The Pilot Way To Grid Resources Using GlideinWMS

Parag Mhashilkar

Fermilab, Batavia, IL

Overview

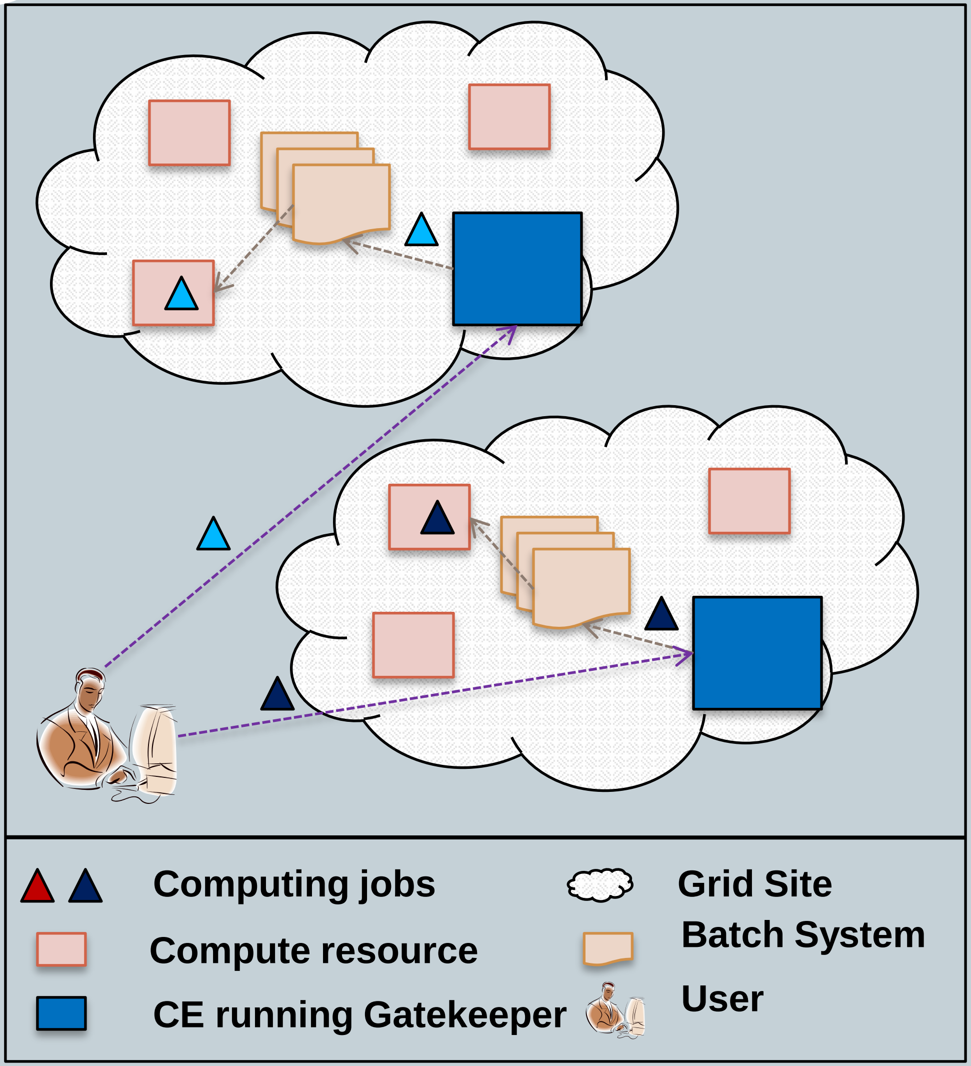
- Grid Computing
- Pilot Workload Management (WMS) Paradigm
- Security Considerations
- GlideinWMS implementation of Pilot Paradigm
- Pseudo-interactive Monitoring using glideinWMS
- Scalability of glideinWMS
- Summary
- References

Grid Computing

- Distributed computing paradigm spanning many administrative domains.
- Widely deployed by the scientific communities with high computing demands
  - High Energy Physics (HEP)
  - Astro Physics Communities
  - Weather Surveys
  - Biology
  - [...]
- General purpose Grids used by the scientific communities
  - Open Science Grid (OSG)
  - European Grid for E-SciEnce (EGEE)
  - [...]

