ARC Advanced Research Computing

Intermediate HPC

Presented by: Jacob Boschee and Venkat Mahadevan

arc.ubc.ca

Summary of Topics

1. Course Prerequisites and Interactive Examples
2. Array Jobs
    a. Examples
3. Parallel and MPI Workflows
    a. OpenMP vs MPI
    b. Benchmarking and Rightsizing
4. Managing Python Environments
    a. Virtualenv
    b. PIP
    c. Exercises
5. Workflow Optimization Exercises
    a. Understanding Benchmarking Results
    b. Improving Storage Performance

Prerequisites

- Familiarity with command line shells
- Connecting to a system via SSH
- Editing files in terminal
- Experience with scheduling jobs on an HPC system

Array Jobs

- A better way to handle hundreds to thousands of similar jobs
- Submit one job script that walks over a range
- Can be used to vary input parameters or file names for software
- Each sub-job will run independently
- Order of completion is not guaranteed by default

Array Jobs – Examples

- #PBS –J 0-99:3
  - Job will index 34 jobs with ids 0,3,6,9 …, 99
- PBS provides $PBS_ARRAY_INDEX to reference the ID during execution
- Can be used to walk over a directory of input files
- Example script is provided in /scratch/tr-summer-2021/HPC/arrayjobs.sh

Parallel and MPI workflows

- Many scientific software packages support parallel processing
- Allows utilization of multiple core or multiple nodes to speed execution
- Optimizing the number of resources is vital
- Schedulers only consider resources requested not actual utilization

Parallel and MPI workflows – OpenMP vs MPI

- OpenMP uses threaded parallelization or ‘local parallelization’
  - Each thread can run independently on a CPU core
  - Threads share the same memory
  - Tends to be easier to implement on home-grown code than MPI
- MPI distributes work via message passing to MPI threads
  - Biggest advantage is the ability to run on multiple distinct nodes
  - MPI connects over the network and can utilize high speed Infiniband fabrics

Parallel and MPI workflows – Benchmarking and Rightsizing

- Important to fully test your parallel code to make best use of resources
- How software divides the work can lead to various ceilings of parallelism
- Test your software with multiple settings to optimize your future runs
- A better understanding will lead to faster queue times and quicker turnaround

Managing Python Environments

- Python is great for rapid prototyping and has a large variety of packages for scientific computing.
- However, there are many versions of Python and associated packages, which makes keeping track of versions and dependencies difficult.
- Python virtual environments are a way to manage this complexity.

Managing Python Environments

- Anaconda is not well suited for clusters:
  - Installs packages which already exist on the cluster.
  - Installs binaries not optimized for the processor architecture.
  - Makes references to system libraries that are incorrect on clusters.

- Hence it is recommended to use a Python virtual environment instead.

Managing Python Environments - Virtualenv

- Virtualenv is a tool used to create a Python virtual environment.

- Creates a self-contained Python environment with all dependencies that packages can be installed into.

Managing Python Environments - PIP

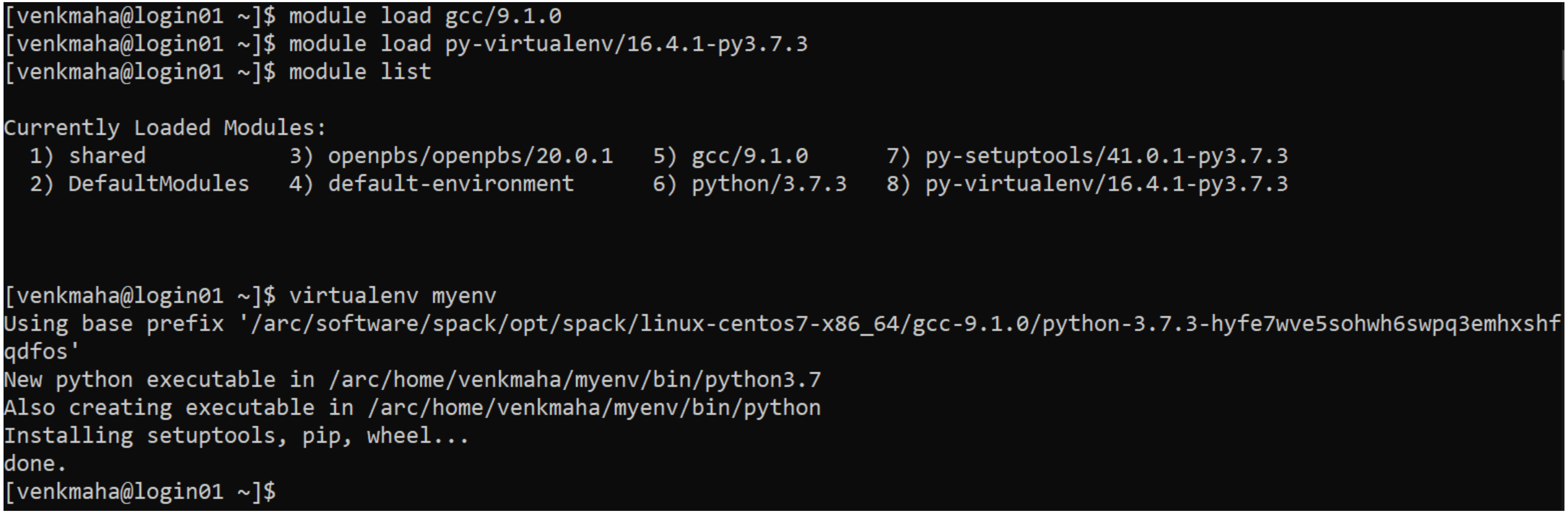
- PIP is the package installer for Python.

- It can be used to install packages from the Python Package Index (PyPI) and other indexes.

- Can be used with virtual environments.

