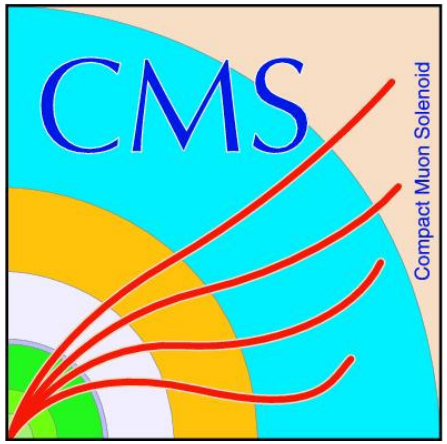
Efficiency Improvement at tier-2 IFCA

Facilities&Operations Meeting 21/09/2009



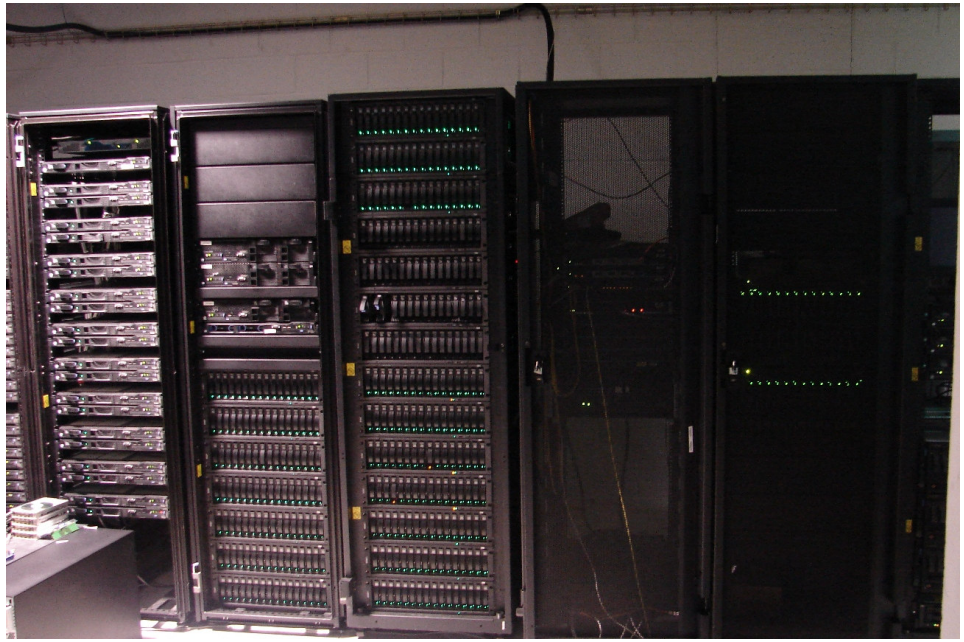
A.Y. Rodríguez Marrero
Instituto de Física de Cantabria (IFCA)
Spain

I.Cabrillo Bartolomé
Instituto de Física de Cantabria (IFCA)
Spain

IFCA

- Multi-VO: HEP, Astrophysics, Cosmology, Statistical Physics, …
  - Half of the Spanish T2. Uses GPFS to store the data + StoRM for the SRM layer.
  - It is working in different Computing projects.
- Noticed the behavior of cpu efficiency was lower (**35%**) than expected for analysis jobs, running at IFCA.
- The goal of this study was to detect and solve the possible problems to optimize the resources at IFCA

Instituto de Física de Cantabria



Efficiency Comparison Between IFCA and another T2 I

- The Job used is a typical cms skimming job:
  - running over an initial Dataset of 110 GB (split in 16 jobs, ~90000 events/each)
  - and for different number of events (from 100 to 100000)

- Two dedicated WN used: Usually all slots filled
  - IBM blade HS21 (cms28)
    - Intel(R) Xeon(R) CPU E5345 @ 2.33GHz
    - cache size : 4096 KB
    - 2 processors /4 cores per processor
  - IBM blade HS21 (gcsic012)
    - Intel(R) Xeon(R) CPU E5420 @ 2.50GHz
    - cache size : 6144 KB
    - 2 processors /4 cores per processor

Efficiency Comparison Between IFCA and another T2 II

- Running the same job (several times) at Tier-2 IFCA and another Tier-2 site (known to have a good efficiency reported at the CMS dashboard) and similar computing hardware, before any optimization at Tier-2 IFCA. Data is collected from CRAB output.

