CMS Tier-2 Computing Tutorial

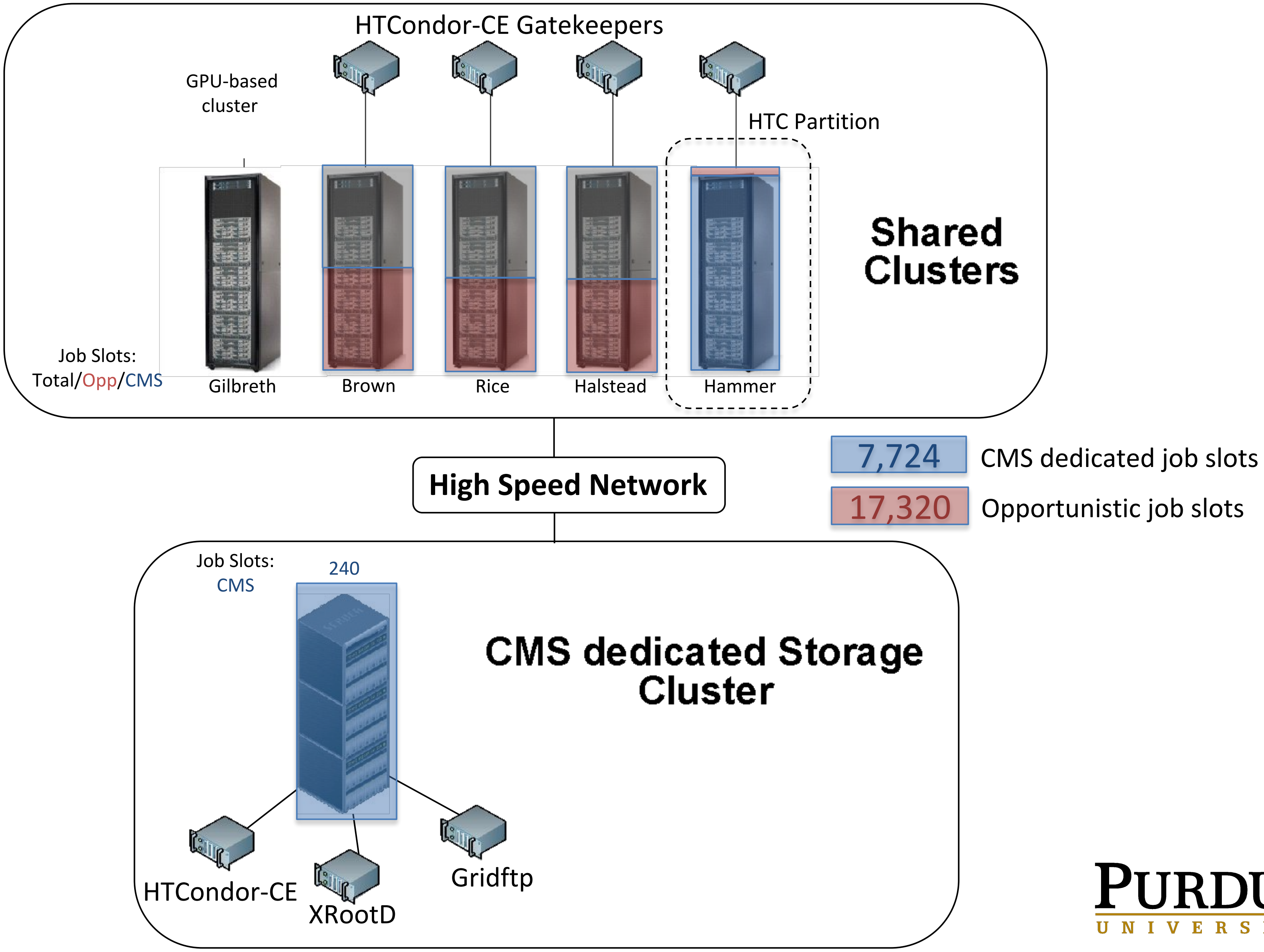
March 6, 2020

Stefan Piperov

spiperov@purdue.edu

PURDUE UNIVERSITY | Research Computing INFORMATION TECHNOLOGY

PURDUE T2 Overview



COMPUTE

- **Dedicated compute resources**
  - **CMS storage cluster**
    - Provides a limited number of batch slots via HTCondor
    - Read-only access to HDFS storage
    - File access protocols (XRootD/Gridftp)
  - **CMS owned Community Cluster nodes**
    - Provides dedicated batch slots via SLURM
    - Our (CMS) main usage is of Hammer cluster
- **Purdue opportunistic resources**
  - A 'standby' queue on each Community Cluster provides short (4h) job slots. If you scale your jobs correctly (less than 4h run time), you get access to a lot of free job slots through CRAB and CMS-Connect

PURDUE UNIVERSITY

STORAGE

- **Home area - private area, source code, development, small**
  - Available on all clusters
  - /home/<username>
- **Data Depot Group space - intermediate, read-write, medium size**
  - Available on all clusters. ***Shared*** by all CMS users!
  - /depot/cms/
  - Individual users may get dedicated sub-directories - e.g.: /depot/cms/users/spiperov/
  - Individual sub-groups have dedicated sub-directories, with *additional* quotas - e.g.: /depot/cms/top
- **Hadoop Distributed Filesystem (HDFS) - long term, large files/datasets**
  - Our main long-term storage space
  - /mnt/hadoop on Hammer and CMS cluster (read-only)

- /scratch - The community clusters provide large, fast local filesystems for *temporary* storage. Cleaned up periodically
- /tmp - on Hammer, this is the main *temporary* space (no /scratch there!)

