GlideinWMS

Marco Mambelli

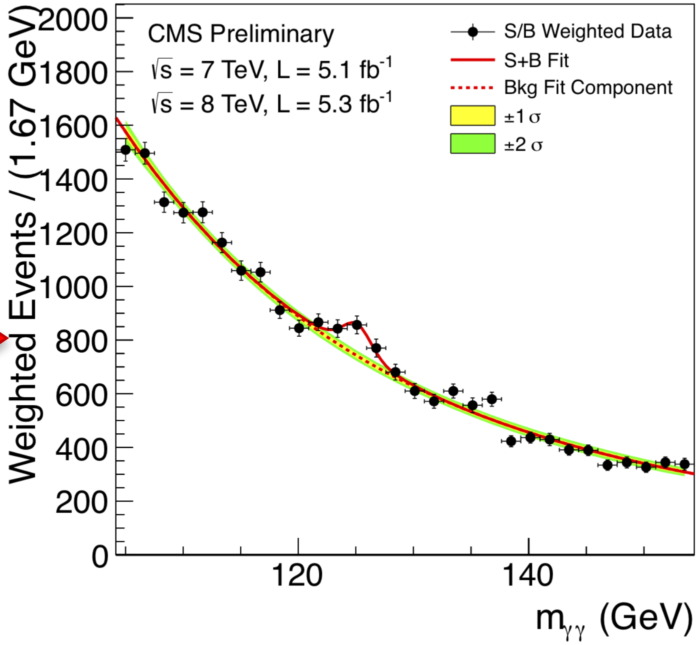
Summer Student Guidelines

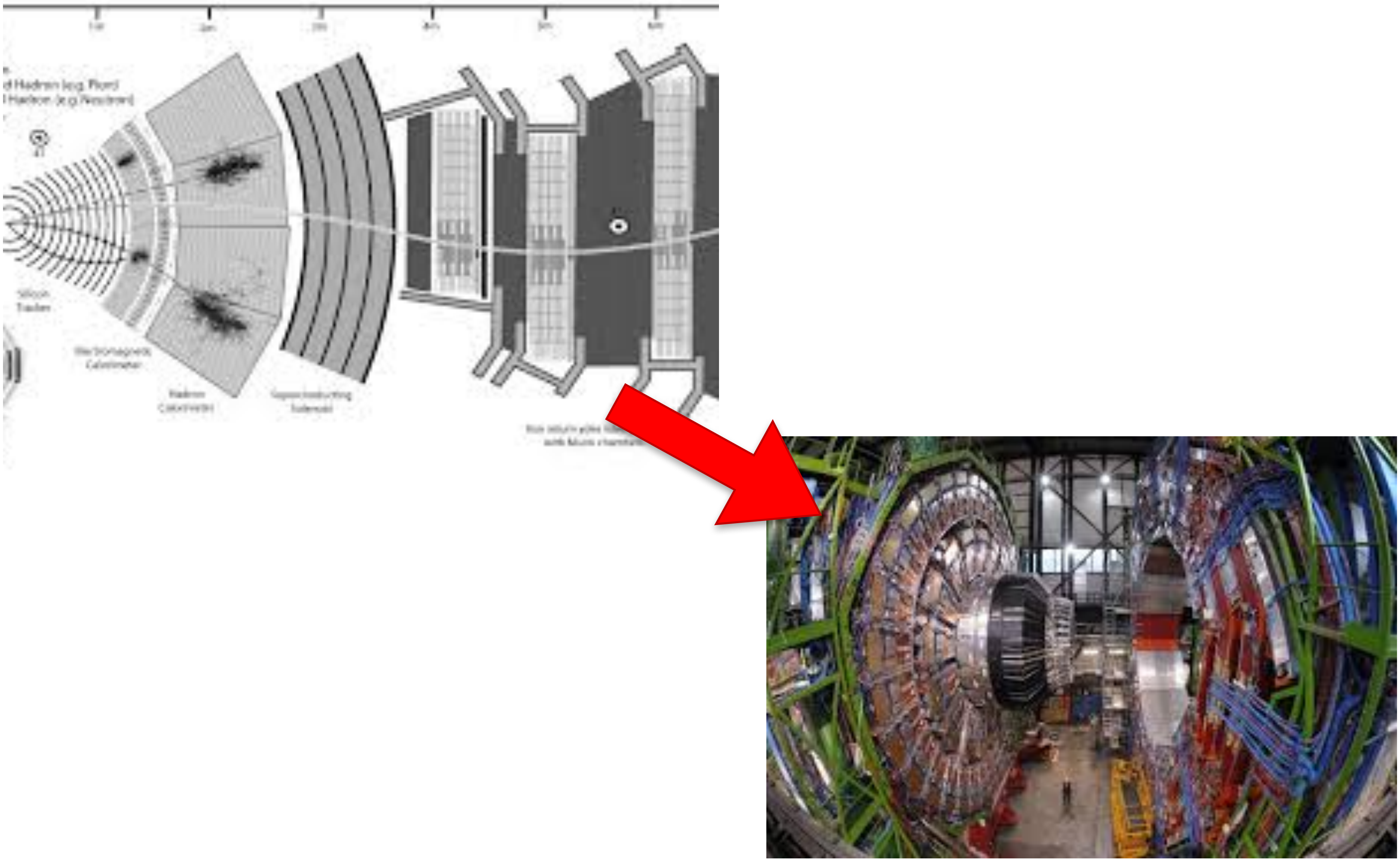
May 2020

Outline

- Scientific computing
- GlideinWMS
- HTCondor
- Resources
- Monitoring
- Links
- Demo

- **Scientific computing**
- GlideinWMS
- HTCondor
- Resources
- Monitoring
- Links
- Demo


CMS Preliminary
√s = 7 TeV, L = 5.1 fb⁻¹
√s = 8 TeV, L = 5.3 fb⁻¹
S/B Weighted Data
S+B Fit
Bkg Fit Component
±1 σ
±2 σ
Weighted Events / (1.67 GeV)
0
200
400
600
800
1000
1200
1400
1600
1800
2000
120
140
mγγ (GeV)


1m
2m
3m
4m
5m
6m
d Hadron (e.g. Pion)
l Hadron (e.g. Neutron)
Silicon Tracker
Electromagnetic Calorimeter
Hadron Calorimeter
Superconducting Solenoid
Iron return yoke
with Muon chambers

HEP experiments require computing!

- Accelerator and detector simulations
- Data reconstruction
- Data analysis

When one computer is not enough

✓ Supercomputer

✓ Cluster (Batch Systems)

✓ Grid

✓ Cloud

Fermilab

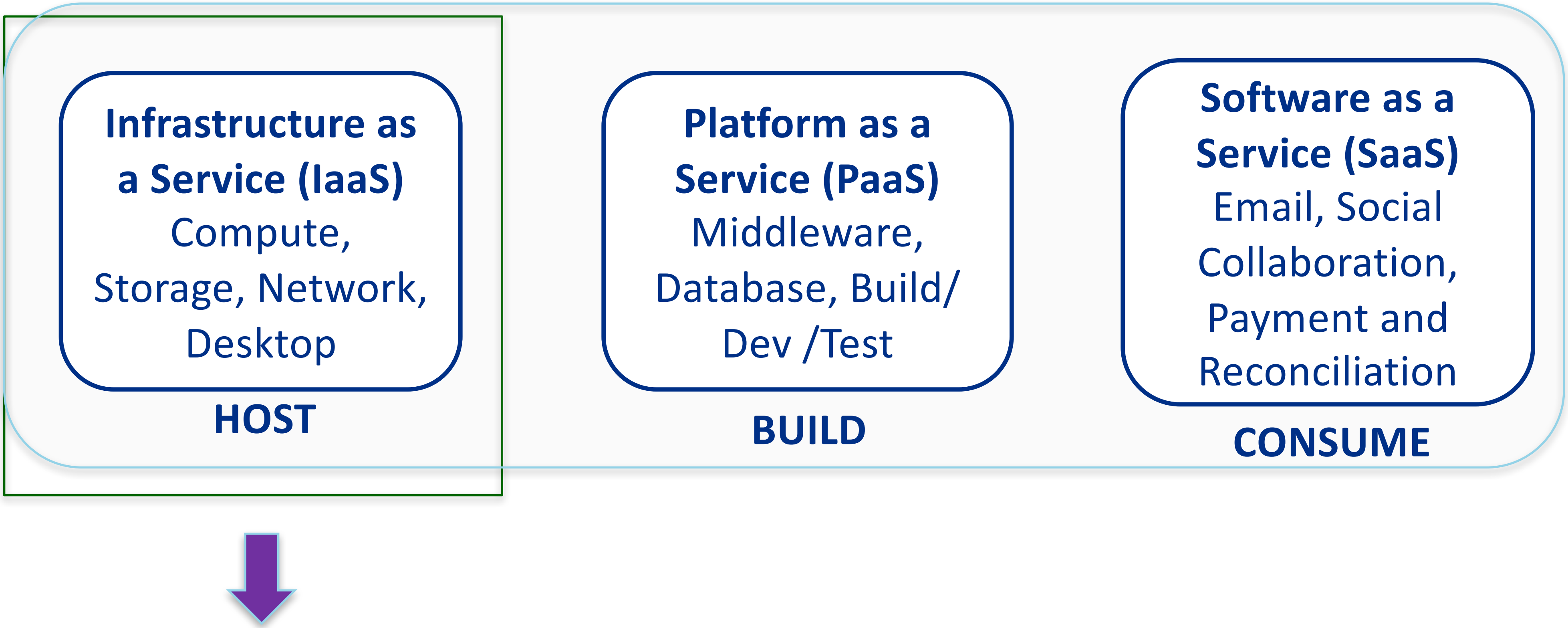
Computing resources

- Supercomputer
  - Special purpose computer fine tuned to achieve elevated number of operations per second
- Cluster (batch system)
  - Collection of parallel or distributed computers which are interconnected among themselves using high-speed networks
  - Local Resource Manager (or batch system) is the software managing the computers in the cluster (e.g. PBS, SLURM, HTCondor, SGE, LSF, ...)
- Grid (e.g. Open Science Grid)
  - Combines computers from multiple administrative domains to reach common goals, to solve tasks
  - System that coordinates resources which are not subject to centralized control, using standard, open, general-purpose protocols and interfaces to deliver nontrivial qualities of service

Computing resources (cont)

- Cloud
  - Refers to both the applications delivered as services over the Internet and the hardware and system software in the data centers that provide those services
  - aka Elastic computing, available or paid only when used
  - Software as a Service (SaaS) is a kind of services where in many users can make use of the software hosted by the service provider and pay only for time its being used (Workday, Slack)
  - Platform as a Service (PaaS) provides a high-level integrated environment to design, build, test, deploy and update online custom applications (Amazon orchestration, Google AE)
  - Infrastructure as a Service (IaaS) refers to the services provided to the users to use processing power, storage, network and other computing resources, to run any software including operating systems and applications (AWS, Google CE, Fermicloud)

Basic concepts – Service Models



Infrastructure as a Service (IaaS)
Compute, Storage, Network, Desktop

HOST

Platform as a Service (PaaS)
Middleware, Database, Build/ Dev /Test

BUILD

Software as a Service (SaaS)
Email, Social Collaboration, Payment and Reconciliation

CONSUME

Services provided to the users to use processing power, storage, network and other computing resources, to run any software including operating systems and applications (AWS, Google CE)

HTC problem: Growing needs and Trends

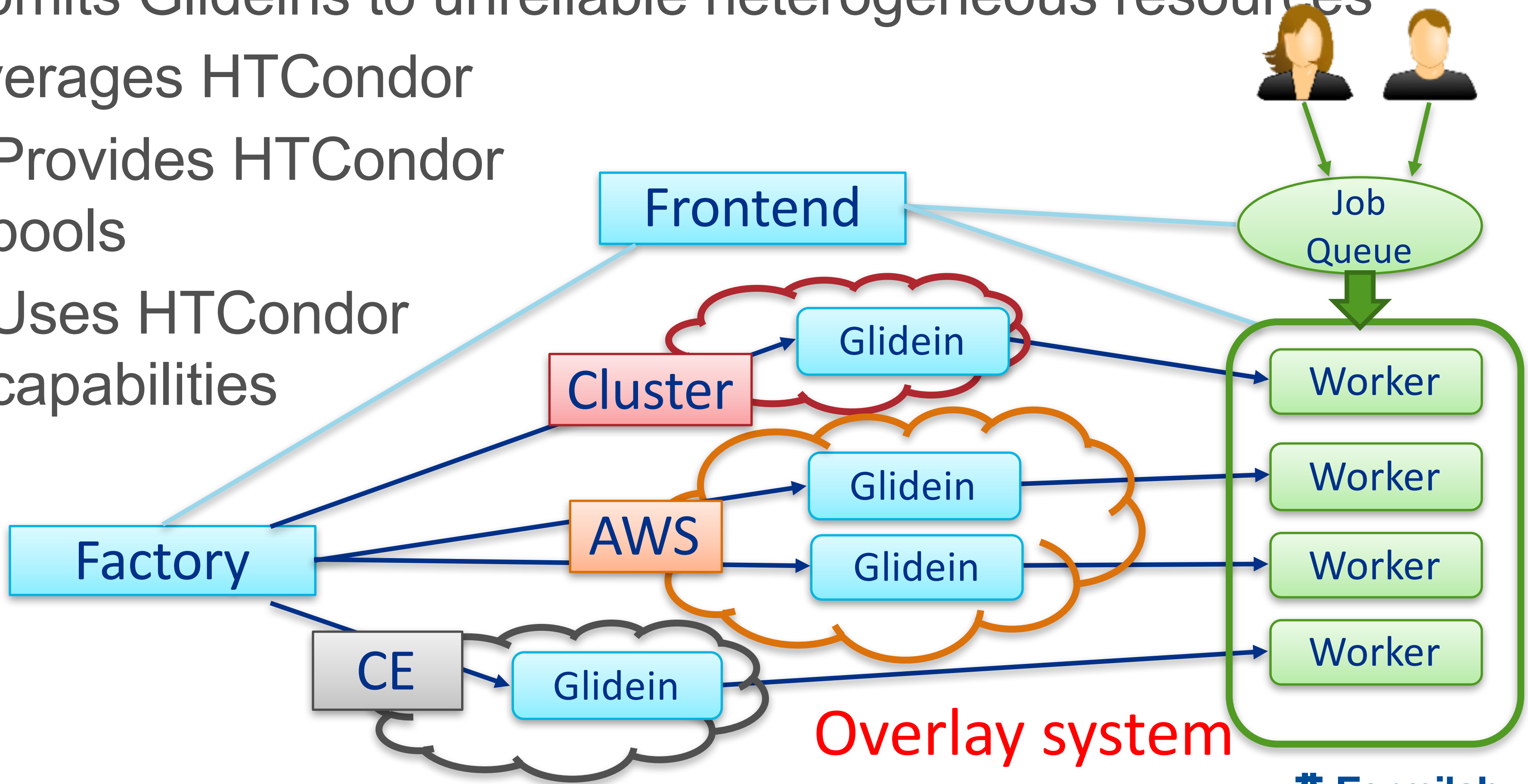
- High Throughput Computing
  - use of many computing resources over long periods of time to accomplish a computational task
- Need for more resources
  - Scale to more jobs
  - Access more resources
  - Simplify the management
- Less structured resources and infrastructure
  - Multiple organizations
  - Different systems
  - Less infrastructure
  - Different authentications

- Scientific computing
- **GlideinWMS**
- HTCondor
- Resources
- Monitoring
- Links
- Demo

GlideinWMS

GlideinWMS is a pilot based resource provisioning tool for distributed High Throughput Computing

- Provides reliable and uniform virtual clusters
- Submits Glideins to unreliable heterogeneous resources
- Leverages HTCondor
  - Provides HTCondor pools
  - Uses HTCondor capabilities



Glidein: node testing and customization

- Scouts for resources and validates the Worker node
  - Cores, memory, disk, GPU, …
  - OS, software installed
  - CVMFS
  - VO specific tests
- Customizes the Worker node
  - Environment, GPU libraries, …
  - Starting containers (Singularity, …)
  - VO specific setup
- Provides a reliable and customized execute node to HTCondor

Factory

- A Glidein Factory knows how to submit to sites
  - Sites are described in a local configuration
  - Only trusted and tested sites are included
- Each site entry in the configuration contains
  - Contact info (hostname, resource type, queue name)
  - Site configuration (startup dir, OS type, …)
  - VOs authorized/supported
  - Other attributes (Site name, core count, max memory, ...)
  - Glideins can also auto-detect resources
- Configuration can be auto-generated (e.g. from CRIC), admin curated, stored in VCS (e.g. GitHub)
- Condor does the heavy lifting of submissions.

Factory: Supported resources

- Remote or local clusters:
  - Can have batch systems other than HTCondor: PBS, SGE, Slurm, all supported.
- Grid sites (CREAM, ARC, HTCondor-CE)
- Hosted CEs
- Commercial cloud (AWS, Google)
- Open Source Cloud (OpenStack, OpenNebula)
- HPC sites
  - Uses an ssh-based system to ssh into HPC sites and submit directly from their login nodes.

Frontend

- Monitors jobs to see how many Glideins are needed
- Compares what entries (sites) are available
- Requests Glideins from the Factory
- Requests Factory to kill Glideins if there are too many
- Pressure-based system
  - Works keeping a certain number of Glideins running or idle at the sites
  - Glideins requests are gradual to avoid spikes and overloads
- Manages credentials and delegates them to the Factory.

