HC Test Performance at AGLT2

B. Ball

March 31, 2010

Tests and Methodology

- Methodology
  - 24 hour Hammer Cloud testing
  - Attempt to sustain constant level of running jobs
  - Runs with 250, 500, 750, 1000, 1250 simultaneous jobs
- Run summary

HC Id	Simul Jobs	Date	Comment
1153	250	3/10 – 3/11	Kept no plots
1163	500	3/11 – 3/12	
1174	750	3/12 -- 3/13	
1188	750	3/14 – 3/15	Kept no plots
1189	1000	3/15 – 3/16	Kept no plots
1191	1250	3/16 – 3/17	Marked as 1000
1245	1000	3/29 – 3/30	
1248	1250	3/30 -- 3/31	

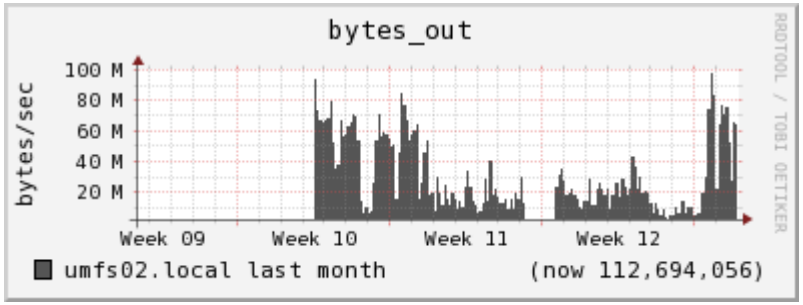
Metrics Examined

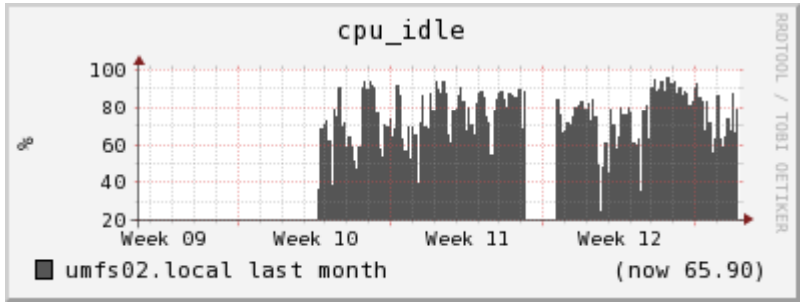
- Examine performance of NFS server hosting $HOME
  - Also hosts all ATLAS software kits
- dCache file server performance
- Worker node performance as a whole
- Single worker node under various loads
- Reported Ganga robot metrics
  - CPU Time/Wall Time
  - Events/WallClk
- Biggest problem – getting consistently high number of auto-pilots at high simultaneous job numbers to keep us filled