PURDUE UNIVERSITY

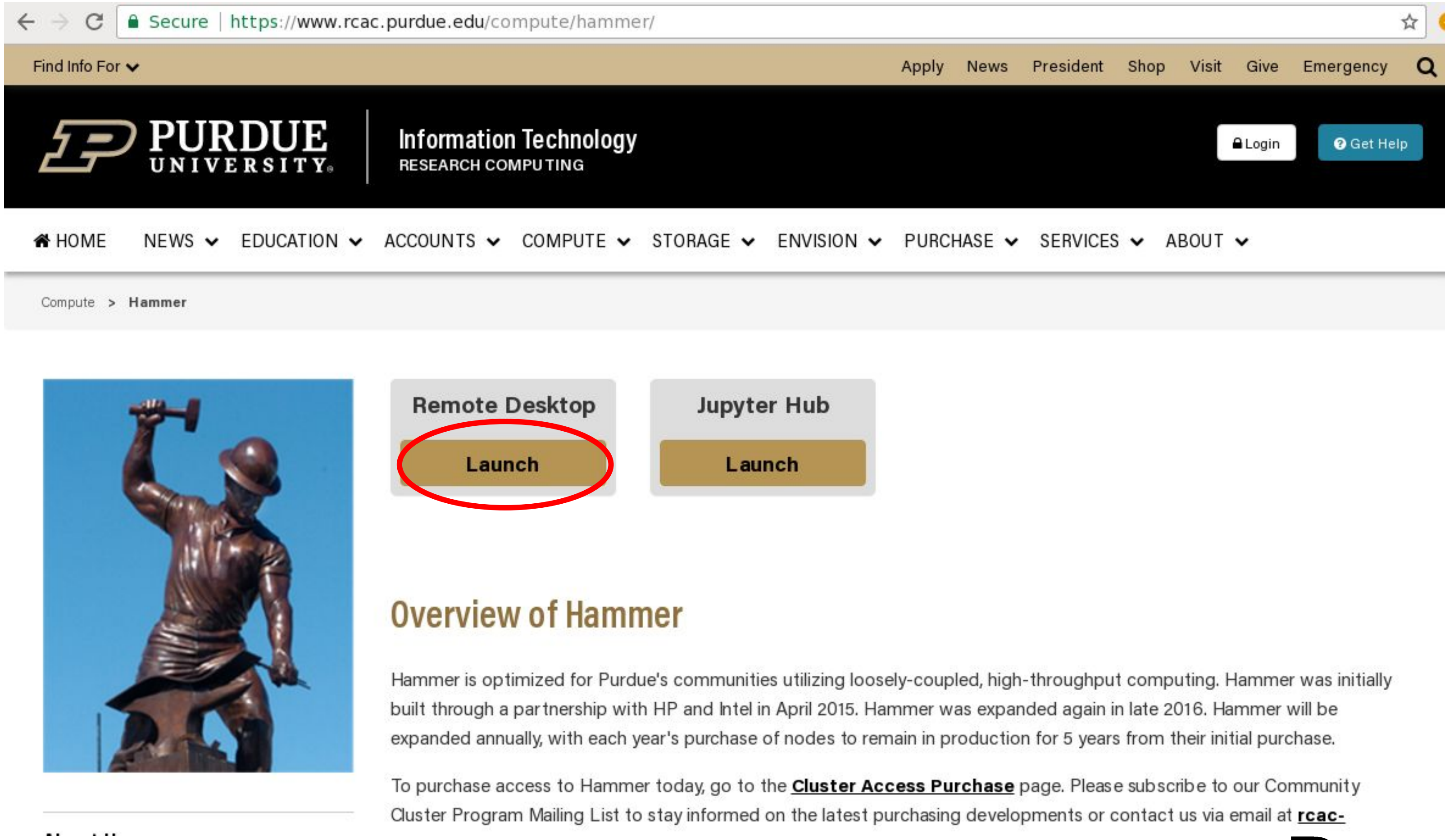
Interactive work

- **SSH to Log-in nodes:**
  - **CMS cluster:**
    - ssh <username>@hep.rcac.purdue.edu
    - ssh <username>@cms.rcac.purdue.edu
  - **Hammer:**
    - ssh <username>@hammer.rcac.purdue.edu
  - **Other Comunity Clusters:**
    (same as Hammer)
    - ssh <username>@halstead.rcac.purdue.edu
    - ssh <username>@rice.rcac.purdue.edu
    - ssh <username>@brown.rcac.purdue.edu
    - ssh <username>@gilbreth.rcac.purdue.edu

PURDUE
UNIVERSITY

Interactive work

- **Remote desktop on Community Clusters**
  - **https://www.rcac.purdue.edu/compute/hammer/**