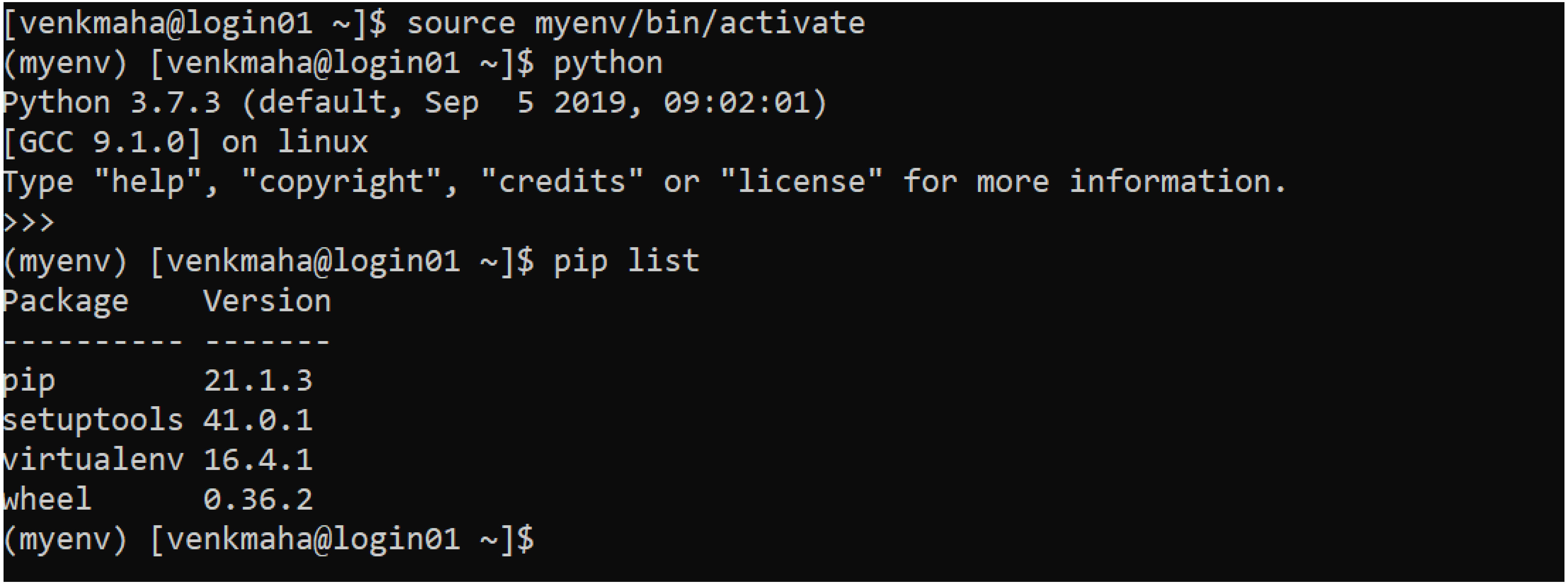
- https://pip.pypa.io/en/stable/cli/

Exercises – create a virtual environment



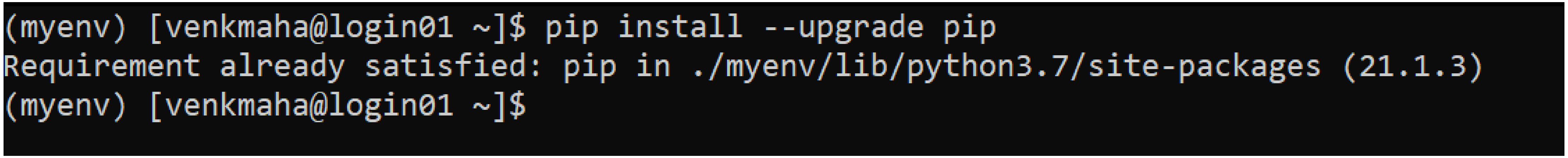
```
[venkmaha@login01 ~]$ module load gcc/9.1.0
[venkmaha@login01 ~]$ module load py-virtualenv/16.4.1-py3.7.3
[venkmaha@login01 ~]$ module list

Currently Loaded Modules:
  1) shared            3) openpbs/openpbs/20.0.1   5) gcc/9.1.0      7) py-setuptools/41.0.1-py3.7.3
  2) DefaultModules    4) default-environment      6) python/3.7.3   8) py-virtualenv/16.4.1-py3.7.3



[venkmaha@login01 ~]$ virtualenv myenv
Using base prefix '/arc/software/spack/opt/spack/linux-centos7-x86_64/gcc-9.1.0/python-3.7.3-hyfe7wve5sohwh6swpq3emhxshf
qdfos'
New python executable in /arc/home/venkmaha/myenv/bin/python3.7
Also creating executable in /arc/home/venkmaha/myenv/bin/python
Installing setuptools, pip, wheel...
done.
[venkmaha@login01 ~]$
```