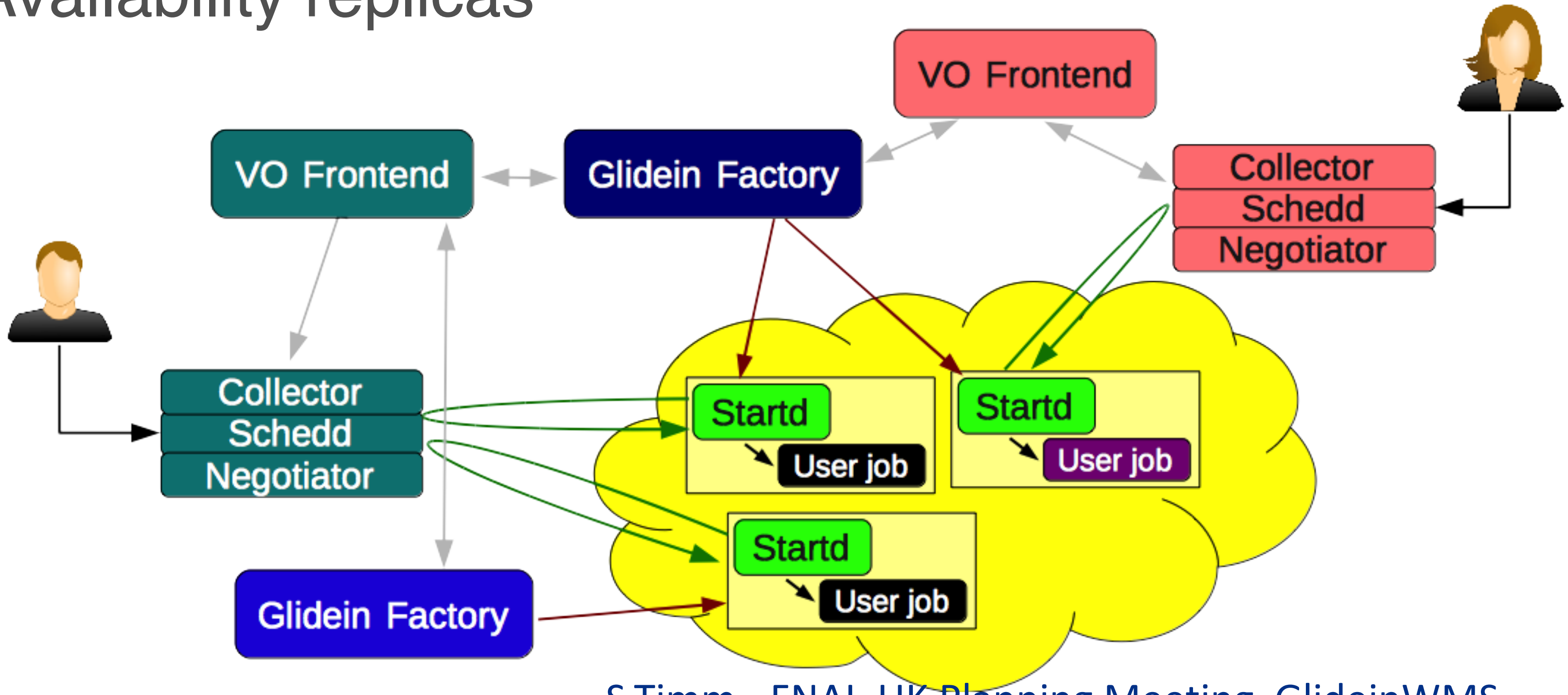
GlideinWMS components

FRONTEND	FACTORY
- Controlled by VO Operators - Main task is to look for user jobs and ask the Glidein Factories to provide glideins, if needed - Decides which glidein Factory should submit the pilot Jobs and how many of them - Configuration xml file: /etc/gwms-frontend/frontend.xml	- Controlled by Factory operators (OSG) - Advertises itself, listen for requests from Frontend clients and submit glideins - Can handle multiple kinds of glidein - Configuration xml file: /etc/gwms-factory/glideinwms.xml

GLIDEIN
- Requested by the Frontend, launched by the Factory and joins the virtual cluster - Property configured execution node submitted as a Grid job. - Defines how multi-core glideins should split their resources - Pilot jobs may run multiple user Jobs - glidein_startup.sh configures and starts the condor startd daemon

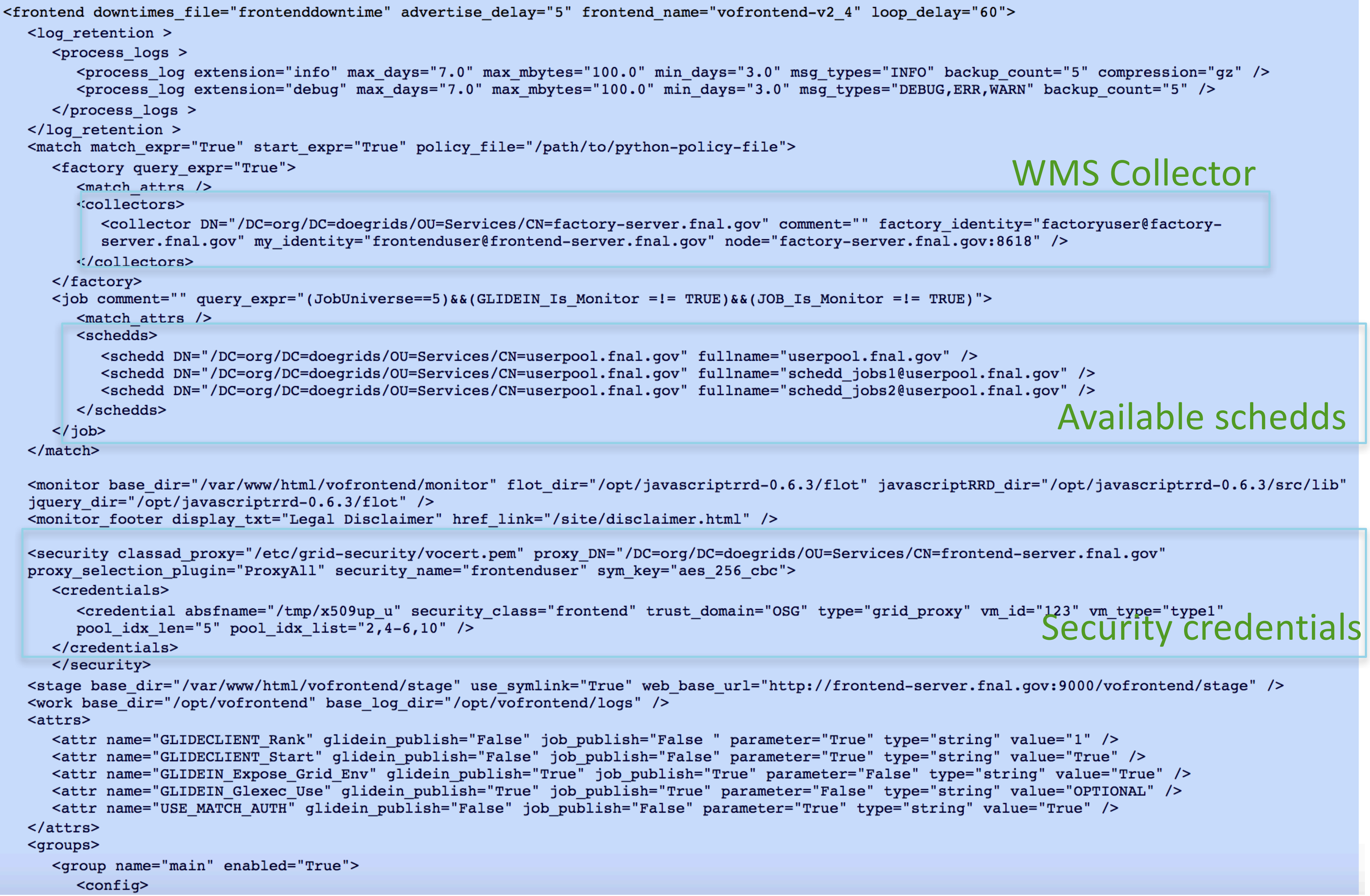
Distributed

- N-to-M relationship
  - Each Frontend can talk to many Factories
  - Each Factory may serve many Frontends
- Multiple User Pools
- High Availability replicas



S.Timm - FNAL-UK Planning Meeting GlideinWMS

Frontend configuration example



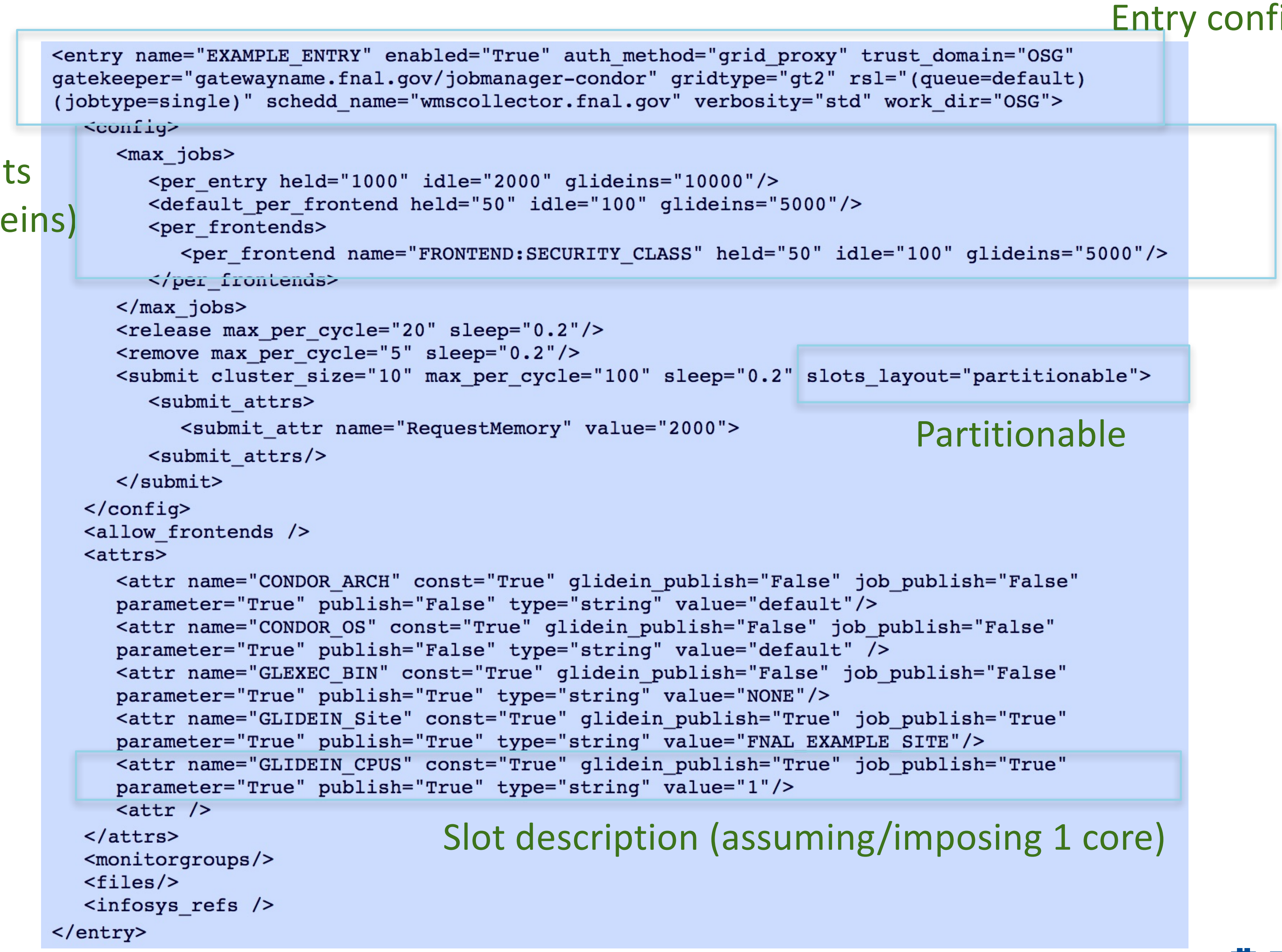
```
<frontend downtimes_file="frontenddowntime" advertise_delay="5" frontend_name="vofrontend-v2_4" loop_delay="60">
   <log_retention >
      <process_logs >
         <process_log extension="info" max_days="7.0" max_mbytes="100.0" min_days="3.0" msg_types="INFO" backup_count="5" compression="gz" />
         <process_log extension="debug" max_days="7.0" max_mbytes="100.0" min_days="3.0" msg_types="DEBUG,ERR,WARN" backup_count="5" />
      </process_logs >
   </log_retention >
   <match match_expr="True" start_expr="True" policy_file="/path/to/python-policy-file">
      <factory query_expr="True">
         <match_attrs />
         <collectors>
            <collector DN="/DC=org/DC=doegrids/OU=Services/CN=factory-server.fnal.gov" comment="" factory_identity="factoryuser@factory-
            server.fnal.gov" my_identity="frontenduser@frontend-server.fnal.gov" node="factory-server.fnal.gov:8618" />
         </collectors>
      </factory>
      <job comment="" query_expr="(JobUniverse==5)&&(GLIDEIN_Is_Monitor =!= TRUE)&&(JOB_Is_Monitor =!= TRUE)">
         <match_attrs />
         <schedds>
            <schedd DN="/DC=org/DC=doegrids/OU=Services/CN=userpool.fnal.gov" fullname="userpool.fnal.gov" />
            <schedd DN="/DC=org/DC=doegrids/OU=Services/CN=userpool.fnal.gov" fullname="schedd_jobs1@userpool.fnal.gov" />
            <schedd DN="/DC=org/DC=doegrids/OU=Services/CN=userpool.fnal.gov" fullname="schedd_jobs2@userpool.fnal.gov" />
         </schedds>
      </job>
   </match>

   <monitor base_dir="/var/www/html/vofrontend/monitor" flot_dir="/opt/javascriptrrd-0.6.3/flot" javascriptRRD_dir="/opt/javascriptrrd-0.6.3/src/lib"
   jquery_dir="/opt/javascriptrrd-0.6.3/flot" />
   <monitor_footer display_txt="Legal Disclaimer" href_link="/site/disclaimer.html" />

   <security classad_proxy="/etc/grid-security/vocert.pem" proxy_DN="/DC=org/DC=doegrids/OU=Services/CN=frontend-server.fnal.gov"
   proxy_selection_plugin="ProxyAll" security_name="frontenduser" sym_key="aes_256_cbc">
      <credentials>
         <credential absfname="/tmp/x509up_u" security_class="frontend" trust_domain="OSG" type="grid_proxy" vm_id="123" vm_type="type1"
         pool_idx_len="5" pool_idx_list="2,4-6,10" />
      </credentials>
      </security>
   <stage base_dir="/var/www/html/vofrontend/stage" use_symlink="True" web_base_url="http://frontend-server.fnal.gov:9000/vofrontend/stage" />
   <work base_dir="/opt/vofrontend" base_log_dir="/opt/vofrontend/logs" />
   <attrs>
      <attr name="GLIDECLIENT_Rank" glidein_publish="False" job_publish="False " parameter="True" type="string" value="1" />
      <attr name="GLIDECLIENT_Start" glidein_publish="False" job_publish="False" parameter="True" type="string" value="True" />
      <attr name="GLIDEIN_Expose_Grid_Env" glidein_publish="True" job_publish="True" parameter="False" type="string" value="True" />
      <attr name="GLIDEIN_Glexec_Use" glidein_publish="True" job_publish="True" parameter="False" type="string" value="OPTIONAL" />
      <attr name="USE_MATCH_AUTH" glidein_publish="False" job_publish="False" parameter="True" type="string" value="True" />
   </attrs>
   <groups>
      <group name="main" enabled="True">
         <config>
```

WMS Collector

Available schedds

Security credentials

Factory configuration example - Entry



Entry configuration

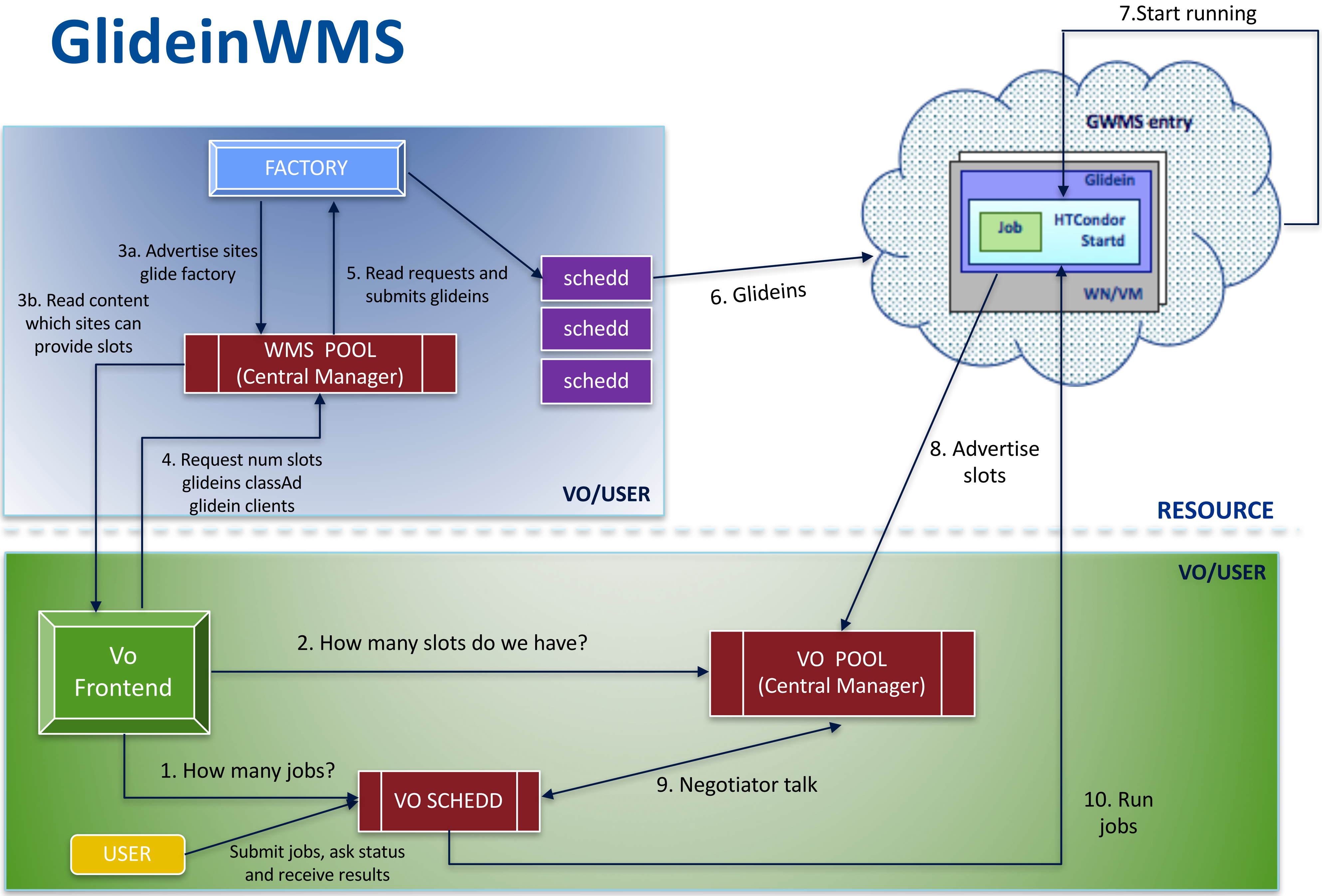
Limits
(in glideins)

```
<entry name="EXAMPLE_ENTRY" enabled="True" auth_method="grid_proxy" trust_domain="OSG"
gatekeeper="gatewayname.fnal.gov/jobmanager-condor" gridtype="gt2" rsl="(queue=default)
(jobtype=single)" schedd_name="wmscollector.fnal.gov" verbosity="std" work_dir="OSG">
   <config>
      <max_jobs>
         <per_entry held="1000" idle="2000" glideins="10000"/>
         <default_per_frontend held="50" idle="100" glideins="5000"/>
         <per_frontends>
            <per_frontend name="FRONTEND:SECURITY_CLASS" held="50" idle="100" glideins="5000"/>
         </per_frontends>
      </max_jobs>
      <release max_per_cycle="20" sleep="0.2"/>
      <remove max_per_cycle="5" sleep="0.2"/>
      <submit cluster_size="10" max_per_cycle="100" sleep="0.2" slots_layout="partitionable">
         <submit_attrs>
            <submit_attr name="RequestMemory" value="2000">
         <submit_attrs/>
      </submit>
   </config>
   <allow_frontends />
   <attrs>
      <attr name="CONDOR_ARCH" const="True" glidein_publish="False" job_publish="False"
      parameter="True" publish="False" type="string" value="default"/>
      <attr name="CONDOR_OS" const="True" glidein_publish="False" job_publish="False"
      parameter="True" publish="False" type="string" value="default" />
      <attr name="GLEXEC_BIN" const="True" glidein_publish="False" job_publish="False"
      parameter="True" publish="True" type="string" value="NONE"/>
      <attr name="GLIDEIN_Site" const="True" glidein_publish="True" job_publish="True"
      parameter="True" publish="True" type="string" value="FNAL EXAMPLE SITE"/>
      <attr name="GLIDEIN_CPUS" const="True" glidein_publish="True" job_publish="True"
      parameter="True" publish="True" type="string" value="1"/>
      <attr />
   </attrs>
   <monitorgroups/>
   <files/>
   <infosys_refs />
</entry>
```

