glideinWMS

Stakeholder Meeting

October 30, 2013

Burt Holzman

Challenges of Grid Computing: Distributed Compute Resources

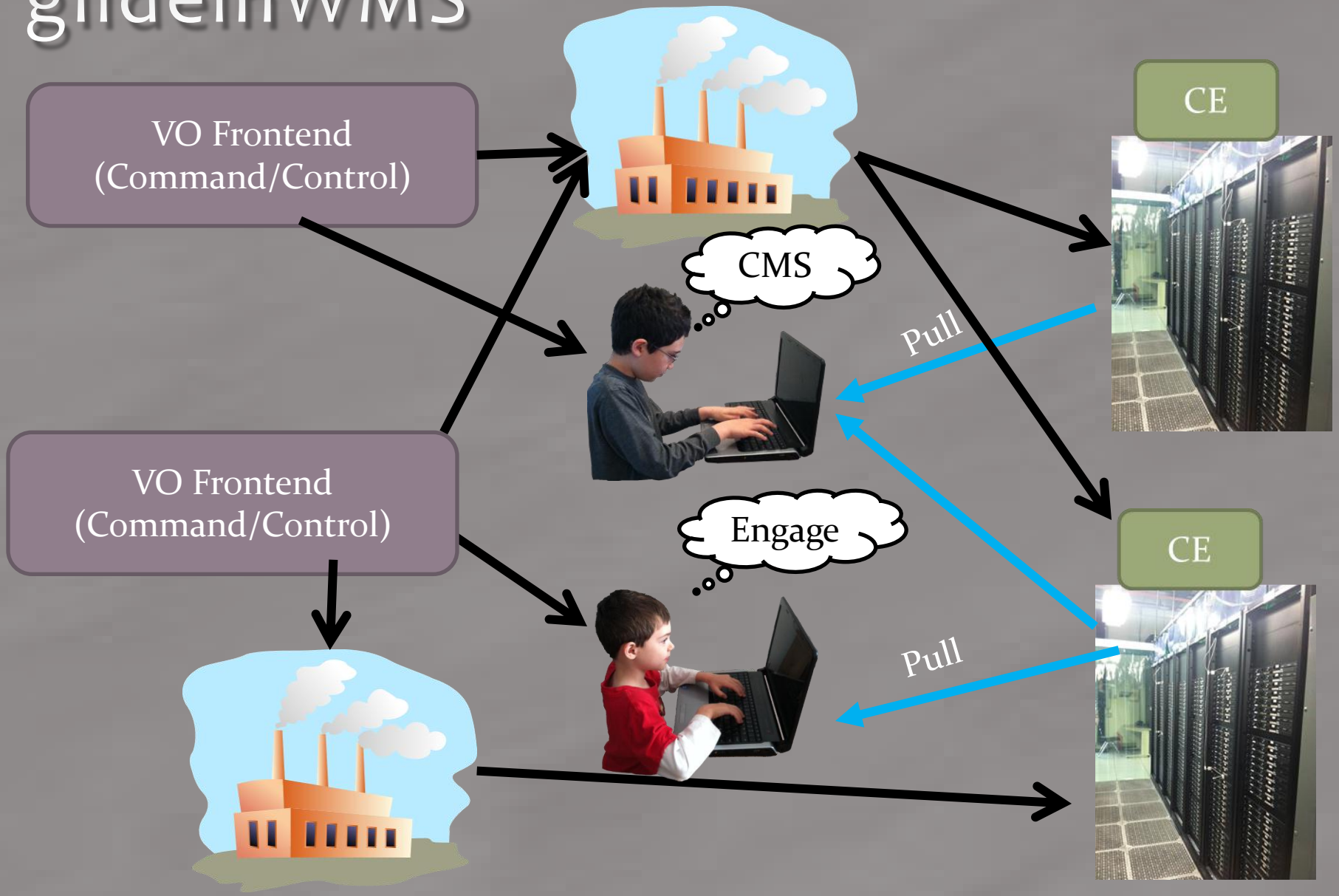
3 Compute Elements are manageable By hand

