VC3

DOE Collaborative Projects Meeting
Sponsored by the U.S. Department of Energy
Office of Advanced Scientific Computing Research
September 23–25, 2019

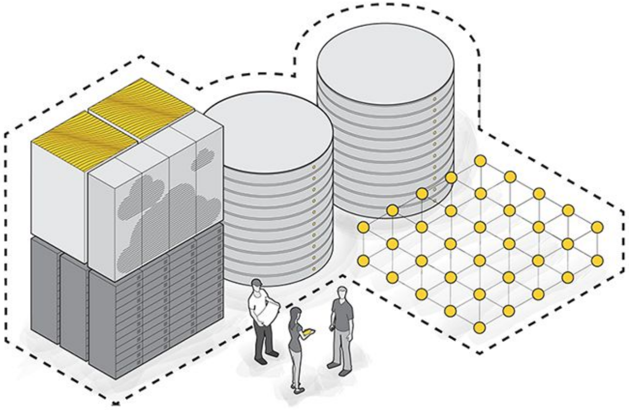
Virtual Clusters for Community Computation

Douglas Thain, University of Notre Dame
Rob Gardner, University of Chicago
John Hover, Brookhaven National Lab

Outline

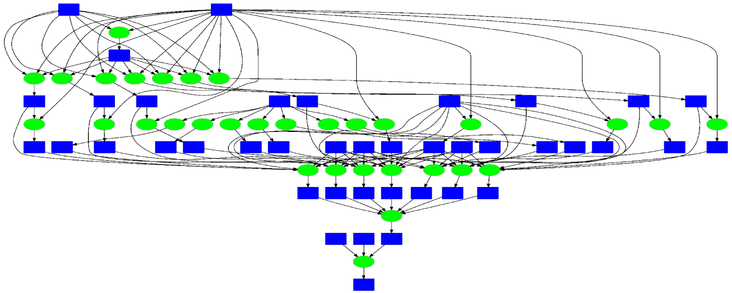
- Intro to the Virtual Cluster Concept
- Self-service Provisioning of Middleware Across Heterogeneous Resources
- Dealing with Software Environments
- Clustering Middleware Examples
- New Applications and Configurations
- Thoughts and Lessons Learned

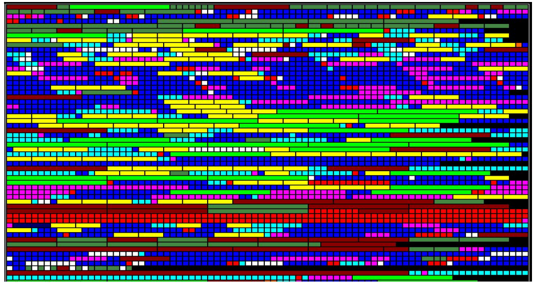
Introduction



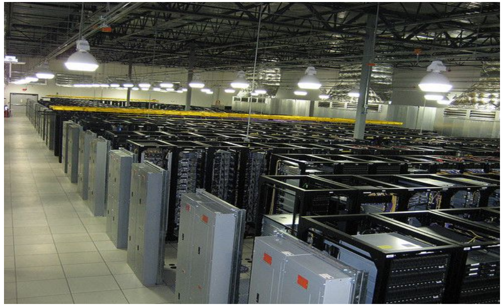
VC3: A platform for provisioning cluster frameworks over heterogeneous resources for collaborative science teams

You have developed a complex workload which runs successfully at one site, perhaps your home university.





Now, you want to migrate and expand that application to national-scale infrastructure.
And allow others to easily access and run similar workloads.

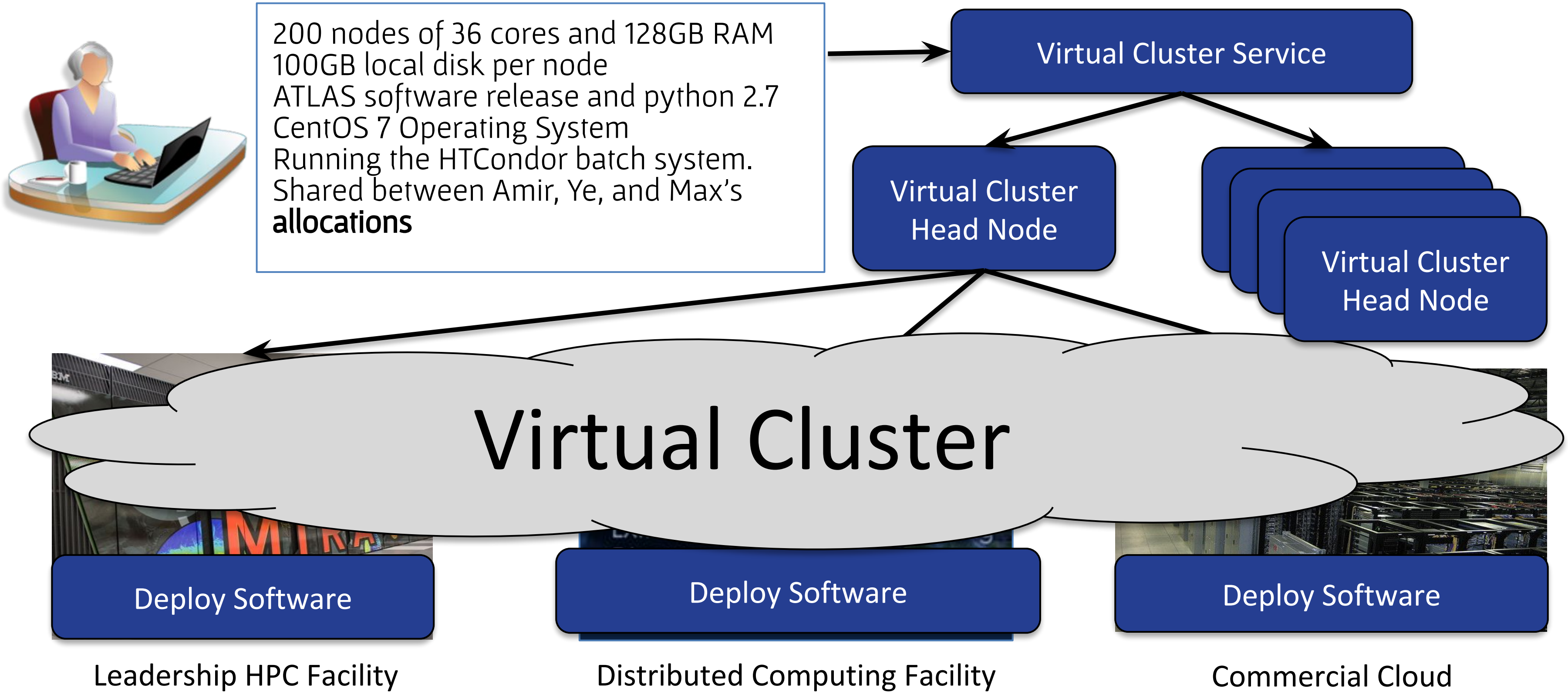


Leadership HPC Facility

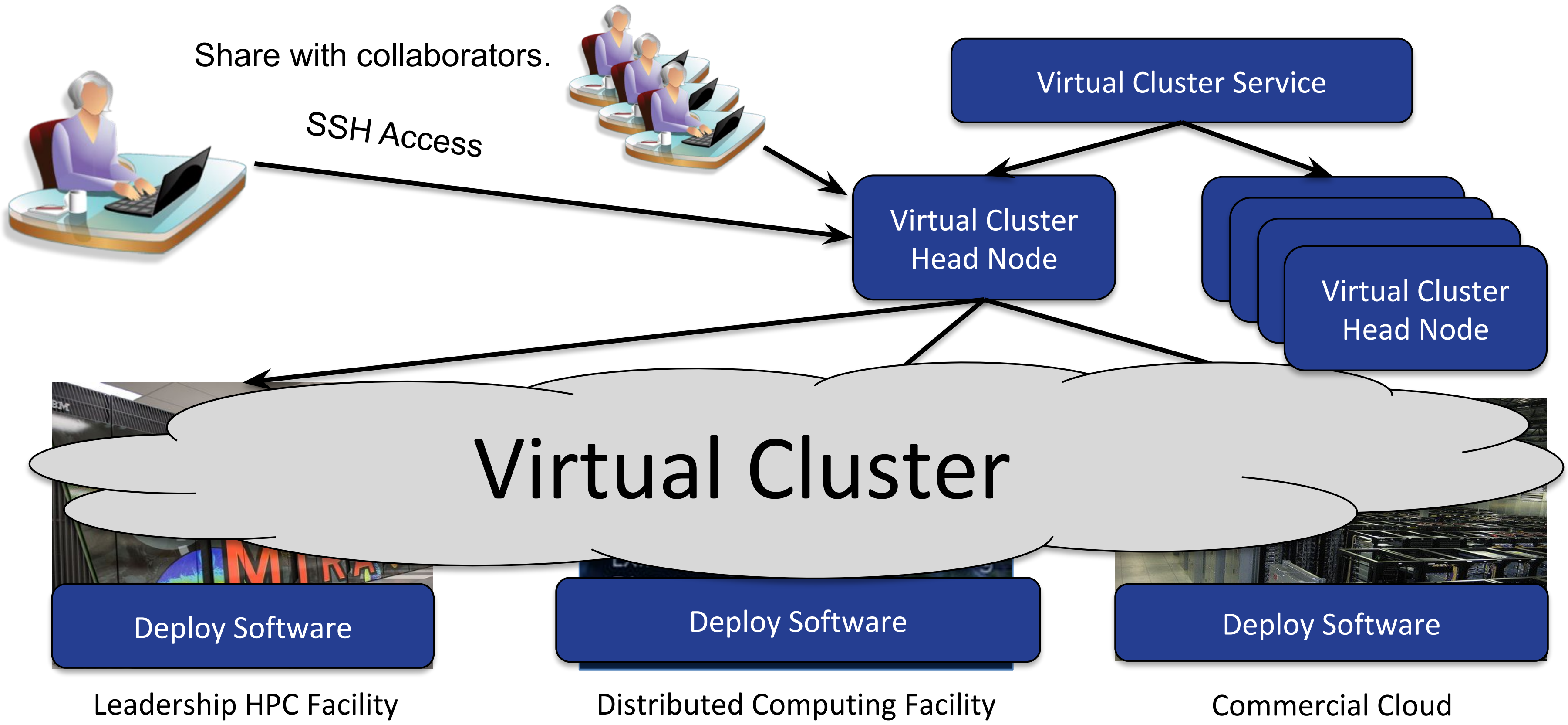
Distributed Computing Facility

Commercial Cloud

Concept: Virtual Cluster

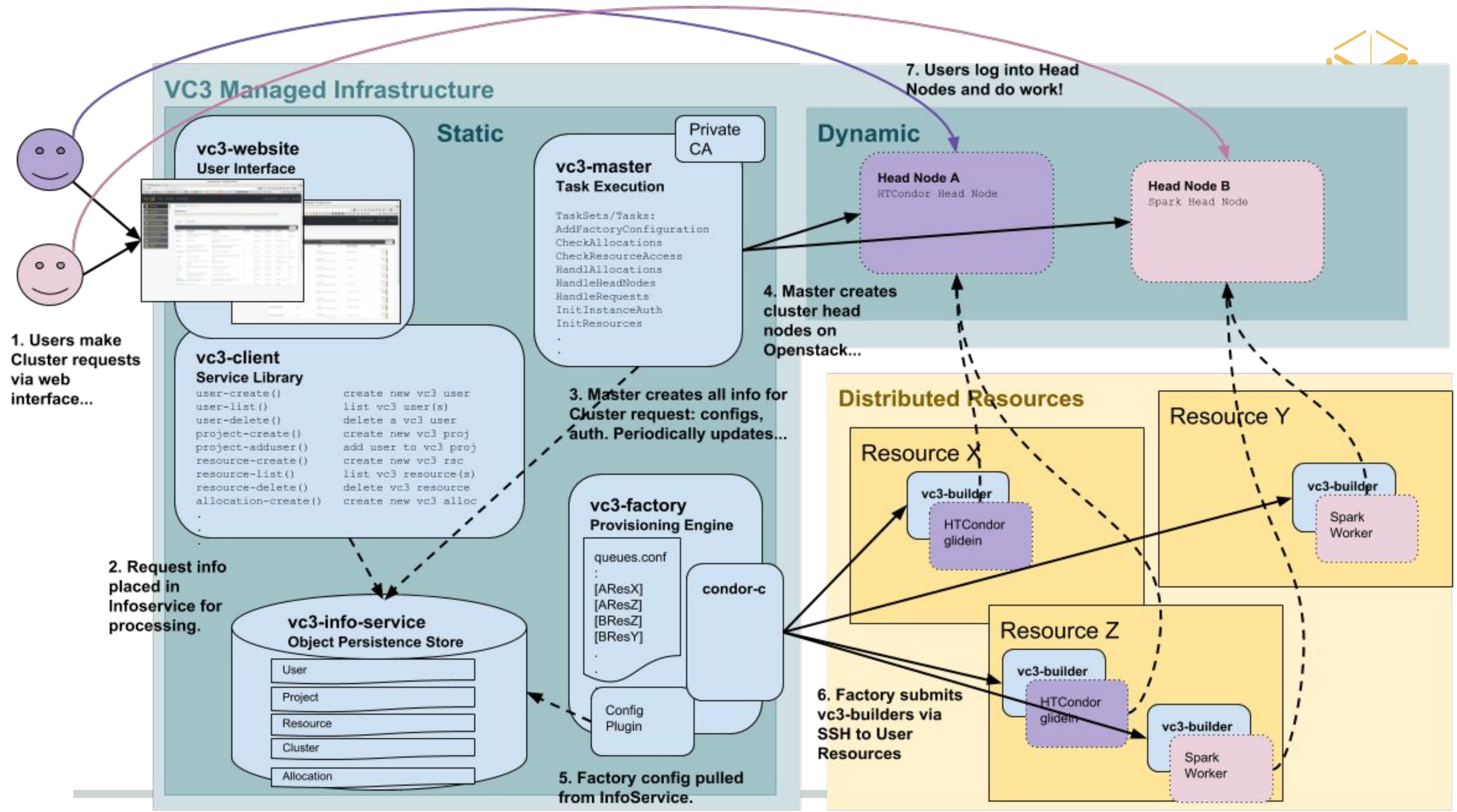


Concept: Virtual Cluster



VC3: Virtual Clusters for Community Computation

- **VC3 is an interactive service for creating/sharing/using virtual clusters.**
- A virtual cluster consists of:
  - 1 x **head node** for interactive access to the cluster. (SSH, Jupyter,..)
  - N x **worker nodes** for executing your workload.
  - **Middleware** to manage the cluster. (HTCondor, Makeflow, Spark, ...)
  - **Application** software to do real work. (BLAST, CMSSW, etc...)
- A virtual cluster is created using:
  - Your standard accounts/credentials on existing facilities.
  - Plain ssh/qsub/srun access on each facility.
  - Container technology (if available) or user-level software builds (otherwise).
  - (No special privileges or admin access required on the facility.)