Efficiency = CPU time / EXE time
(Crab CPU Percentage)

Exec 100 Events	IFCA	T2
Exec Time (s)	51	49
Crab User CPU Time (s)	22.47	20.44
Crab Sys CPU Time (s)	0.65	1.27
Crab Wrapper Time (s)	97	92
Crab Stageout Time (s)	16	24
Crab CPU percentage	***45%***	***44%***

Exec 1000 Events	IFCA	T2
Exec Time (s)	69	70
Crab User CPU Time (s)	38.44	35.68
Crab Sys CPU Time (s)	1.01	1.37
Crab Wrapper Time (s)	103	110
Crab Stageout Time (s)	19	23
Crab CPU percentage	***56%***	***51%***

Similar or even better results for IFCA and few events

Efficiency Comparison Between IFCA and another T2 III

Exec 10000 Events	IFCA	T2
Exec Time (s)	476	238
Crab User CPU Time (s)	204.6	192.4
Crab Sys CPU Time (s)	4.56	5.02
Crab Wrapper Time (s)	500	268
Crab Stageout Time (s)	14	24
Crab CPU percentage	*43%*	*81%*

Exec 50000 Events	IFCA	T2
Exec Time (s)	3011	1618
Crab User CPU Time (s)	990.5	943.1
Crab Sys CPU Time (s)	22.37	56.96
Crab Wrapper Time (s)	3073	1976
Crab Stageout Time (s)	35	350
Crab CPU percentage	*33%*	*58%*

Exec 100000 Events	IFCA	T2
Exec Time (s)	7296	3880
Crab User CPU Time (s)	2015	2087
Crab Sys CPU Time (s)	47.80	126.6
Crab Wrapper Time (s)	7425	4563
Crab Stageout Time (s)	78	670
Crab CPU percentage	*28%*	*54%*

For longer jobs (>1000 events):

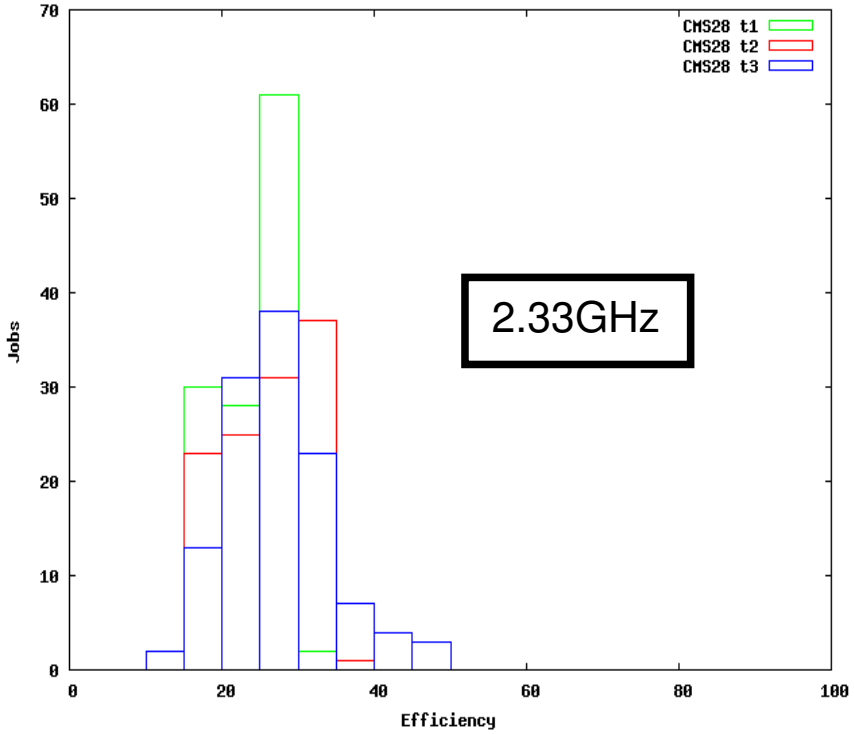
- **CPU Times** are very similar
- **Exec Times** are larger by a factor ~ 2
- Initial diagnostic: Most probably the **problem is at I/O**
  - File System (GPFS)
  - Network
  - Storage Hardware

