CMS multicore pilot model and its implications on accounting

2nd Accounting TF meeting

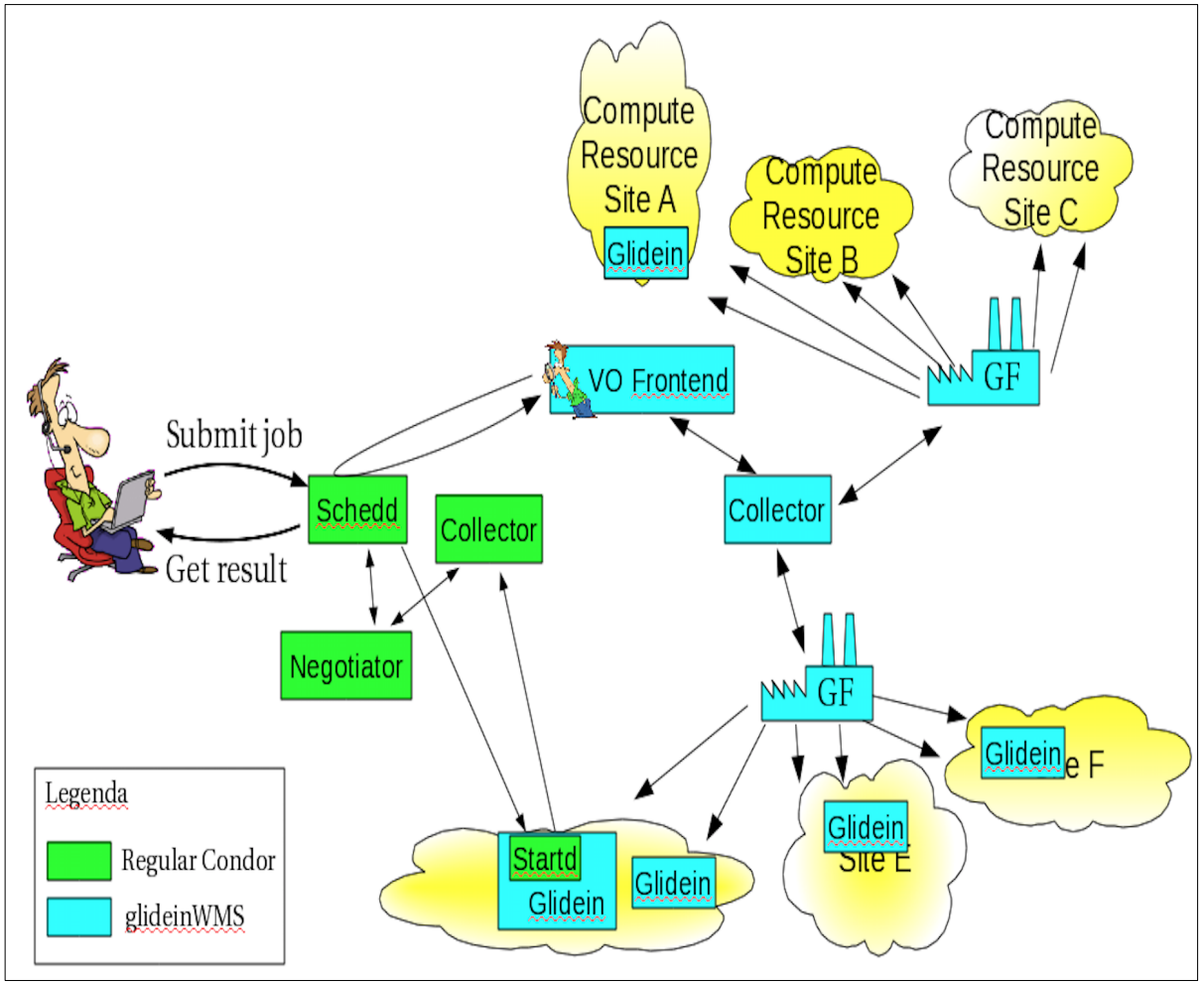
Antonio Pérez-Calero Yzquierdo,

06/23/2016

Ciemat
Centro de Investigaciones Energéticas, Medioambientales y Tecnológicas

PIC
port d'informació científica

CMS Submission Infrastructure



GlideinWMS is the main tool used to create a (**HTCondor**) virtual pool of resources (***pilots***)

Two stages of matchmaking

- Request to site resources: **launch new pilots** based on demand

- **Jobs to running pilots**, based on matching classads

A single **global pool** handles all jobs, centralized production workflows and analysis tasks

CMS model to schedule multithreaded jobs

- CMS, like other VOs needs to run, but not only, **multicore** jobs!