Partitionable

Slot description (assuming/imposing 1 core)

GlideinWMS


7.Start running
GWMS entry
Glidein
Job
HTCondor Startd
WN/VM
FACTORY
3a. Advertise sites glide factory
5. Read requests and submits glideins
schedd
schedd
schedd
6. Glideins
3b. Read content which sites can provide slots
WMS POOL (Central Manager)
4. Request num slots glideins classAd glidein clients
VO/USER
8. Advertise slots
RESOURCE
VO/USER
Vo Frontend
2. How many slots do we have?
VO POOL (Central Manager)
1. How many jobs?
VO SCHEDD
9. Negotiator talk
10. Run jobs
USER
Submit jobs, ask status and receive results

Major GlideinWMS Deployments

- Beta version was called “GlideCAF” in CDF
  - Began testing in 2005
- CMS Global Pool—regularly 200000+ cores
  - Redundant master nodes at CERN and Fermilab
  - Combines production and analysis jobs
- FIFEBATCH / FermiGrid
  - Integrates 18000 on-site cores of FermiGrid with up to 12000 offsite cores.
  - DUNE is using it for standard production and analysis
  - Demonstrated a pool with 2.01 million cores (NOVA 2018)
- Open Science Grid
  - Multi-VO structure shares the same Factory at UCSD
- HEPCloud
  - Portal integrating multiple resources

How it is used?

- Can be used directly
  - HTCondor
- Integrates well in hybrid systems
  - OSG-Connect
  - FermiGrid
- Used by workload/workflow managers:
  - JobSub
  - ProdAgent, CRAB
  - POMS
  - Pegasus
  - HEPCloud

- Scientific computing
- GlideinWMS
- **HTCondor**
- Resources
- Monitoring
- Links
- Demo

HTCondor

- HTCondor is a Workload Management System
  - i.e.: batch system or Local Resource Manager

- Open-source batch system implementation
  - Fault tolerant
  - Robust feature set
  - Flexible
  - Local Center for High Throughput Computing (UW Madison)

HTCondor ClassAds

- HTCondor principles: two parts of the equation
  - Jobs: quanta of work
  - Machines: available resources

- ClassAds is a language for objects (jobs and machines) to
  - Express attributes about themselves
  - Express what they require/desire in a match (similar to personal classified ads)
  - Structure
    - Set of attribute name/value pairs
    - Value : Literals (string, bool, int, float or an expression)

Example Match

Pet Ad

MyType = “Pet”

TargetType = “Buyer”

Requirements =

DogLover =?= True

Rank = 0

PetType = “Dog”

Color = “Brown”

Price = 75

Breed = "Saint Bernard"

Size = "Very Large"

...

Buyer Ad

MyType = “Buyer”

TargetType = “Pet”

Requirements =

(PetType == “Dog”) &&

(TARGET.Price <= MY.AcctBalance)
&&

(Size == "Large"||Size == "Very Large")

Rank = (Breed == "Saint Bernard")

AcctBalance = 100

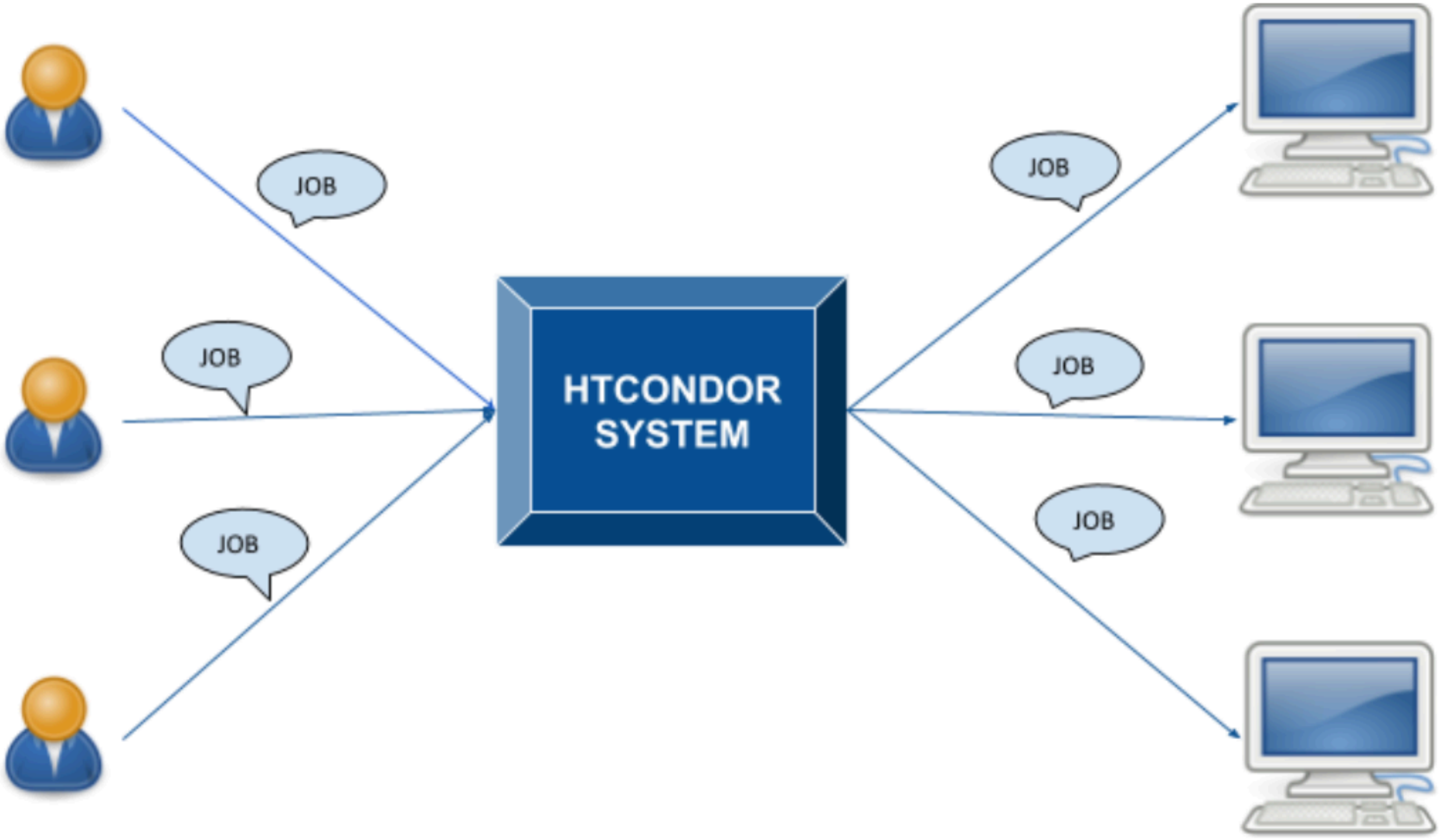
DogLover = True

. . .

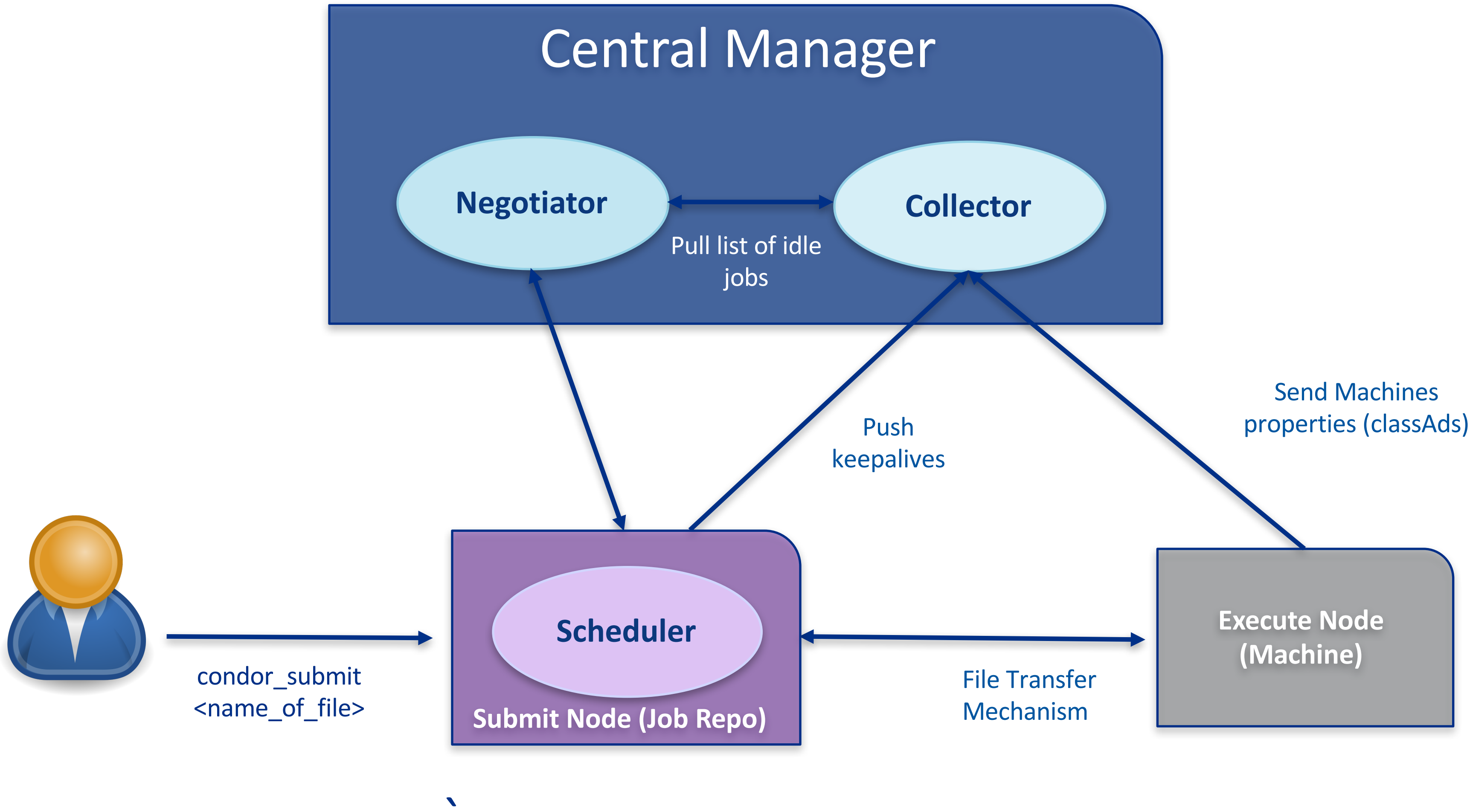
Dog == Resource ~= Machine

Buyer ~= Job

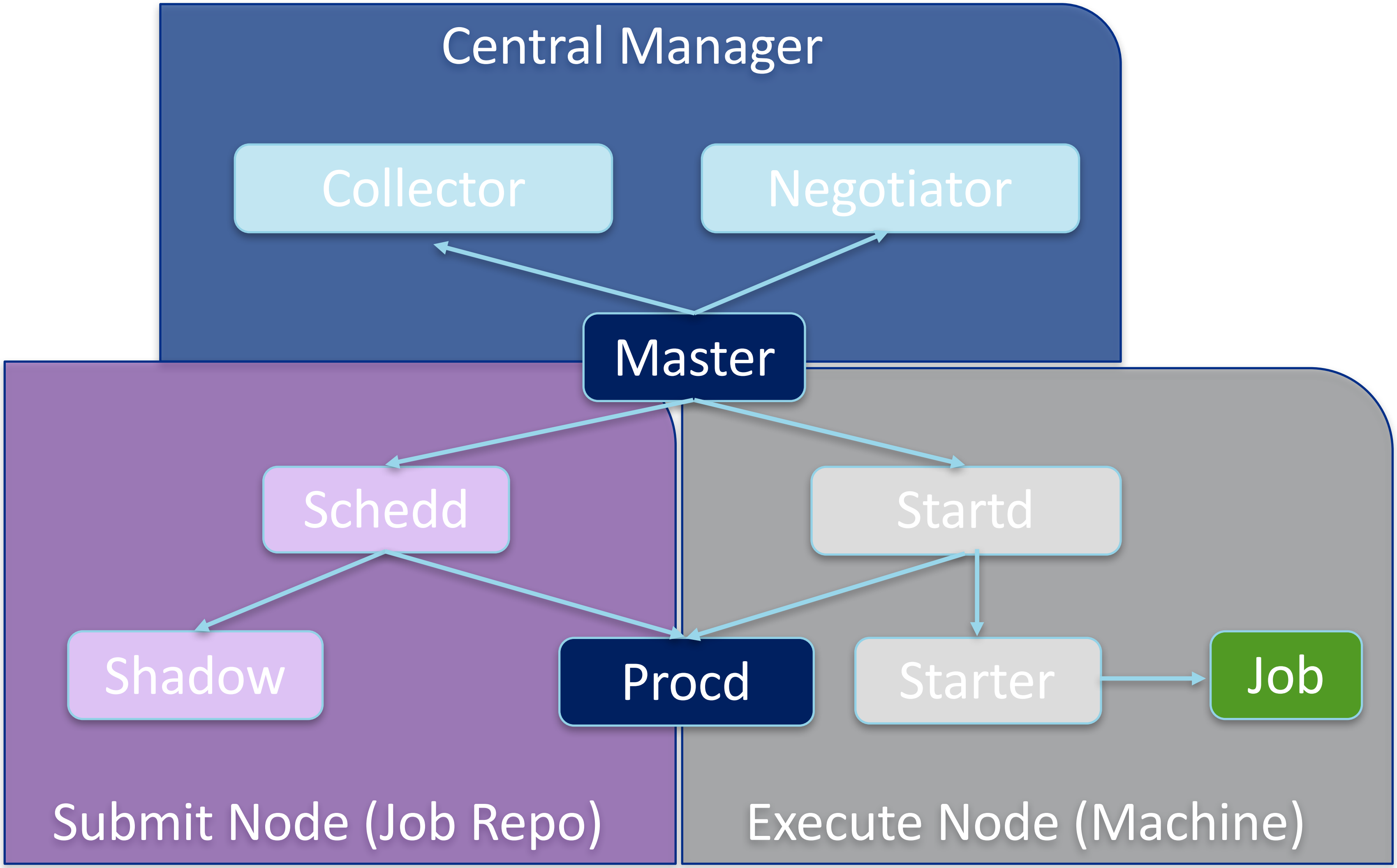
HTCondor Workflow



HTCondor components



HTCondor components (daemons)



- Scientific computing
- GlideinWMS
- HTCondor
- **Resources**
- Monitoring
- Links
- Demo

Glideins run on Execute Nodes

- This is a machine (worker node, host, node, resource), managed by a (Local) Resource Manager
- More frequently virtual than not
- Characterized by its resources (dimensions):
  - CPUs (or total number of cores)
  - RAM (memory)
  - Disk
- There can be other special resources that the node provides: GPUs, access to devices, software, …
- The Glidein will receive all the node or part of it
- Sometime is not easy to identify everything used by a job

Simple (confusing) scenario

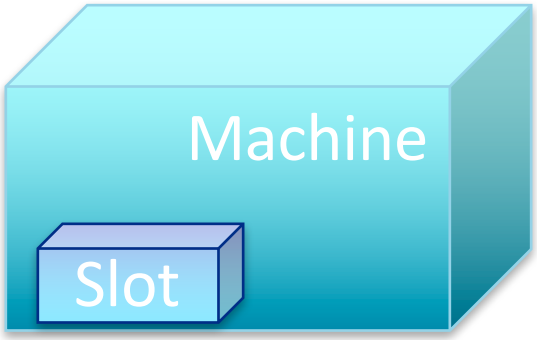
Historically 1 job was running on 1 glidein on 1 worker node using 1 CPU with 1 core.