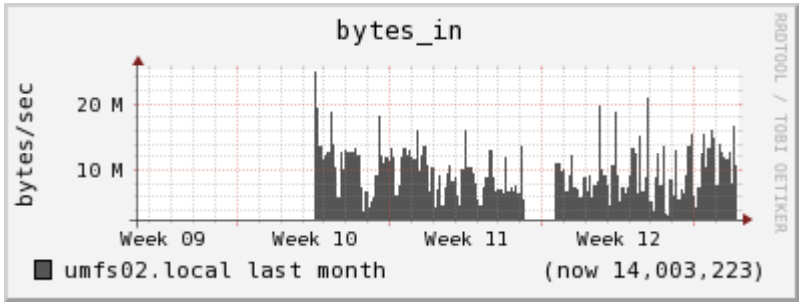
Performance of NFS Server

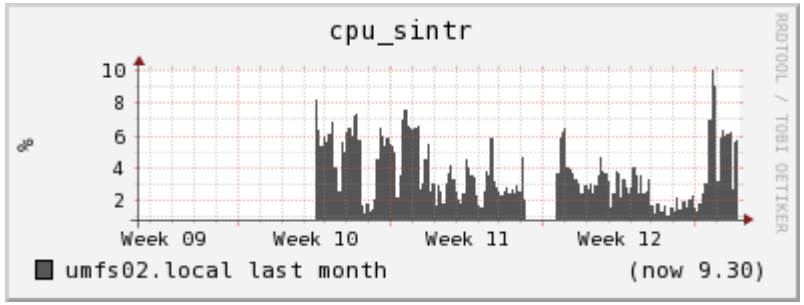
- umfs02.grid.umich.edu
  - Home for usatlasN accounts
  - Hosts all ATLAS software kits
- cpu_nice is minimal
- cpu_system is ~15% for any number of Analysis jobs
- cpu_wio is ~20% for any number of Analysis jobs
- bytes_out is ~40-60MB/sec, for any number of Anal jobs

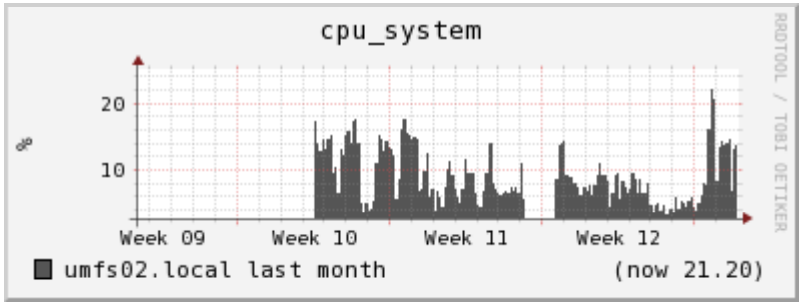
Umfs02 data

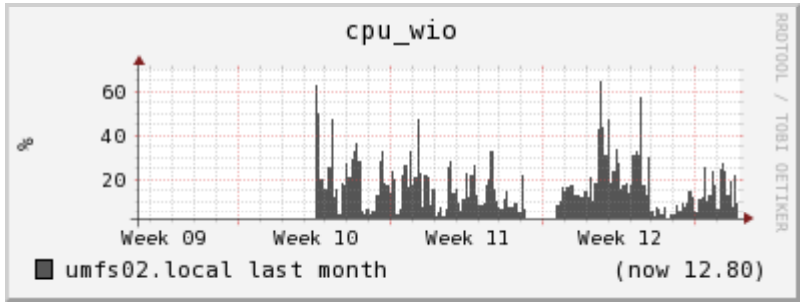

bytes_out
bytes/sec
100 M
80 M
60 M
40 M
20 M
Week 09
Week 10
Week 11
Week 12
umfs02.local last month
(now 112,694,056)
RRDTOOL / TOBI OETIKER


cpu_idle
%
100
80
60
40
20
Week 09
Week 10
Week 11
Week 12
umfs02.local last month
(now 65.90)
RRDTOOL / TOBI OETIKER


bytes_in
bytes/sec
20 M
10 M
Week 09
Week 10
Week 11
Week 12
umfs02.local last month
(now 14,003,223)
RRDTOOL / TOBI OETIKER


cpu_sintr
%
10
8
6
4
2
Week 09
Week 10
Week 11
Week 12
umfs02.local last month
(now 9.30)
RRDTOOL / TOBI OETIKER

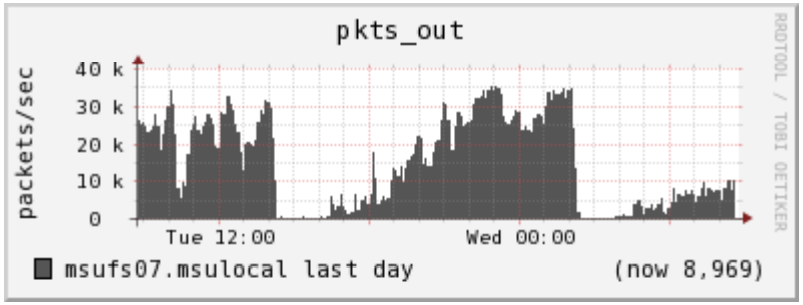

cpu_system
%
20
10
Week 09
Week 10
Week 11
Week 12
umfs02.local last month
(now 21.20)
RRDTOOL / TOBI OETIKER

