wolf Vollprecht (QuantStack) · Jannis Leidel (Anaconda) · Jaime Rodríguez-Guerra (Quansight)

EuroSciPy, 2022 · Aug 31st

We are part of the conda-forge core team

Wolf Vollprecht

wolfv · wuoulf

CTO at QuantStack
Initiated the mamba project to make conda-forge faster and the RoboStack project to package ROS for conda

Jannis Leidel

jezdez

Staff Software Engineer on the conda team at Anaconda. Previously co-founded the Python Packaging Authority and serves on the PSF board of directors.

Jaime Rodríguez-Guerra

jaimergp · jaime_rgp

Software eng, PhD in Biotech. Conda enabled my research on molecular simulation and biotechnology.

QuantStack
Scientific Computing

ANACONDA.

QUANSIGHT

Outline

~30 mins

1. A brief overview of conda-forge
2. Organization and maintenance
3. Growth-driven community innovation
4. The future of the conda ecosystem

1. A brief overview of conda-forge

Historical context for conda

B. C.

- Install Python across platforms
- User permissions?
- Compiled extensions?
- Good luck, specially on Windows!

2012

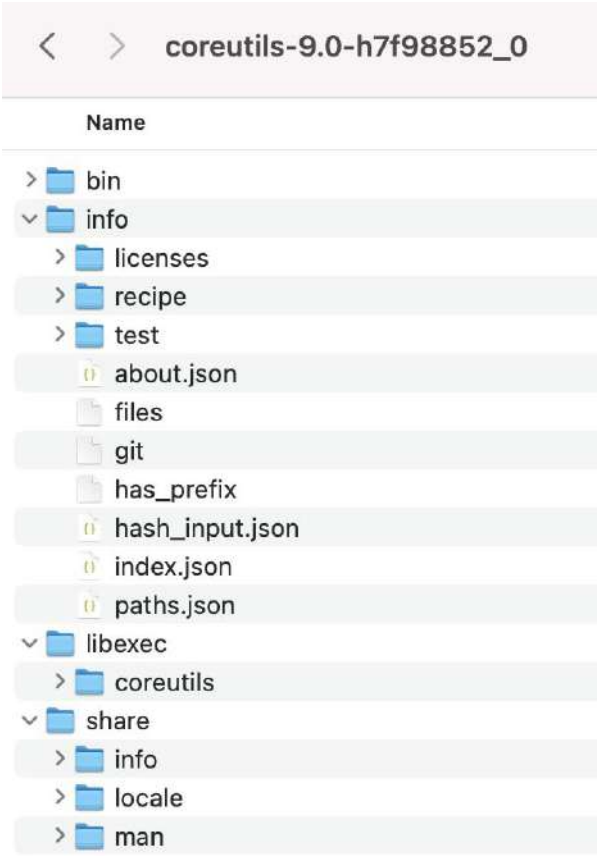
- Continuum Analytics introduces Anaconda
- Solves all those problems!

2013

- conda and conda-build tools

No sudo | Cross-platform | One-click

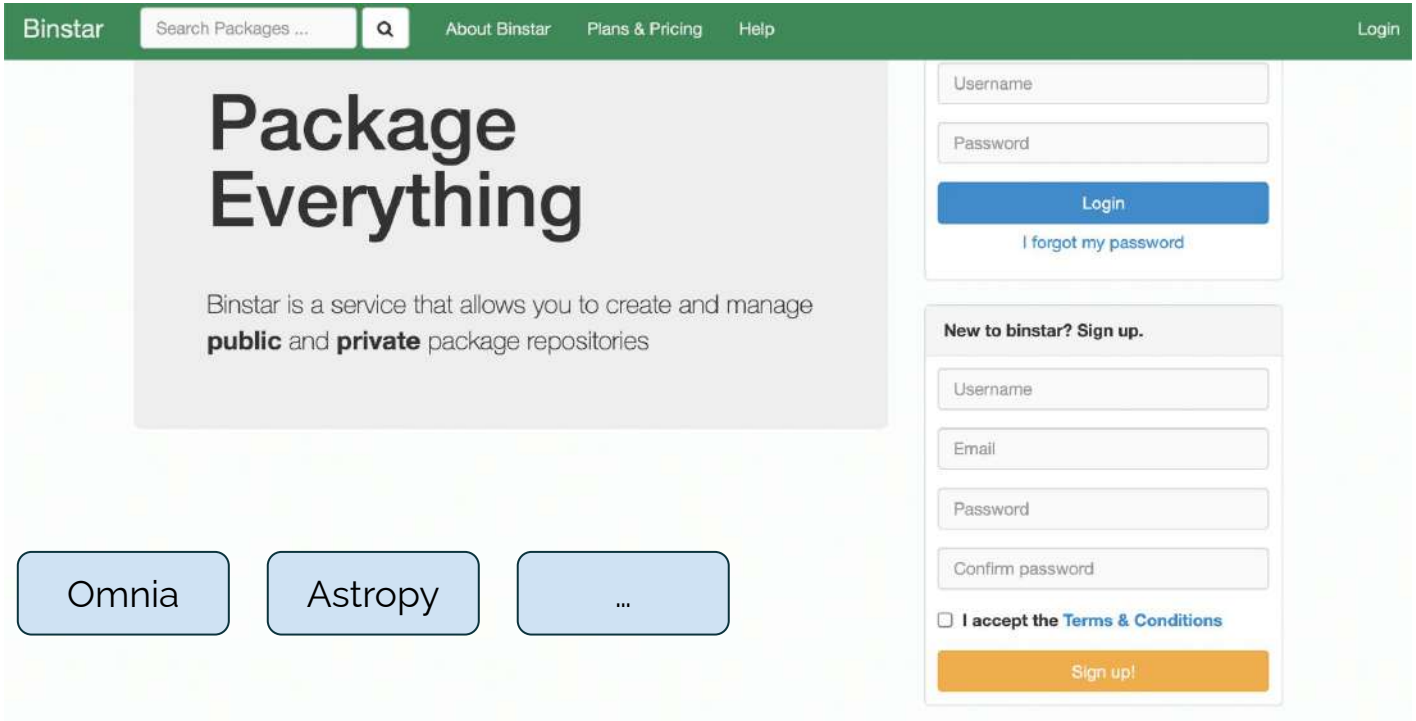
Dependency solver | Pre-compiled | Language agnostic