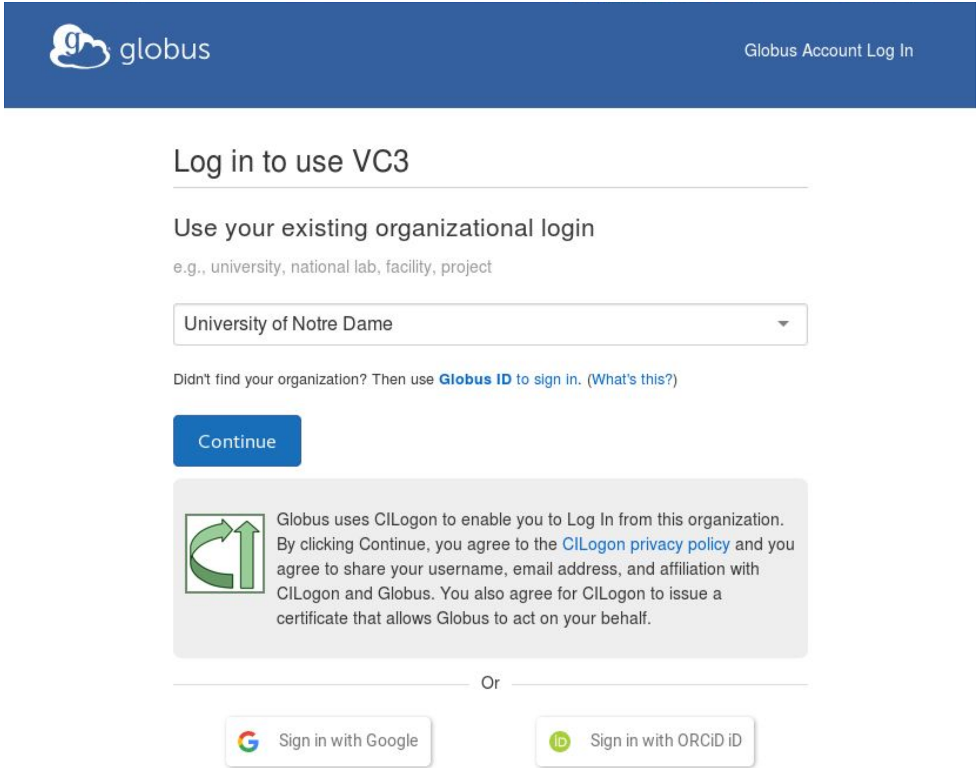
VC3 Managed Infrastructure
Static
Dynamic
vc3-website
User Interface
vc3-master
Task Execution
Private CA
TaskSets/Tasks:
AddFactoryConfiguration
CheckAllocations
CheckResourceAccess
HandlAllocations
HandleHeadNodes
HandleRequests
InitInstanceAuth
InitResources
vc3-client
Service Library
user-create() create new vc3 user
user-list() list vc3 user(s)
user-delete() delete a vc3 user
project-create() create new vc3 proj
project-adduser() add user to vc3 proj
resource-create() create new vc3 rsc
resource-list() list vc3 resource(s)
resource-delete() delete vc3 resource
allocation-create() create new vc3 alloc
vc3-factory
Provisioning Engine
queues.conf
[AResX]
[AResZ]
[BResZ]
[BResY]
condor-c
Config Plugin
vc3-info-service
Object Persistence Store
User
Project
Resource
Cluster
Allocation
Head Node A
HTCondor Head Node
Head Node B
Spark Head Node
Distributed Resources
Resource X
Resource Y
Resource Z
vc3-builder
HTCondor glidein
Spark Worker
1. Users make Cluster requests via web interface...
2. Request info placed in Infoservice for processing.
3. Master creates all info for Cluster request: configs, auth. Periodically updates...
4. Master creates cluster head nodes on Openstack...
5. Factory config pulled from InfoService.
6. Factory submits vc3-builders via SSH to User Resources
7. Users log into Head Nodes and do work!

Self-Service Provisioning

www.virtualclusters.org

login with a vc3 account

Institutional Identity (CI-Logon)



Curated Resources

Resource Profiles

Filter

Name	Organization	Description	Cores	Memory	Storage	Native OS	Features
Cori	National Energy Research Scientific Computing Center (NERSC)	Cori Supercomputer at NERSC	32	4000 MB	10000 MB	suse:v12	Shifter
MWT2	Midwest Tier 2	ATLAS Midwest Tier 2 Center job gateway (UChicago)	4	1000 MB	1000 MB	scientificlinux:v6.9	N/A
Midway	University of Chicago Research Computing Center (RCC)	Midway cluster at the University of Chicago Research Computing Center (RCC)	64	4000 MB	10000 MB	scientificlinux:v6.7	N/A
Stampede 2	Texas Advanced Computing Center (TACC)	Stampede 2 Super Computer	96	2000 MB	10000 MB	centos:v7.4	Singularity
CoreOS	University of Chicago	CoreOS Cluster	4	1000 MB	1000 MB	scientificlinux:v6.9	Singularity
UCT3	University of Chicago	UChicago ATLAS Tier 3	4	1000 MB	1000 MB	scientificlinux:v6.9	N/A
ND CCL	University of Notre Dame Cooperative Computing Lab	ND-CCL login none	4	1000 MB	10000 MB	redhat:v7	Singularity
Bridges	Pittsburgh Supercomputing Center	Bridges Supercomputer at PSC	28	4000 MB	35000 MB	centos:v7.3	Singularity
VC3 Test Pool	VC3	VC3 Test Pool	4	1000 MB	1000 MB	centos:v6.9	N/A
UCLA Hoffman2	University of California, Los Angeles	UCLA Hoffman2	8	1000 MB	10000 MB	centos:v6.9	N/A
OSG Connect	Open Science Grid	Open Science Grid (SL7)	4	1000 MB	1000 MB	Unknown	N/A

Allocations

Step 1: Log Into Resource

In a terminal, type:

`ssh btovar@cclvm05.crc.nd.edu`

Step 2: Access Resource

Enter your password for `cclvm05.crc.nd.edu` for access

Step 3: Add Allocation SSH Public Key to Resource

Once the SSH key is generated below, click `Copy to Clipboard` and paste the following line into your SSH session. You will only need to do this once per allocation.

/GTjhj8YrCyX6UhqG+S3nOxOf+ewxx3RSIMf9lsFZpDNdXwJi1YD1dyRCYy8TwNhBg9GlkCxEKMqfOgo
L6ROpicuUhFY6yT9apKG0x1mPSM
/94ETHxIkBmNK8Ph926fuT+F+QQToSQV0v9o9hWLGiGdNoztW8OUkSFzZ6uZE5zfFpP0xq45a4+FYE
TorljRapgPsjmSjmSB7TeD+qs1ECilwrrg3iJP0RBOEMMeLf7rwgIXXjxtZkBUQ72lkq5ltXTUAyeu0CbGglI
Q7ZHGHrNTyKkSPLI7rXEi7nnz6ofgUJCU3L7hr2VKKy84RcHPSfep64qV3ilOCw1o6SPvu6IwRYeqhfe
A0o
/yKp1IvapyfM7Ptuy+6yWZ7grZlb9AtBolcoBC0lpi964MR8T4D8RKp196onCG5ltXwC4mmPSgffQ0fOl
WJom7TudG+yTWouWikup0ieObZX5w8SKFc0H

Copy to Clipboard

Step 4: Validate Allocation

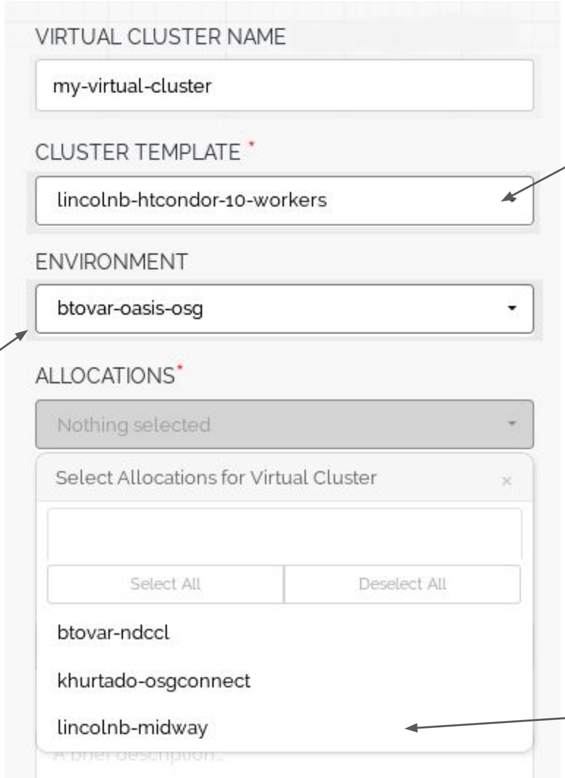
Projects

Project Profiles

Filter

Name	Members	Allocations	Description
vc3-team	**Benjamin Tovar (Owner) - btovar@nd.edu** Lincoln Bryant (UChicago) Jeremy Van (UChicago) Robert Gardner (UChicago) Kenyi Hurtado (University of Notre Dame)	btovar-ndccl khurtado-osgconnect lincolnb-midway	Currently no description
btovar	**Benjamin Tovar (Owner) - btovar@nd.edu** Benjamin Tovar (University of Notre Dame)	btovar-ndccl	Currently no description

Launching a Virtual Cluster



shared cluster definition

workers will have this environment installed

allocations available in this project

Cluster Status

My Virtual Clusters

Filter

Name	State	Cluster Template	Workers	Head Node
my-virtual-cluster	Running All requested compute workers are running.	lincolnb-htcondor-10-workers	Requested: 10 Running: 7 Queued: 3 Error: 0	**128.135.158.187**

Workers from many sites

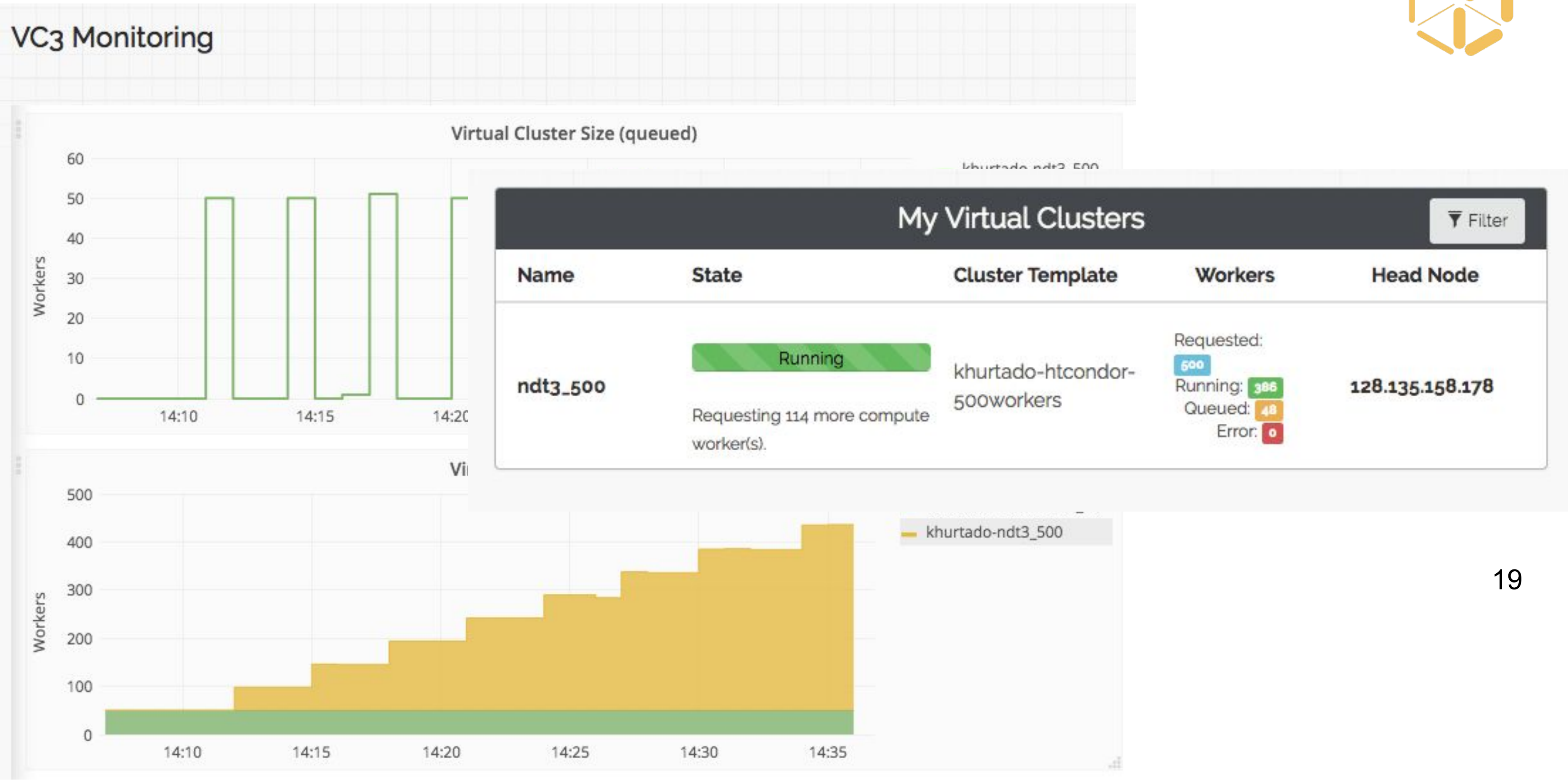
```
[btovar@btovar-my-virtual-cluster ~]$ ip addr | grep 128.135.158.187
    inet 128.135.158.187/25 brd 128.135.158.255 scope global dynamic eth0
[btovar@btovar-my-virtual-cluster ~]$ condor_status
Name                                    OpSys      Arch   State     Activity LoadAv Mem   ActvtyTime

slot1@glidein_21791@camd01.crc.nd.edu   LINUX      X86_64 Unclaimed Idle      5.120 4013  0+00:19:37
slot1@glidein_29106@camd01.crc.nd.edu   LINUX      X86_64 Unclaimed Idle      5.120 4013  0+00:19:37
slot1@glidein_91802@camd05.crc.nd.edu   LINUX      X86_64 Unclaimed Idle      5.260 4013  0+00:19:37
slot1@glidein_39133@iut2-c257.iu.edu    LINUX      X86_64 Unclaimed Idle     34.620 3223  0+00:19:48
slot1@glidein_61297@lnxfarm275.colorado.edu LINUX  X86_64 Unclaimed Idle      6.990 3002  0+00:14:36
slot1@glidein_28373@midway091.rcc.local LINUX      X86_64 Unclaimed Idle      8.170 2013  0+00:19:36
slot1@glidein_71179@midway098.rcc.local LINUX      X86_64 Unclaimed Idle      7.480 2013  0+00:19:36
slot1@glidein_46364@midway260.rcc.local LINUX      X86_64 Unclaimed Idle      8.040 2013  0+00:19:35
slot1@glidein_39282@midway324.rcc.local LINUX      X86_64 Unclaimed Idle      8.750 2013  0+00:19:36
slot1@glidein_39133@uct2-c373.mwt2.org  LINUX      X86_64 Unclaimed Idle     34.080 2415  0+00:19:33

                    Machines Owner Claimed Unclaimed Matched Preempting  Drain

        X86_64/LINUX      10     0       0        10       0          0      0

               Total      10     0       0        10       0          0      0
[btovar@btovar-my-virtual-cluster ~]$
```