- Terms got mixed up
- Systems were handling these interchangeably

This is no more

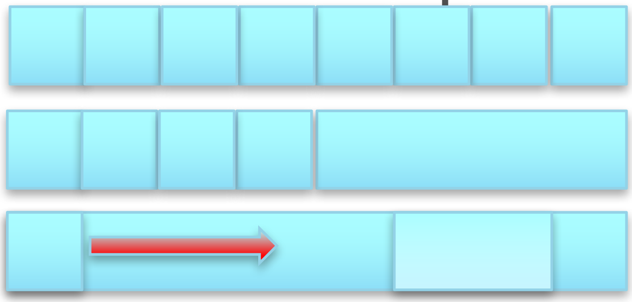
- Systems are more flexible
- We need to know what we want to count or control

Units of work and resources



Terms used by HTCondor

- Job
- Machine (Startd)
- Slot (vm, Starter): multidimensional partition of a machine
  - Static
  - Partitionable
  - Dynamic



Glidein

- Pilot sent on a Machine (or more)
- Allows partitioning policies
- Job for the Factory

Job and Machine ‘dimensions’

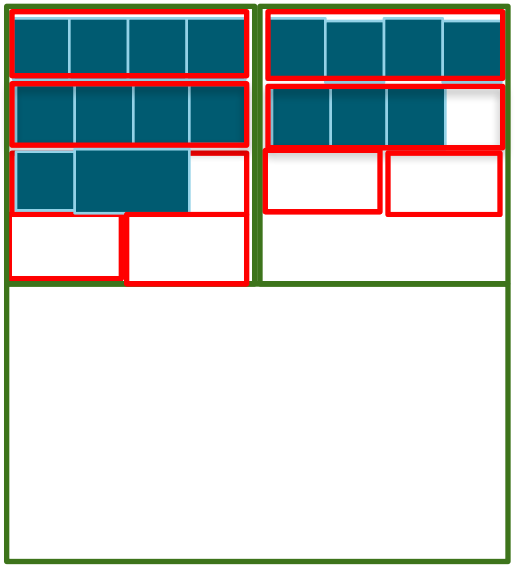
- Job request
  - request_cpus: number of cores, integer, default 1.
  - request_disk: amount of disk space in Kbytes, default to sum of sizes of the job's executable and all input files (or image size)
  - request_memory: amount of memory space in Mbytes, default to executable size
- Machine
  - Cpus: number of cores, integer, by default the available cores
  - Disk: amount of disk space on this machine available for the job in KiB, by default the available space
  - Memory: amount of RAM in MiB in this slot

- Over and Under provision are possible

Partitioning in an overlay system

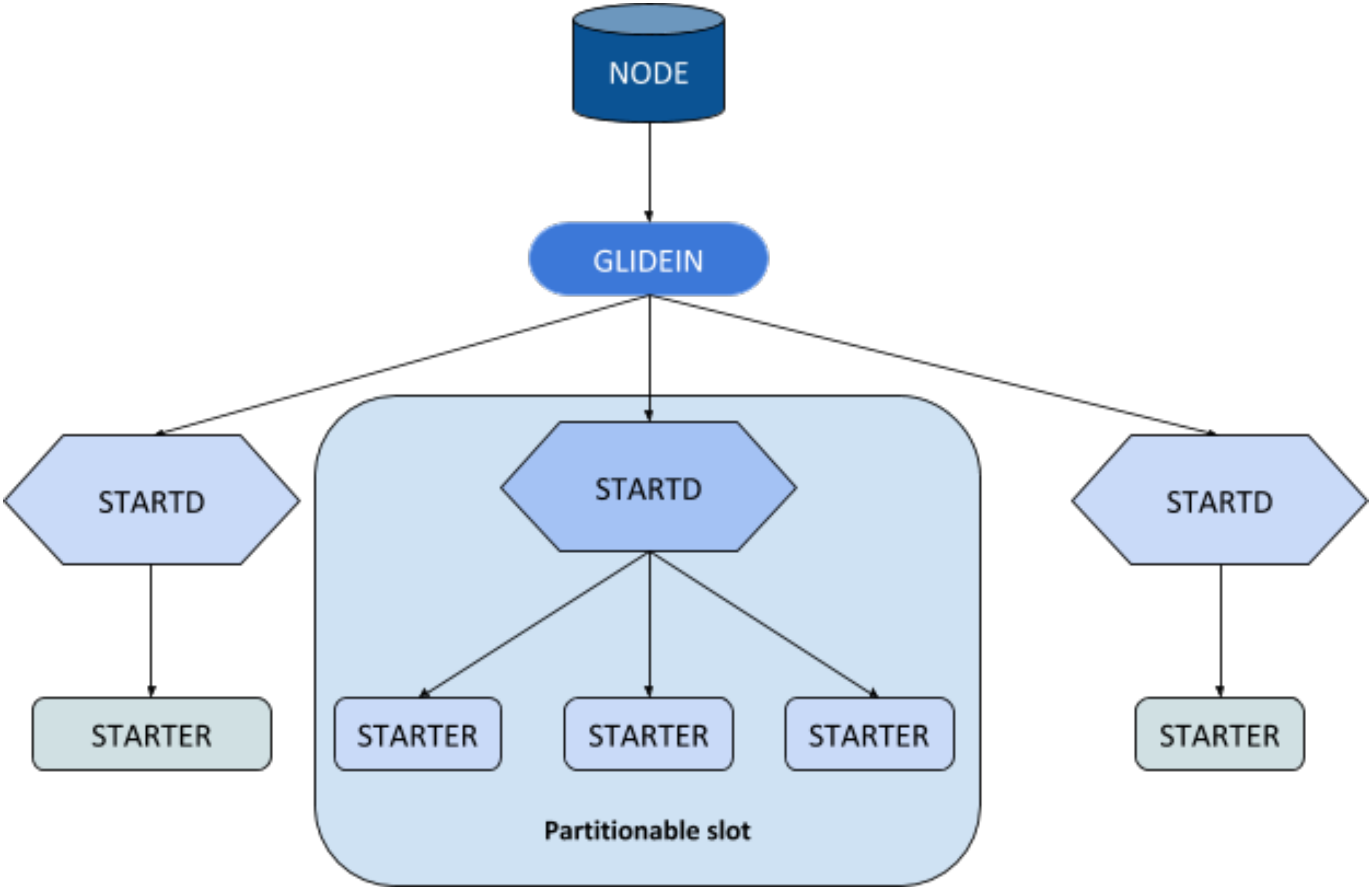
- Dimensions: Cores, Memory, Disk, Lifetime
- The resource (e.g. GPGrid) partitions its Execute nodes
- GlideinWMS further partitions the resources it receives

- E.g. 64 Cores machine split in 16 or 32 cores cluster slots; 16 or 12 cores Glideins in 4 or 2 cores partitionable slots; 2 or 1 core jobs



- Issues
  - Fragmentation (unused)
  - Flexibility (vs Complexity)
  - Under or over provisioning (overbooking or be prudent)
  - Scaling (big slots, fewer slices)

Units of work and resources



- Scientific computing
- GlideinWMS
- HTCondor
- Resources
- **Monitoring**
- Links
- Demo

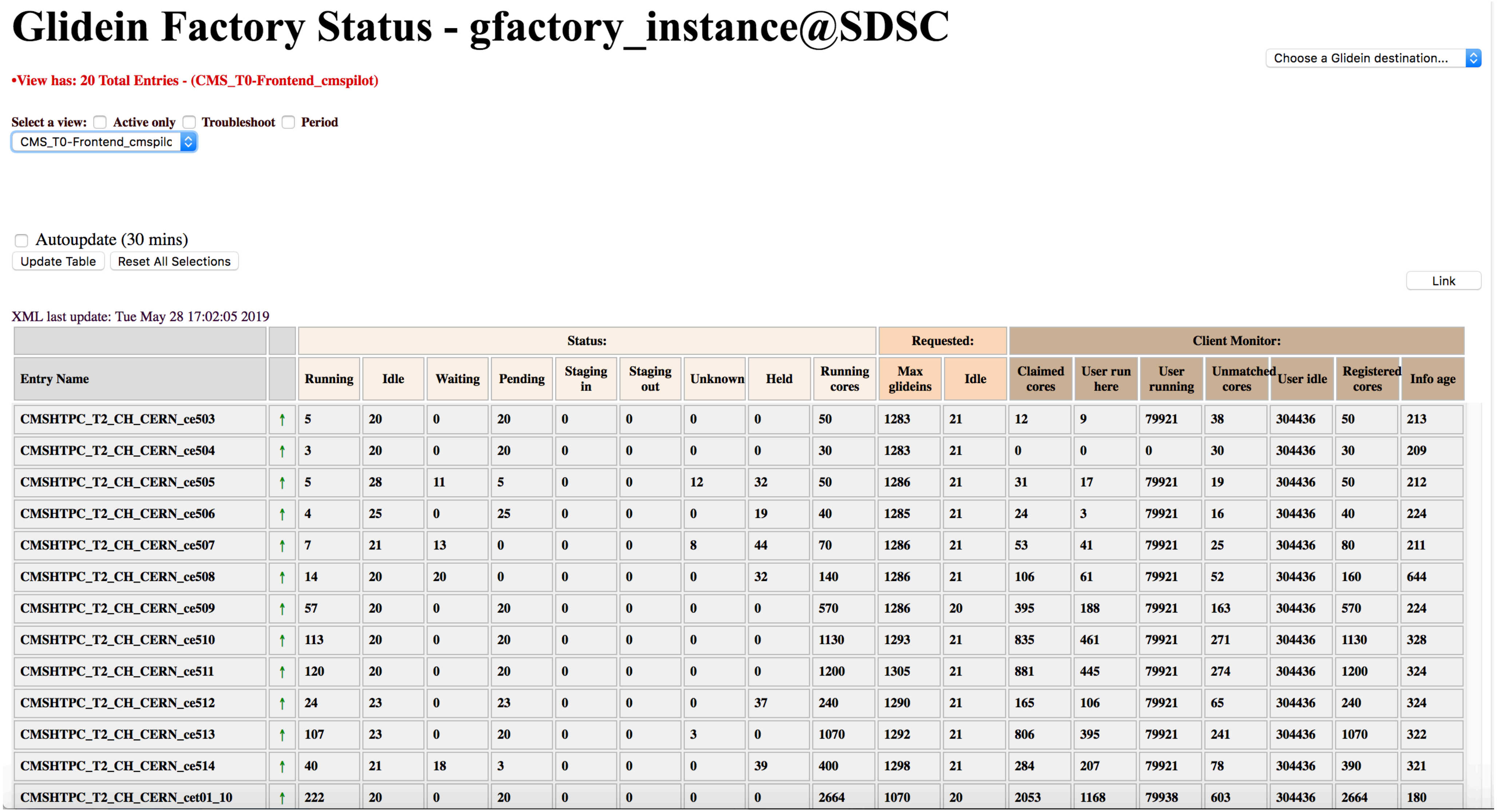
Why monitoring is so important?

- Something may break, what?
- CE
  - May refuse to accept glideins
  - May not start glideins
  - Fail to tell us what the status of the job is
- The worker node may be broken/misconfigured (validation fails)
- Networking may not work properly
- Central Manager never hears from the Startd
- Schedd cannot talk to Startd
- Security infrastructure could be broken (CAs missing)
- Jobs not matching

Monitoring resources

- Logs
  - Frontend
  - Factory
  - HTCondor
  - CE
  - Glidein
- ClassAds
- GlideinWMS monitoring
- Monitoring FIFEMON

GlideinWMS monitoring



Glidein Factory Status - gfactory_instance@SDSC

•View has: 20 Total Entries - (CMS_T0-Frontend_cmspilot)

Select a view: □ Active only □ Troubleshoot □ Period

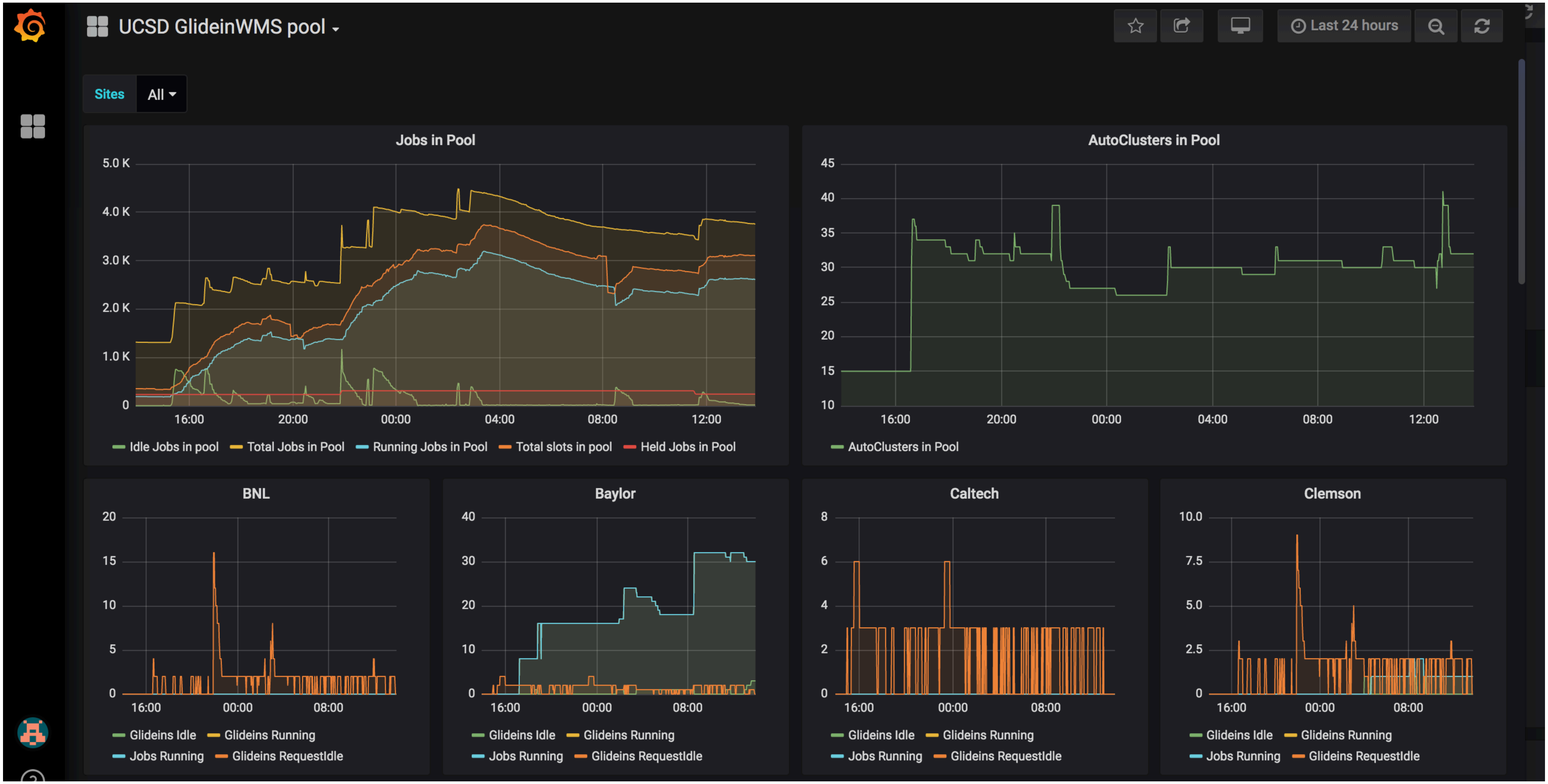
CMS_T0-Frontend_cmspilc

□ Autoupdate (30 mins)

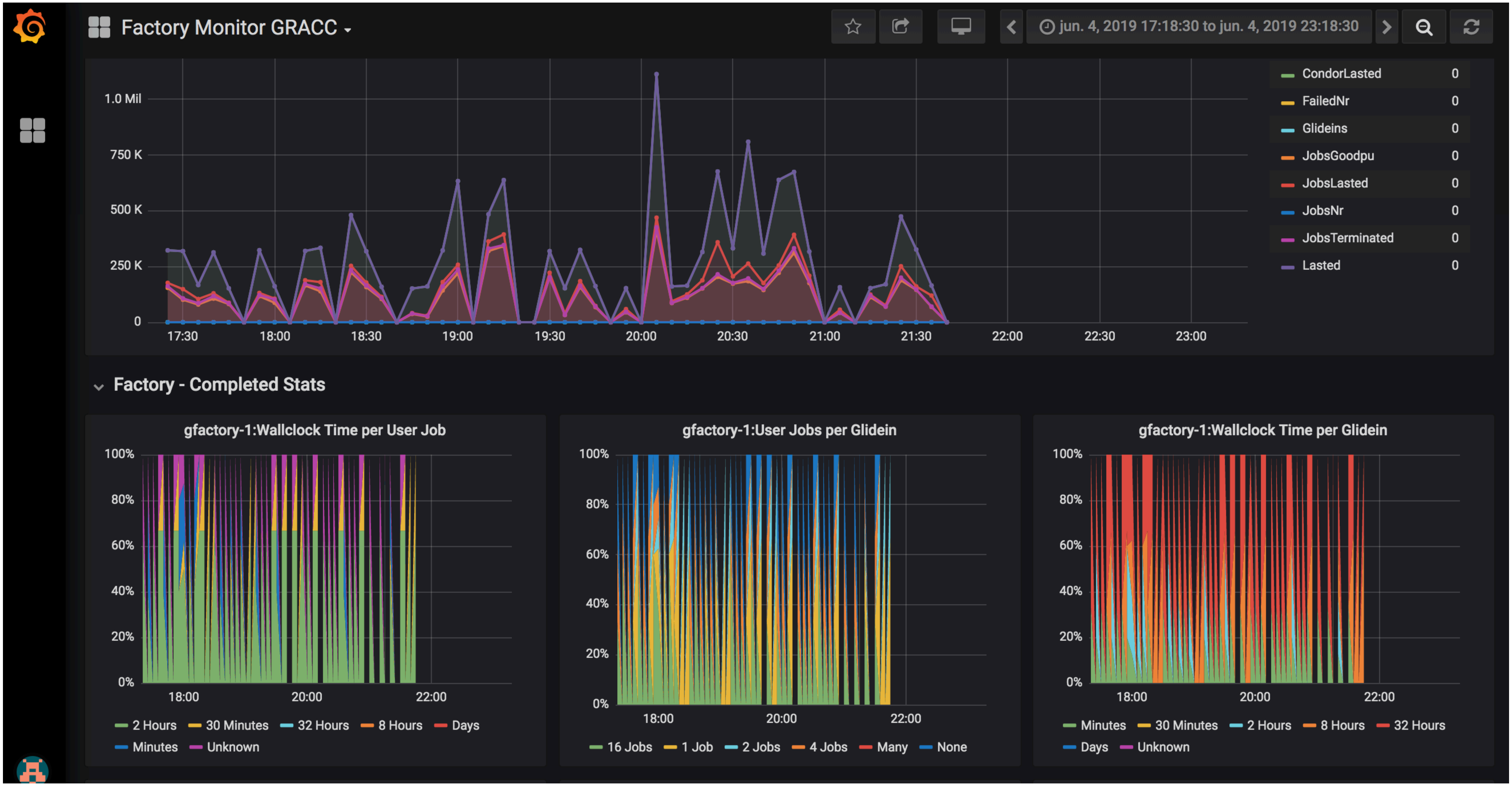
XML last update: Tue May 28 17:02:05 2019