Binstar (Anaconda.org) empowers communities

- Single vendor bottleneck! Non-mainstream, niche packaging needs?

2014



IOOS

Bioconda

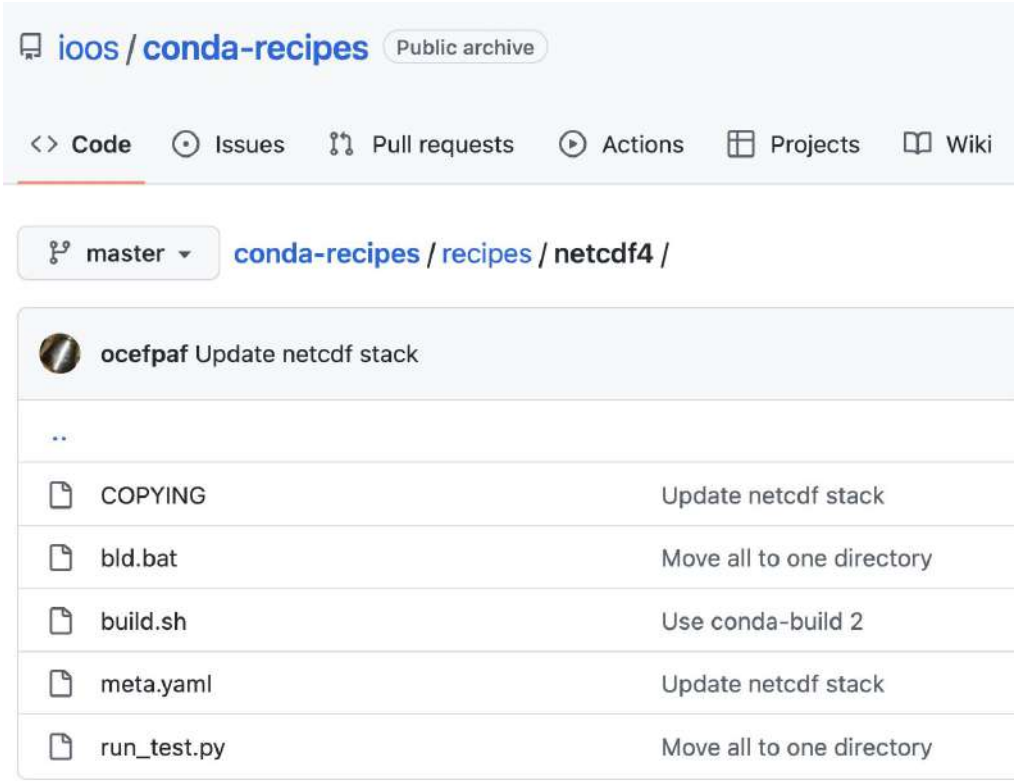
Scitools

Omnia

Astropy

...

Channel maintenance in pre-conda-forge times



Set up like this:

- Mono-repo
- All recipes
- CI-driven
- Conda-build-all

Problems:

- Duplication of effort
- Subtle incompatibilities
- CI limits / scaling issues
- Permission granularity

Emergence of conda-forge

2015

- Key idea: one recipe per repository!
  - CI limits ✅
  - Granular permissions ✅
  - ⚠️ Cross-repository tooling needed!

- conda-forge / conda-smithy designed to create and update “feedstocks”, repositories with this structure:

onda build recipe (meta.yaml + build scripts)
I workflows
lobal configuration for conda build
upporting scripts and metadata

2. Organization and maintenance

Foundational principles

Metadata and scripts
Build logs
Code reviews
Meeting minutes

Transparency

Compatibility

Co-installability
Centralized pinnings

Version updates
Changes in pinnings
New CI services or features

Automation

Distributed ownership

Each package has its own separate and fully operational repository with full permissions

7 years of successful growth!