- **Single-core** still needed for
  - **auxiliary tasks** (e.g. output merge, log collect, etc)
  - **analysis jobs**

- CMS model: use a **single type of pilot to schedule and run all types of jobs**

  - Evolves also from the **CMS global pool** idea: analysis and centralized production jobs running in the same pilots

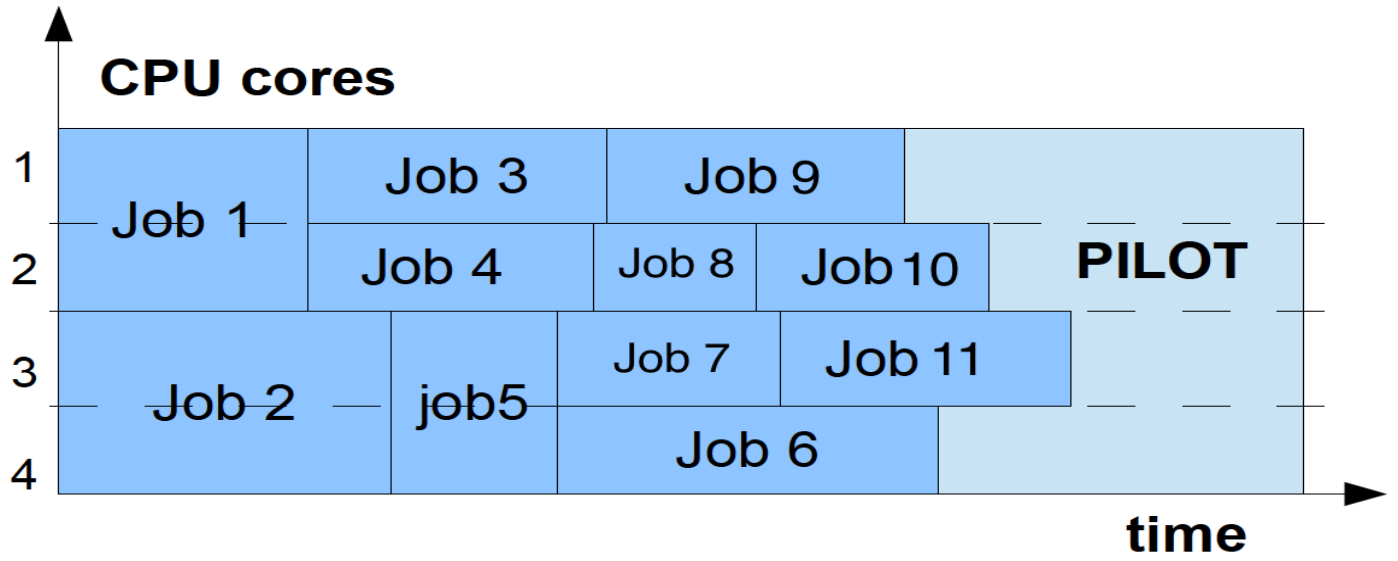
  - CMS pilots can run **payloads from multiple users** (glexec)

  - Allows for maximum **flexibility of the use of resources in control of the VO**

