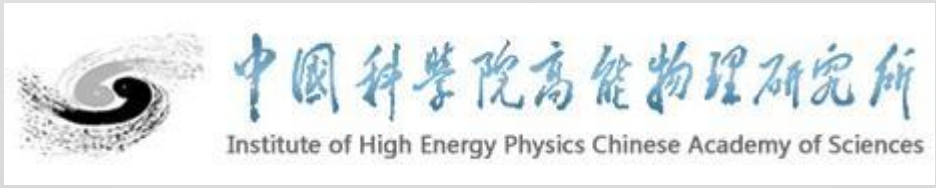
Use ATLAS@home to exploit extra CPU from busy Tier2 site

Wenjing Wu[1], David Cameron[2]

1. Computer Center, IHEP, China
2. University of Oslo, Norway

2017-9-21



Outline

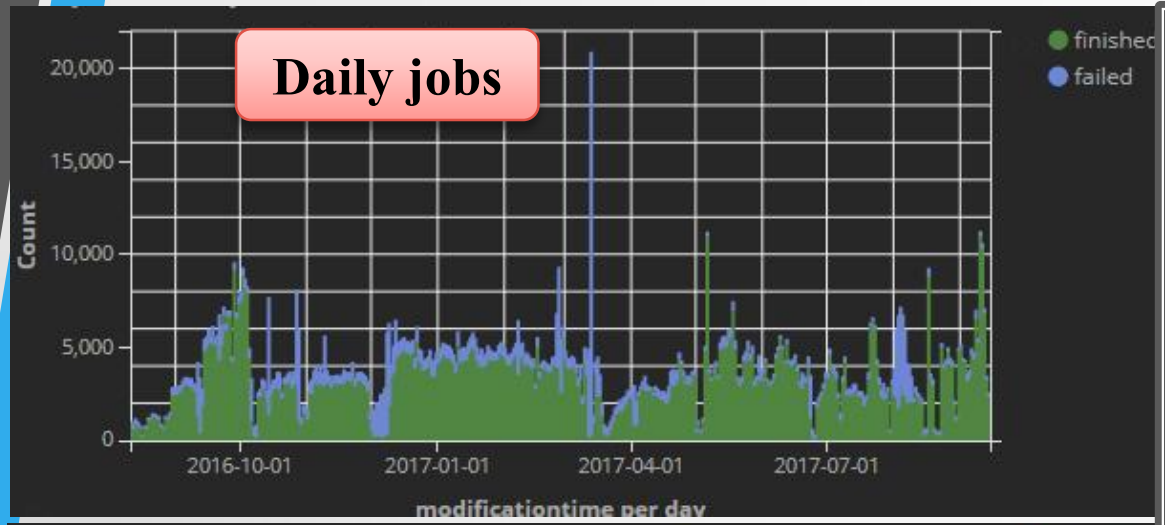
- ATLAS@home
  - Running status
  - New features/improvements
- Use ATLAS@home to exploit extra CPU at ATLAS Tier2
  - Why running on Tier2?
  - How much can be exploited?
  - Impact on Tier2?
  - Can it run on more Tier2 site?
- Summary

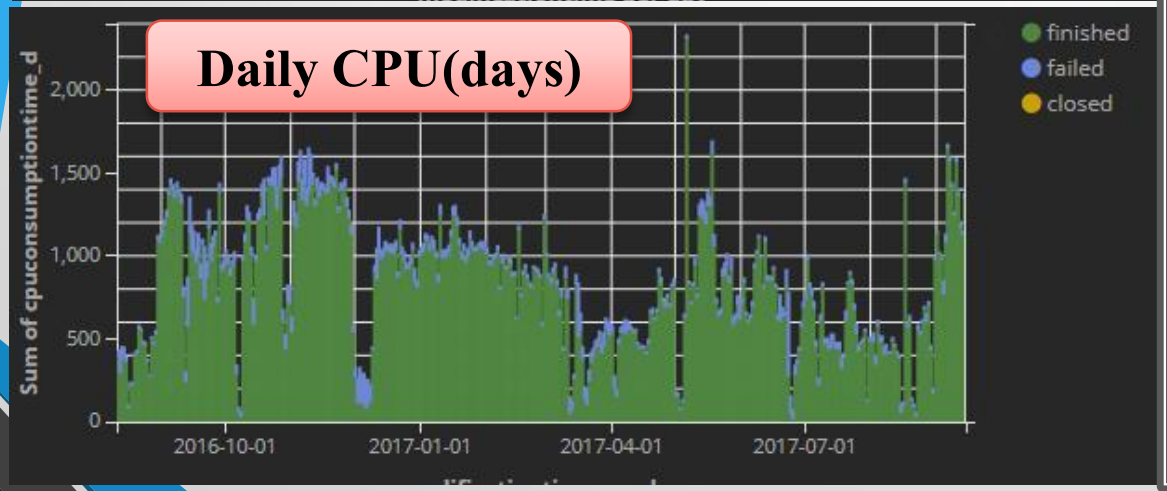
ATLAS@home running status

ATLAS@home

- Started to run in production from Dec 2013 as ATLAS@home, has been in steady running ever since
- Merged into LHC@home as “ATLAS” app in 2017.3
- Main feature:
  - Use virtualbox to virtualize heterogeneous volunteer hosts to run short and less urgent ATLAS simulation jobs (100 events/job)
- Recently added new features
  - Support multiple core jobs in virtualbox (2016.6)
  - Support native running (2017.7)
    - Run ATLAS directly on SLC6 hosts
    - Run through Singularity on other Linux hosts

Long term running status (1)

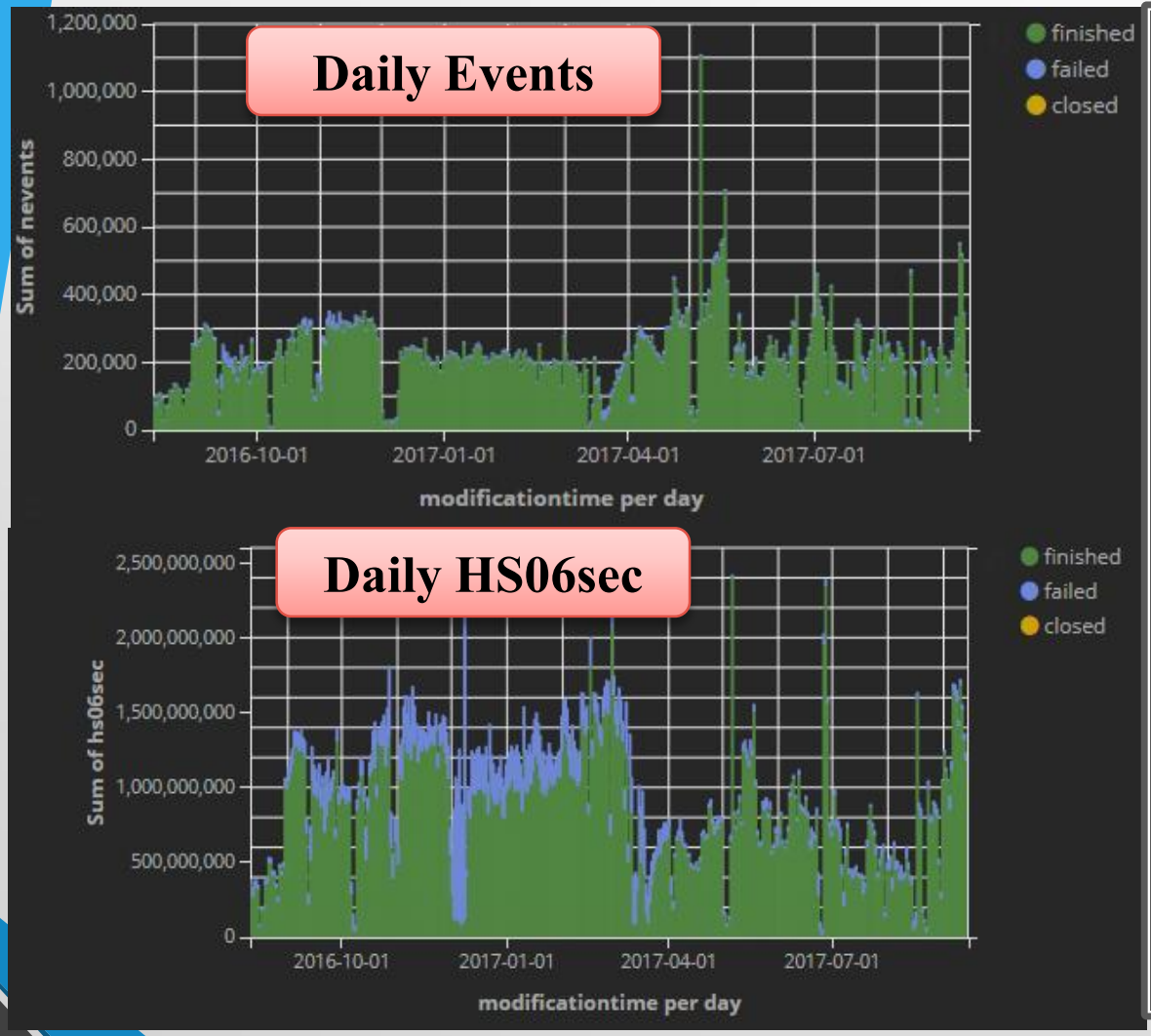




Avg. Per day from the past 400 days:

- **3300** jobs (finished)
- **852** CPU days
- **0.22** Mevents
- **1 B** HS06sec , equal to **1157** CPU days on a core with 10 HS06

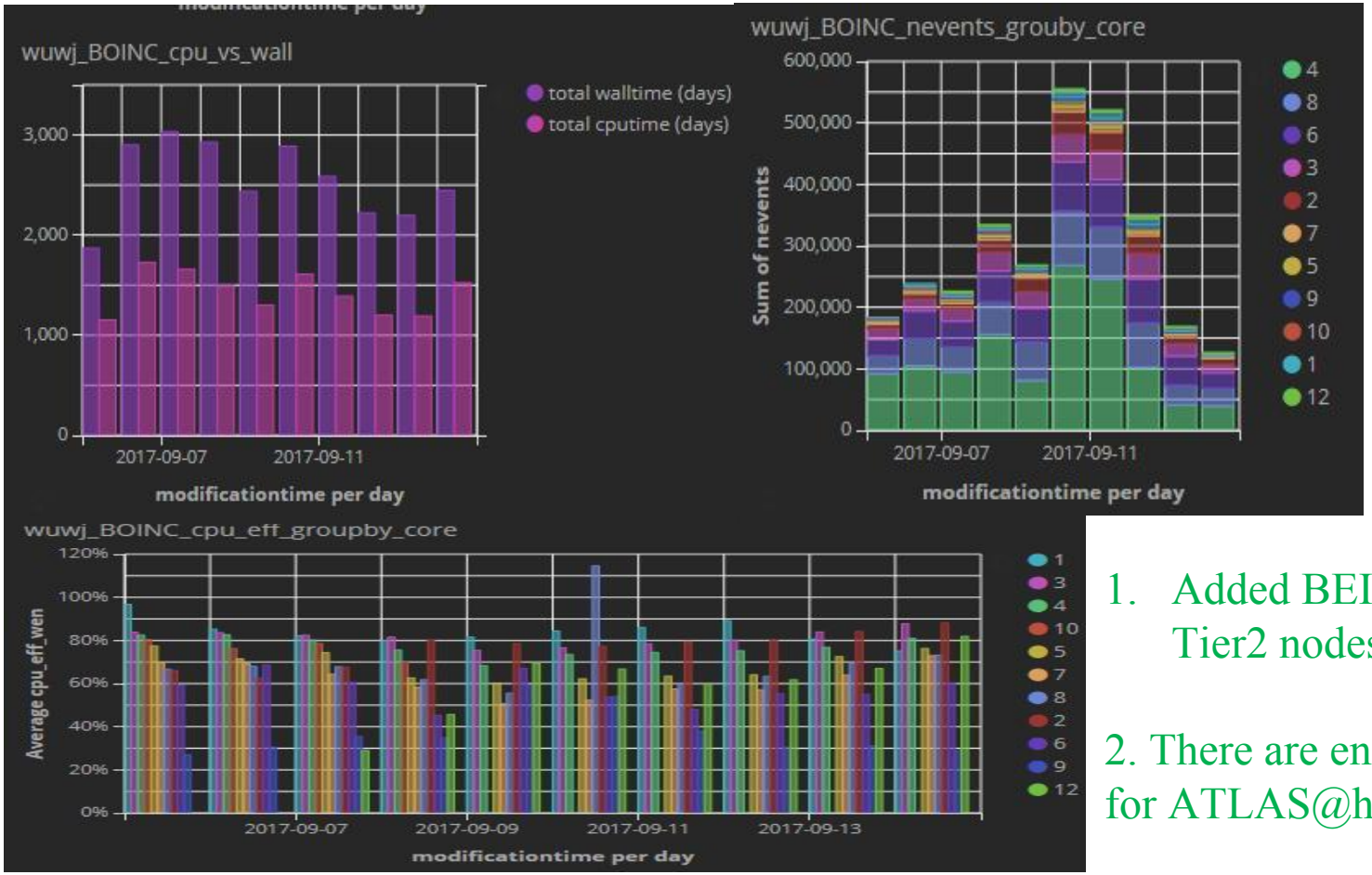
Long term running status (2)