🎂 2015/04/11 🎂

🎉 **7+** years 🎉

4.2K maintainers,
26 active core,
15 staged-recipes

17.2K
repositories,
19.1K packages,
1M artifacts

6 platforms
Linux (x64, PPC, ARM)
MacOS (x64, ARM)
Windows (x64)

>172K commits
(157K by bots)

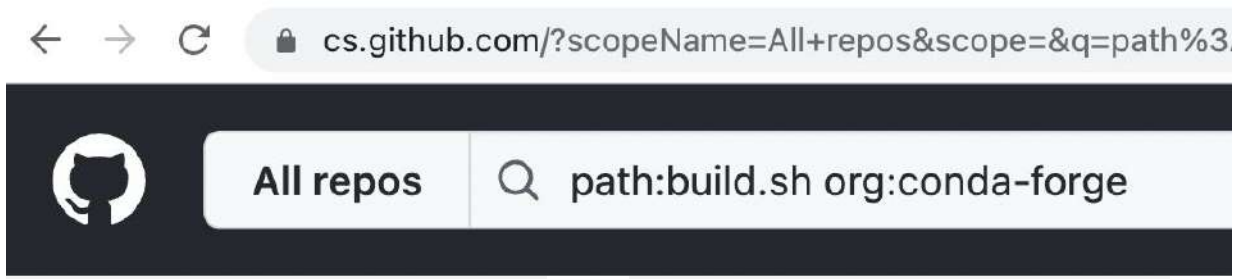
247.3K
issues/PRs
(236K closed)

5 TB
storage*

300M dls, **1.4 PB**
bandwidth
per month*

* Last data for February 2022

Key benefits



Standardized build environments

Preconditioned system state (Docker images, preparation scripts)

Cross-repository conda build configuration

ABI-aware pinnings

Better collaboration model

Per-package issue tracker

Granular interest

Focused help / onboarding

Bigger, unified community

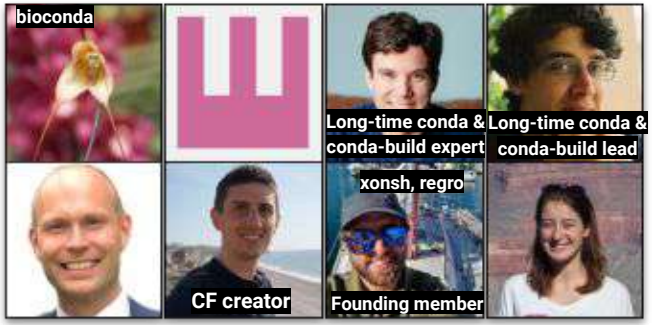
Sum > individual parts

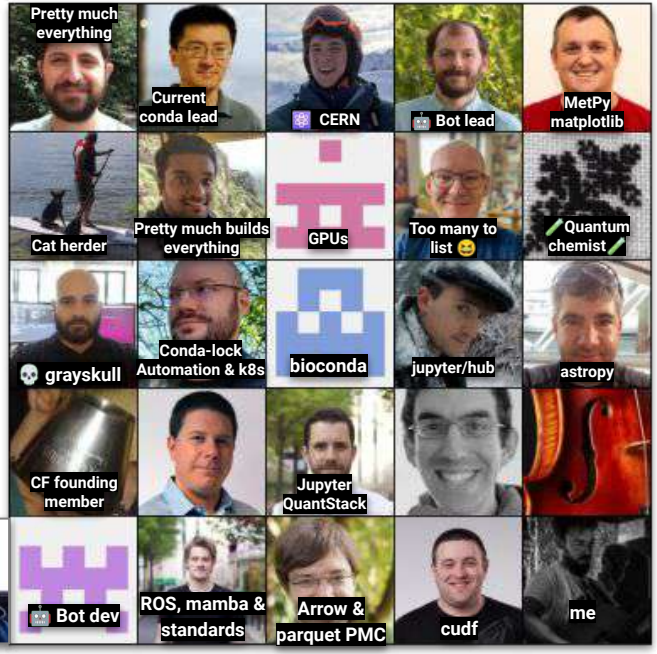
Ever-growing corpus of knowledge

Programmatic insights, build scripts wiki!