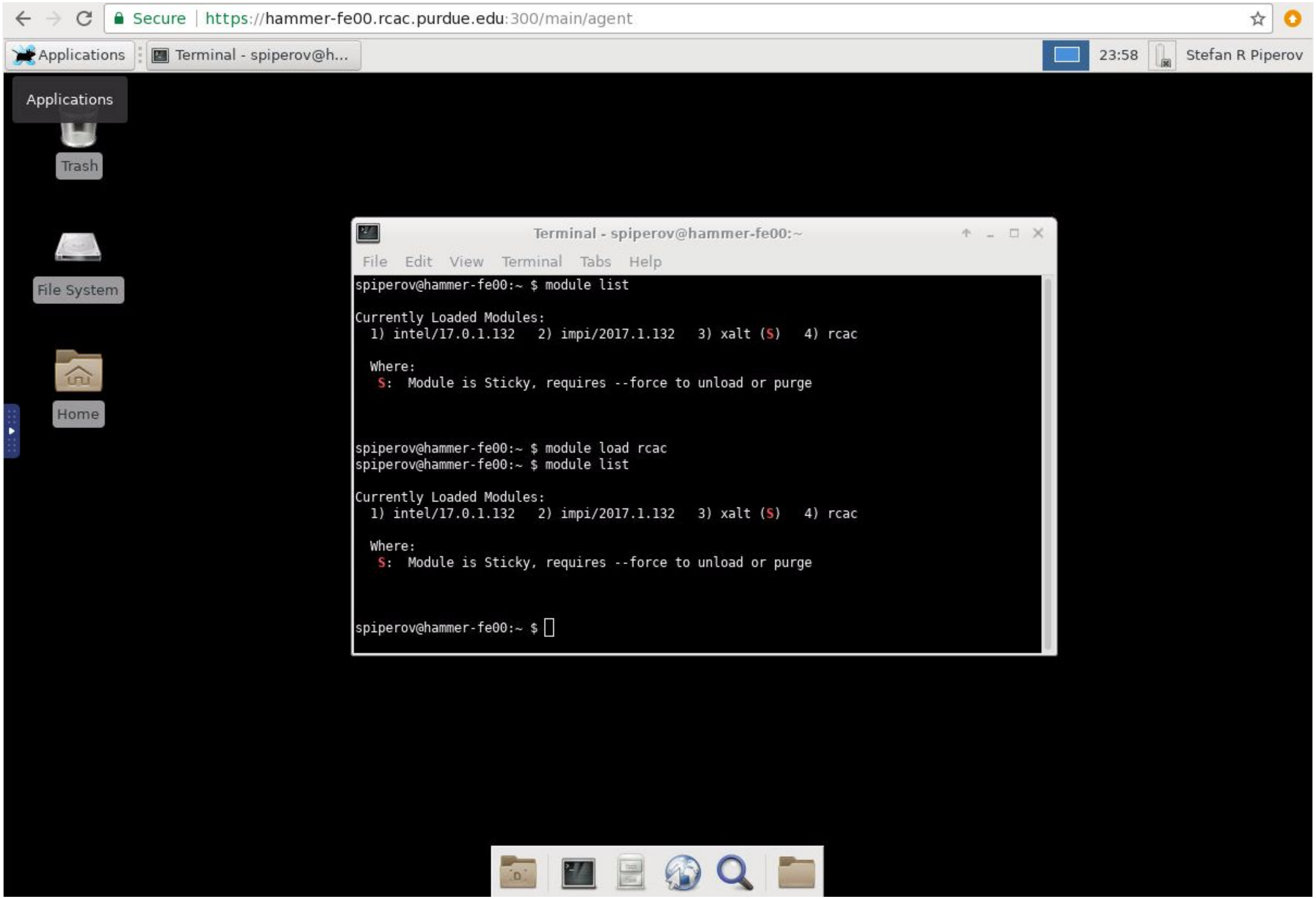

Secure | https://www.rcac.purdue.edu/compute/hammer/
Find Info For
Apply News President Shop Visit Give Emergency
PURDUE UNIVERSITY
Information Technology
RESEARCH COMPUTING
Login
Get Help
HOME NEWS EDUCATION ACCOUNTS COMPUTE STORAGE ENVISION PURCHASE SERVICES ABOUT
Compute > Hammer
Remote Desktop
Launch
Jupyter Hub
Launch

Overview of Hammer

Hammer is optimized for Purdue's communities utilizing loosely-coupled, high-throughput computing. Hammer was initially built through a partnership with HP and Intel in April 2015. Hammer was expanded again in late 2016. Hammer will be expanded annually, with each year's purchase of nodes to remain in production for 5 years from their initial purchase.

To purchase access to Hammer today, go to the **Cluster Access Purchase** page. Please subscribe to our Community Cluster Program Mailing List to stay informed on the latest purchasing developments or contact us via email at **rcac-**

Interactive work


Secure | https://hammer-fe00.rcac.purdue.edu:300/main/agent
Applications
Terminal - spiperov@h...
23:58
Stefan R Piperov
Trash
File System
Home
Terminal - spiperov@hammer-fe00:~
File Edit View Terminal Tabs Help

```
spiperov@hammer-fe00:~ $ module list

Currently Loaded Modules:
  1) intel/17.0.1.132   2) impi/2017.1.132   3) xalt (S)   4) rcac

  Where:
   S:  Module is Sticky, requires --force to unload or purge



spiperov@hammer-fe00:~ $ module load rcac
spiperov@hammer-fe00:~ $ module list

Currently Loaded Modules:
  1) intel/17.0.1.132   2) impi/2017.1.132   3) xalt (S)   4) rcac

  Where:
   S:  Module is Sticky, requires --force to unload or purge



spiperov@hammer-fe00:~ $
```

Best Practices

- Setup CMS environment on the local machine:
  - FrontEnd nodes (cms, hep, hammer, ...)
  - laptop/desktop
  - LXPLUS
- Develop and test analysis code locally
- Test analysis code on a small local dataset
- When convinced that analysis runs correctly locally - Submit multiple batch jobs:
  - either to local clusters via Condor and SLURM
  - or remotely via CRAB and CMS-Connect

- NB. As shared resources, the Front-End (login) nodes deliberately limit the resources available per individual user (20% RAM, 80% CPU)
- Full-size, long, production jobs should never be run there.
- It is much better to start an interactive whole-node SLURM job and have all the memory and CPUs to yourself, than to compete/interfere with everyone else on the FE

Setup CMS environment

- **After logging into a FrontEnd machine**

```
$ export SCRAM_ARCH=slc7_amd64_gcc700
$ source /cvmfs/cms.cern.ch/cmsset_default.sh
$ export CMSSW_GIT_REFERENCE=/cvmfs/cms.cern.ch/cmssw.git.daily
$ mkdir MyAnalysis      #(only if creating it for a first time)
$ cd MyAnalysis
$ cmsrel CMSSW_10_6_4     #(only if setting up for a first time)
$ cd CMSSW_10_6_4/src
$ cmsenv
$ git cms-init
$ echo $CMSSW_BASE
```