The gaps are mostly due to lack of tasks for ATLAS@home

Due to “long tail” tasks, only less urgent tasks are assigned to ATLAS@home

Recent usage (recent 10 days)



1. Added BEIJING Tier2 nodes

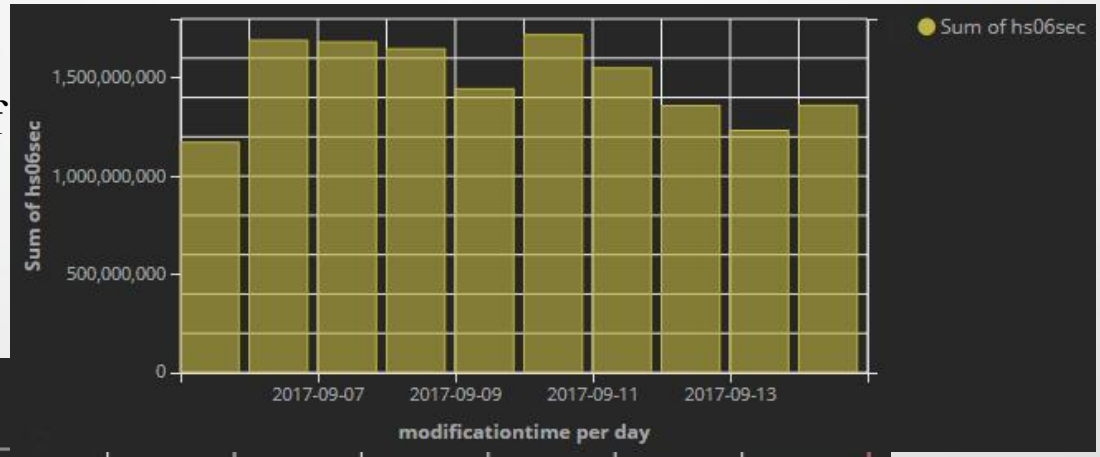
2. There are enough tasks for ATLAS@home

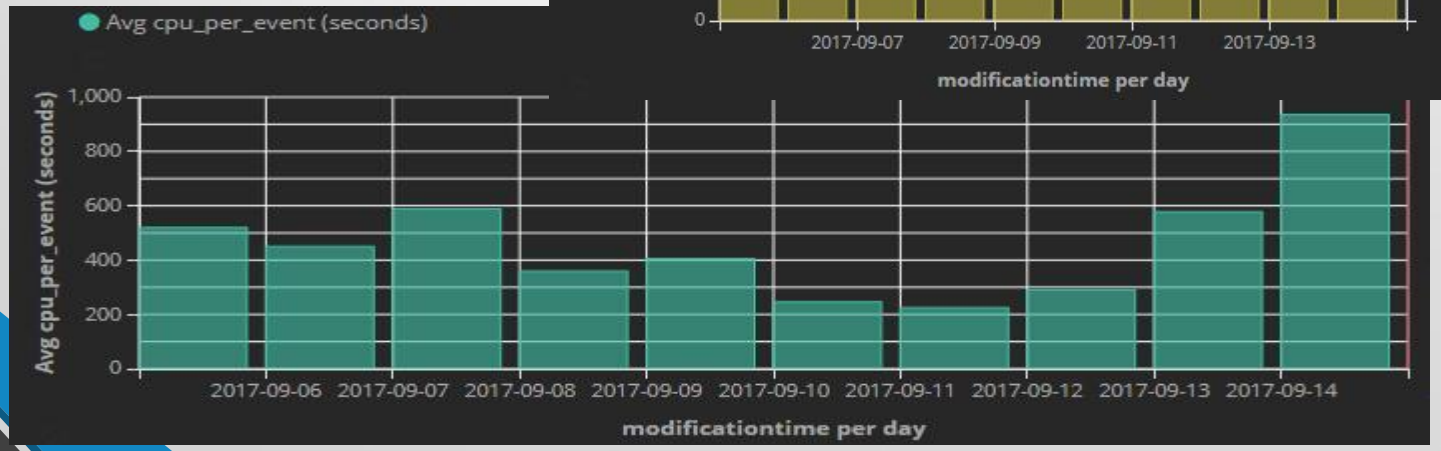
Daily Walltime: **2200** CPU days, Daily CPUtime: **1405** CPU days,

Daily events: **0.3** Mevents

Avg. CPU efficiency is around 75% (cpu_time/wall_time)

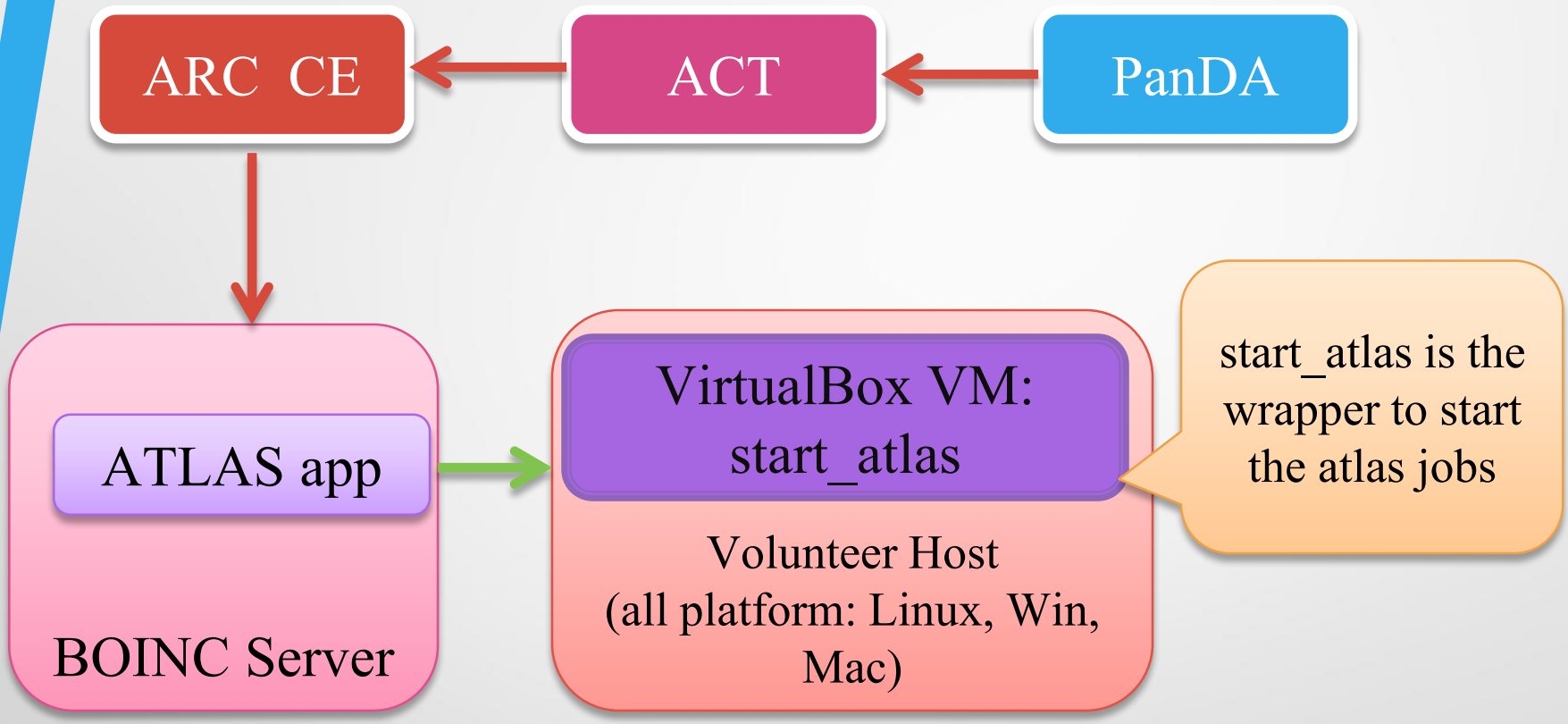
- CPU time does not consider the speed of CPU (HS06)
- CPU time perevent varies from task to task by a few times even on the same CPU.
- HS06sec is more comparable, avg. 1.6B HS06sec/day
  - 18517 HS06 days/day
  - Equal to 1851 Cores (10 HS06/core, which is the case for most grid site)
- Equivalent to a Grid site of **2700** CPU cores for simulation jobs! (avg. 0.70 CPU utilization)
- cpu_util= wall_util * job_eff





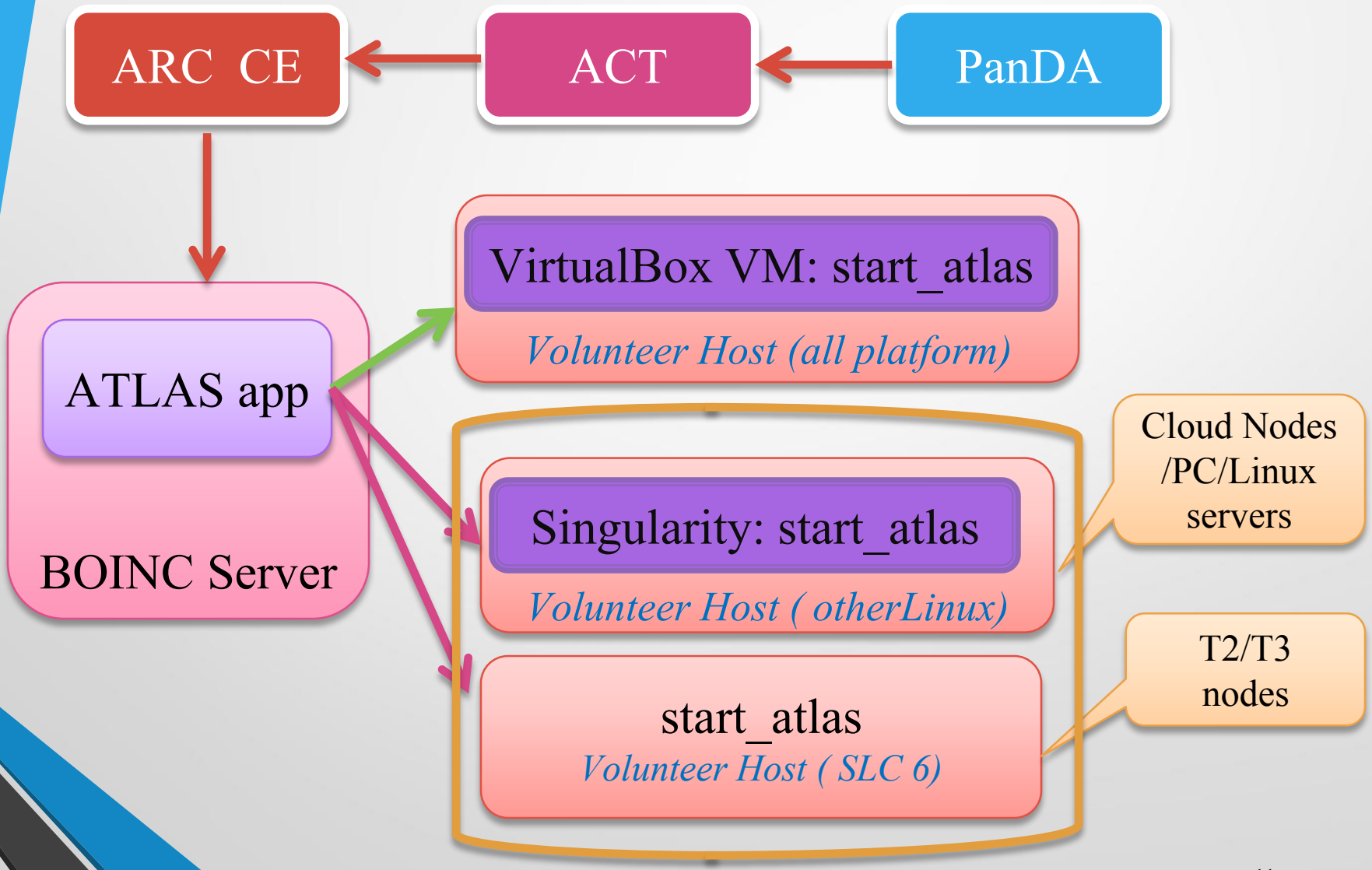
New Features for ATLAS@home

ATLAS@home with VirtualBox



Traditional way to run ATLAS@home

VirtualBox and Native



Native

- For Linux nodes with SLC6, atlas jobs are directly run without containers/Virtualbox
- For other Linux (Cent OS 7, Ubuntu), Singularity is used to run the atlas jobs
  - Some CERN cloud nodes use Cent OS 7
  - A lot of volunteer hosts use Ubuntu
- It improves CPU/IO/Network/RAM performance compared to using Virtualization
- Jobs can be configured to be kept in memory, so they do not lose the previous work during suspending.
- Requirements on the nodes
  - Have **CVMFS** installed
  - Have **singularity** installed and bind mount CVMFS (Cent OS 7/Ubuntu)