Who runs conda-forge

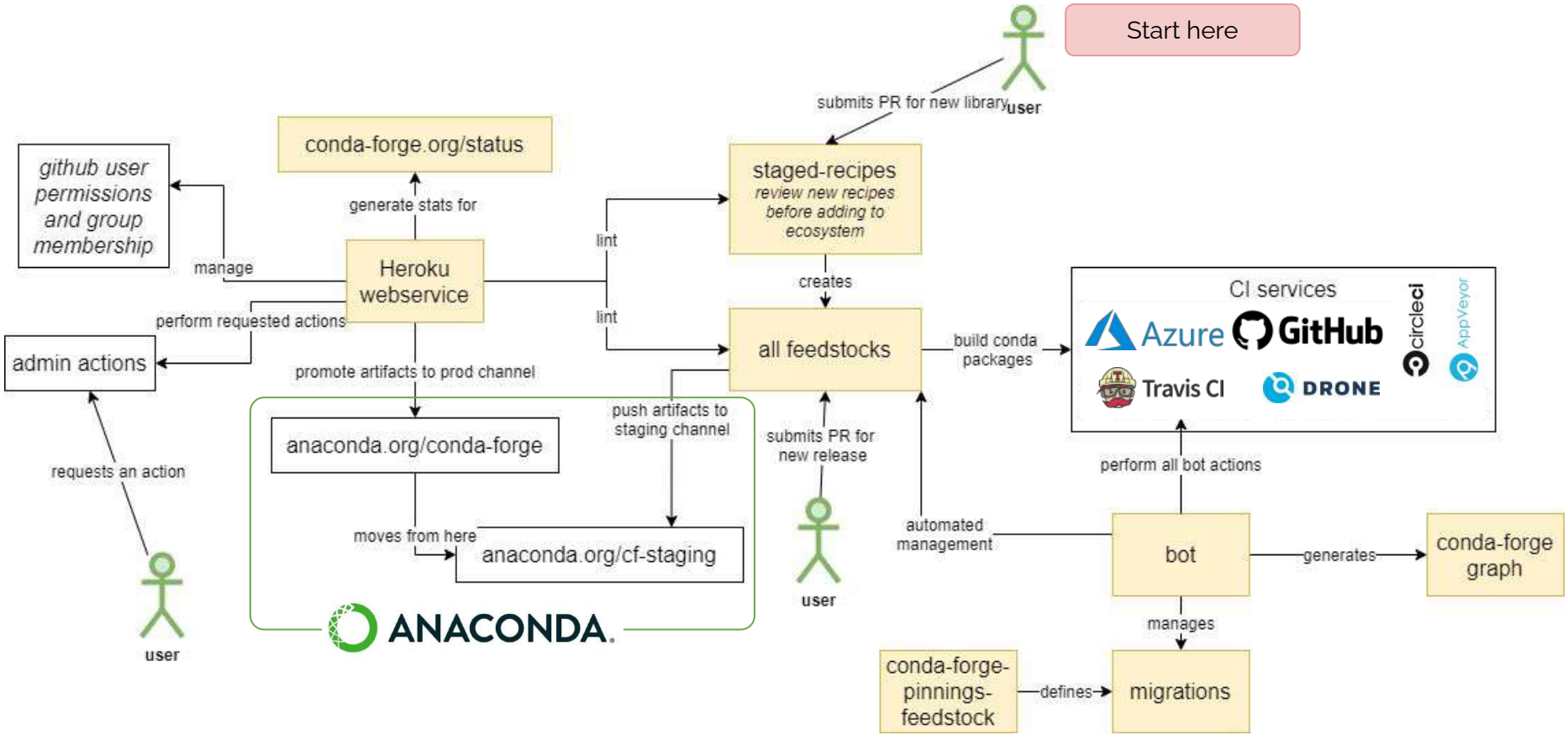
- 100% community driven on GitHub
- Core team: 8 emeritus, 26 active
- 15 Staged-recipes + domain-specific help teams
- 4.2k maintainers, managing 17k repositories
- 🤖🤖🤖 : autotick, linter, user commands, admin requests, migrations, staged to feedstock, artifact validation, repodata collection…