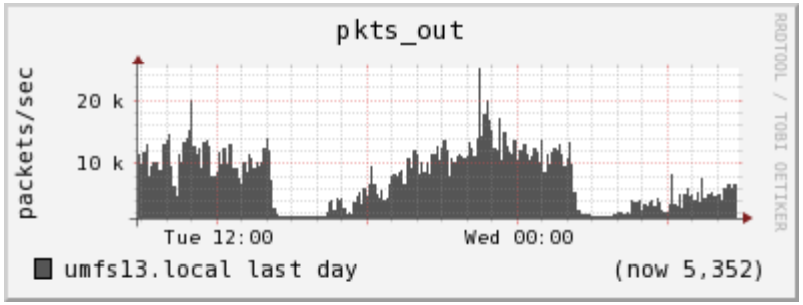

cpu_wio
%
60
40
20
Week 09
Week 10
Week 11
Week 12
umfs02.local last month
(now 12.80)
RRDTOOL / TOBI OETIKER

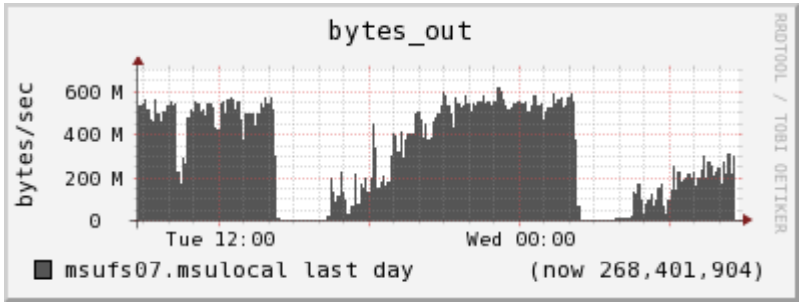
dCache Performance

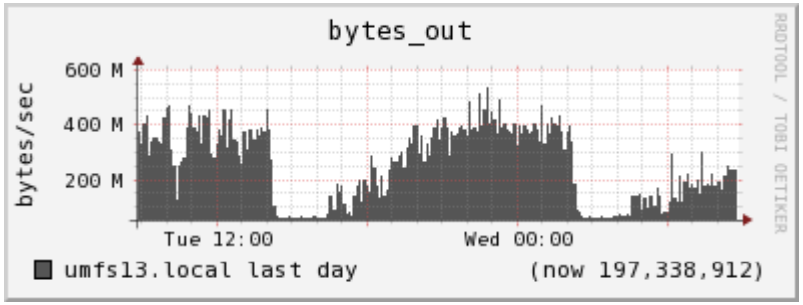
- Determined that umfs13 and msufs07 are carrying balance of file service load
- Network output from each is nearly linear up to 750 simultaneous jobs. For 1100 jobs we see….
  - msufs07 – 30k packets/sec, 500MB/sec
  - umfs13 – 11k packets/sec, 375MB/sec
- Cpu very busy. With 1100 running
  - 25% system
  - 50% wait on io