Exercises – activate the virtual environment



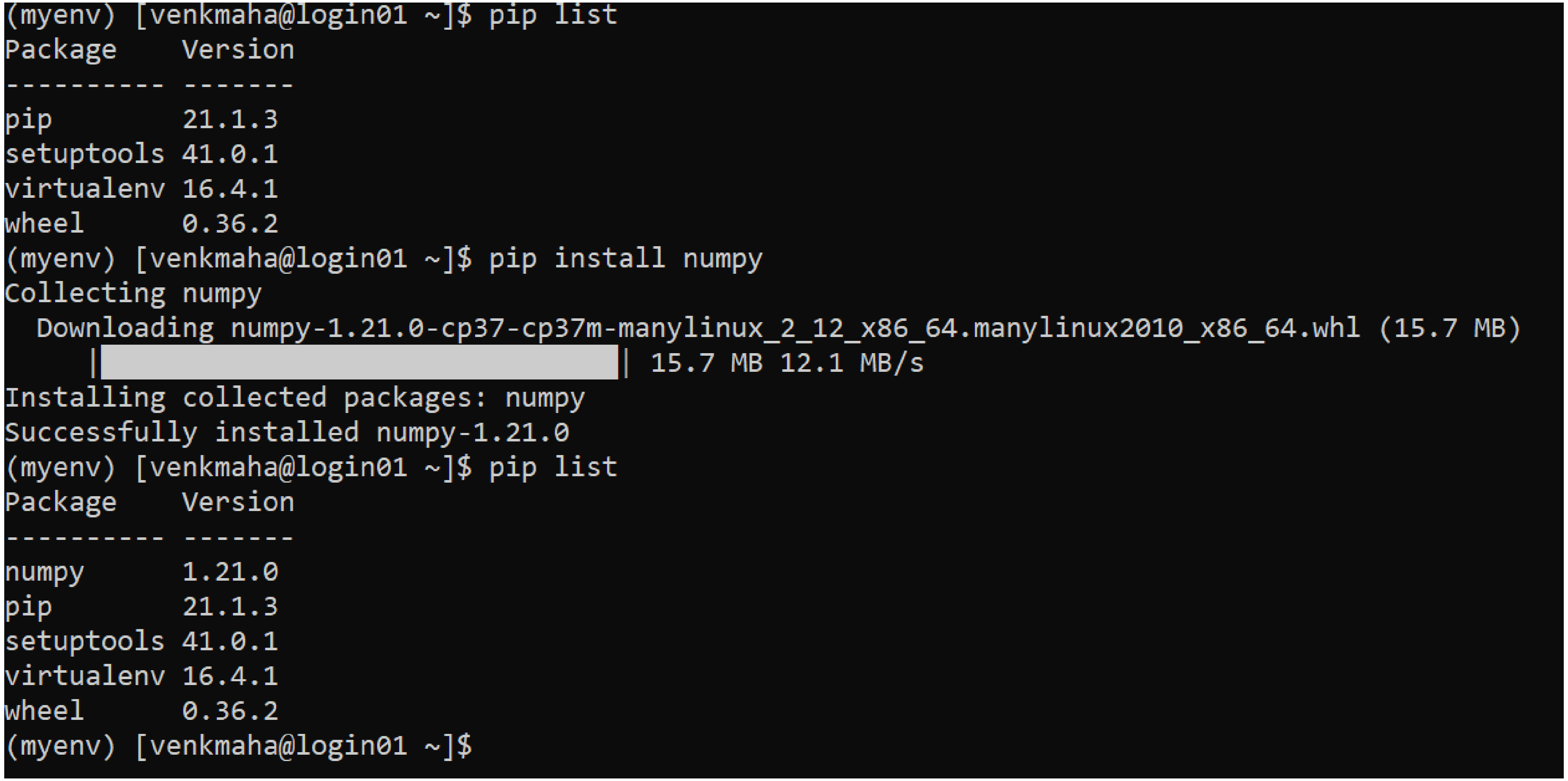
```
[venkmaha@login01 ~]$ source myenv/bin/activate
(myenv) [venkmaha@login01 ~]$ python
Python 3.7.3 (default, Sep  5 2019, 09:02:01)
[GCC 9.1.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>>
(myenv) [venkmaha@login01 ~]$ pip list
Package    Version
---------- -------
pip        21.1.3
setuptools 41.0.1
virtualenv 16.4.1
wheel      0.36.2
(myenv) [venkmaha@login01 ~]$
```

Exercises – upgrade PIP



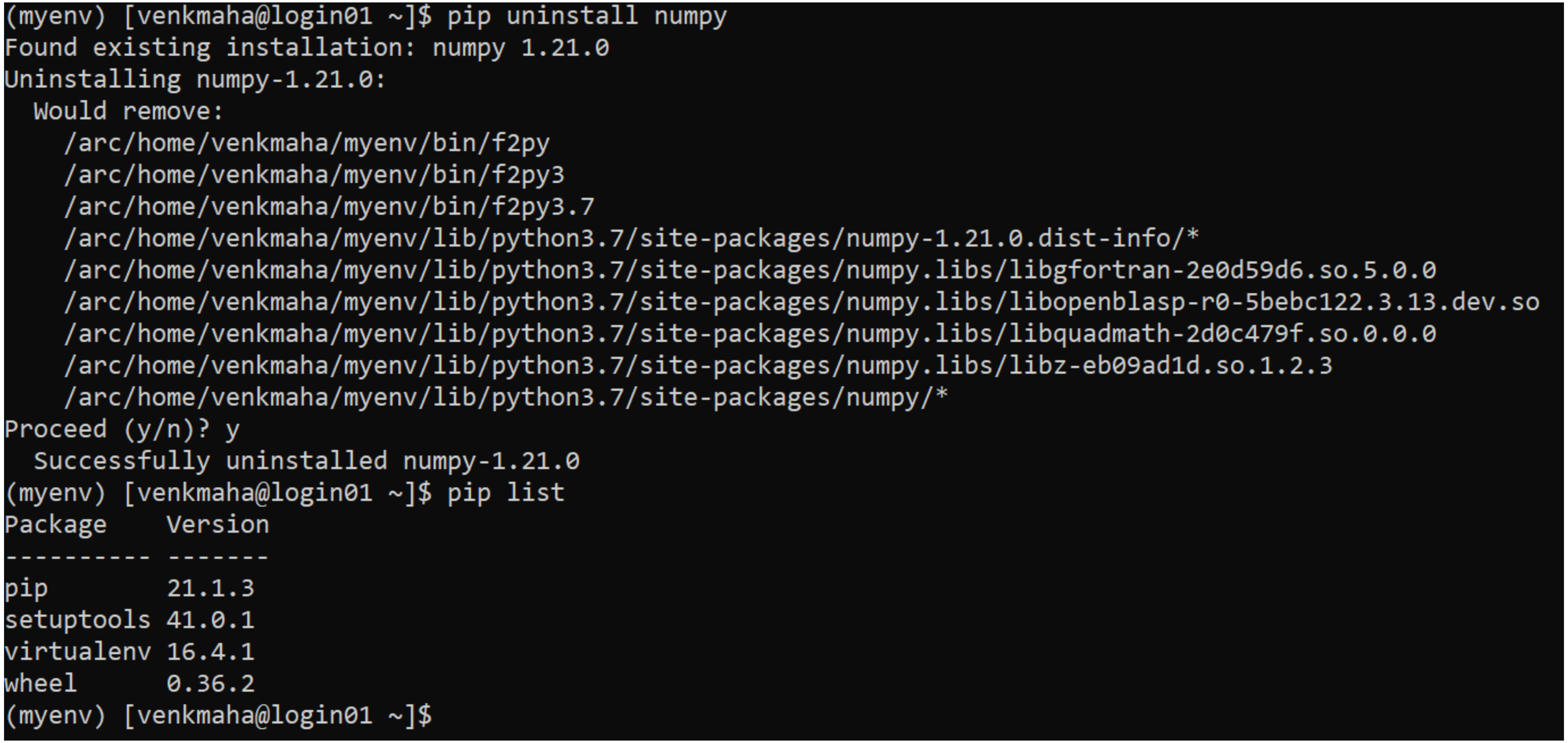
```
(myenv) [venkmaha@login01 ~]$ pip install --upgrade pip
Requirement already satisfied: pip in ./myenv/lib/python3.7/site-packages (21.1.3)
(myenv) [venkmaha@login01 ~]$
```