Why running on Tier 2 site

Why: Tier2 is not always fully loaded

- Site downtime
- Conservative scheduling around downtime (draining jobs/ ramp up jobs slowly)
- Not enough pilots at site (occasionally)
- What is the utilization for typical Tier2 site?
  - Choose a few sites with different scale/region
  - Calculate its walltime/(total_cores*24) in the past 100 days, break into every 10 days (to avoid spikes from jobs running over days)
  - Asked the sites for their number of dedicated cores for ATLAS
  - Avg. 85% wall utilization, 70% CPU utilization

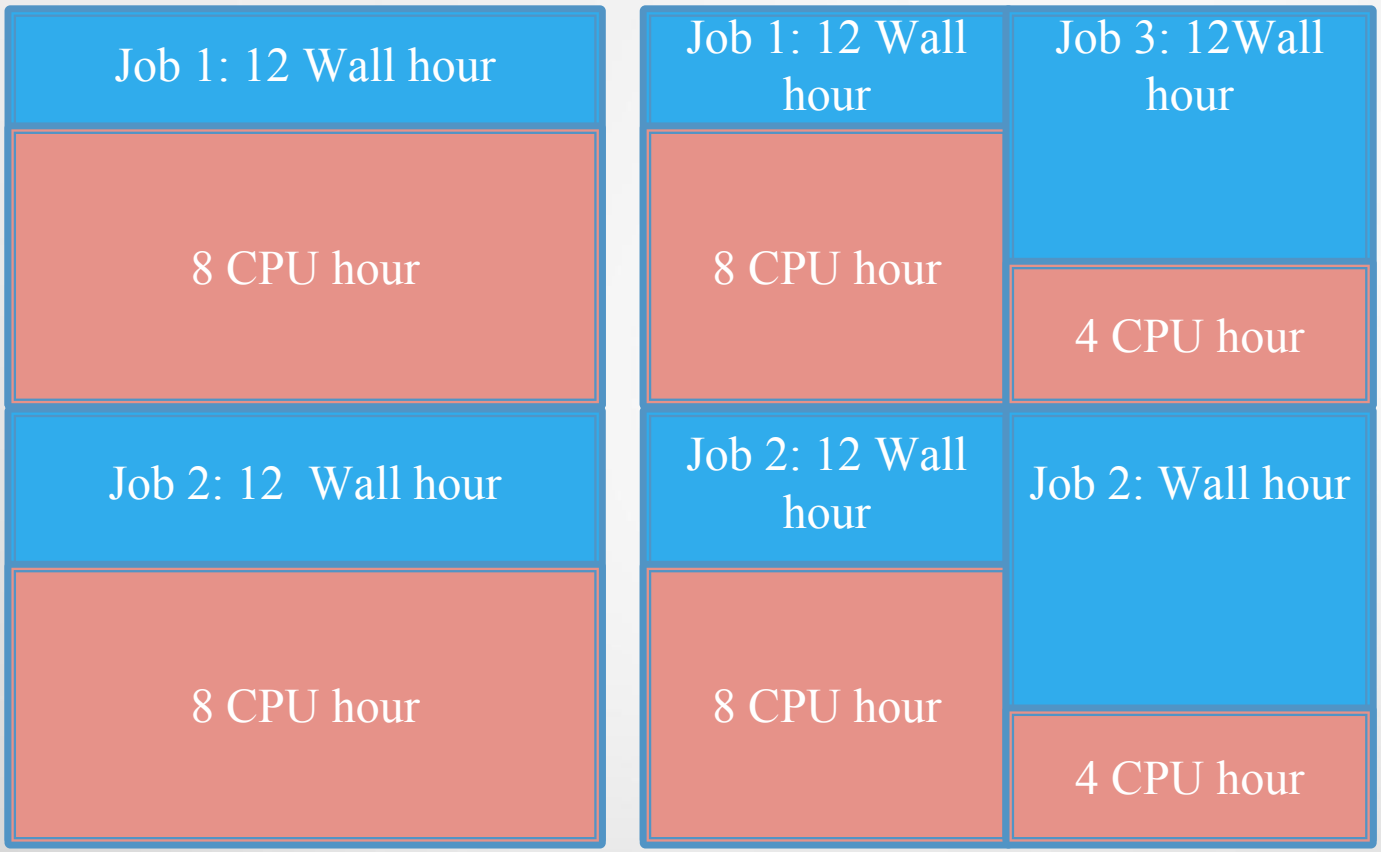
Tier 2 site utilization

Site CPU utilization in 100 days						
Site	cores	HS06	avg wall utilization	avg cpu utilization	HS06 utilization	avg job eff
BEIJING	634	11780	0.68	0.55	0.63	0.81
TOKYO	6144	111267	0.85	0.72	0.71	0.85
SiGNET	5288	51822	0.88	0.68	0.77	0.77
MWT2	16250	162500	0.83	0.70	0.73	0.84
AGLT2	10224	104414	0.72	0.61	0.63	0.84

Wall_util=wall_time/(cpu_cores*24)
CPU_util=cpu_time/(cpu_cores*24)
Based on data from the past 100 days

Data source: http://atlas-kibana-dev.mwt2.org/

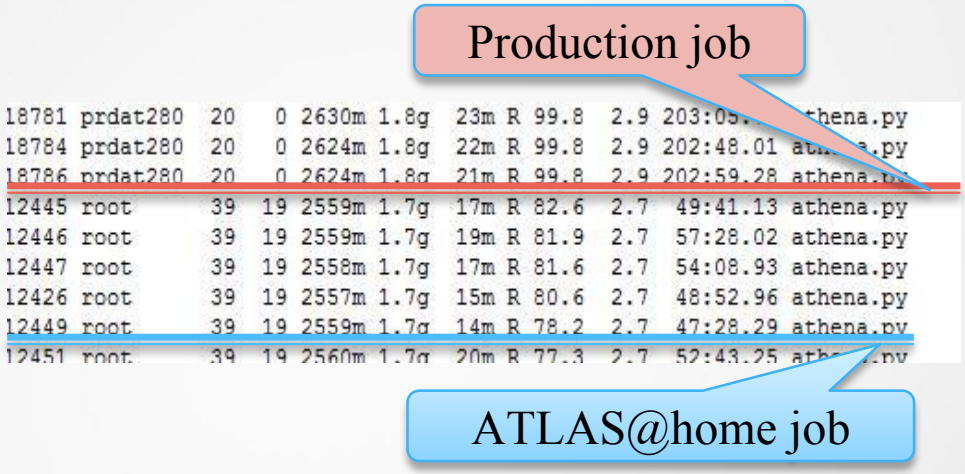
100% wall != 100% CPU utilization due to job efficiency



One work node
With job 1-2, 100% wall utilization, assume job eff 75%,then 25% CPU is wasted
With job 1-4, 200% wall utilization, 100% cpu utilization, job eff 75% and 25%

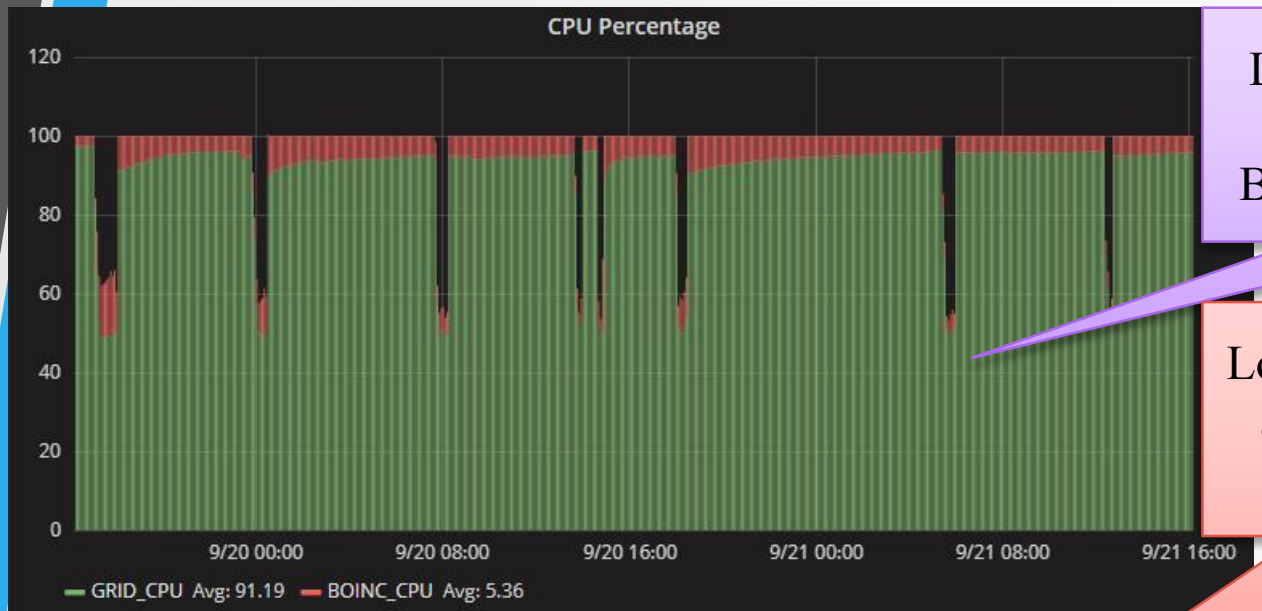
Put more jobs on work nodes?

- Not easy to manage with local scheduler
  - Define different job priority?
  - Define more job slots than available cores?



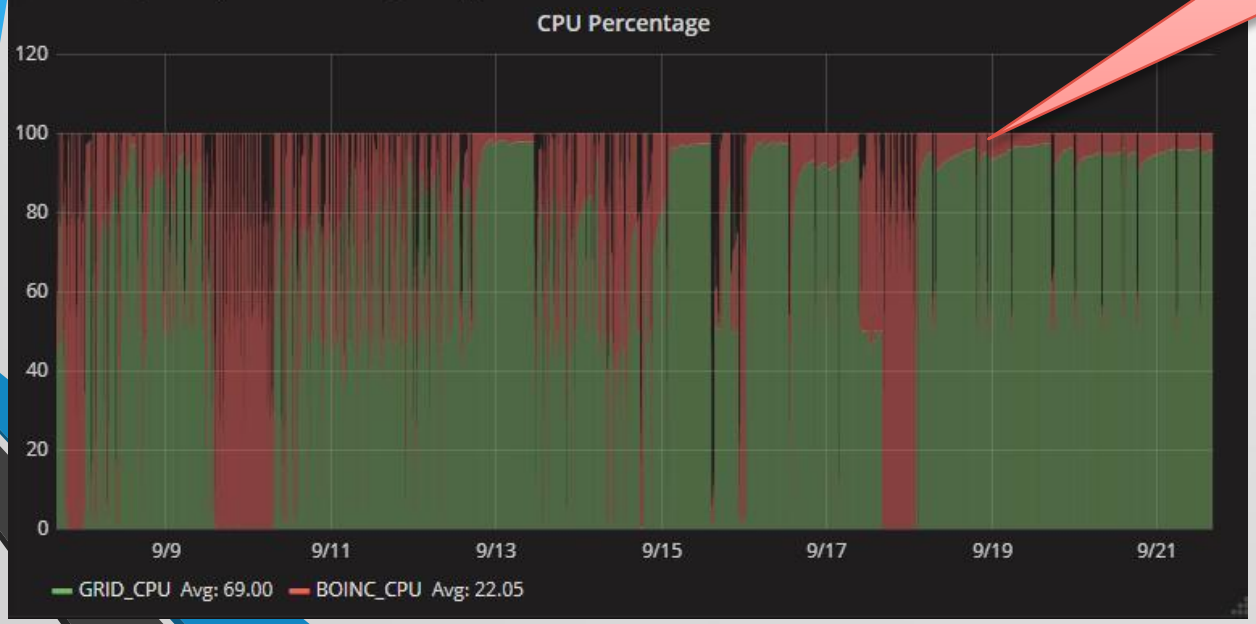
- With ATLAS@home jobs
  - BOINC is a second scheduler(requesting jobs from ATLAS@home), LRMS do not see it, can still push enough production jobs in.
  - BOINC jobs have the lowest priority in OS (PR=39), production jobs has the highest priority by default (PR=20)
  - On Linux OS, non-preemptive CPU scheduling: High priority process occupies CPU until it voluntarily releases it.
  - BOINC jobs release CPU/memory whenever the production jobs require
  - However, we configure BOINC job to be suspended in memory, so it don't lose the previous work.