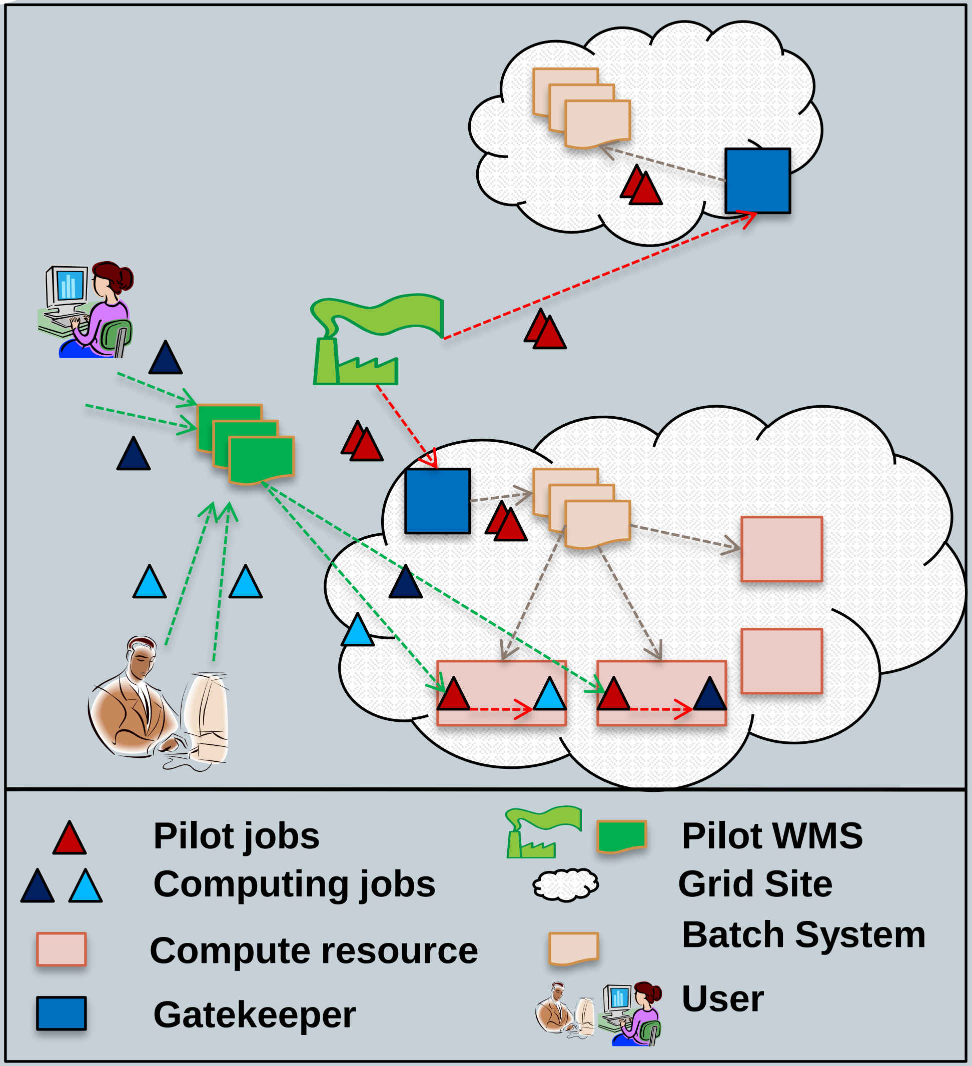
Typical Grid Use Case

- Grid Site: An administrative domain
  - Administrators deploy Grid middleware with following components-
    - Compute Element (CE) running a gatekeeper, executes jobs on behalf of users
    - Gatekeeper forwards the job to Local Batch System (BS)
    - Job runs on one of the worker nodes
- From user's perspective
  - Pros
    - Large pool of resources to satisfy their computing needs
  - Cons
    - Middleware problems in managing the job
    - Monitoring the job is complicated
    - Issues related to heterogeneity of the resources over the Grid
    - Need a meta WMS to manage Grid jobs