Exercises – install numpy



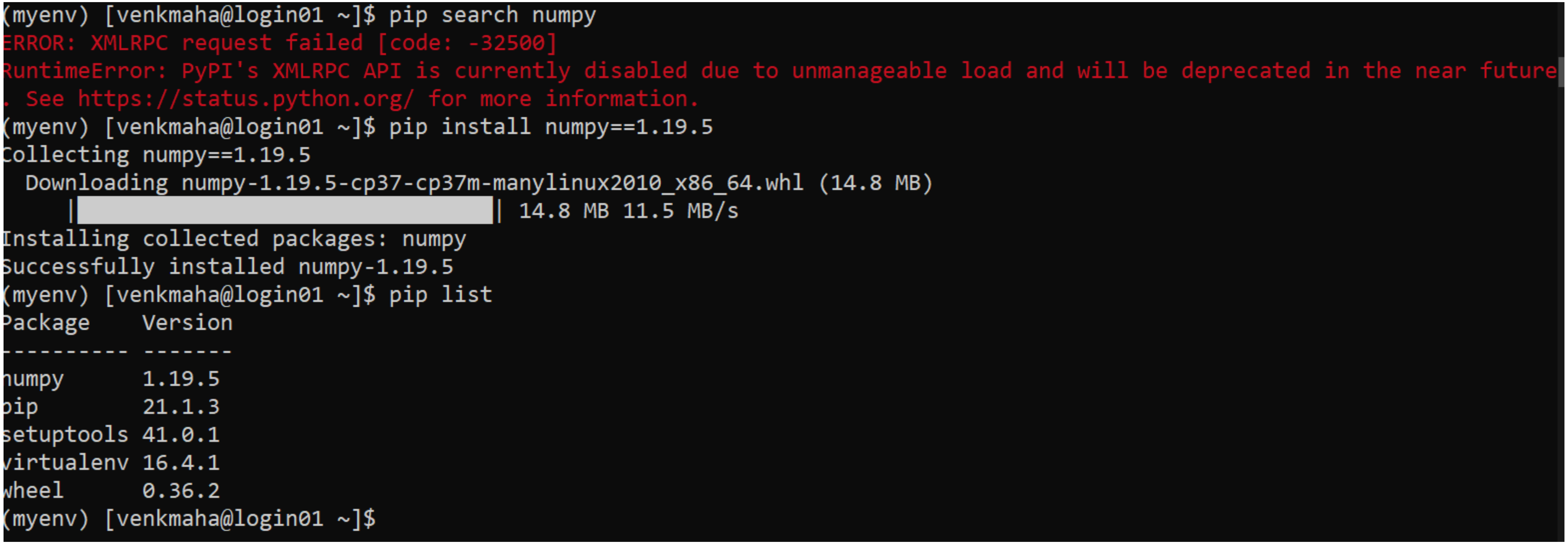
```
(myenv) [venkmaha@login01 ~]$ pip list
Package    Version
---------- -------
pip        21.1.3
setuptools 41.0.1
virtualenv 16.4.1
wheel      0.36.2
(myenv) [venkmaha@login01 ~]$ pip install numpy
Collecting numpy
  Downloading numpy-1.21.0-cp37-cp37m-manylinux_2_12_x86_64.manylinux2010_x86_64.whl (15.7 MB)
     |████████████████████████████████| 15.7 MB 12.1 MB/s
Installing collected packages: numpy
Successfully installed numpy-1.21.0
(myenv) [venkmaha@login01 ~]$ pip list
Package    Version
---------- -------
numpy      1.21.0
pip        21.1.3
setuptools 41.0.1
virtualenv 16.4.1
wheel      0.36.2
(myenv) [venkmaha@login01 ~]$
```

