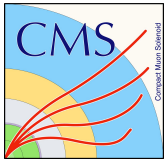
CMS Offline and Computing Week



VC3

CMS experience using VC3

for provisioning spark clusters and deploying T3s on top of campus resources at the user level

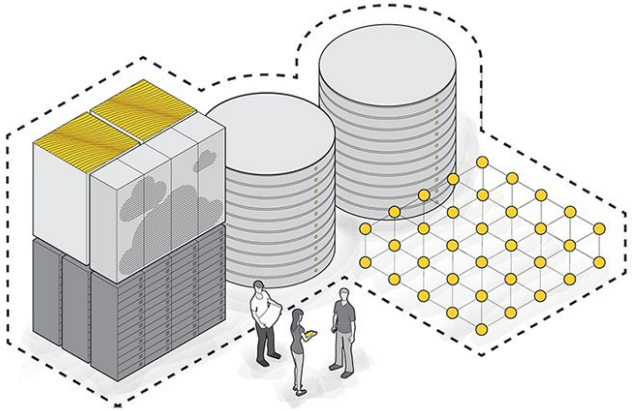
Kenyi Hurtado
University of Notre Dame
khurtado@nd.edu

October 25, 2018

Outline

- What is VC3?
- VC3 CMS use case examples
  - Building Spark clusters on top of Global Pool Resources
  - Deploying a Tier 3 on top of campus resources (with no grid-friendly environment or root access level to the resource)
- Conclusions

What is VC3?



VC3: A platform for provisioning cluster frameworks over heterogeneous resources for collaborative science teams

https://www.virtualclusters.org

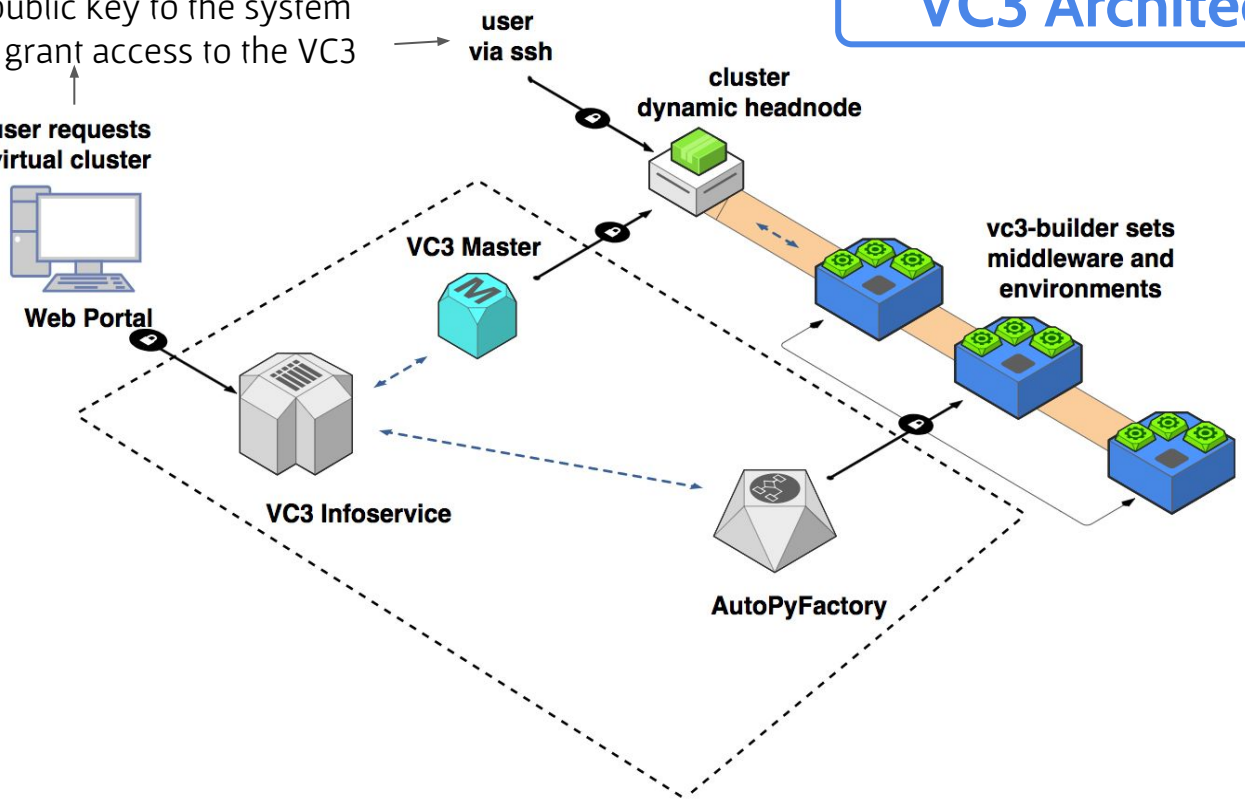
What is VC3?