Look into one work node



Look for 2 days, it is very full and CPU efficient, BOINC gets only 5% CPU

Look for 2 weeks, it is NOT always full, BOINC gets 22%



This node runs multi core production jobs (simul and pile), more cpu efficient than other type of jobs

Methodology

- BEIJING ATLAS Tier 2 site has dedicated 420 cores. (ATLAS jobs only), started to run from 25th August 2017
- Generate and visualize the results from ATLAS job stats (ES+Kibana)
  - For each job, cputime, walltime, cpu_eff, cputime_perevent, hs06sec etc.
  - Kibana: http://atlas-kibana-dev.mwt2.org/
- Deployed local monitor collects information (Agent+ES+Grafana)
  - #Job :
    - # ATLAS/CMS jobs, # of processes
    - # BOINC running jobs, # BOINC processes, # BOINC queued jobs
  - Node health:
    - system load/ free MEM/ used SWAP/ BOINC_used_MEM/idle CPU/
  - CPU utilization:
    - Grid CPU usage, BOINC CPU usage

How much can be exploited?

Tier2 Utilization : data from ATLAS Job stats.

BEIJING_LCG2 daily utilization in 3 weeks										
	#cores	HS06	Walltime (days)	CPUtime (days)	HS06 (days)	Job CPU eff (%)	Wall Util (%)	CPU Util (%)	HS06 Util (%)	Extra Cores
Grid	420	7560	359	283	5591	78.83	85.48	67.38	73.96	
BOINC	420	7560	367	97	1730	26.43	87.38	23.10	22.88	144.0
Total	420	7560	726	380	7321		172.86	90.48	96.84	
Note: site has 420 dedicated cores for ATLAS, numbers are the average in a consective 3 weeks (starting 2017-08-25), BOINNC processes **0.03 M events/day**										

BEIJING_LCG2 Work Nodes						
	#cores	RAM/core	Job	max PSS/core	avg PSS/core	Total cores
12score	12	2 GB	analysis/evgen	1.8/1.6 GB	0.6/1.2 GB	84
12mcore	12	2 GB	simul/pile	1.7 GB	1.2GB	96
24mcore	24	2.6 GB	simul/pile	1.7 GB	1.2GB	240
Total						420

BEIJING Tier2 site:

- 420 Cores (No HT), 420 job slots, PBS managed
- 84 cores running single core (15HS06)
- 336 cores running multi core (18HS06), 12 core/job

Exploited an extra **144** Cores out from **420** Cores in the past 3 week continuous running

CPU utilization reaches 90%! (CPU utilization from Grid jobs is 67%)

Tier 2 utilization: data from local monitor

BEIJING ATLAS Tier2 Work Nodes status in 24 hours										
	#cores	RAM/core	running Grid jobs	Grid Util	Grid CPU pct	BOINC CPU pct	Total CPU util	Max(load / used_swap / BOINC_MEM)	Avg(load / used swap / BOINC_MEM)	BOINC Config
12score	12	2 GB	11.2	93.33	58.89	28.12	87.01	37.4 / 7% / 3.4GB	18.3 / 2.4% / 1.5GB	1(job) x 9(core)
12mcore	12	2 GB	12	100.00	88.25	7.13	95.38	34.7 / 1.6% / 7.1GB	24 / 0.7% / 4.5GB	2(job) x 6(core)
24mcore	24	2.6 GB	22.2	92.50	80.43	12.8	93.23	58 / 0.6% / 6.5GB	41 / 0.4% / 4.5GB	4(job) x 6(core)
avg	420		395	94.05	77	15	92			
Note: Data taken in 1 day when work nodes are fully loaded with grid jobs										

BEIJING ATLAS Tier2 Work Nodes status in 3 weeks										
	#cores	RAM/core	running Grid jobs	Grid Util	Grid CPU pct	BOINC CPU pct	Total CPU util	Max(load / used_swap / BOINC_MEM)	Avg(load / used swap / BOINC_MEM)	BOINC Config
12score	12	2 GB	10.62	88.50	54.1	28.81	82.91	51.7 / 27.9% / 3.4GB	18.5 / 3.98% / 1.5GB	2(job) x 6(core)
12mcore	12	2 GB	10.3	85.83	67.07	22.35	89.42	28.4 / 7.5% / 4.7GB	21.8 / 2.1% / 3.1GB	2(job) x 6(core)
24mcore	24	2.6 GB	20.53	85.54	68.94	21.27	90.21	59 / 7% / 10GB	40 / 0.6% / 4.4GB	4(job) x 6(core)
avg	420		361	85.95	64.18	23.73	87.91			
Note: Data taken in a 3 week period										

When production wall utilization is over **94%**, ATLAS@home exploits **15%** CPU time

When production is continuously around **86%**, ATLAS@home exploits **24%** CPU time

Work nodes are “calm” in terms of load, memory usage, swap usage

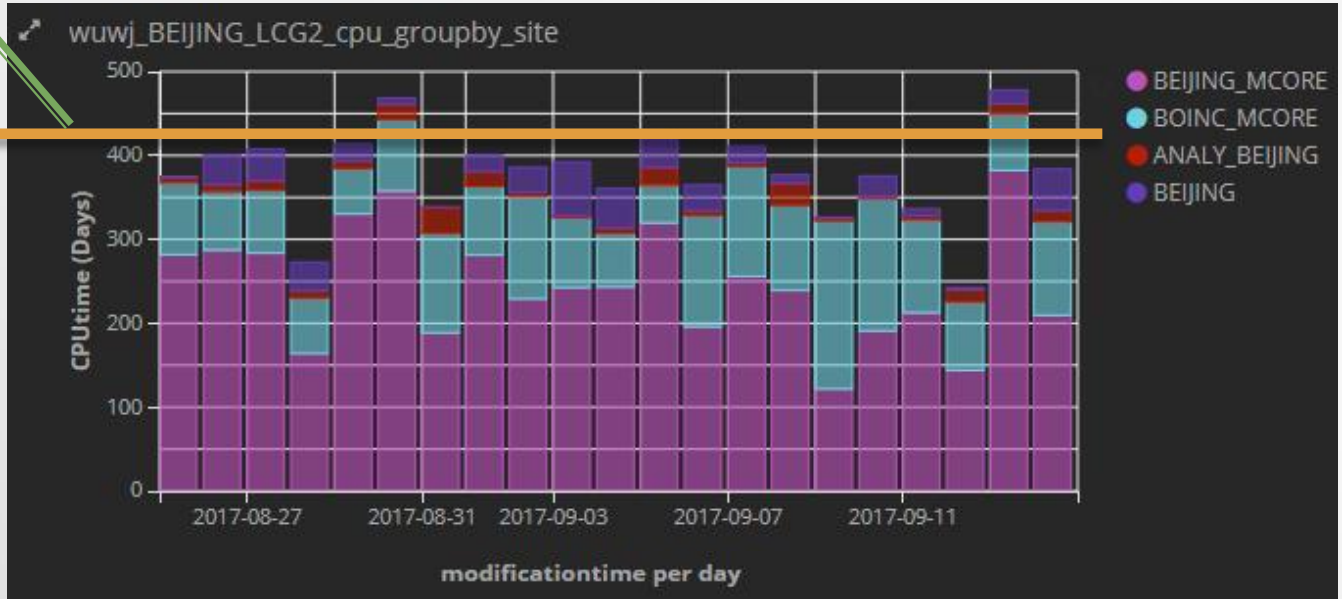
Avg. CPU utilization is **88%**

Data comparison

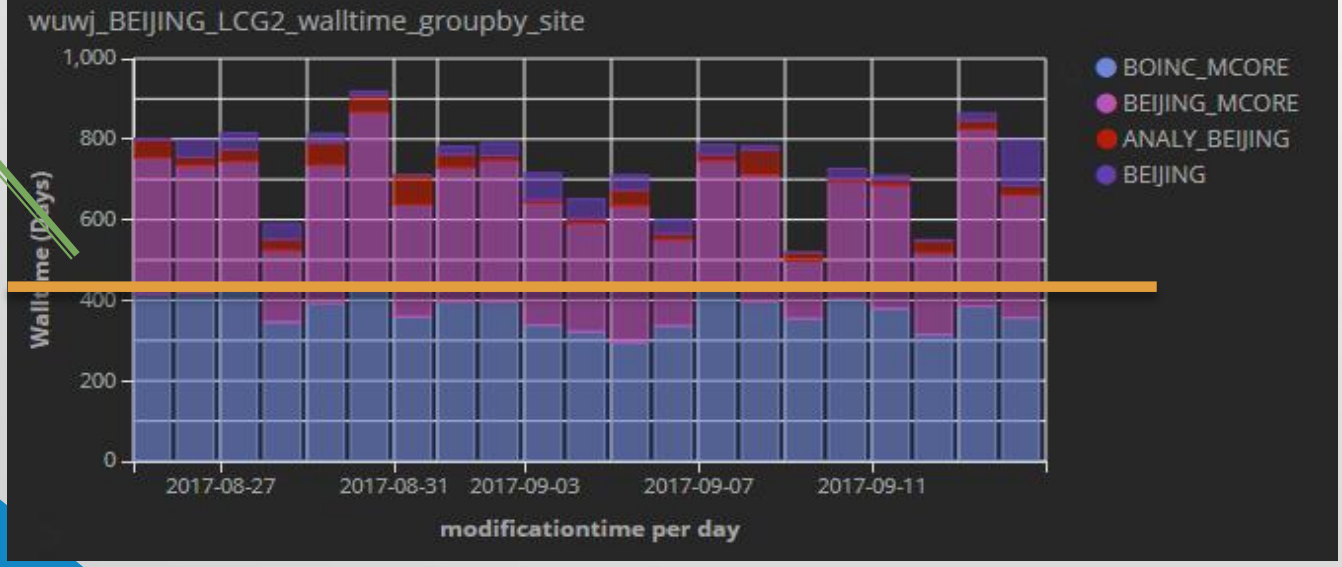
- Data source: ATLAS Job stats (ES+Kibana) vs. Local Monitor (Agent+ES+Grafana)
- Consistent
  - Grid Jobs:
    - Walltime utilization is 86% in both
    - CPU time utilization is 67%(ATLAS) vs. 64% (Local)
  - BOINC Jobs:
    - CPU time utilization is 23% in both
  - Work nodes total CPU time utilization
    - 90% (ATLAS) vs. 88% (Local)

Utilization from ATLAS Job stats.