Notre Dame

OSG

UChicago

18


VC3 Monitoring
Virtual Cluster Size (queued)
Workers
60
50
40
30
20
10
0
14:10
14:15
14:20
My Virtual Clusters
Filter
Name
State
Cluster Template
Workers
Head Node
ndt3_500
Running
Requesting 114 more compute worker(s).
khurtado-htcondor-500workers
Requested: 500
Running: 386
Queued: 48
Error: 0
128.135.158.178
Workers
500
400
300
200
100
0
14:10
14:15
14:20
14:25
14:30
14:35
khurtado-ndt3_500

19

Dealing with Software Environments

Deploying Software Environments

The **vc3-builder**, a command-line tool for deploying software environments on clusters.

```
vc3-builder
   --require-os centos:7
   --mount /scratch=/data
   --require /cvmfs
   --require python:2.7 -- myapp ...my args...
```

https://github.com/vc3-project/vc3-builder

Basic Use Case: Application Software

```
vc3-builder --require ncbi-blast
..Plan:    ncbi-blast => [, ]
..Try:     ncbi-blast => v2.2.28
....Plan:    perl => [v5
....Try:     perl => v5.
....could not add any
....Try:     perl => v5.
....could not add any
....Try:     perl => v5.
......Plan:    perl-vc3-
......Try:     perl-vc3-r
......Success: perl-vc
....Success: perl v5.
....Plan:    python => [v2.006, ]
```

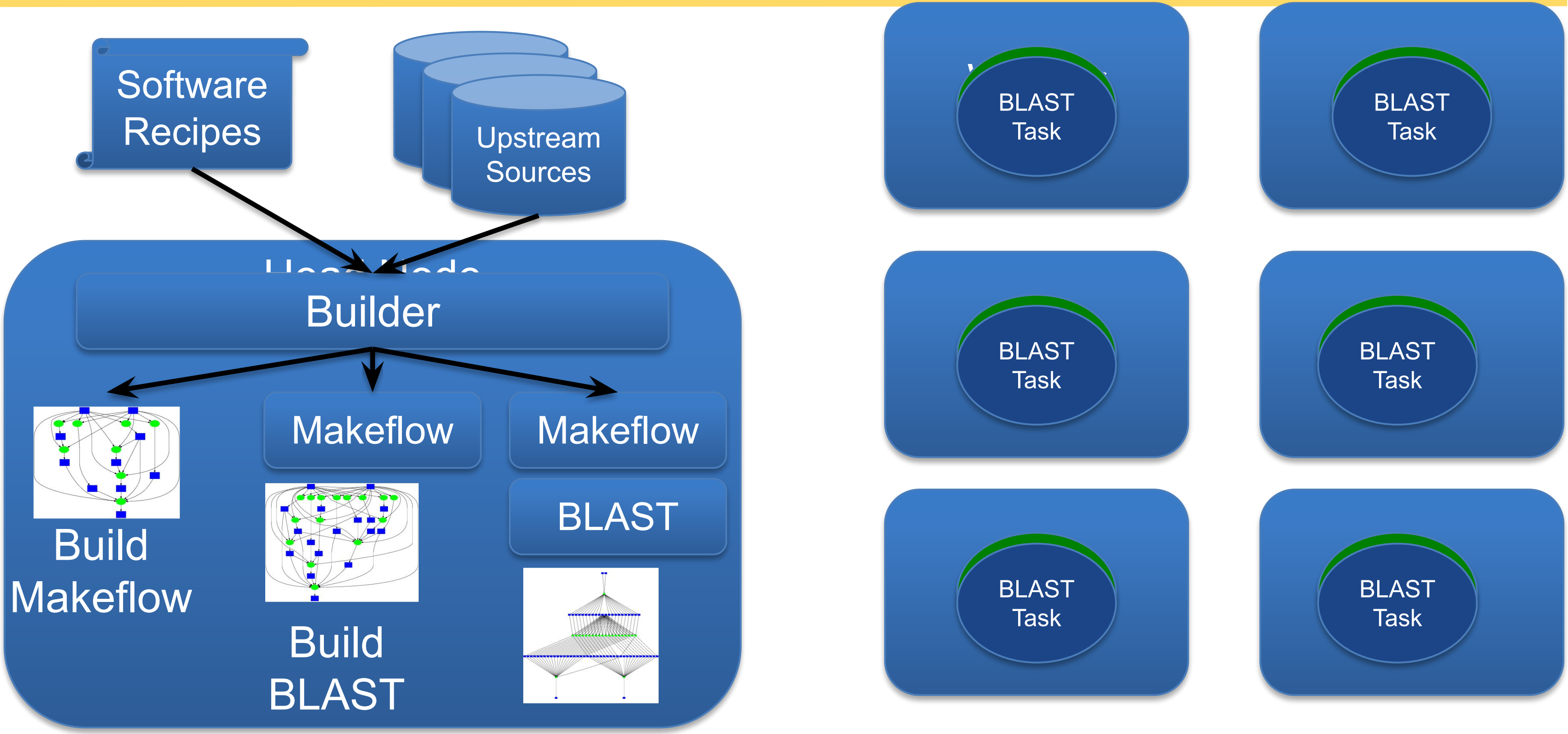
(New Shell with Desired Environment)

```
bash$  which blastx
/tmp/test/vc3-root/x86_64/redhat6/ncbi-blast/v2.2.28/bin
/blastx

bash$ blastx –help
USAGE
  blastx [-h] [-help] [-import_search_strategy filename]
```

Problem: Long Build on Head Node

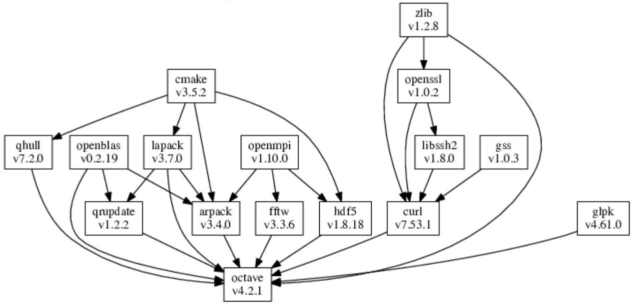
- Many facilities limit the amount of work that can be done on the head node, so as to maintain quality of service for everyone.
- Solution: Move the build jobs out to the cluster nodes. (Which may not have network connections.)
- Idea: Reduce the problem to something we already know how to do: *Workflow!* But how do we bootstrap the workflow software? *With the builder!*

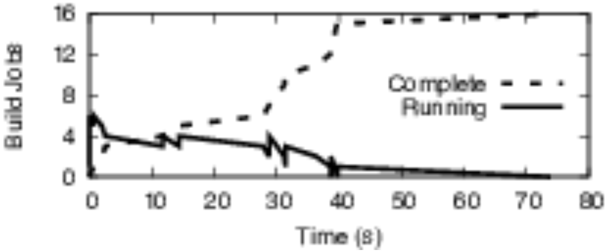
Bootstrapping Workflows and Apps



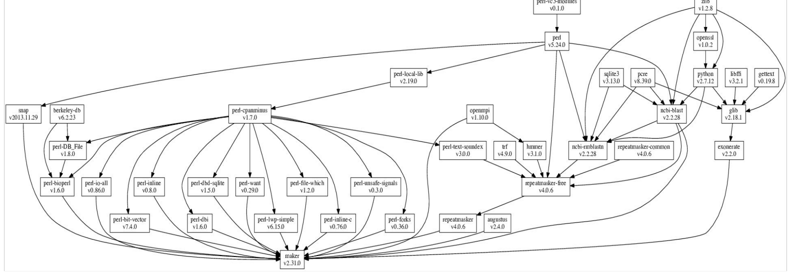
Example Applications

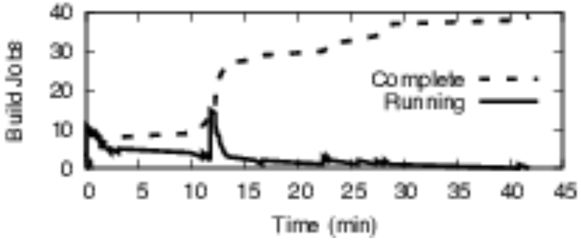
Octave





MAKER





Typically, 2-3x faster overall. But more importantly, filesystem-intensive jobs run on the cluster resources, not on the head node!

Controlling Cluster Size

Dynamic Cluster Resizing

VC3 Supports several mechanisms for setting the number of workers in a given virtual cluster.

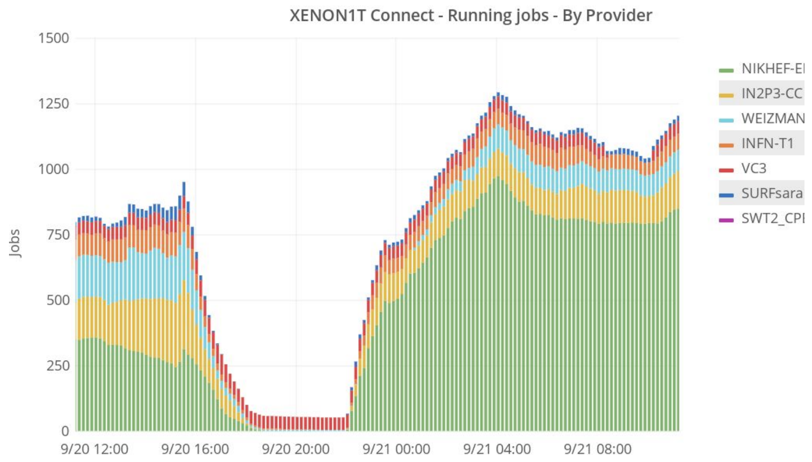
- Set static size for cluster upon creation.
- Manually change cluster size via web interface. The provisioning factory automatically adds or removes worker jobs at resources.
- (For HTCondor cluster) Scale the cluster dynamically based on idle jobs on the head node.

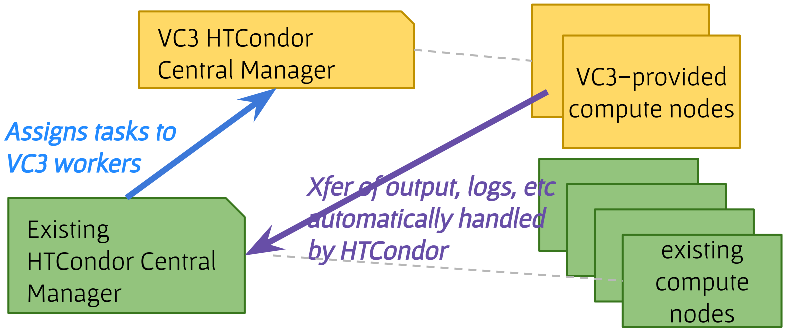
Current Resizing Cases

		Job Removal Setting	
		Peaceful = True	Peaceful = False
Source of desired number of workers	User (via web interface)	**Increasing:** Works as expected. **Decreasing**: Workers will terminate when they have finished current job. By default, the younger workers go away first.	**Decreasing**: Running jobs may be killed. But cluster can be reduced rapidly.
	VC3 (automatic, dynamic, based on user's job pressure on head node.)	**Increasing:** Works as expected. **Decreasing**: Workers will terminate when they have finished current job. By default, the younger workers go away first.	

Scaling Out Production Clusters

- Collaborations who have existing HTCondor pools can extend them by adding more worker nodes via VC3
- Add XSEDE resources, Open Science Grid, and campus HPC clusters
- End-users can transparently use additional resources



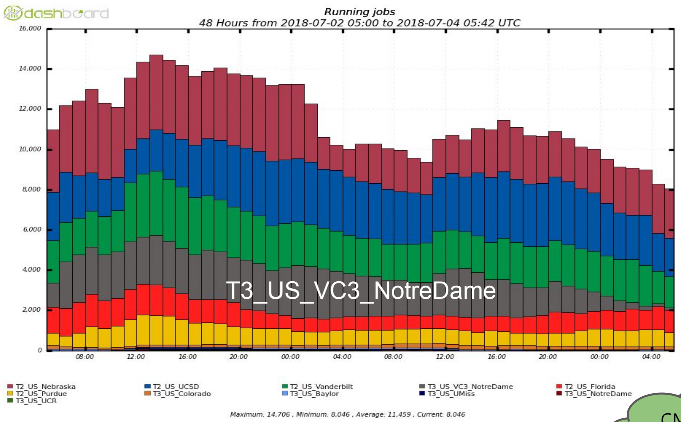


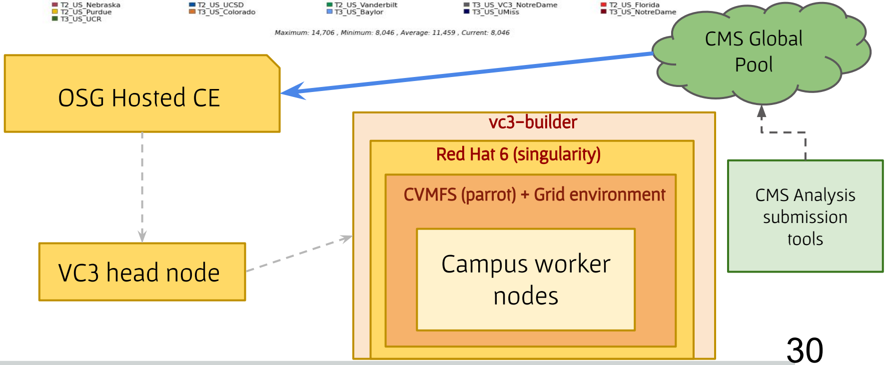
Building Tier 3s on top of campus resources

- Although different computing resources are often available at universities, meeting all requirements to deploy a valid Tier 3 able to run CMS workflows on the grid is not an easy task to achieve without the intervention of a system administrator with root access.
- 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)
- 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.

