

Tier-1

Resource Utilization and Performance

Andrea, Chris, Pepe, Peter, Isidro, José



Monitoring Tier-1s

- Monitoring more closely Tier-1 activity, performance and utilization
 - <http://indico.cern.ch/getFile.py/access?contribId=7&sessionId=2&resId=1&materialId=slides&confId=91930>
 - <https://twiki.cern.ch/twiki/bin/viewauth/CMS/CmsTier1MonitoringProject>
- Consolidate/integrate various monitoring tools/pages
 - Try to show in a single page monitoring information from the multiple monitoring sources
 - <http://asciaba.web.cern.ch/asciaba/tier1/monitoring.html>
- Review each Monday at the Operations Meeting



Tier-1 utilization & performance

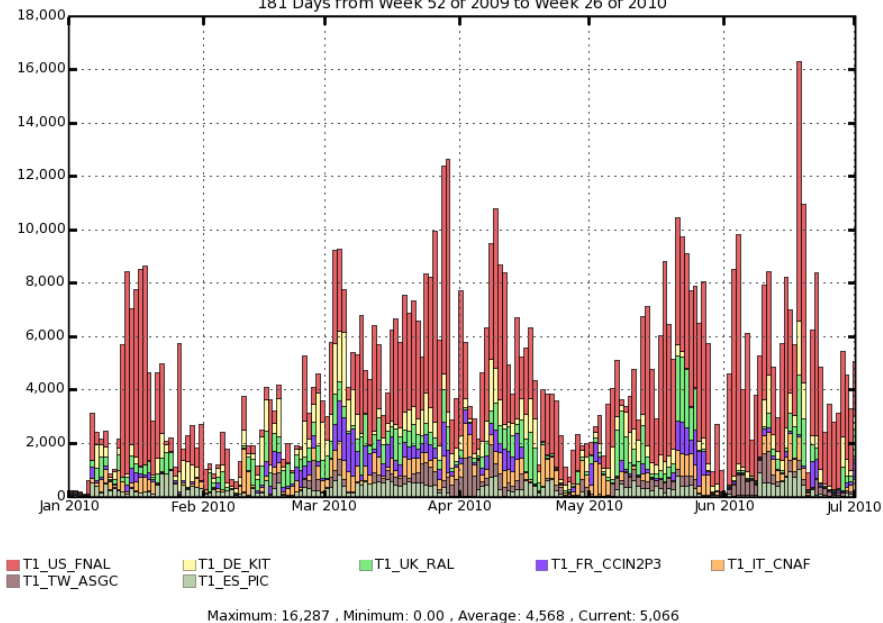
- Monitor
 - Utilization
 - Slot usage, processing share among sites, utilization level wrt pleges
 - Our dataset-based data placement does not guarantee processing balance
 - Performance
 - Job success rates, CPU efficiency
- Summary plots that should allow us to see at a glance how we are using our T1 resources
- Low utilization of the resources will be a convolution of several effects: site problems, WMS tools/operations inefficiencies, lack of processing work, imbalance of data, etc