Network output at 1100 jobs

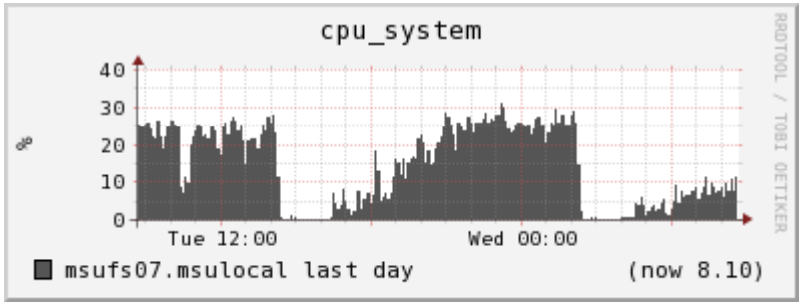


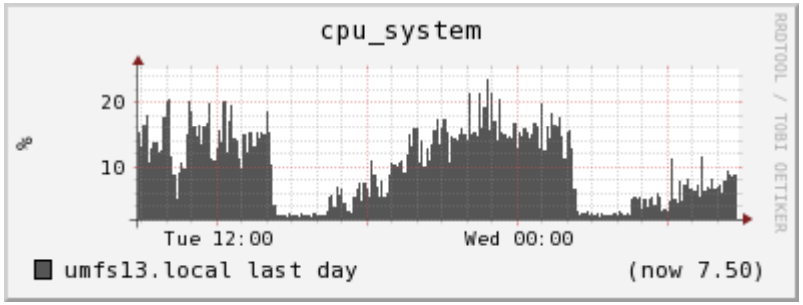


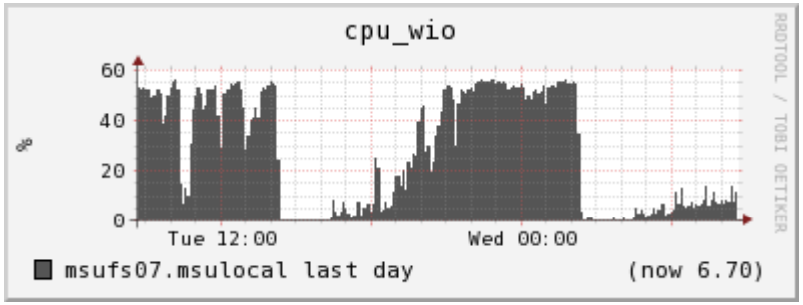


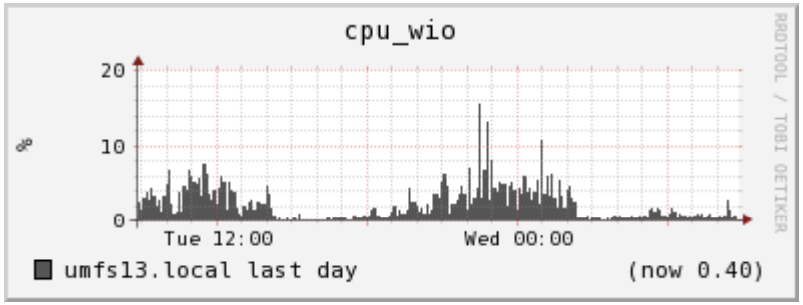


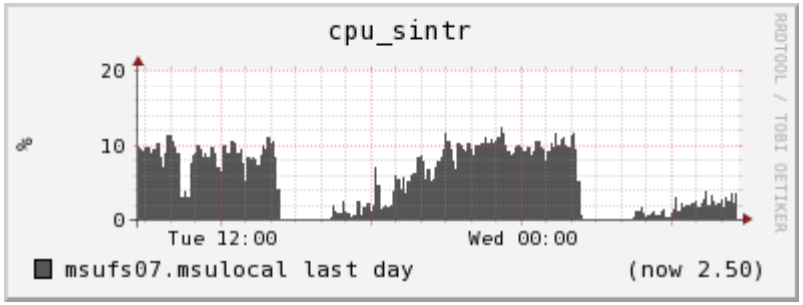
Cpu load on msufs07 with 1100 jobs

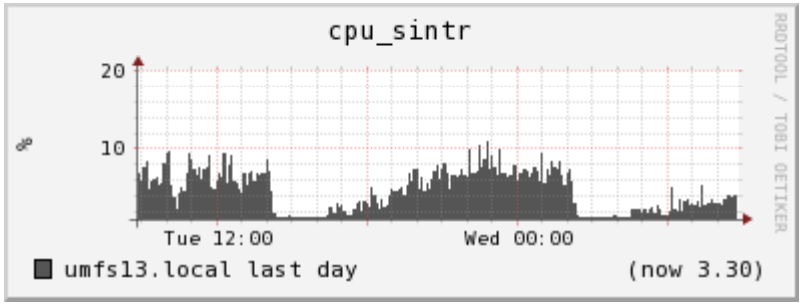








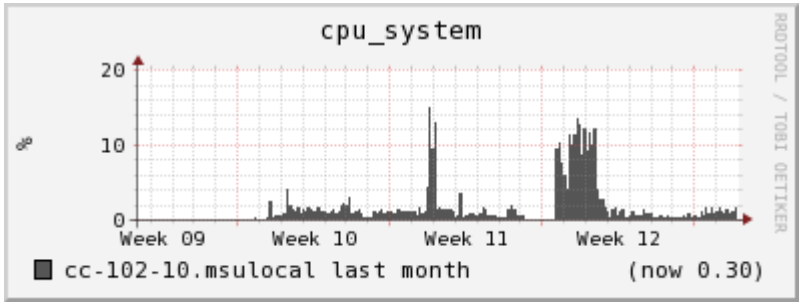


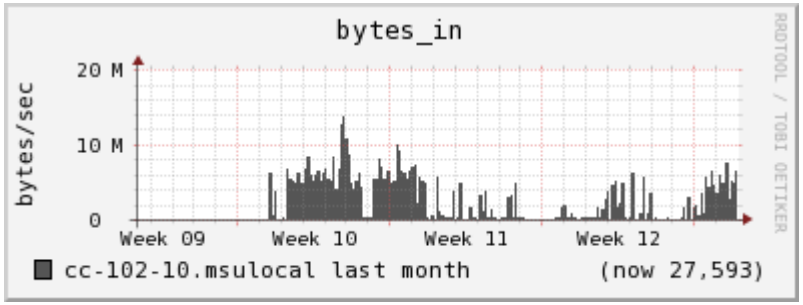