Old CMSSW, OS versions

- **Until recently, default OS for CMS was ScientificLinux6 (SL6,RHEL6)**
    - **reaching its ‘end-of-life’**
    - **replaced everywhere with CentOS7 (CC7,RHEL7).**
    - **However, many CMSSW releases still need SL6 environment**

- **For compatibility, we provide it via Singularity containers:**
    - /depot/itap/singularity/cms/cmssw-slc6

After starting the container, you should set up your CMSSW environment in the usual way, but for the older `SCRAM_ARCH=slc6...`

```
$ export SCRAM_ARCH=slc6_amd64_gcc630
$ source /cvmfs/cms.cern.ch/cmsset_default.sh
$ cmsrel CMSSW_9_3_2 (if necessary)
$ cd CMSSW_9_3_2/src
$ cmsenv
$ git cms-init
$ source /cvmfs/cms.cern.ch/crab3/crab.sh
$ voms-proxy-init -voms cms -valid 168:00
..etc.
```

Setup Python environment

- **On Community Clusters - multiple named environments**

```
$ module spider anaconda
    Versions:
    anaconda/5.1.0-py27
    anaconda/5.3.1-py37

$ module load anaconda/5.3.1-py37
$ conda create --name test_coffea python=3.7
$ source activate test_coffea
$ pip install --upgrade coffea
$ conda install -c conda-forge xrootd
$ conda install nb_conda

$ source deactivate

$ module load anaconda/5.1.0-py27
$ conda create --prefix ~/test/ml/uprootenv python=2.7
$ source activate ~/test/ml/uprootenv
$ conda install -c conda-forge uproot
$ conda install -c conda-forge tensorflow keras numpy pandas
```

See more examples of managing packages and exnvironments

PURDUE
UNIVERSITY

Copy small dataset locally

- **To develop/test your analysis code, you only need one (or a few) input files, and not the complete dataset.**
- **You can use these commands to copy such files locally:**

```
$ voms-proxy-init -voms cms -valid 168:00   #(in case you have not done so already)
```

Note: redirector - no need to know the exact location!

```
$ xrdcp
root://cmsxrootd.fnal.gov//store/relval/CMSSW_10_6_4/RelValZMM_13/MINIAODSIM/PUpmx25ns_106X_upgrade201
8_realistic_v9-v1/10000/EC35B5C1-A0A7-574F-8D7A-FCB4C3FBECBE.root ./
```

Note: Precise locations needed!

```
$ gfal-copy
gsiftp://cms-gridftp.rcac.purdue.edu//store/relval/CMSSW_10_6_1/RelValZEE_13/MINIAODSIM/PU25ns_106X_mc
2017_realistic_forECAL_v6_HS-v3/20000/83E6A167-6CD2-BF48-8937-AC79791ACC72.root
file:///home/spiperov/DAS2020/CMSSW_10_6_4/src/
```

*But, can do **recursion**!*

```
$ gfal-copy -r
gsiftp://cms-gridftp.rcac.purdue.edu//store/relval/CMSSW_10_6_1/RelValZEE_13/MINIAODSIM/PU25ns_106X_mc
2017_realistic_forECAL_v6_HS-v3  file:///home/spiperov/DAS2020/CMSSW_10_6_4/src/
```

NB. Sometimes the OSG tools are conflicting with CMSSW tools, and gfal-copy starts crashing. To fix that, execute:

```
$ source /cvmfs/oasis.opensciencegrid.org/osg-software/osg-wn-client/3.5/current/el7-x86_64/setup.sh
$ source /cvmfs/oasis.opensciencegrid.org/osg-software/osg-wn-client/3.4/current/el6-x86_64/setup.sh
```

Working with Datasets

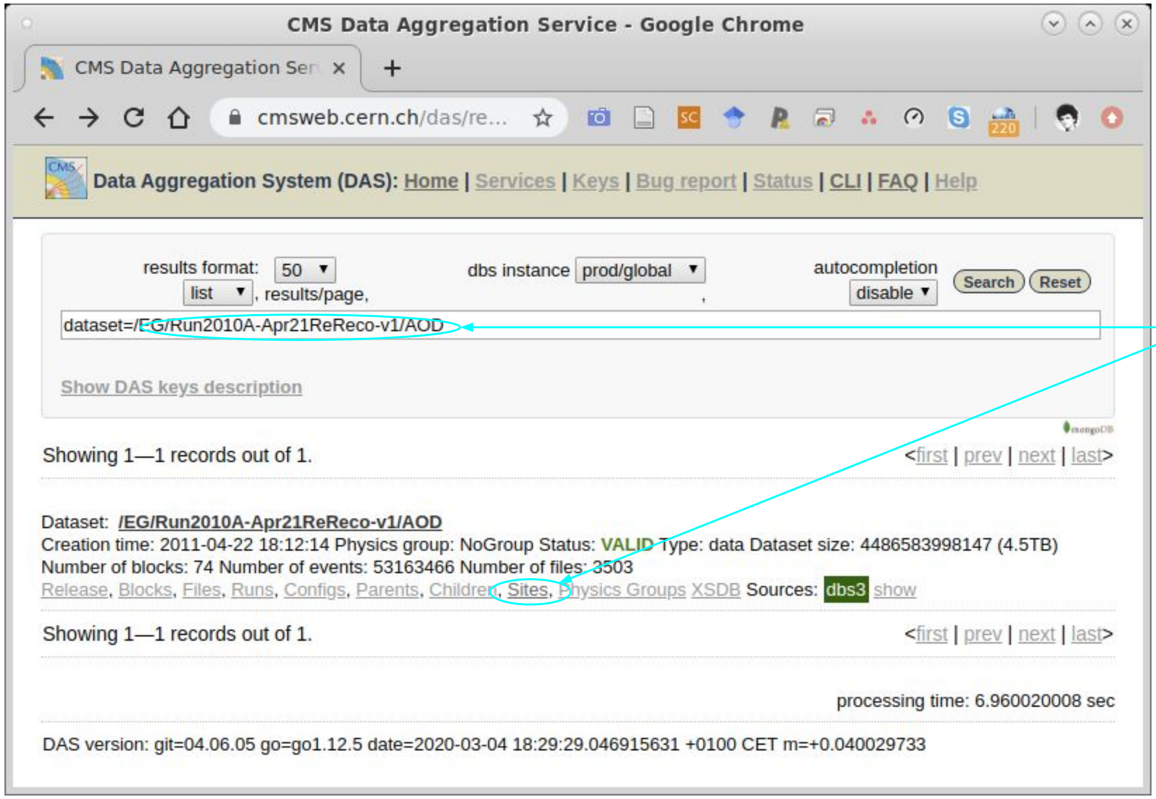
- Main navigational tool - Data Aggregation System (DAS) at CERN (https://cmsweb.cern.ch/das/)
    - Universal and (somewhat) intuitive
    - Slow!
- Command-line tool - dasgoclient
    - Much faster
    - Very flexible, when combined with other UNIX-shell tools

```
$ voms-proxy-init -voms cms -valid 168:00 -rfc
$ /cvmfs/cms.cern.ch/common/dasgoclient -examples
$ /cvmfs/cms.cern.ch/common/dasgoclient -query="dataset=/EG/Run2010A*/AOD"
```

