Introduction to conda

For Scientific Robotic Software in Research and Development

2021-06-09

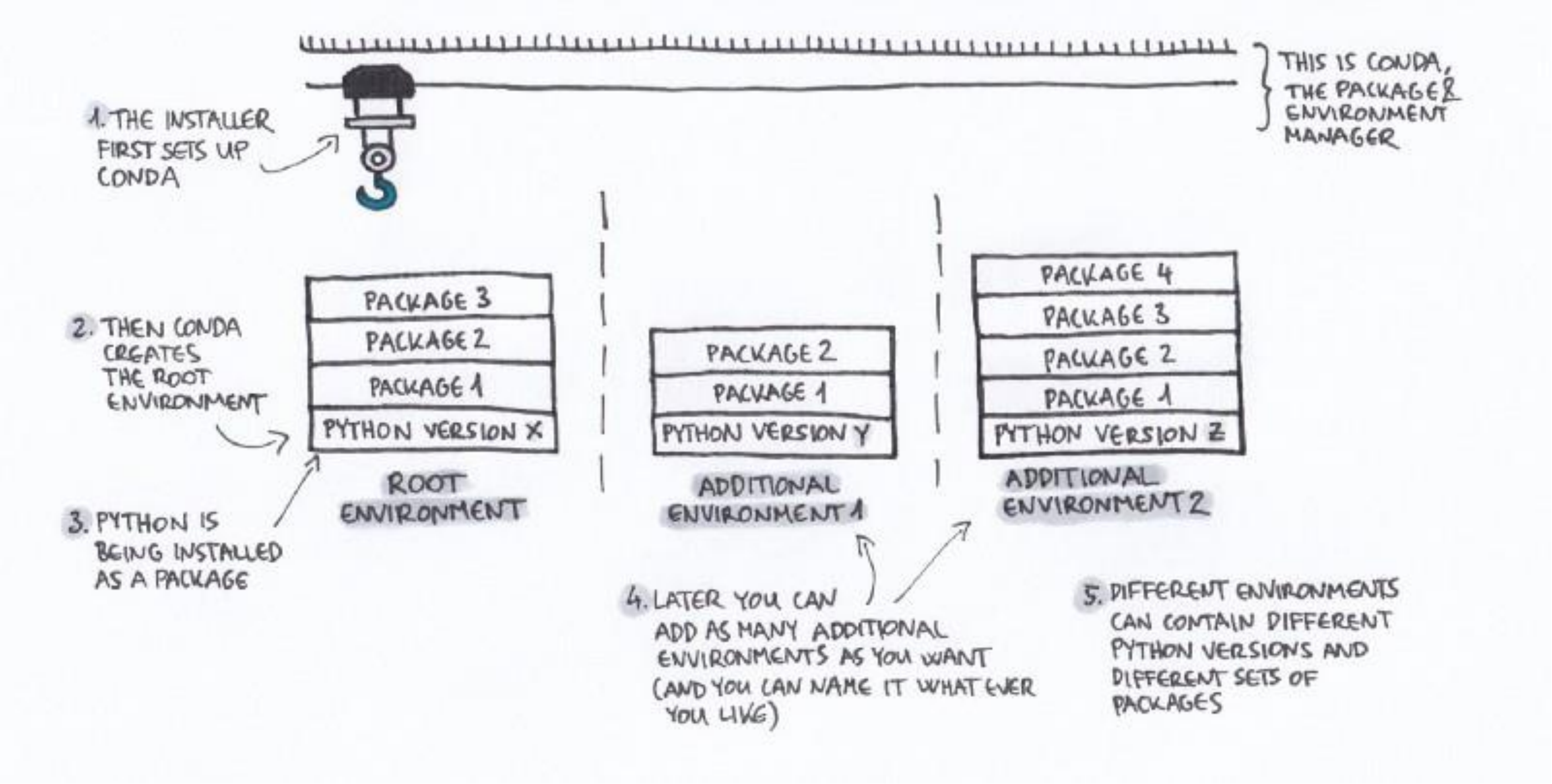
Silvio Traversaro

What is Conda?

- A **package manager** (like apt, pip, vcpkg, homebrew…)
  - Open source
  - Cross-platform (Linux, macOS, Windows)
  - Language agnostic (you can install C++, Python, MATLAB packages)
  - Packaging binaries that natively run on your system; no additional layer as in e.g. Docker
- An **environment management system**
  - Each "environment" do not interfere with each other
  - Living in your **home directory,** no root access required
  - This can be naturally used to easily switch between different "environments" corresponding to different projects. In this way you can easily switch between one project and another, even having **different versions** of dependencies in each environment.
- Based on **channels**
  - Software in conda can be installed from multiple **channels,** similar to PPAs on Ubuntu
  - The most used channel in the open source community is **conda-forge,** that is effectively a a cross-platform distribution
  - Other channels mentioned in this presentation will be **robostack** (channel with ROS packages) and **robotology** (channel with software from robotology-superbuild)

