Backfilling the Grid with Containerized BOINC in the ATLAS computing

Wenjing Wu[1], David Cameron[2] , Andrej Filipcic[3]

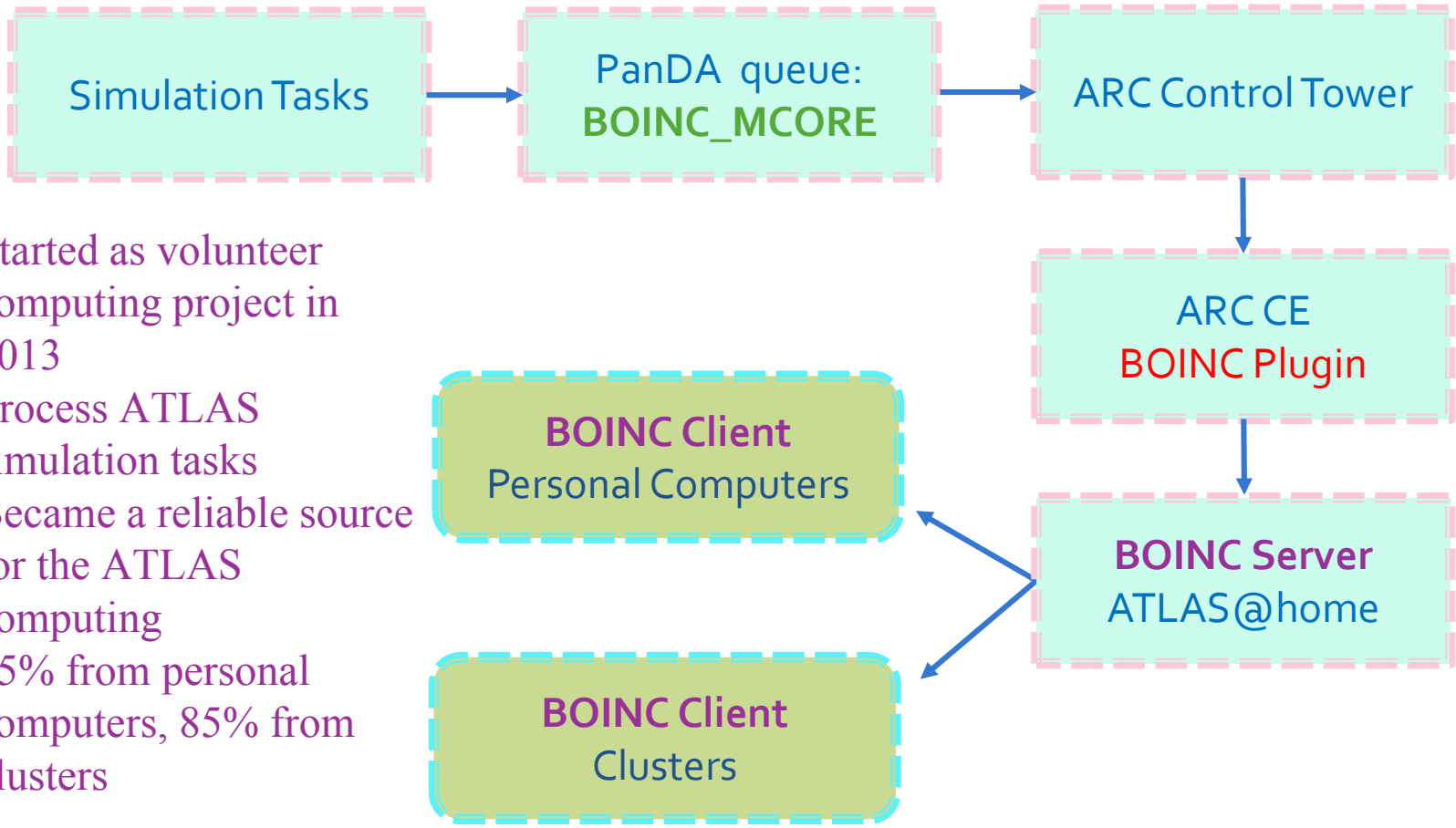
1. Computer Center, IHEP, China
2. University of Oslo, Norway
3. Jozef Stefan Institute, Slovenia

2018-07-10

Outline

- ATLAS@home : Virtualization vs. Containerization
- Performance measurement
- Use cases
  - Backfilling the grid sites
  - CERN IT cloud cluster
- Summary