Slot Usage

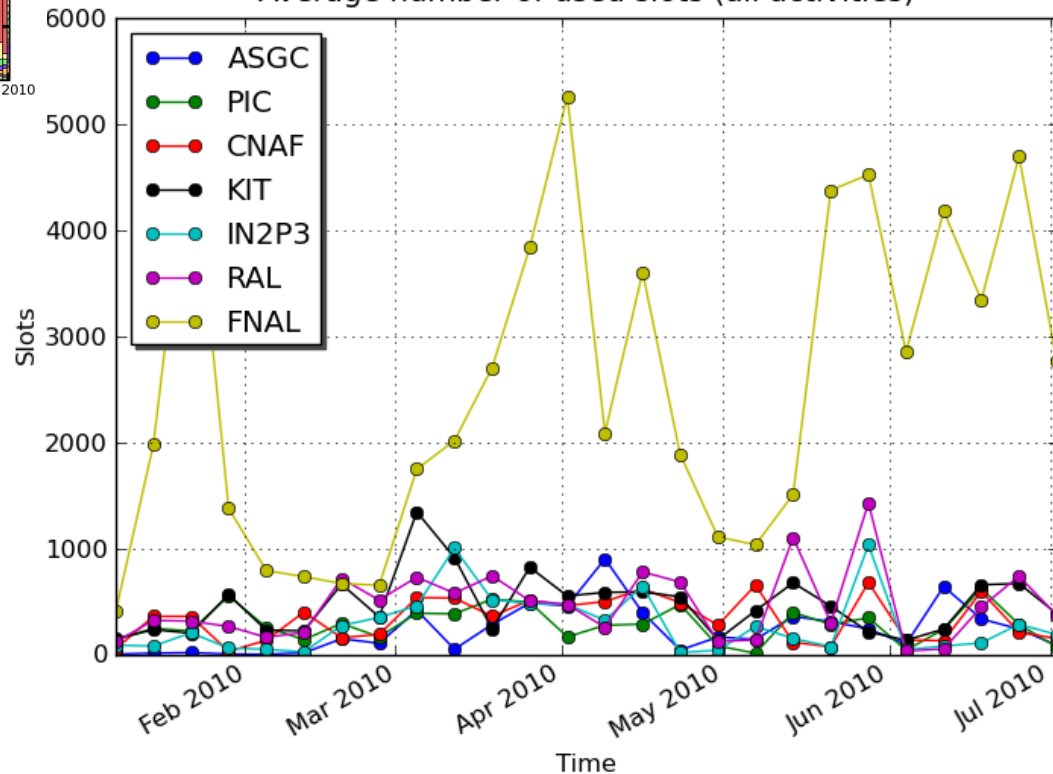
Running jobs

181 Days from Week 52 of 2009 to Week 26 of 2010



- Average number of slots occupied
- Does not tell how processing is shared or utilization level

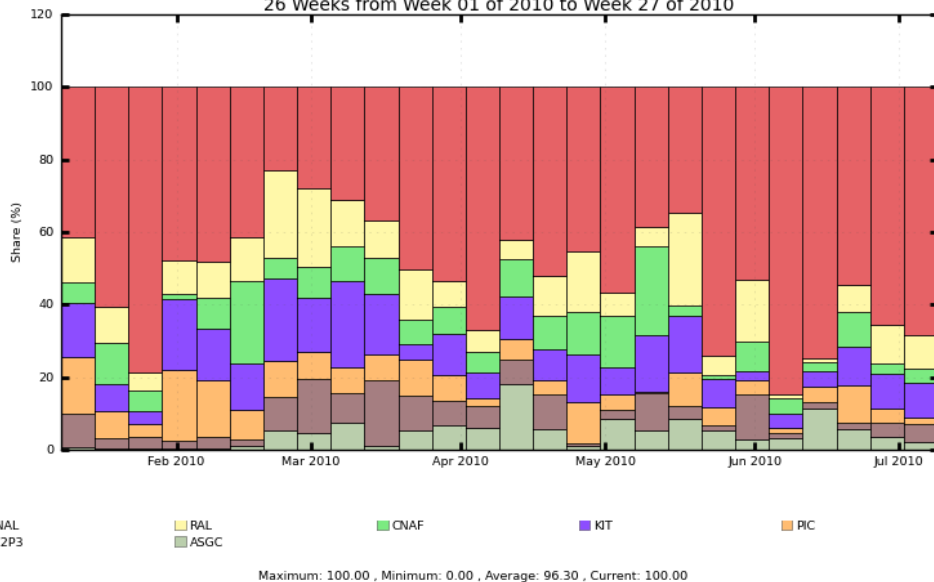
Average number of used slots (all activities)



