Managing Software Packages with

CONDA Virtual Environment

Jason Li
HPC User Services
LSU HPC / LONI
sys-help@loni.org
Louisiana State University
Baton Rouge
Mar 29, 2023
1. **Why Conda?**
   - 1) Scenarios
   - 2) Concepts

2. **Basic Usage**
   - 1) Get Conda
   - 2) Typical workflow
   - 3) Creating a virtual environment
   - 4) Installing software packages

3. **Advanced Tips**
1. Why Conda?
   1) Scenarios
   2) Concepts

2. Basic Usage
   1) Get Conda
   2) Typical workflow
   3) Creating a virtual environment
   4) Installing software packages

3. Advanced Tips
1) Scenarios

- Facing difficulties installing / managing your software packages on HPC?
- Some typical scenarios…
a) Permission denied
1) Scenarios

a) Permission denied

```
[jasonli3@smic2 ~]$ module load python/3.6.2-anaconda-tensorflow
[jasonli3@smic2 ~]$ module list
Currently Loaded Modulefiles:
1) python/3.6.2-anaconda-tensorflow
```
1) Scenarios

a) Permission denied

```
[jasonli3@smic2 ~]$ module load python/3.6.2-anaconda-tensorflow
[jasonli3@smic2 ~]$ module list
Currently Loaded Modulefiles:
  1) python/3.6.2-anaconda-tensorflow
[jasonli3@smic2 ~]$ pip install geos
Collecting geos
  Downloading https://files.pythonhosted.org/packages/49/5b/b8arf74c01187a36aa41b6523deb9b2a59c...
100% [==================================================] 409KB 3.0MB/s
```
a) Permission denied

```
[JasonLi3@smic2 ~]$ module load python/3.6.2-anaconda-tensorflow
[JasonLi3@smic2 ~]$ module list
Currently Loaded Modulefiles:
  1) python/3.6.2-anaconda-tensorflow
[JasonLi3@smic2 ~]$ pip install geos
Collecting geos
  Downloading https://files.pythonhosted.org/packages/49/5b/b8a9c01187a36aa41b6523db9baa59c5c0419d9b09025be9db9ca08e/geos-2.14.1.tar.gz (409kB)
  100% |████████████████████████████████| 409kB 3.0MB/s
File "usr/local/packages/python/3.6.2-anaconda/lib/python3.6/site-packages/pip/utils/__init__.py", line 220, in makedirs
    os.makedirs(path)
File "usr/local/packages/python/3.6.2-anaconda/lib/python3.6/os.py", line 220, in makedirs
You are using pip version 9.0.1, however version 20.0.1 is available.
You should consider upgrading via the 'pip install --upgrade pip' command.
[JasonLi3@smic2 ~]$ 
```
1) Scenarios

a) **Permission denied**

```bash
$ sudo yum install ...
$ sudo apt-get install ...
$ sudo make install
```
1) Scenarios

b) Conflicted packages

• What if I need two packages w/ conflicted dependencies?

Tensorflow 1.13

PyTorch > 1.5
c) Need a different version?

```
[jasonli3@smic2 ~]$ module av python/ r/ perl/
-------- /usr/local/packages/Modules/default/modulefiles/linux-rhel7-ivybridge--------
python/2.7.7-anaconda  python/2.7.13-anaconda-tensorflow  python/3.8.5-anaconda-ood
python/2.7.7/GCC-4.9.0  python/3.6.2-anaconda-tensorflow
python/2.7.10-mkl-mic  python/3.8.5-anaconda
-------- /usr/local/packages/Modules/default/modulefiles/linux-rhel7-ivybridge--------
r/4.0.3/intel-19.0.5  r/4.1.2/gcc-9.3.0
-------- /usr/local/packages/Modules/default/modulefiles/linux-rhel7-ivybridge--------
perl/5.32.0/intel-19.0.5
```
1) Scenarios

d) Over complicated dependencies

Prerequisites

- Yade compilation and execution rely on a number of mandatory and optional external softwares; the
- cmake build system
- gcc compiler (g++); other compilers will not work; you need g++>=4.2 for openMP support
- boost 1.47 or later
- Qt library
- freetype
- libGLViewer
- python, numpy, ipython, sphinx
- matplotlib
- eigen algebra library (minimal required version 3.2.1)
- gdb debugger
- sqlite3 database engine
1) Scenarios

e) Sharing / Migrating your environment

- Huge effort & large disk quota to install

  • What if my colleagues want to use?