CMS Analysis activity per Tier Site

Note T3_US_VC3_NotreDame performing at the scale of the Tier 2s!



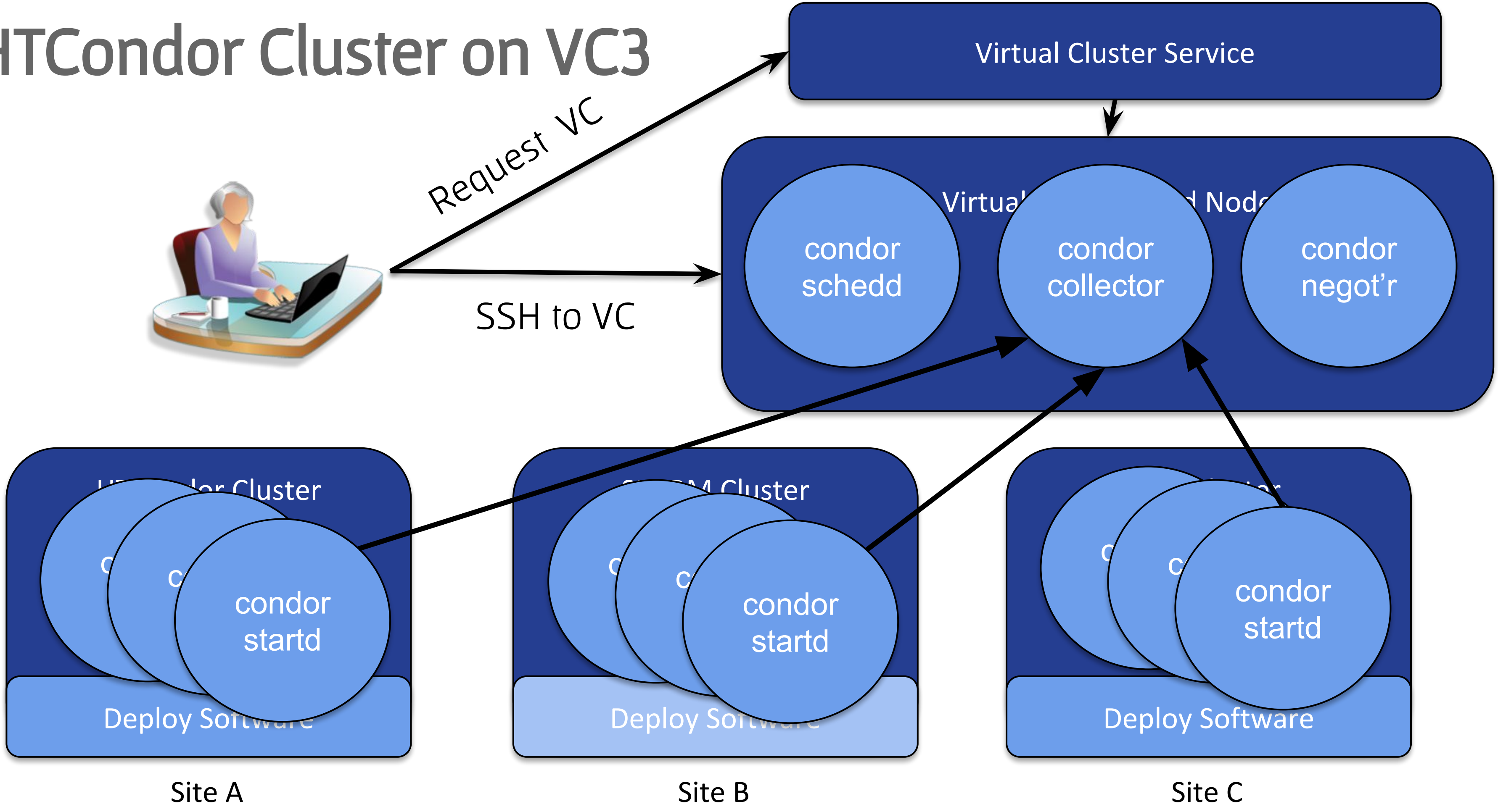


Examples of clustering middleware

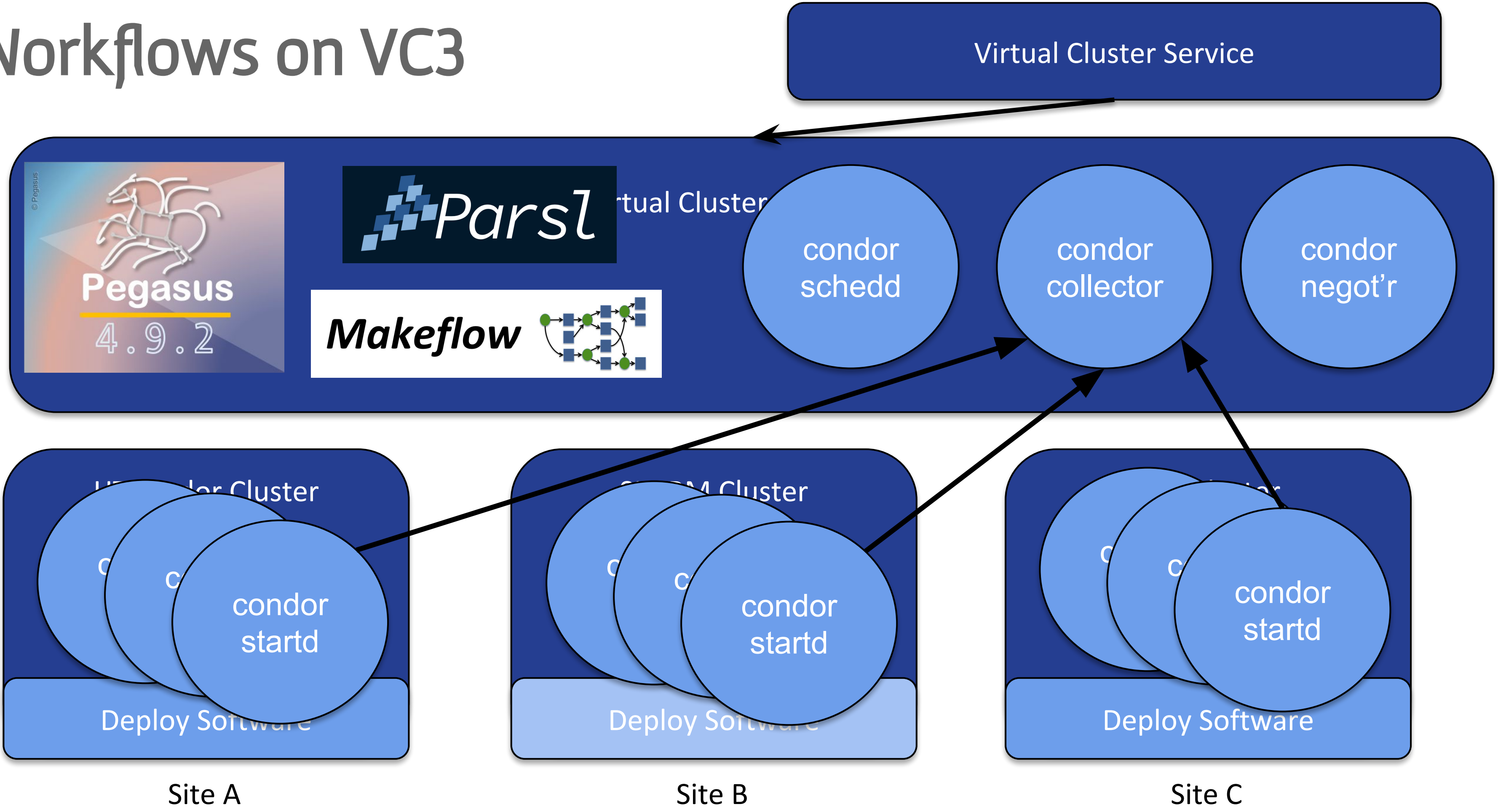
Use Cases

- HTCondor Pool
  - Already demonstrated self-service, fully automated provisioning clusters, binding allocations into projects
- Workflow Technologies
  - Makeflow + Work Queue Pegasus + HTCondor,
  - Parsl + iPyParallel
- Spark
  - **D**eployment of a Spark cluster for data analysis on top of existing schedulers
- JupyterLab
  - **Popular interactive** analytics environment - provisionable by VC3
- SCAILFIN + REANA
  - Complex set of REANA services deployed on minikube on head node + HTCondor on Stampede, Blue Waters and PSC.
- KOTO
  - Helping a new collaboration with no establishes resources to run on BNL HPC, KEK HPC + OSG HTC + campus resources.