VC3 Architecture

User adds an SSH public key to the system that will be used to grant access to the VC3 dynamic headnode

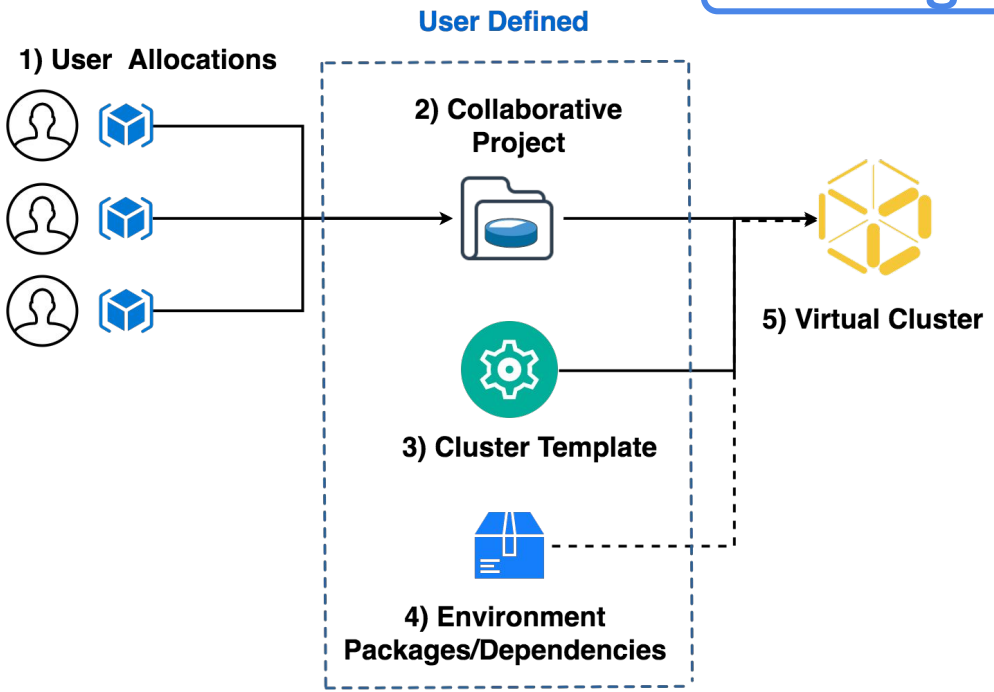
start here

User defines an allocation and is handed a public ssh key to add to his/her submit node resource



What is VC3?

Creating a Virtual Cluster



Use case example 01
Provisioning Spark clusters on top of Global Pool resources

Creating a Spark Cluster – Step 1

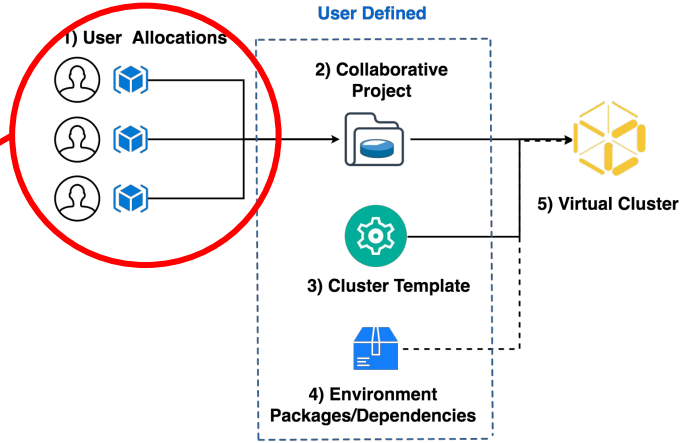
https://www.virtualclusters.org/resource

VC3 News Community Documentation

- Resources
- Allocations
- Projects
- Environments (beta)
- Virtual Clusters
- Monitoring
- Admin

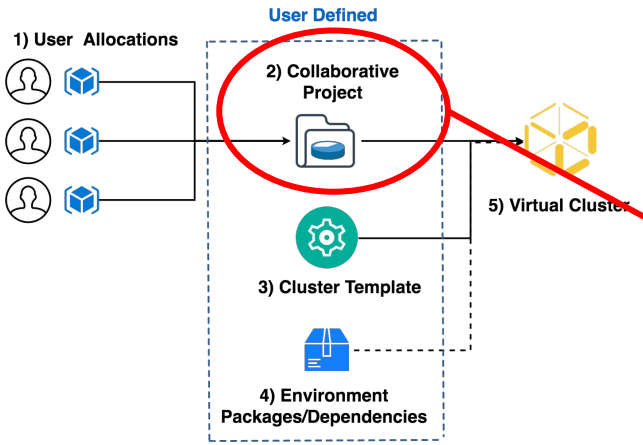
	Computing Center (RCC)	Computing Center (RCC)
Stampede2	Texas Advanced Computing Center (TACC)	Stampede2 Super Computer
CMS Connect	CMS	CMS Connect
CoreOS	University of Chicago	CoreOS/Kubernetes Cluster with HTCondor Overlay
UCT3	University of Chicago - Enrico Fermi Institute	UChicago ATLAS Tier 3
ND CCL	University of Notre Dame Cooperative Computing Lab (CCL)	Notre Dame CCL Job Gateway