Processing share

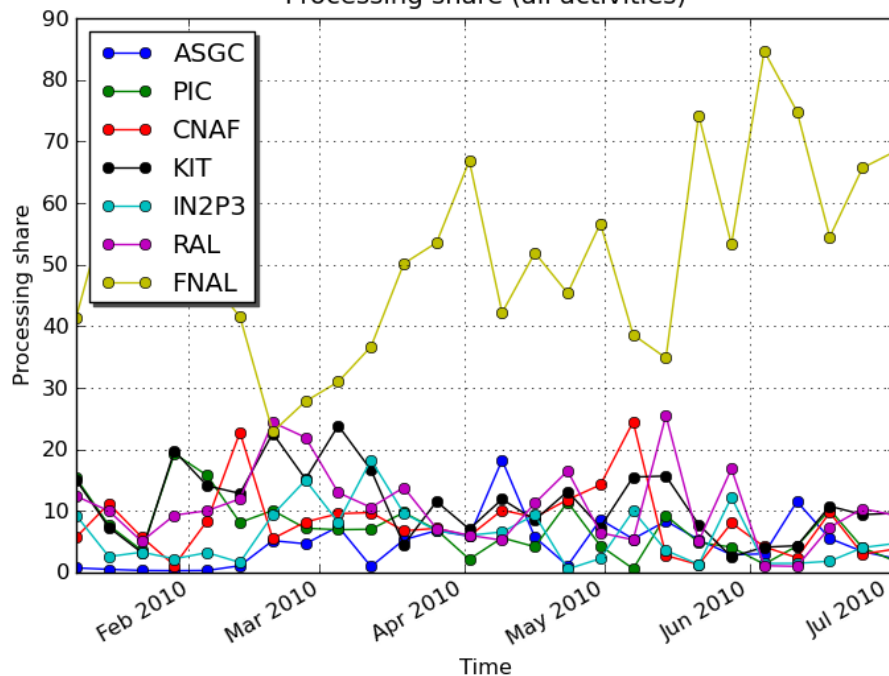
T1 processing share

26 Weeks from Week 01 of 2010 to Week 27 of 2010



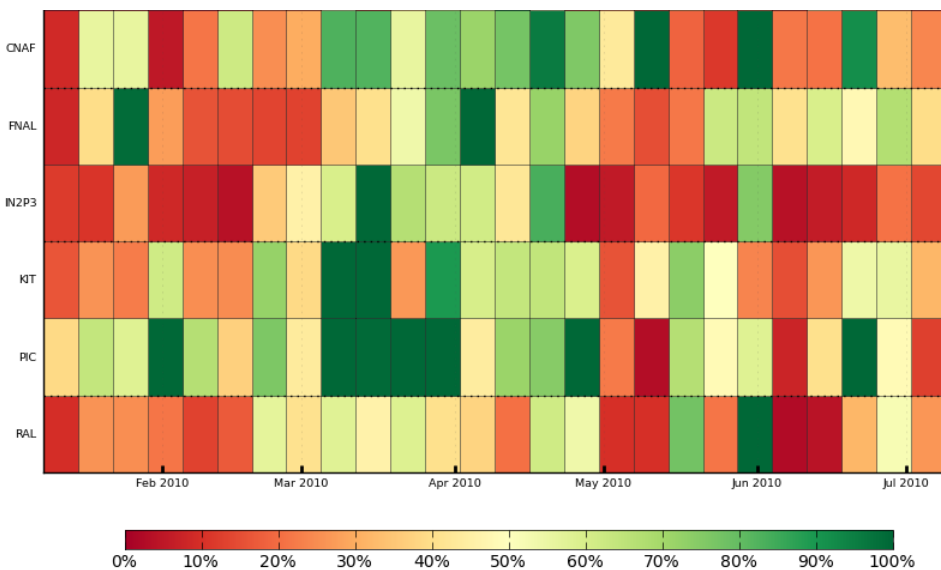
- Fraction of the processing done at each Tier-1
- Does not tell the level of utilization but how the processing is shared
- Useful to balance resource usage

Processing share (all activities)