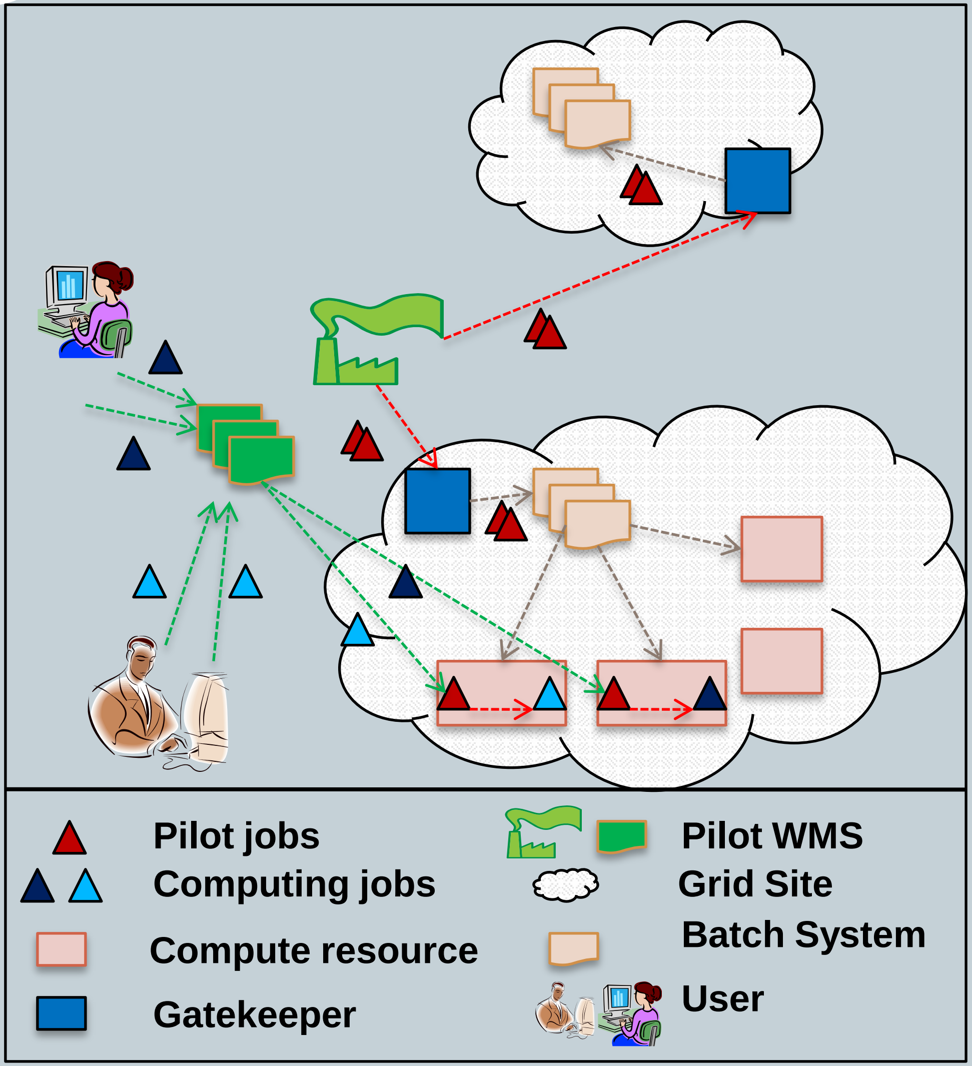
Pilot WMS Paradigm

- Pilot or just-in-time paradigm
  - Pilot factory submits pilot jobs to different grid sites
  - Pilots start running on the compute resources and fetch user jobs from the user job queue of WMS
- Advantages of pilot based WMS
  - Forms a private pool of compute resources based on the demand
  - Partially hides heterogeneity of grid sites from the user.
  - Pilot jobs are dispensable
    - If the environment is bad, pilot exits, preventing the user job to start and thus fail
    - Act as a wrapper and makes sure that the environment is right for the user job to execute.