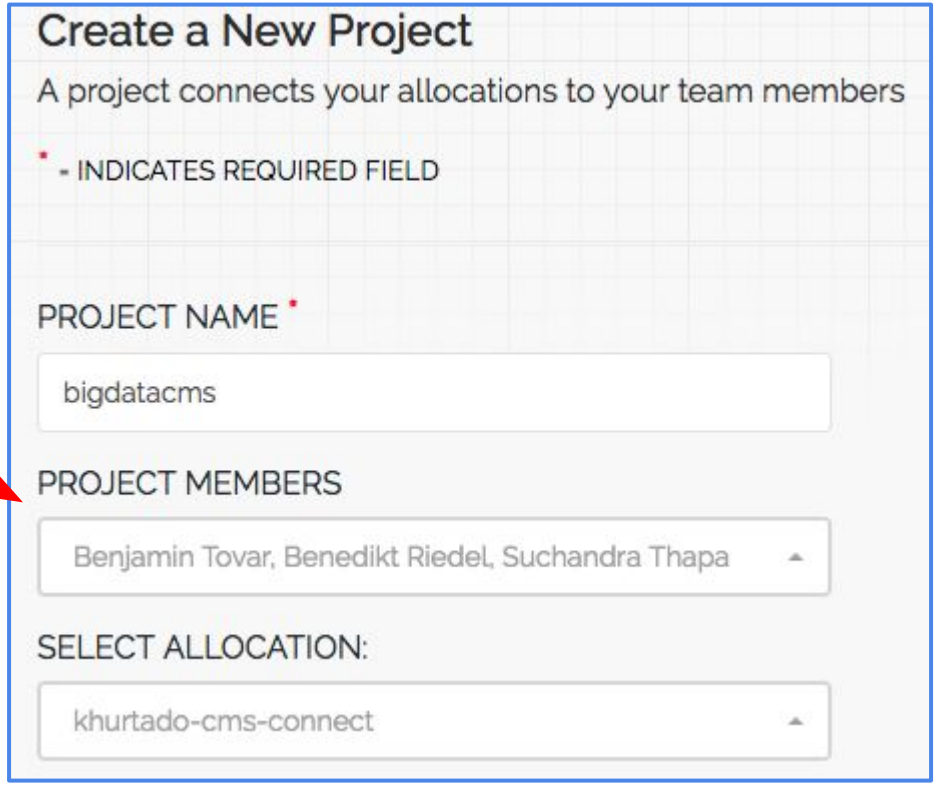
VC3 website:
https://www.virtualclusters.org/



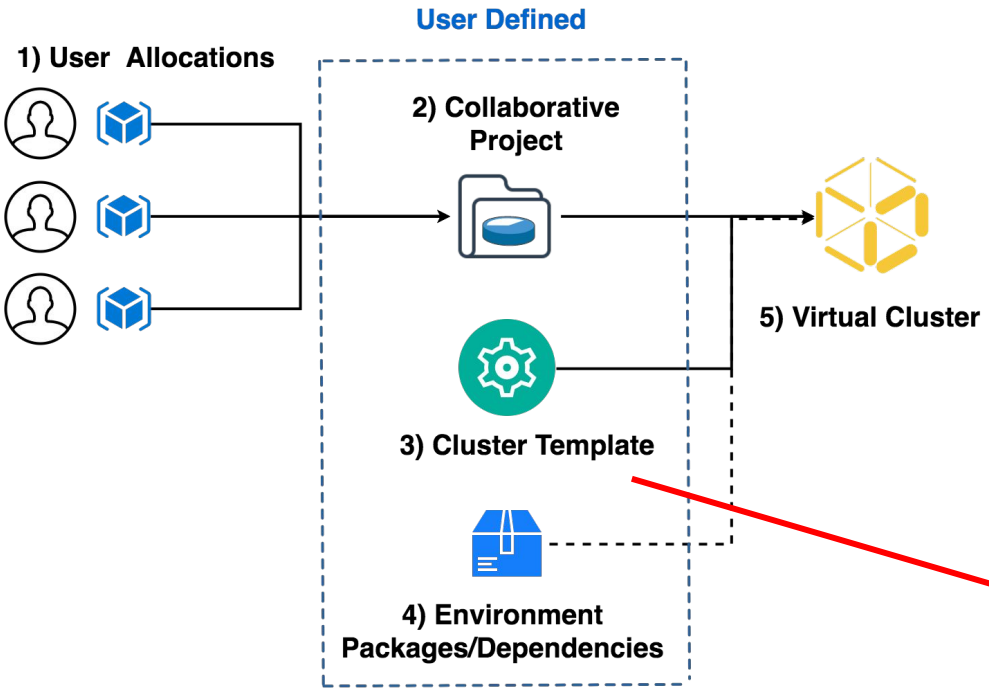
Using CMS Connect to access the CMS Global Pool