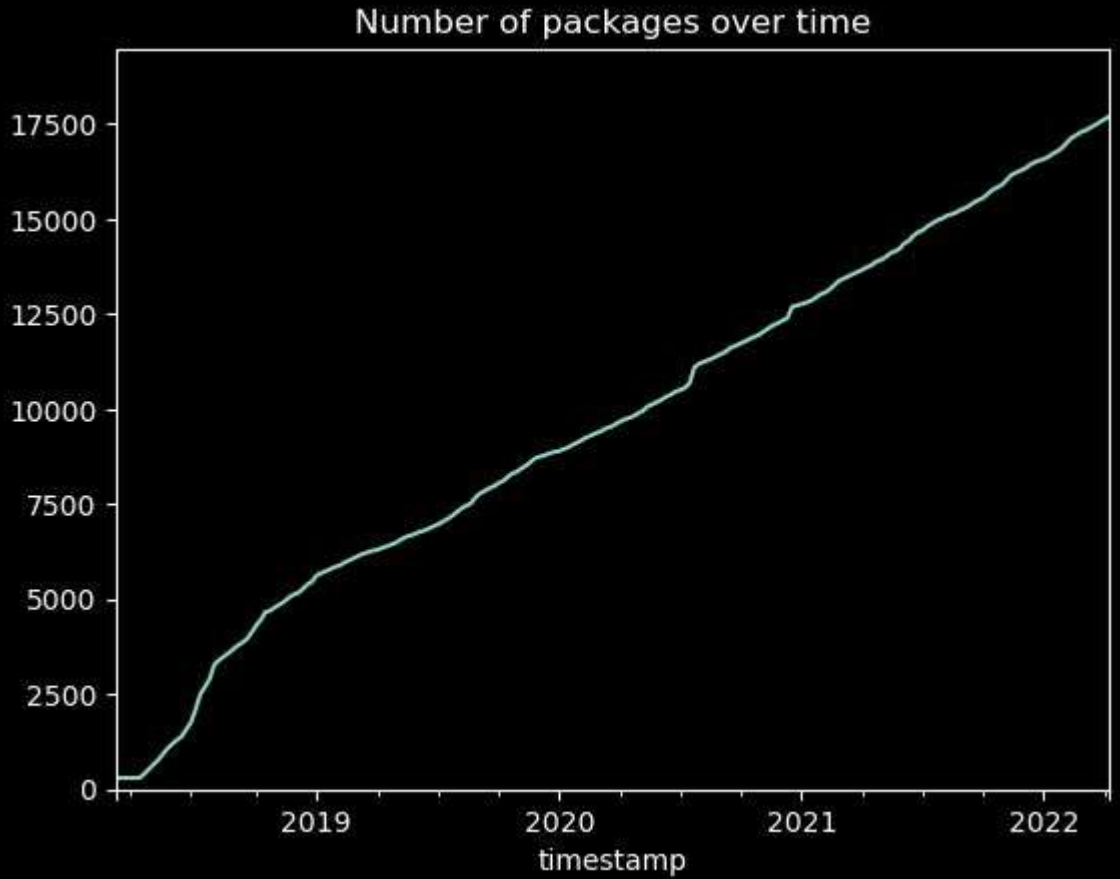


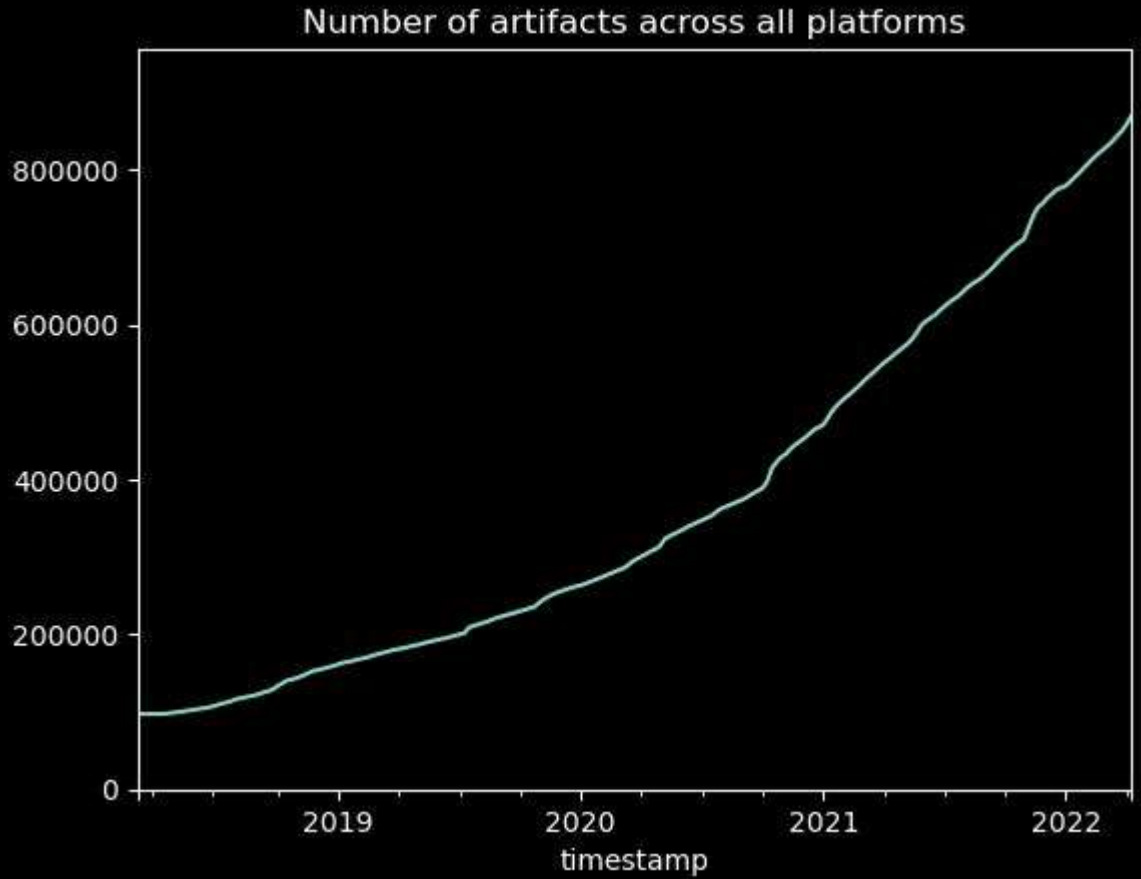
How does it work?