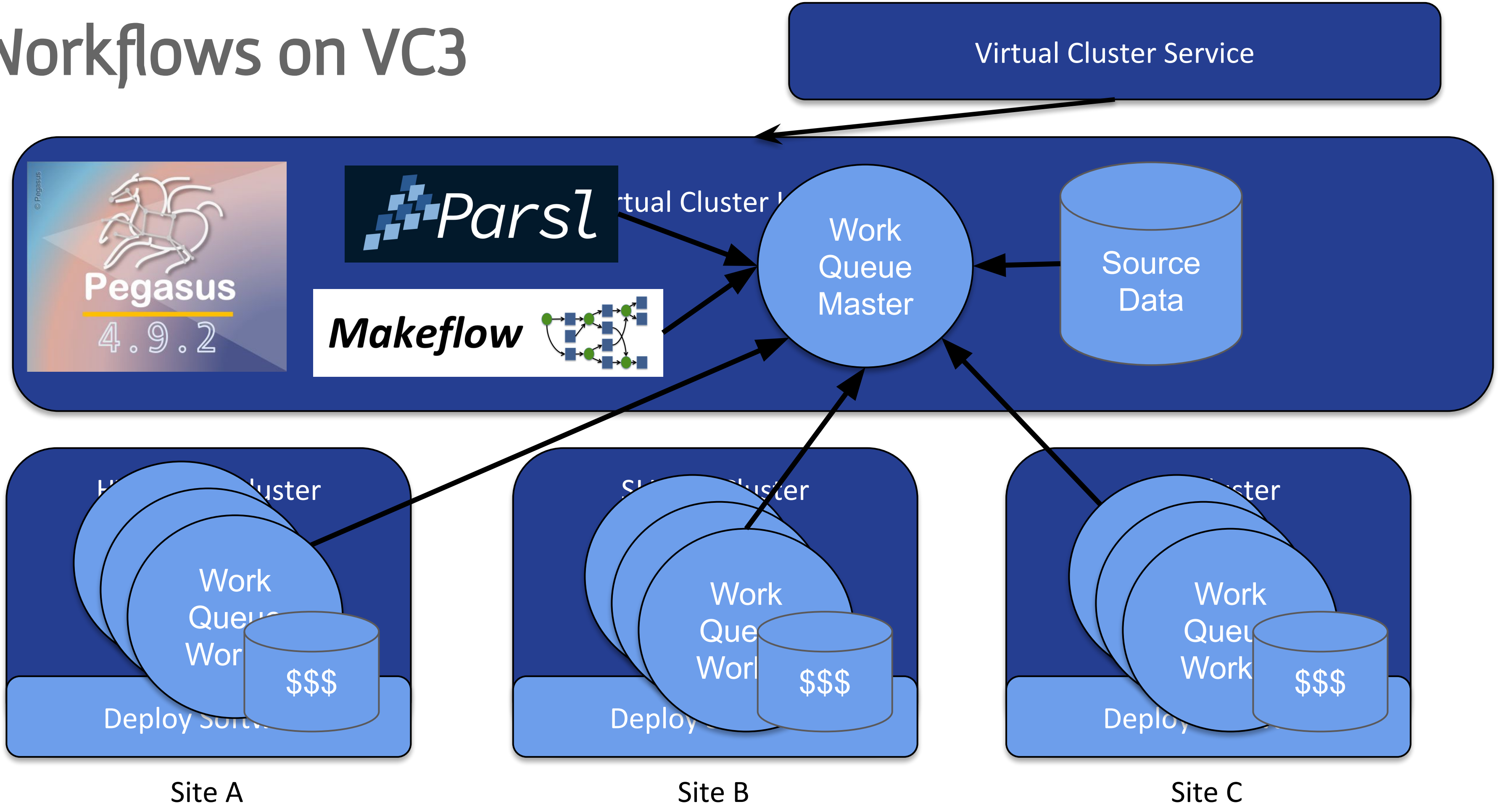
HTCondor Cluster on VC3



Workflows on VC3



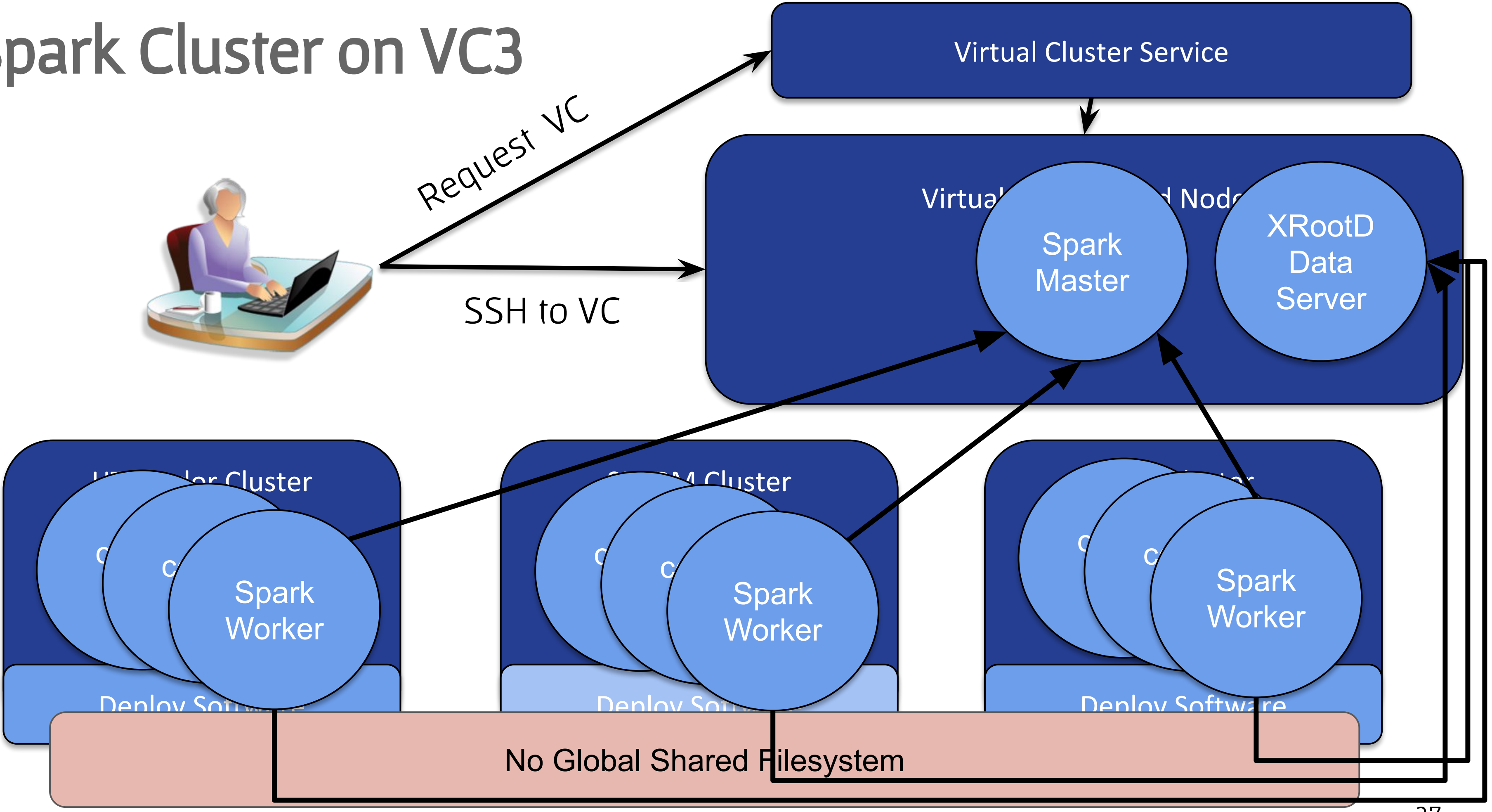
Workflows on VC3



Spark Middleware Integration

- Apache Spark now a first-class supported middleware in VC3
- Spark master runs on the virtual cluster head-node
- Virtual cluster workers are spark slaves
    - Java JRE, spark, scala, pyspark are deployed as-needed by the target resource.
- A shared secret secures connection to the master from workers and application clients
- No shared filesystem, use of an IO driver is needed.

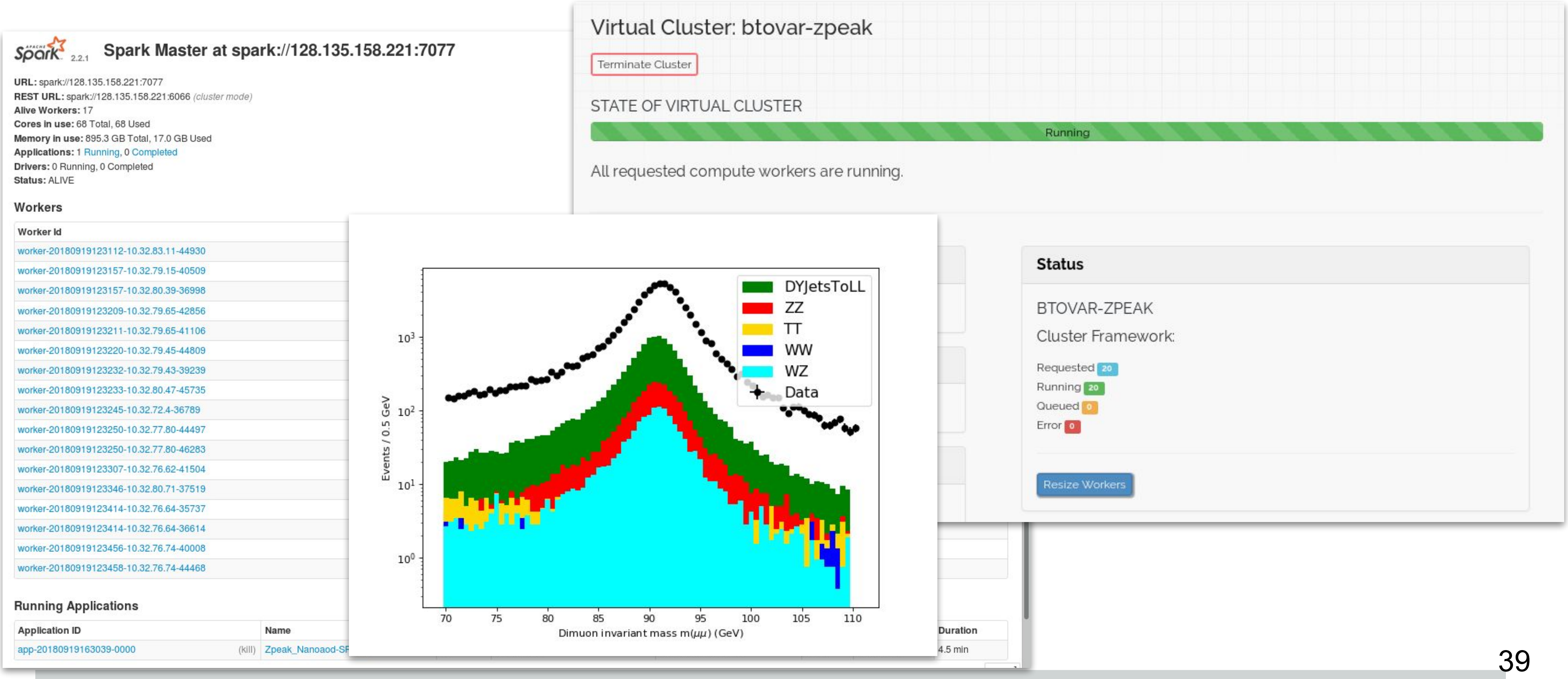
Spark Cluster on VC3



Spark & CMS Analysis

- Late-stage custom analysis code processing a Mini-AOD file to produce a histogram.
- Application written in python, using pyspark
- Data IO with the spark-xrootd plugin
  - Automatically installed in headnode and workers
- Ran with 20 workers, 4 cores each.
- Maximum concurrency was 80 tasks.

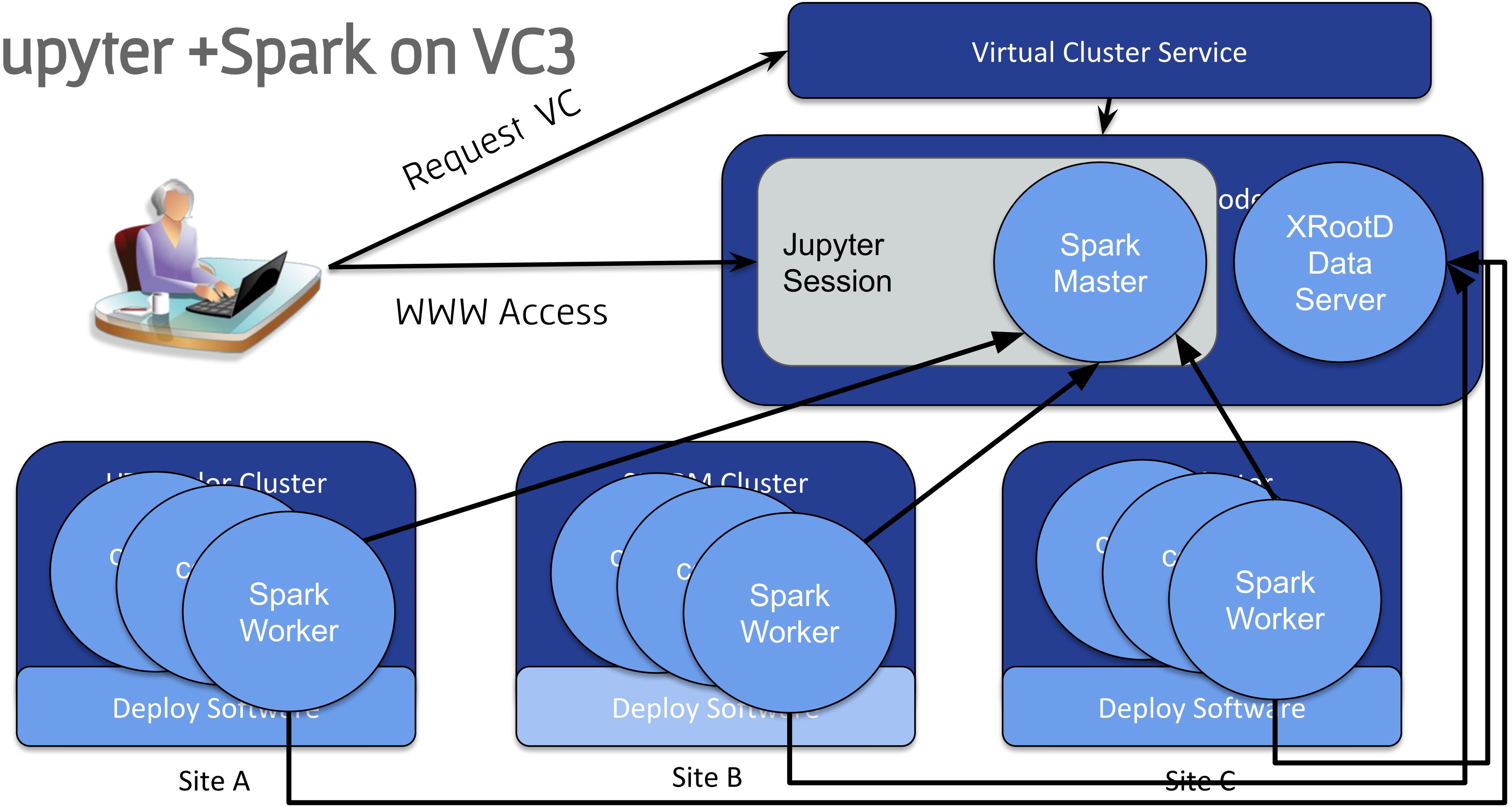
Spark & CMS Analysis



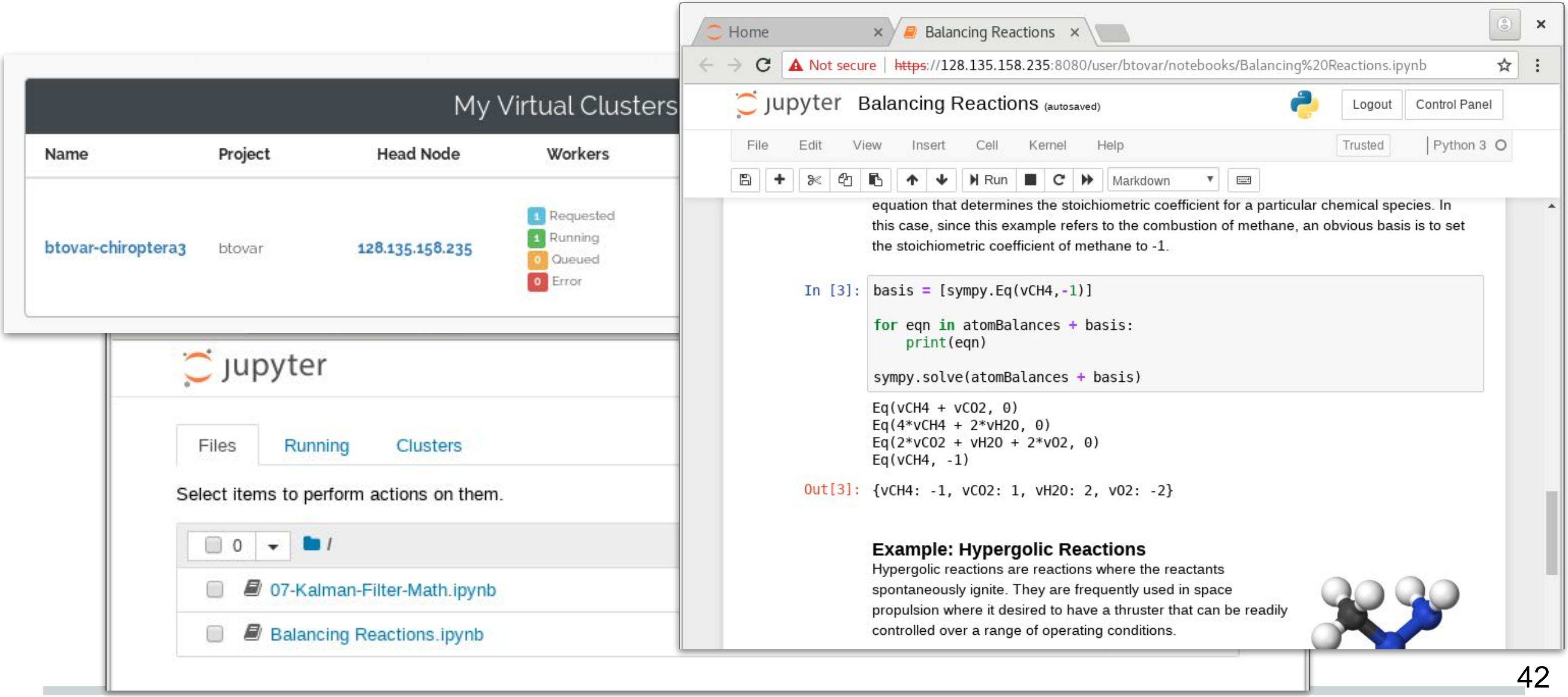
Provisioning JupyterLab Notebooks

- JupyterLab-based head nodes now launchable in VC3 development branch
- Web-based notebook interface for Python
- Uses Globus identity plugin to login
- Integration with HTCondor and other middleware types