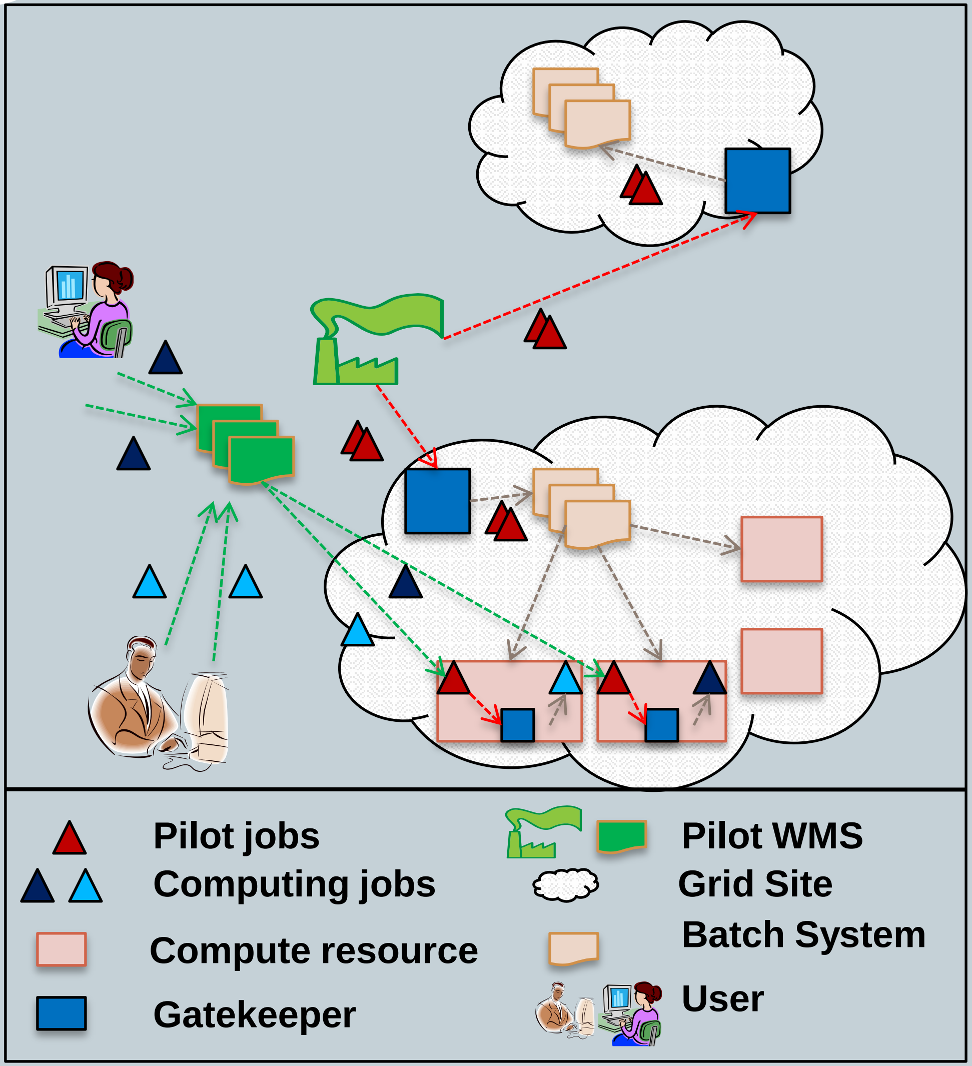
Security Considerations

- Pilots are authenticated and authorized by the site gatekeeper.
- Concerns with pilot based WMS
  - User jobs do not traverse through the site gatekeeper:
    - Does not fit well with the Grid model of authenticating / authorizing / accounting of user jobs.
  - Since pilot bootstraps the user job, both pilot and user job run under same OS user
    - This allows a malicious user to compromise the pilot job infrastructure.



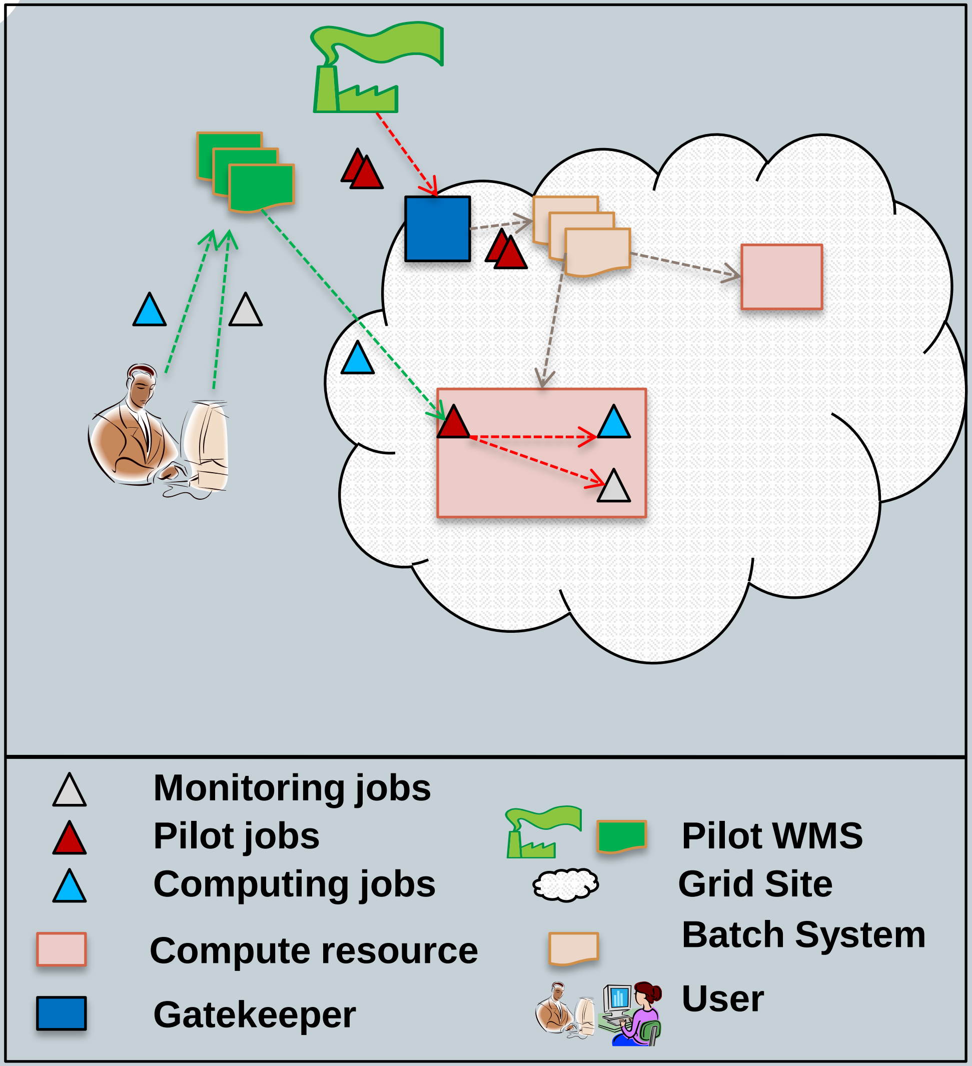
Security Considerations

- Possible Solution
  - Deploy mini-gatekeepers on the worker nodes to authenticate/authorize user jobs.
    - OSG and EGEE sites deploy gLExec, which acts as a mini gate-keepers on worker nodes to authenticate / authorize user jobs.