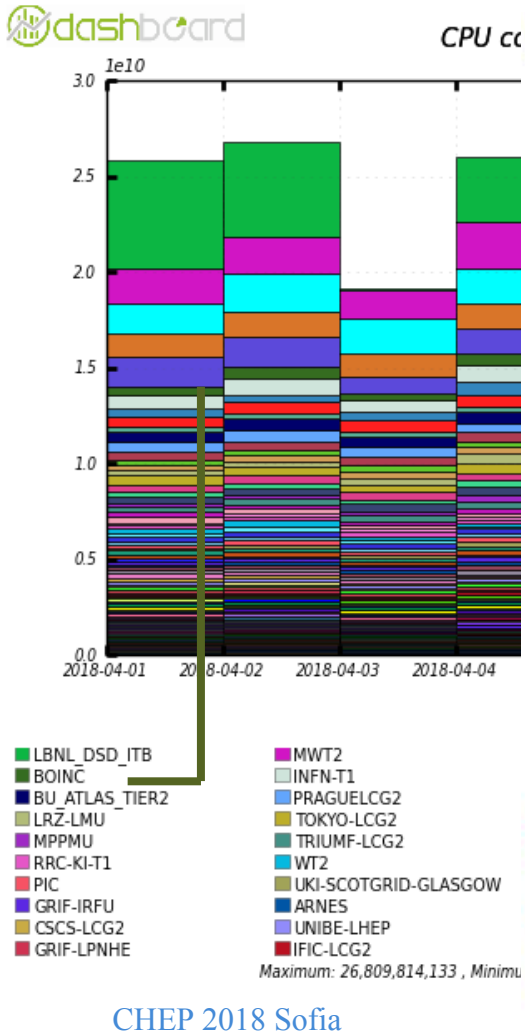
ATLAS@home Architecture

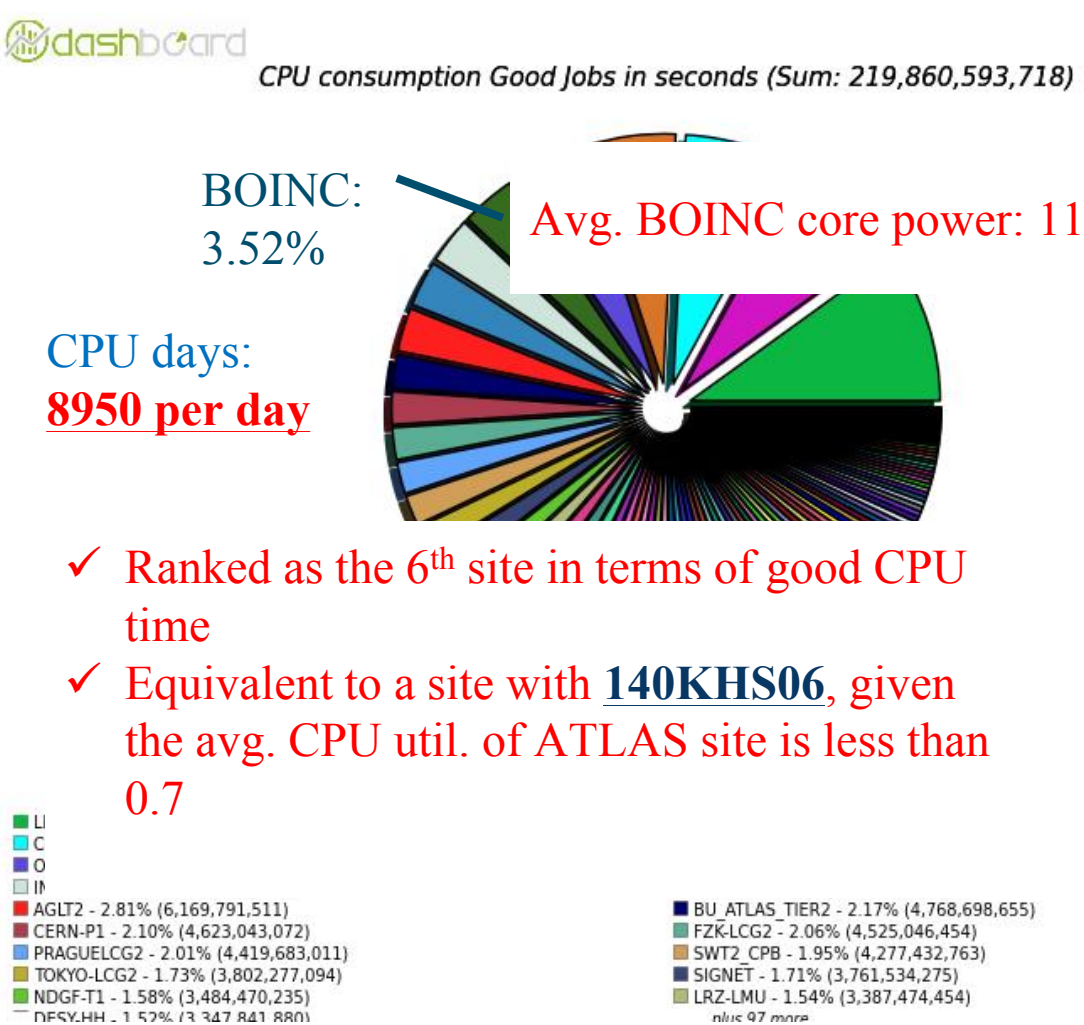


- Started as volunteer computing project in 2013
- Process ATLAS simulation tasks
- Became a reliable source for the ATLAS computing
- 15% from personal computers, 85% from clusters

Current scale (1)