Creating a Spark Cluster – Step 2

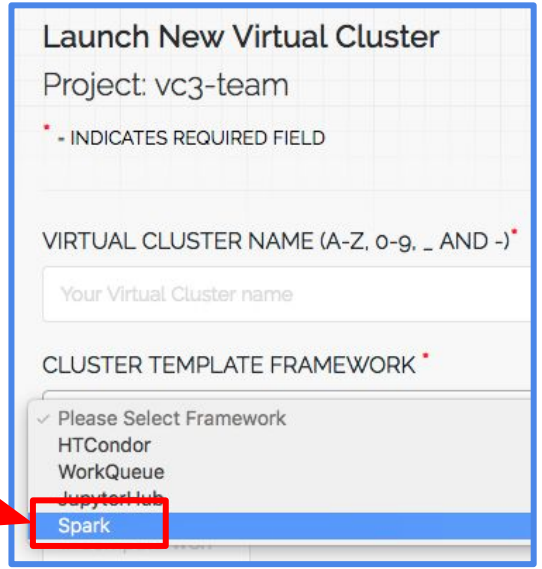




Creating a Spark Cluster – Step 3



Use CMS Connect to access the CMS Global Pool



Creating a Spark Cluster – Step 4

Spark Master at spark://128.135.158.246:7077

URL: spark://128.135.158.246:7077
REST URL: spark://128.135.158.246:6066 *(cluster mode)*
Alive Workers: 10
Cores in use: 20 Total, 0 Used
Memory in use: 74.2 GB Total, 0.0 B Used
Applications: 0 Running, 0 Completed
Drivers: 0 Running, 0 Completed
Status: ALIVE

Workers

Worker Id	Addres
worker-20181024210235-169.228.132.112-39013	169.22
worker-20181024210254-169.228.132.112-33352	169.22
worker-20181024210842-169.228.131.229-37426	169.22
worker-20181024212048-169.228.132.142-39969	169.22
worker-20181024212233-169.228.130.186-30117	169.22
worker-20181024212246-169.228.130.186-5423	169.22
worker-20181024212423-169.228.131.46-42113	169.22
worker-20181024212423-169.228.131.46-43409	169.22

Virtual Cluster: khurtado-spark_ucsd

Terminate Cluster

STATE OF VIRTUAL CLUSTER

Running

All requested compute workers are running.

Owner

Kenyi Anampa

Project

sparkcms

Status

KHURTADO-SPARK_UCSD

Cluster Framework: spark

Requested 10
Running 10
Queued 0
Error 0

Using the Spark Cluster

https://github.com/SiewYan/PadovaBIGDATA/blob/docker_dev/Docker_dev/notebooks/Zpeak_Nanoaod-SPARK.ipynb

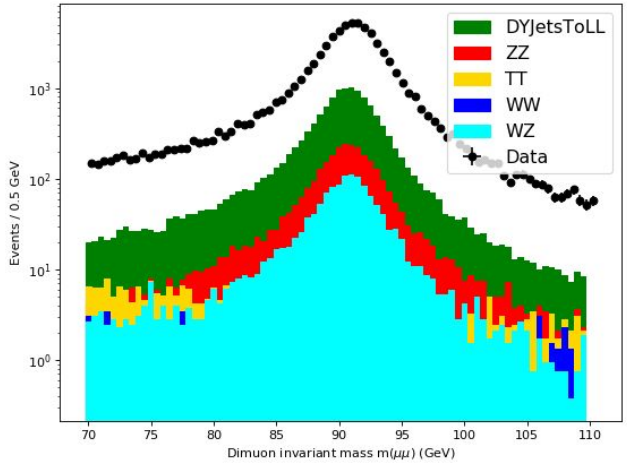
Spark 2.2.1 **Application: Zpeak_Nanoaod-SPARK**

CMS BIG-DATA group example

ID: app-20181024213924-0000
Name: Zpeak_Nanoaod-SPARK
User: khurtado
Cores: Unlimited (20 granted)
Executor Limit: Unlimited (10 granted)
Executor Memory: 7.0 GB
Submit Date: 2018/10/24 21:39:24
State: RUNNING
Application Detail UI

```
vc3-spark-submit --executor-memory 7G --conf spark.sql.caseSensitive=true --conf
spark.serializer=org.apache.spark.serializer.KryoSerializer --packages
org.diana-hep:spark-root_2.11:0.1.16,org.diana-hep:histogrammar-sparksql_2.11:1.0.4
Zpeak_Nanoaod-SPARK.py
```

