GlideinWMS: making life easier on the grid

Burt Holzman

Fermi National Laboratory

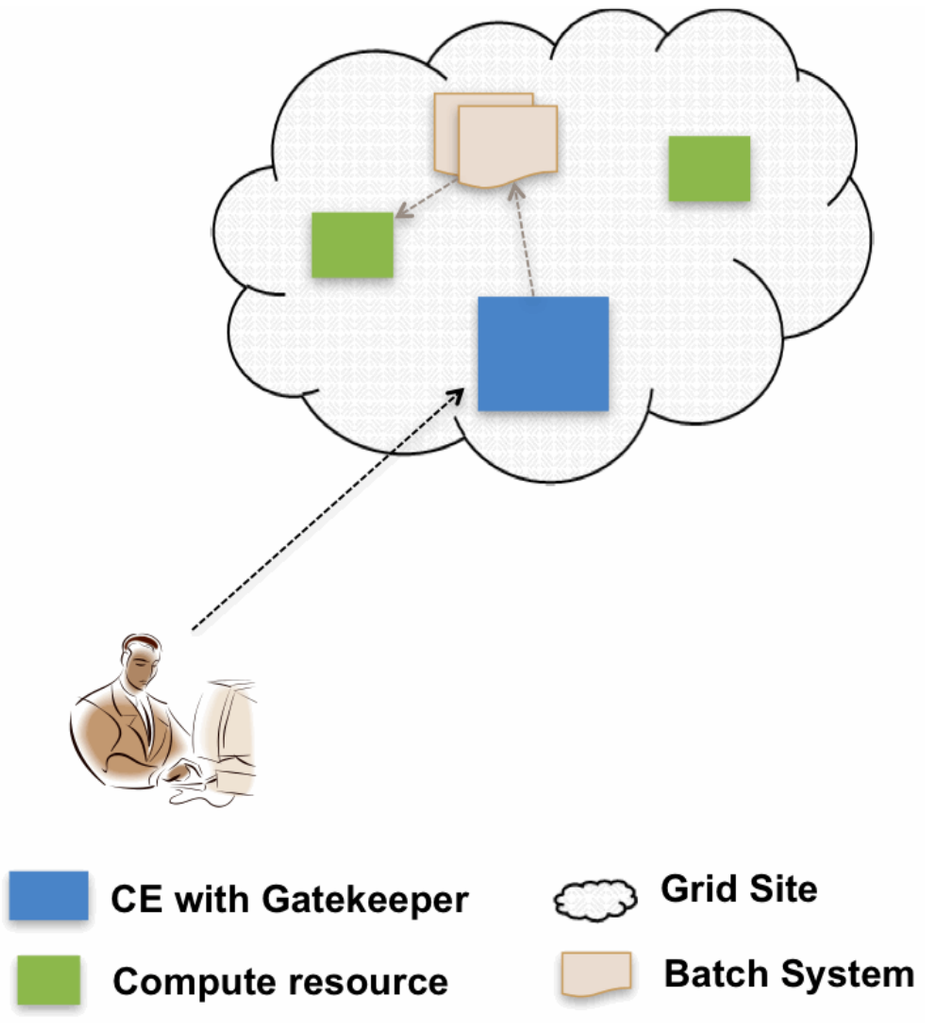
burt@fnal.gov

Fermilab

HTPC on the Grid

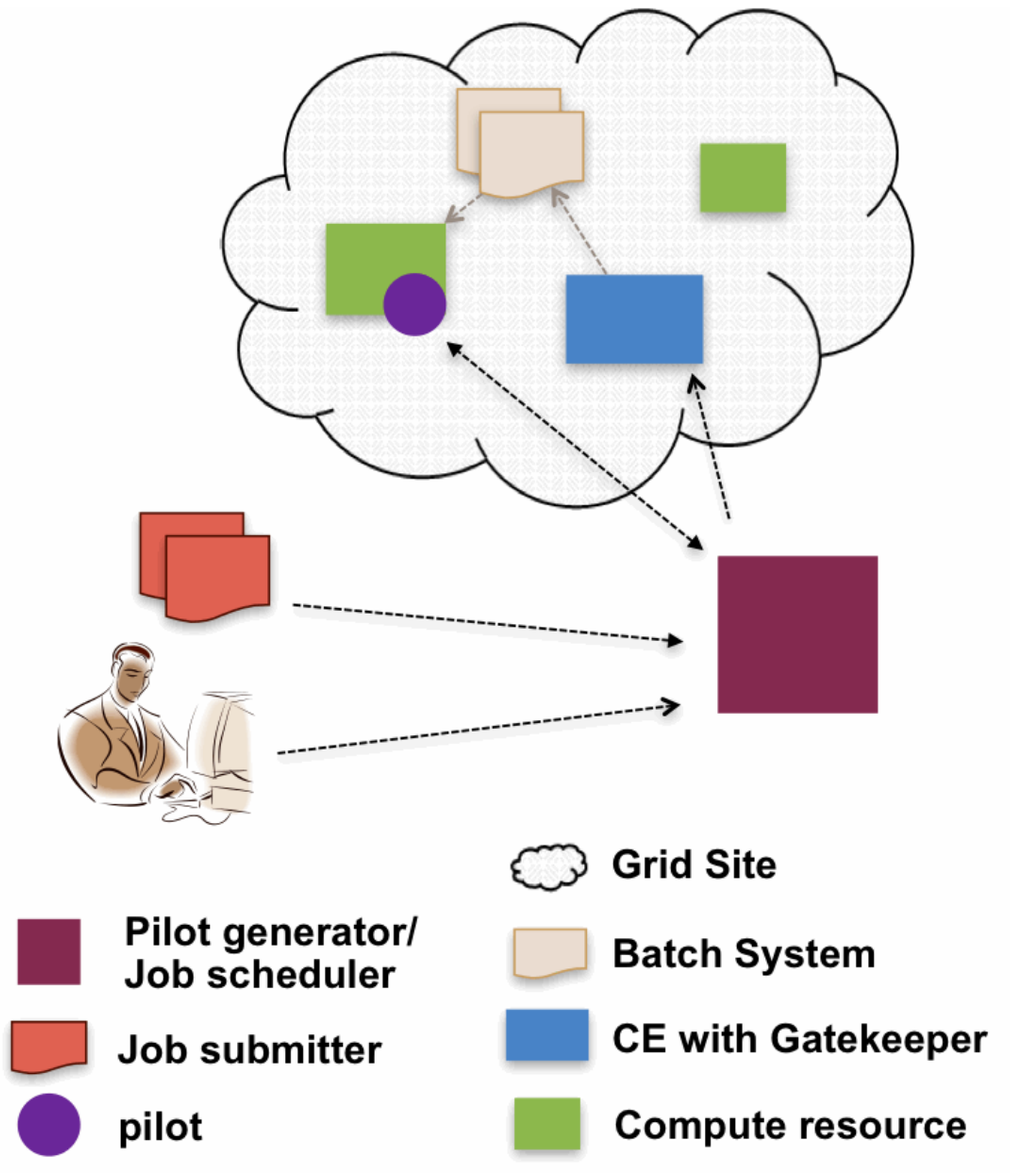
User has access to a large pool of resources, but

- Middleware has problems managing jobs
- Monitoring jobs is complicated
- Heterogeneous grid resources can cause issues
- Queueing and scheduling delays
- Software overheads and scheduling policies