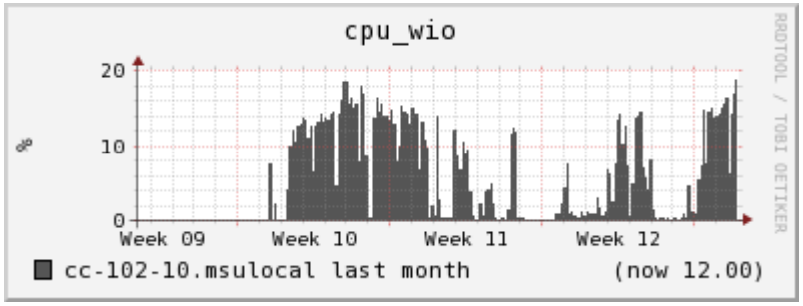
Look at Worker Nodes

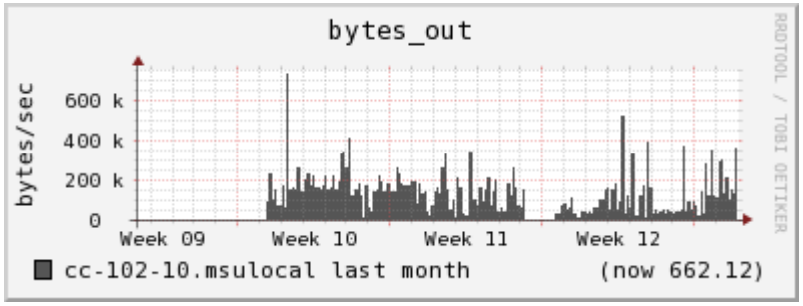
- c-102-10 is guaranteed 4 Analysis slots out of 8 available
  - Balance of cores running production or other jobs
- Reaches 4k packets/sec (6MB/sec) incoming network
  - Independent of total number of Analysis jobs on the system
- Cpu_wio 12-15%
- System cpu ~1%
- Cpu_nice ~55%
- Running process count is ~5-6 out of 8 cores