420 Cores



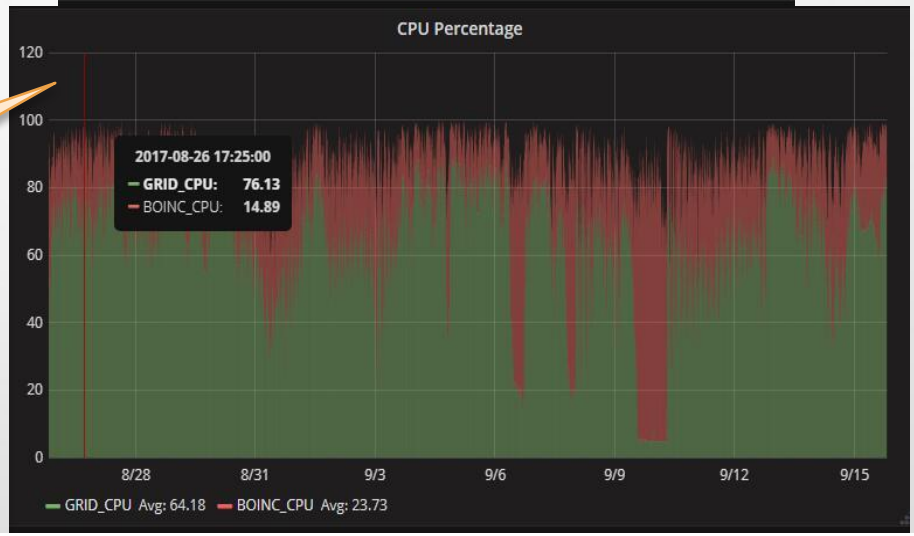
420 Cores



Utilization from Local monitor

Running Processes and Jobs of ATLAS Grid jobs

CPU utilization
ATLAS Grid jobs: 64.18%
BOINC: 23.73%

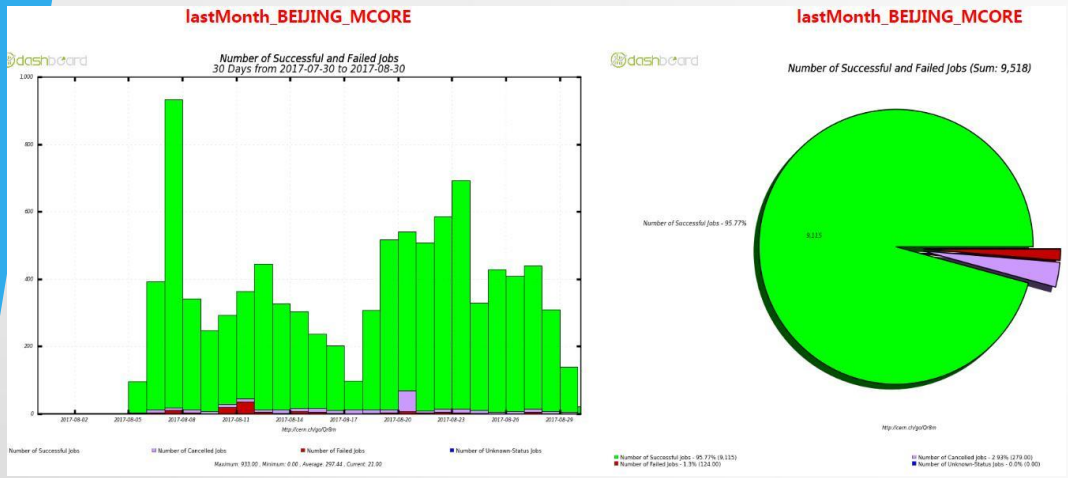


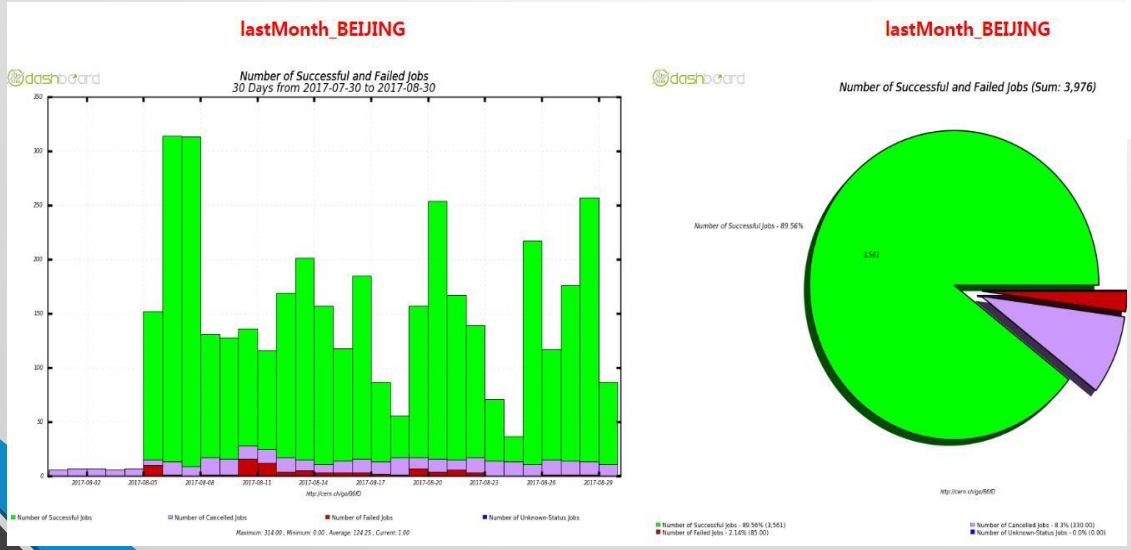
Two scheduled downtime (each 6 hours) during the 3 weeks

Impact on Tier2?

- Stability
  - It does not affect the production job failure rate (1-2%)
  - It does not affect the throughput of production jobs
    - with running BOINC, ATLAS grid jobs walltime utilization is around 90%
  - Work nodes have reasonable load/swap usage
- Efficiency
  - For production jobs , it does not make it less efficient
    - CPU efficient is less than 1% difference or NONE?!
  - For BOINC jobs, it does not make it less efficient
    - CPU time per event is 0.02% difference between dedicated nodes and nodes mixed with production jobs.
- Manpower to maintain
  - After the right configuration, no manual intervention is needed

Production jobs

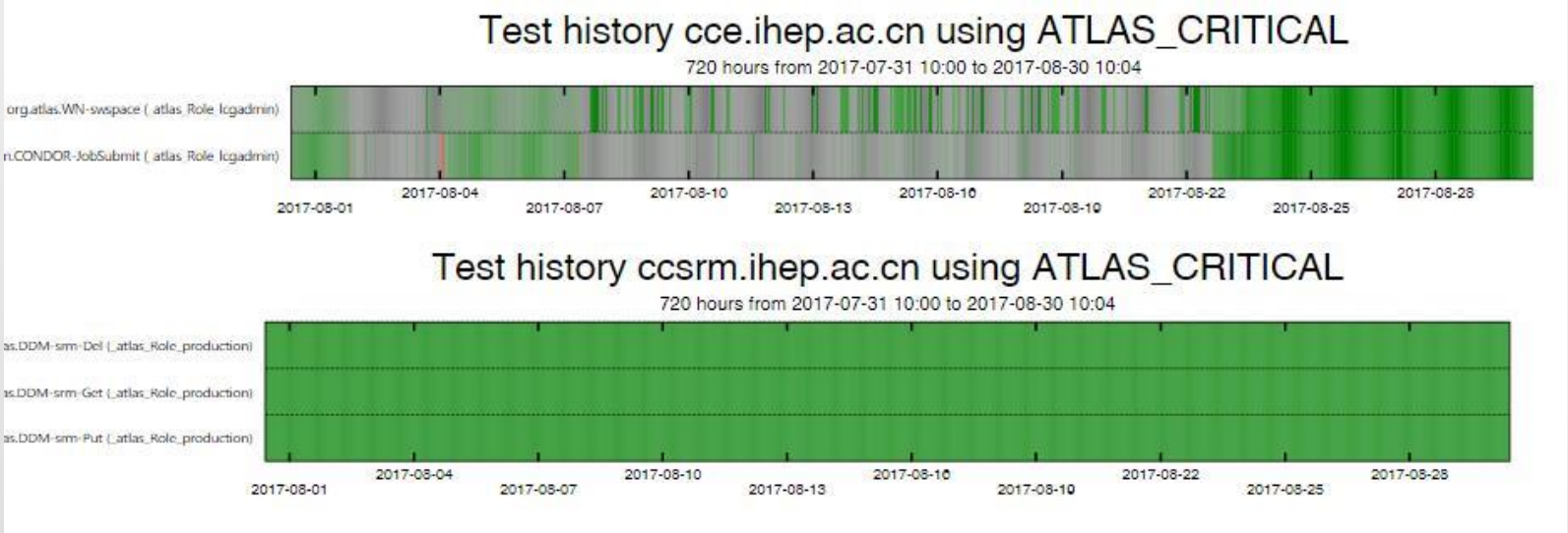




1. Production jobs in a month: (with BOINC jobs) 1-2% failure rate

2. In recent 3 weeks, production jobs uses 90% of the walltime

Site SAM Tests in a month



Site Reliability : **99.29%**

Job Efficiency compare

	Average cpu_eff per job	Average cpu_eff
Production Job (without BOINC)	0.9429	0.9573
Production Job (with BOINC)	0.9384	0.9441
with VS. without	-0.48%	-1.38%

select jobs with : jobstatus=finished, processingtype=simul, nevents=1000, actualcorecount=12, avg_cpu_eff=avg_cputime_day/avg_walltime_day

	Average cputime_day per job	Average cpu_sec_perevent
BOINC Job (Mixed production)	0.2002	306.0081
BOINC Job (Dedicated)	0.1983	305.9520
Dedicated VS. Mixed	-0.94%	-0.02%

select jobs with : jobstatus=finished, processingtype=simul, avg_cpu_eff=avg_cputime_day/avg_walltime_day, nevents=(50,100), both mixed and dedicated nodes are of the same model, 5 mixed nodes, 2 dedicated nodes, mixed nodes all run single core production jobs

Production Job Efficiency compare

6 weeks running **without** BOINC jobs

8 weeks running **with** BOINC jobs

During different period, run different tasks, cpu time per event can be very different, so we only compare cpu_eff

BEIJING MCORE Simul Job Efficiency compare(with vs. without BOINC jobs)						
modificationtime per week	Average cputime_day	Average walltime_day	Average nevents per job	Average cpu_sec_perevent	Average cpu_eff per job	Average cpu_eff
June 12th 2017	2.685	2.873	1,000	231.984	89.83%	0.9346
June 19th 2017	6.531	6.719	1,000	564.2784	97.19%	0.9720
June 26th 2017	1.811	1.914	1,000	156.4704	93.58%	0.9462
July 3rd 2017	3.32	3.441	1,000	286.848	94.39%	0.9648
July 10th 2017	2.258	2.367	1,000	195.0912	93.73%	0.9540
July 17th 2017	5.347	5.499	1,000	461.9808	97.04%	0.9724
Avg(without BOINC)	3.6587	3.8022	1000	316.1088	0.9429	0.9573
July 24th 2017	3.267	3.377	1,000	282.2688	95.78%	0.9674
July 31st 2017	1.485	1.578	1,000	128.304	94.02%	0.9411
August 7th 2017	2.134	2.24	1,000	184.3776	95.08%	0.9527
August 14th 2017	2.131	2.263	1,000	184.1184	93.79%	0.9417
August 21st 2017	1.368	1.484	1,000	118.1952	91.96%	0.9218
August 28th 2017	2.953	3.125	1,000	255.1392	93.20%	0.9450
September 4th 2017	2	2.136	1,000	172.8	93.03%	0.9363
September 11th 2017	3.43	3.623	1,000	296.352	93.87%	0.9467
Avg(with BOINC)	2.3460	2.4783	1000	202.6944	0.9384	0.9441
with VS. without	-35.88%	-34.82%	0.00%	-35.88%	-0.48%	-1.38%
select jobs with : jobstatus=finished, processingtype=simul, nevents=1000, actualcorecount=12, avg_cpu_eff=avg_cputime_day/avg_walltime_day						

BOINC Job Efficiency compare

9 weeks running on nodes mixed with production jobs

9 weeks running on same nodes dedicated for BOINC jobs

Same period (same tasks, cpu time per event is similar), same nodes, so we compare cpu time per event

BOINC MCORE Simul Job Efficiency compare(mixed vs. dedicated resource)