Exercises – uninstall numpy



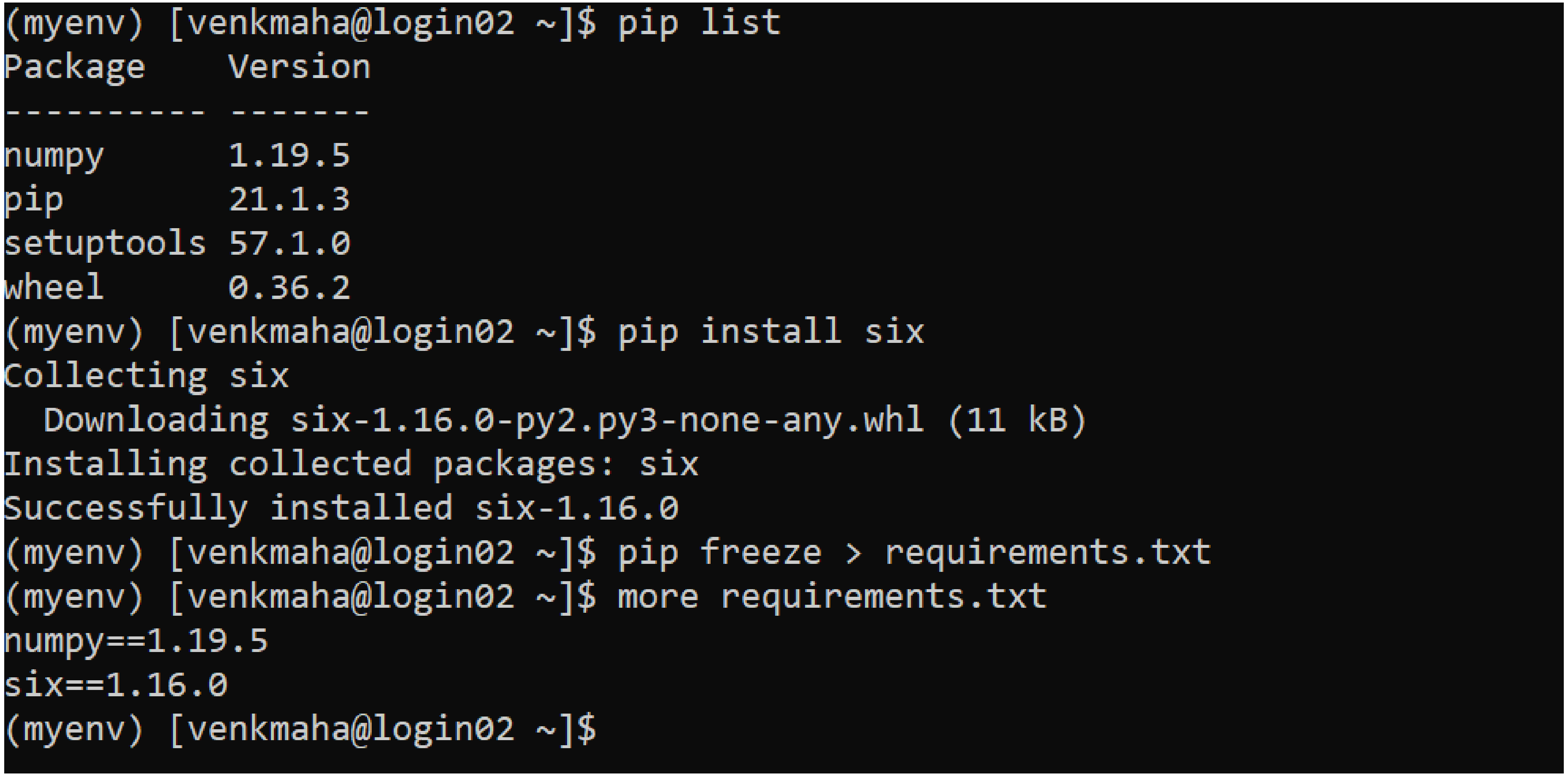
```
(myenv) [venkmaha@login01 ~]$ pip uninstall numpy
Found existing installation: numpy 1.21.0
Uninstalling numpy-1.21.0:
  Would remove:
    /arc/home/venkmaha/myenv/bin/f2py
    /arc/home/venkmaha/myenv/bin/f2py3
    /arc/home/venkmaha/myenv/bin/f2py3.7
    /arc/home/venkmaha/myenv/lib/python3.7/site-packages/numpy-1.21.0.dist-info/*
    /arc/home/venkmaha/myenv/lib/python3.7/site-packages/numpy.libs/libgfortran-2e0d59d6.so.5.0.0
    /arc/home/venkmaha/myenv/lib/python3.7/site-packages/numpy.libs/libopenblasp-r0-5bebc122.3.13.dev.so
    /arc/home/venkmaha/myenv/lib/python3.7/site-packages/numpy.libs/libquadmath-2d0c479f.so.0.0.0
    /arc/home/venkmaha/myenv/lib/python3.7/site-packages/numpy.libs/libz-eb09ad1d.so.1.2.3
    /arc/home/venkmaha/myenv/lib/python3.7/site-packages/numpy/*
Proceed (y/n)? y
  Successfully uninstalled numpy-1.21.0
(myenv) [venkmaha@login01 ~]$ pip list
Package    Version
---------- -------
pip        21.1.3
setuptools 41.0.1
virtualenv 16.4.1
wheel      0.36.2
(myenv) [venkmaha@login01 ~]$
```

