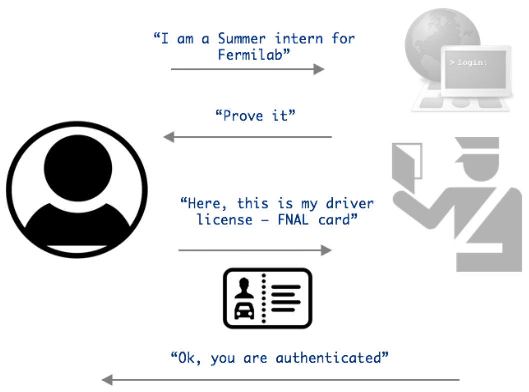
Basic Concepts - A GlideinWMS Glossary

Glossary of basic terms useful to understand GlideinWMS. When adding a new term, respect the alphabetic ordering. GlideinWMS and HTCondor have their own sub-section with terms specific to those systems.

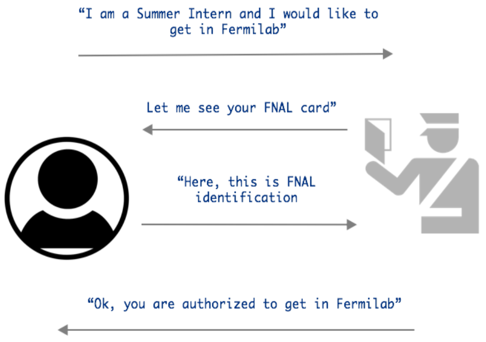
Authentication

Process where the identity of a subject (user, machine, service..) is confirmed


"I am a Summer intern for Fermilab"
> login:
"Prove it"
"Here, this is my driver license - FNAL card"
"Ok, you are authenticated"

Authorization

Process that determines whether an authenticated subject who has requested an action has the right to do so.


"I am a Summer Intern and I would like to get in Fermilab"
Let me see your FNAL card"
"Here, this is FNAL identification
"Ok, you are authorized to get in Fermilab"

Central Manager

Host that manages the pool of resources in a cluster, sometimes also the Job queue. In HTCondor, the host running the collector and negotiator daemons and performing the matchmaking.

Certificate

File written in a standard format (X.509) which confirms a subject’s identity. Differences:

- .PEM: Container format that may include just the public certificate or may include an entire certificate chain including public key, private key and root certificates.
- .KEY: PEM formatted file containing just the private key of a specific certificate and is merely a conventional name and not a standardized one.
- .P12: Passworded container format that contains both public and private certificate pairs. It’s fully encrypted.

Cluster

Collection of parallel or distributed computers which are interconnected among themselves using high-speed networks.

Compute Node

Machine that is dedicated to run Jobs (or portions of them). It is characterized by its number of CPU cores, the memory (RAM) capacity, local file system size (disk) and eventual special resources available on the node (GPUs, …). Its description can include also Operating Systems and software installed on it.
OS-level virtualization refers to an operating system paradigm in which the kernel allows the existence of multiple isolated user space instances

Container

OS-level virtualization refers to an operating system paradigm in which the OS (kernel) allows the existence of multiple instances isolated on different dimensions (user space, network space, file system, …). It can be seen as a finer level virtualization compared to Virtual Machines. Common container technologies are Docker, Podman, Singularity.

Core

Part of a CPU that receives instructions and performs calculations, or actions, based on those instructions. In GlideinWMS, will be the size available/requested for the user job. A Glidein can discover dynamically the available cores or assume them:

- NODE: Use all the cores available on the physical (or virtual) machine
- SLOT: Use the number of Cores made available by the batch system
- GLIDEIN_CPU=N (Where N is a Natural number): assume to have N Cores available

Compute Element

Site grid gatekeeper technology which represents the entry point for local resources.Provides remote access for VOs to submit “pilot jobs” to your local batch system.

Credentials

Set of files proving identity and privilege of a user. These can be Certificates, Keys, Tokens

Daemon

A computer program that runs as a background process, rather than being under the direct control of an interactive user.