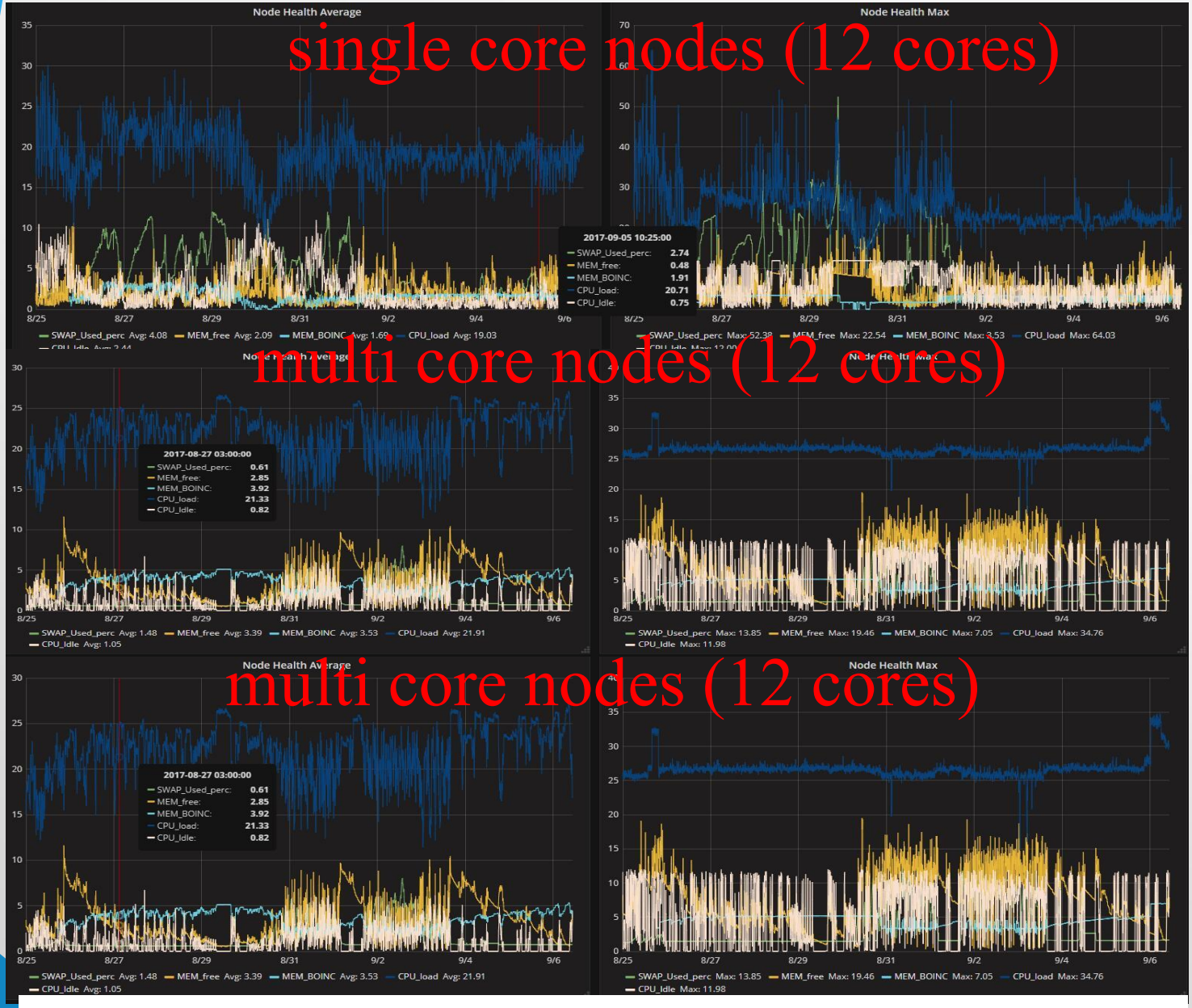
modificationtime per week	Average cputime_day	Average walltime_day	Average nevents per job	Average cpu_sec_perevent	Average cpu_eff per job	Average cpu_eff
July 17th 2017	0.18	0.284	50	311.04	72.51%	0.6338
July 24th 2017	0.179	0.216	52.601	294.017224	84.49%	0.8287
July 31st 2017	0.196	0.231	99.438	170.3010921	87.44%	0.8485
August 7th 2017	0.201	0.81	100	173.664	29.54%	0.2481
August 14th 2017	0.187	0.245	50.38	320.69869	80.38%	0.7633
August 21st 2017	0.142	0.486	50	245.376	41.37%	0.2922
August 28th 2017	0.195	0.552	50	336.96	51.46%	0.3533
September 4th 2017	0.244	0.567	50	421.632	57.43%	0.4303
September 11th 2017	0.278	0.744	50	480.384	54.53%	0.3737
Avg (mixed BOINC with production)	0.2002	0.4594	61.3799	306.0081	0.6213	0.5302
July 17th 2017	0.179	0.278	50	309.312	75.05%	0.6439
July 24th 2017	0.178	0.203	50	307.584	90.26%	0.8768
July 31st 2017	0.194	0.302	100	167.616	82.98%	0.6424
August 7th 2017	0.189	0.271	100	163.296	71.69%	0.6974
August 14th 2017	0.182	0.192	50	314.496	94.63%	0.9479
August 21st 2017	0.137	0.18	50	236.736	76.66%	0.7611
August 28th 2017	0.205	0.248	50	354.24	81.32%	0.8266
September 4th 2017	0.243	0.288	50	419.904	81.95%	0.8438
September 11th 2017	0.278	0.325	50	480.384	81.46%	0.8554
Avg (Dedicated BOINC)	0.1983	0.2541	61.1111	305.9520	0.8178	0.7884
Dedicated VS. Mixed	-0.94%	-44.69%	-0.44%	-0.02%	31.63%	48.69%

select jobs with : jobstatus=finished, processingtype=simul, avg_cpu_eff=avg_cputime_day/avg_walltime_day, nevents=(50,100), both mixed and dedicated nodes are of the same model, 5 mixed nodes, 2 dedicated nodes, mixed nodes all run single core production jobs

Work nodes health status

BEIJING ATLAS Tier2 Work Nodes status in 3 weeks										
	#cores	RAM/core	running Grid jobs	Grid Util	Grid CPU pct	BOINC CPU pct	Total CPU util	Max(load / used_swap / BOINC_MEM)	Avg(load / used swap / BOINC_MEM)	BOINC Config
12score	12	2 GB	10.94	91.17	59.04	25.68	84.72	51.7 / 27.9% / 3.4GB	18.5 / 2.98% / 1.5GB	2(job) x 6(core)
12mcore	12	2 GB	10.76	89.67	69.94	19.11	89.05	34.7 / 4.5% / 10GB	21.8 / 2.1% / 3.1GB	2(job) x 6(core)
24mcore	24	2.6 GB	22.2	92.50	70.73	18.9	89.63	59.6 / 0.6% / 6.5GB	39.7 / 0.83% / 4.46GB	4(job) x 6(core)
avg	420		356	84.76	64.21	24.33	88.54			
Note: Data taken in a 3 week period										

- Avg. sys load is less than 2.5 times of the CPU cores
- Swap usage is less than 3% (no memory competition)
- BOINC jobs not use much memory



Monitor: Load, BOINC_used_memory, used_swap, idle_CPU

Can this be tested on other Tier2 site?

- Scripts available for fast deployment on a cluster
- Local monitor scripts available
- ATLAS@home scalability:
  - The current ATLAS@home server can survive 20K jobs/day, bottle neck from submitting to ARC CE (IO)
  - Can run multiple ARC CE with multiple BOINC servers to receive jobs from BOINC_MCORE (PanDA que)
- The extra exploited CPU time can be “counted” to sites in ATLAS Job stats. (Kibana visualization)

Summary

- ATLAS@home has been providing continuous resource since 2013
  - Equal to a Grid site with 2700 Cores (10 HS06)
- With recent optimization in scheduling and newly added Tier2 work nodes (serving as stable host group) , it can run more important tasks
- An 23% of extra CPU can be exploited from a full loaded Tier2 site (with 90% Wall utilization), this also increases the CPU utilization to 90%
- The extra BOINC jobs do not impact the health of work nodes or throughput of production jobs on the Tier2 site, and it does not require extra manpower to maintain once it is setup appropriately.

Acknowledgements

- Andrej Filipcic (Jozef Stefan Institute, Slovenia)
- Nils Hoimyr (CERN IT, BOINC Project)
- Ilija Vukotic (University of Chicago)
- Frank Berghaus (University of Victoria)
- Douglas Gingrich (University of Victoria)
- Paul Nilsson (BNL)
- Rod Walker (LMU-München)

Thanks!

Backup slides

Tier 2 Utilization in details

BEIJING									
cores	HS06	mtime per 10 days	10 days walltime	10 days cputime	10 days HS06	walltime utilization	cpu utilization	HS06 utilization	job cpu eff
634	11780	May 14th 2017	7,243.77	6,002.40	115,658.73	1.14	0.95	0.98	0.83
634	11780	May 24th 2017	6,802.46	6,089.43	112,091.12	1.07	0.96	0.95	0.90
634	11780	June 3rd 2017	7,213.58	6,091.04	120,323.64	1.14	0.96	1.02	0.84
634	11780	June 13th 2017	2,125.76	1,447.02	38,959.49	0.34	0.23	0.33	0.68
634	11780	June 23rd 2017	4,968.71	4,095.54	87,060.96	0.78	0.65	0.74	0.82
634	11780	July 3rd 2017	2,653.50	2,039.65	45,407.56	0.42	0.32	0.39	0.77
634	11780	July 13th 2017	5,175.79	4,370.64	113,886.89	0.82	0.69	0.97	0.84
634	11780	July 23rd 2017	1,226.11	1,060.23	24,384.51	0.19	0.17	0.21	0.86
634	11780	August 2nd 2017	1,936.42	1,126.01	28,781.12	0.31	0.18	0.24	0.58
634	11780	August 12th 2017	3,539.47	2,404.54	55,101.38	0.56	0.38	0.47	0.68
634	11780	Average	4288.56	3472.65	74165.54	**0.68**	**0.55**	**0.63**	0.81
Note: group data in every 10 days, to avoild jobs running across the day(s)									

Tokyo									
cores	HS06	mtime per 10 days	10 days walltime	10 days cputime	10 days HS06	walltime utilization	cpu utilization	HS06 utilization	job cpu eff
6144	111267	May 14th 2017	59,937.18	50,263.77	921,379.72	0.98	0.82	0.83	0.84
6144	111267	May 24th 2017	44,589.70	34,340.65	598,310.12	0.73	0.56	0.54	0.77
6144	111267	June 3rd 2017	42,597.56	37,666.91	664,571.78	0.69	0.61	0.60	0.88
6144	111267	June 13th 2017	58,369.24	50,039.98	918,773.13	0.95	0.81	0.83	0.86
6144	111267	June 23rd 2017	58,750.68	50,734.78	904,148.19	0.96	0.83	0.81	0.86
6144	111267	July 3rd 2017	54,700.64	46,657.37	832,151.42	0.89	0.76	0.75	0.85
6144	111267	July 13th 2017	58,588.21	54,982.41	924,000.53	0.95	0.89	0.83	0.94
6144	111267	July 23rd 2017	41,755.10	36,259.31	641,469.05	0.68	0.59	0.58	0.87
6144	111267	August 2nd 2017	46,609.56	37,657.49	707,925.34	0.76	0.61	0.64	0.81
6144	111267	August 12th 2017	53,775.70	40,939.76	827,854.21	0.88	0.67	0.74	0.76
6144	111267	Average	51967.36	43954.24	794058.35	**0.85**	**0.72**	**0.71**	0.85

SiGNET									
cores	HS06	mtime per 10 days	10 days walltime	10 days cputime	10 days HS06	walltime utilization	cpu utilization	HS06 utilization	job cpu eff
5288	51822	May 14th 2017	43,837.56	33,186.65	381,668.74	0.83	0.63	0.74	0.76
5288	51822	May 24th 2017	56,107.74	47,816.65	496,579.71	1.06	0.90	0.96	0.85
5288	51822	June 3rd 2017	56,426.82	48,449.91	493,351.91	1.07	0.92	0.95	0.86
5288	51822	June 13th 2017	49,475.25	38,947.49	425,916.12	0.94	0.74	0.82	0.79
5288	51822	June 23rd 2017	42,683.41	31,299.46	358,490.39	0.81	0.59	0.69	0.73
5288	51822	July 3rd 2017	31,702.16	21,065.26	262,378.37	0.60	0.40	0.51	0.66
5288	51822	July 13th 2017	40,319.96	28,992.77	347,505.45	0.76	0.55	0.67	0.72
5288	51822	July 23rd 2017	53,089.23	41,201.41	456,109.62	1.00	0.78	0.88	0.78
5288	51822	August 2nd 2017	50,605.24	37,026.78	429,010.54	0.96	0.70	0.83	0.73
5288	51822	August 12th 2017	41,359.06	32,367.40	361,261.17	0.78	0.61	0.70	0.78
5288	51822	Average	46560.64	36035.38	401227.20	**0.88**	**0.68**	**0.77**	0.77

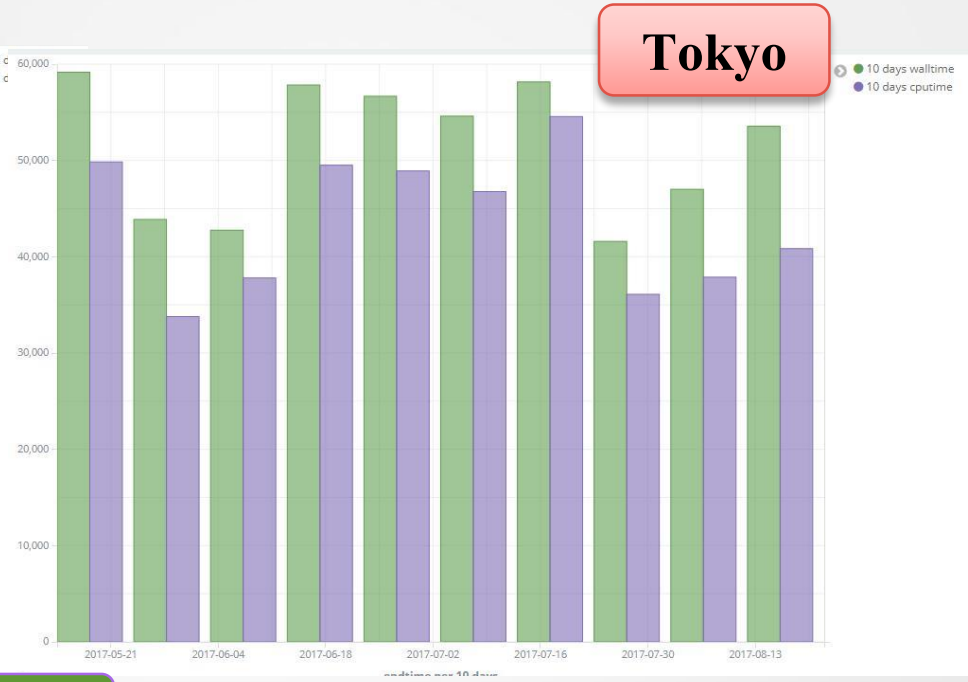
CPU/Wall/HS06 time are presented in “days” and grouped by every 10 days