- python APIs - cmssw_das_client.py,DBS client examples
    - can be integrated in directly in your code

Working with Datasets

- Do we *have* this dataset at Purdue?



CMS Data Aggregation Service - Google Chrome

cmsweb.cern.ch/das/re...

Data Aggregation System (DAS): Home | Services | Keys | Bug report | Status | CLI | FAQ | Help

results format: 50, list, results/page, dbs instance prod/global, autocompletion disable, Search, Reset

dataset=/EG/Run2010A-Apr21ReReco-v1/AOD

Show DAS keys description

Showing 1—1 records out of 1. <first | prev | next | last>

Dataset: /EG/Run2010A-Apr21ReReco-v1/AOD
Creation time: 2011-04-22 18:12:14 Physics group: NoGroup Status: VALID Type: data Dataset size: 4486583998147 (4.5TB)
Number of blocks: 74 Number of events: 53163466 Number of files: 3503
Release, Blocks, Files, Runs, Configs, Parents, Children, Sites, Physics Groups XSDB Sources: dbs3 show

Showing 1—1 records out of 1. <first | prev | next | last>

processing time: 6.960020008 sec

DAS version: git=04.06.05 go=go1.12.5 date=2020-03-04 18:29:29.046915631 +0100 CET m=+0.040029733

Ask DAS:
- Enter Dataset Name
- Click “Sites”

- Or ask the DAS GO Client:

```
$ dasgoclient -query="site dataset=/EG/Run2010A-Apr21ReReco-v1/AOD"
T1_UK_RAL_Buffer
T1_UK_RAL_MSS
T3_CH_CERN_OpenData
T3_TW_NTU_HEP
```

Working with Datasets

Do we *need* this dataset at Purdue?

Well, it depends:

- **If it exists at another Tier-2, and you only want to run over a few events/files, then - NO. You can access it remotely via XRootD (AAA):**
  - **directly in ROOT:**

```
TFile *f
=TFile::Open("root://cmsxrootd.fnal.gov//store/mc/SAM/GenericTTbar/GEN-SIM-RECO/CMSSW_5_3_1_START53_V5-v1/
0013/CE4D66EB-5AAE-E111-96D6-003048D37524.root");
```

  - **or in CMSSW:**

Just prepend the address of the XRootD redirector

```
process.source = cms.Source("PoolSource",
   fileNames = cms.untracked.vstring('root://cmsxrootd.fnal.gov//store/myfile.root')
                           )
```

- **But if you plan to run your analysis multiple times, on the entire dataset, then - YES, it's better to copy it here at Purdue.**
  - **Create a PhEDEx "Transfer Request" - very easy!**
  - **Copy the files yourself (tedious, but only option for privately produced datasets at other sites, not registered in PhEDEx)**

```
E.g.:
xrdcp root://stormgf3.pi.infn.it:1094//store/user/PrivateProd/... root://xrootd.rcac.purdue.edu//store/user/piperov/
```

PURDUE
UNIVERSITY

Submitting Jobs

- Distributed:
  - CRAB
    - Send CMSSW jobs to many CMS sites
  - CMS-Connect
    - Send Condor jobs to many CMS sites

- Local:
  - Condor
    - Send jobs to our local CMS cluster
  - SLURM
    - Send jobs to Hammer cluster