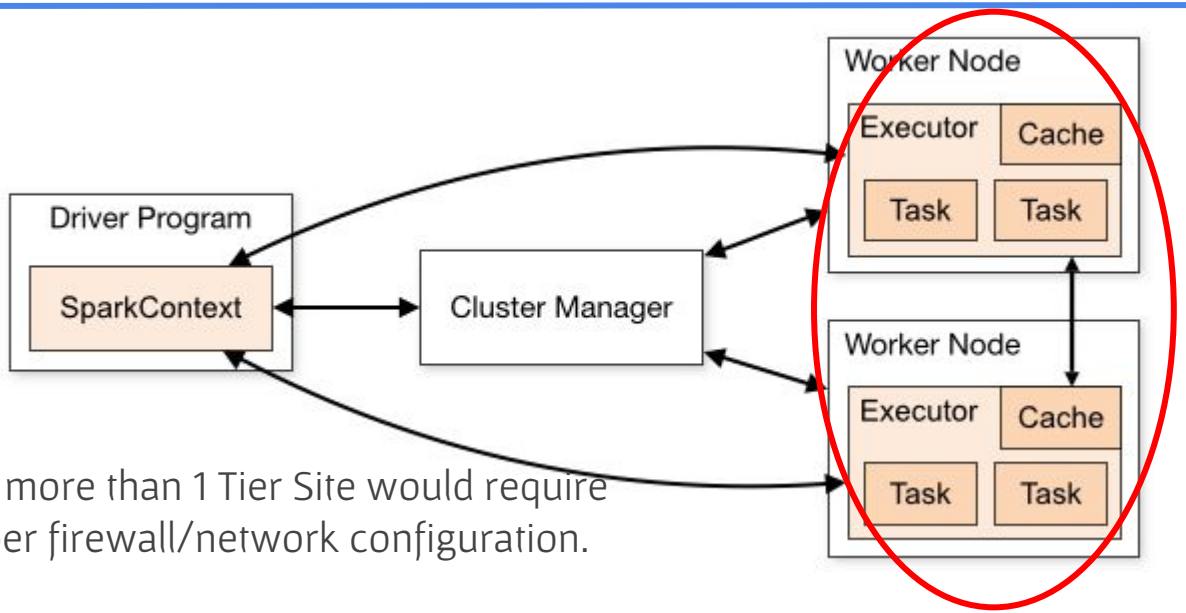
Executor Summa



ExecutorID	Cores	Memory	State	Logs
7	2	7168	RUNNING	stdout stderr
8	2	7168	RUNNING	stdout stderr
0	2	7168	RUNNING	stdout stderr
4	2	7168	RUNNING	stdout stderr
9	2	7168	RUNNING	stdout stderr
3	2	7168	RUNNING	stdout stderr
5	2	7168	RUNNING	stdout stderr
2	2	7168	RUNNING	stdout stderr
6	2	7168	RUNNING	stdout stderr
1	2	7168	RUNNING	stdout stderr

Limitations

1. Spark workers need to communicate between each other for some operations (e.g.: shuffling/repartition)



That means, spark clusters using more than 1 Tier Site would require workers with public IPs and proper firewall/network configuration.

Due to the above, only 1 Tier Site per spark cluster is used in this use case.
This is currently done by exporting an environment variable in login.uscms.org (`CONDOR_DEFAULT_DESIRED_SITES`)

Limitations

2. We use singularity containers in the Global Pool. There is a bug in singularity when using the "--contain" (used in CMS) option that affects running spark workers/slaves.

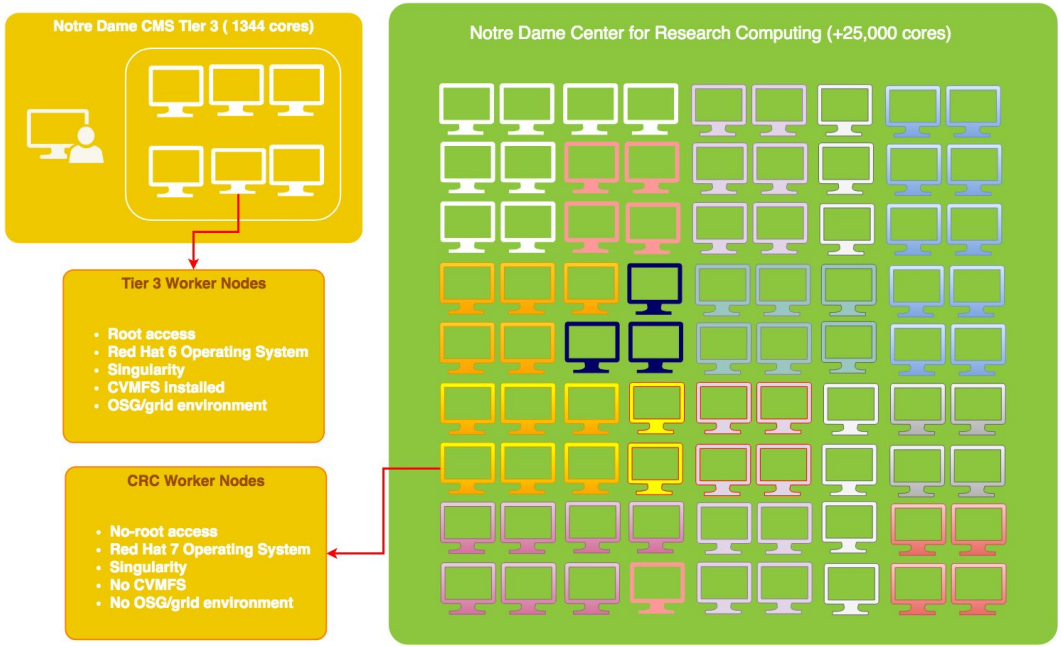
```
09/18/2018 07:17:13 PM [ERROR] There was an error executing job: /srv/x86_64/redhat6.10/spark/v2.2.1/bin/spark-class: line 80: /dev/fd/62: No such file or directory
```

https://github.com/sylabs/singularity/pull/1420

Due to the above, only Sites with Singularity 2.5.2+ will work.
Good news is, 2.6.0 is already available in e.g.: OSG repositories.