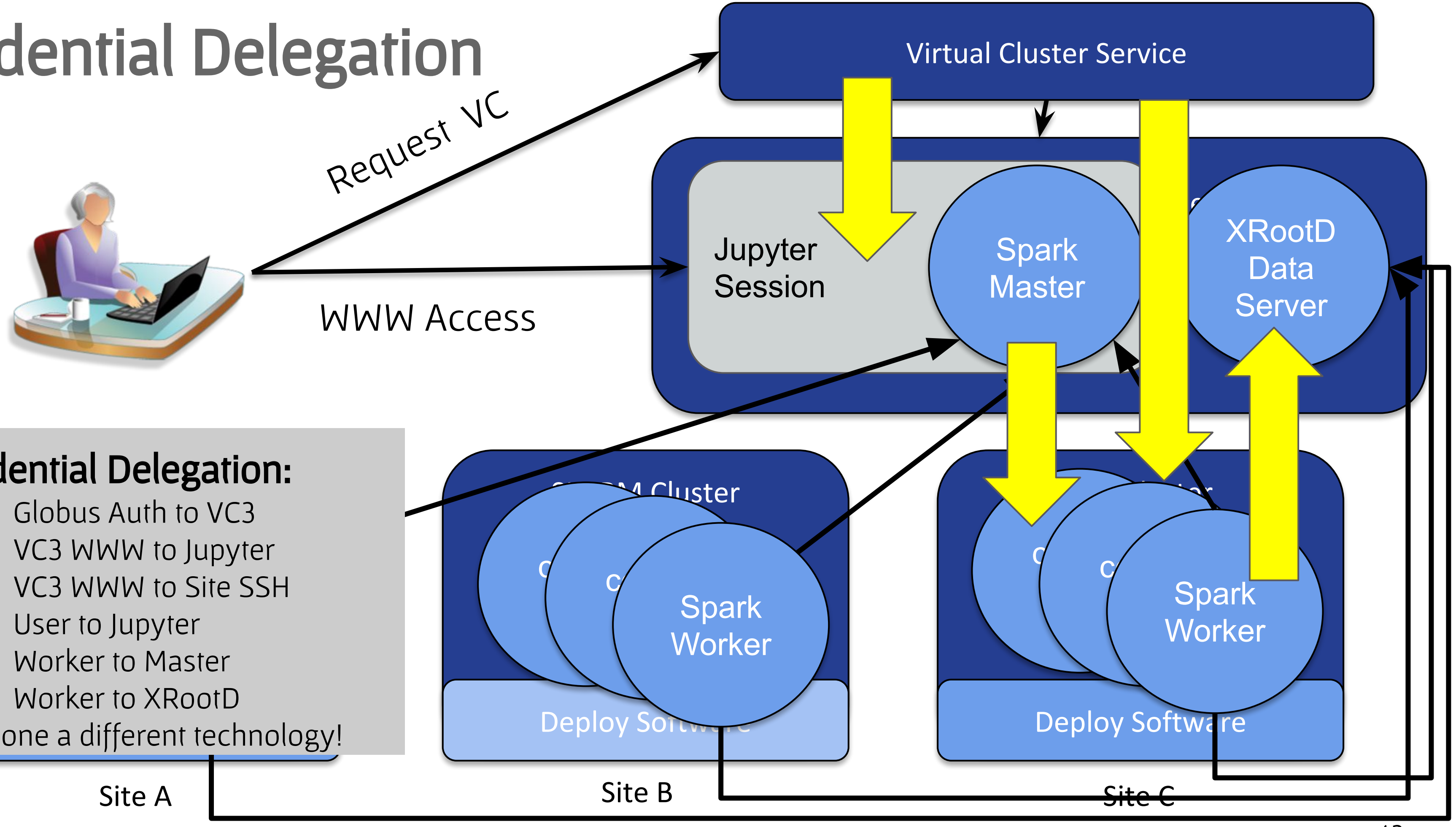
Jupyter +Spark on VC3



Provisioning JupyterLab Notebooks



Credential Delegation



Credential Delegation:

- Globus Auth to VC3
- VC3 WWW to Jupyter
- VC3 WWW to Site SSH
- User to Jupyter
- Worker to Master
- Worker to XRootD

Each one a different technology!

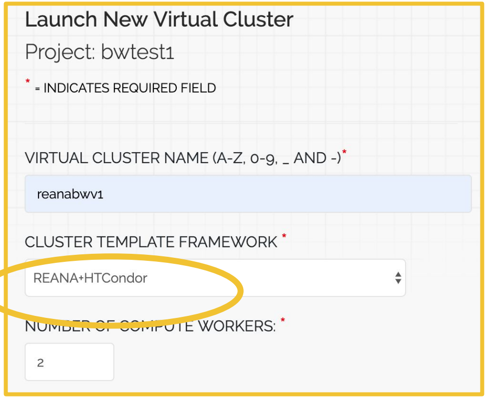
Scalable cyberinfrastructure on HPC facilities

SCAILFIN: Scalable CyberInfrastructure for Artificial Intelligence and Likelihood Free Inference

The SCAILFIN project aims to deploy artificial intelligence and likelihood-free inference techniques and software using scalable cyberinfrastructure (CI) that is developed to be integrated into existing CI elements, such as the **REANA** system, to work on **HPC facilities**. → VC3

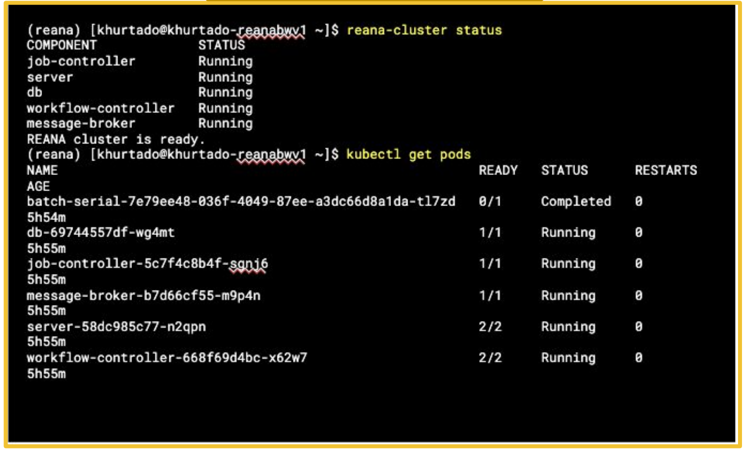
reana

Reproducible research data analysis platform



REANA is automatically deployed and integrated with HTCondor in VC3

VC3 headnode



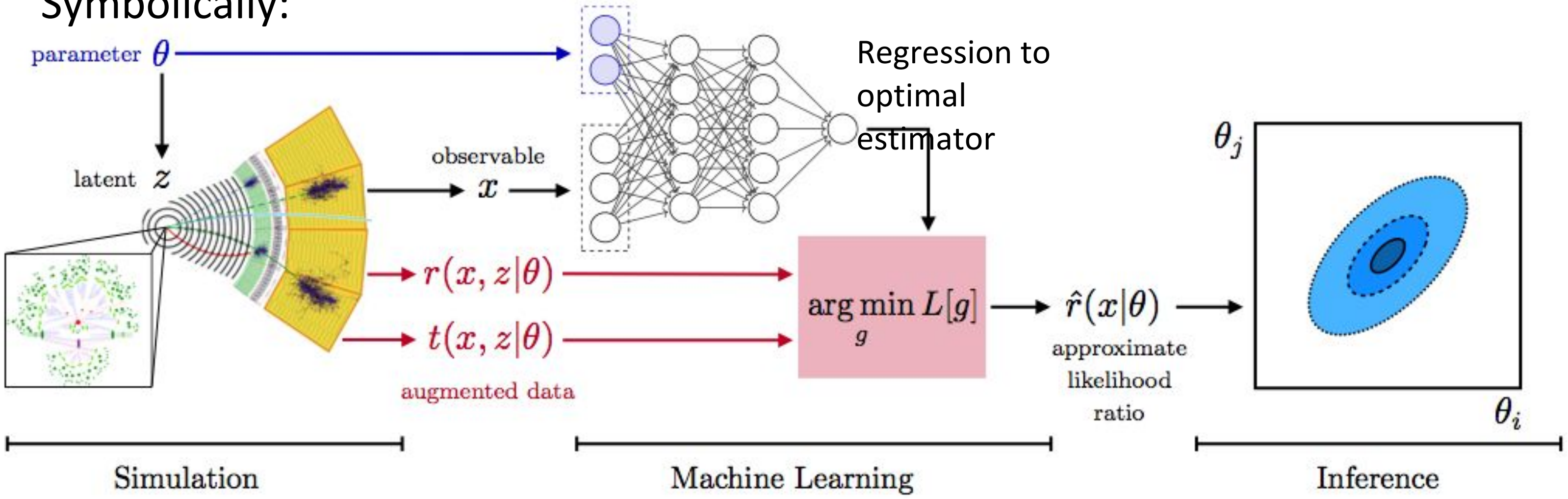
```
(reana) [khurtado@khurtado-reanabwv1 ~]$ reana-cluster status
COMPONENT           STATUS
job-controller      Running
server              Running
db                  Running
workflow-controller Running
message-broker      Running
REANA cluster is ready.
(reana) [khurtado@khurtado-reanabwv1 ~]$ kubectl get pods
NAME                                                          READY   STATUS      RESTARTS
AGE
batch-serial-7e79ee48-036f-4049-87ee-a3dc66d8a1da-tl7zd       0/1     Completed   0
5h54m
db-69744557df-wg4mt                                           1/1     Running     0
5h55m
job-controller-5c7f4c8b4f-sgnj6                               1/1     Running     0
5h55m
message-broker-b7d66cf55-m9p4n                                1/1     Running     0
5h55m
server-58dc985c77-n2qpn                                       2/2     Running     0
5h55m
workflow-controller-668f69d4bc-x62w7                          2/2     Running     0
5h55m
```

REANA components are started via kubernetes(minikube)

Simulation-Based Likelihood Free Inference

J. Brehmer, K. Cranmer, G. Louppe, and J. Pavez. arXiv:1805.00020, 2018.

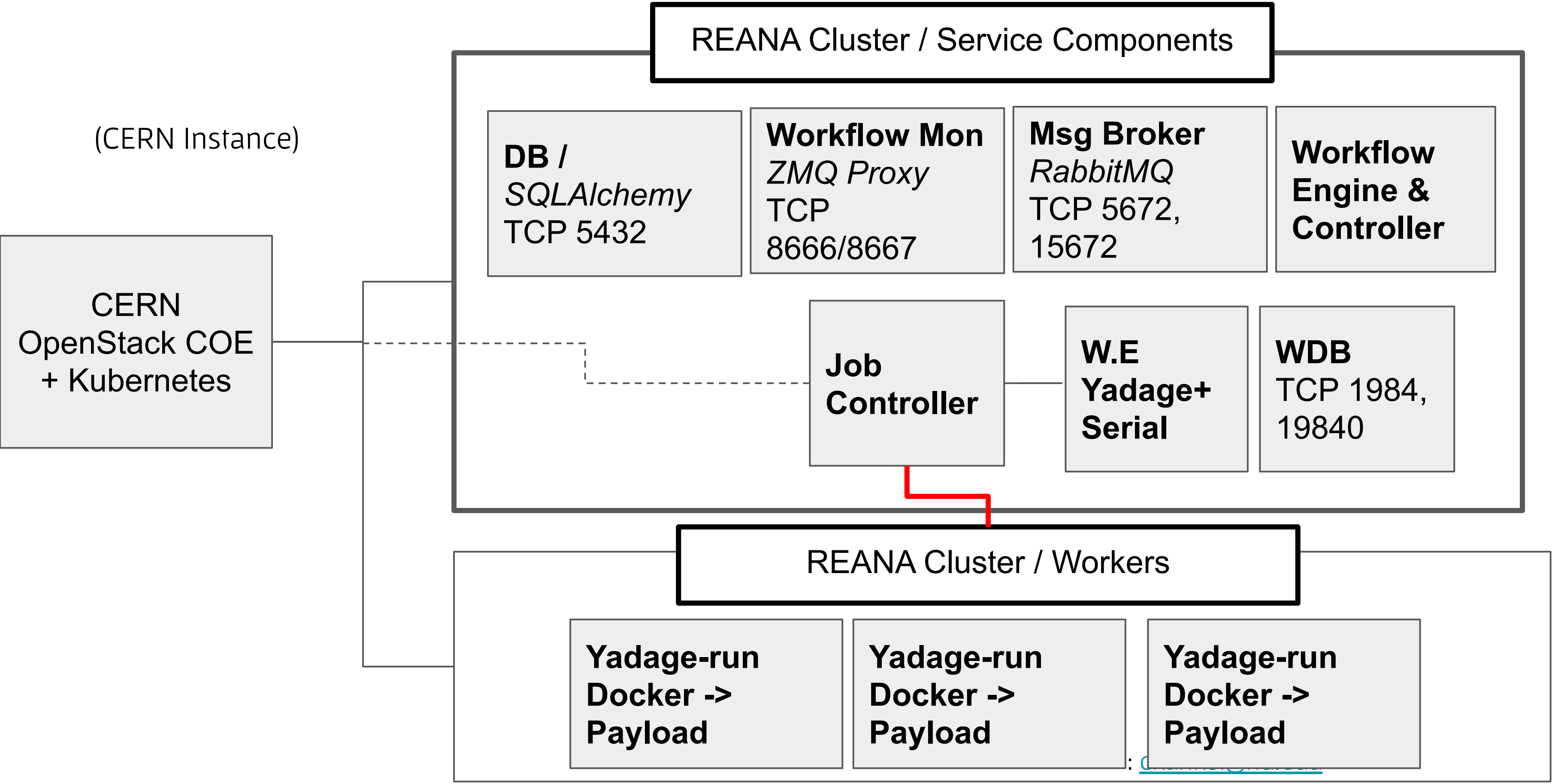
Symbolically:



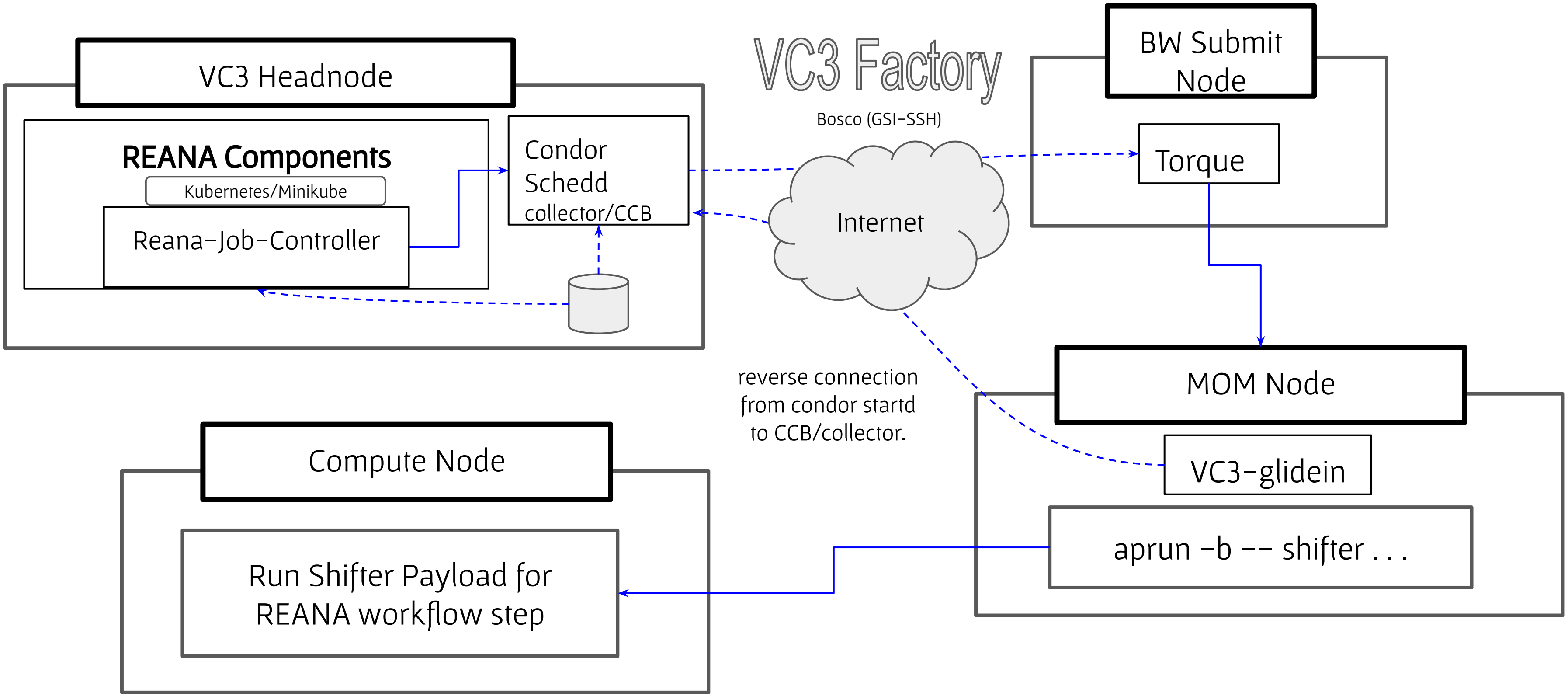
Estimation of optimal estimator lends itself to ML methods:

- Training data derived from simulations
- Can be guided by optimal sampling based on phase space density of generator, sensitivity to physics under study

reana-cluster - Simplified Diagram

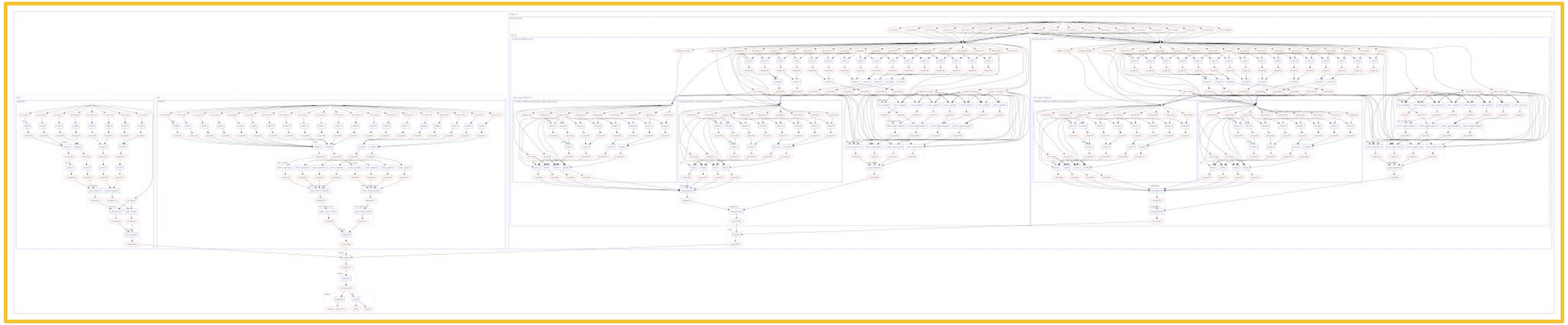


SCAILFIN on Blue Waters via VC3



SCAILFIN on Blue Waters via VC3

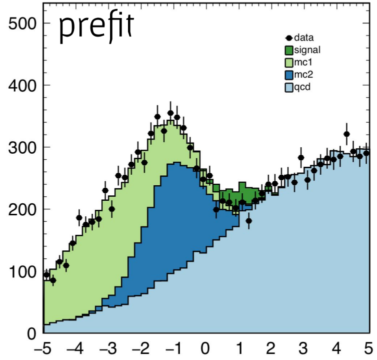
Complex Beyond Standard Model example (ran at BW via REANA + HTCONDO Rvia VC3)

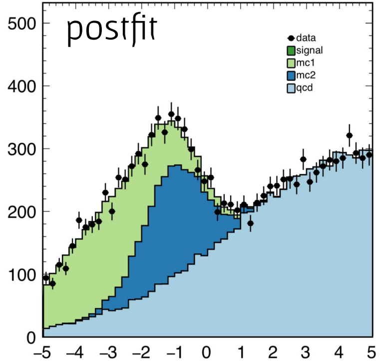


Data is generated/emulated according to Standard Model expectations.

After processing, a statistical model involving both signal and control regions is built and the model is fitted against the observed data.

The signal sample is scaled down significantly to fit the data, which is expected since the data was emulated in accordance with a SM-only scenario

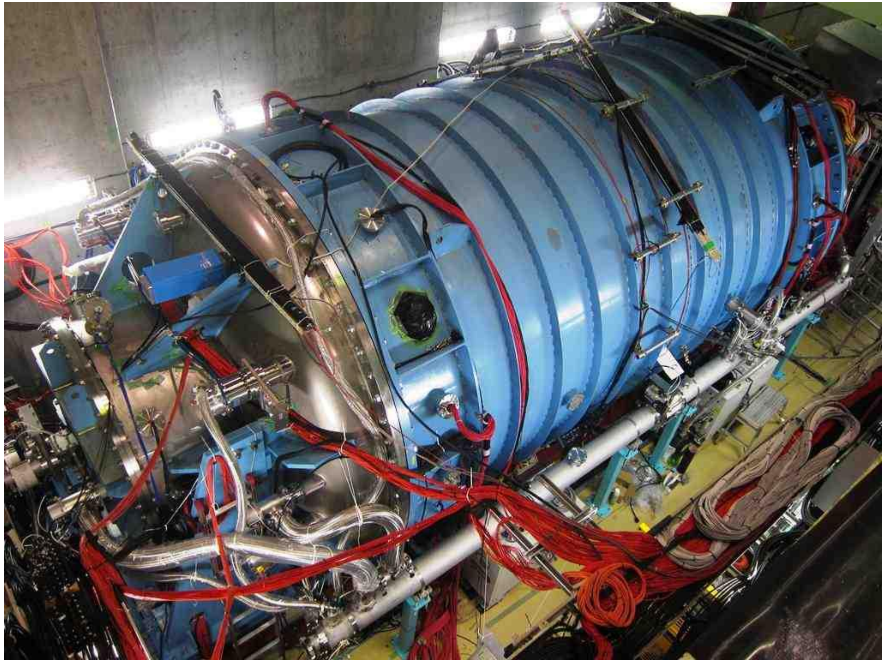




KOTO experiment

High energy physics experiment at the J-PARC (Japan Proton Accelerator Research Complex) Facility in Japan - http://koto.kek.jp/

Experiment measures the decay rate of neutral K-mesons (kaons) into neutral pi-mesons and a pair of neutrinos



Requirements

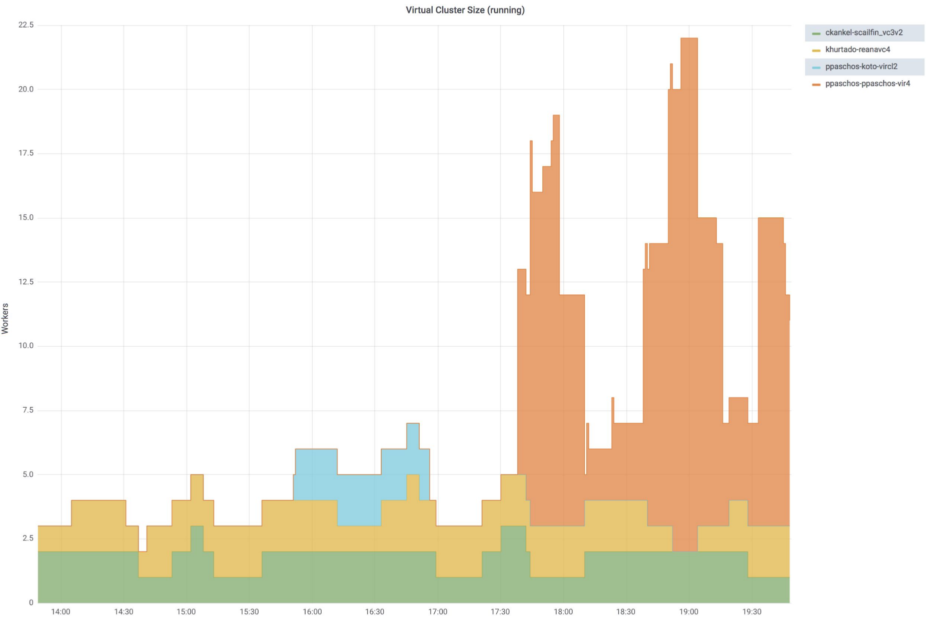
Storage: Estimated storage for accommodating the MC and MC/Analysis portions of the pipeline for a full experiment cycle is in the order of 200 TB per experiment cycle

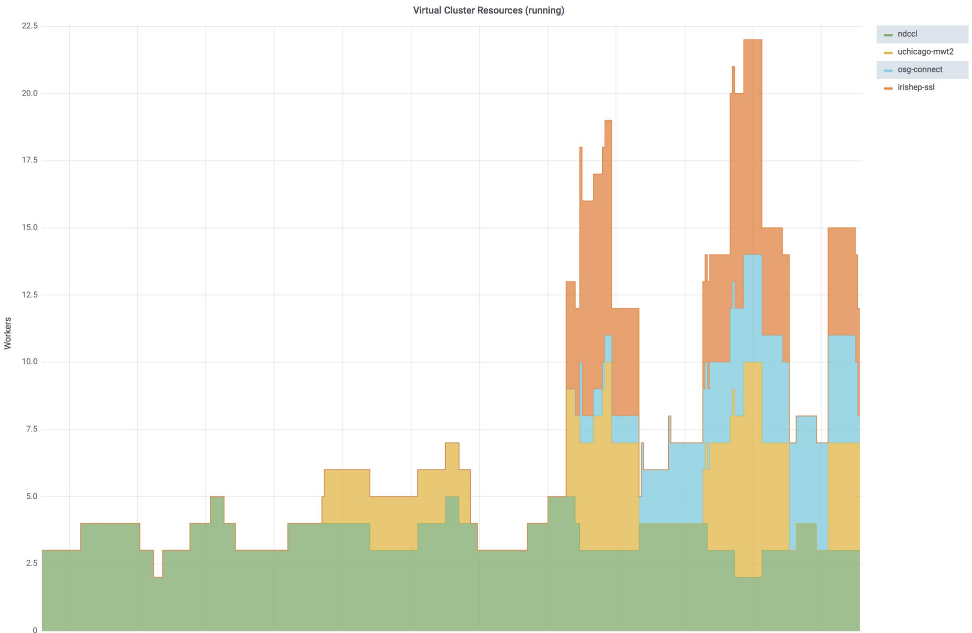
Software stack: There are two applications running on the KEKCC machine that needed to be rebuilt on the OSG side under RHEL7. A KOTO GEANT package and an analysis collection of tools. Most of the effort so far is to provide the environment for the software to be built and run on remote OSG sites

Submission scripts: KEKCC uses the LSF scheduler (bsub). Submission scripts needed to be modified to HTCondor

VC3 job submission for KOTO

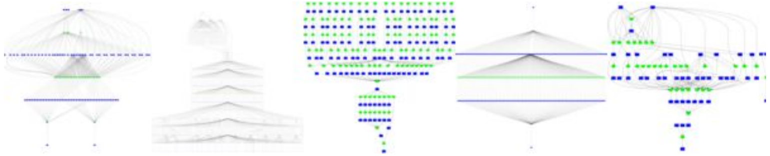
- OSG Connect (blue), UC (yellow), IRIS-HEP-SSL (orange)
- VC3 evenly distributes the load across resources

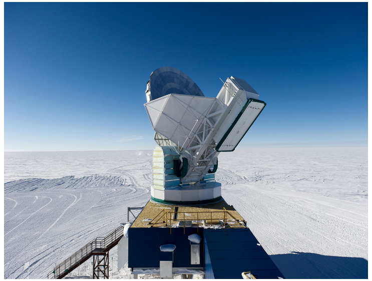


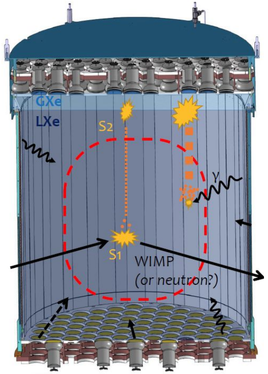


Working Middleware and Applications

- Various Bioinformatics Workflows
  - Makeflow + HTCondor + BWA, Shrimp, BLAST.
- Lobster CMS Data Analysis
  - Work Queue + Builder + CVMFS
- Lifemapper
  - Work Queue + Builder
- Spark CMS Data Analysis
- South Pole Telescope (SPT-3G) Analysis Framework
  - HTCondor Jobs + Docker/Shifter + CVMFS
- XENON1T Analysis Framework
  - Pegasus + HTCondor + CVMFS
- MAKER Bioinformatics Pipeline
  - Work Queue + Builder
- IceCube Simulation Framework
  - HTCondor







Thoughts and Lessons Learned

Successful Ideas that Worked

- Automated Provisioning of Cluster Frameworks Across Heterogeneous HPC Resources
- Automated Deployment of User-Level Software
  - (Build itself may require substantial cluster resources)
- Internal Architecture for Robust Service Deployment
  - (Surprisingly complex state machine.)
- Multi-Layer Configuration Enables Collaborations
  - Credentials go deep into the framework

Things That were Harder than Expected

- New Security Requirements (MFA) Spread Very Quickly
  - Working Idea: Let user login and “pull” the cluster blueprint to the local site where it can be deployed.
- Difficulty of Curating Site/Cluster Details due to Churn
  - Choice: Either invest more in active curation, or involve users in the configuration process.
- Debugging Complex Service Deployments
  - Working Idea: Test before using, expose stages of success to user, give concrete feedback for fixing.