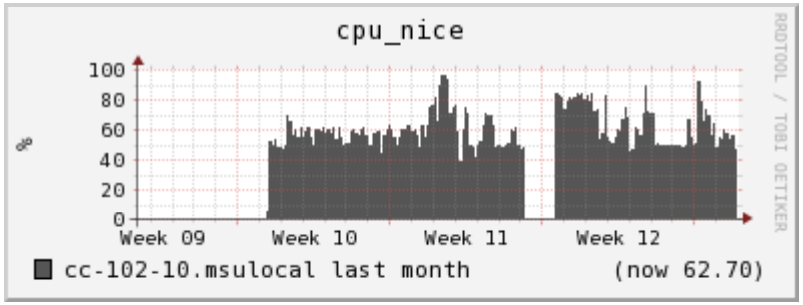
c-102-10 performance last month

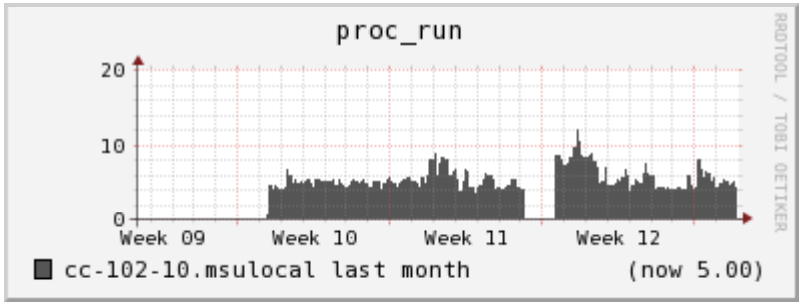








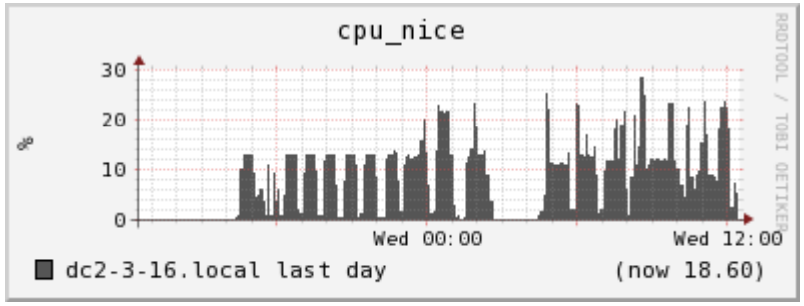


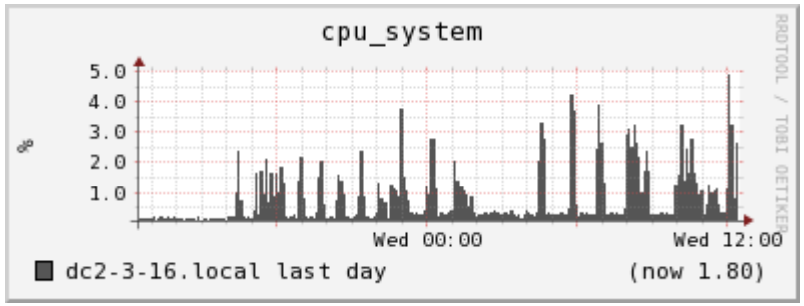


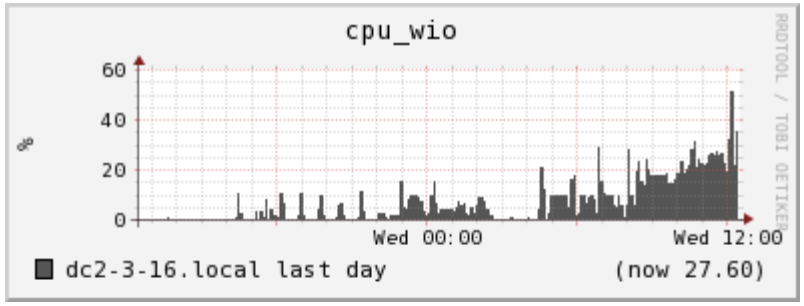
Analysis only on c-3-16

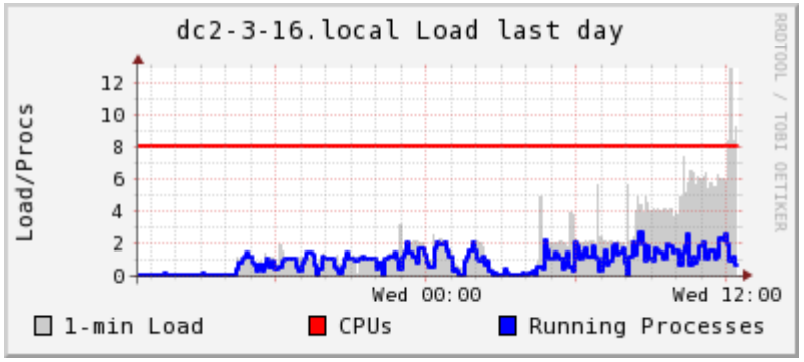
- Select an 8 core machine and run ONLY Analysis jobs
  - Using 1, 2, 4, 6, 8 simultaneous with no other interfering usage
- Still in progress, 1, 2, 4, 6 complete, 8 now running
- As count increases, time spent in wio increases
- Load_one/running_processes grows quickly
  - Two jobs gives load_one about 1.3 times Running Process count
  - Four jobs gives a ratio of 2.6