  • What if I want to migrate a different cluster?
Any of those apply to you?
1. Why Conda?
   1) Scenarios
   2) Concepts

2. Basic Usage
   1) Get Conda
   2) Typical workflow
   3) Creating a virtual environment
   4) Installing software packages

3. Advanced Tips
2) Concepts

Virtual Environment

CONDA
2) Concepts

a) Conda

- A software

- Installs / updates / manages packages & dependencies

- Creates / loads / switches between virtual environments

- Initially for Python → General purposes

- Advantage: Does NOT need sudo permission!
### 2) Concepts

#### b) Virtual Environment (VE)

- An **environment**
- **Isolated** and **self-contained** to install and manage packages & dependencies

---

![Virtual Environment Diagram]

- **Python 3.8**
  - numpy 1.13
  - scipy 0.19

- **Python 3.10**
  - numpy 1.23
  - pandas 1.5

- **Perl 5.16**
  - BioPerl 1.7

- **R 4.2.2**
  - Seurat 4.3
2) Concepts

b) Virtual Environment (VE)

- An environment
- Isolated and self-contained to install and manage packages & dependencies

Whatever happens in a VE stays in that VE...

OS

<table>
<thead>
<tr>
<th>Python 3.8</th>
<th>Python 3.10</th>
<th>Perl 5.16</th>
</tr>
</thead>
<tbody>
<tr>
<td>numpy 1.13</td>
<td>numpy 1.23</td>
<td>BioPerl 1.7</td>
</tr>
<tr>
<td>scipy 0.19</td>
<td>pandas 1.5</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Perl 5.8</td>
<td>BioPerl 1.6</td>
</tr>
<tr>
<td></td>
<td></td>
<td>R 4.2.2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Seurat 4.3</td>
</tr>
</tbody>
</table>
2) Concepts

b) Virtual Environment (VE)

- An environment
- Isolated and self-contained to install and manage packages & dependencies

All dependencies are installed within the VE

<table>
<thead>
<tr>
<th>OS</th>
<th>Python 3.8</th>
<th>Python 3.10</th>
<th>Perl 5.16</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>numpy 1.13</td>
<td>numpy 1.23</td>
<td>BioPerl 1.7</td>
</tr>
<tr>
<td></td>
<td>scipy 0.19</td>
<td>pandas 1.5</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>R 4.2.2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Perl 5.8</td>
<td>Seurat 4.3</td>
</tr>
</tbody>
</table>
2) Concepts

• Relation

- Conda is a tool to create / manage
- Conda is not the only tool to create / manage
- Usually works with

Virtual Environment
Before we continue…

- Rule of thumb:

  If a software package you need is managed by Conda, you (most likely) can install / manage it by yourself without a problem.
Outlines

1. Why Conda?
   1) Scenarios
   2) Concepts

2. Basic Usage
   1) Get Conda
   2) Typical workflow
   3) Creating a virtual environment
   4) Installing software packages

3. Advanced Tips
1. Why Conda?
   1) Scenarios
   2) Concepts

2. Basic Usage
   1) Get Conda
   2) Typical workflow
   3) Creating a virtual environment
   4) Installing software packages

3. Advanced Tips
1) Get Conda

- First install a conda in your user environment ...?
1) Get Conda

a) Use conda that comes with system-wide python module

   - No installation / disk quota required.

   - Sufficient for most user cases.

   $ module load python

   **Step 1: Can use Conda**

   $ conda init

   **Step 2: Can use Conda without loading python module (recommended)**
1) Get Conda

b) Install miniconda

– Latest version: [https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86_64.sh](https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86_64.sh)

```bash
$ wget https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86_64.sh  # Step 1: Download miniconda
$ chmod u+x Miniconda3-latest-Linux-x86_64.sh  # Step 2: Allow execution
$ ./Miniconda3-latest-Linux-x86_64.sh  # Step 3: Run and follow prompts
```
1. Why Conda?
   1) Scenarios
   2) Concepts

2. Basic Usage
   1) Get Conda
   2) Typical workflow
   3) Creating a virtual environment
   4) Installing software packages

3. Advanced Tips
2) Typical workflow

- Key:

Always use a virtual environment!
2) Typical workflow

To install ...