CPU time of good jobs of All ATLAS sites in the past 10 days





BOINC: 3.52%

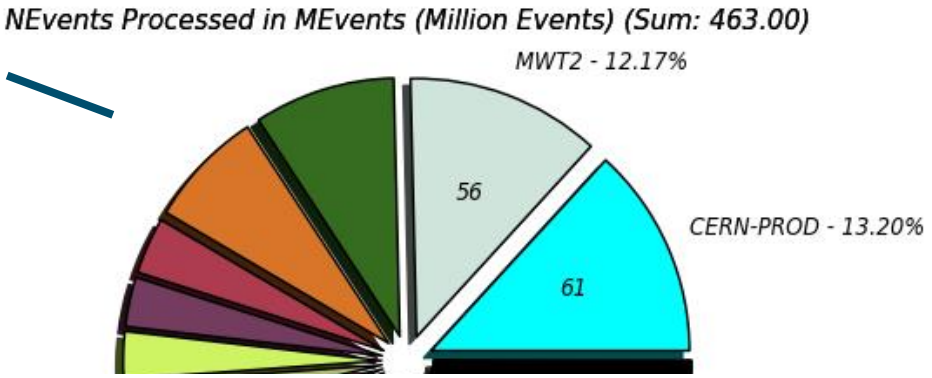
Avg. BOINC core power: 11HS06

CPU days: **8950 per day**

- ✓ Ranked as the 6th site in terms of good CPU time
- ✓ Equivalent to a site with **140KHS06**, given the avg. CPU util. of ATLAS site is less than 0.7

ATLAS@home as a simulation site

BOINC: **8.83%**
Avg. **4.1M** events per day



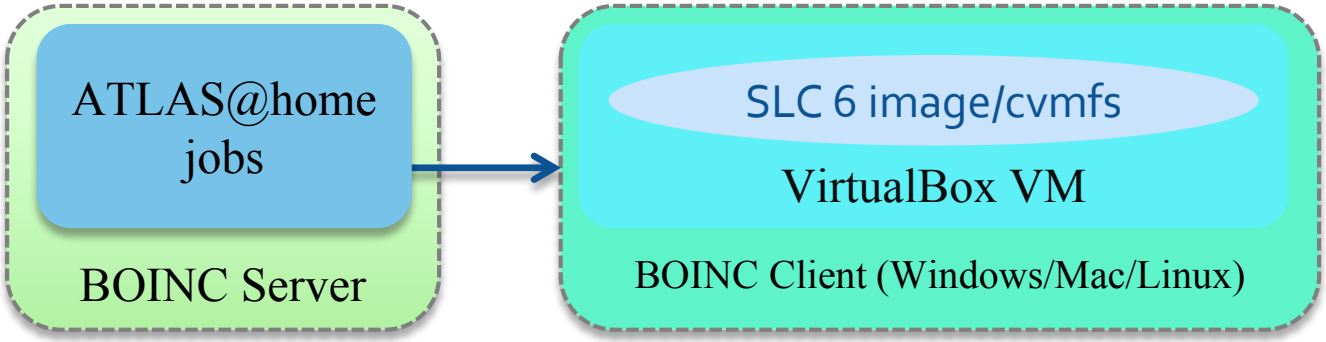
- ✓ Runs only simulation jobs, CPU intensive
- ✓ Ranked as the 2nd site for simulation jobs
- ✓ Over 50% of the ATLAS computing resources are used by simulation

CERN-PROD - 13.20% (61.00)
BOINC - 8.83% (41.00)
CERN-P1 - 3.54% (16.00)
LBNL_DSD_ITB - 2.96% (14.00)
TOKYO-LCG2 - 2.50% (12.00)
INFN-T1 - 1.95% (9.00)
SIGNET - 1.52% (7.00)
PRAGUELCG2 - 1.41% (7.00)
CA-VICTORIA-WESTGRID-T2 - 1.31% (6.00)
LRZ-LMU - 1.24% (6.00)

MWT2 - 12.17% (56.00)
BNL-ATLAS - 7.35% (34.00)
IN2P3-CC - 3.09% (14.00)
OLCF - 2.51% (12.00)
MPPMU - 2.09% (10.00)
AGLT2 - 1.72% (8.00)
BU_ATLAS_TIER2 - 1.46% (7.00)
RO-07-NIPNE - 1.34% (6.00)
TECHNION-HEP - 1.31% (6.00)
plus 83 more

Virtualization

- The hosts are heterogeneous in terms of OS, including different versions of Windows, Mac, Linux
- Virtualization was used upon all hosts to provide a unified software environment ATLAS simulation jobs require.
- Workflow
  - BOINC Client caches the image (Only once unless the image has updates)
  - BOINC client clones the image and launches VirtualBox VM for each job
  - Each ATLAS job is executed inside the VM



Disadvantages of Virtualization

- Creating the VM image is tedious
- Requires updates on the image on the BOINC server and client sides for new release of software
- Requires downloading/caching the VM image (~500MB after compression) on the BOINC client
- Over-heading time (for every new job) :
  - Clone the image (1.5GB, takes about 1 minute) before starting BOINC job
  - creation and termination of VM (5-10 minutes) before starting ATLAS job
  - can significantly reduce the CPU Efficiency of the short wall time jobs, which is common for multi-core simulation jobs
- Performance penalty (IO, CPU)
- Most of the software is cached in the image, but extra files(conditional database) might need to be downloaded on the fly, and they can't be cached and shared by different jobs on the client

Containerization

- As a solution for containerization, Singularity is widely used in the HEP field, especially in ATLAS computing
  - Standard images (CentOS 6/7) for ATLAS computing stored in CVMFS
  - A few sites (with non-standard Linux OS) use Singularity to run grid jobs on their clusters
- However Singularity “only” works on Linux systems, so keep the virtualization for Windows and Mac machines