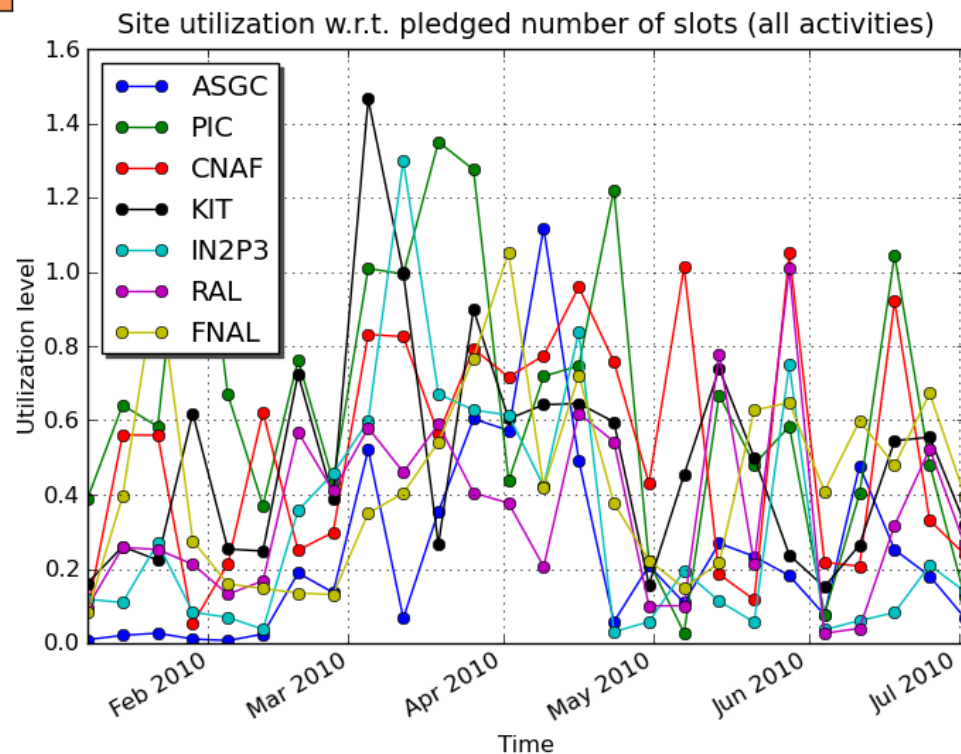




Utilization level



- Fraction of the pledged number of slots used
- Pledges taken from SiteDB and updated numbers from Chris
- NB: site utilizations > 100% plotted as 100% in map above