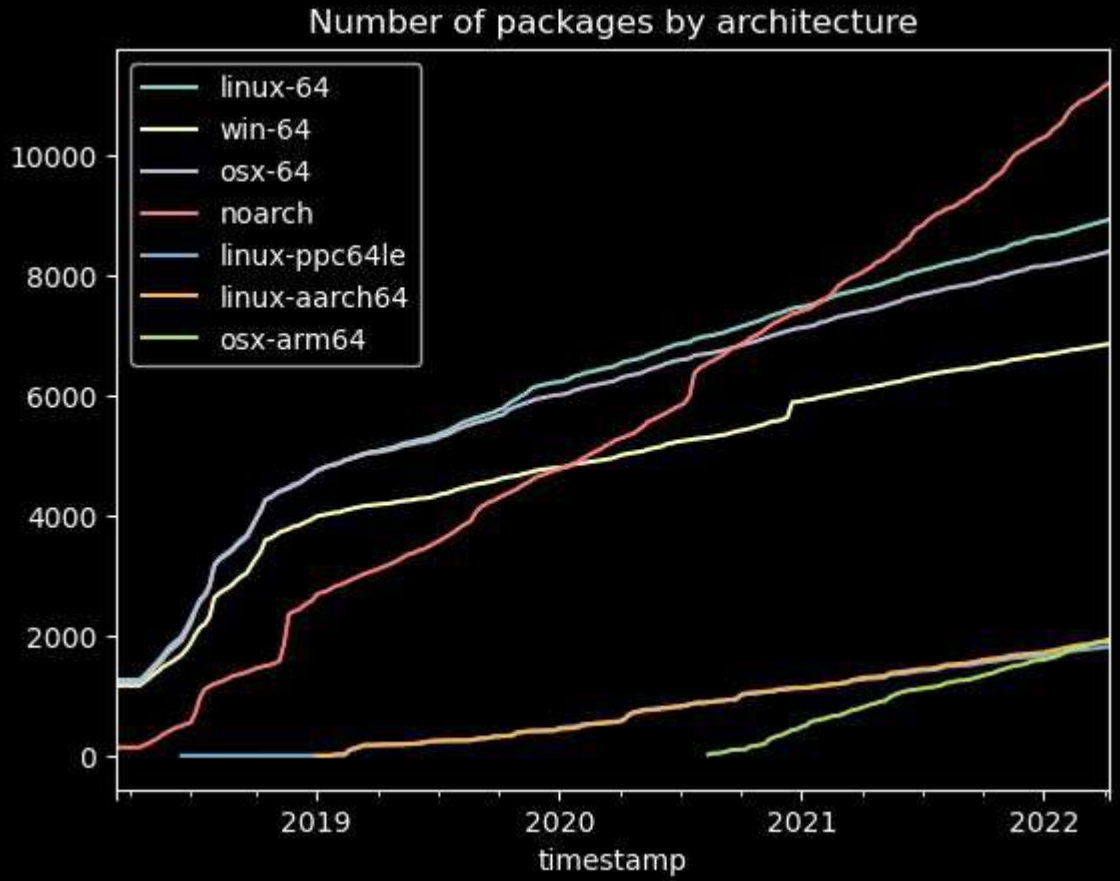
```
conda install -c conda-forge numpy
```