Docker

Docker is a platform to develop, deploy, and run applications inside containers.

Execute Host

Same as Compute Node.

Glidein

Pilot (job). It's just a property configured execution node submitted as a Grid/Cloud job. Defines how multi-core glideins should split their resources

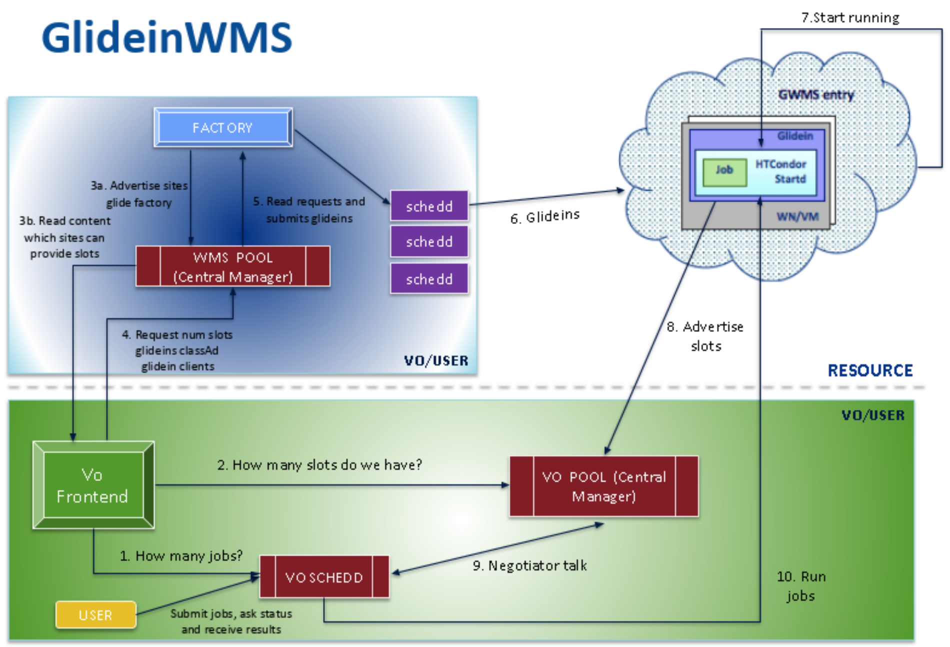
- Fixed: Create N single core slots, and split memory and disk equally among them.
- Partitionable: Start up with a single partitionable slot and let HTCondor do the splitting as needed.

GlideinMonitor

GlideinMonitor [https://github.com/glideinWMS/glideinmonitor] is a Web application that allows to view GlideinWMS's Glidein log files: it provides a user interface, tools to do quick searches and to decode the log content; it provides an efficient managed archive of the log files and a framework to add log processing, e.g. log sanitation.

GlideinWMS (GWMS)

Glidein based Workload Management System. It's a workload manager for scientific jobs: it provisions resources from different sources and makes them available to users jobs in a single virtual cluster


GlideinWMS
FACTORY
3a. Advertise sites glide factory
3b. Read content which sites can provide slots
5. Read requests and submits glideins
schedd
schedd
schedd
WMS POOL (Central Manager)
4. Request num slots glideins classAd glidein clients
VO/USER
6. Glideins
7.Start running
GWMS entry
Glidein
Job
HTCondor Startd
WN/VM
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

Entry

A computing resource, as represented in the Factory configuration and operation. It includes a description of key features, information and limits to submit Glideins, how to monitor the status. Typical entries are a Grid Computing Element, a Cloud resource (e.g. AWS zone), an HPC cluster (e.g. Leadership machines). You do require at least one Entry to use a resource.

Factory

Receives requests from the Frontend(s) and submits a HTCondor startd wrapper (glidein) to entry points (grid sites).

Frontend

It looks at the user Jobs queued (in HTCondor schedds) and matches them to resources (Entries) that can run them. It sends requests to the Factory to provision all needed Glideins.