		Status:									Requested:		Client Monitor:						
Entry Name		Running	Idle	Waiting	Pending	Staging in	Staging out	Unknown	Held	Running cores	Max glideins	Idle	Claimed cores	User run here	User running	Unmatched cores	User idle	Registered cores	Info age
CMSHTPC_T2_CH_CERN_ce503	↑	5	20	0	20	0	0	0	0	50	1283	21	12	9	79921	38	304436	50	213
CMSHTPC_T2_CH_CERN_ce504	↑	3	20	0	20	0	0	0	0	30	1283	21	0	0	0	30	304436	30	209
CMSHTPC_T2_CH_CERN_ce505	↑	5	28	11	5	0	0	12	32	50	1286	21	31	17	79921	19	304436	50	212
CMSHTPC_T2_CH_CERN_ce506	↑	4	25	0	25	0	0	0	19	40	1285	21	24	3	79921	16	304436	40	224
CMSHTPC_T2_CH_CERN_ce507	↑	7	21	13	0	0	0	8	44	70	1286	21	53	41	79921	25	304436	80	211
CMSHTPC_T2_CH_CERN_ce508	↑	14	20	20	0	0	0	0	32	140	1286	21	106	61	79921	52	304436	160	644
CMSHTPC_T2_CH_CERN_ce509	↑	57	20	0	20	0	0	0	0	570	1286	20	395	188	79921	163	304436	570	224
CMSHTPC_T2_CH_CERN_ce510	↑	113	20	0	20	0	0	0	0	1130	1293	21	835	461	79921	271	304436	1130	328
CMSHTPC_T2_CH_CERN_ce511	↑	120	20	0	20	0	0	0	0	1200	1305	21	881	445	79921	274	304436	1200	324
CMSHTPC_T2_CH_CERN_ce512	↑	24	23	0	23	0	0	0	37	240	1290	21	165	106	79921	65	304436	240	324
CMSHTPC_T2_CH_CERN_ce513	↑	107	23	0	20	0	0	3	0	1070	1292	21	806	395	79921	241	304436	1070	322
CMSHTPC_T2_CH_CERN_ce514	↑	40	21	18	3	0	0	0	39	400	1298	21	284	207	79921	78	304436	390	321
CMSHTPC_T2_CH_CERN_cet01_10	↑	222	20	0	20	0	0	0	0	2664	1070	20	2053	1168	79938	603	304436	2664	180

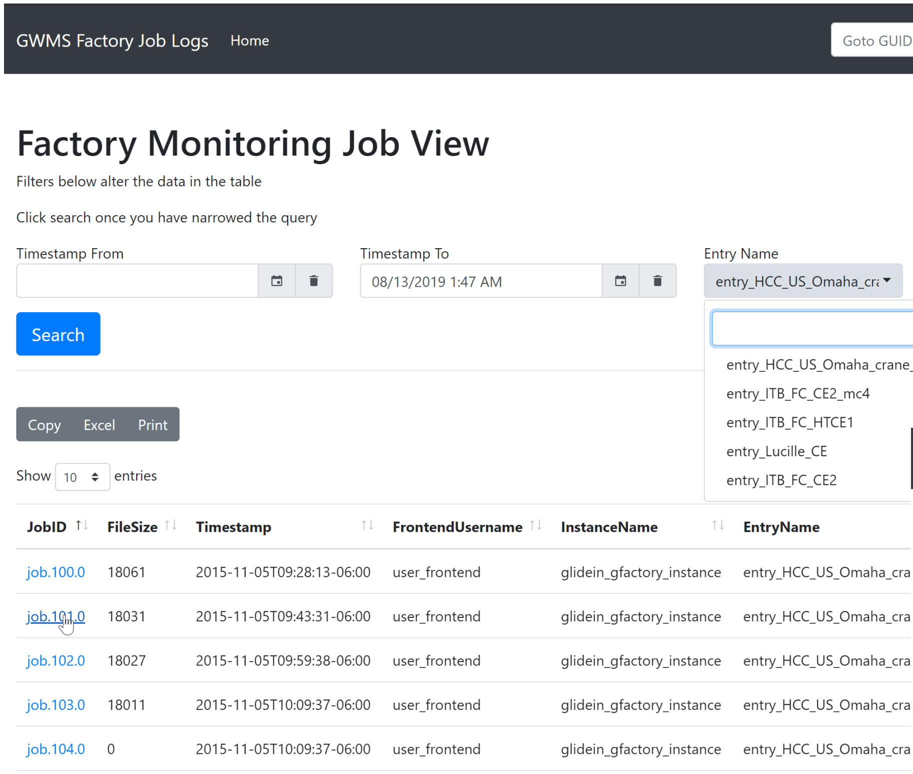
GlideinWMS monitoring - Kibana



GlideinWMS monitoring - GRACC and Graphana



GlideinMonitor



Factory Monitoring Job View

Filters below alter the data in the table

Click search once you have narrowed the query

Timestamp From:

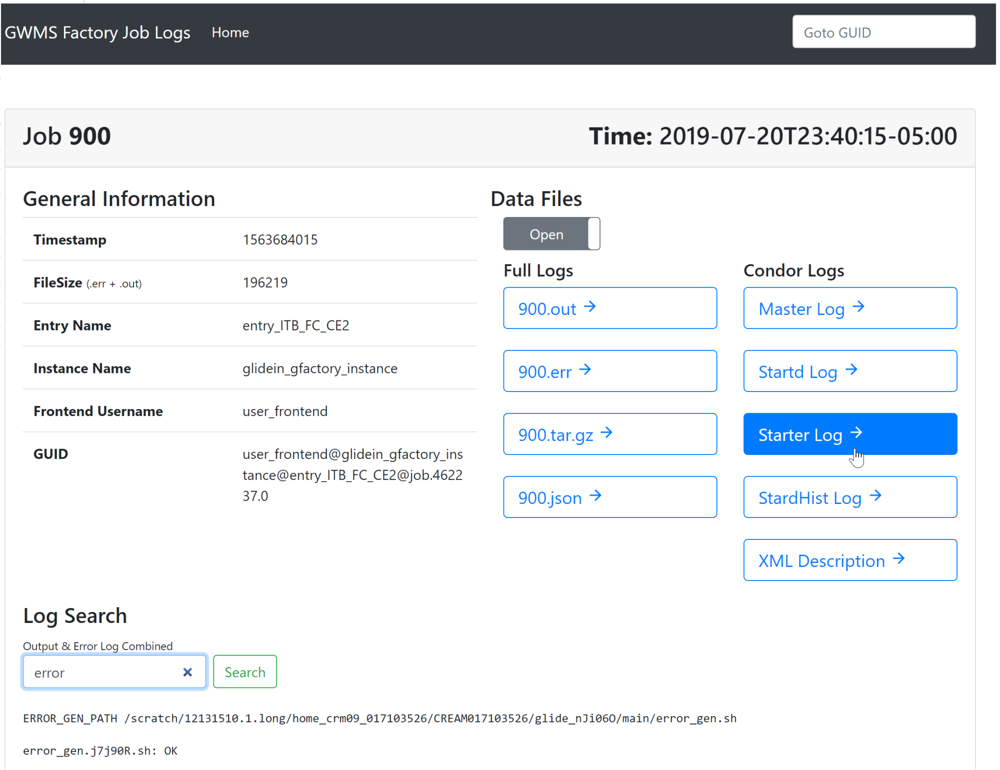
Timestamp To: 08/13/2019 1:47 AM

Entry Name: entry_HCC_US_Omaha_cra

- entry_HCC_US_Omaha_crane_gpu ✓
- entry_ITB_FC_CE2_mc4 ✓
- entry_ITB_FC_HTCE1
- entry_Lucille_CE
- entry_ITB_FC_CE2

Show 10 entries

JobID	FileSize	Timestamp	FrontendUsername	InstanceName	EntryName
job.100.0	18061	2015-11-05T09:28:13-06:00	user_frontend	glidein_gfactory_instance	entry_HCC_US_Omaha_cra
job.101.0	18031	2015-11-05T09:43:31-06:00	user_frontend	glidein_gfactory_instance	entry_HCC_US_Omaha_cra
job.102.0	18027	2015-11-05T09:59:38-06:00	user_frontend	glidein_gfactory_instance	entry_HCC_US_Omaha_cra
job.103.0	18011	2015-11-05T10:09:37-06:00	user_frontend	glidein_gfactory_instance	entry_HCC_US_Omaha_cra
job.104.0	0	2015-11-05T10:09:37-06:00	user_frontend	glidein_gfactory_instance	entry_HCC_US_Omaha_cra



Job 900

Time: 2019-07-20T23:40:15-05:00

General Information

Timestamp	1563684015
FileSize (.err + .out)	196219
Entry Name	entry_ITB_FC_CE2
Instance Name	glidein_gfactory_instance
Frontend Username	user_frontend
GUID	user_frontend@glidein_gfactory_instance@entry_ITB_FC_CE2@job.462237.0

Data Files

Full Logs

Condor Logs

Log Search

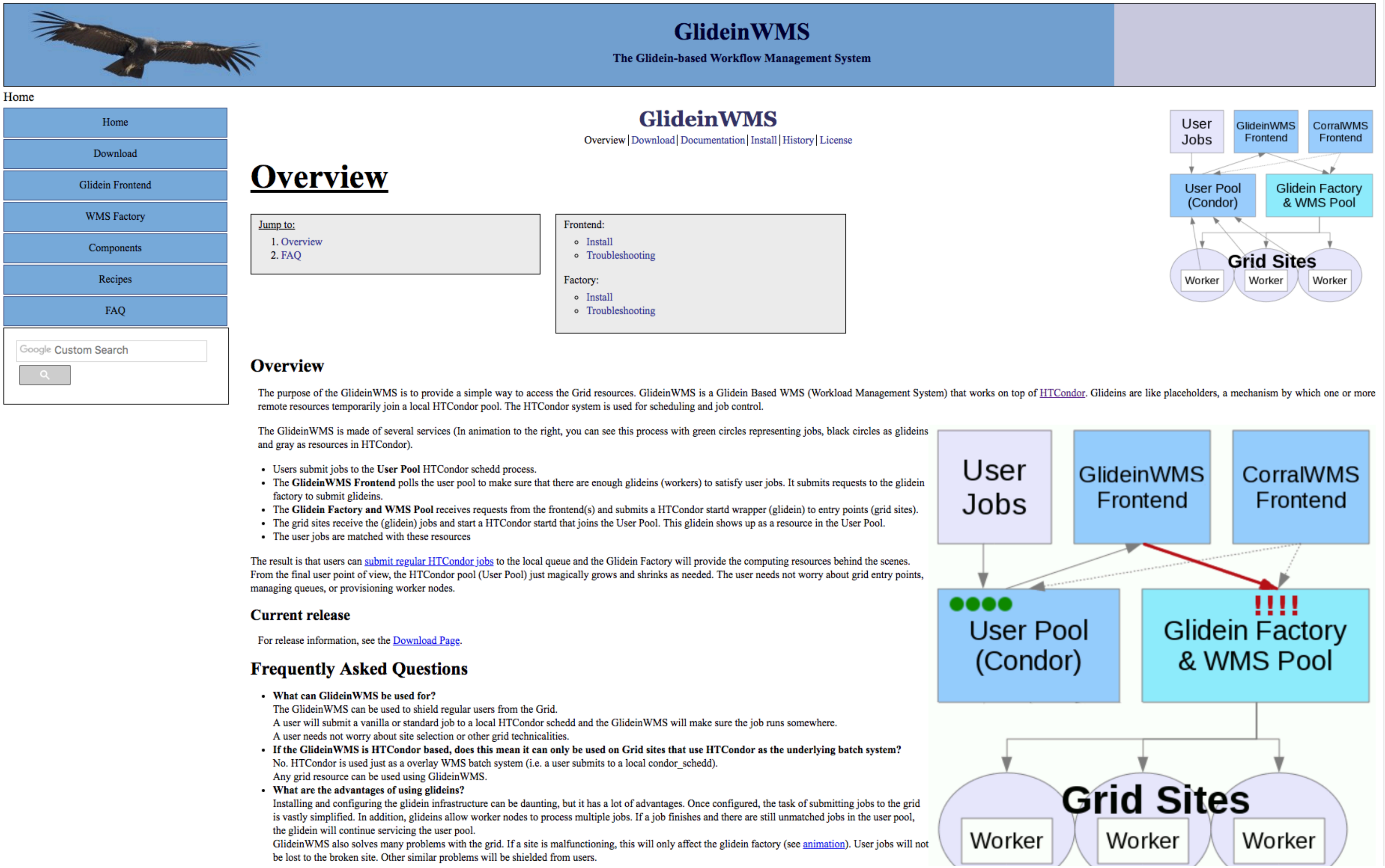
Output & Error Log Combined: error

```
ERROR_GEN_PATH /scratch/12131510.1.long/home_crm09_017103526/CREAM017103526/glide_nJi06O/main/error_gen.sh

error_gen.j7j90R.sh: OK
```

- Scientific computing
- GlideinWMS
- HTCondor
- Resources
- Monitoring
- **Links**
- Demo

Links – GlideinWMS Website

glideinwms.fnal.gov



GlideinWMS

The Glidein-based Workflow Management System

Home

GlideinWMS

Overview | Download | Documentation | Install | History | License

Overview

Jump to:

1. Overview
2. FAQ

Frontend:

- Install
- Troubleshooting

Factory:

- Install
- Troubleshooting

Overview

The purpose of the GlideinWMS is to provide a simple way to access the Grid resources. GlideinWMS is a Glidein Based WMS (Workload Management System) that works on top of HTCondor. Glideins are like placeholders, a mechanism by which one or more remote resources temporarily join a local HTCondor pool. The HTCondor system is used for scheduling and job control.

The GlideinWMS is made of several services (In animation to the right, you can see this process with green circles representing jobs, black circles as glideins and gray as resources in HTCondor).

- Users submit jobs to the **User Pool** HTCondor schedd process.
- The **GlideinWMS Frontend** polls the user pool to make sure that there are enough glideins (workers) to satisfy user jobs. It submits requests to the glidein factory to submit glideins.
- The **Glidein Factory and WMS Pool** receives requests from the frontend(s) and submits a HTCondor startd wrapper (glidein) to entry points (grid sites).
- The grid sites receive the (glidein) jobs and start a HTCondor startd that joins the User Pool. This glidein shows up as a resource in the User Pool.
- The user jobs are matched with these resources

The result is that users can submit regular HTCondor jobs to the local queue and the Glidein Factory will provide the computing resources behind the scenes. From the final user point of view, the HTCondor pool (User Pool) just magically grows and shrinks as needed. The user needs not worry about grid entry points, managing queues, or provisioning worker nodes.

Current release

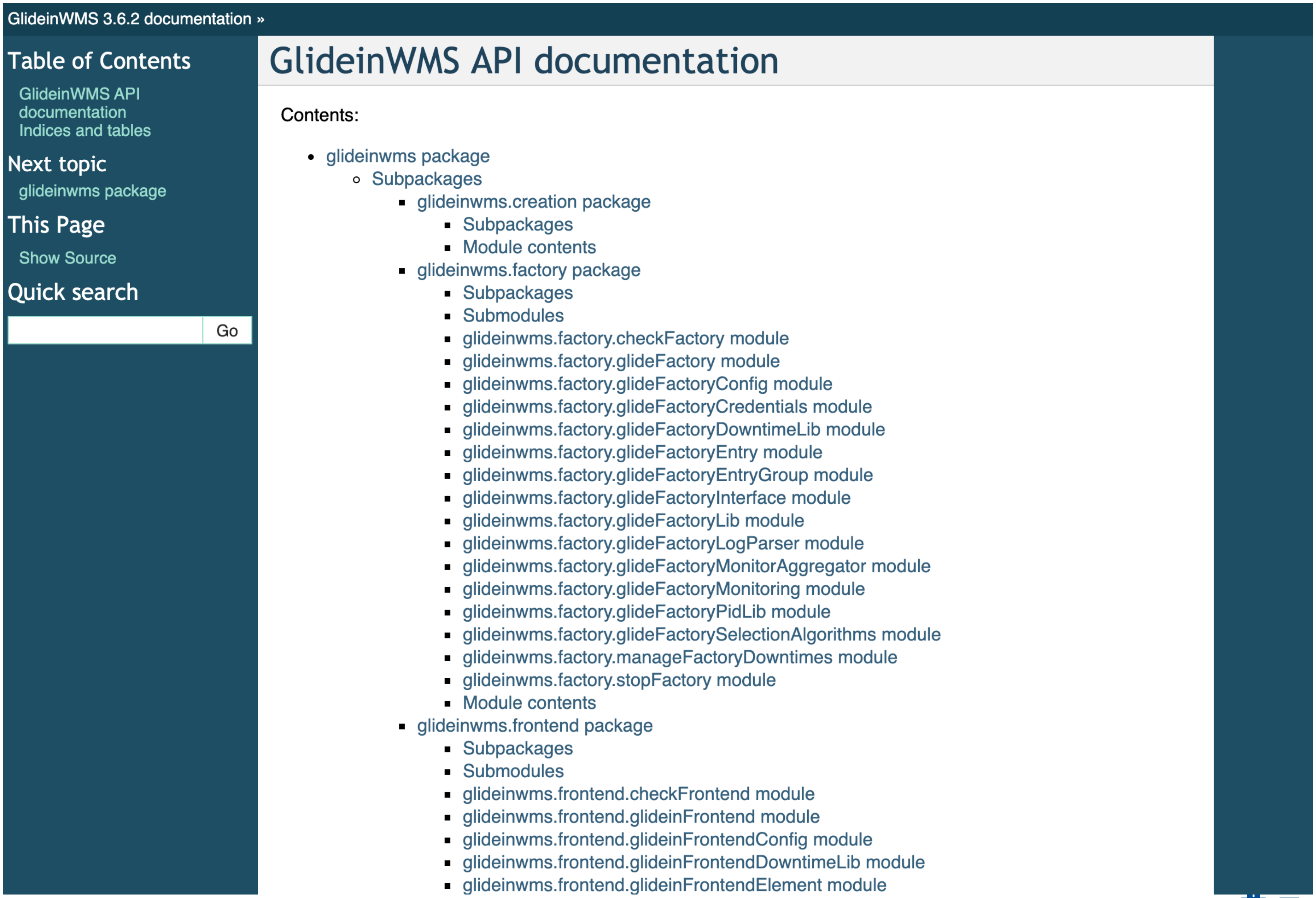
For release information, see the Download Page.