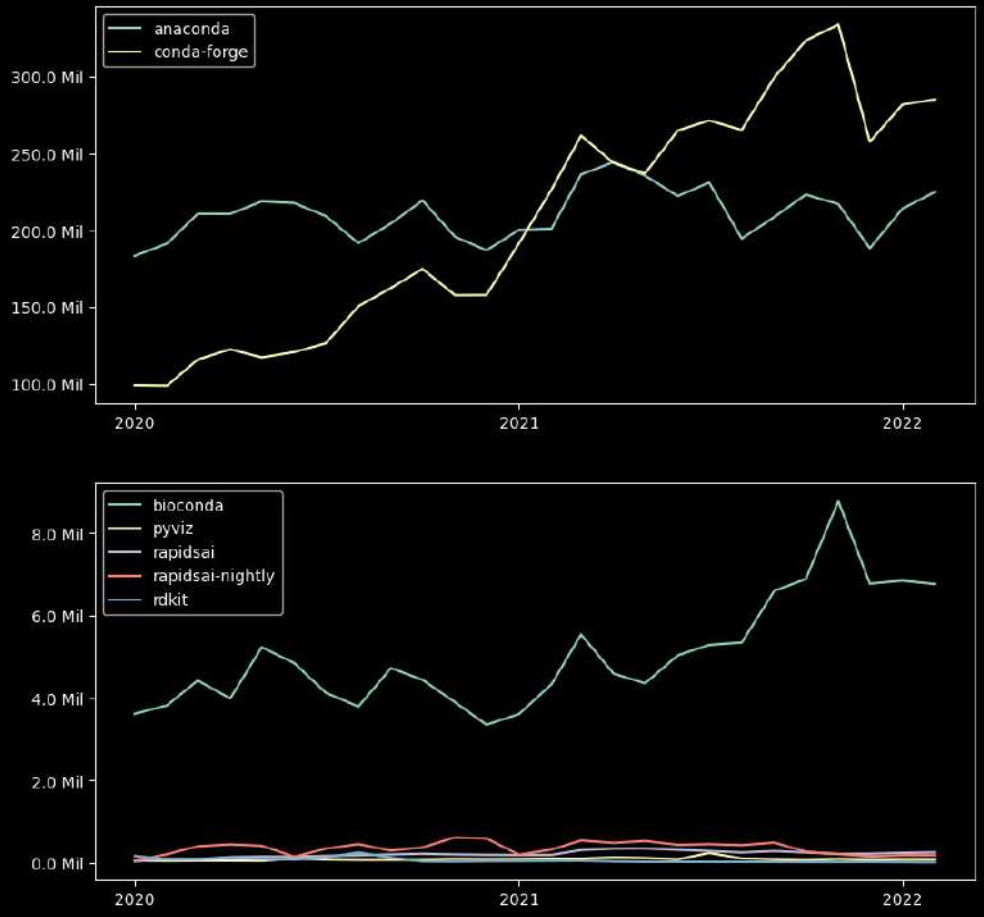
3. Growth-driven innovation


Number of packages over time
17500
15000
12500
10000
7500
5000
2500
0
2019
2020
2021
2022
timestamp


Number of artifacts across all platforms
800000
600000
400000
200000
0
2019
2020
2021
2022
timestamp


Number of packages by architecture
linux-64
win-64
osx-64
noarch
linux-ppc64le
linux-aarch64
osx-arm64
10000
8000
6000
4000
2000
0
2019
2020
2021
2022
timestamp

Monthly Downloads by channel



Growing pains ⚡

Infrastructure

- Many concurrent package builds
- CDN “delay”: the speed with which the channel is re-indexed & travels to CDN
- Managing repository integrity without rebuilding the world

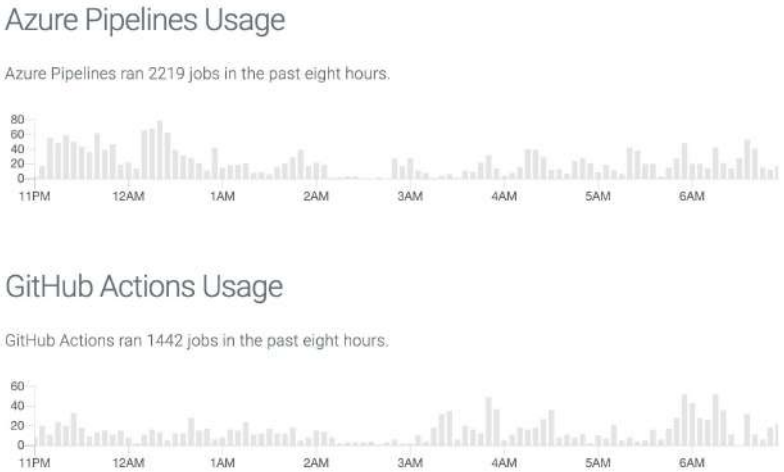
Client side

- Large index files to be downloaded each time
  - ~120 Mb for Linux / 23 Mb gzip’d
- Huge solution space for the satisfiability solver (SAT solver)
  - long solving times

ON THE **CLOUD** SIDE

Azure, GitHub Actions, cirun.io

- conda-forge “burned” through many CI services: Travis, Circle CI, Drone, …
- Running at conda-forge scale: 200 parallel Azure runners
- Since a couple weeks: 60 parallel Github Actions runners
- Currently investigating cirun.io for long running builds
  - We have a time limit of 3 hours & 2 cores for builds on Azure
  - Qt, TensorFlow, Pytorch run longer → need for custom runners
- Pre-test PRs using our bots & mamba to avoid unsolvable PRs



ON THE **CLOUD** SIDE

Failovers: more mirrors

- Setup an OCI-registry mirror on Github packages (similar to Homebrew)
  - OCI registry is a vendor neutral spec implemented by many hosts
  - Available on https://github.com/orgs/channel-mirrors/packages
- Might lead to a faster *CDN sync* and faster build/migration times
- Ideally: Linux-style mirrors around the globe (like Universities, Telecom providers, …)

https://github.com/mamba-org/conda_oci_mirror
https://github.com/regro/cf-oci-mirror-action

ON THE **CLOUD / CLIENT** SIDE

Repodata workarounds in place

current_repodata.json

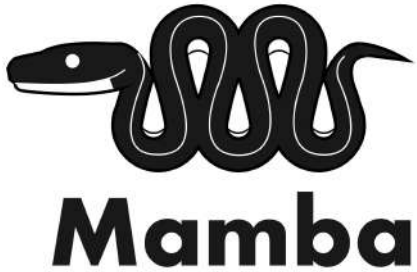
- A smaller package index only containing the latest packages
  - the latest version of each package
  - any earlier versions of dependencies needed to make the latest versions satisfiable

conda-forge-repodata-patches

- Repodata patching
  - Instead of rebuilding packages with “corrected” metadata we patch the repodata
  - Missing upper bounds on dependencies, wrong prioritization, …

ON THE **CLIENT** SIDE

Mamba



- Faster solving, downloading and extracting
- Using
  - C++ as implementation language
  - libsolv for fast package resolution (also used by OpenSuse / RedHat's dnf)
  - libcurl for parallel downloads
  - libarchive for parallel extraction

- libsolv is using a back-tracking SAT solver vs. global optimization employed by conda

https://github.com/mamba-org/mamba

ON THE **CLIENT** SIDE

Mamba CZI grant: more improvements coming!