Pilot Based Workload Management Systems

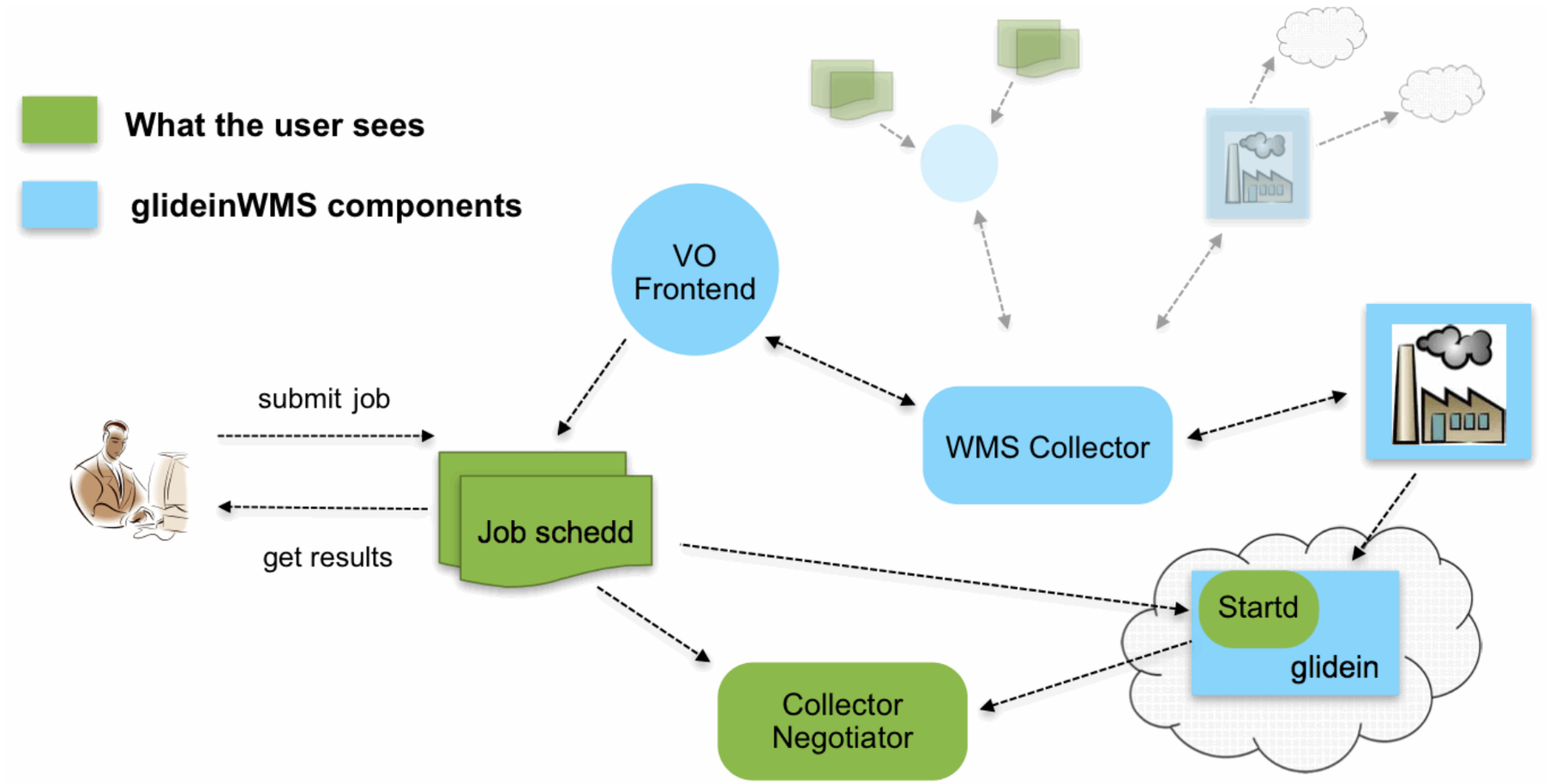
- Pilot generator submits pilots to the grid sites
- Pilots start running on the compute resources before the user job
  - Pilot can run several checks
  - Hides some diversity of grid resources
  - Overlays personal cluster on top of the grid
- Pilots fetch user jobs from a scheduler and execute
- Issues with scalability
  - Central queue can be resource intensive
  - Security handshake can be expensive



glideinWMS

- glideinWMS a thin layer on top of Condor
- Uses glideins (i.e. pilot jobs)
  - a glidein is a Condor Startd submitted as a grid job
- All network traffic authenticated and integrity checked
- Pseudo-interactive job monitoring is included
- Scalability considerations
  - Multiple user queues can spread the load
  - Increased memory of the machine hosting the schedd service
  - Multiple slave collectors can reduce communication issues

glideinWMS (II)



- Glidein Factories know about grid sites, how to submit glideins
- VO Frontends know about job details, number and kind of glideins needed
- Factories and VO Frontends communicate through common (Condor) WMS Collector

Why?

We want to off-load the "hard part" of grid computing to a glideinWMS factory as much as possible – hetereogeneity of grid sites, lack of late-binding validation, bottlenecking at grid gatekeepers. The user sees a local virtual batch system.

We want to put power in the hands of the VOs to allow them to prioritize their workflows rather than leave it at the mercy of a remote grid system administrator.

glideinWMS in production

- Active factory deployments
  - UCSD (CMS Analysis)
  - UCSD (OSG-supported)
  - Fermilab (CMS Production)
  - Fermilab (CDF)
  - Fermilab (DZero)
  - CERN (CMS testbed)
  - USC/ISI (Corral deployment)

CorralWMS project

- CorralWMS integrates a Corral Frontend component into the glideinWMS software project
- Corral project has close ties with Pegasus
- Scope is wide; includes compatibility with OSG and Teragrid and scientific and commercial clouds
- Outreach & education will be ongoing: we will be actively recruiting domain users this year, HPC-style applications obviously welcome!

glideinWMS: some things coming up

- "Sky computing" / cloud submission
  - We have successfully executed jobs via the glideinWMS framework to Amazon EC2. We are working to integrate this in a general fashion into the existing infrastructure. We will continue to work with Magellan on utilizing their resources. We also need to understand stakeholders' use cases for the cloud.
- Inclusion in the VDT
  - The VDT will include glideinWMS as an option in their software repository. When feasible, we will provide native (RPM) packaging for a VO-friendly frontend preconfigured to work with OSG-supported factories