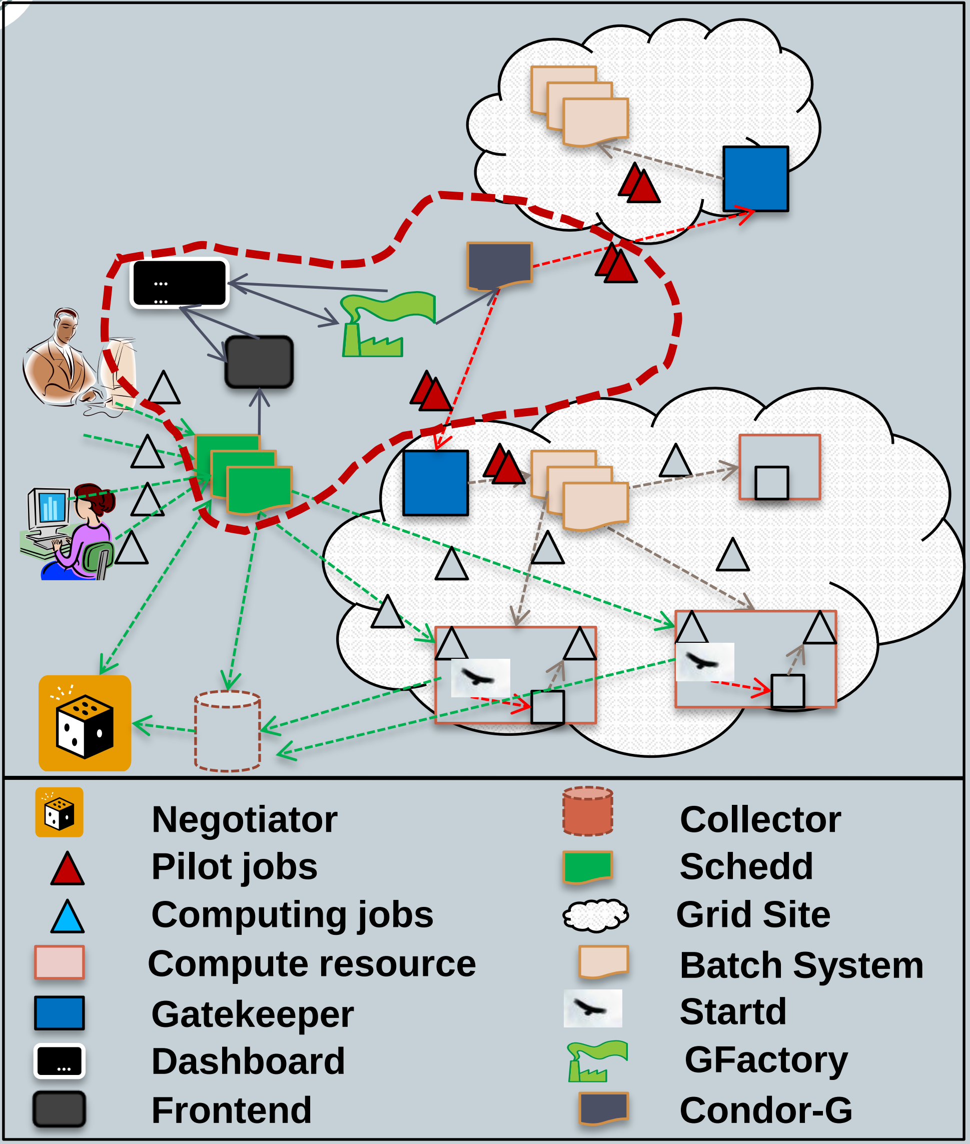
Pseudo-interactive Monitoring in Pilot WMS

- Users need more info –
  - When something goes wrong
  - In case of very long running jobs
- Information useful to the user -
  - What processes are running (ps)
  - Peek at the log files (cat/tail)
  - What files have been created (ls)
  - Peek at the process stack (gdb bt)
  - Is the machine thrashing? (top)
- Above information can be obtained through batch jobs
  - Pilot starts another job that acts as a monitoring job