Frequently Asked Questions

- **What can GlideinWMS be used for?**
  The GlideinWMS can be used to shield regular users from the Grid.
  A user will submit a vanilla or standard job to a local HTCondor schedd and the GlideinWMS will make sure the job runs somewhere.
  A user needs not worry about site selection or other grid technicalities.
- **If the GlideinWMS is HTCondor based, does this mean it can only be used on Grid sites that use HTCondor as the underlying batch system?**
  No. HTCondor is used just as a overlay WMS batch system (i.e. a user submits to a local condor_schedd).
  Any grid resource can be used using GlideinWMS.
- **What are the advantages of using glideins?**
  Installing and configuring the glidein infrastructure can be daunting, but it has a lot of advantages. Once configured, the task of submitting jobs to the grid is vastly simplified. In addition, glideins allow worker nodes to process multiple jobs. If a job finishes and there are still unmatched jobs in the user pool, the glidein will continue servicing the user pool.
  GlideinWMS also solves many problems with the grid. If a site is malfunctioning, this will only affect the glidein factory (see animation). User jobs will not be lost to the broken site. Other similar problems will be shielded from users.

Links – GlideinWMS API

glideinwms.fnal.gov/api/



GlideinWMS 3.6.2 documentation »

Table of Contents

GlideinWMS API documentation
Indices and tables

Next topic

glideinwms package

This Page

Show Source

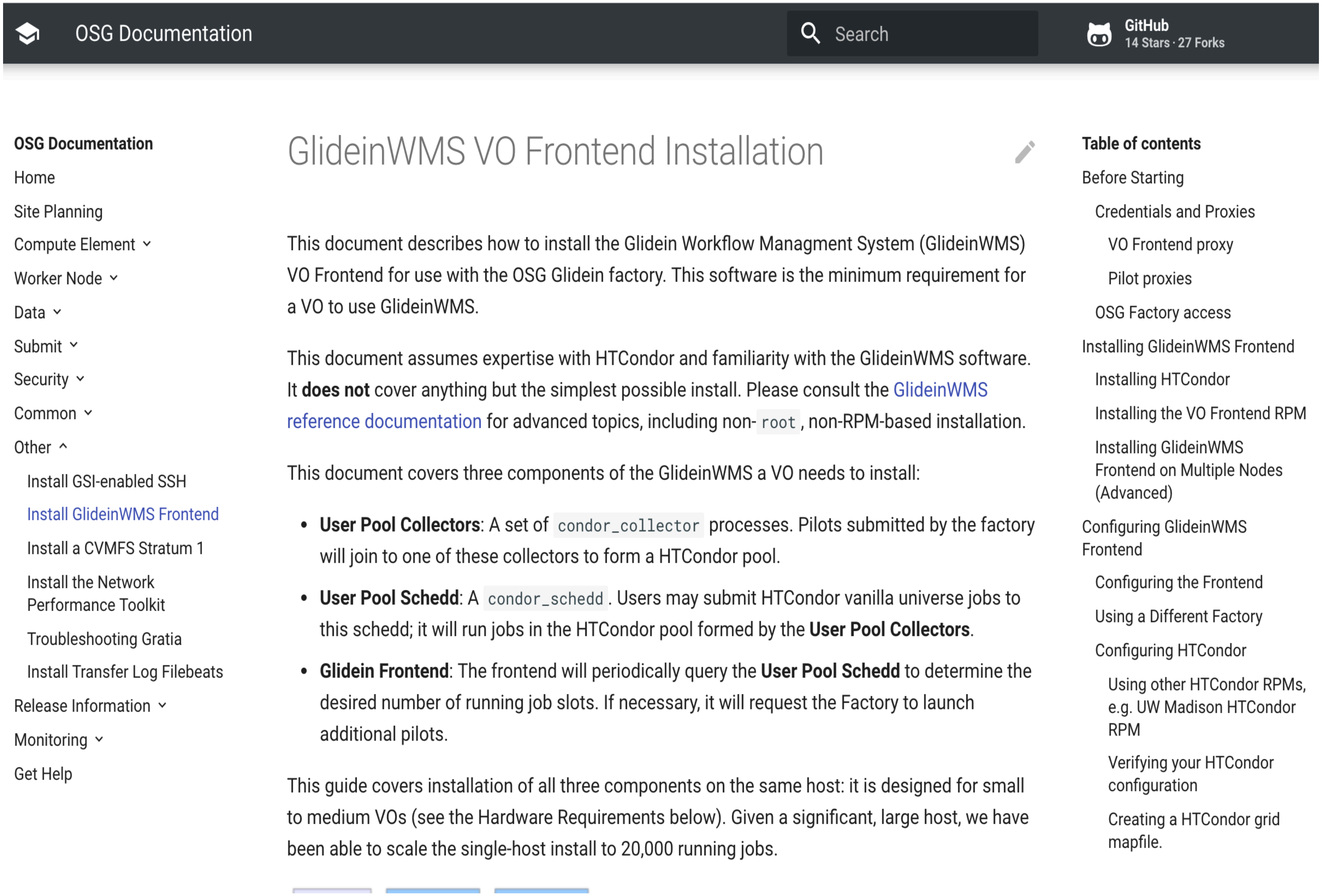
Quick search

GlideinWMS API documentation

Contents:

Links – OSG Docs: install

https://opensciencegrid.org/docs/other/install-gwms-frontend/



OSG Documentation

- Home
- Site Planning
- Compute Element
- Worker Node
- Data
- Submit
- Security
- Common
- Other
  - Install GSI-enabled SSH
  - Install GlideinWMS Frontend
  - Install a CVMFS Stratum 1
  - Install the Network Performance Toolkit
  - Troubleshooting Gratia
  - Install Transfer Log Filebeats
- Release Information
- Monitoring
- Get Help

GlideinWMS VO Frontend Installation

This document describes how to install the Glidein Workflow Managment System (GlideinWMS) VO Frontend for use with the OSG Glidein factory. This software is the minimum requirement for a VO to use GlideinWMS.

This document assumes expertise with HTCondor and familiarity with the GlideinWMS software. It **does not** cover anything but the simplest possible install. Please consult the GlideinWMS reference documentation for advanced topics, including non- `root` , non-RPM-based installation.

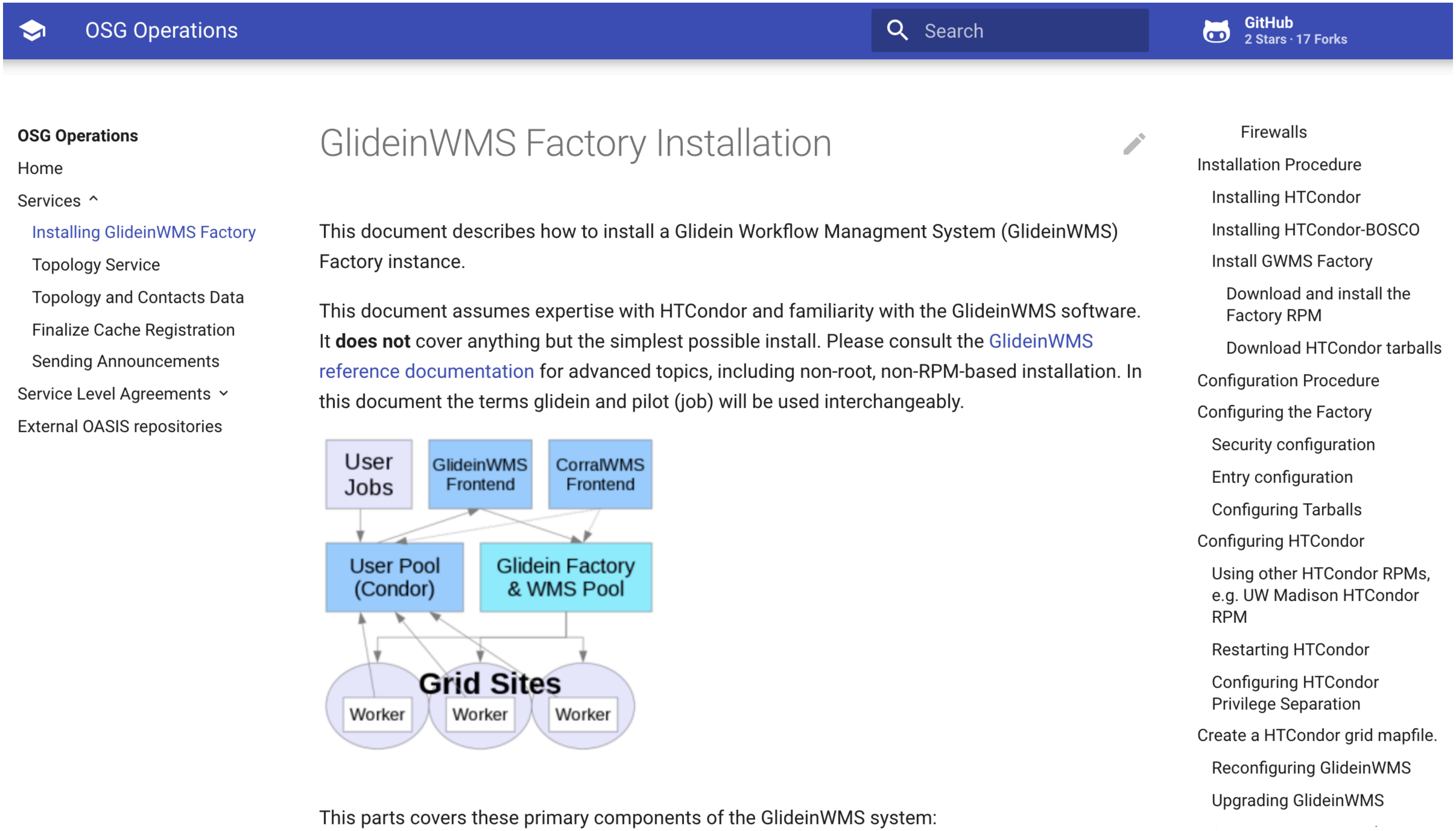
This document covers three components of the GlideinWMS a VO needs to install:

- **User Pool Collectors**: A set of `condor_collector` processes. Pilots submitted by the factory will join to one of these collectors to form a HTCondor pool.
- **User Pool Schedd**: A `condor_schedd` . Users may submit HTCondor vanilla universe jobs to this schedd; it will run jobs in the HTCondor pool formed by the **User Pool Collectors**.
- **Glidein Frontend**: The frontend will periodically query the **User Pool Schedd** to determine the desired number of running job slots. If necessary, it will request the Factory to launch additional pilots.

This guide covers installation of all three components on the same host: it is designed for small to medium VOs (see the Hardware Requirements below). Given a significant, large host, we have been able to scale the single-host install to 20,000 running jobs.

Links – OSG Docs: install

https://opensciencegrid.org/operations/services/install-gwms-factory/



OSG Operations

- Home
- Services ⌃
  - Installing GlideinWMS Factory
  - Topology Service
  - Topology and Contacts Data
  - Finalize Cache Registration
  - Sending Announcements
- Service Level Agreements ⌄
- External OASIS repositories

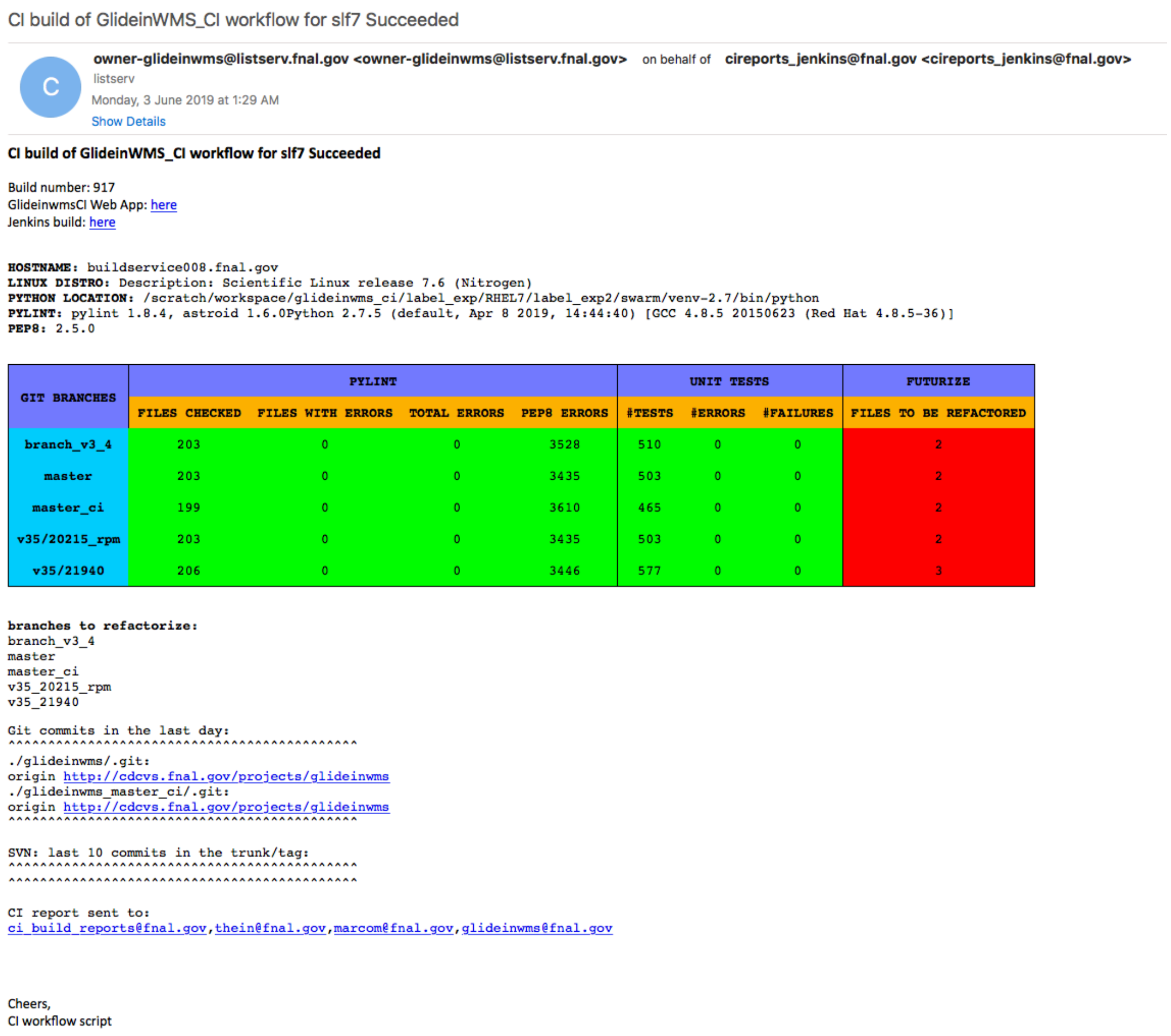
GlideinWMS Factory Installation

This document describes how to install a Glidein Workflow Managment System (GlideinWMS) Factory instance.

This document assumes expertise with HTCondor and familiarity with the GlideinWMS software. It **does not** cover anything but the simplest possible install. Please consult the GlideinWMS reference documentation for advanced topics, including non-root, non-RPM-based installation. In this document the terms glidein and pilot (job) will be used interchangeably.