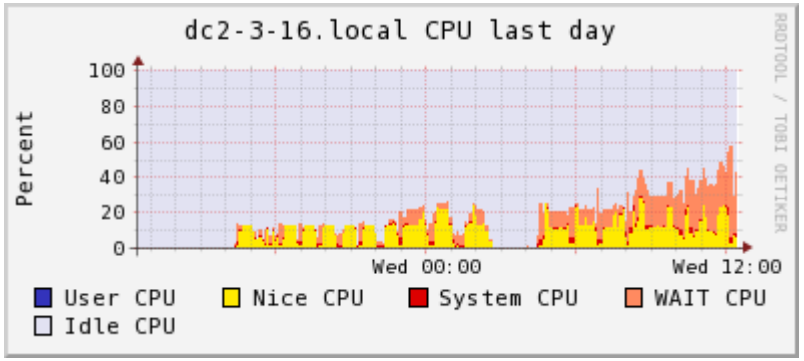
Preliminary c-3-16 metrics











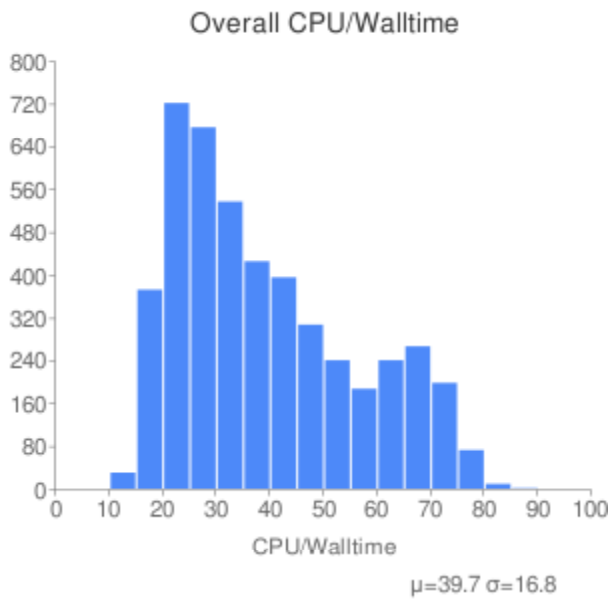
c-3-16 Table of Results

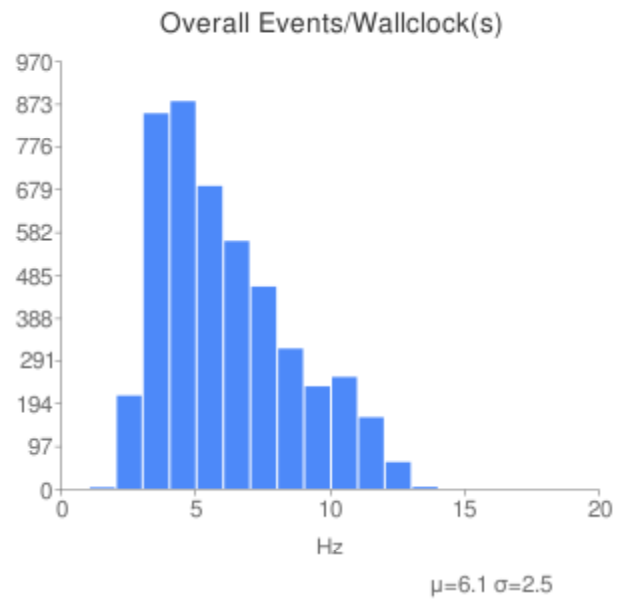
Slots	Cpu_idle	Cpu_nice	Cpu_sys	Cpu_wio	Load_one/run_procs
1	0.87	0.13	<0.01	0.01	1.0
2	0.77	0.11	0.01	0.06	1.3
4	0.70	0.11	0.015	0.18	2.6
6	0.57	0.08	0.01	0.22	3.7
8	0.45	0.04	0.01	0.30	5.3

Ganga Robot metrics

- Two plots recorded at typical Ganga Robot page
  - Eg, http://gangarobot.cern.ch/hc/1248/test/
  - Overall CPU/Walltime
  - Overall Events/Wallclock
- Using data from these we see CPU/Walltime dropping as simultaneous events rises to ~750, then levelling out
- Overall Events/Wallclock exhibits similar behaviour
- Problem: at high simultaneous job counts, cannot sustain that count on an active T2
  - Need consistent auto-pilot submission, a problem of late
- Must either dedicate the T2, or “filter” the HC results, keeping only jobs that start and end during some window of time.

Typical Ganga Robot report





HC Metric Summary

- Eight HC test runs
- Data with more than 750 simultaneous jobs have discussed problems
- Run 1191 reported 1250, marked on web as 1000

HC ID	Simul Jobs	CPU/Wall	Evts/Wall
1153	250	54.2	8.2
1163	500	49.4	7.5
1174	750	44.0	6.7
1188	1000	43.3	6.6
1191	1250*	38.2	5.8
1245	1000	51.2	8.0
1248	1250	43.1	6.6

An Observation

- On slide 10 we see a higher cpu utilization during week 12
- These are real Analysis jobs, with 1250-1300 running at that time
- Cpu and dCache usage are different than the HC usage
  - More and better cpu utilization
  - Less dCache access
- One must ask, “Is the HC test really representative of Analysis usage that we will see?”

Conclusions

- NFS access to VO home directories and ATLAS kits is unchanged (saturated?) for all HC job counts
  - Likely need better service here
  - Split kits and VO home directories
  - Use a better server (current PE1950/MD1000), eg, SUNNAS system or solid state drives
- dCache performs well, but would be aided by better distribution over available server systems
- Limiting number of Analysis jobs per worker node is good
  - Best number for a PE1950 is TBD
  - Using half the cores is a good rule of thumb as a maximum
    - May want to decrease this even further