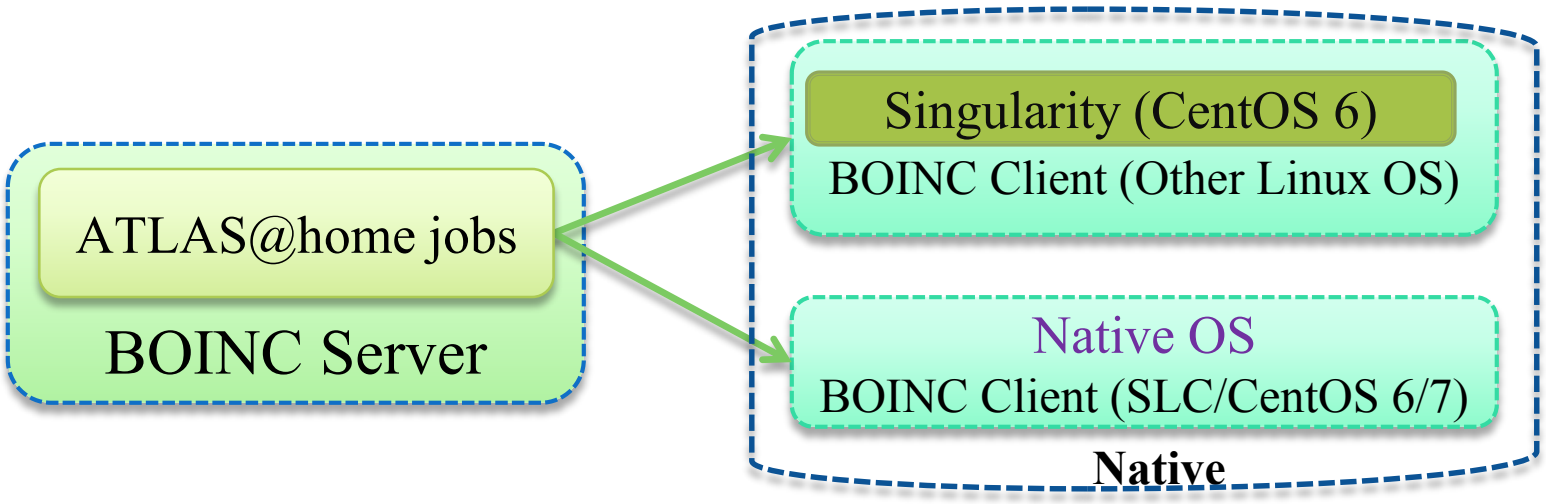
Local Performance Test

- Performance test
  - Host: RAM=8GB; Native OS : CentOS 7; Singularity Image: CentOS 7
- Test tools:
  - Read: dd if=/dev/zero of=/tmp/test.out bs=1M count=40000
  - Write: dd if=/tmp/test.out of=/dev/zero bs=1M
  - CPU: time echo "scale=5000;4*a(1)"|bc -l -q
  - ATLAS jobs: Simulation jobs with 3 events
- There is no Performance Loss between Singularity and native host

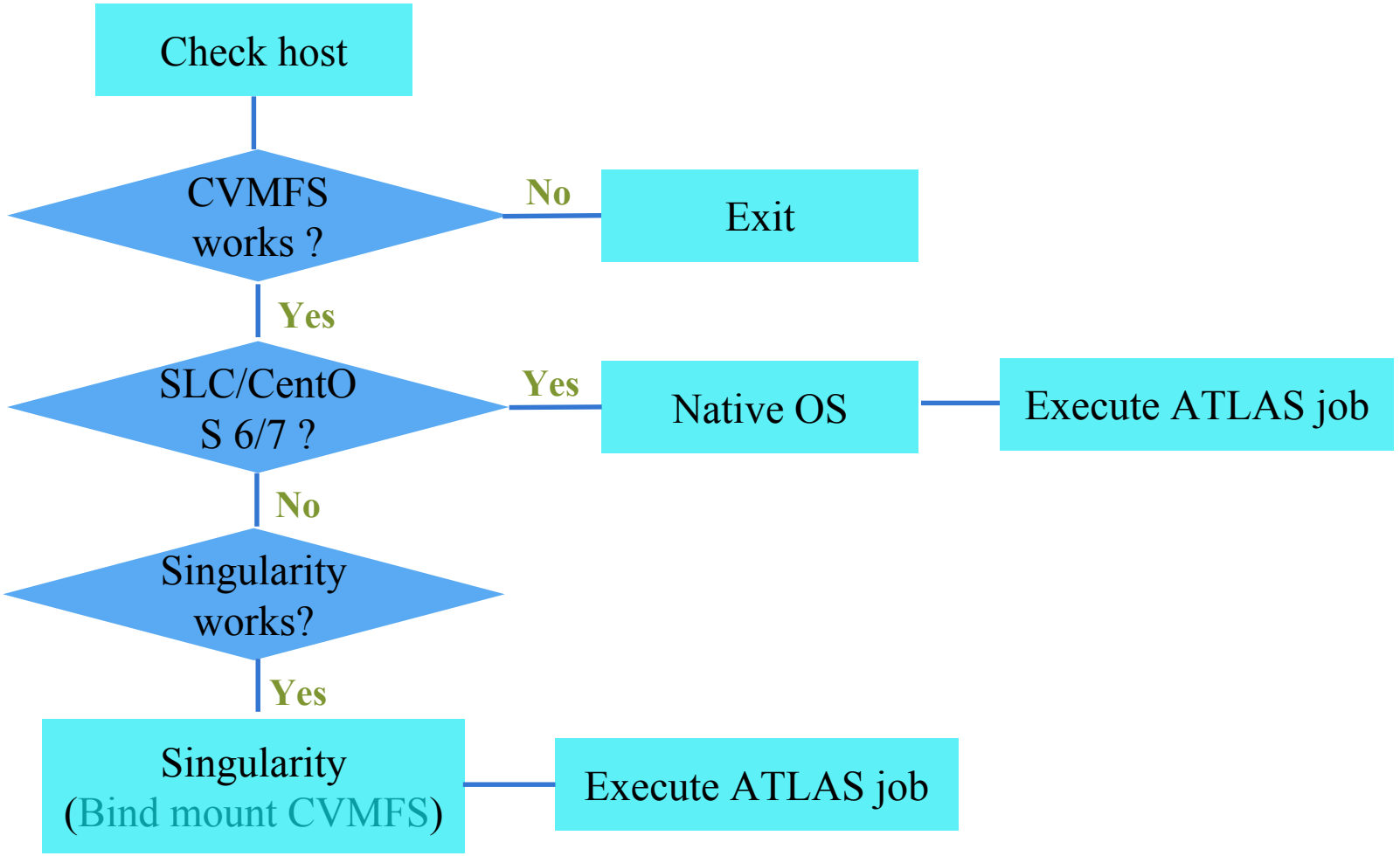
	READ	WRITE	CPU	ATLAS job
Singularity (CentOS 7)	113 MB/s	116 MB/s	26.975s	587s
Native Host(CentOS 7)	113 MB/s	114 MB/s	27.742s	594s
Performance Loss	0	-1.7%	-2.7%	1.17%