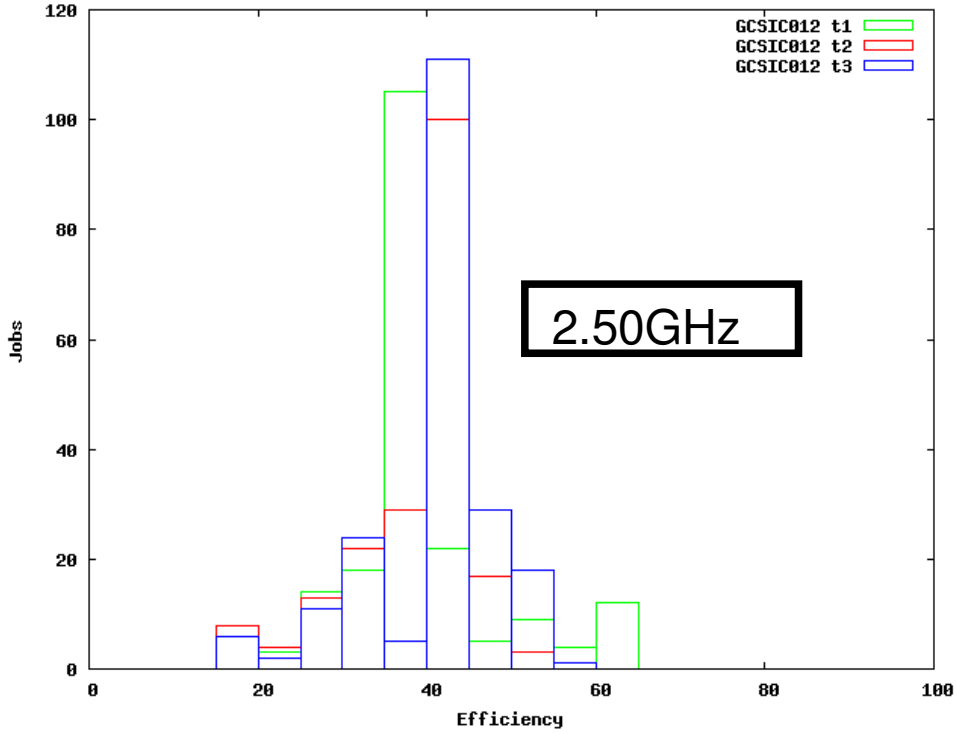
Looking Into GPFS I

- GPFS has a few parameters that may improve the file system I/O
  - **The Pagepool** determines the size of the GPFS file data cache.
  - **PrefetchThreads** indicate the number of threads that GPFS daemon should use for read or write operations .
  - **Worker1Threads** indicate the maximum number of threads that can be used for controlling sequential write-behind.

Looking Into GPFS II

- Pagepool set to 512Mb (initial), 1Gb and 2Gb (300 Jobs for each study case)