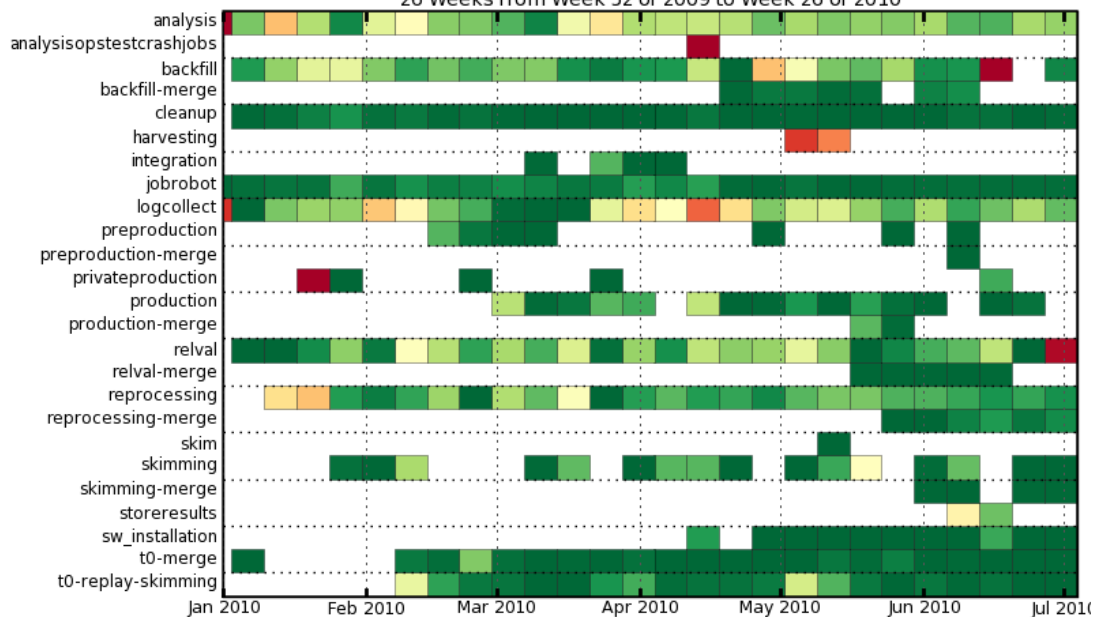




Job success rates

Efficiency based on success/failures

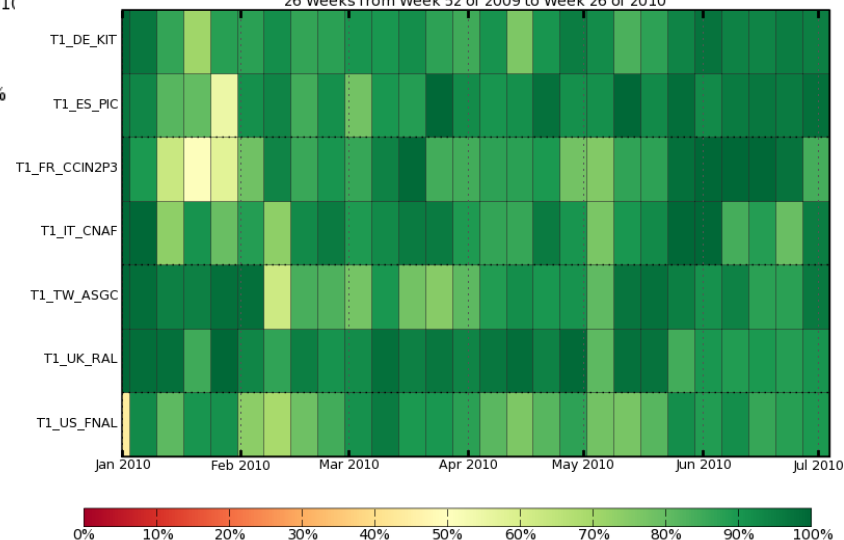
26 Weeks from Week 52 of 2009 to Week 26 of 2010



- In average ~90%
- Looks good!

Efficiency based on success/failures

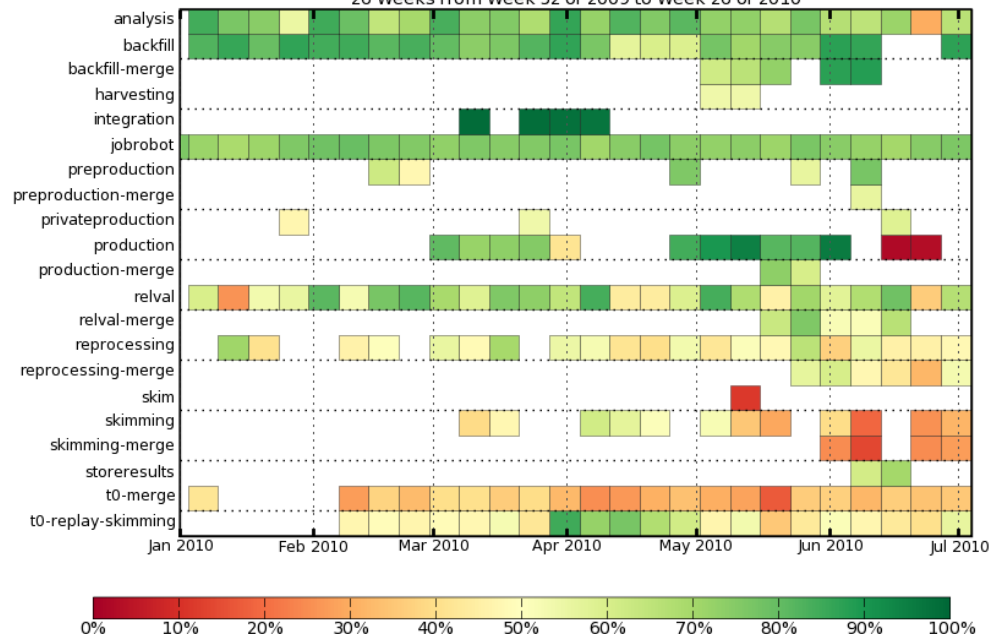
26 Weeks from Week 52 of 2009 to Week 26 of 2010