Use case example 02
Building a Tier 3 on top of campus resources

The Notre Dame campus cluster



The Notre Dame CMS group operates a Tier 3 with about 1,300 cores for local and grid analysis jobs.

In addition to this, the Center for Research Computing (CRC) provides an opportunistic campus cluster with over 25K cores of computing power researchers have access to, but these resources lack the software components and environment needed by CMS analysis workflows.

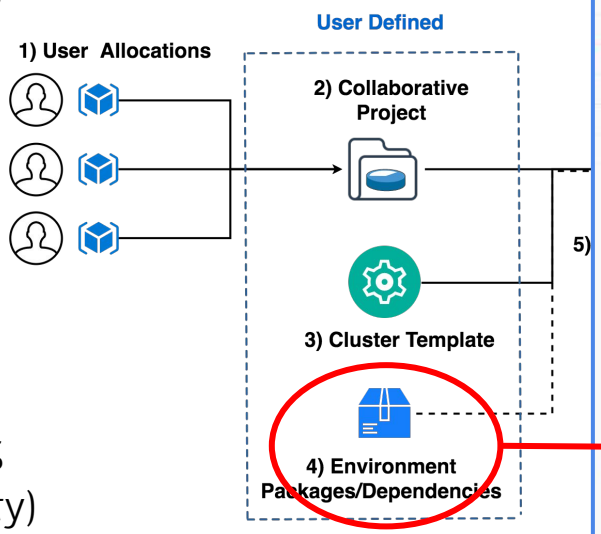
The Notre Dame campus cluster

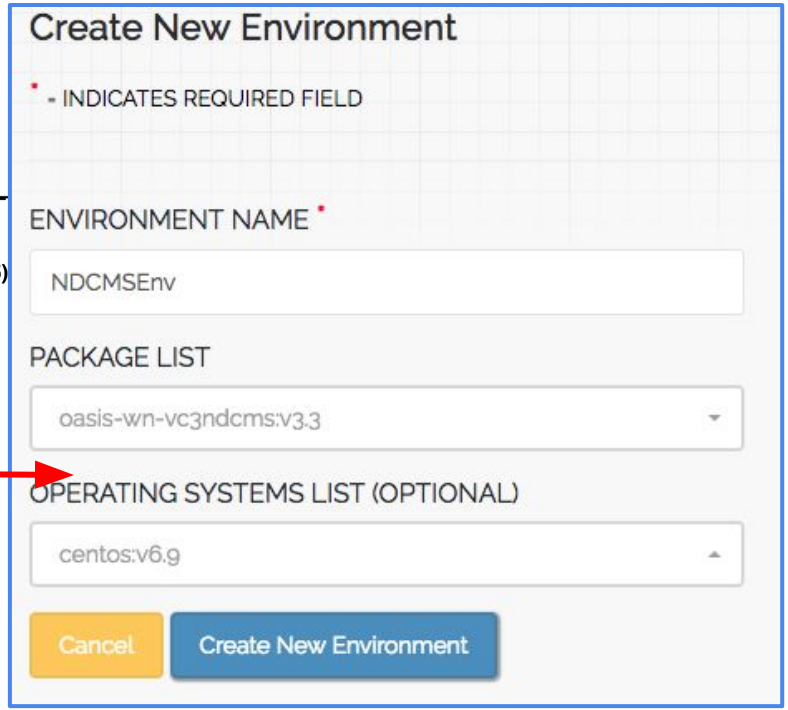
ND CRC cluster characteristics:

- ✔ – HTcondor cluster with +25K cores
- ✔ – Singularity available
- ✘ – **No CVMFS** → CMS glideins use singularity containers located via CVMFS. CMSSW also located here.
- ✘ – **No grid environment (voms, gfal tools, xrootd, etc)** → Can't use VOMS proxies, needed to submit to the Global Pool and access data.
- ✘ – **The ND CMS has no root access to it, just user allocations.** → Can't use yum to install the above using the OSG repositories
- ✘ – **No Compute Element** → Can't install/maintain HTCondor-CE for this resource

Building a Tier 3 on top of campus resources – Step 1

- VC3 allows the provisioning at user-level of:
  - The CERN File System (CVMFS) (via parrot)
  - The OSG grid environment on the worker nodes (via CVMFS)
  - Customized Operating Systems (via singularity)