Containerized ATLAS@home

- The targets are various Linux machines
- Two different approaches of OS provision depending on the target host OS
- Its workflow is controlled by the ATLAS@home native job wrapper



Workflow: ATLAS@home native wrapper



Advantages of Containerization

- No need to maintain the VM image on the server side
- Significantly saves downloading and disk space on the client side
  - Software is cached in CVMFS of the target OS, can be shared by jobs
  - No need to clone and store a image (~1.5GB) for each job
- Eliminates the over-heading time for each job
  - Clone image (~1 min)
  - Creation/destroy of the VM (5~10 min)
- No performance penalty on IO/CPU

Performance Comparison (1)

- Test jobs are of the same size (total cputime required), the more cores per job, the shorter the walltime (on all cores).
- ATLAS Job CPU Eff.
  - CPU_Eff = cputime / walltime
  - walltime includes IO time (improved), cputime (improved), extra software/database files downloading time (eliminated)
  - CPU_Eff. is improved by **4~10%**

ATLAS Job CPU Eff. VM vs. Native			
core per job	CPU Eff. (VM)	CPU Eff. (Native)	CPU Eff. Offset
2	86%	91%	5%
4	72%	82%	10%
8	71%	75%	4%

Performance Comparison (2)

- BOINC Job CPU Eff.
  - BOINC_CPU_Eff = BOINC_cputime / BOINC_walltime
  - BOINC_walltime includes includes IO time (improved), cputime (improved), extra software/database files downloading time (eliminated) and VM over-heading time (eliminated)
  - BOINC_CPU_Eff is improved by **1~12%**

BOINC Job CPU Eff. (VM vs. Native)			
core per job	CPU Eff. (VM)	CPU Eff. (Native)	CPU Eff. Offset
2	90.4%	91.6%	1.2%
4	75.1%	81.3%	6.2%
8	66.3%	78.2%	11.9%

Use cases of containerized ATLAS@home

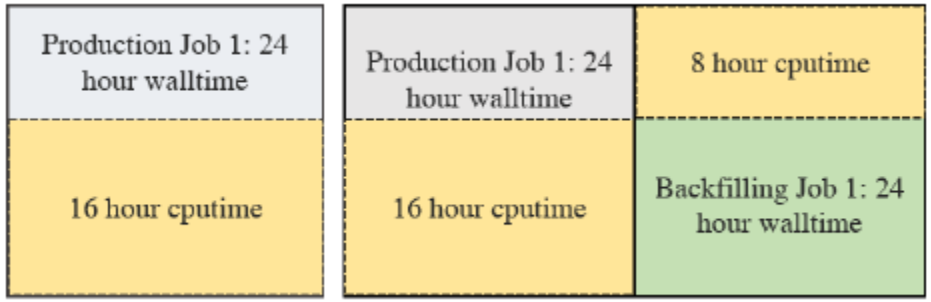
- Provides a lightweight way to run ATLAS@home on clusters (both physical nodes and cloud nodes)
  - Backfilling the busy ATLAS grid sites
  - Fulfill the idle cloud nodes from CERN IT
- It is too tedious and inefficient to run ATLAS@home vm version on the cluster with physical nodes (requiring installation of VirtualBox)
- It is impossible to run ATLAS@home vm version on the cloud nodes

