glideinWMS

Stakeholder Meeting

February 19, 2014

Burt Holzman

Challenges of Grid Computing: Distributed Compute Resources

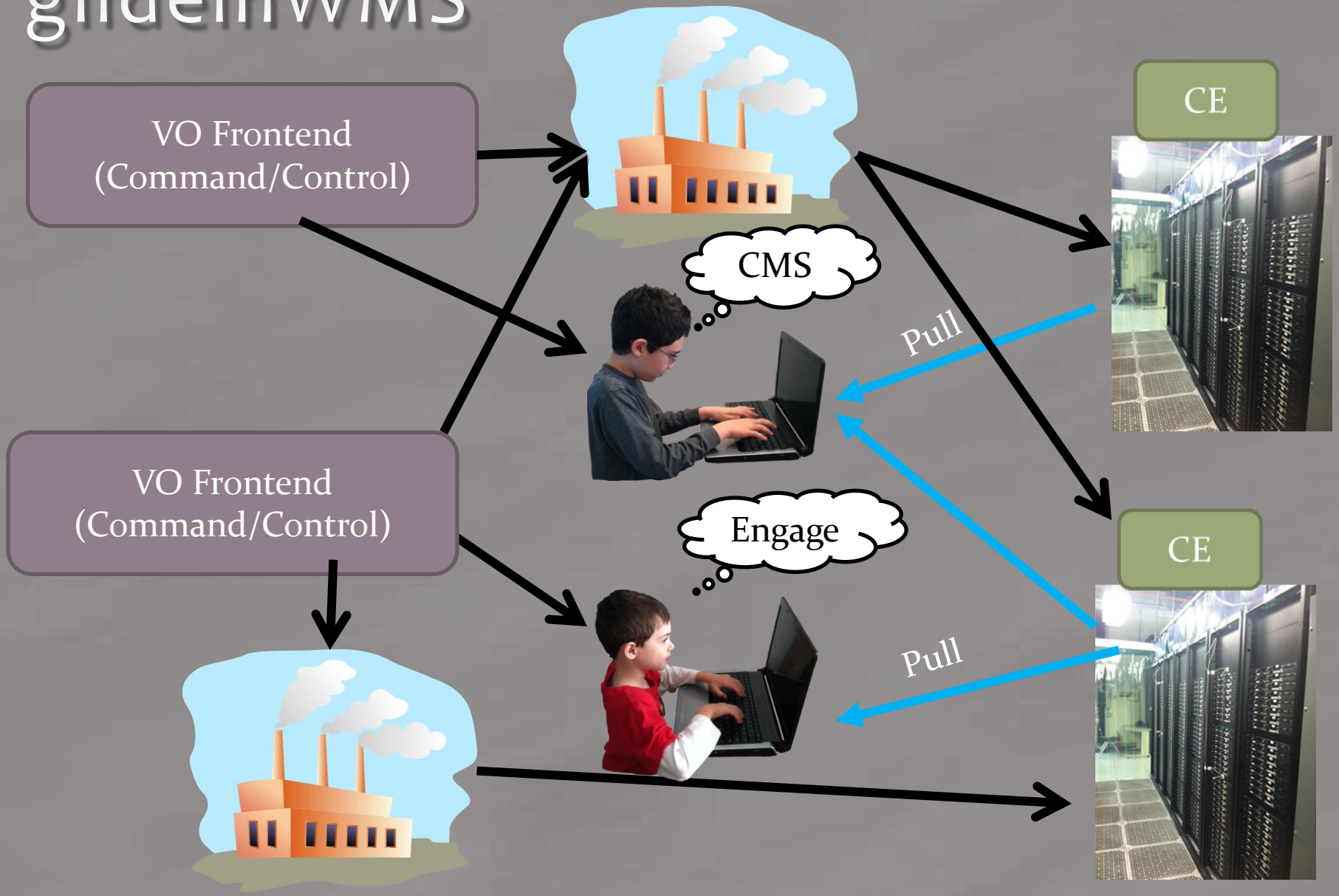
3 Compute Elements are manageable By hand

Challenges of Grid Computing: Distributed Compute Resources

?