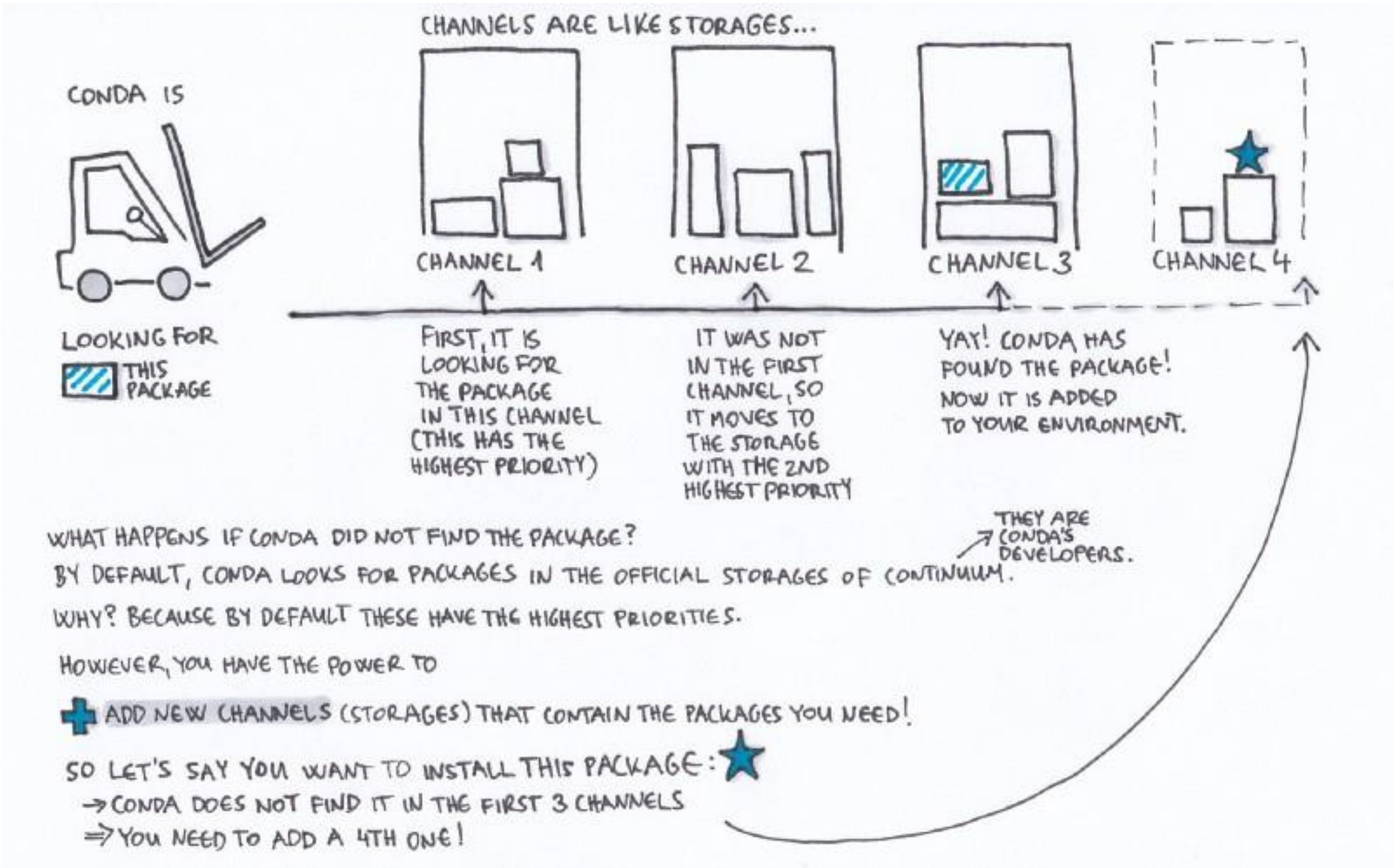
Based on https://www.youtube.com/watch?v=4sVeqG3eP0A

Environments in conda


THIS IS CONDA, THE PACKAGE & ENVIRONMENT MANAGER
1. THE INSTALLER FIRST SETS UP CONDA
2. THEN CONDA CREATES THE ROOT ENVIRONMENT
3. PYTHON IS BEING INSTALLED AS A PACKAGE
PACKAGE 3
PACKAGE 2
PACKAGE 1
PYTHON VERSION X
ROOT ENVIRONMENT
PACKAGE 2
PACKAGE 1
PYTHON VERSION Y
ADDITIONAL ENVIRONMENT 1
PACKAGE 4
PACKAGE 3
PACKAGE 2
PACKAGE 1
PYTHON VERSION Z
ADDITIONAL ENVIRONMENT 2
4. LATER YOU CAN ADD AS MANY ADDITIONAL ENVIRONMENTS AS YOU WANT (AND YOU CAN NAME IT WHAT EVER YOU LIKE)
5. DIFFERENT ENVIRONMENTS CAN CONTAIN DIFFERENT PYTHON VERSIONS AND DIFFERENT SETS OF PACKAGES