Storing and Sharing Data

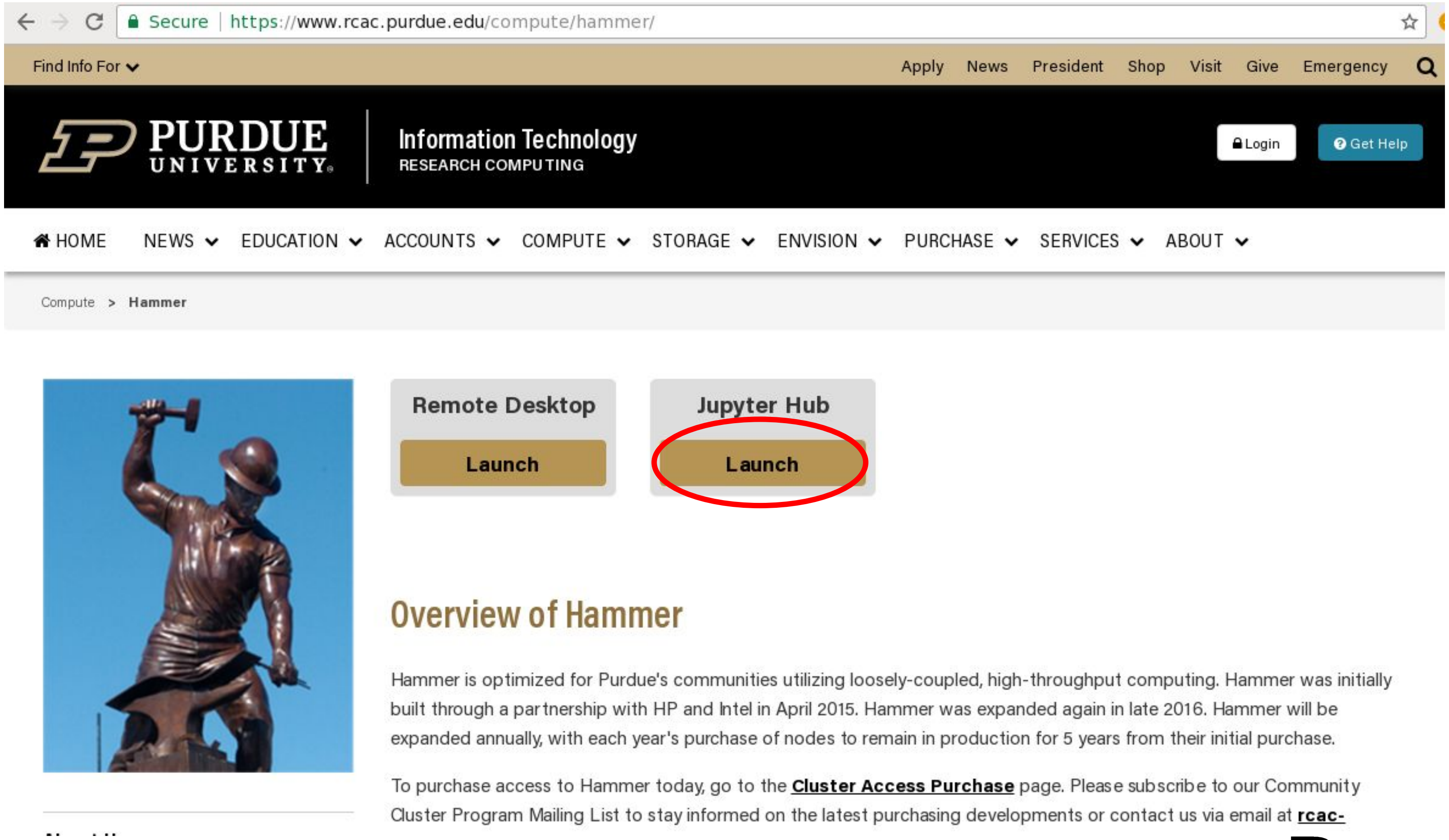
- **Once the jobs are finished, where to put the Ntuples?**
  - **NOT in `/tmp` or `/scratch` - for sure!**
    - **those get cleaned - frequently**
  - **Home directory - probably too small**
  - **Data Depot - for R/W access and sharing with the group**
  - **HDFS - best for long-term storage (R/O) and sharing with the collaboration worldwide**
    - **just point them to your /store/user or /store/group directory**

```
$ xrdcp root://xrootd.rcac.purdue.edu//store/user/piperov/...
```

    - `HDFS is already the default stage-out location for your CRAB jobs`
    - `for local jobs - add a bunch of gfal-copy commands at the end of the job`
  - **Fortress - for archival storage on tape**
    - **HTAR/HSI commands**

Jupyter Notebooks

- **Jupyter Hubs on Community Clusters**
  - **https://www.rcac.purdue.edu/compute/hammer/**