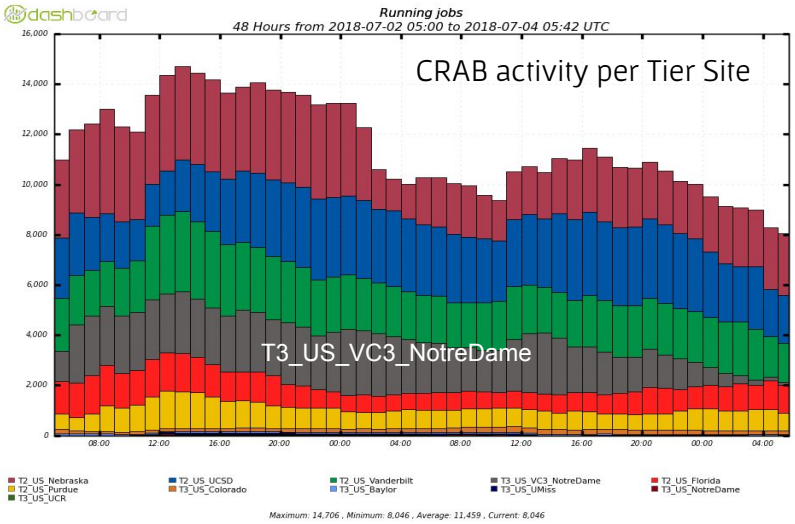


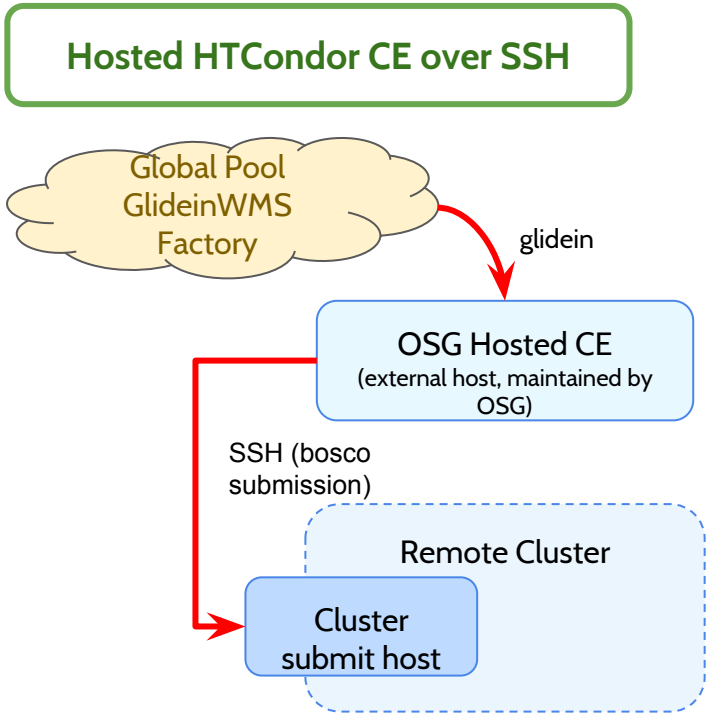


Also, oasis-wn-vc3ndcms links `/cvmfs/cms.cern.ch/SITECONF/local` to: https://gitlab.cern.ch/SITECONF/T3_US_VC3_NotreDame

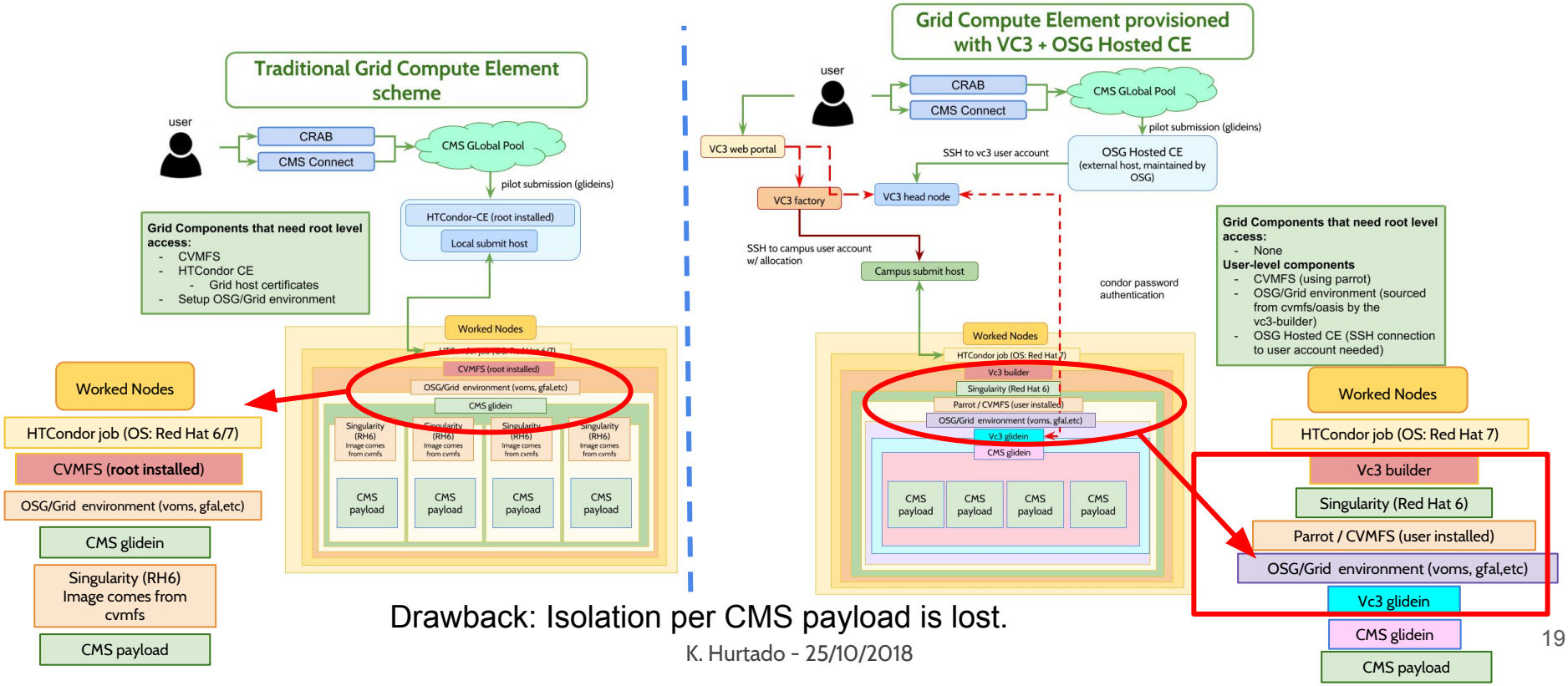
Building a Tier 3 on top of campus resources – Step 2