From https://geohackweek.github.io/datasharing/01-conda-tutorial/

Channels in conda



FIRST, IT IS LOOKING FOR THE PACKAGE IN THIS CHANNEL (THIS HAS THE HIGHEST PRIORITY)

IT WAS NOT IN THE FIRST CHANNEL, SO IT MOVES TO THE STORAGE WITH THE 2ND HIGHEST PRIORITY

YAY! CONDA HAS FOUND THE PACKAGE! NOW IT IS ADDED TO YOUR ENVIRONMENT.

WHAT HAPPENS IF CONDA DID NOT FIND THE PACKAGE?

BY DEFAULT, CONDA LOOKS FOR PACKAGES IN THE OFFICIAL STORAGES OF CONTINUUM. → THEY ARE CONDA'S DEVELOPERS.

WHY? BECAUSE BY DEFAULT THESE HAVE THE HIGHEST PRIORITIES.

HOWEVER, YOU HAVE THE POWER TO

✚ ADD NEW CHANNELS (STORAGES) THAT CONTAIN THE PACKAGES YOU NEED!

SO LET'S SAY YOU WANT TO INSTALL THIS PACKAGE: ★

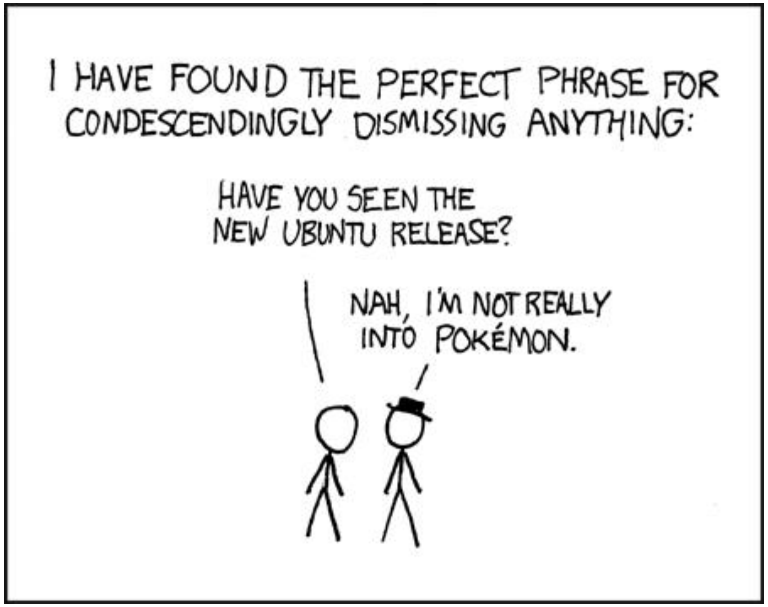
→ CONDA DOES NOT FIND IT IN THE FIRST 3 CHANNELS

⇒ YOU NEED TO ADD A 4TH ONE!

From https://geohackweek.github.io/datasharing/01-conda-tutorial/

Terminology (to avoid confusion)

- **conda:** Open source package manager (like apt).
- **conda-forge**: Distribution of software (like Debian) in a forma of a **conda channel,** run by **volunteers** (mostly scientist and research software engineers)
- **Anaconda**: Both the commercial US **company** that created conda and a installer based on conda with many data science packages (**not based** on conda-forge)
- **Miniforge**: A minimal installer for conda, that contains packages from **conda-forge**. It is the suggested installer for this tutorial.


I HAVE FOUND THE PERFECT PHRASE FOR CONDESCENDINGLY DISMISSING ANYTHING:
HAVE YOU SEEN THE NEW UBUNTU RELEASE?
NAH, I'M NOT REALLY INTO POKÉMON.

https://xkcd.com/178/

What is conda-forge channel?