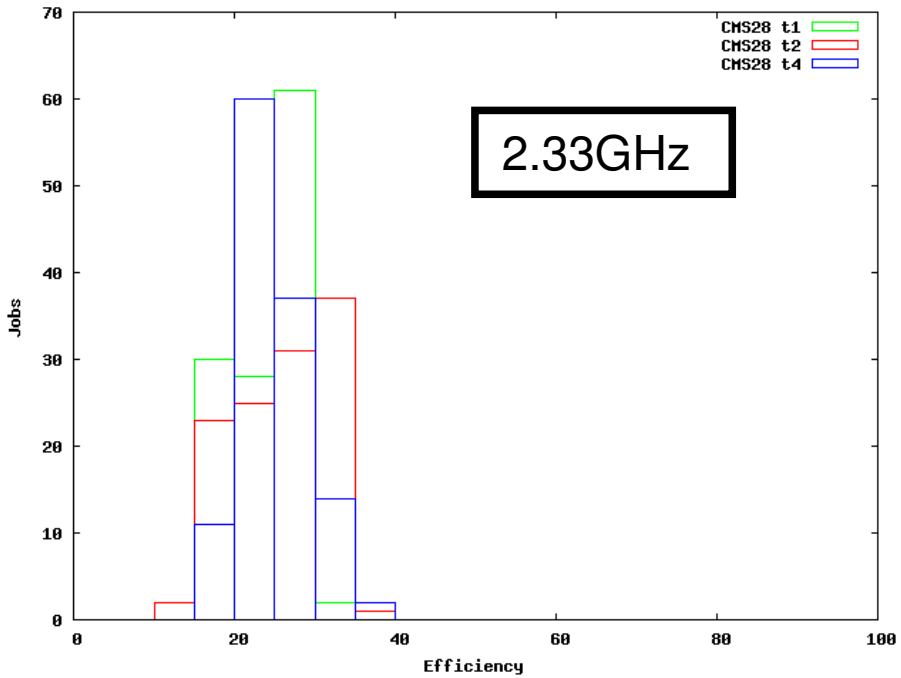
Best results for 1 Gb, improvement of relative 10%

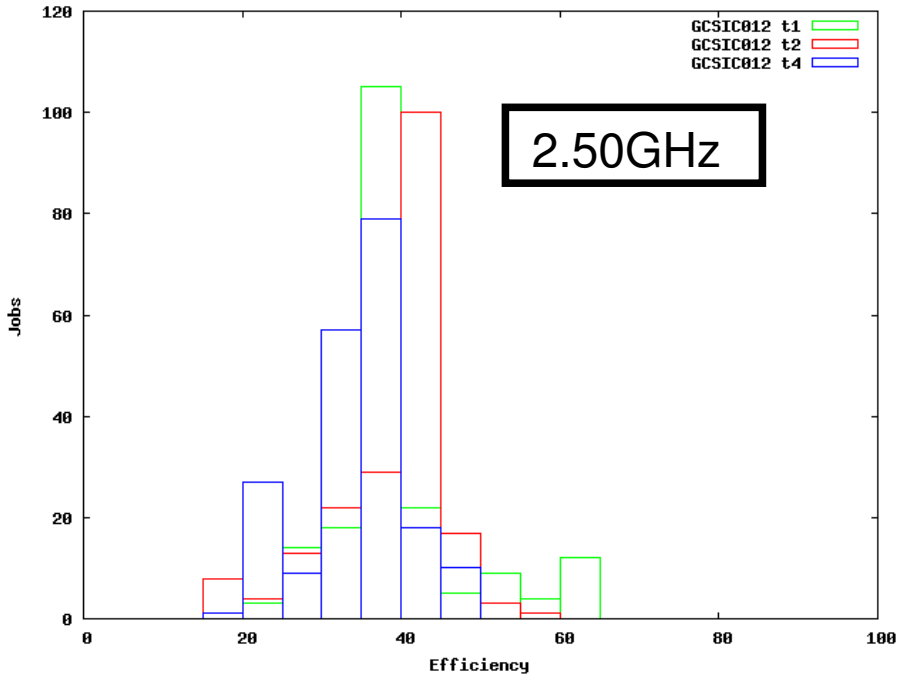
Mean (2.33 GHz) = 25.3

Mean (2.50 Ghz) = 38.1

Looking Into GPFS III

- PrefetchThreads from 50 to 100 (maintain 1Gb of pagepool)