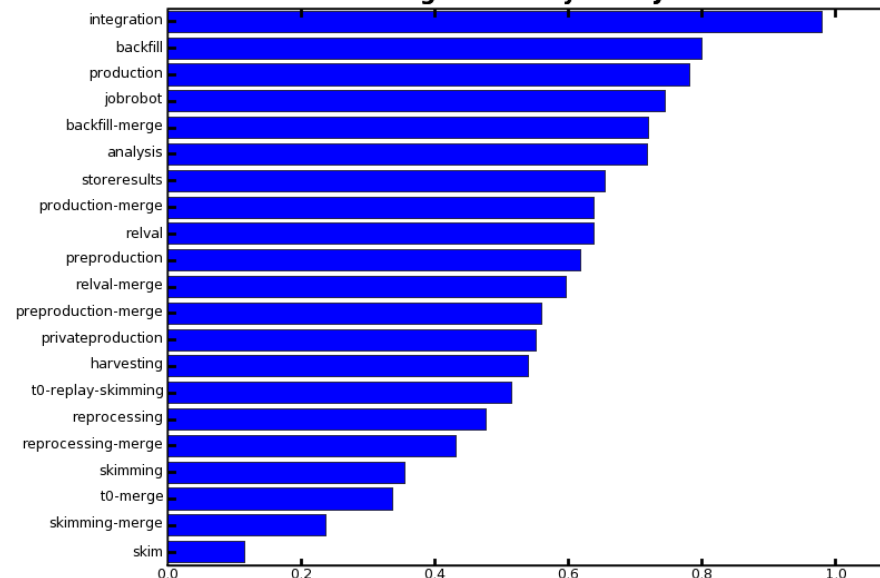
Job CPU efficiency (per activity)