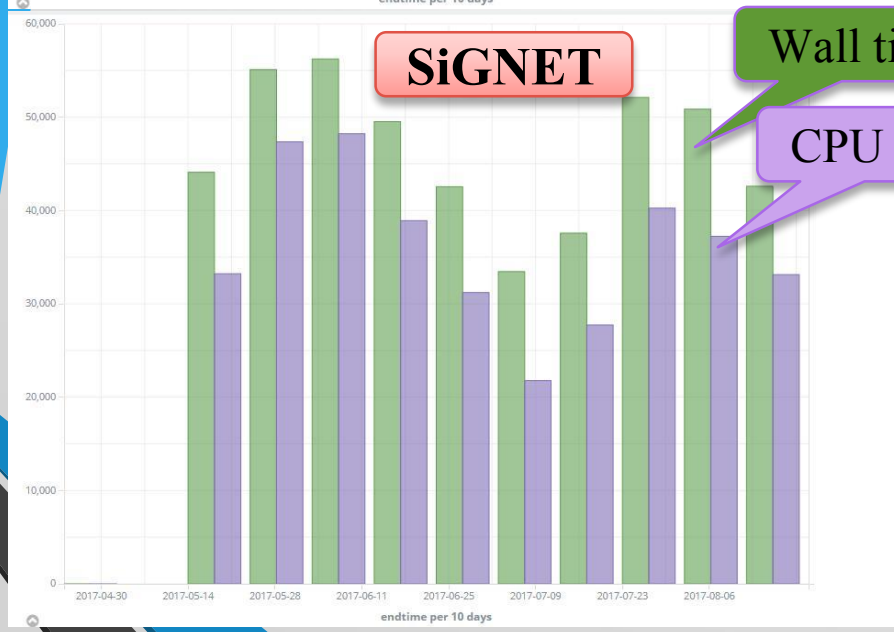
MWT2									
cores	HS06	mtime per 10 days	10 days walltime	10 days cputime	10 days HS06	walltime utilization	cpu utilization	HS06 utilization	job cpu eff
16250	162500	May 14th 2017	102,096.17	78,304.62	908,609.18	0.63	0.48	0.56	0.77
16250	162500	May 24th 2017	131,945.71	117,034.11	1,164,518.99	0.81	0.72	0.72	0.89
16250	162500	June 3rd 2017	159,019.16	139,162.17	1,439,723.33	0.98	0.86	0.89	0.88
16250	162500	June 13th 2017	144,684.53	121,960.79	1,294,674.63	0.89	0.75	0.80	0.84
16250	162500	June 23rd 2017	142,468.27	116,697.84	1,149,473.81	0.88	0.72	0.71	0.82
16250	162500	July 3rd 2017	156,036.04	128,261.24	1,389,739.82	0.96	0.79	0.86	0.82
16250	162500	July 13th 2017	159,113.56	141,542.55	1,381,017.71	0.98	0.87	0.85	0.89
16250	162500	July 23rd 2017	137,467.48	120,017.21	1,226,565.33	0.85	0.74	0.75	0.87
16250	162500	August 2nd 2017	147,819.50	120,317.84	1,269,273.12	0.91	0.74	0.78	0.81
16250	162500	August 12th 2017	70,404.53	54,576.39	597,448.97	0.43	0.34	0.37	0.78
16250	162500	Average	135105.49	113787.48	1182104.49	**0.83**	**0.70**	**0.73**	0.84

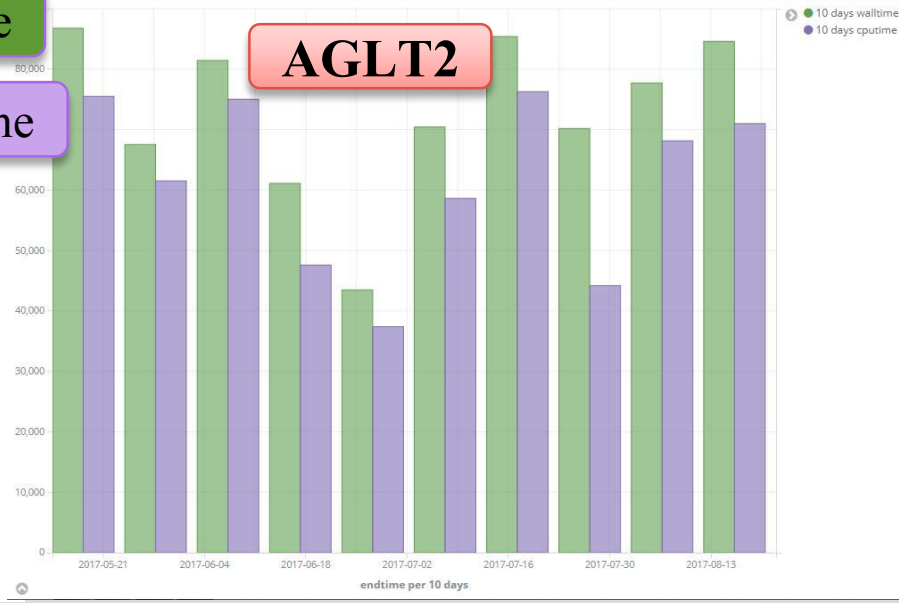
Tier 2 Utilization in details

AGLT2									
cores	HS06	mtime per 10 days	10 days walltime	10 days cputime	10 days HS06	walltime utilization	cpu utilization	HS06 utilization	job cpu eff
10224	104414	May 14th 2017	86,875.84	75,543.79	781,167.12	0.85	0.74	0.75	0.87
10224	104414	May 24th 2017	68,924.17	62,704.09	620,776.06	0.67	0.61	0.59	0.91
10224	104414	June 3rd 2017	81,087.97	74,660.99	732,012.98	0.79	0.73	0.70	0.92
10224	104414	June 13th 2017	62,139.56	48,565.15	557,444.11	0.61	0.48	0.53	0.78
10224	104414	June 23rd 2017	43,033.62	36,960.71	380,756.92	0.42	0.36	0.36	0.86
10224	104414	July 3rd 2017	73,071.43	60,900.61	649,634.52	0.71	0.60	0.62	0.83
10224	104414	July 13th 2017	83,403.15	74,860.50	758,270.75	0.82	0.73	0.73	0.90
10224	104414	July 23rd 2017	72,099.21	45,585.11	614,101.48	0.71	0.45	0.59	0.63
10224	104414	August 2nd 2017	77,606.26	67,880.90	695,835.44	0.76	0.66	0.67	0.87
10224	104414	August 12th 2017	85,707.36	71,811.43	770,093.47	0.84	0.70	0.74	0.84
10224	104414	Average	73394.86	61947.33	656009.29	**0.72**	**0.61**	**0.63**	0.84

BEIJING
8,000
7,000
6,000
5,000
4,000
3,000
2,000
1,000
0
2017-05-21
2017-06-04
2017-06-18
2017-07-02
2017-07-16
2017-07-30
2017-08-13
endtime per 10 days

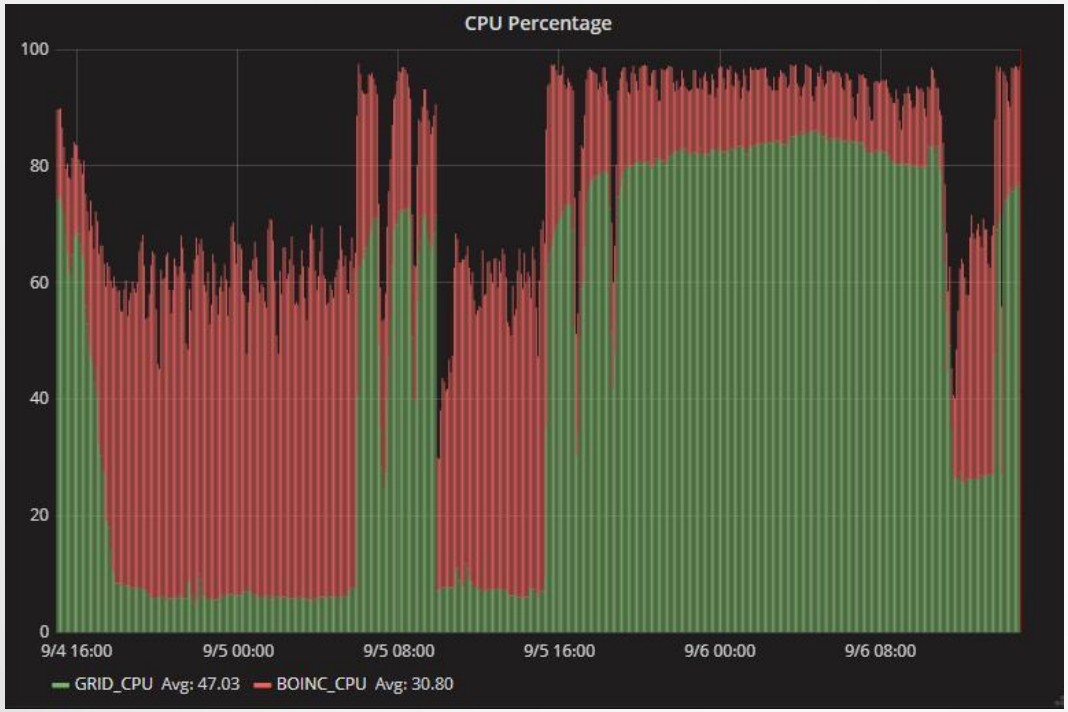

Tokyo
10 days walltime
10 days cputime
60,000
50,000
40,000
30,000
20,000
10,000
0
2017-05-21
2017-06-04
2017-06-18
2017-07-02
2017-07-16
2017-07-30
2017-08-13

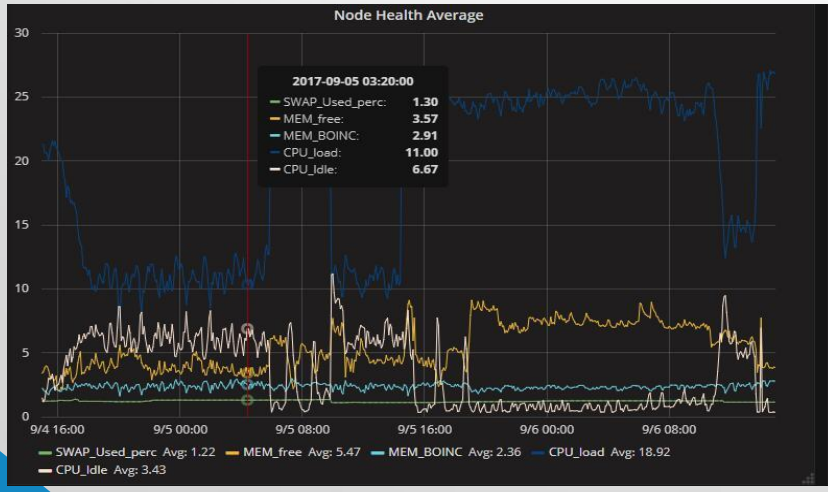

SiGNET
Wall time
CPU time
60,000
50,000
40,000
30,000
20,000
10,000
0
2017-04-30
2017-05-14
2017-05-28
2017-06-11
2017-06-25
2017-07-09
2017-07-23
2017-08-06
endtime per 10 days

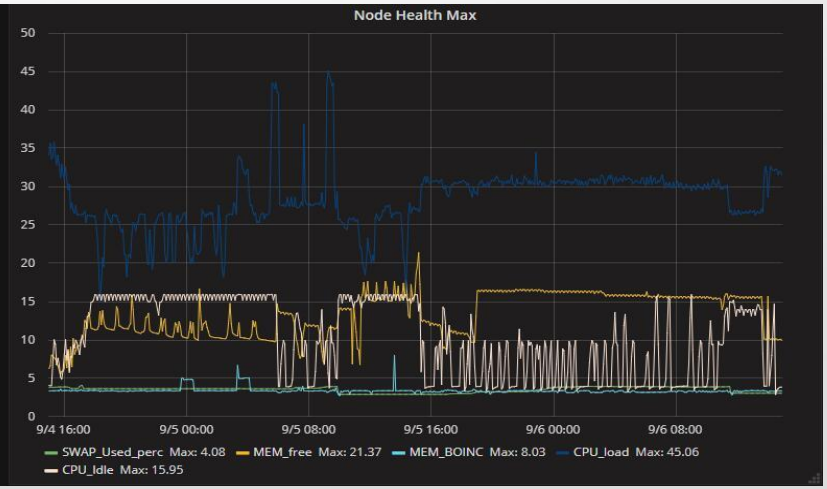

AGLT2
10 days walltime
10 days cputime
80,000
60,000
50,000
40,000
30,000
20,000
10,000
0
2017-05-21
2017-06-04
2017-06-18
2017-07-02
2017-07-16
2017-07-30
2017-08-13
endtime per 10 days

How about Tier3?