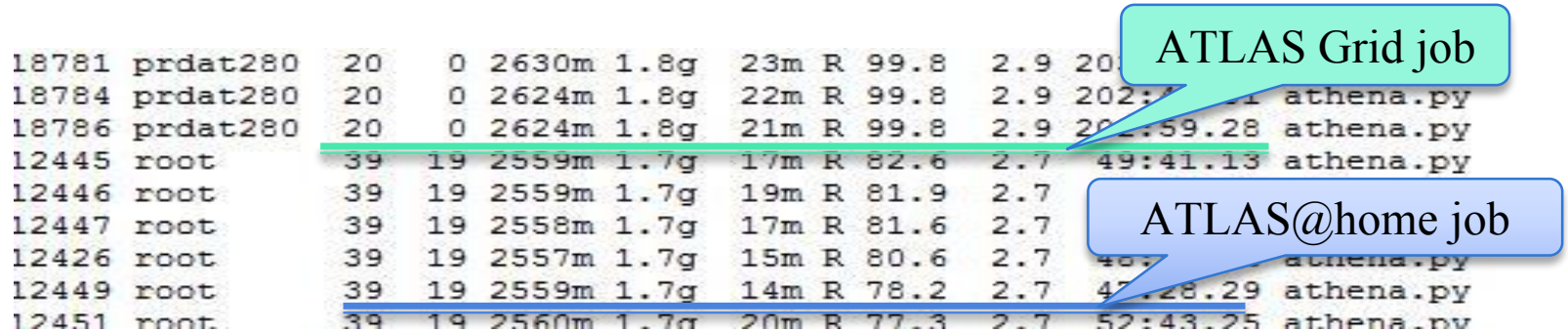
Backfilling the grid sites

Site	Core number	Avg. wall_util	Avg. cpu_util	Avg. cpu_eff
BEIJING	634	68%	55%	81%
TOKYO	6144	85%	72%	85%
SiGNET	5288	88%	68%	77%
MWT2	16250	83%	70%	84%
AGLT2	10224	72%	61%	84%

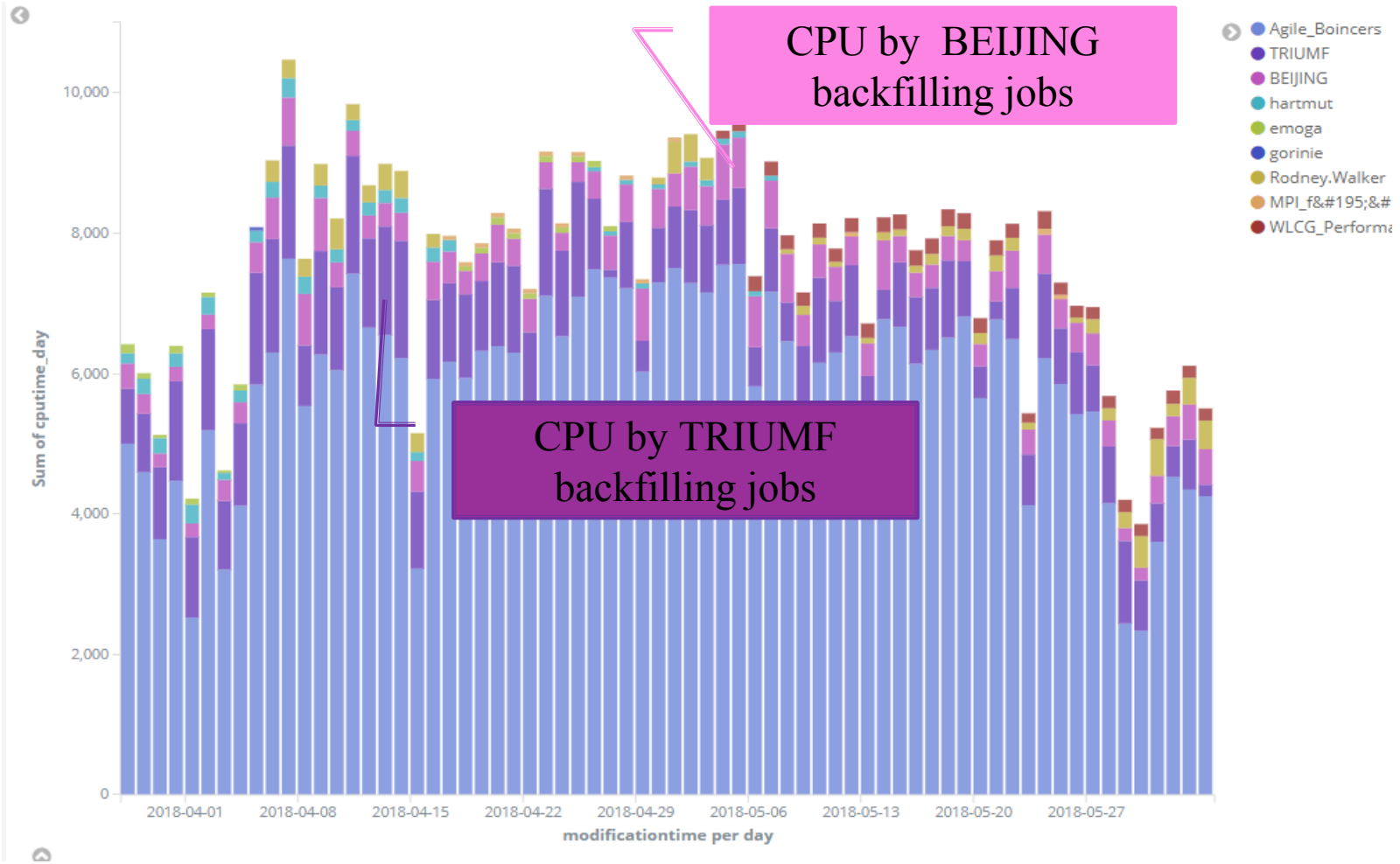
Table 1: The average utilization of typical ATLAS grid sites in a period of 100 days