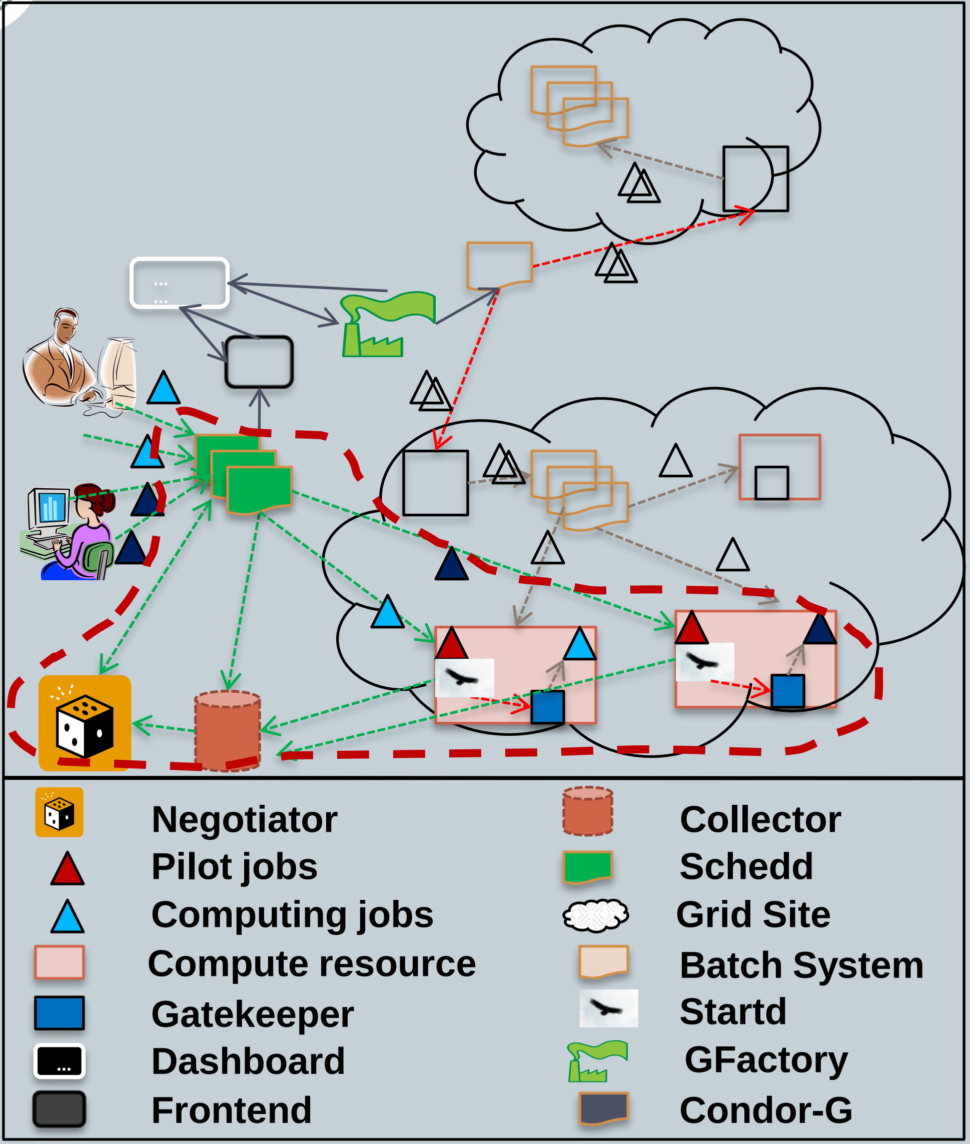
glideinWMS Implementation of the Pilot Paradigm

- glideinWMS is based on Condor with the VO Frontend and the Factory sending pilot jobs (i.e. glideins) to the grid sites.
- Condor as a user job WMS
- **glideinWMS Factory**
  - Creates and submits pilot jobs to the grid sites using CondorG
  - Condor collector acts as a dashboard for message exchanging
  - Factory receives orders from the VO frontend via the dashboard
- **VO Frontend**
  - VO frontend monitors the CondorWMS and regulates the number of pilot jobs sent by glidein factories via the dashboard
  - Frontend acts as a match maker for the glideins
- All network traffic is authenticated and integrity checked
- Support pseudo-interactive monitoring out of the box