- Use *zchunk* for repodata – this will allow smaller repodata downloads
  - Only download updated chunks of repodata
- Support mirrors for distributed, fast & reliable downloads
  - Mirror support with automatic fastest mirror selection
  - OCI registries and S3 buckets support
  - Ongoing work to use Github packages OCI registry as cf-mirror
- Better error messages, inspired by PubGrub
  - Make error messages less confusing and add relevant information

Reusable implementation in **libpowerloader**

https://wolfv.medium.com/the-mamba-project-and-the-czi-grant-ec88fb27c25
https://github.com/mamba-org/powerloader

mambabuild & boa

conda mambabuild

- Monkey-patching conda-build to use libmamba as solver
- Default on conda-forge now

boa build …

- Introduces a new recipe format (pure YAML)
- Use mamba as a solver

https://github.com/mamba-org/boa

Continuing the growth

Platforms:

- osx-arm64, linux-aarch64 & linux-ppc64le already available
- Experimental emscripten-32 support (emscripten-forge)

Programming languages:

- Python, C/C++, R – historically good support among many others
- Rust and Go have great support now
- Lots of other up-and-coming ecosystems: Julia, Zig, Nim, …

4. The future of the conda ecosystem

Behind conda-forge: challenges in conda

- Governance of "*conda and friends*" trails conda-forge
- Code bases need to cater to changing user base
- CEPs are central in future
- New maintenance and release strategy (CEP 8)
- Focus on and support of community efforts
- *Maintenance is a process*

Open conda and conda-build issues over the years

Supporting mamba from conda

- conda-forge has additional requirements given its size and scope compared to Anaconda's channels or other smaller conda channels
- mamba clearly helping users with improved user experience
- What if parts of mamba would be integrated in conda?
  - 2021 work on building a integration layer between libmamba and conda
  - Available via conda-forge and Anaconda's defaults channel
  - Experimental release of conda-libmamba-solver in Q1/2022 to get feedback
  - Stable release expected in Q4/2022 with further optimizations
- More opportunities for collaboration on user-facing features, e.g. error reporting, I/O backend and similar user experience improvements

Scaling for the community needs

- Lots of other components in the larger *“conda and friends”* ecosystem
- New plugin API (CEP 4) to achieve goals:
  - Improve usage of conda APIs and cater to “Hyrum's Law”:

    With a sufficient number of users of an API,

    it does not matter what you promise in the contract:

    all observable behaviors of your system

    will be depended on by somebody.

  - Facilitate additional use cases that don’t fit conda’s scope
  - Allow more community innovation to happen
  - Comprehensive documentation of explicit plugin APIs

Updated conda/-incubator governance policy

- conda-forge **pioneered** community-driven innovation and required governance structure to scale it
- **Leaner steering council** with maximum number of 2 members per “funding source” to correct historic imbalance and prevent capture
- Provisional memberships to **onboard underrepresented community members** and organizations
- Submitted application for **fiscal sponsorship via NumFOCUS nonprofit** to pave the way for further project independence
- New Code of Conduct with **dedicated Code of Conduct team** for a more inclusive community
- Project and federated teams to provide **organizational infrastructure for community projects**

conda-forge provides packages for all

How can you help conda-forge?

If you like:

- **Adding** new functionality
- **Keeping** things running
- **Automation** (like you're obsessed with automation, maybe *too* obsessed)
- Financially **supporting**

Then you should consider:

- Staged-recipes review team
- General Q&A in gitter
- General Feedstock maintenance
- Documentation
- Bot team (docs, source, gitter)
- NumFOCUS donation, earmarked for conda-forge
- Donations of servers, VMs or cloud storage (Anaconda, NVIDIA, OVHCloud, Quansight)

FINANCIAL SUPPORT

INFRASTRUCTURE SUPPORT

DEVELOPER SUPPORT

Thanks!

www.conda-forge.org

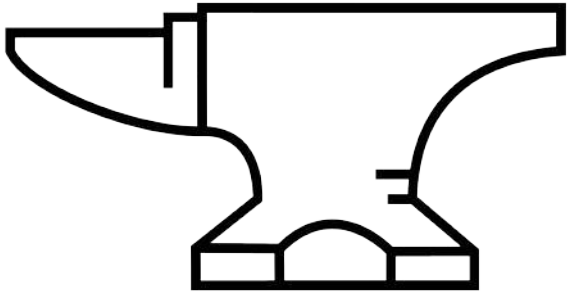
github.com/conda-forge

twitter.com/condaforge

gitter.im/conda-forge/conda-forge.github.io

PyLadies Lunch @ EuroScipy:
Wednesday 31st August – Room 028

everyone is welcome



conda-forge

supporting the growth of the volunteer-driven, community-based packaging project

Wolf Vollprecht (QuantStack) · Jannis Leidel (Anaconda) · Jaime Rodríguez-Guerra (Quansight)

EuroSciPy, 2022 · Aug 31st