- The Avg. CPU utilization of ATLAS grid sites is below 70%
- Backfilling is to run more than one process on each core
- Backfilling processes have lower priority than grid jobs, hence they only use the fragmental CPU cycles which are not being used by the grid jobs
- Experimented in 2 ATLAS sites:
  - BEIJING Tier 2 (464 cores)
  - TRIUMF Tier 1 (4688 cores)





CPU from backfilling jobs



- Avg. CPU time (CPU days/day) : TRIUMF 860 , BEIJING 230
- Accounting for 25% of the ATLAS@home computing power

Exploitation of extra CPU in backfilling

TRIUMF Site CPU Utilization(4816 CPU cores in 7 days)

Mon Mar 12 00:00:00 2018 to Mon Mar 19 00:00:00 2018

	suc_rate	cpu_eff	cpu_util	wall_util
BOINC	NaN	NaN	0.00	0.00
Grid	0.9	0.8	0.69	0.88
All	0.9	0.8	0.69	0.88

Before Backfilling

TRIUMF Site CPU Utilization(4816 CPU cores in 7 days)

Thu Apr 12 00:00:00 2018 to Thu Apr 19 00:00:00 2018

	suc_rate	cpu_eff	cpu_util	wall_util
BOINC	0.97	0.29	0.27	0.91
Grid	0.95	0.50	0.65	0.97
All	0.95	0.50	0.92	1.88

After Backfilling

BEIJING Site CPU Utilization(464 CPU cores in 7 days)

Thu Apr 12 00:00:00 2018 to Thu Apr 19 00:00:00 2018

	suc_rate	cpu_eff	cpu_util	wall_util
BOINC	1.00	0.17	0.15	0.88
Grid	0.99	0.53	0.80	0.93
All	0.99	0.53	0.95	1.81

Busy Week

BEIJING Site CPU Utilization(464 CPU cores in 7 days)

Fri Apr 6 00:00:00 2018 to Fri Apr 13 00:00:00 2018

	suc_rate	cpu_eff	cpu_util	wall_util
BOINC	1.00	0.47	0.42	0.88
Grid	0.96	0.61	0.48	0.62
All	0.98	0.61	0.90	1.50