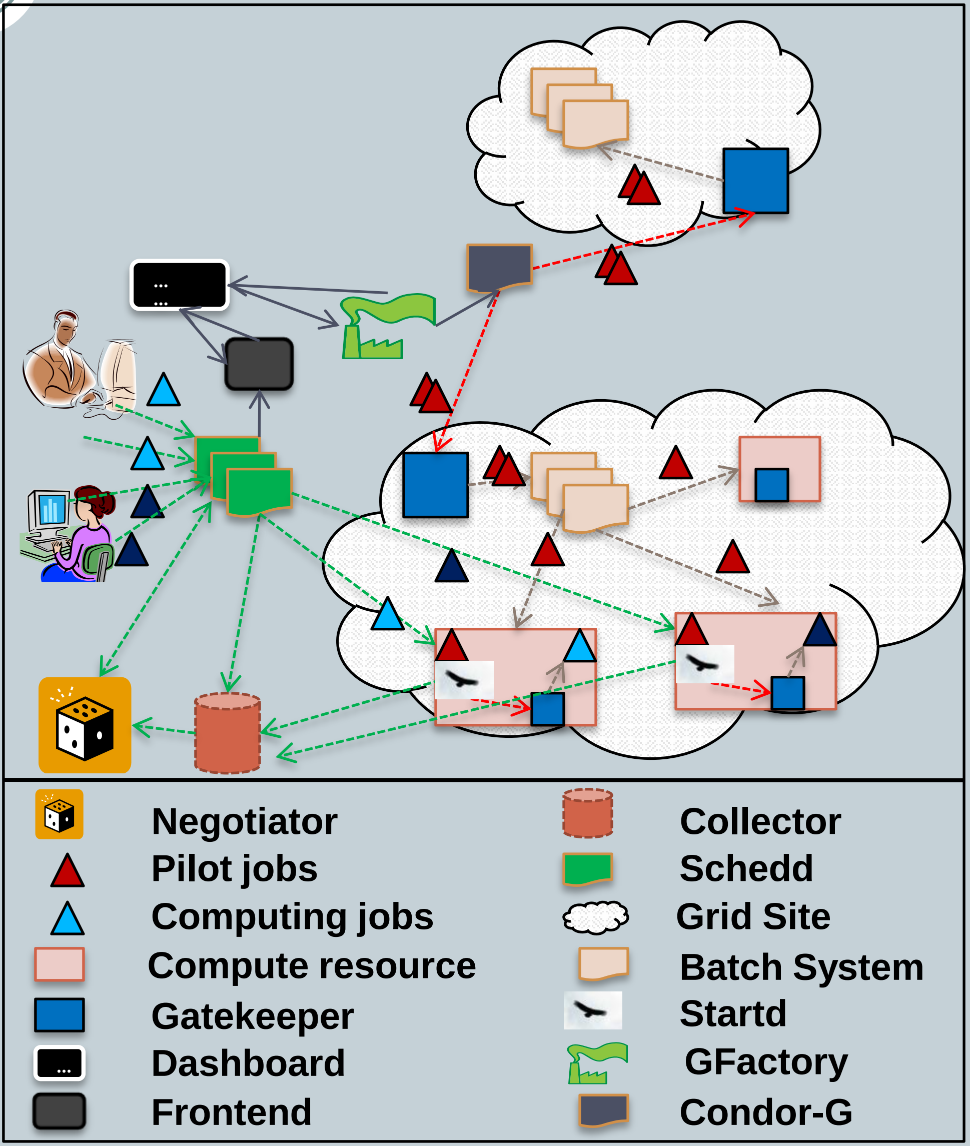
glideinWMS Implementation of the Pilot Paradigm

- glideinWMS is based on Condor with the VO Frontend and the Factory sending pilot jobs (i.e. glideins) to the grid sites.
- **Condor as a user job WMS**
  - Condor collector acts as an information database
  - Condor startd manages the compute resource
  - Condor schedd acts as the job queue for users jobs
  - Startd and schedd advertise the resource and jobs respectively to the collector using condor classAds
  - Condor negotiator acts as a match maker between compute resources and user jobs
- glideinWMS Factory
- VO Frontend
- All network traffic is authenticated and integrity checked
- Support pseudo-interactive monitoring out of the box



glideinWMS Implementation of the Pilot Paradigm

- glideinWMS is based on Condor with the VO Frontend and the Factory sending pilot jobs (i.e. glideins) to the grid sites.
- Condor as a user job WMS
  - Condor collector acts as an information database
  - Condor startd manages the compute resource
  - Condor schedd acts as the job queue for users jobs
  - Startd and schedd advertise the resource and jobs respectively to the collector
  - Condor negotiator acts as a match maker between compute resources and user jobs
- glideinWMS Factory
  - Creates and submits pilot jobs to the grid sites using CondorG
  - Condor collector acts as a dashboard for message exchanging
  - Factory receives orders from the VO frontend via the dashboard
- VO Frontend
  - VO frontend monitors the CondorWMS and regulates the number of pilot jobs sent by glidein factories via the dashboard
  - Frontend acts as a match maker for the glideins
- All network traffic is authenticated and integrity checked
- Support pseudo-interactive monitoring out of the box