Create virtual environment

Activate virtual environment

Install software packages

To use ...

Activate virtual environment

Use software packages
Outlines

1. Why Conda?
   1) Scenarios
   2) Concepts

2. Basic Usage
   1) Get Conda
   2) Typical workflow
   3) Creating a virtual environment
   4) Installing software packages

3. Advanced Tips
3) Creating a virtual environment

- Most frequently used commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>conda create -n ENVIRONMENT</code></td>
<td>Create a virtual environment</td>
</tr>
</tbody>
</table>

### 3) Creating a virtual environment

- Most frequently used commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>conda create -n ENVIRONMENT</td>
<td>Create a virtual environment</td>
</tr>
<tr>
<td><code>source</code> activate ENVIRONMENT</td>
<td>Activate a virtual environment</td>
</tr>
</tbody>
</table>

```
(base) [jasonli3@smic2 ~]$ source activate myenv
(myenv) [jasonli3@smic2 ~]$
```

3) Creating a virtual environment

- Most frequently used commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>conda create -n ENVIRONMENT</code></td>
<td>Create a virtual environment</td>
</tr>
<tr>
<td><code>source activate ENVIRONMENT</code></td>
<td>Activate a virtual environment</td>
</tr>
<tr>
<td><code>conda deactivate</code></td>
<td>Deactivate a virtual environment</td>
</tr>
</tbody>
</table>

3) Creating a virtual environment

- Most frequently used commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>conda create -n ENVIRONMENT</td>
<td>Create a virtual environment</td>
</tr>
<tr>
<td>source activate ENVIRONMENT</td>
<td>Activate a virtual environment</td>
</tr>
<tr>
<td>conda deactivate</td>
<td>Deactivate a virtual environment</td>
</tr>
<tr>
<td>conda env list</td>
<td>List all virtual environments</td>
</tr>
</tbody>
</table>

### 3) Creating a virtual environment

- Most frequently used commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>conda create -n ENVIRONMENT</td>
<td>Create a virtual environment</td>
</tr>
<tr>
<td>source activate ENVIRONMENT</td>
<td>Activate a virtual environment</td>
</tr>
<tr>
<td>conda deactivate</td>
<td>Deactivate a virtual environment</td>
</tr>
<tr>
<td>conda env list</td>
<td>List all virtual environments</td>
</tr>
<tr>
<td>conda env remove -n ENVIRONMENT</td>
<td>Remove a virtual environment and all installed packages</td>
</tr>
</tbody>
</table>

**CAUTION! NO CONFIRMATION! IRREVERSIBLE!**

3) Creating a virtual environment

- Most frequently used commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>conda create -n ENVIRONMENT</td>
<td>Create a virtual environment</td>
</tr>
<tr>
<td><code>source activate ENVIRONMENT</code></td>
<td>Activate a virtual environment</td>
</tr>
<tr>
<td>conda deactivate</td>
<td>Deactivate a virtual environment</td>
</tr>
<tr>
<td>conda env list</td>
<td>List all virtual environments</td>
</tr>
<tr>
<td>conda env remove -n ENVIRONMENT</td>
<td>Remove a virtual environment and all installed packages</td>
</tr>
</tbody>
</table>

1. Why Conda?
   1) Scenarios
   2) Concepts

2. Basic Usage
   1) Get Conda
   2) Creating a virtual environment
   3) Typical workflow
   4) Installing software packages

3. Advanced Tips
4) Installing software packages

- Before installation…

**Make sure a virtual environment is activated!**
## 4) Installing software packages

- Most frequently used commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>conda install NAME</td>
<td>Install a software package</td>
</tr>
</tbody>
</table>