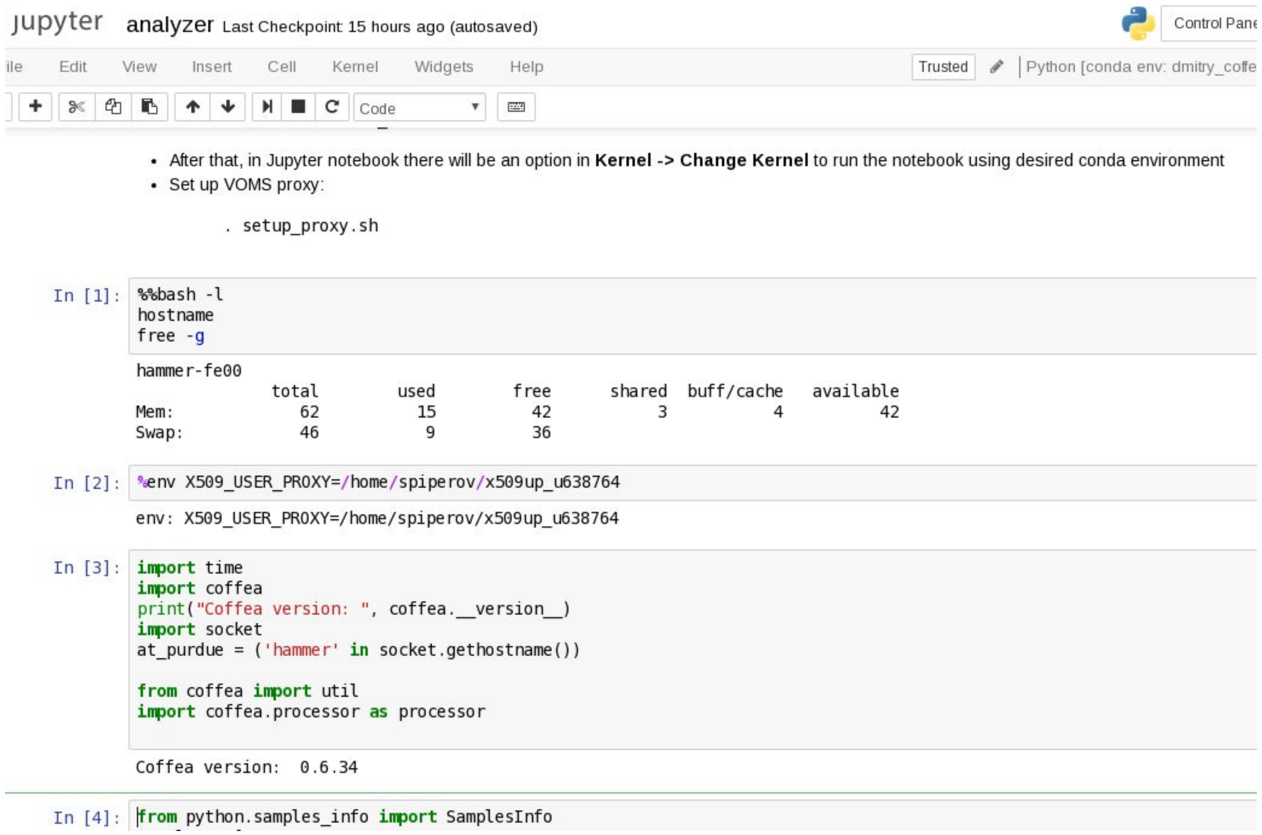
Overview of Hammer

Hammer is optimized for Purdue's communities utilizing loosely-coupled, high-throughput computing. Hammer was initially built through a partnership with HP and Intel in April 2015. Hammer was expanded again in late 2016. Hammer will be expanded annually, with each year's purchase of nodes to remain in production for 5 years from their initial purchase.

To purchase access to Hammer today, go to the **Cluster Access Purchase** page. Please subscribe to our Community Cluster Program Mailing List to stay informed on the latest purchasing developments or contact us via email at **rcac-**

Jupyter Notebooks

- "...web application that allows you to create and share documents that contain live code, equations, visualizations and narrative text."


jupyter
analyzer Last Checkpoint: 15 hours ago (autosaved)
Control Pane
ile Edit View Insert Cell Kernel Widgets Help
Trusted
Python [conda env: dmitry_coffe
Code

- After that, in Jupyter notebook there will be an option in **Kernel -> Change Kernel** to run the notebook using desired conda environment
- Set up VOMS proxy:

```
. setup_proxy.sh
```

```
In [1]: %%bash -l
        hostname
        free -g

hammer-fe00
              total        used        free      shared  buff/cache   available
Mem:             62          15          42           3           4          42
Swap:            46           9          36

In [2]: %env X509_USER_PROXY=/home/spiperov/x509up_u638764

env: X509_USER_PROXY=/home/spiperov/x509up_u638764

In [3]: import time
        import coffea
        print("Coffea version: ", coffea.__version__)
        import socket
        at_purdue = ('hammer' in socket.gethostname())

        from coffea import util
        import coffea.processor as processor

Coffea version:  0.6.34

In [4]: from python.samples_info import SamplesInfo
```

- Nice introductory tutorial and a gallery of interesting examples
- Best experienced *live*! (watch the *COFFEA* demonstration)

PURDUE UNIVERSITY

Speeding things up

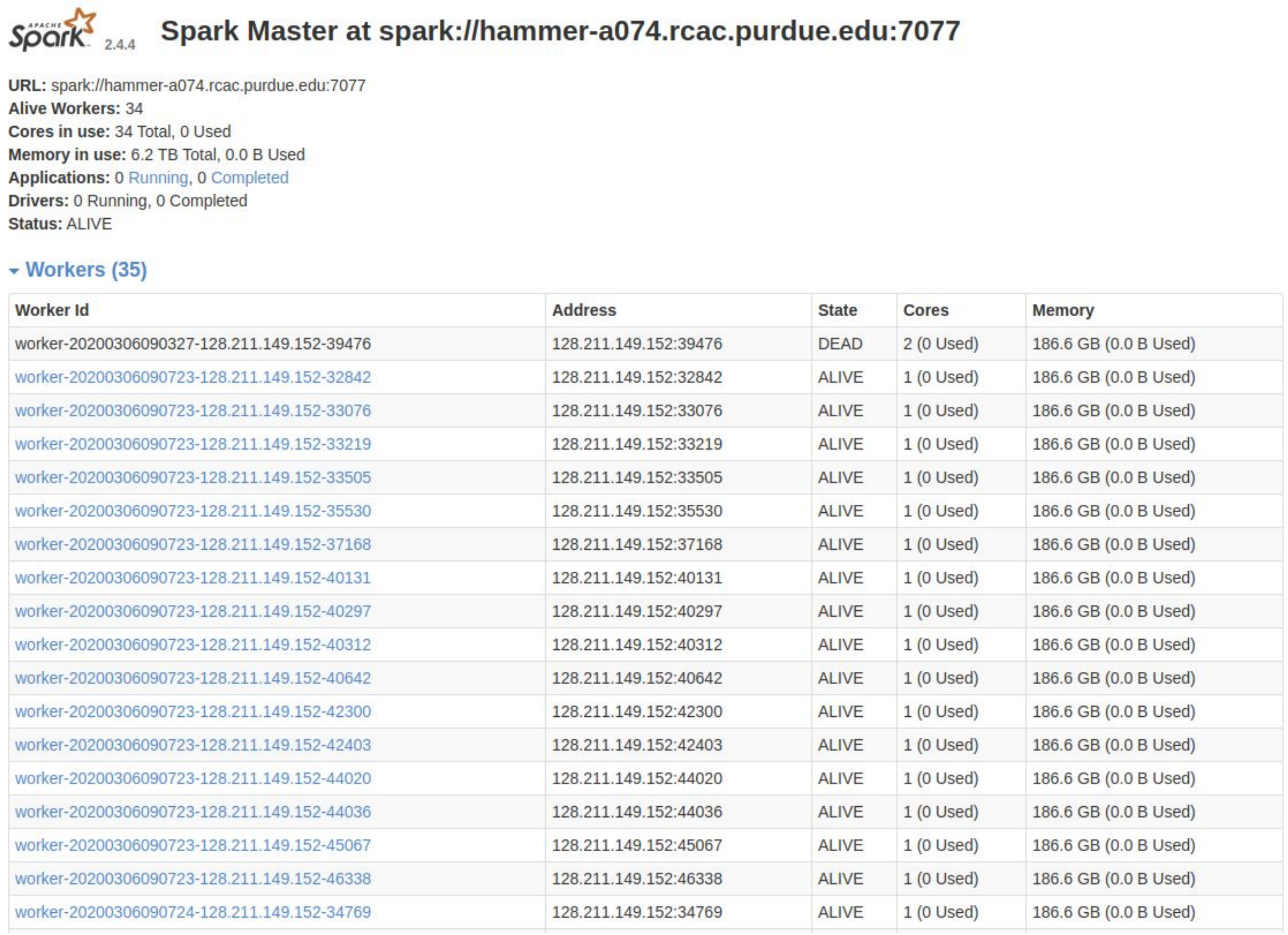
When your analysis gets too big to fit in RAM, or on one CPU or node…