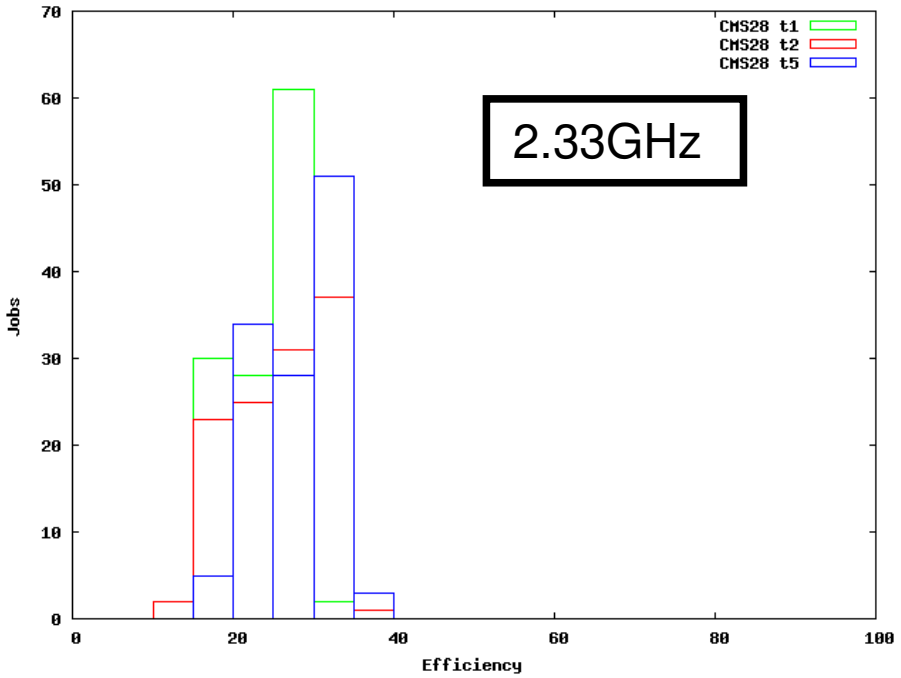


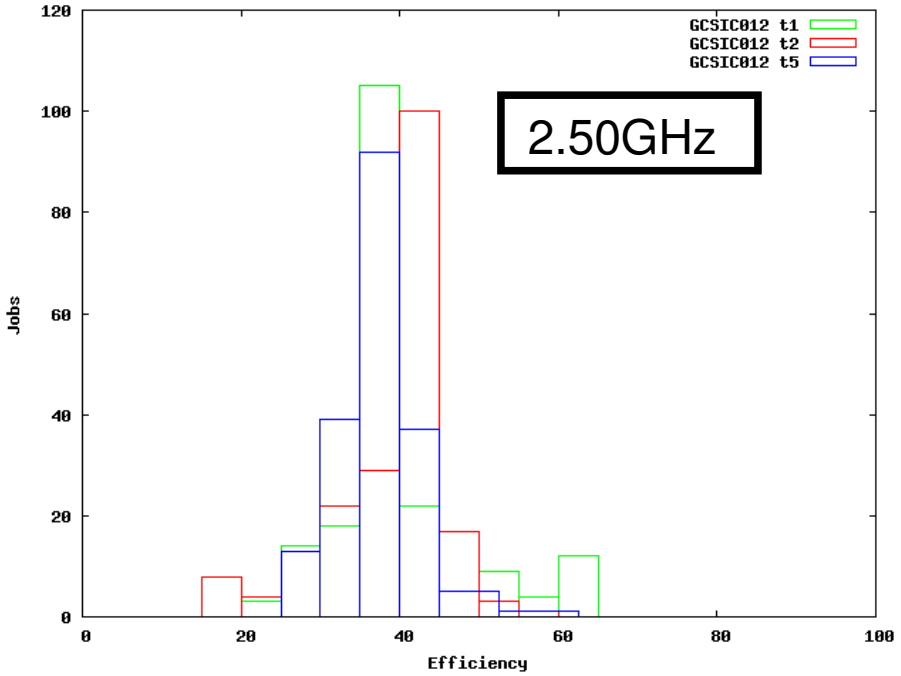


No improvement detected for this change : keep the initial values