CMS model to schedule multithreaded jobs

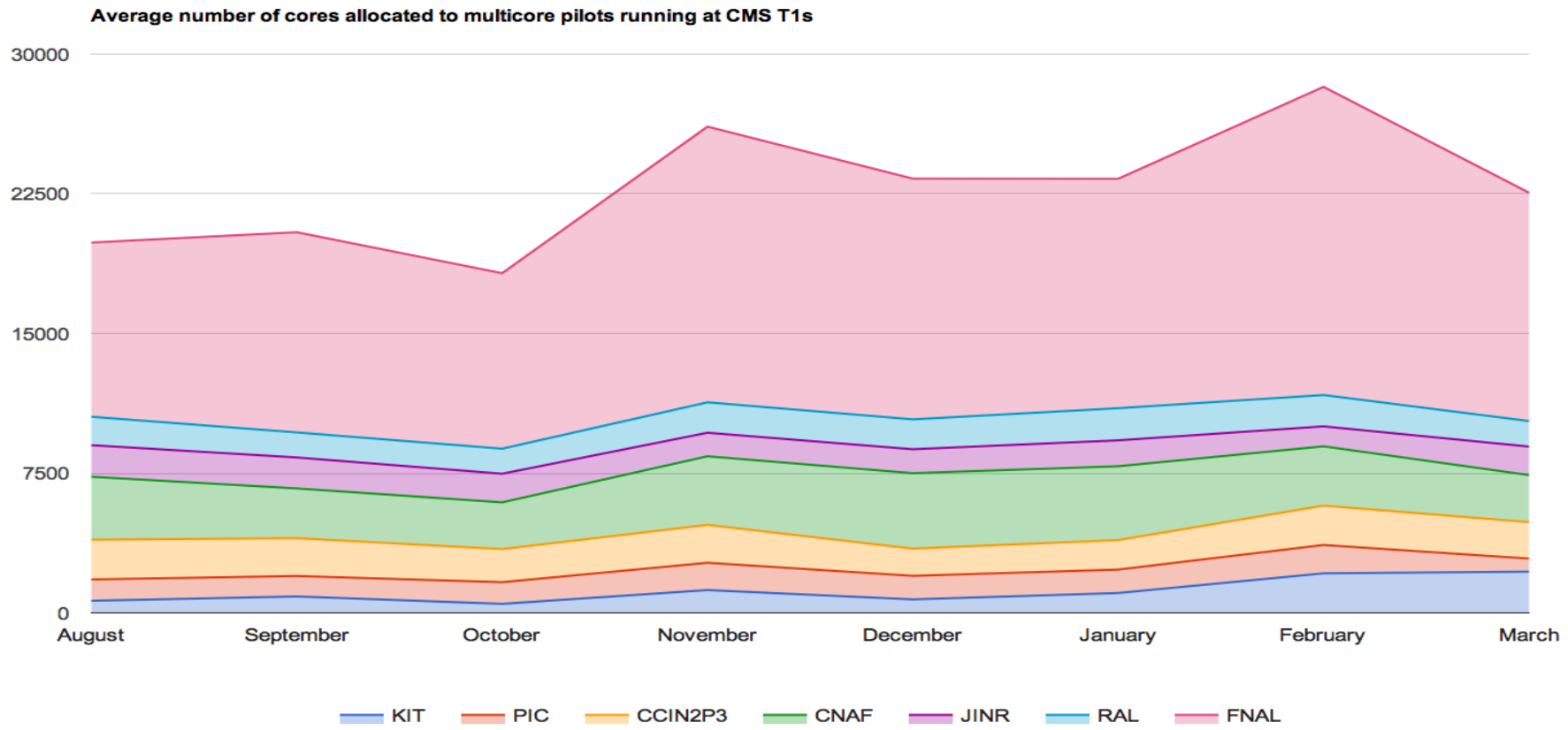
- Main tool : **multicore pilot** with dynamic partitioning of resources
  - Inherited from HTCondor **partitionable slots**
  - Enables **common allocation of single and multicore jobs**
  - Enables use of the resources in a **flexible way at the discretion of the VO**:
    - e.g. high memory tasks with no requirements from the sites

Advantage to sites: **a single unified request for all tasks** and standard across sites