Group

A way to cluster (payload) Jobs. Different Groups can use different credentials, belong to different users or VOs, use different submit hosts, be a different accounting group, or simply have a set of common requirements (e.g. run only on some Entries, use GPUs, …). All Jobs in the same Group share the same Submit hosts, pilot Credentials and Glideins, i.e. the Glidein provisioned for one of the Jobs in the group can run (in parallel or sequentially) also other jobs from the same Group. The accounting group could differ since the Glidein can report the actual owner of the Job.

Metasite

Aka Entry_set, is a group of Entries usually sharing the same physical resource. It is used in the Factory configuration and operation to provide common descriptions and limits, e.g. to limit the number of Glideins submitted to a cluster that has multiple gateways (Entries).

High Performance Computing (HPC)

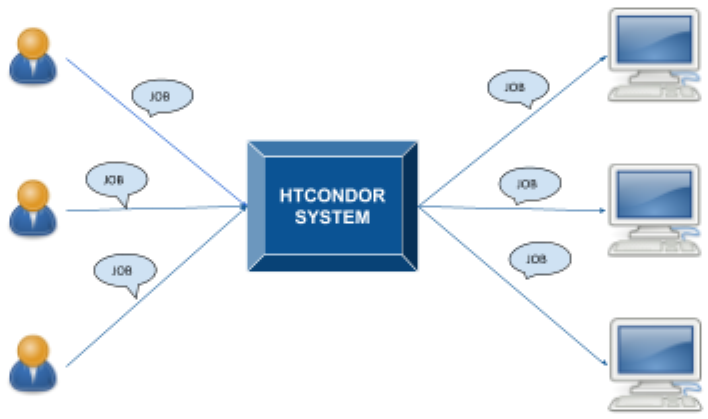
Is the ability to process data and perform complex calculations at high speeds. It is also the use of super computers and parallel processing techniques for solving complex computational problems. Tasks are characterized as needing large amounts of computing power and communication bandwidth for short periods of time, measuring the amount of basic operations per second (FLOPS). Typical HPC resources are the Top 500 supercomputers, or the machines at Leadership Computing Facilities at National Labs

High Throughput Computing (HTC)

The use of many computing resources over long periods of time to accomplish a computational task. Also, a computing paradigm that focuses on the efficient execution of a large number of loosely-coupled tasks. The main performance metric is the ability to complete many complex operations (jobs) per day or month. HTC is generally performed orchestrating together many off-the-shelf computers, e.g. using Grids or Clouds.

HTCondor

Software that schedules and runs computing tasks on computers. Quoting the website [https://research.cs.wisc.edu/htcondor/description.html]: HTCondor is a specialized workload management system for compute-intensive jobs. Like other full-featured batch systems, HTCondor provides a job queueing mechanism, scheduling policy, priority scheme, resource monitoring, and resource management. Users submit their serial or parallel jobs to HTCondor, HTCondor places them into a queue, chooses when and where to run the jobs based upon a policy, carefully monitors their progress, and ultimately informs the user upon completion.



ClassAd

Internal data representation. It is the language for objects(jobs and machine) to express attributes about themselves and what they require/desire in a "match".

Cluster

Set of related jobs. Each cluster has a cluster number, an unsigned integer value unique to the job queue on a submit host. Each individual job within a cluster is given a process number, and process numbers always start at zero.

Daemons

In HTCondor instances of the following daemons interact to perform its functions:

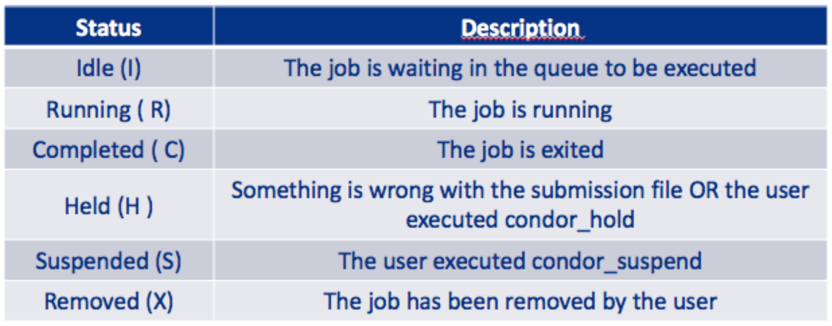
- Collector: Collects information about the status of the pool as well as it controls the management of user jobs and matches them to machines generated by the Glidein Factory. The VO will query the Collector to determine how many glideins should be created.
- Master: Every condor machine needs a mater. It's responsible for keeping all the rest of the HTCondor daemons running on each machine in the pool.
- Negotiator: Responsible of matching of jobs to resources.
- Schedd: Runs on the submit machine and it allows users to submit jobs. It represents resource requests to the HTCondor pool.
- Shadow: When a job starts to run, this process starts up on the submit machine. It's the process by which remotely executing jobs can access the environment from which it was submitted, such as input and output files.
- Startd: Represents the resource/machine on which it is running to the HTCondor pool. It advertises resource attributes to the rest of the condor pool
- Starter: It's the entity that actually spawns the HTCondor job on a give machine

DAGMan

Meta-scheduler for HTCondor. It manages dependencies between Jobs a higher level than the HTCondor Scheduler, using Direct Acyclic Graphs (DAGs)

Job

In HTCondor, it is a representation of a piece of work. As it runs, each job requires an input (from infile), the executable program (instructions) and the output(to outfile), needing resources when it runs.



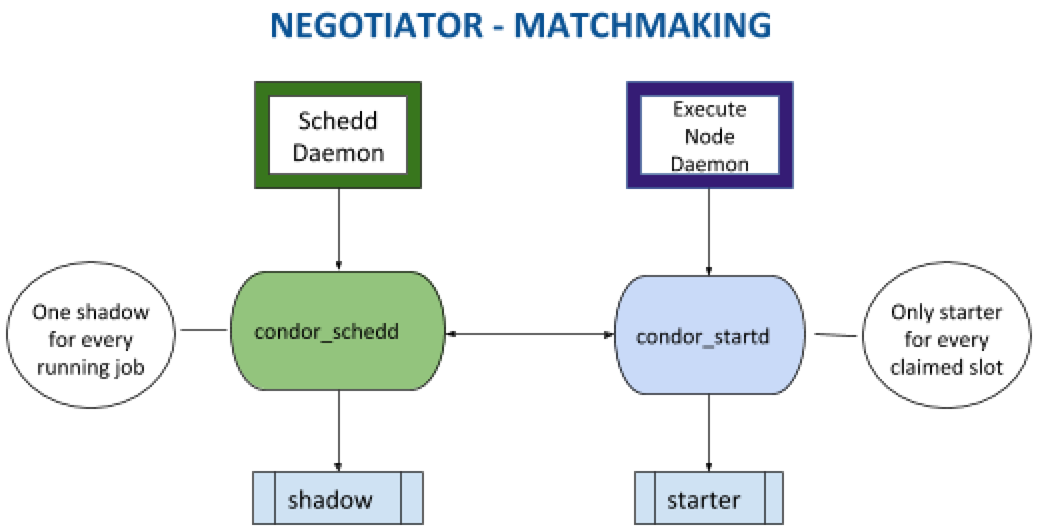
Status	Description
Idle (I)	The job is waiting in the queue to be executed
Running (R)	The job is running
Completed (C)	The job is exited
Held (H)	Something is wrong with the submission file OR the user executed condor_hold
Suspended (S)	The user executed condor_suspend
Removed (X)	The job has been removed by the user

Machine

Available resources or computers than can do processing. Each machine can have 1 or more Cores (called CPUs by HTCondor).