Looking Into GPFS IV

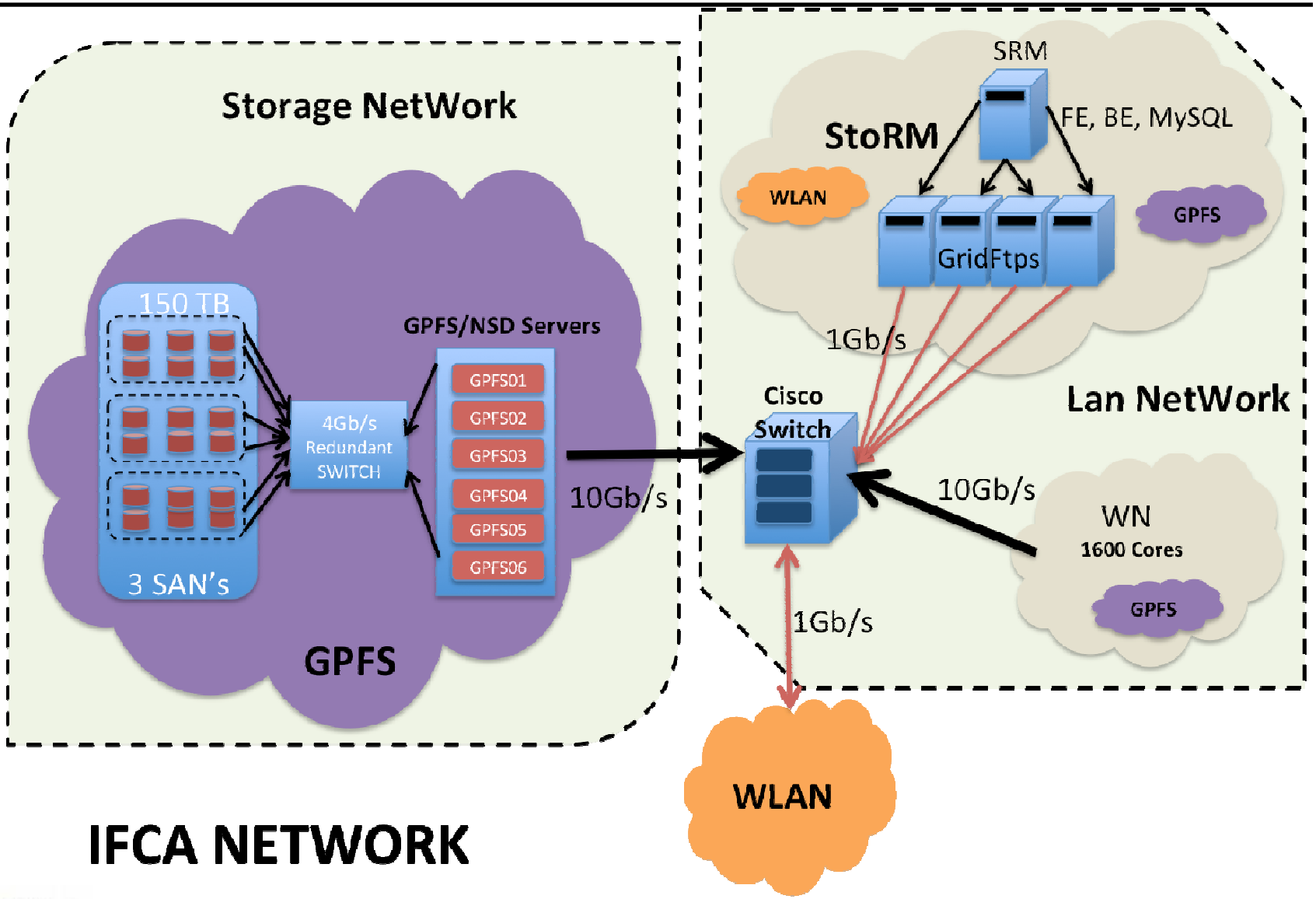
- Worker1Threads from 100 to 200 (maintain 1Gb of pagepool)