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

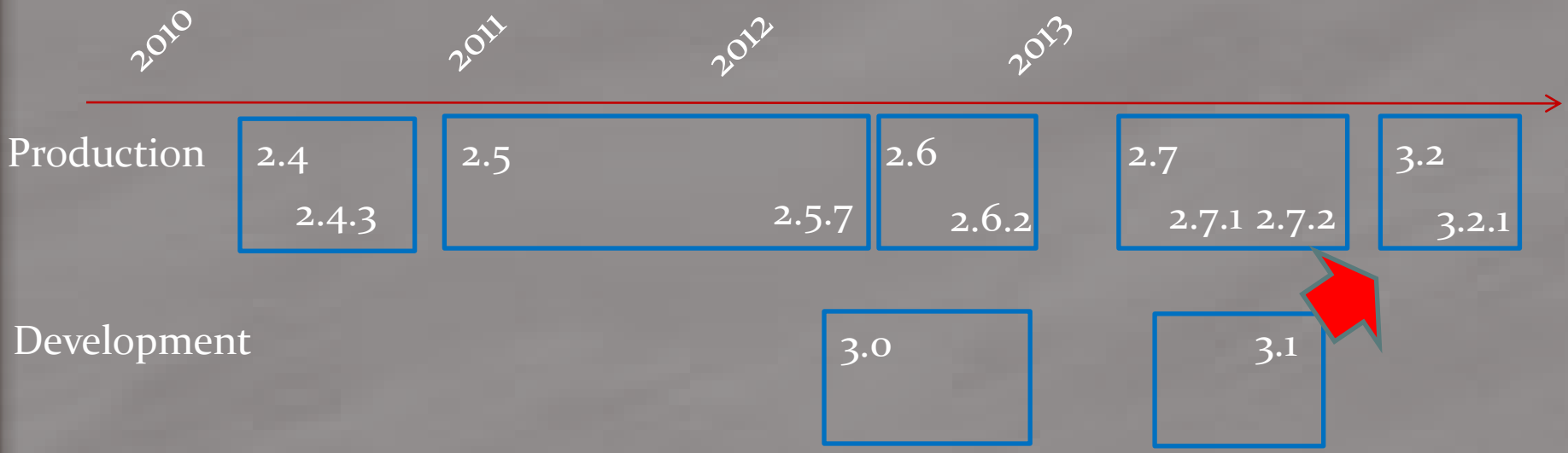
glideinWMS: quick facts

- 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

- V2.7.2 release – Sep 10
  - Bug fixes, all correlated with race conditions on reconfig
- V3.1 release – Aug 1
  - First v3 production release!
- V3.2 release – Oct 10
  - Tons of new bug fixes – thanks to all factory and frontend operators for helping to test at scale with many configs
- V3.2.1 release – early next week
  - XSLT transforms
  - Log cleanup improvement (decoupled from main loop, improved logic)
  - Improved frontend logging

glideinWMS: version timeline



- 2.4.x: privilege separation, aggregate monitoring, glexec control, glidein lifetime control
- 2.5.x: HTCondor TCP bulk updates, efficiency improvements, factory limits per frontend, excess glidein removal, shared ports, better user pool matchmaking
- 2.6.x: Better multislot support, ARC CE, more glidein lifetime controls, factory limits per frontend security class
- 2.7.x: Refactor for factory scaling, performance fixes, partitionable slot support

- 3.x: Cloud support, CorralWMS frontend support

Action items (1/2)

- https://cdcvs.fnal.gov/redmine/projects/glideinwms/wiki/StakeholderInfo
  - Describe what is included in the next major gWMS release
    - Redmine roadmap to gWMS 3.3: http://tinyurl.com/gwms33
      (this will be revised a week or two after this meeting)
  - ensure good communication of plans and activities between cycle computing monitoring NDA and GWMS project
    - Dialog was initiated between the UCSD/Cycle activity and the project.
  - request capability from HTCondor for differing types of class ads to be sent to different collectors.
    - This capability was requested from HTCondor. A solution via COLLECTOR_REQUIREMENTS was suggested; we need to evaluate to see if that's suitable for the use cases.

Action items (2/2)

- https://cdcvs.fnal.gov/redmine/projects/glideinwms/wiki/StakeholderInfo
  - include more specific info on the support functions/feedback/effort etc next time
    - See following slide
  - thinking on how to present the separate functions and components of Scheduling and Factory concepts in GlideinWMS more clearly.
    - We are thinking about it – but nothing concrete yet apart from our current approach
  - for or before next qtrly meeting have meeting of stakeholders to discuss control and policy mechanisms, options, futures in the factory and/or front end.
    - We weren't ready for this yet, but we could do this before the next meeting.

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

Proposed Plan for next glideinWMS 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.

Stakeholder Input on plan