### 4) Installing software packages

- **Most frequently used commands**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>conda install NAME</td>
<td>Install a software package</td>
</tr>
<tr>
<td>conda install NAME=VERSION</td>
<td>Install a specific version</td>
</tr>
</tbody>
</table>

### 4) Installing software packages

- **Most frequently used commands**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>conda install NAME</td>
<td>Install a software package</td>
</tr>
<tr>
<td>conda install NAME=VERSION</td>
<td>Install a specific version</td>
</tr>
<tr>
<td>conda install NAME -c CHANNEL</td>
<td>Install from a specific channel (e.g., conda-forge, bioconda, nvidia, ...)</td>
</tr>
</tbody>
</table>

4) Installing software packages

- Most frequently used commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>conda install NAME</code></td>
<td>Install a software package</td>
</tr>
<tr>
<td><code>conda install NAME=VERSION</code></td>
<td>Install a specific version</td>
</tr>
<tr>
<td><code>conda install NAME -c CHANNEL</code></td>
<td>Install from a specific channel (e.g., conda-forge, bioconda, nvidia, …)</td>
</tr>
<tr>
<td><code>conda install NAME1 NAME2 ...</code></td>
<td>Install multiple packages at once (let conda work out dependencies)</td>
</tr>
</tbody>
</table>

## 4) Installing software packages

- Most frequently used commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>conda install NAME</td>
<td>Install a software package</td>
</tr>
<tr>
<td>conda install NAME=VERSION</td>
<td>Install a specific version</td>
</tr>
<tr>
<td>conda install NAME -c CHANNEL</td>
<td>Install from a specific channel (e.g., conda-forge, bioconda, nvidia, …)</td>
</tr>
<tr>
<td>conda install NAME1 NAME2 ...</td>
<td>Install multiple packages at once (let conda work out dependencies)</td>
</tr>
<tr>
<td>conda list</td>
<td>List all installed software package</td>
</tr>
</tbody>
</table>

### 4) Installing software packages

- **Other useful commands**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>conda search NAME</td>
<td>Search available package versions</td>
</tr>
</tbody>
</table>

### 4) Installing software packages

- **Other useful commands**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>conda search NAME</td>
<td>Search available package versions</td>
</tr>
<tr>
<td>conda search NAME -c CHANNEL</td>
<td>Search available package versions in a specific channel</td>
</tr>
</tbody>
</table>

4) Installing software packages

• Other useful commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>conda search NAME</td>
<td>Search available package versions</td>
</tr>
<tr>
<td>conda search NAME -c CHANNEL</td>
<td>Search available package versions in a specific channel</td>
</tr>
<tr>
<td>conda search NAME --info</td>
<td>Search available package versions with details</td>
</tr>
</tbody>
</table>

4) Installing software packages

- Other useful commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>conda search NAME</td>
<td>Search available package versions</td>
</tr>
<tr>
<td>conda search NAME -c CHANNEL</td>
<td>Search available package versions in a specific channel</td>
</tr>
<tr>
<td>conda search NAME --info</td>
<td>Search available package versions with details</td>
</tr>
<tr>
<td>conda update/upgrade NAME</td>
<td>Update a package to the latest available version</td>
</tr>
</tbody>
</table>

## 4) Installing software packages

- **Other useful commands**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>conda search NAME</td>
<td>Search available package versions</td>
</tr>
<tr>
<td>conda search NAME -c CHANNEL</td>
<td>Search available package versions in a specific channel</td>
</tr>
<tr>
<td>conda search NAME --info</td>
<td>Search available package versions with details</td>
</tr>
<tr>
<td>conda update/upgrade NAME</td>
<td>Update a package to the latest available version</td>
</tr>
<tr>
<td>conda uninstall/remove NAME</td>
<td>Remove a package</td>
</tr>
</tbody>
</table>

4) Installing software packages

- Bonus: Hot packages!

  a) PyTorch (w/ GPU support)

$ # Start an interactive job (using qsub or srun)

$ conda create -n torch
$ source activate torch
$ conda install -c pytorch -c nvidia pytorch==1.13.1 torchvision==0.14.1 torchaudio==0.13.1 pytorch-cuda=11.6

4) Installing software packages

• Bonus: Hot packages!