No general improvement detected for this change : keep the initial values

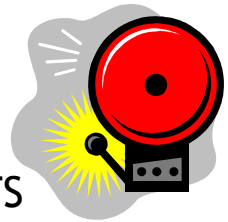
Looking into Network I



IFCA NETWORK

Looking into Network II

- Check Storage Network performance (Iperf):
  - GPFS Servers ←→ GPFS Servers 2 Gb/s Should be 4 -5 Gb/s
    - **Switch off Iptables (Private Network) : Improve to 4.5 Gb/s** (OK)
  - Gridftp pools ←→ GPFS Servers 1Gb/s (OK)
  - SE (StoRM) ←→ GPFS Servers 1Gb/s (OK)
  - WN's ←→ WN's 1 Gb/s (OK)
  - Gridftp pools ←→ GPFS Servers 1Gb/s (OK)
  - WN's → GPFS Servers 1Gb/s (OK)
  - **GPFS Servers → WN's: 250 - 500 Mb/s Should be 1 Gb/s!!!**
    - **net.ipv4.tcp_sack (tcp selective acknowledgements) = 0** (GPFS tuning recommendations), **turning it to 1 GPFS Servers → WN's 1Gb/S** (OK)
    - Gridftp's and StoRM have net.ipv4.tcp_sack = 0 and there is no problem, but if we connect this machines through WN switches GPFS Servers → Gridftp's 250 – 500 Mb/s
    - **Seems to be any malfunction between this parameter and WN's Switches** (under investigation)