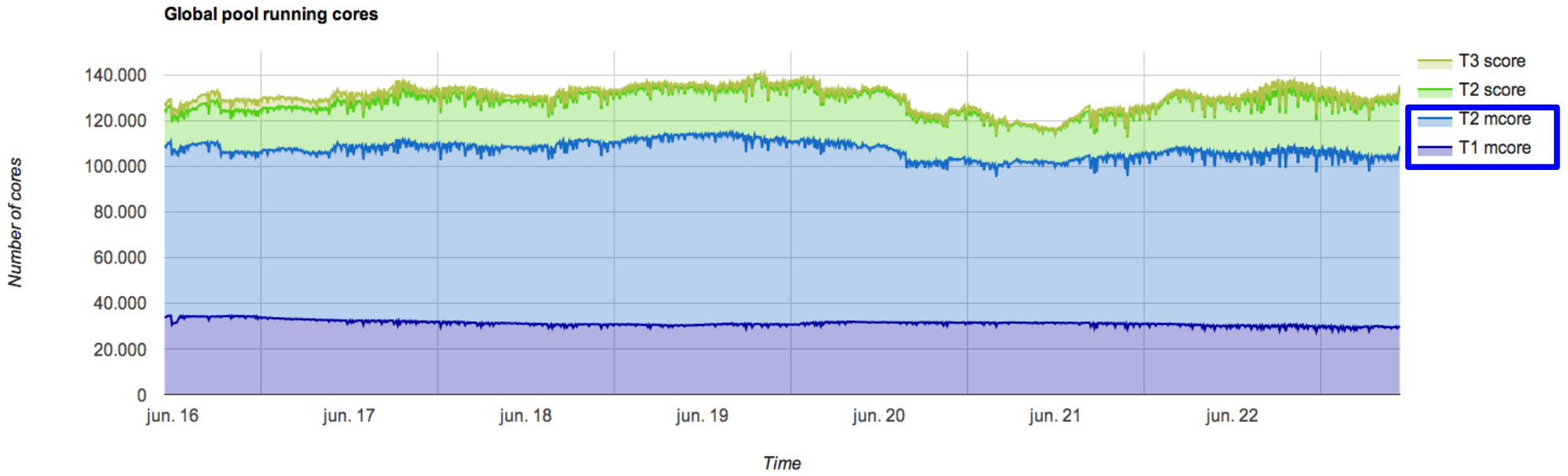
Multicore at T1s

- Deployment and use of multicore resources at CMS **Tier-1s started in 2014**
- Stable use, and increasing through 2015 and 2016
  - Finished transition to fully mcore (KIT & JINR)
  - Increased CPU pledges for 2016



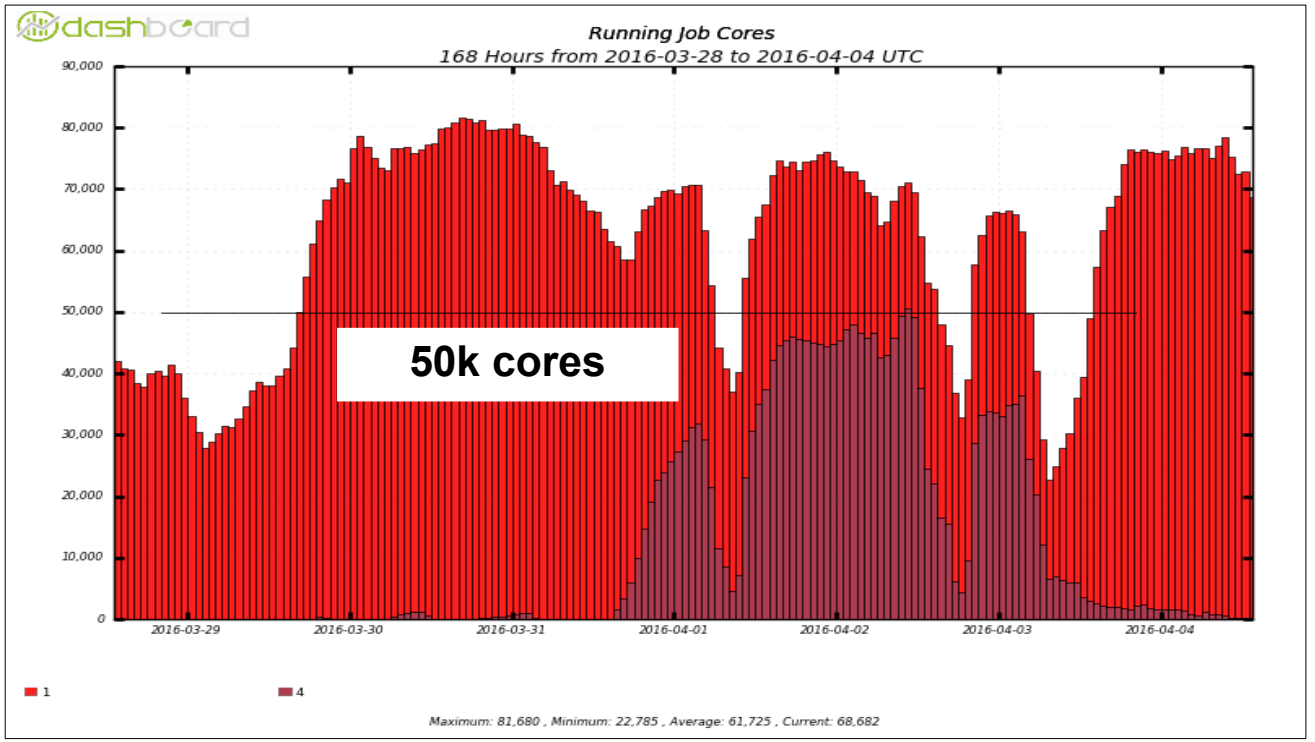
Deployment of multicore to CMS T1+T2 sites