Efficiency Good Jobs

26 Weeks from Week 52 of 2009 to Week 26 of 2010

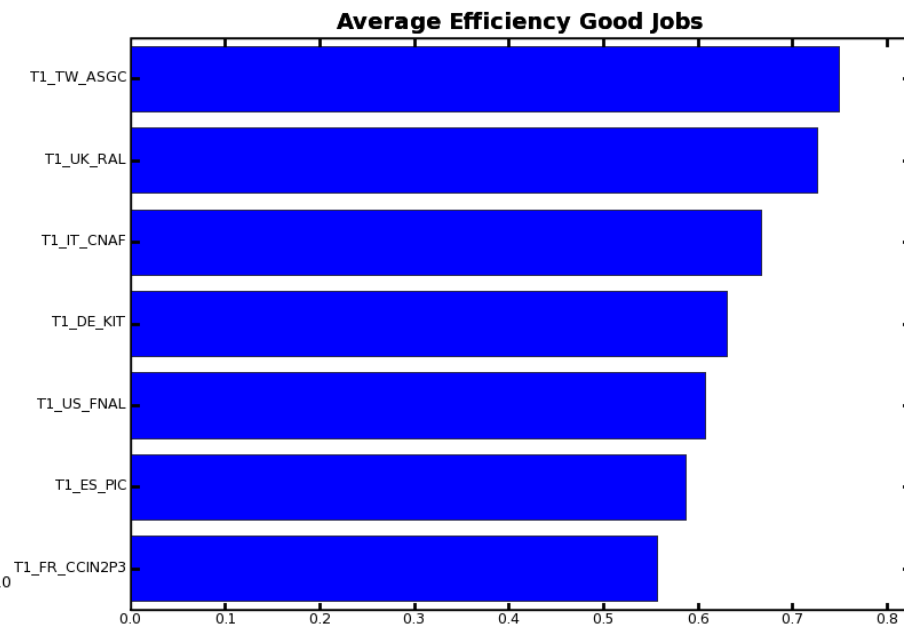
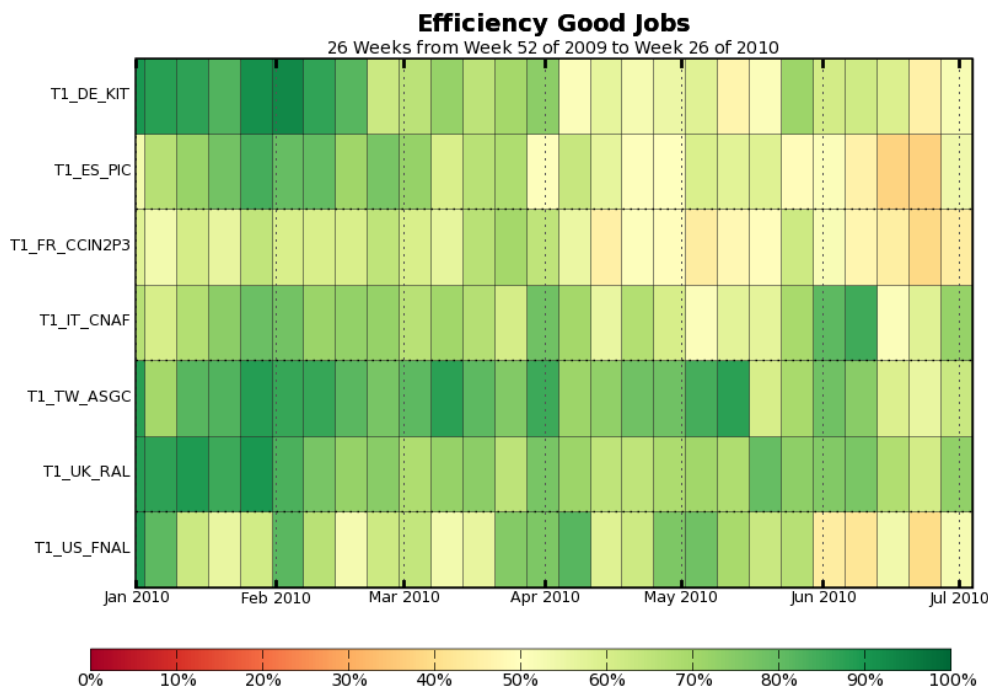


Average Efficiency Good Jobs



- `Cpu_time/wallclock_time` including stagein and stageout
- Low CPU efficiency for reprocessing, (prompt-)skimming, merge
- I/O bound jobs, overload of data serving infrastructure, stagein, application inefficiency reading data (file layout, read-ahead), etc

Job CPU efficiency (per T1)