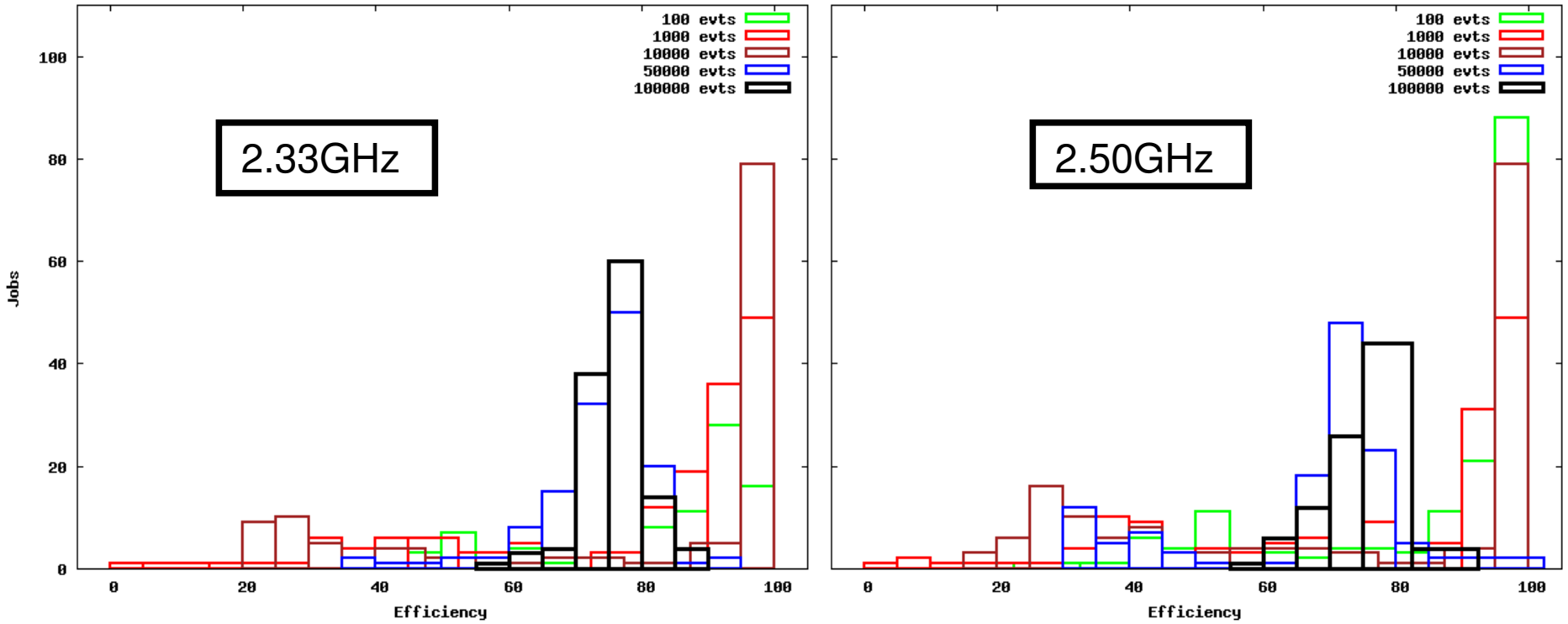


Looking into Storage Hardware I

- SAN's IBM
  - DS4700 Controllers and EXP810 expansion enclosures
    - Redundant FC 4 Gb/s connection
    - FC and SATA HDD support (SATA for IFCA case)
    - Support For 112 HDD slots
    - RAID5
  - Cache Parameters
    - **Read Caching Enable** (default Enable)
    - **Read-ahead multiplier** (prefetch) **Enable** (default Disable)
    - **Write caching Disable** (default Enabled)
  - Modification Priority **Low** (default High)
    - *modification priority* rates to determine their process priority.

Current Situation

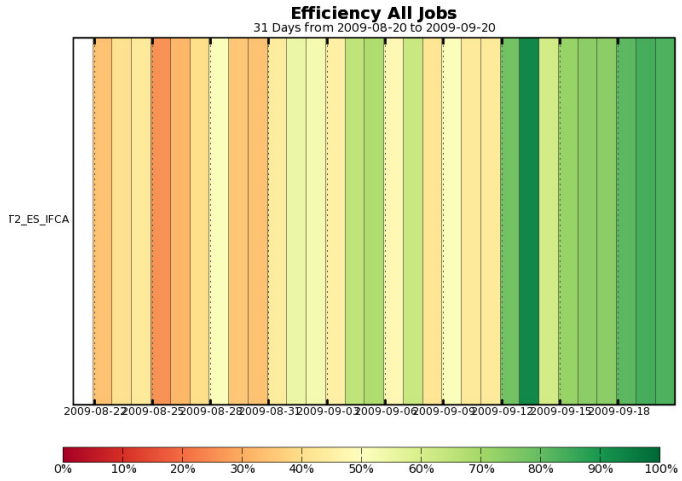
- After all these changes:
- Efficiency plots for ~120 (2.33GHz) & ~ 150 (2.55 GHz) jobs of each type



Mean (1.e5 events & 2.33 GHz) = 75.1

Mean (1.e5 events & 2.50 Ghz) = 73.2

Average CPU Efficiency



Pagepool 1Gb PrefetchThreads 50 Worker1Threads 100	2.33 GHz (~120 jobs)	2.5 GHz (~ 150 jobs)
100 evts	81.7	85.3
1000 evts	80.3	75.4
10000 evts	82.3	73.3
50000 evts	73.4	65.6
100000 evts	75.1	73.2

Need to run over same number of jobs to make a final comparison

- among CPU efficiencies for jobs runing over different number of events
- between CPU efficiencies for the two kind of WNs

Conclusions

- By changing a few parameters in several sites we manage to double the CPU efficiency for typical CMS analysis job.
  - **Pagepool = 1Gb**
  - **Net.ipv4.tcp_sack = 1**
  - **Read Caching Enable Read-ahead multiplier Enable**
  - **Write caching Disable**
  - **Modification Priority Low**
- Some of the changes are dependent on the storage technology, but some others might be usefull for non-GPFS sites: Network parameters.
- None of the modifications have affected the stability and/or reliability of the site.
- Improving the CPU efficiency has in fact result in a better handling of the whole T2 as the CEs can release their jobs sooner.

The End

¡Thank you very much!