Portable Ideas to Keep and Use Elsewhere

- Importance of the Head Node as Infrastructure
  - Allocatable, Personal, Shareable, Configurable
- Self-Service Deployment Model
  - Generalization of Infrastructure-as-Code:
  - OpSys + Software + Middleware + Entry Point
- A Model for Sharing Institutional Resources
  - Classic: Submit My Work to Remote Environments
  - VC3: Deploy My Environment on Remote Resources

Technology Evolution 2016–2020

Evolution from VMs to Containers:

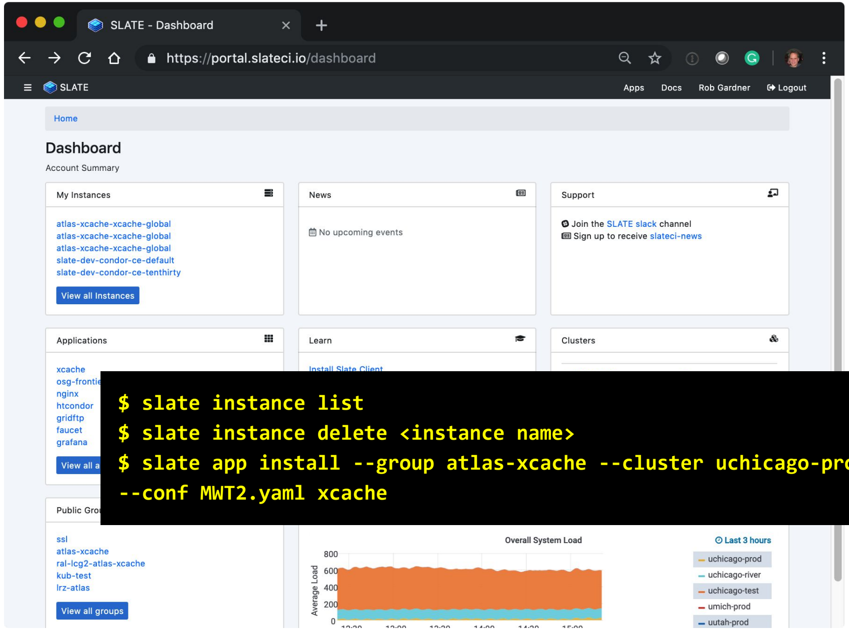
- Swap out OpenStack in favor of Kubernetes, which is lightweight and better supports process automation.
- Docker is being supplanted by Singularity, and other container environments.

Evolution from credentials to capabilities:

- ssh-keygen -> Upload Pubkey -> RSA-SSH (Standalone)
- Globus Auth -> GSI Token -> GSI-SSH (DCDE Fed.)
- Globus Auth -> Web Tokens -> WebTokens-SSH (Web Native)
- SciTokens? (scitokens.org)

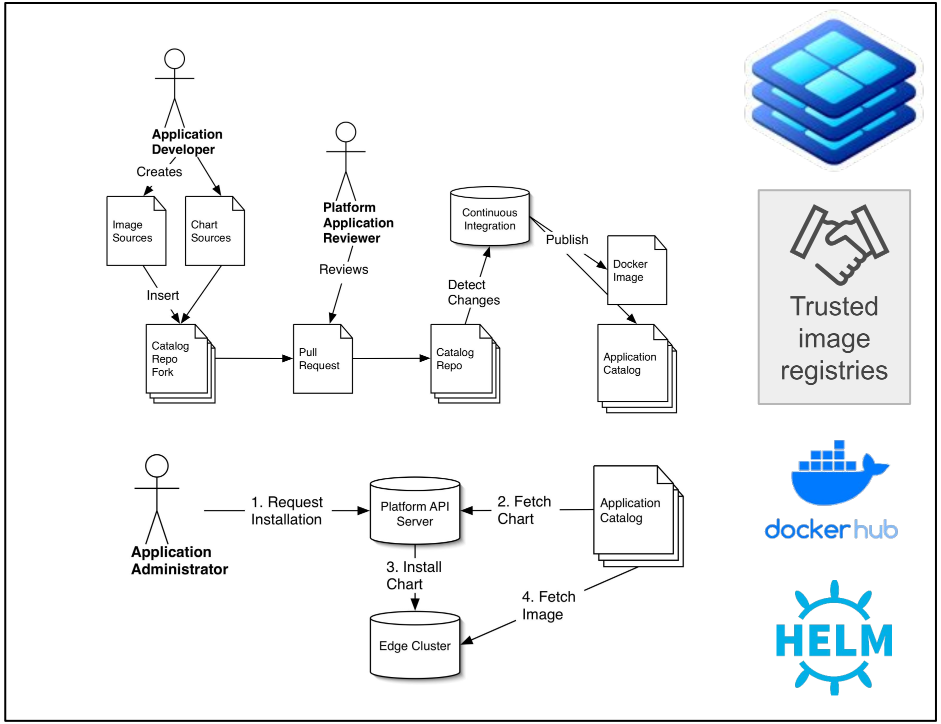
Towards Federated Ops with SLATE

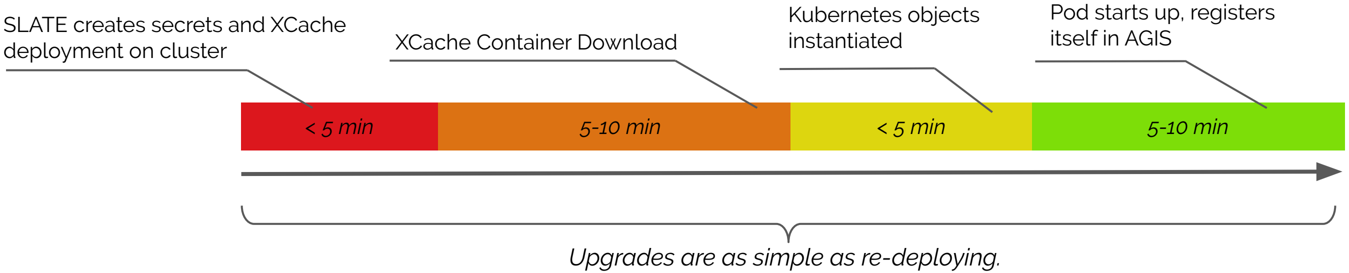
- Remotely manage edge services at sites by **expert teams** from **trusted organizations**
- Deploy updates more quickly & introduce new services more easily
- Save time and effort for local site admins -- towards **OSG NoOps**
- Edge federation via lightweight server/client overlay using **Kubernetes**, the industry leading container orchestration platform
- Software catalog, with push button deploy using vetted **Helm** charts




```
$ slate instance list
$ slate instance delete <instance name>
$ slate app install --group atlas-xcache --cluster uchicago-prod
--conf MWT2.yaml xcache
```

- [SLATE](#) - a value added K8s distribution
  - Support for CVMFS, ingress controller (multi-tenant, scoped privileges), Prometheus monitoring, **curated application catalog w/ Jenkins CI**
- Site security & policy conscious
  - SLATE works as an unprivileged user
  - Single entrypoint via **institutional identity**
  - Site owner controls group whitelists & service apps; **retains full control**
- With OSG, WLCG, [trustedci.org](#) & others working to establish a "CISO compliant" security posture and **new trust delegation model**

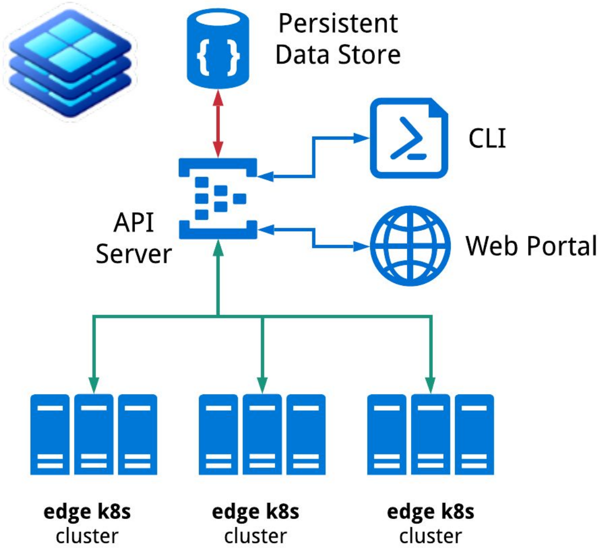


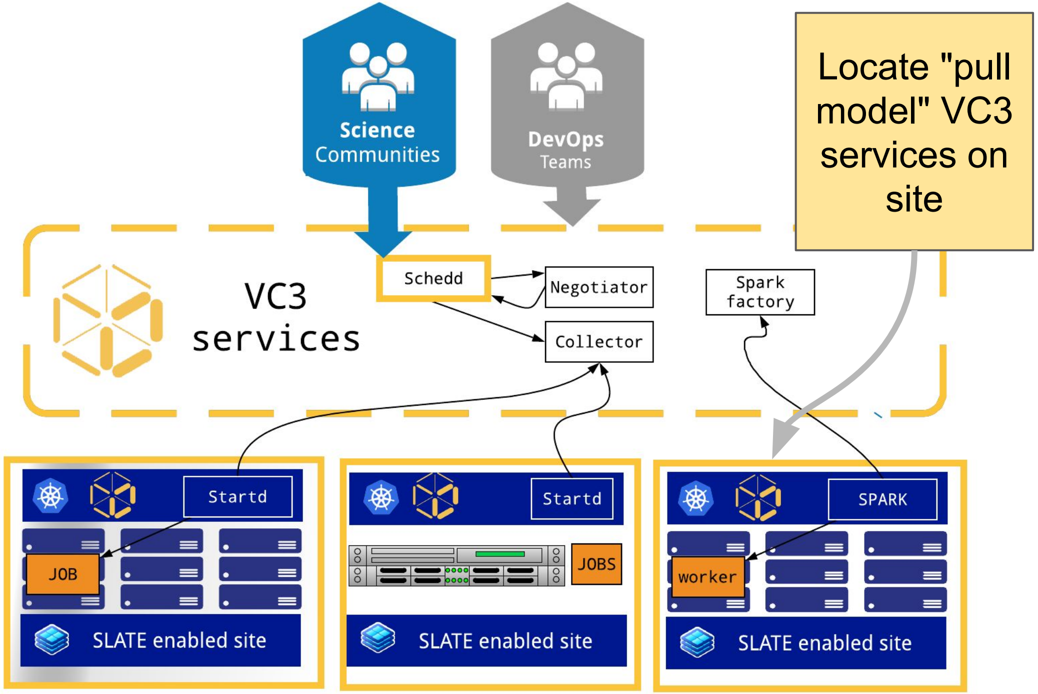


A data caching network deployed in less than 20 minutes.

VC3 and SLATE synergies

SLATE: federated service orchestration in the SciDMZ

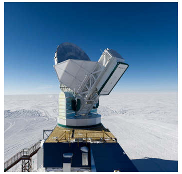




Open Problems in Federated Systems

- Troubleshooting Across Systems
  - Many stakeholders, system churn, lack of evidence.
  - Troubleshooting posed as a distributed database query?
- State Management and Garbage Collection
  - Remote creation of containers for local sites, how long must they be kept around?
- Socio-Technical Understanding of Delegation
  - User X submits jobs to user Y's account on site S. Ok?
  - Technical ability to represent non-local account?

Collaborators and Connections



Xe XENON Dark Matter Project

CMS Compact Muon Solenoid

ATLAS EXPERIMENT

reana

Makeflow

AutoPyFactory

HTCondor High Throughput Computing

Parsl

Pegasus 4.9.2

SGCI Science Gateways Community Institute

globus

Open Science Grid

ci connect

VC3

Virtual Clusters for Community Computation

https://www.virtualclusters.org

@virtualclusters

UNIVERSITY OF NOTRE DAME

THE UNIVERSITY OF CHICAGO

BROOKHAVEN NATIONAL LABORATORY

New users signup: http://bit.ly/vc3-signup

Register your HPC: http://bit.ly/vc3-new-resource

Supported by the Department of Energy Office of Advanced Scientific Computing Research and Next Generation Networking Services, Solicitation DE-FOA0001344 (DDRM), Proposal 0000219942.

VC3 Funding and Team

Funded by DOE Office of Advanced Scientific Computing Research (ASCR) and NSF Next Generation Networking Services (NGNS)

PIs: Rob Gardner (UC), Douglas Thain (ND), and John Hover (BNL)

co-PIs: David Miller (UC), Paul Brenner (ND), Mike Hildreth (ND), Kevin Lannon (ND)

dev-team: Lincoln Bryant (UC), Benedikt Riedel (UC), Suchandra Thapa (UC), Jeremy Van (UC), Kenyi Hurtado Anampa (ND), Ben Tovar (ND), Jose Caballero Bejar (BNL).

U.S. DEPARTMENT OF ENERGY | Office of Science

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