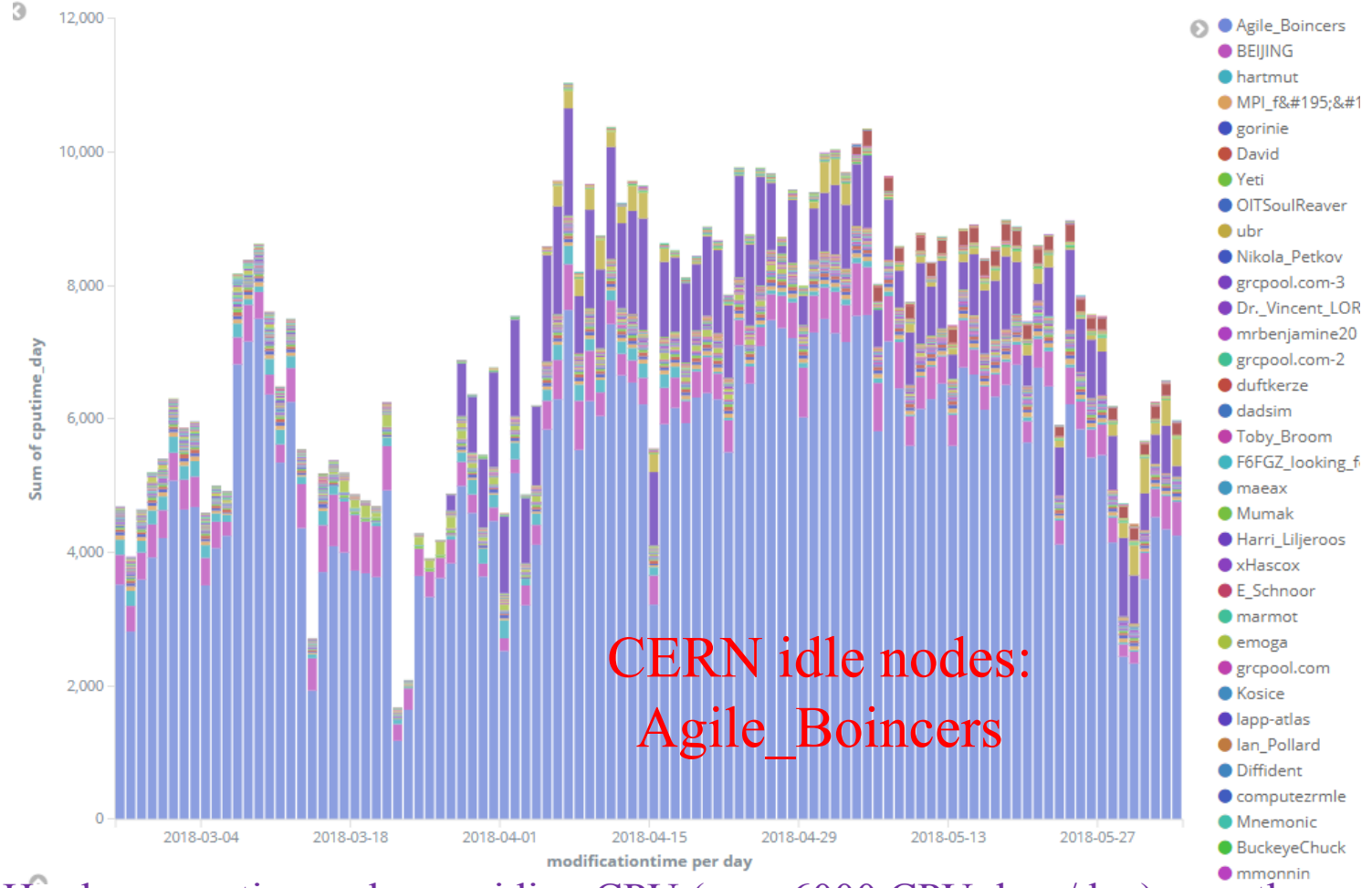
Idle Week

- The CPU utilization of TRIUMF site is **improved by 23%** (from 69% to 92%, note the grid workload is even higher after backfilling)
- BEIJING site, the Avg. CPU utilization improvement (in 6 months period) is **26%**
- The Peak CPU utilization reaches **95%** in a week basis, remains **90%** in long term

Cloud nodes from CERN IT

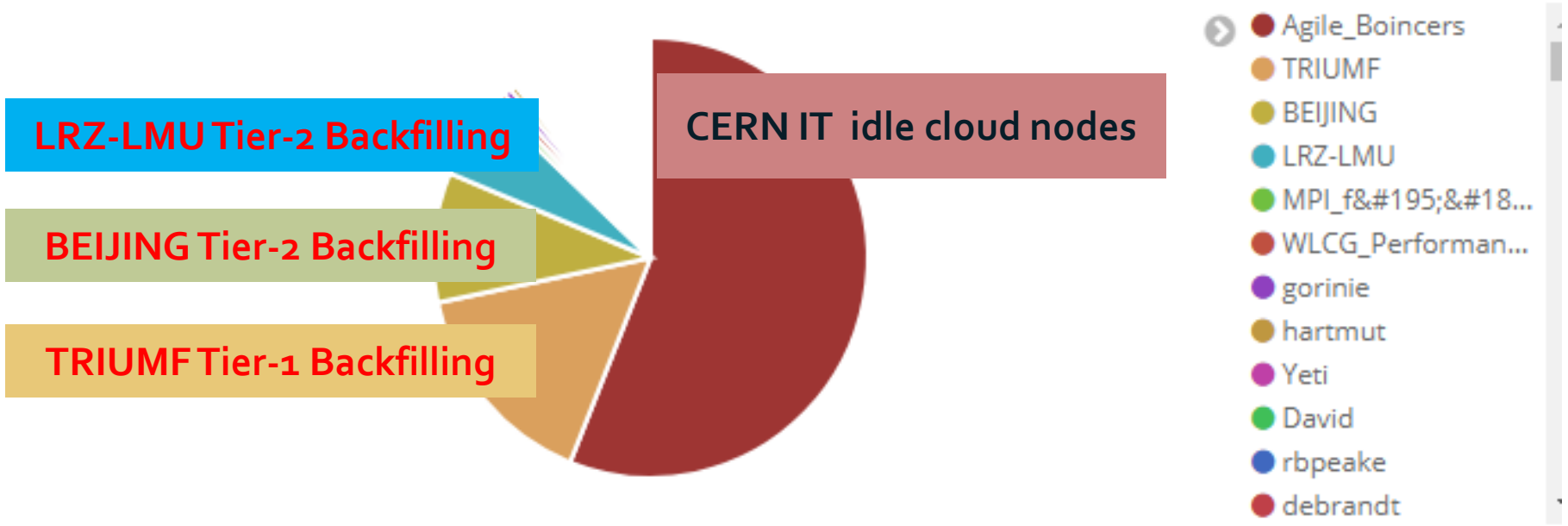
- There are a very considerable amount of Idle computers from CERN IT (not being used for anything)
  - Physical Nodes near retiring
  - Cloud nodes before being delivered to the users/experiments
- Cloud nodes also need to run CPU intensive benchmarks, the ATLAS@home jobs fit into that requirement
- BOINC client is in puppet, can easily be deployed to nodes

CPU time delivered by CERN idle nodes



- Has been continuously providing CPU (avg. 6000 CPU days/day) over the past 6 months, accounting for 60% of the CPU time for ATLAS@home
- Utilizes the nodes which can't be used by other experiments!

Containerized ATLAS@home becomes the majority resource



85% of the ATLAS@home CPU time is provided by the containerized version, and **25%** is from the grid sites backfilling

Summary

- Containerized ATLAS@home provides a lightweight solution compared to its VM version
  - Reduces the over-heading time, and improves the job CPU Efficiency by 5-10%
  - Reduces the usage of disk space, network from the clients, and the maintenance work from server
  - Makes it possible for two use cases (backfilling grid sites and fulfilling cloud nodes) which significantly increases the available resources for ATLAS@home
- Over 85% of the ATLAS@home computing power is from the containerized ATLAS@home

Thanks!