Match

Process of associating a Job with a Machine



Match expression

Will be evaluated when matching lidein(pilot) request to entries. This will determine which sites are submitted to

Pool

Collection of resources (machines) and resources requests (jobs). It comprises a machine which serves as Central Manager (collector and negotiator) and an arbitrary number of other machines that have joined the pool as computing resources (startd/machines). A pool can also be used to allow demons to communicate exchanging HTCondor ClassAds.
In GlideinWMS you have:

- (VO) User Pool: HTCondor pool (at least collector and negotiator daemons) where all the user jobs (for a VO) are matched to Glideins and managed
- WMS Pool : HTCondor Pool (at least collector, Frontend and Factory) used by a Frontend to communicate with a Factory, i.e. to request and manage Glideins submissions

Process

Within HTCondor, each job in a single submission. Each HTCondor job is identified by a Job ID, e.g. 201.5, consisting of two numbers, the Cluster ID (identifies a submission to a schedd) and the Process ID (distinguishing the jobs within the same submission).

Requirements

Needs for the job. Example: require a machine with python installed.

Rank

Preferences for the job. Example: prefer a machine with the most of memory.

Slot

Describes the portion of a Machine which is matched to a Job in HTCondor. Each slot may contain one or more cores. Each slot has his own machine ClassAd

- Static: when the Glidein is split at startup and the partition remain the same for its duration
- Partitionable: a slot that takes ownership of multiple node resources (Cores, memory) and provides portions of these to Jobs according to their requests, creating Dynamic slots. No job is running directly on the Partitionable slot, it owns the leftovers, unassigned portion.
- Dynamic(ally created): Created for a specific Job, as a sub-slot of a Partitionable slot, used by the Job, disappears at the end of it

Start expression

Will be evaluated when jobs are matched to pilots. This will determine, once Glideins are up and running, which jobs will run on them.

State

Machine’s HTCondor state

- Claimed: The machine is claimed by a remote schedd and it’s probably running a job.
- Unclaimed: The machine is available to run HTCondor jobs, but a good match is either not available or not yet found
- Matched: Central Manager has found a good match for the resource, but and scheduler has not yet claimed it.
- Drained: The slot doesn’t accept jobs, because the machine is being drained

Submit file

Provides commands on how to execute the job.

Universe

Type of runtime environment that controls how HTCondor handles jobs.

Job (single/multi core)

Single computing task representation, like a Unix process or element of a workflow. The number of cores corresponds to the number of computing units that the job will use to do the processing.

Job manager

Controls and/or monitor job execution. It is automatically submitted as a task by the Batch service when the job is created and scheduled to execute before the other tasks in a job.

Keytab

Files used to authenticate without typing a Kerberos password

Kerberos

Temporary identification token given to a user. Kerberos commands:

- kinit create a Kerberos ticket
- klist: Retrieves and list details about your Kerberos ticket
- kdestroy: Deletes your Kerberos tickets from your local machine

OSG (Open Science Grid)

Worldwide grid of technologies resources, which facilitates distributed computing for scientific research. It provides common service and support for resource providers and scientific institution using a distributed fabric of high throughput computational services.

Singularity

Container platform designed for use on computational resources without giving extra privileges to unprivileged users.

Submit Host

Node from which Jobs are submitted to GlideinWMS or a batch system. Usually the only node of a cluster that allows interactive login. In HTCondor it runs the schedd daemon.

Proxy

Temporary credential derived from an existing certificate. We use it for job submission and adta movement

- VO FE proxy: Used to authenticate with the other GlideinWMS services.
- Pilot Proxies: Used/Delegated to the factory services and submitted on the GlideinWMS pilots.

- Both must be owned by user “frontend”

Virtual Machine (VM)

"Emulation of a computer system. Virtual machines are based on computer architectures and provide functionality of a physical computer. Their implementations may involve specialized hardware, software, or a combination." - from Wikipedia
VMs can be instantiated and managed in cloud systems. Commercial clouds like AWS and GCE or private clouds. Fermicloud is a private cloud and provides virtual machines that can be used for code development and integration.

VOMS (Virtual Organization Membership Service)

VOMS proxy is an X509 proxy that also contains information about your role and group memberships in the experiment (Virtual Organization - VO). I.e. special permissions connected with your identity. More information at https://italiangrid.github.io/voms/documentation/voms-clients-guide/. Commands:

- voms-proxy-info -all
- voms-proxy-init --rfc --voms YOURVO -valid 192:00

This technology is being phased out and will be replaced by tokens.