Exercises – install a specific version of numpy



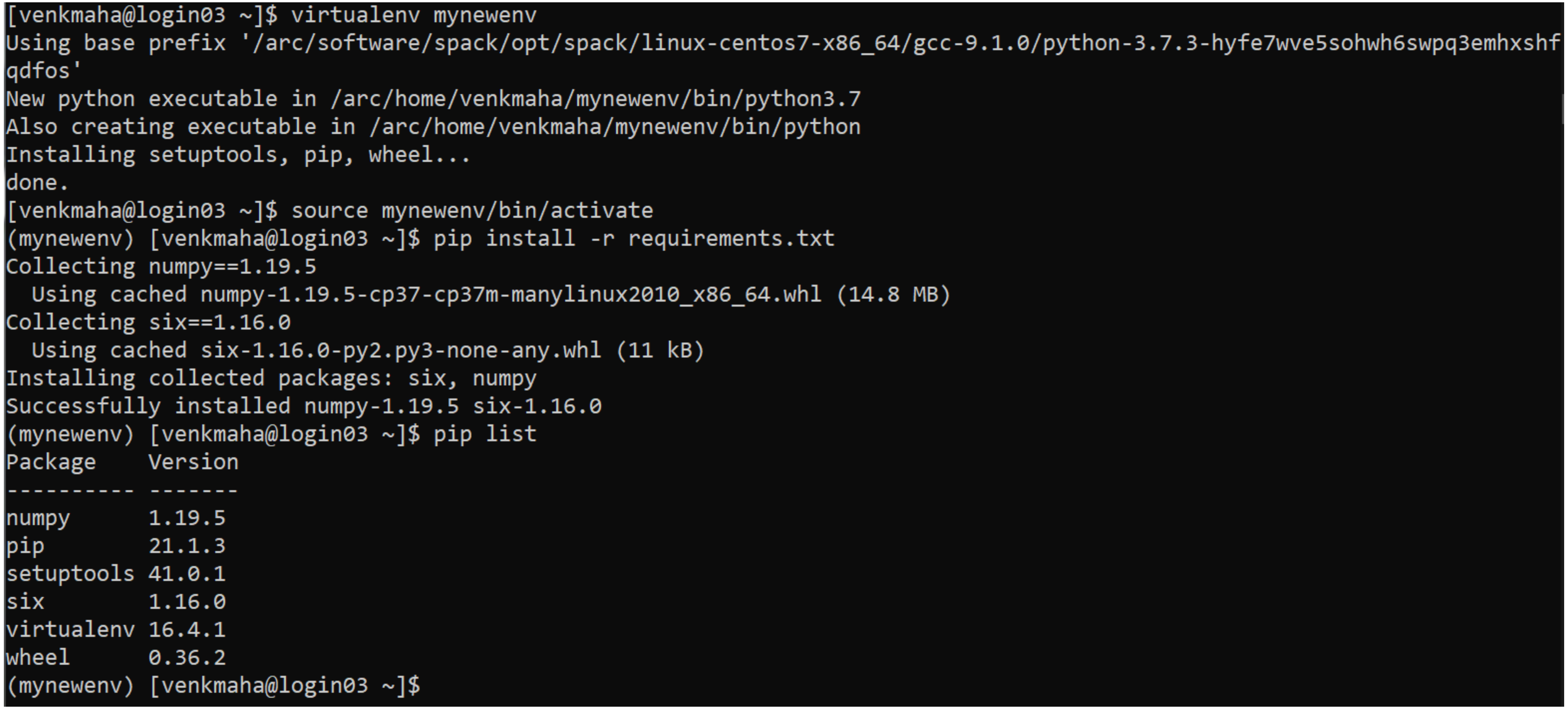
```
(myenv) [venkmaha@login01 ~]$ pip search numpy
ERROR: XMLRPC request failed [code: -32500]
RuntimeError: PyPI's XMLRPC API is currently disabled due to unmanageable load and will be deprecated in the near future
. See https://status.python.org/ for more information.
(myenv) [venkmaha@login01 ~]$ pip install numpy==1.19.5
Collecting numpy==1.19.5
  Downloading numpy-1.19.5-cp37-cp37m-manylinux2010_x86_64.whl (14.8 MB)
     |████████████████████████████████| 14.8 MB 11.5 MB/s
Installing collected packages: numpy
Successfully installed numpy-1.19.5
(myenv) [venkmaha@login01 ~]$ pip list
Package    Version
---------- -------
numpy      1.19.5
pip        21.1.3
setuptools 41.0.1
virtualenv 16.4.1
wheel      0.36.2
(myenv) [venkmaha@login01 ~]$
```

Exercises – freeze a list of requirements



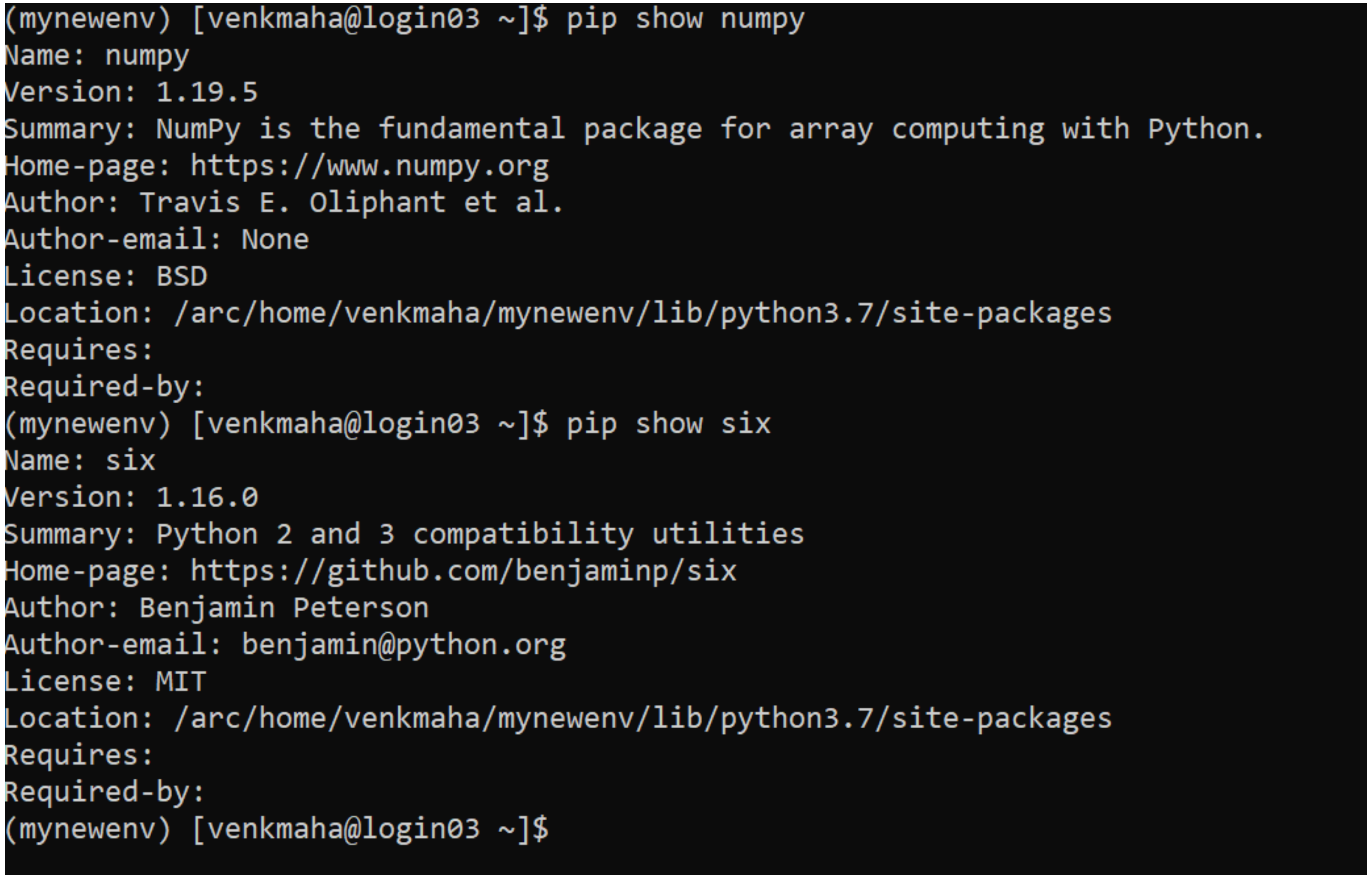
```
(myenv) [venkmaha@login02 ~]$ pip list
Package    Version
---------- -------
numpy      1.19.5
pip        21.1.3
setuptools 57.1.0
wheel      0.36.2
(myenv) [venkmaha@login02 ~]$ pip install six
Collecting six
  Downloading six-1.16.0-py2.py3-none-any.whl (11 kB)
Installing collected packages: six
Successfully installed six-1.16.0
(myenv) [venkmaha@login02 ~]$ pip freeze > requirements.txt
(myenv) [venkmaha@login02 ~]$ more requirements.txt
numpy==1.19.5
six==1.16.0
(myenv) [venkmaha@login02 ~]$
```