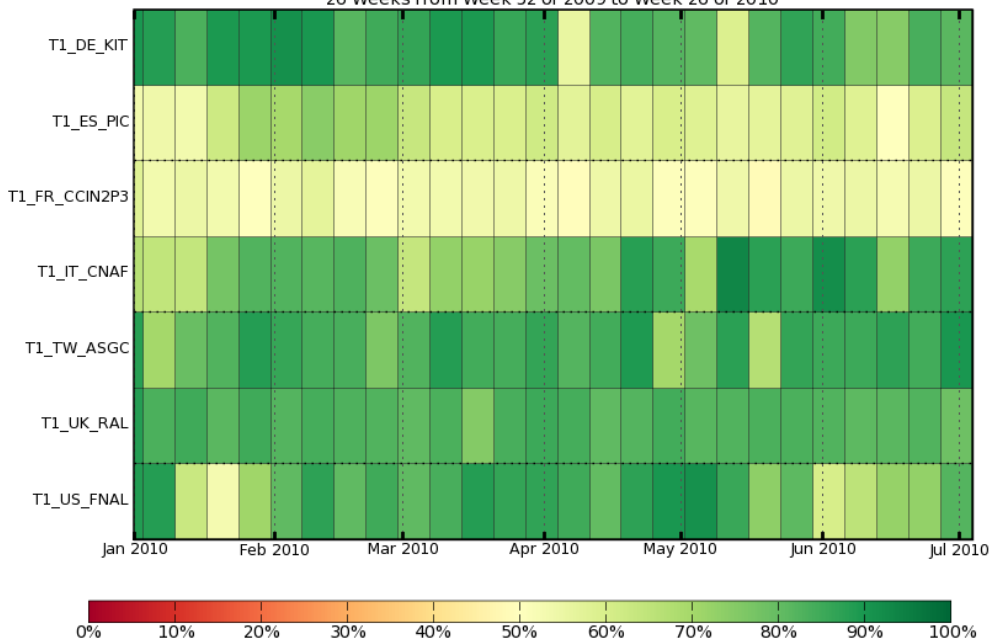


- CPU efficiency has worsened in the last month. Load? Workflow type?
- Better CPU efficiency for Castor/storm sites? LazyDownload

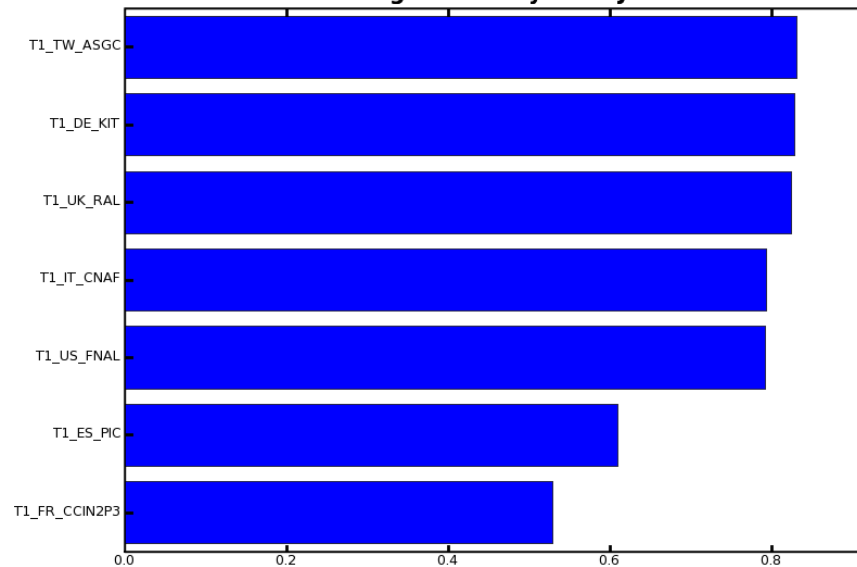
Job CPU efficiency (JobRobot)

Efficiency Good Jobs

26 Weeks from Week 52 of 2009 to Week 26 of 2010



Average Efficiency Good Jobs



- CPU efficiency consistently lower for PIC and IN2P3 for standard JobRobot jobs
- Observed at PIC after migrating WNs to SL5 (!?)



Summary

- Closely monitor resource utilization and performance
- Investigate inefficiencies and disentangle the various effects
- Try to balance resource utilization to make the most efficient use of the available resources
 - Important once we become resource-constrained
- We'll be reviewing resource utilization and performance every week at the Computing Operations meeting