b) Tensorflow (w/ GPU support)

```bash
$ # Start an interactive job on *GPU* nodes (using qsub or srun)

Step 1: Start an interactive job on *GPU* nodes (otherwise will fail)
$ conda create -n tf
$ source activate tf
$ conda install -c conda-forge -c nvidia tensorflow-gpu=2.11 cudatoolkit=11.6 cuda-nvcc=11.6

Step 2: Create a VE and install tensorflow & dependencies

$ mkdir -p $CONDA_PREFIX/etc/conda/activate.d
$ echo 'export XLA_FLAGS=--xla_gpu_cuda_data_dir=$CONDA_PREFIX' >> $CONDA_PREFIX/etc/conda/activate.d/env_vars.sh

Step 3: Run these commands to set up environment

[1] https://anaconda.org/anaconda/tensorflow-gpu
```
• Your workflow should mostly look like…

To install ...

$ conda create ...
$ source activate ...
$ conda install ...

To use ...

$ source activate ...
$ # Do whatever you need to do with the packages
Exercise

- Create a virtual environment
- Search for **SciPy** version and install the second-latest version (as well as dependencies)
- After you are done, type in chat the installed **SciPy** and **Python** version
1. Why Conda?
   1) Scenarios
   2) Concepts

2. Basic Usage
   1) Get Conda
   2) Typical workflow
   3) Creating a virtual environment
   4) Installing software packages

3. Advanced Tips
A little more than the basics...
1) Change Conda path

- Default Conda path:
  - Environments: ~/.conda/envs/
  - Downloaded packages: ~/.conda/pkgs/

```
[jasonli3@smic ~]$ balance
User filesystem quotas for jasonli3 (uid 15827):
  Filesystem      MB used   MB quota
  /home           950        10000
  /work/project   329639      0

Storage allocation      MB used   MB quota
```

Next page...
1) Change Conda path

- **Solution:**

  **Step 1:** Ask your PI to apply for a storage allocation (/project) and add you to it

  **Step 2:** Set up Conda to create / find virtual environments in /project

  **Step 3:** Create your virtual environment and install software package
1) Change Conda path

- Solution:

**Step 1:** Ask your PI to apply for a storage allocation (/project) and add you to it

**Step 2:** Set up Conda to create / find virtual environments in /project

**Step 3:** Create your virtual environment and install software package
1) Change Conda path

a) Method 1: Command lines

$ conda config --add envs_dirs /path/to/envs
Add path to environments

$ conda config --add pkgs_dirs /path/to/pkgs
Add path to downloaded packages

1) Change Conda path

b) Method 2: Configuration file

- Use any text editor to open: ~/.condarc

$ vi ~/.condarc

```
envs_dirs:
- /project/jasonli3/.conda/envs
pkgs_dirs:
- /project/jasonli3/.conda/pkgs
```

2) Share virtual environment

- Scenario:

  I made a huge effort to install an extensive collection of software packages for our group's research needs. I don't want to do it all over again for everyone in our group. Is it possible to just share the virtual environment with them?
2) Share virtual environment

• Solution:

Step 1: Ask your PI to apply for a storage allocation (/project) and add you and your colleagues to it.

Step 2: Set up Conda to create / find virtual environments in /project.

Step 3: Create your virtual environment and install software packages.

Step 4: Ask your colleague to repeat [Step 2].
3) Migrate / clone virtual environment

- Scenario:
  - I have been using LSU HPC cluster. But now I want to switch to LONI and run the exactly same software. How can I do that?

  - I am leaving my current position. But I may continue doing similar research. How can I replicate my environment to a different HPC system in a different institute?
3) Migrate / clone virtual environment

- Solution

<table>
<thead>
<tr>
<th>To …</th>
<th>Run command</th>
</tr>
</thead>
</table>

3) Migrate / clone virtual environment environment

- Solution

To ...