Exercises – build and install requirements



```
[venkmaha@login03 ~]$ virtualenv mynewenv
Using base prefix '/arc/software/spack/opt/spack/linux-centos7-x86_64/gcc-9.1.0/python-3.7.3-hyfe7wve5sohwh6swpq3emhxshf
qdfos'
New python executable in /arc/home/venkmaha/mynewenv/bin/python3.7
Also creating executable in /arc/home/venkmaha/mynewenv/bin/python
Installing setuptools, pip, wheel...
done.
[venkmaha@login03 ~]$ source mynewenv/bin/activate
(mynewenv) [venkmaha@login03 ~]$ pip install -r requirements.txt
Collecting numpy==1.19.5
  Using cached numpy-1.19.5-cp37-cp37m-manylinux2010_x86_64.whl (14.8 MB)
Collecting six==1.16.0
  Using cached six-1.16.0-py2.py3-none-any.whl (11 kB)
Installing collected packages: six, numpy
Successfully installed numpy-1.19.5 six-1.16.0
(mynewenv) [venkmaha@login03 ~]$ pip list
Package    Version
---------- -------
numpy      1.19.5
pip        21.1.3
setuptools 41.0.1
six        1.16.0
virtualenv 16.4.1
wheel      0.36.2
(mynewenv) [venkmaha@login03 ~]$
```

Exercises – show package details



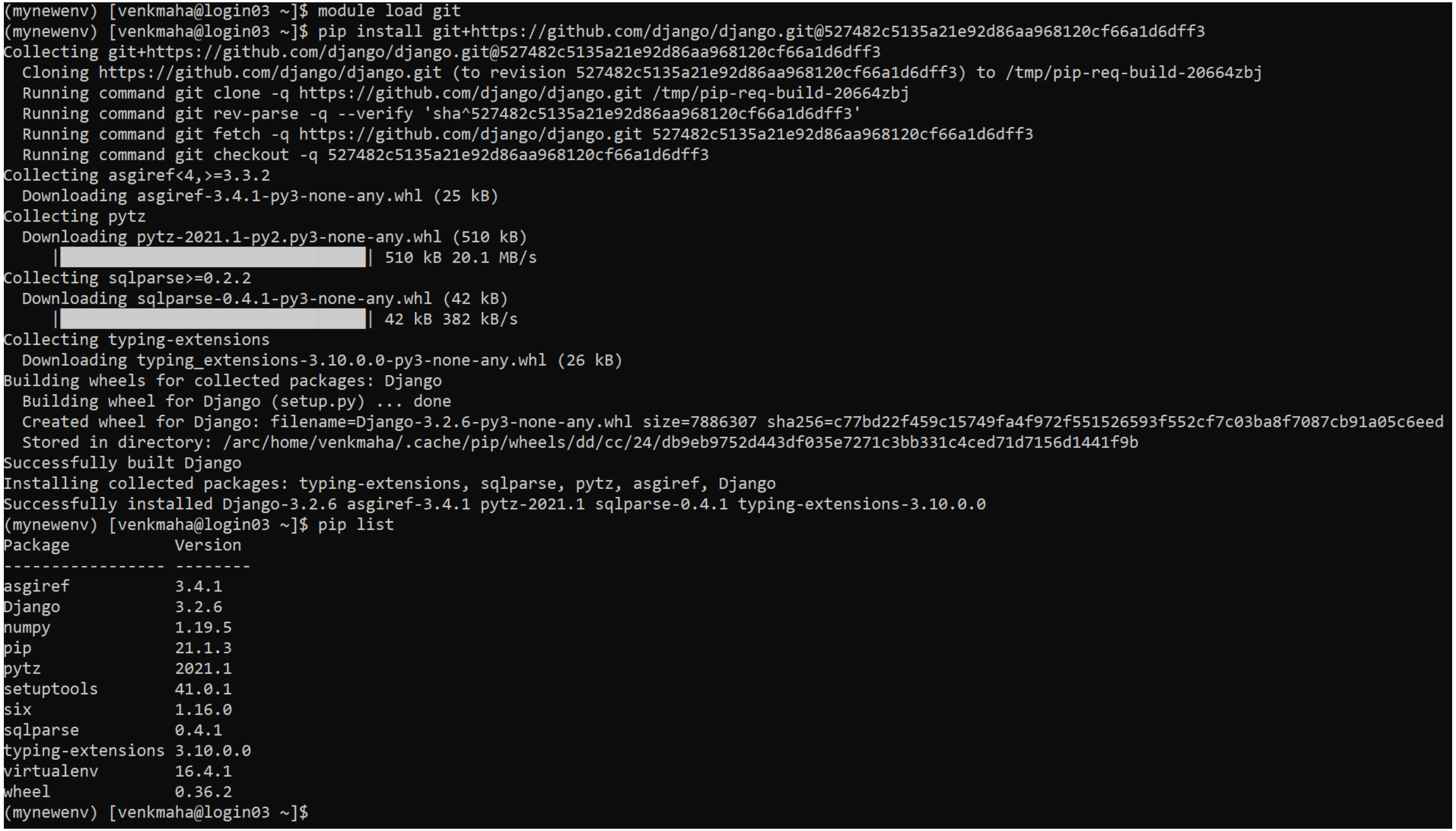
```
(mynewenv) [venkmaha@login03 ~]$ pip show numpy
Name: numpy
Version: 1.19.5
Summary: NumPy is the fundamental package for array computing with Python.
Home-page: https://www.numpy.org
Author: Travis E. Oliphant et al.
Author-email: None
License: BSD
Location: /arc/home/venkmaha/mynewenv/lib/python3.7/site-packages
Requires:
Required-by:
(mynewenv) [venkmaha@login03 ~]$ pip show six
Name: six
Version: 1.16.0
Summary: Python 2 and 3 compatibility utilities
Home-page: https://github.com/benjaminp/six
Author: Benjamin Peterson
Author-email: benjamin@python.org
License: MIT
Location: /arc/home/venkmaha/mynewenv/lib/python3.7/site-packages
Requires:
Required-by:
(mynewenv) [venkmaha@login03 ~]$
```