Use 10 ATLS Tier3 nodes for testing


CPU Percentage
100
80
60
40
20
0
9/4 16:00
9/5 00:00
9/5 08:00
9/5 16:00
9/6 00:00
9/6 08:00
GRID_CPU Avg: 47.03
BOINC_CPU Avg: 30.80


Node Health Average
2017-09-05 03:20:00
SWAP_Used_perc: 1.30
MEM_free: 3.57
MEM_BOINC: 2.91
CPU_load: 11.00
CPU_Idle: 6.67
30
25
20
15
10
5
0
9/4 16:00
9/5 00:00
9/5 08:00
9/5 16:00
9/6 00:00
9/6 08:00
SWAP_Used_perc Avg: 1.22
MEM_free Avg: 5.47
MEM_BOINC Avg: 2.36
CPU_load Avg: 18.92
CPU_Idle Avg: 3.43

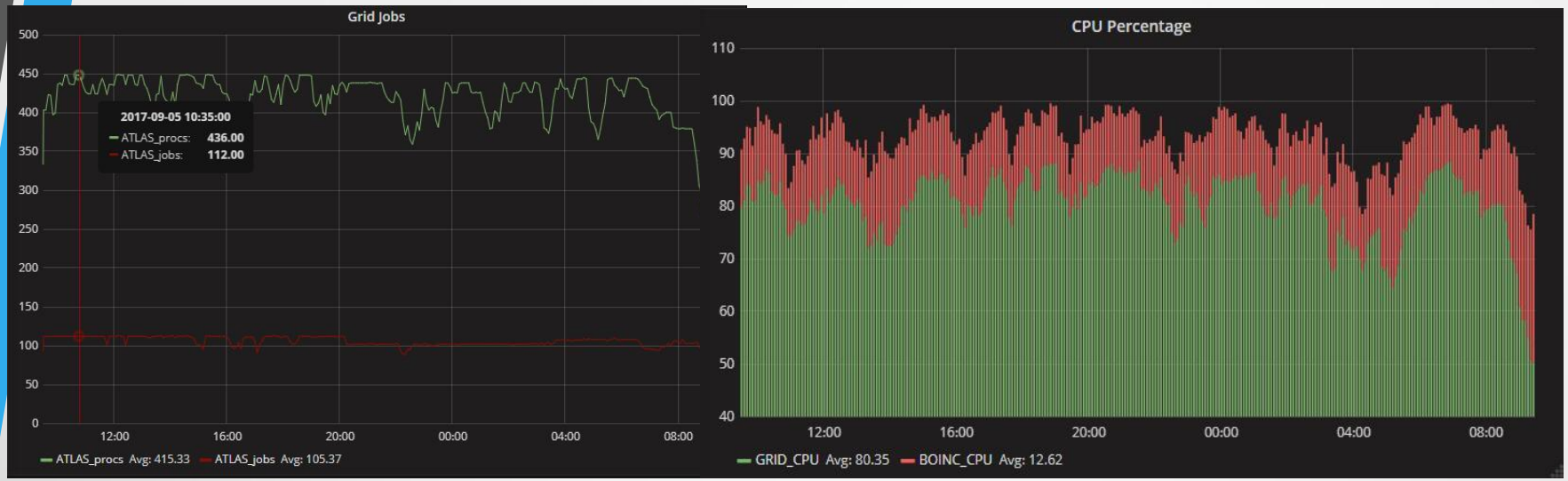

Node Health Max
50
45
40
35
30
25
20
15
10
5
0
9/4 16:00
9/5 00:00
9/5 08:00
9/5 16:00
9/6 00:00
9/6 08:00
SWAP_Used_perc Max: 4.08
MEM_free Max: 21.37
MEM_BOINC Max: 8.03
CPU_load Max: 45.06
CPU_Idle Max: 15.95

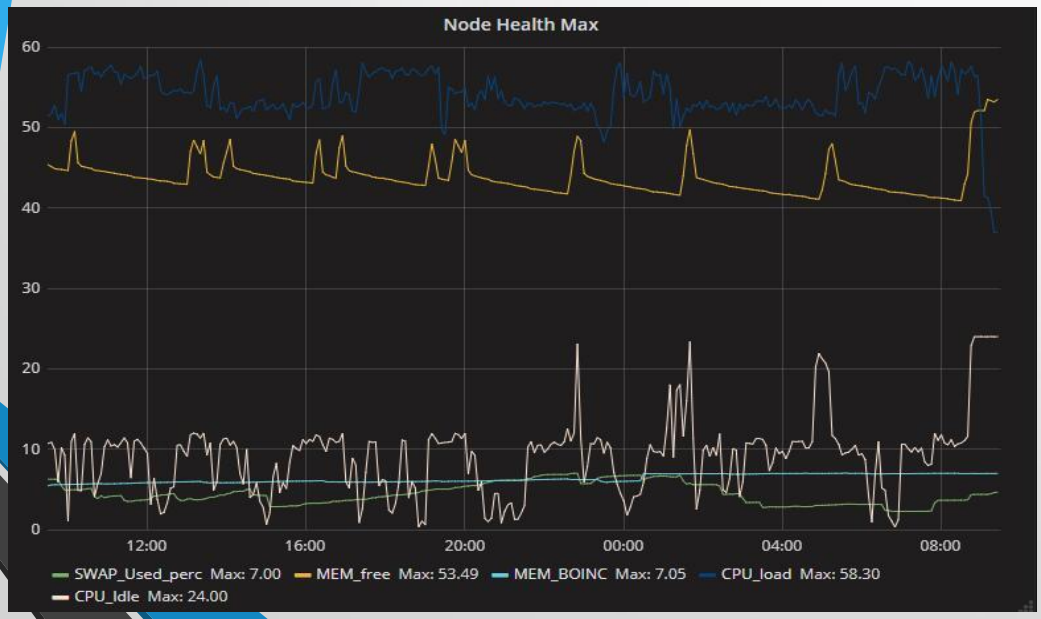
Tier3 node utilization summary

BEIJING ATLAS Tier2 Work Nodes status in 2 weeks										
	#cores	RAM/core	running Grid jobs	Grid Util	Grid CPU pct	BOINC CPU pct	Total CPU util	Max(load / used_swap / BOINC_MEM)	Avg(load / used swap / BOINC_MEM)	BOINC Config
12score	12	2 GB	10.94	91.17	59.04	25.68	84.72	51.7 / 27.9% / 3.4GB	18.5 / 2.98% / 1.5GB	2(job) x 6(core)
12mcore	12	2 GB	10.76	89.67	69.94	19.11	89.05	34.7 / 4.5% / 10GB	21.8 / 2.1% / 3.1GB	2(job) x 6(core)
24mcore	24	2.6 GB	22.2	92.50	70.73	18.9	89.63	59.6 / 0.6% / 6.5GB	39.7 / 0.83% / 4.46GB	4(job) x 6(core)
avg	420		374	89.05	67.2	20.87	88.07			
Note: Data taken in a 2 week period										

Tier3 Work Nodes status in 2 days										
	#cores	RAM/core	running Grid jobs	Grid Util	Prod CPU pct	BOINC CPU pct	Total CPU util	Max(load / used_swap / BOINC_MEM)	Avg(load / used swap / BOINC_MEM)	BOINC Config
aws nodes	12	2 GB		0.00	47.03	30.8	77.83	45 / 4% / 8GB	18.9 / 1.2% / 2.41GB	2(job) x 6(core)
Note: Data taken in a 2 day period										

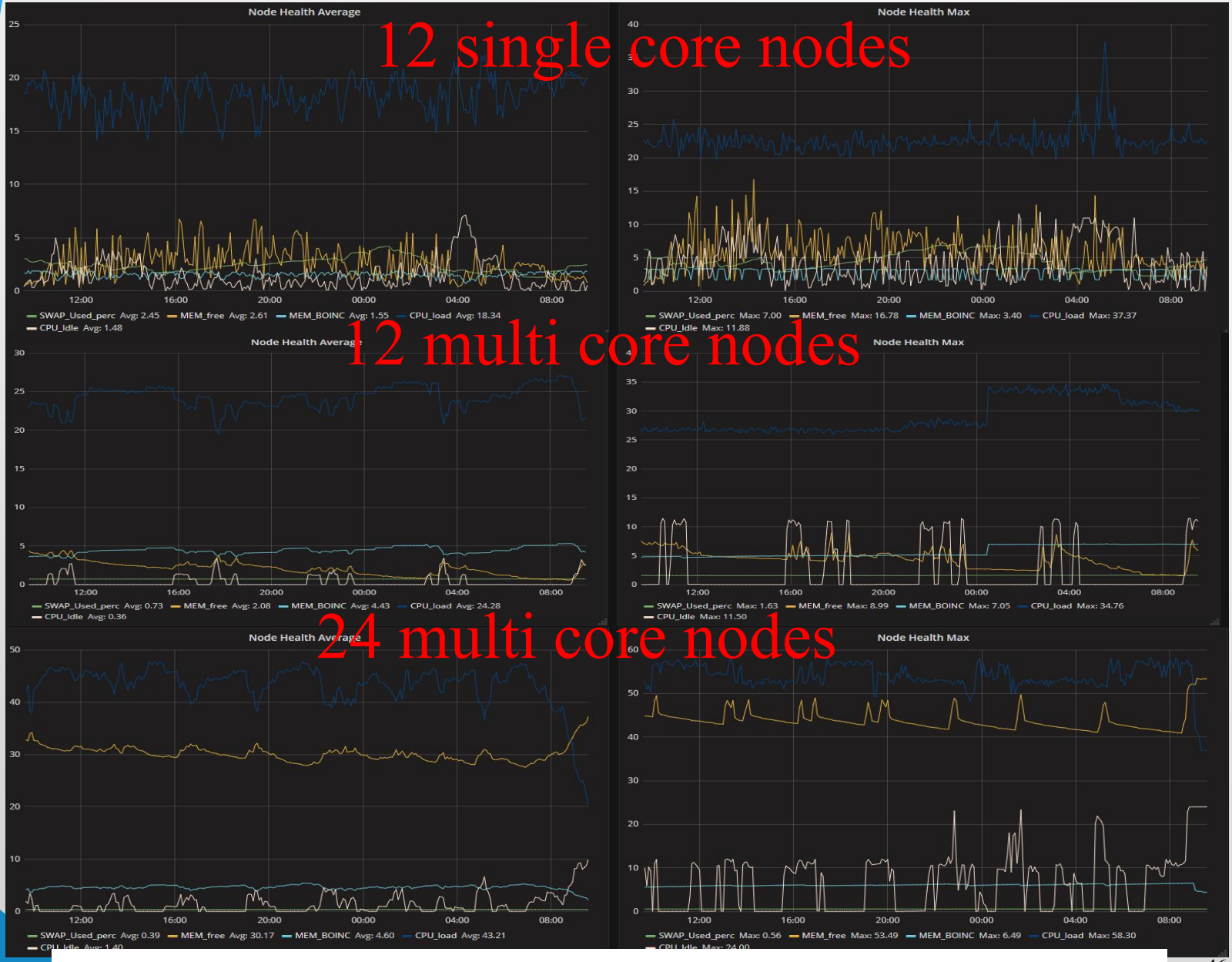
When production is over 94% in a single day





Production wall utilization is over 94% (avg. 415 jobs running out of 420 jobs slots)

ATLAS@home still can exploit **12.62%** CPU time



Monitor: Load, BOINC_used_memory, used_swap, idle_CPU