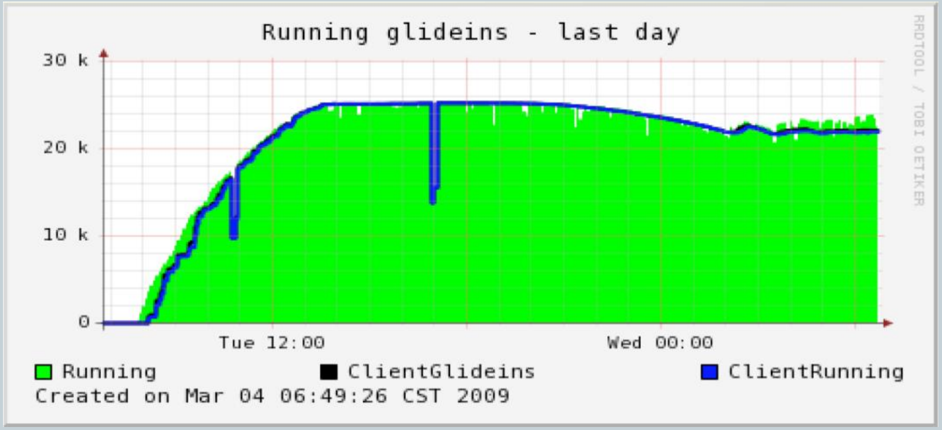


Scalability of glideinWMS

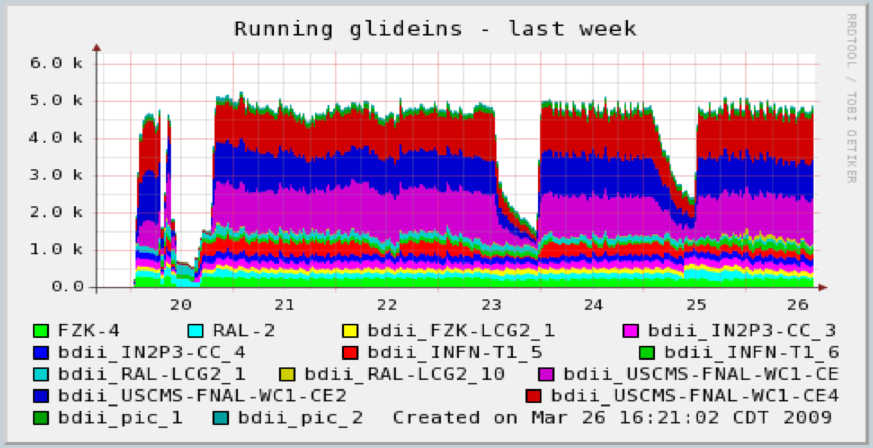
- Centralized WMS are generally less scalable
- glideinWMS scalability issues found
  - The centralized user queue keeping track of thousands of running jobs is memory exhaustive.
  - Security handshake in establishing communication between different components could be expensive in case of high network latency
- glideinWMS addresses these scalability issues by
  - Deploying multiple instances of the user queue service to spread the load
  - Increasing the memory of the machine that hosts schedd service
  - Deploying multiple slave collectors to reduce the impact of communication issues because of high network latency
- Table below summarizes the scalability achieved with a deployment running 1 Master collector, 70 slave collectors and using system with 16GB of memory to host the schedd service.

Criteria	Design goal	Achieved so far
Total number of user jobs in the queue at any given time	100k	200k
Number of glideins in the system at any given time	10k	~26k
Number of running jobs per schedd at any given time	10k	~23k
Grid sites handled	~100	~100

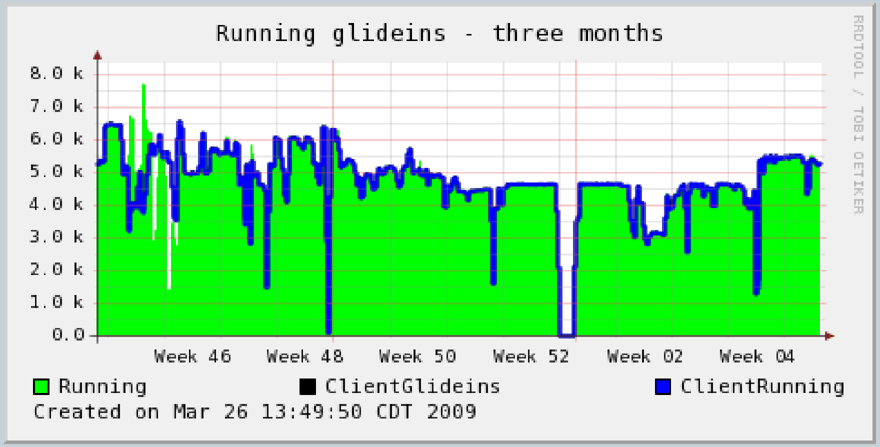
glideinWMS in CMS Operations



Running more than 20k glideins at any given time



CMS operations using glideinWMS at it’s seven archival storage sites



CMS operations at Tier1 site at Fermilab

Summary

- Grid computing provides large computing power at the expense of complexity for the users
- Pilot based WMS can alleviate some of the complexity by forming virtual private pool of compute resources for the users
- glideinWMS is a pilot WMS based on Condor with a thin layer of software on top of Condor to send pilot jobs
- The current release of glideinWMS has been tested to handle > 20k batch slots from ~100 Grid sites with ~200k jobs in the queue with average job start up rate of 1Hz

Acknowledgements

- glideinWMS infrastructure is developed in Fermilab in collaboration with the Condor team from Wisconsin and High Energy Physics experiments like CMS, CDF and MINOS.
- Most of the glideinWMS development work is funded by USCMS (part of CMS) experiment.
- Fermilab is operated by Fermi Research Alliance, LLC under Contract No. DE-AC02-07CH11359 with the United States Department of Energy.
- The paper was partial sponsored by the National Science Foundation under Grant No. PHY-0533280 (DISUN) and PHY-0427113 (RACE).
- The Open Science Grid (OSG)

References

- glideinWMS Homepage: http://www.uscms.org/SoftwareComputing/Grid/WMS/glideinWMS
- glideinWMS publications and presentations: http://www.uscms.org/SoftwareComputing/Grid/WMS/glideinWMS/doc.html
- Fermi National Accelerator Laboratory: http://www.fnal.gov
- The Open Science Grid: http://www.opensciencegrid.org
- EGEE Homepage:

  http://www.eu-egee.org