Exercises – install from Git repo

- module load git

- pip install git+https://github.com/django/django.git@527482c5135a21e92d86aa968120cf66a1d6dff3

Exercises – install from Git repo



```
(mynewenv) [venkmaha@login03 ~]$ module load git
(mynewenv) [venkmaha@login03 ~]$ pip install git+https://github.com/django/django.git@527482c5135a21e92d86aa968120cf66a1d6dff3
Collecting git+https://github.com/django/django.git@527482c5135a21e92d86aa968120cf66a1d6dff3
  Cloning https://github.com/django/django.git (to revision 527482c5135a21e92d86aa968120cf66a1d6dff3) to /tmp/pip-req-build-20664zbj
  Running command git clone -q https://github.com/django/django.git /tmp/pip-req-build-20664zbj
  Running command git rev-parse -q --verify 'sha^527482c5135a21e92d86aa968120cf66a1d6dff3'
  Running command git fetch -q https://github.com/django/django.git 527482c5135a21e92d86aa968120cf66a1d6dff3
  Running command git checkout -q 527482c5135a21e92d86aa968120cf66a1d6dff3
Collecting asgiref<4,>=3.3.2
  Downloading asgiref-3.4.1-py3-none-any.whl (25 kB)
Collecting pytz
  Downloading pytz-2021.1-py2.py3-none-any.whl (510 kB)
     |████████████████████████████████| 510 kB 20.1 MB/s
Collecting sqlparse>=0.2.2
  Downloading sqlparse-0.4.1-py3-none-any.whl (42 kB)
     |████████████████████████████████| 42 kB 382 kB/s
Collecting typing-extensions
  Downloading typing_extensions-3.10.0.0-py3-none-any.whl (26 kB)
Building wheels for collected packages: Django
  Building wheel for Django (setup.py) ... done
  Created wheel for Django: filename=Django-3.2.6-py3-none-any.whl size=7886307 sha256=c77bd22f459c15749fa4f972f551526593f552cf7c03ba8f7087cb91a05c6eed
  Stored in directory: /arc/home/venkmaha/.cache/pip/wheels/dd/cc/24/db9eb9752d443df035e7271c3bb331c4ced71d7156d1441f9b
Successfully built Django
Installing collected packages: typing-extensions, sqlparse, pytz, asgiref, Django
Successfully installed Django-3.2.6 asgiref-3.4.1 pytz-2021.1 sqlparse-0.4.1 typing-extensions-3.10.0.0
(mynewenv) [venkmaha@login03 ~]$ pip list
Package           Version
----------------- --------
asgiref           3.4.1
Django            3.2.6
numpy             1.19.5
pip               21.1.3
pytz              2021.1
setuptools        41.0.1
six               1.16.0
sqlparse          0.4.1
typing-extensions 3.10.0.0
virtualenv        16.4.1
wheel             0.36.2
(mynewenv) [venkmaha@login03 ~]$
```

Workflow Optimization Exercises

- Files for exercises are in /scratch/tr-summer-2021/HPC/workflow
- Job scripts will write out to /scratch/tr-summer-2021/HPC/output/JOB_ID_output.txt
- Find your previous jobs with `qstat –xu USERNAME’ and full job info with `qstat –xf JOB_ID’

Workflow Optimization Exercises – Benchmarking Results

- Utilizing the LAMMPS script benchmarking_exercise.sh determine the optimal number of cores and nodes to run a large number of similar jobs in the future.
- Options are 8 CPUS, 16 CPUs, 32CPUs, and 2 nodes with 32CPUs.
- To save on system load we will present the 16, 32 and multi-node examples and give job ids to inspect the total load.
- Use `qstat –xf JOB_ID’ after completion to look at actual runtime.
- Consider what point adding more processing power provides minimal speedup

Workflow Optimization Exercises – Storage Optimization

- Utilizing the script [[insert script name.sh]] edit it to point the output to $PBS_TMPDIR and copy the file JOB_ID_final.txt to /scratch/tr-summer-2021/HPC/output/ before the job script exits.
- Due to the local nature of the $TMPDIR writes of files are much better performing for frequent file output.
- Once your job is completed move the file from /scratch/tr-summer-2021/HPC/output/ to /arc/project/tr-summer-2021/HPC/StorageOp
- Final results should not be stored on /scratch and should instead be moved to your project space

Workflow Optimization Exercises – ??

arc.ubc.ca

UBC Advanced Research Computing

ARC Advanced Research Computing

arc.ubc.ca

email: arc.info@ubc.ca