We need middleware – specifically
a Workload Management System
(and more specifically, "glideinWMS")

glideinWMS



VO Frontend can talk to multiple factories

glideinWMS: quick facts

- glideinWMS is an open-source Fermilab Scientific Computing Division product driven by CMS
- Heavy reliance on HTCondor from UW Madison and we work closely with them
- http://tinyurl.com/glideinWMS
- Effort:
  - Burt Holzman (.2 US CMS)
  - Anthony Tiradani (.25 US CMS)
  - Marco Mambelli (.5 US CMS)
  - John Weigand (.5 US CMS – Contractor)
  - Parag Mhashilkar (.5 SCD – was Corral)
  - Krista Larson (US CMS)
  - Mats Rynge (ISI/USC)
  - Derek Weitzel (UNL)
  - Igor Sfiligoi (US CMS)

Since we met last

- V3.2.1 release – October 30
  - XSLT transforms
  - Log cleanup improvement (decoupled from main loop, improved logic)
  - Improved frontend logging
- V3.2.2 release – November 8
  - Patch to support new ARC CE state
  - Bug fix for factory crashes
- V3.2.3 release – February 3
  - Site batch slot tracking
  - Configurable log rotation
  - Bug fixes for renewable credentials, partitionable slots, log cleanup, performance improvements
- V3.2.4 release – early April – so far:
  - Updated JavascriptRRD
  - Some frontend aggregation in monitoring

Support

- Support Mailing list: glideinwms-support@fnal.gov
- Issues are tracked in the redmine issue tracker
  - Categorization and prioritization based on impact, urgency, and requester
  - Issues are assigned based on developer's expertise and other workload
  - Entire development team is responsible for support
- Development list: glideinwms@fnal.gov
  - Anyone is welcome to join/participate, although discussions are generally technical
- Working with Fermilab Computing Sector Office of Project Management for project support

From last time: gWMS milestones for next release

- Better prevention of "black hole" workers
  - Periodic validation, exponential backoffs, tunable thresholds for job starts
- "Why is my job not running"?
  - New monitoring plots, frontend level tools ("why does my job not invoke glidein requests")
- Frontend scalability
  - Queries to condor schedulers scales (at least) linearly with the number of frontend groups – for busy schedulers these queries are very expensive
- Factory/frontend Configurability
  - We need to implement better hierarchical configuration options to simplify the operational load of configuring a factory and frontend
- Aggregrate monitoring
  - We need to pull together the monitoring across multiple factories, and across multiple frontends.

Prevention of blackhole workers

- Issue: https://cdcvs.fnal.gov/redmine/issues/4988
- Commonly identified 3 failure modes. Each requires different solution.
  1. Insufficient validation of worker nodes
  2. Worker nodes start experiencing problems after job starts
  3. Failures specific to type of user jobs
- Current scheme
  - Use validation/custom scripts to identify potential problems before glidein starts HTCondor daemons
  - Limitation: Problems can creep in after job starts.
    Example: Required mounted partitions (like CVMFS) becomes intermittently unavailable.

Prevention of blackhole workers

- Proposed schemes
  - Insufficient validation:
    - Requires VO input
    - Smarter validation scripts
  - Nodes failing after validation:
    - Run some type of validation scripts periodically (HTCondor cron)
    - More complex matching and start dynamics (but this wastes cycles)
    - Out-of-band notification to site

“Why is my job not running”?

- https://cdcvs.fnal.gov/redmine/issues/4989

- Working on a tool – functionality similar to ‘condor_q -analyze’
- Currently requires access to VO frontend configuration
  - Executes VO frontend logic to identify
    - Jobs to analyze
    - Available factory entry points (aka sites) where to request glideins
  - Correlates the information to detect possible anomalies like:
    - Are there any sites satisfying given constraints to request glideins?
    - Is frontend requesting enough glideins?
    - Are we over the threshold?
    - Are requested glideins starting on the site?
    - Do glideins that start report back to the collector?
    - […]

Frontend scalability

- Queries to condor schedulers scales (at least) linearly with the number of frontend groups – for busy schedulers these queries are very expensive
  - Implement fork & collect with locks – attacks symptom, but not disease. Do we need to do better?

Factory/frontend Configurability

- Factory/frontend Configurability
  - We need to implement better hierarchical configuration options to simplify the operational load of configuring a factory and frontend
  - We have selected a scheme for simplifying complex match policies – changes should be backwards-compatible, targeted for the next major release.
  - We are evaluating a few different proposals for factory configuration; all are easier if we can phase out SL5 support so we can use modern XML parsing.

Aggregate monitoring

- We need to pull together the monitoring across multiple factories, and across multiple frontends.
- Proposed: a server that takes the hostnames and URLs of existing monitoring and aggregates the output.

What else is planned?

- Simplifying operations – decoupling and decentralizing "site support" functions from day-to-day operation of a glideinWMS factory

- Improvements to provisioning of "allocation-based" resources (HPC, Cloud)

Stakeholder Input