Export virtual environment recipe to file

```yaml
name: spyder
channels:
  - defaults
dependencies:
  - _libgcc_mutex=0.1=main
  - _openmp_mutex=5.1=1_gnu
  - arrow=1.2.3=py310h06a4308_1
  - astroid=2.14.2=py310h06a4308_0
  - attrs=22.1.0=py310h06a4308_0
  - babel=2.11.0=py310h06a4308_0
  - beautifulsoup4=4.11.1=py310h06a4308_0
  - black=22.6.0=py310h06a4308_0
  - blas=1.0=mkl
  - bottleneck=1.3.5=py310ha9d4c09_0
  - brotli=1.0.9=h5eea18b_7
```

3) Migrate / clone virtual environment

- Solution

<table>
<thead>
<tr>
<th>To ...</th>
<th>Run command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Export virtual environment recipe to file</td>
<td>conda env export &gt; myenv.yml</td>
</tr>
<tr>
<td>Create a virtual environment from file</td>
<td>conda env create -f myenv.yml</td>
</tr>
</tbody>
</table>

4) Use virtual environment on Open OnDemand

LSU HPC (SMIC)

LONI (QB2)

Welcome to the LSU HPC OnDemand portal!

With the OnDemand web portal, you can:

- Manage, download and upload files to the HPC systems (click links in the "files" on the top of this page)
- Check allocation balances
- Check disk usage and quotas
- Check job status
- Submit jobs using templates
- Access HPC systems via a terminal
- Run interactive apps such as Jupyter Notebook/Lab and RStudio (click links in the "Interactive Apps" on the top of this page)

Getting started

1. Why Conda?
2. Basic Usage
3. Advanced Tips

[1] https://youtu.be/xk5q8p6QQ_k
4) Use virtual environment on Open OnDemand

[1] https://youtu.be/xk5q8p6QQ_k

1. Why Conda?
2. Basic Usage
3. Advanced Tips
4) Use virtual environment on Open OnDemand

How to:

Step 1: ssh to the cluster you want to use

Step 2: Activate the virtual environment you want to use in Jupyter

Step 3: Install ipykernel

Step 4: Start a Jupyter session in Open OnDemand, and choose the environment in kernel

[1] https://youtu.be/xk5q8p6QQ_k
5) More than Python

- Scenario
  - I need software packages other than Python (R / Perl / Lua / …)
  - I need a different version than the system modules
  - I am using the system’s R module, but having trouble installing some packages (e.g., rgdal)

```r
> install.packages("rgdal")
warning in install.packages("rgdal"):
  "lib = "/home/packages/r/4.1.2/Sk5jengl/lib/R/library", is not writable"

 configure: error: gdal-config not found or not executable.
ERROR: configuration failed for package 'rgdal'
* removing '/home/jasonli3/R/x86_64-pc-linux-gnu-library/4.1/rgdal'
The downloaded source packages are in
  '/tmp/Rtmpd2ctsho/downloaded_packages'
Warning message:
In install.packages("rgdal"):
  installation of package 'rgdal' had non-zero exit status
```
### 5) More than Python

- **Solutions**

  Many non-python packages are managed by Conda too!

<table>
<thead>
<tr>
<th>Languages</th>
<th>To install ...</th>
<th>Run command ...</th>
</tr>
</thead>
<tbody>
<tr>
<td>R</td>
<td></td>
<td>conda install R</td>
</tr>
<tr>
<td>Perl</td>
<td></td>
<td>conda install perl</td>
</tr>
<tr>
<td>Julia</td>
<td></td>
<td>conda install julia -c conda-forge</td>
</tr>
</tbody>
</table>

[1] https://anaconda.org/
Many non-python packages are managed by Conda too!

<table>
<thead>
<tr>
<th>To install ...</th>
<th>Run command ...</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Languages</strong></td>
<td></td>
</tr>
<tr>
<td>R</td>
<td>conda install R</td>
</tr>
<tr>
<td>Perl</td>
<td>conda install perl</td>
</tr>
<tr>
<td>Julia</td>
<td>conda install julia -c conda-forge</td>
</tr>
<tr>
<td><strong>Dependencies</strong></td>
<td></td>
</tr>
<tr>
<td>hdf5</td>
<td>conda install hdf5</td>
</tr>
<tr>
<td>netcdf</td>
<td>conda install libnetcdf -c conda-forge</td>
</tr>
<tr>
<td>FFTW</td>
<td>conda install fftw</td>
</tr>
</tbody>
</table>
## 5) More than Python

- Solutions

Many non-python packages are managed by Conda too!