- Progression of CMS multicore pilot deployment over the last months:
  - **T1s use stable at about 35k cores (all of them)**
  - **T2s use increasing up to 80k cores (about 30 sites)**
- The CMS global pool is now running **~85% of the resources as multicore pilots**
  - Continuously and **regardless of the type of payload**, single or multicore



Multicore payloads

- Readiness for **running multicore jobs at T1s demonstrated in tests since early 2015**
- 2015 end-of-year **data reprocessing** executed as **multicore jobs** (4 cores) in **8-core pilots**
- Since then, CMS finished adapting **MC generation** software **to efficiently run multithreaded**
  - **Now data and MC can be processed as multicore**
  - But not used at scale yet: multicore pilots filled with single core jobs



CMS model and accounting

Scheduling of payloads, being internal to pilots, involve stages which are hidden from sites, which only detect (are account for) the overall net effect

=> Payload accounting (dashboard) can't be compared to site reports or EGI accounting

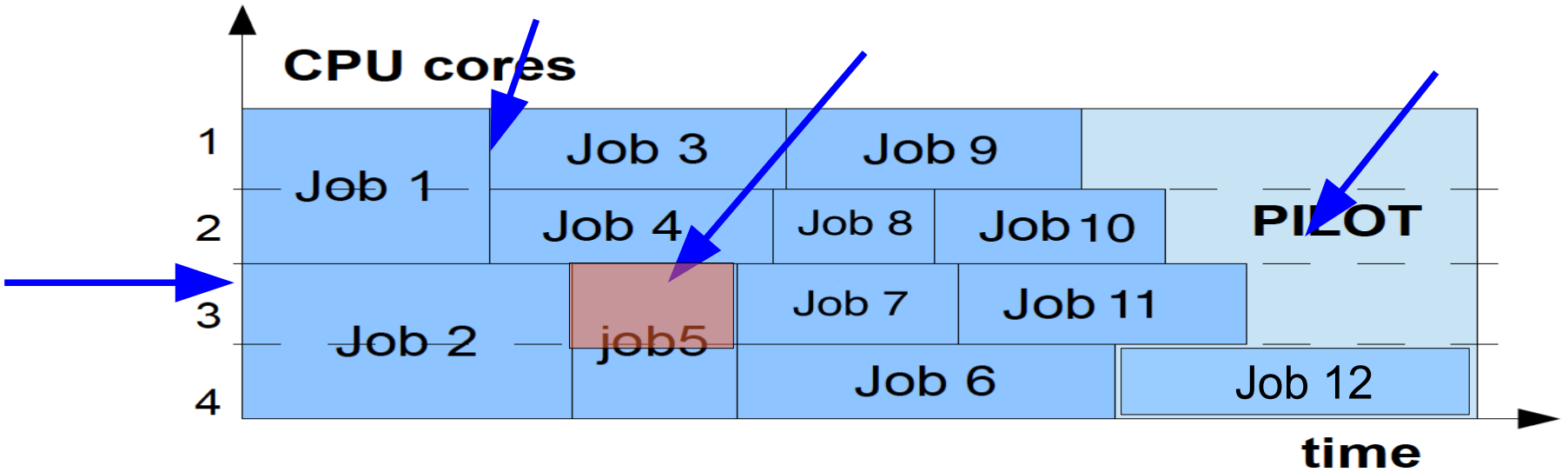
Some effects:

• **Pilot start**: pilot env. validation tests, contact the global pool CM for slot-resource negotiation (payload assignment)

• **Matching of new payloads**: again, as each payload finishes

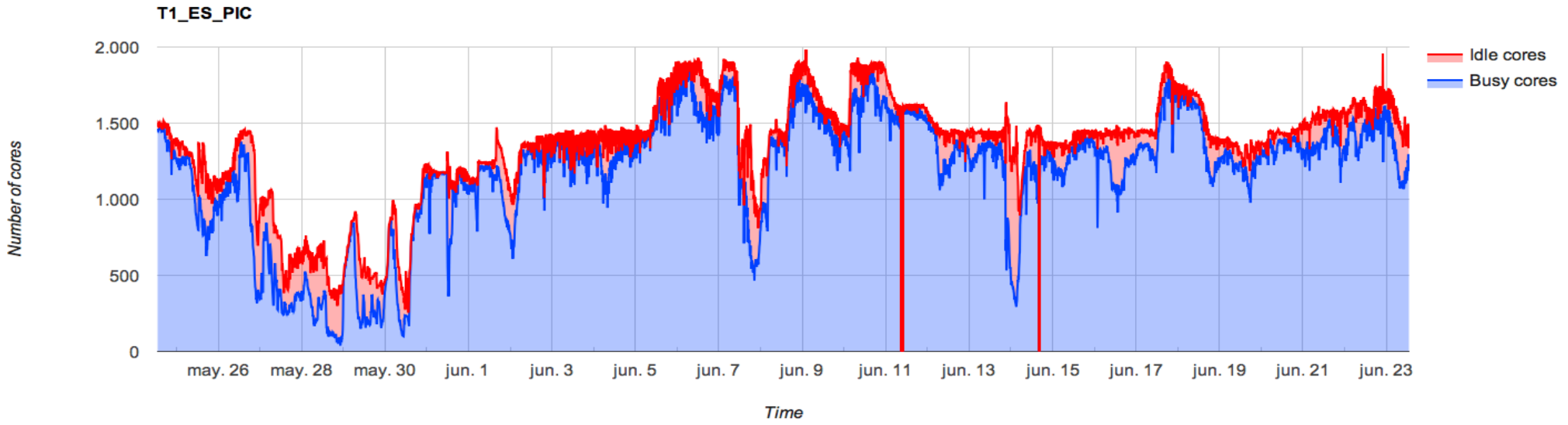
• **High resource request jobs**: Ex. payloads using more than 2 GB/core can be allocated to the ad hoc slot, even if using only one CPU

• **Pilot draining**: after a fraction of time, pilots stops accepting new payloads to finish in a clean way before reaching max allowed walltime



CMS model and accounting

- All these steps in internal payload scheduling have an effect on the net time the pilot was running payloads
- We don't have at the moment report per pilot; in discussion with HTCondor and GlideinWMS developers
- Statistically however, on average, ~90% of the cores are in use by payloads at any point in time
  - Walltime difference (sum of) payload vs pilots



Conclusions

- CMS experience with multicore pilots goes back to 2014, running them regularly at T1s since
- For 2016, CMS has deployed **multicore pilots to main T2 sites**
  - About 85% of global pool cores now used in multicore mode, ~110k cores
  - **Still filled with single core payloads for the most part**
- Effects in **internal scheduling of payloads is hidden to site view**
  - Pilot start, succession of payloads, high memory jobs, pilot draining, etc
  - **Dashboad (payload) measurements can't be directly translated to site (pilot) accouting**
- Effects on accouting still to be properly measured in terms of CPU time and walltime
  - CMS multicore pilots using about 90% of the cores at any point in time: direct effect on walltime