

# CMS readiness for multi-core workload scheduling

**Antonio Pérez-Calero Yzquierdo,**

**on behalf of the CMS Collaboration, Computing and  
Offline, Submission Infrastructure Group**

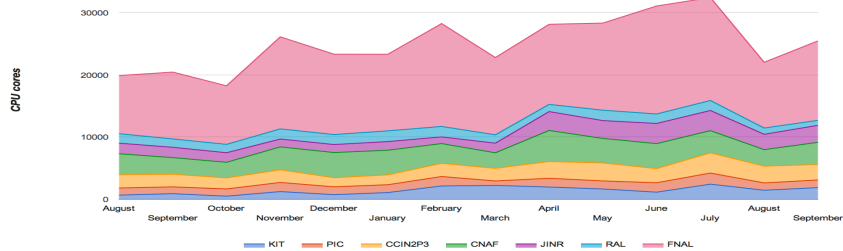
**CHEP 2016 San Francisco, USA  
October 10th, 2016**

**Ciemat**  
Centro de Investigaciones  
Energéticas, Medioambientales  
y Tecnológicas

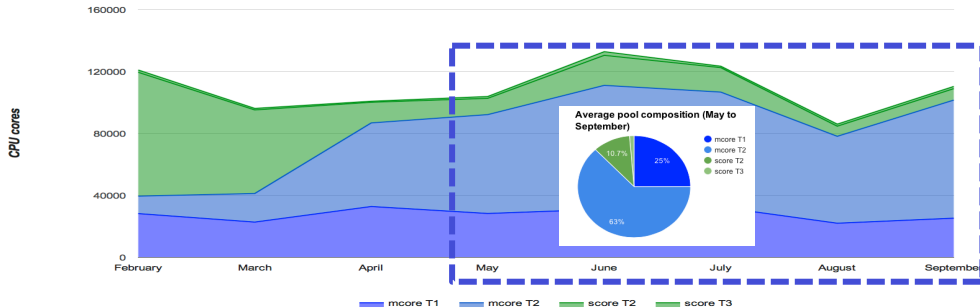


- CMS needs to run multicore jobs to overcome memory/core and time/event limitations and profit from current CPU architectures. However, many jobs will remain single core. CMS has chosen the approach of a single type of multicore partitionable pilot to allocate resources to both single and multicore tasks, profiting from the native capabilities of HTCondor and GlideinWMS
- The multicore pilot model provides increased job scheduling flexibility and control to CMS. However the scheduling inefficiencies typically observed at LRMS are passed into the VO responsibility. Monitoring and accounting of allocated and used resources is also more complex than in the case of one payload type per pilot models
- Multicore pilot deployment to CMS sites started to T1s in 2015. 90% of the CMS pledged resources, including T1s and big T2s, were moved to multicore by Spring 2016, covering a wide variety of CE and batch system technologies

Monthly average number of cores allocated to multicore pilots running at CMS T1s (2015/16)

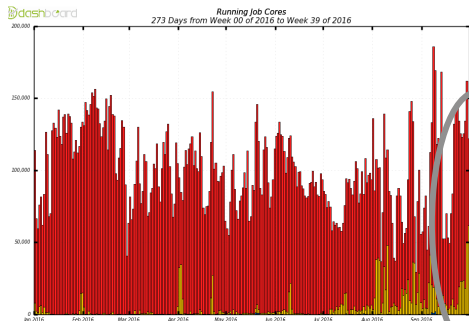


CMS global pool composition by type of pilot and Tiers (monthly averages 2016)

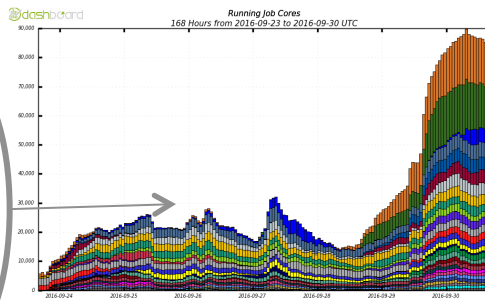


- Results: CMS is successfully running multithreaded jobs together with single core. The amount of CPU dedicated to data and MC reconstruction multicore jobs has been increasing along 2016, although MC simulation and analysis jobs remain mostly single core.
- Performance: Observed inefficiency is originated mainly by idle cores in drainage period near the end of pilot running time (5% of 150k pool cores) as well as available but unmatched slots (typically also ~5%), due to a variety of reasons affecting various elements in the job submission and resource allocation agents
- Those inefficiencies are continuously being reduced by means of a more efficient and faster job-to-pilot matchmaking, improved workflow management and job-submit components, as well as several strategies being worked on by the CMS Offline & Computing submission infrastructure team together with HTCondor and GlideinWMS developers teams

**Number of cores in running jobs by cores per job, during 2016**



**Number of cores in running 4-core data reco. jobs for last week of September 2016**



**CMS Global pool 24h view for 150k cores**