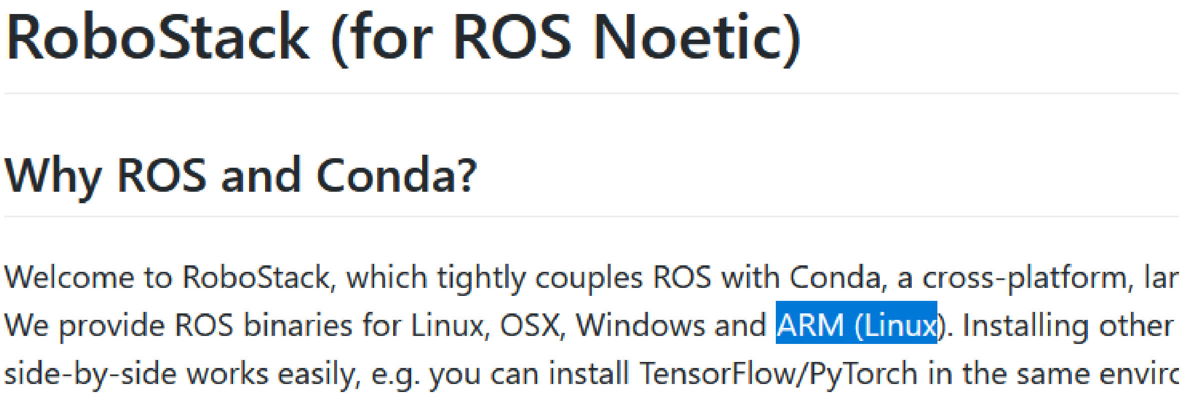
- **conda-forge** is conda channel that contains a scientific distribution of software . Thinks about **Debian** (if you know what it is, more like **Arch**).
- It is built and maintain by people doing scientific software.
- It started (with anaconda) as a **Python distribution**, but not can be used just for C++ without even the need to install Python.
- It is maintained by scientist and research software engineers for different fields, for examples see https://www.nature.com/articles/s41592-018-0046-7 , a paper on **Nature** on the application of **conda-forge for life sciences**, or https://indico.cern.ch/event/759388/contributions/3306849/attachments/1816254/2968550/root_conda_forge.pdf a presentation on how conda-forge is used at **CERN** for physics software.

CONDA-FORGE
A community-led collection of recipes, build infrastructure and distributions for the conda package manager.

https://conda-forge.org/

What is robostack channel?

ROS packages for Linux (amd64/arm64)/macOS/Windows are available in the **robostack** channel : https://github.com/RoboStack/ros-noetic .



RoboStack (for ROS Noetic)

Why ROS and Conda?

Welcome to RoboStack, which tightly couples ROS with Conda, a cross-platform, lar
We provide ROS binaries for Linux, OSX, Windows and ARM (Linux). Installing other
side-by-side works easily, e.g. you can install TensorFlow/PyTorch in the same envirc

The ROS packages are not distributed directly on conda-forge because they tend to break **ABI** (Application Binary Interface) quite frequently, so it is convenient to rebuild them together.

However, **robostack** depends on **conda-forge** for any non-ROS package.

What is robotology channel?

Conda binary packages for the software in the robotology-superbuild are available on the **robotology** conda channel. It contains software related to the iCub robot.

Details: https://github.com/robotology/robotology-superbuild/blob/master/doc/conda-recipe-generation.md

The robotology-superbuild packages are not distributed directly on conda-forge because they tend to break **ABI** (Application Binary Interface) quite frequently, so it is convenient to rebuild them together.
As **robostack**, also **robotology** depends on **conda-forge** for any non-robotology channel, ensuring mutual compatibility with ROS packages (use same OpenCV or PCL)

conda & co advantages

- **Use of binaries:** fast installation, no slow setup as vcpkg
- **ROS available on Windows, macOS and Linux:** provided by RoboStack
- **No need for root permissions or to change the system state:** convenient to install on HPC systems
- **Simplicity of adding a new package in conda-forge:** a new package can be added in conda-forge in few days, as opposed to years in apt/Ubuntu.
- **Updated dependencies:** conda-forge has recent versions of the dependencies, such you can use a recent VTK or PCL (or event a recent compiler) on an old Linux distro.
- **Reproducible enviroments:** conda has built in support for installing exactly the same version of the packages you were using in the past, up to the patch version. This is quite important for reproducibility in scientific research. See https://www.nature.com/articles/d41586-020-02462-7 .

conda & conda-forge limitations

- On Linux, **conda-forge acts as a complete distribution**, so you should not mix binaries/libraries installed with **apt** and the one installed by **conda-forge** .

- On Windows, only **Release** libraries are installed, so if a library uses **C++ STL** you can't build a **Debug** version of your application (just a **RelWithDebInfo**). For installing both **Release** and **Debug** version of a library on Windows, vcpkg remains a better alternative.

- **conda-forge** is a **rolling release distribution**. Older version of software can be always installed for reproducibility, but in conda-forge (being focused on R&D applications) no patches for **security vulnerability** for old versions are provided, differently from LTS Linux Distributions

Hands-on

Miniforge installation

- To use conda, you need to install a minimal distribution. We suggest to install miniforge (https://github.com/conda-forge/miniforge) provided by the conda-forge project.

- Follow the docs in https://github.com/robotology/robotology-superbuild/blob/master/doc/install-miniforge.md . Any problem/feedback?

Hands-on examples

https://github.com/dic-iit/training-conda-robotics#hands-on