<table>
<thead>
<tr>
<th>To install …</th>
<th>Run command …</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Languages</strong></td>
<td></td>
</tr>
<tr>
<td>R</td>
<td>conda install R</td>
</tr>
<tr>
<td>Perl</td>
<td>conda install perl</td>
</tr>
<tr>
<td>Julia</td>
<td>conda install julia -c conda-forge</td>
</tr>
<tr>
<td><strong>Dependencies</strong></td>
<td></td>
</tr>
<tr>
<td>hdf5</td>
<td>conda install hdf5</td>
</tr>
<tr>
<td>netcdf</td>
<td>conda install libnetcdf -c conda-forge</td>
</tr>
<tr>
<td>FFTW</td>
<td>conda install fftw</td>
</tr>
</tbody>
</table>

---

[1] [https://anaconda.org/](https://anaconda.org/)
5) More than Python

• One more cool thing...

  – You can use language specific package management tools

<table>
<thead>
<tr>
<th>Language</th>
<th>Tool</th>
</tr>
</thead>
<tbody>
<tr>
<td>Python</td>
<td>pip</td>
</tr>
<tr>
<td>R</td>
<td>install.packages</td>
</tr>
<tr>
<td>Perl</td>
<td>cpan</td>
</tr>
<tr>
<td>Julia</td>
<td>Pkg</td>
</tr>
</tbody>
</table>

  – Packages will be isolated in the virtual environment
5) More than Python

- E.g., Use Conda to solve your R issue
  - Use system’s R module:
    ```
    $ module load r
    $ R
    > install.packages("rgdal")  → Will fail!
    ```
  - Use Conda:
    ```
    $ conda create -n rgdal
    $ source activate rgdal
    $ conda install r-rgdal  → Will succeed!
    ```
6) Dependency resolution

- Scenario

- I need to compile a code from source, but the dependencies are too convoluted

Prerequisites

Yade compilation and execution rely on a number of mandatory and optional external softwares; the

- cmake build system
- gcc compiler (g++); other compilers will not work; you need g++>=4.2 for openMP support
- boost 1.47 or later
- Qt library
- freeglut3
- libQGLViewer
- python, numpy, ipython, sphinx
- matplotlib
- eigen algebra library (minimal required version 3.2.1)
- gdb debugger
- sqlite3 database engine
6) Dependency resolution

- Solution

To install ...

1. Create virtual environment
2. Activate virtual environment
3. Install dependencies
4. Compile code

To use ...

1. Activate virtual environment
2. Use software packages
7) Troubleshooting

a) Conflict with system module

```
(spyder) [jasonli3@smic2 ~]$ module list
Currently Loaded Modulefiles:
  1) python/3.8.5-anaconda

(spyder) [jasonli3@smic2 ~]$ python
Python 3.10.9 (main, Mar 8 2023, 10:47:38) [GCC 11.2.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import numpy as np
/usr/local/packages/python/3.8.5-anaconda/lib/python3.8/site-packages/numpy/__init__.py
Before Intel(R) MKL initialization ensuring correct out-of-the-box operation under cr
ess is not assured. Please install mkl-service to see http://github.com/IntelPy
from . import _distributor_init
 Traceback (most recent call last):
 File "/usr/local/packages/python/3.8.5-anaconda/lib/python3.8/site-packages/numpy/com
 File "/usr/local/packages/python/3.8.5-anaconda/lib/python3.8/site-packages/numpy/com
```

Fail!
7) Troubleshooting

a) Conflict with system module

- Rule of thumb:

    Do **NOT** load system module if you are using your own installation!
b) What if I made a mess?

- I mixed conda / pip back and forth, and broke the environment…

- It may be easier to create a new virtual environment and start fresh…
Take home message

To install …

$ conda create …

$ source activate …

$ conda install …

To use …

$ source activate …

$ # Do whatever you need to do with the packages
Contact us

- Contact user services
  - Email Help Ticket: sys-help@loni.org
  - Telephone Help Desk: +1 (225) 578-0900