- The OSG Compute Element (CE) is then integrated with the VC3 submit host, allowing the creation of a CMS Tier 3 using Notre Dame opportunistic campus resources without any root access level.





Traditional vs VC3+OSG Hosted CE T3 components



Drawback: Isolation per CMS payload is lost.

Conclusions

- In these use case examples, we have successfully used VC3 to:
  - Use resources already available in CMS Global Pool, but that can't be used with Spark as-is.
    - ... by provisioning the spark cluster middleware on top of condor.
  - Add and use resources we normally can't use with CRAB
    - ... by instaling and setting up the grid required environment without root access level and plugging the resource into the Global Pool using an OSG Hosted CE

Feel free to contact me at khurtado@nd.edu If you are interested in using VC3 for any of these two use-cases.

VC3 ad slide

Virtual Clusters for Community Computation

https://www.virtualclusters.org
@virtualclusters

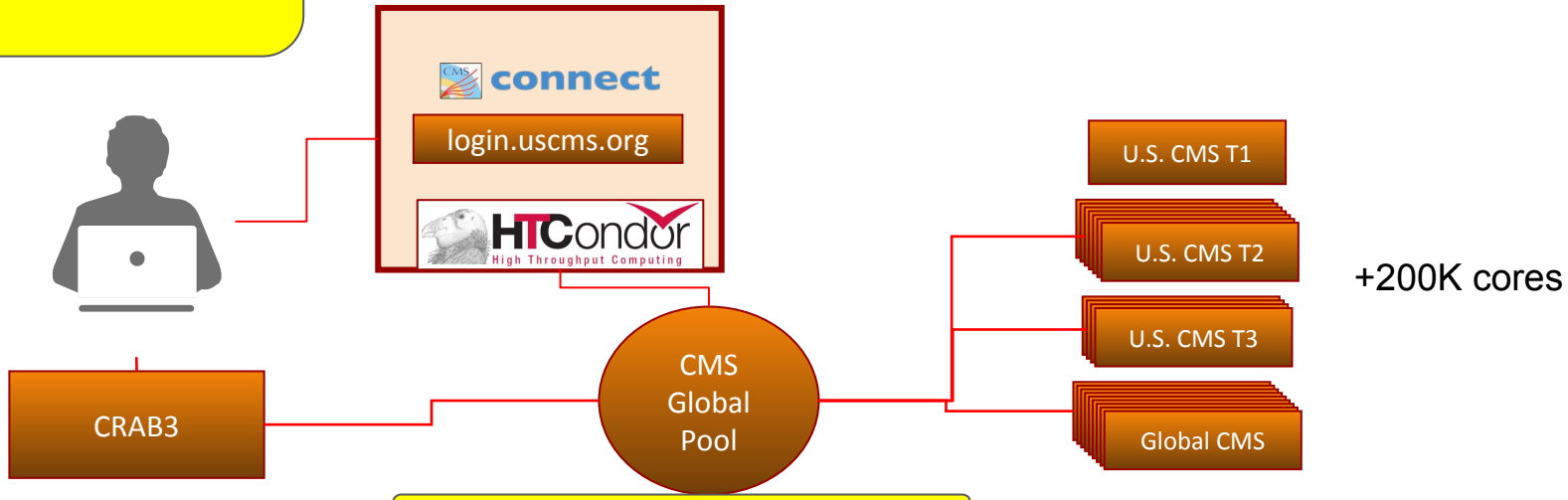
Limited beta signup: http://bit.ly/vc3-signup

Supported by the Department of Energy Office of Advanced Scientific Computing Research and Next Generation Networking Services, Solicitation DE--FOA-0001344 (DDRM), Proposal 0000219942.

Backup slides

CMS Analysis workflows and submission services

CRAB and CMS Connect give analysis users access to the Global Pool



VO CMS grid proxy certificates requires