This parts covers these primary components of the GlideinWMS system:

Links – Weekly CI email



CI build of GlideinWMS_CI workflow for slf7 Succeeded

owner-glideinwms@listserv.fnal.gov <owner-glideinwms@listserv.fnal.gov> on behalf of **cireports_jenkins@fnal.gov <cireports_jenkins@fnal.gov>**
listserv
Monday, 3 June 2019 at 1:29 AM
Show Details

CI build of GlideinWMS_CI workflow for slf7 Succeeded

Build number: 917
GlideinwmsCI Web App: here
Jenkins build: here

```
HOSTNAME: buildservice008.fnal.gov
LINUX DISTRO: Description: Scientific Linux release 7.6 (Nitrogen)
PYTHON LOCATION: /scratch/workspace/glideinwms_ci/label_exp/RHEL7/label_exp2/swarm/venv-2.7/bin/python
PYLINT: pylint 1.8.4, astroid 1.6.0Python 2.7.5 (default, Apr 8 2019, 14:44:40) [GCC 4.8.5 20150623 (Red Hat 4.8.5-36)]
PEP8: 2.5.0
```

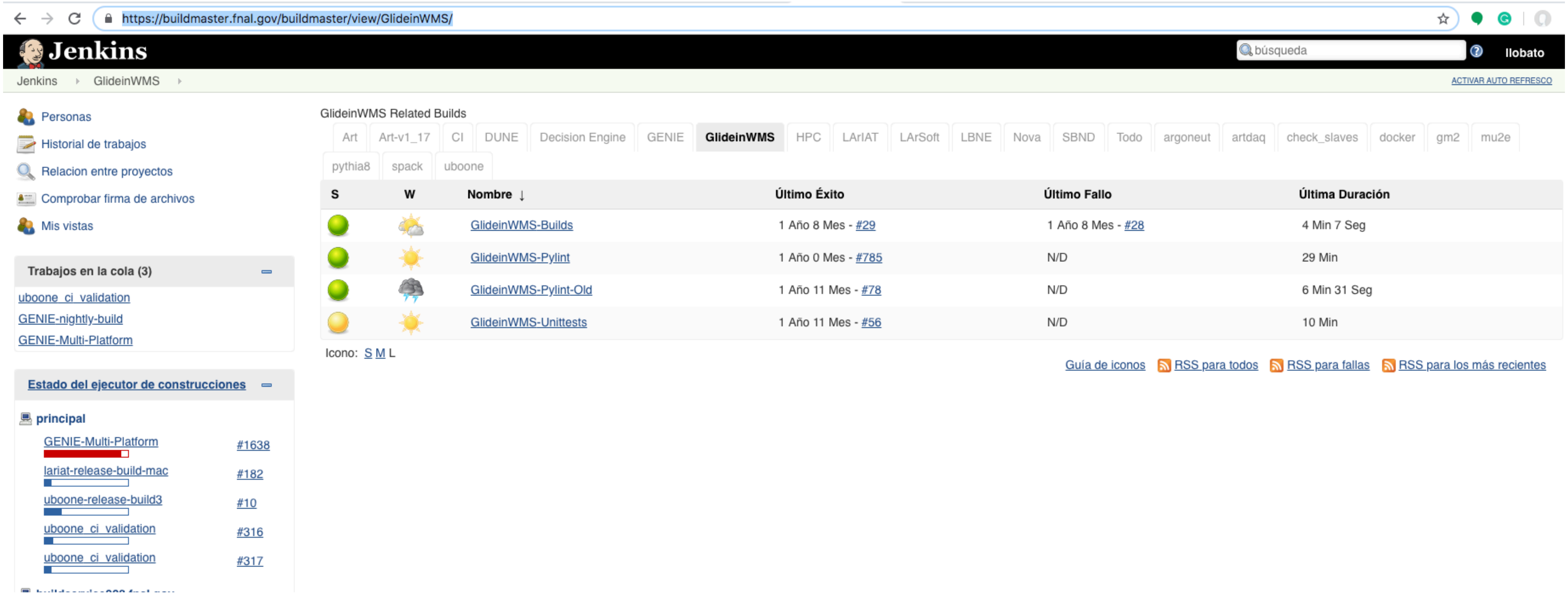
GIT BRANCHES	PYLINT				UNIT TESTS			FUTURIZE
	FILES CHECKED	FILES WITH ERRORS	TOTAL ERRORS	PEP8 ERRORS	#TESTS	#ERRORS	#FAILURES	FILES TO BE REFACTORED
branch_v3_4	203	0	0	3528	510	0	0	2
master	203	0	0	3435	503	0	0	2
master_ci	199	0	0	3610	465	0	0	2
v35/20215_rpm	203	0	0	3435	503	0	0	2
v35/21940	206	0	0	3446	577	0	0	3

```
branches to refactorize:
branch_v3_4
master
master_ci
v35_20215_rpm
v35_21940

Git commits in the last day:
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
./glideinwms/.git:
origin http://cdcvs.fnal.gov/projects/glideinwms
./glideinwms_master_ci/.git:
origin http://cdcvs.fnal.gov/projects/glideinwms
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

SVN: last 10 commits in the trunk/tag:
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

CI report sent to:
ci_build_reports@fnal.gov,thein@fnal.gov,marcom@fnal.gov,glideinwms@fnal.gov
```

Cheers,
CI workflow script

Links – Weekly CI email (cont)

https://buildmaster.fnal.gov/buildmaster/view/GlideinWMS/



https://buildmaster.fnal.gov/buildmaster/view/GlideinWMS/

Trabajos en la cola (3)

- uboone_ci_validation
- GENIE-nightly-build
- GENIE-Multi-Platform

Estado del ejecutor de construcciones

principal

GENIE-Multi-Platform	#1638
lariat-release-build-mac	#182
uboone-release-build3	#10
uboone_ci_validation	#316
uboone_ci_validation	#317

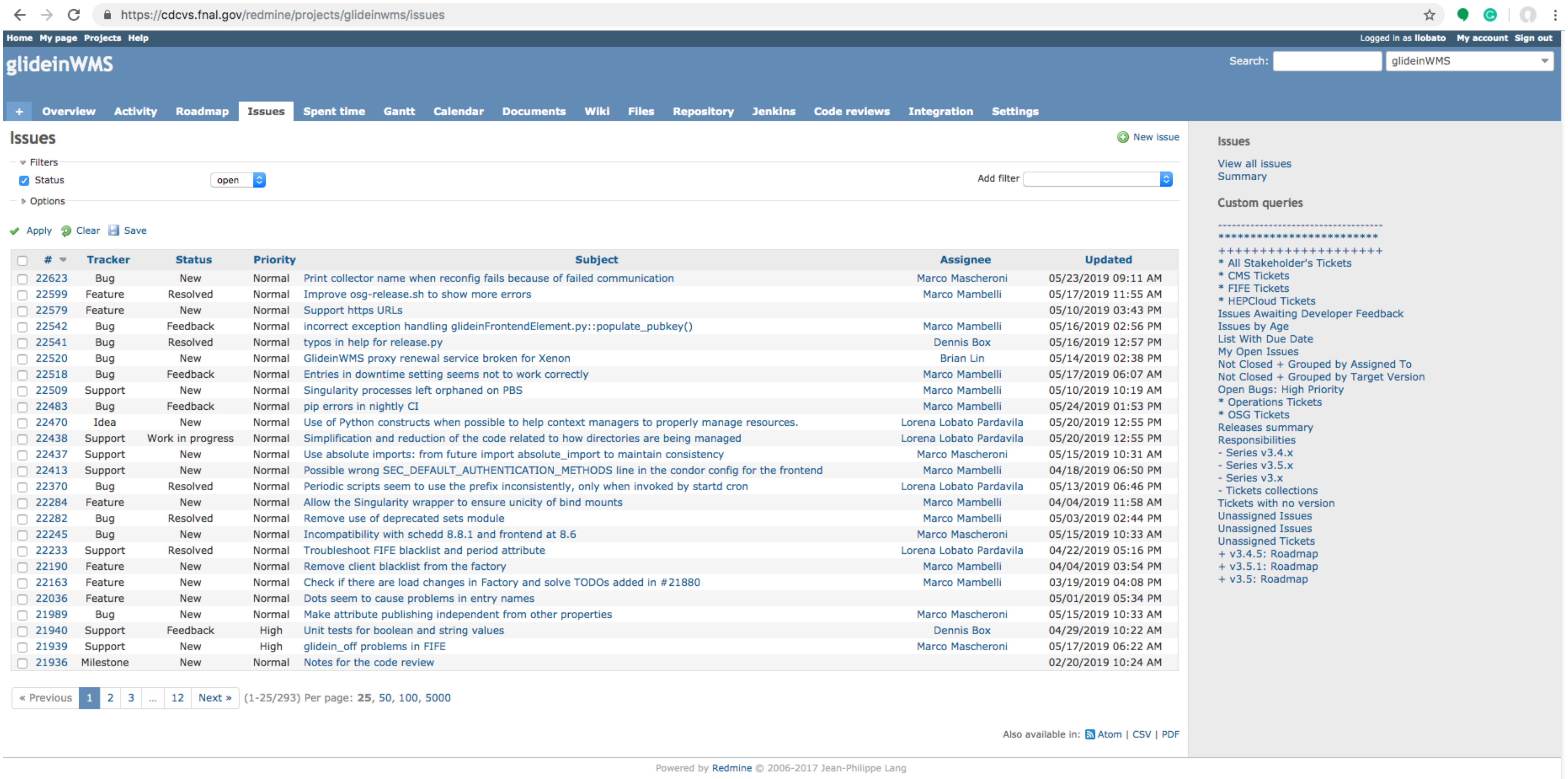
GlideinWMS Related Builds

S	W	Nombre ↓	Último Éxito	Último Fallo	Última Duración
●	⛅	GlideinWMS-Builds	1 Año 8 Mes - #29	1 Año 8 Mes - #28	4 Min 7 Seg
●	☀	GlideinWMS-Pylint	1 Año 0 Mes - #785	N/D	29 Min
●	⛈	GlideinWMS-Pylint-Old	1 Año 11 Mes - #78	N/D	6 Min 31 Seg
●	☀	GlideinWMS-Unittests	1 Año 11 Mes - #56	N/D	10 Min

Icono: S M L

Links - Redmine

https://cdcvs.fnal.gov/redmine/projects/glideinwms/wiki#For-New-Member-Orientation-see-NewMemberOrientation



https://cdcvs.fnal.gov/redmine/projects/glideinwms/issues

Home My page Projects Help — Logged in as llobato My account Sign out

glideinWMS — Search: glideinWMS

+ Overview Activity Roadmap Issues Spent time Gantt Calendar Documents Wiki Files Repository Jenkins Code reviews Integration Settings

Issues

New issue

Filters

☑ Status: open — Add filter

Options

✓ Apply Clear Save

□	#	Tracker	Status	Priority	Subject	Assignee	Updated
□	22623	Bug	New	Normal	Print collector name when reconfig fails because of failed communication	Marco Mascheroni	05/23/2019 09:11 AM
□	22599	Feature	Resolved	Normal	Improve osg-release.sh to show more errors	Marco Mambelli	05/17/2019 11:55 AM
□	22579	Feature	New	Normal	Support https URLs		05/10/2019 03:43 PM
□	22542	Bug	Feedback	Normal	incorrect exception handling glideinFrontendElement.py::populate_pubkey()	Marco Mambelli	05/16/2019 02:56 PM
□	22541	Bug	Resolved	Normal	typos in help for release.py	Dennis Box	05/16/2019 12:57 PM
□	22520	Bug	New	Normal	GlideinWMS proxy renewal service broken for Xenon	Brian Lin	05/14/2019 02:38 PM
□	22518	Bug	Feedback	Normal	Entries in downtime setting seems not to work correctly	Marco Mambelli	05/17/2019 06:07 AM
□	22509	Support	New	Normal	Singularity processes left orphaned on PBS	Marco Mambelli	05/10/2019 10:19 AM
□	22483	Bug	Feedback	Normal	pip errors in nightly CI	Marco Mambelli	05/24/2019 01:53 PM
□	22470	Idea	New	Normal	Use of Python constructs when possible to help context managers to properly manage resources.	Lorena Lobato Pardavila	05/20/2019 12:55 PM
□	22438	Support	Work in progress	Normal	Simplification and reduction of the code related to how directories are being managed	Lorena Lobato Pardavila	05/20/2019 12:55 PM
□	22437	Support	New	Normal	Use absolute imports: from future import absolute_import to maintain consistency	Marco Mascheroni	05/15/2019 10:31 AM
□	22413	Support	New	Normal	Possible wrong SEC_DEFAULT_AUTHENTICATION_METHODS line in the condor config for the frontend	Marco Mambelli	04/18/2019 06:50 PM
□	22370	Bug	Resolved	Normal	Periodic scripts seem to use the prefix inconsistently, only when invoked by startd cron	Lorena Lobato Pardavila	05/13/2019 06:46 PM
□	22284	Feature	New	Normal	Allow the Singularity wrapper to ensure unicity of bind mounts	Marco Mambelli	04/04/2019 11:58 AM
□	22282	Bug	Resolved	Normal	Remove use of deprecated sets module	Marco Mambelli	05/03/2019 02:44 PM
□	22245	Bug	New	Normal	Incompatibility with schedd 8.8.1 and frontend at 8.6	Marco Mascheroni	05/15/2019 10:33 AM
□	22233	Support	Resolved	Normal	Troubleshoot FIFE blacklist and period attribute	Lorena Lobato Pardavila	04/22/2019 05:16 PM
□	22190	Feature	New	Normal	Remove client blacklist from the factory	Marco Mambelli	04/04/2019 03:54 PM
□	22163	Feature	New	Normal	Check if there are load changes in Factory and solve TODOs added in #21880	Marco Mambelli	03/19/2019 04:08 PM
□	22036	Feature	New	Normal	Dots seem to cause problems in entry names		05/01/2019 05:34 PM
□	21989	Bug	New	Normal	Make attribute publishing independent from other properties	Marco Mascheroni	05/15/2019 10:33 AM
□	21940	Support	Feedback	High	Unit tests for boolean and string values	Dennis Box	04/29/2019 10:22 AM
□	21939	Support	New	High	glidein_off problems in FIFE	Marco Mascheroni	05/17/2019 06:22 AM
□	21936	Milestone	New	Normal	Notes for the code review		02/20/2019 10:24 AM

« Previous 1 2 3 ... 12 Next » (1-25/293) Per page: **25**, 50, 100, 5000

Also available in: Atom | CSV | PDF

Issues

View all issues
Summary

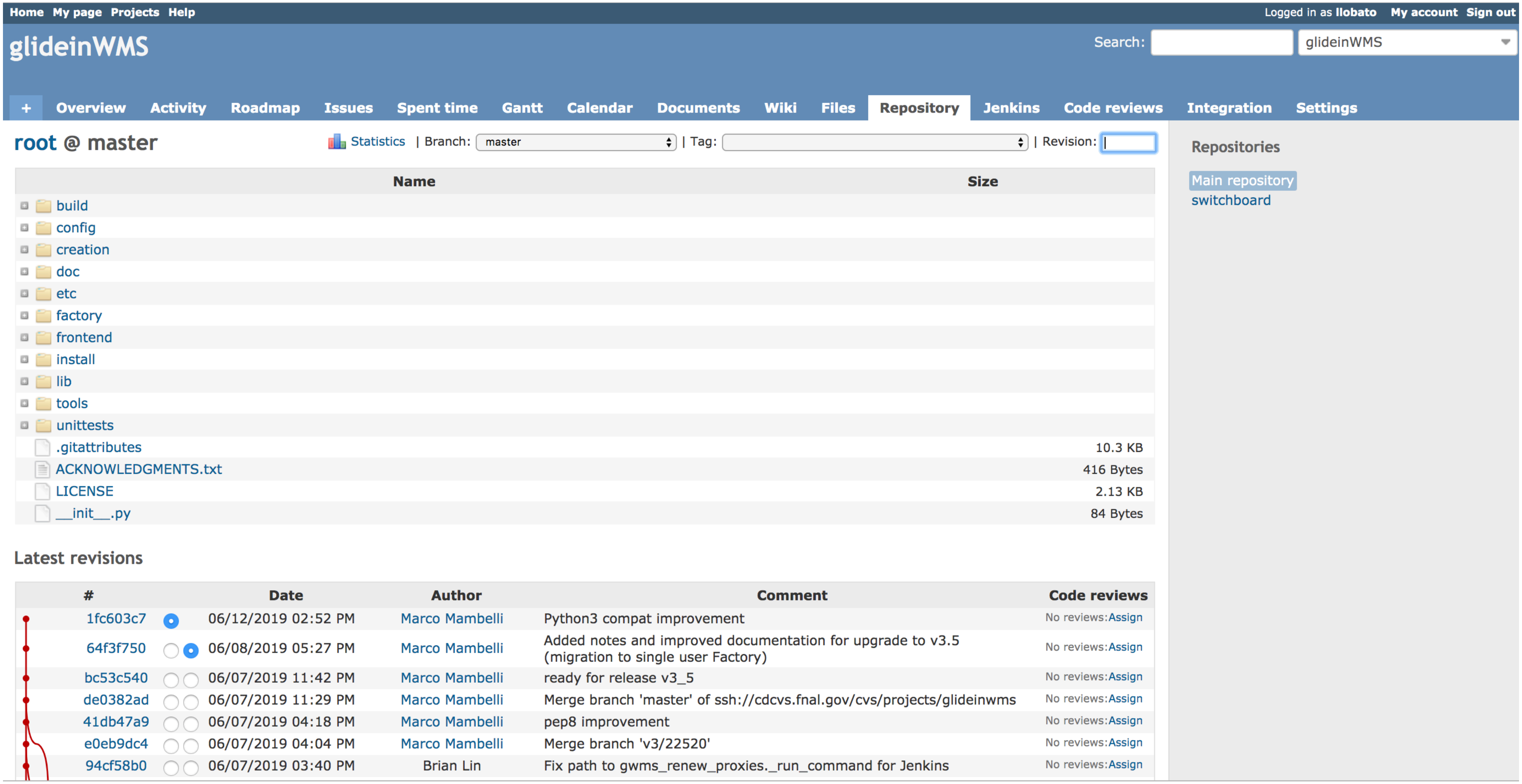
Custom queries

****
++++++++++++++++++++
* All Stakeholder's Tickets
* CMS Tickets
* FIFE Tickets
* HEPCloud Tickets
Issues Awaiting Developer Feedback
Issues by Age
List With Due Date
My Open Issues
Not Closed + Grouped by Assigned To
Not Closed + Grouped by Target Version
Open Bugs: High Priority
* Operations Tickets
* OSG Tickets
Releases summary
Responsibilities
- Series v3.4.x
- Series v3.5.x
- Series v3.x
- Tickets collections
Tickets with no version
Unassigned Issues
Unassigned Issues
Unassigned Tickets
+ v3.4.5: Roadmap
+ v3.5.1: Roadmap
+ v3.5: Roadmap

Powered by Redmine © 2006-2017 Jean-Philippe Lang

Links – Git repo

```
git clone ssh://p-glideinwms@cdcvs.fnal.gov/cvs/projects/glideinwms
```


Home My page Projects Help
Logged in as llobato My account Sign out
glideinWMS
Search:
glideinWMS
+ Overview Activity Roadmap Issues Spent time Gantt Calendar Documents Wiki Files Repository Jenkins Code reviews Integration Settings
Statistics
Repositories
Main repository
switchboard

root @ master

| Branch: master | Tag: | Revision:

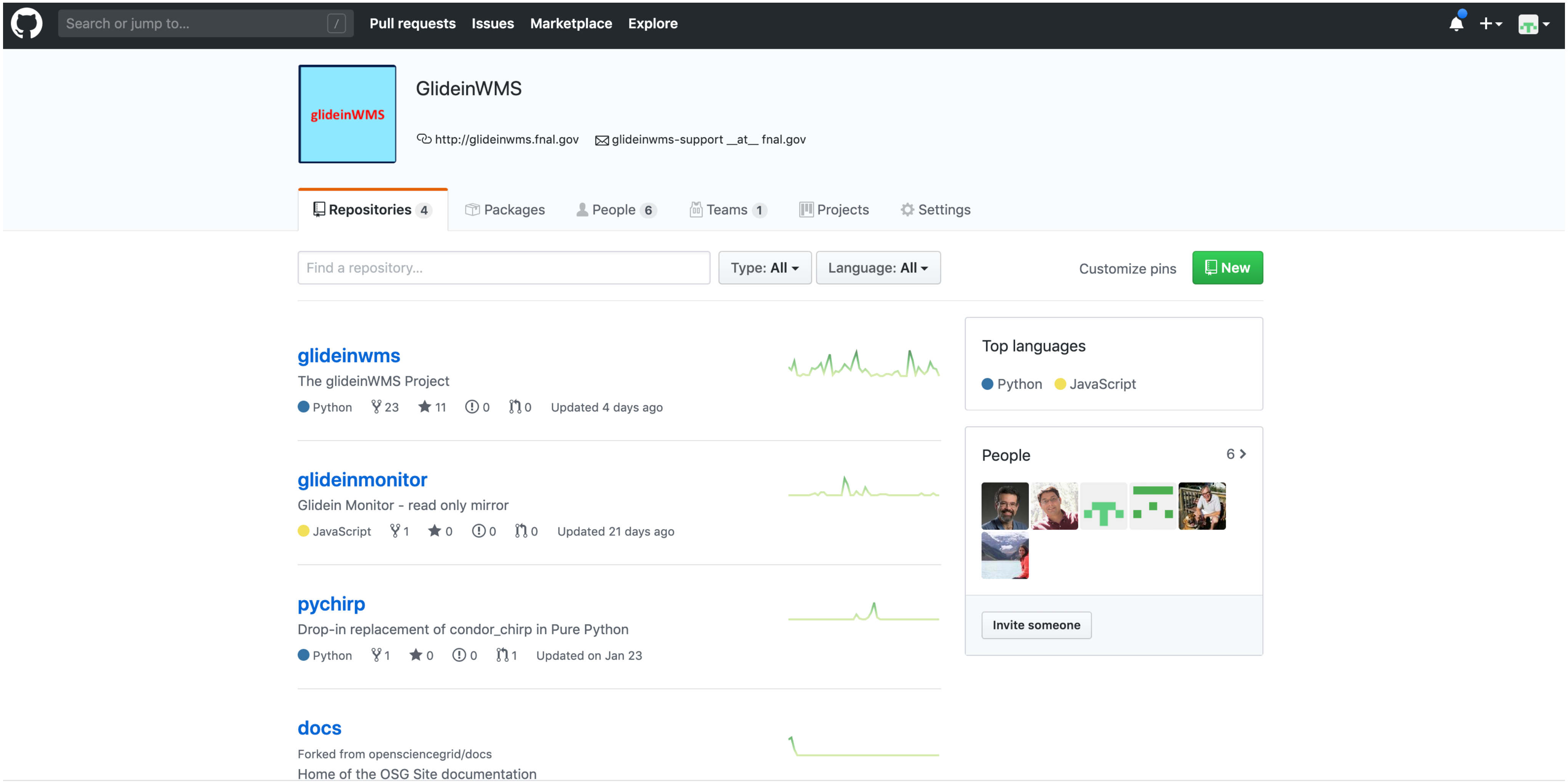
Name	Size
build	
config	
creation	
doc	
etc	
factory	
frontend	
install	
lib	
tools	
unittests	
.gitattributes	10.3 KB
ACKNOWLEDGMENTS.txt	416 Bytes
LICENSE	2.13 KB
__init__.py	84 Bytes

Latest revisions

#		Date	Author	Comment	Code reviews
1fc603c7	●	06/12/2019 02:52 PM	Marco Mambelli	Python3 compat improvement	No reviews:Assign
64f3f750	○ ●	06/08/2019 05:27 PM	Marco Mambelli	Added notes and improved documentation for upgrade to v3.5 (migration to single user Factory)	No reviews:Assign
bc53c540	○ ○	06/07/2019 11:42 PM	Marco Mambelli	ready for release v3_5	No reviews:Assign
de0382ad	○ ○	06/07/2019 11:29 PM	Marco Mambelli	Merge branch 'master' of ssh://cdcvs.fnal.gov/cvs/projects/glideinwms	No reviews:Assign
41db47a9	○ ○	06/07/2019 04:18 PM	Marco Mambelli	pep8 improvement	No reviews:Assign
e0eb9dc4	○ ○	06/07/2019 04:04 PM	Marco Mambelli	Merge branch 'v3/22520'	No reviews:Assign
94cf58b0	○ ○	06/07/2019 03:40 PM	Brian Lin	Fix path to gwms_renew_proxies._run_command for Jenkins	No reviews:Assign

Links – GitHub

https://github.com/glideinWMS



GlideinWMS

http://glideinwms.fnal.gov glideinwms-support __at__ fnal.gov

glideinwms

The glideinWMS Project

● Python 23 ★ 11 0 0 Updated 4 days ago

glideinmonitor

Glidein Monitor - read only mirror

● JavaScript 1 ★ 0 0 0 Updated 21 days ago

pychirp

Drop-in replacement of condor_chirp in Pure Python

● Python 1 ★ 0 0 1 Updated on Jan 23

docs

Forked from opensciencegrid/docs

Home of the OSG Site documentation

Top languages

● Python ● JavaScript

People 6

- Scientific computing
- GlideinWMS
- HTCondor
- Resources
- Monitoring
- Links
- **Demo**

Summary

- Scientific computing requires a lot of computing
- GlideinWMS is a pilot based resource provisioning tool for distributed High Throughput Computing
  - Provides reliable and uniform virtual clusters
  - Submits Glideins to unreliable heterogeneous resources
- HTCondor provides the tools
  - ClassAds, schedd, startd, collector
- Worker nodes on Resources
  - cores, memory and disk
  - Can run multiple jobs
- Good monitoring helps troubleshooting
- http://glideinwms.fnal.gov/

References

- HTCondor slides are based in part on a presentation by Todd Tannenbaum and the HTCondor team http://www.cs.wisc.edu/condor/
- GlideinWMS slides are based in part on previous presentations by the GlideinWMS project developers

Extra slides

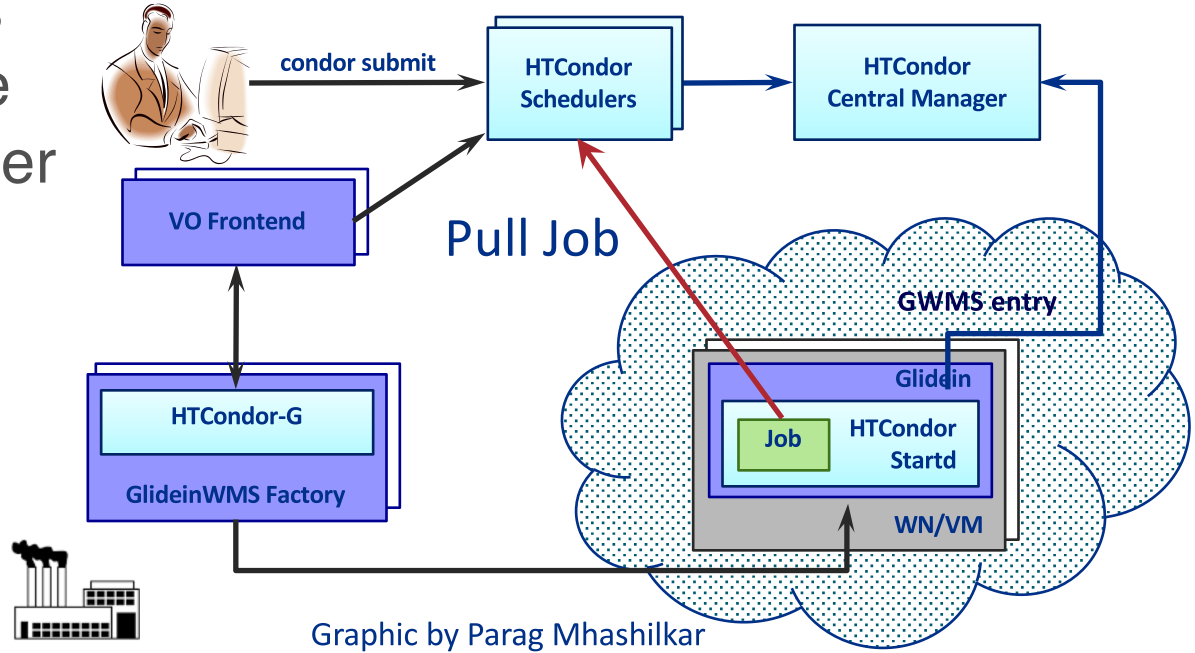
Glidein based Workload Management System

- The Grid is a sum of thousands of independent Grid sites
- Choosing where to try to run the jobs is not a trivial task -> Extra operations
  - To keep in mind: reliability, scalability, priorities, location, quotas…
- Resources are located in independent pools
  - How minimizing the waste?
- Need abstraction layer for submission to any site
- GlideinWMS makes it easier and transparent to the users
  - Division of operations
- Advantages of a local batch system (HTCondor based overlay system)

Glidein based Workload Management System

GWMS is a workload manager for scientific jobs: it provisions resources from different sources and makes them available to users jobs in a single virtual cluster

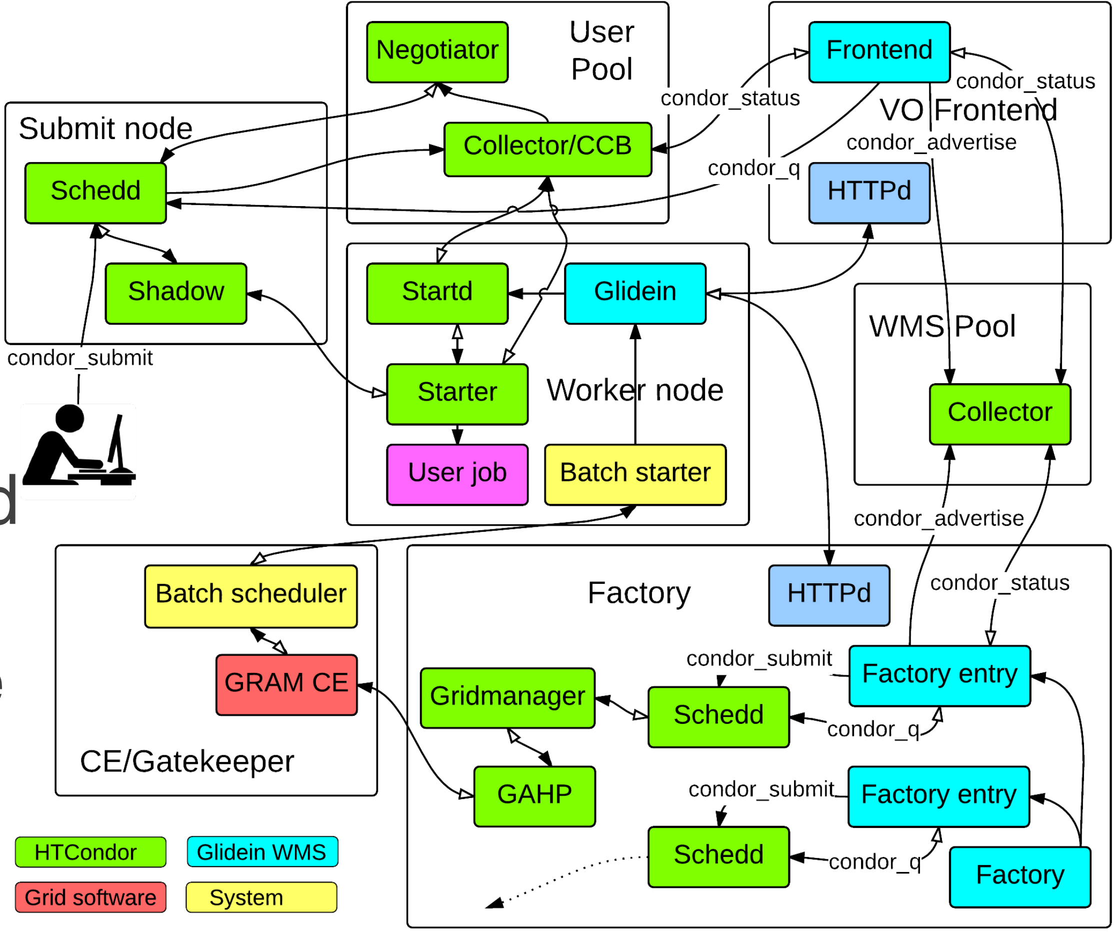
- The cluster provided to you is virtual because the hosts are in different places and sometime not even there until you need them
- Uses Glideins or pilots to acquire and prepare the resources, or worker nodes, where the jobs run
- Based on HTCondor



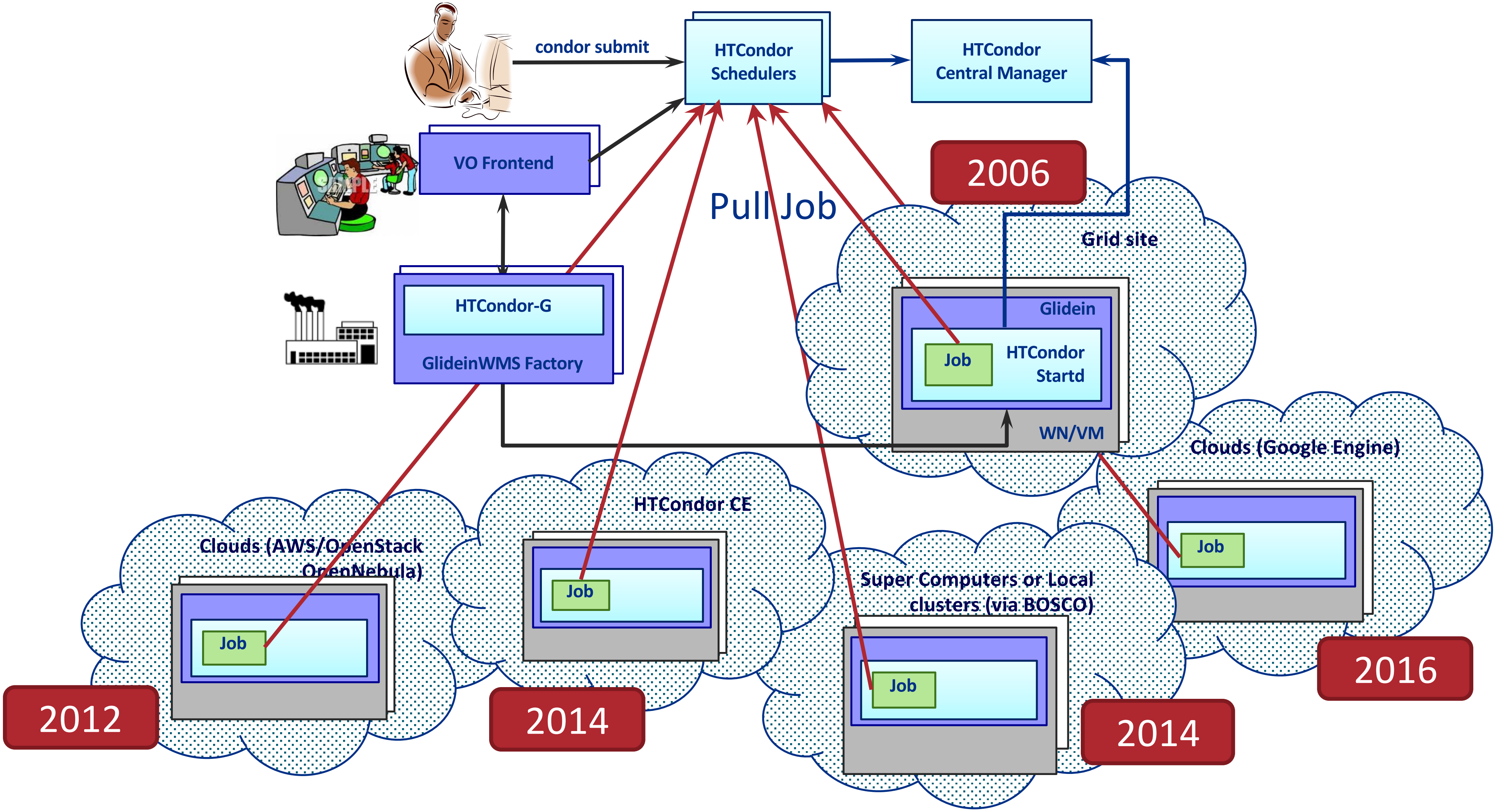
Graphic by Parag Mhashilkar

HTCondor building blocks in Glidein WMS

- The Factory works with an HTCondor pool, WMS pool, to submit Glideins to different resources
- The HTCondor Glideins are pilots that launch a startd that registers on a second HTCondor pool, User pool
- User jobs are matched and execute on the resources
- The Frontend monitors the user schedds and notifies the Factory about the need for more Glideins



New resources added over time



Job and Machine ‘dimensions’

- Job request
  - request_cpus: number of cores, integer, default 1.
  - request_disk: amount of disk space in Kbytes, default to sum of sizes of the job's executable and all input files (or image size)
  - request_memory: amount of memory space in Mbytes, default to executable size
- Machine
  - Cpus: number of cores, integer, by default the available cores
  - Disk: amount of disk space on this machine available for the job in KiB, by default the available space
  - Memory: amount of RAM in MiB in this slot
- Over and Under provision