➔ run it in parallel!

- ◆ Apache Spark
  - big, general purpose Big Data framework
  - well integrated with the rest of Apache's ecosystem (e.g. Hadoop)
  - best for Machine Learning
  - requires a cluster manager and a distributed storage system

- ◆ DASK
  - purely Python library, designed for parallel computing - either on the laptop, or on a cluster
  - dynamic task scheduling
  - "Big Data" types of collections for distributed environments
  - smaller, lightweight, runs on your laptop

Speeding things up - Spark



Spark 2.4.4 **Spark Master at spark://hammer-a074.rcac.purdue.edu:7077**

URL: spark://hammer-a074.rcac.purdue.edu:7077
Alive Workers: 34
Cores in use: 34 Total, 0 Used
Memory in use: 6.2 TB Total, 0.0 B Used
Applications: 0 Running, 0 Completed
Drivers: 0 Running, 0 Completed
Status: ALIVE

Workers (35)

Worker Id	Address	State	Cores	Memory
worker-20200306090327-128.211.149.152-39476	128.211.149.152:39476	DEAD	2 (0 Used)	186.6 GB (0.0 B Used)
worker-20200306090723-128.211.149.152-32842	128.211.149.152:32842	ALIVE	1 (0 Used)	186.6 GB (0.0 B Used)
worker-20200306090723-128.211.149.152-33076	128.211.149.152:33076	ALIVE	1 (0 Used)	186.6 GB (0.0 B Used)
worker-20200306090723-128.211.149.152-33219	128.211.149.152:33219	ALIVE	1 (0 Used)	186.6 GB (0.0 B Used)
worker-20200306090723-128.211.149.152-33505	128.211.149.152:33505	ALIVE	1 (0 Used)	186.6 GB (0.0 B Used)
worker-20200306090723-128.211.149.152-35530	128.211.149.152:35530	ALIVE	1 (0 Used)	186.6 GB (0.0 B Used)
worker-20200306090723-128.211.149.152-37168	128.211.149.152:37168	ALIVE	1 (0 Used)	186.6 GB (0.0 B Used)
worker-20200306090723-128.211.149.152-40131	128.211.149.152:40131	ALIVE	1 (0 Used)	186.6 GB (0.0 B Used)
worker-20200306090723-128.211.149.152-40297	128.211.149.152:40297	ALIVE	1 (0 Used)	186.6 GB (0.0 B Used)
worker-20200306090723-128.211.149.152-40312	128.211.149.152:40312	ALIVE	1 (0 Used)	186.6 GB (0.0 B Used)
worker-20200306090723-128.211.149.152-40642	128.211.149.152:40642	ALIVE	1 (0 Used)	186.6 GB (0.0 B Used)
worker-20200306090723-128.211.149.152-42300	128.211.149.152:42300	ALIVE	1 (0 Used)	186.6 GB (0.0 B Used)
worker-20200306090723-128.211.149.152-42403	128.211.149.152:42403	ALIVE	1 (0 Used)	186.6 GB (0.0 B Used)
worker-20200306090723-128.211.149.152-44020	128.211.149.152:44020	ALIVE	1 (0 Used)	186.6 GB (0.0 B Used)
worker-20200306090723-128.211.149.152-44036	128.211.149.152:44036	ALIVE	1 (0 Used)	186.6 GB (0.0 B Used)
worker-20200306090723-128.211.149.152-45067	128.211.149.152:45067	ALIVE	1 (0 Used)	186.6 GB (0.0 B Used)
worker-20200306090723-128.211.149.152-46338	128.211.149.152:46338	ALIVE	1 (0 Used)	186.6 GB (0.0 B Used)
worker-20200306090724-128.211.149.152-34769	128.211.149.152:34769	ALIVE	1 (0 Used)	186.6 GB (0.0 B Used)

```
$ spiperov@hammer-a074:~ $ spark-submit --total-executor-cores 30 --executor-memory 2G pi.py 100
```

User Guides and Contact

- Main page of User's guide: https://www.physics.purdue.edu/Tier2/user-info/
- Community Clusters docs
- For most CMSSW related issues - the CMS WorkBook
- Email us for support:
- cms-support@lists.purdue.edu

PURDUE UNIVERSITY

CMS Cluster in MATH

PURDUE UNIVERSITY