How Fair is my Fair-Sharing?

Exposing Some Hidden Behavior Through Workload Analysis

Fréderic Azevedo, Dalibor Klusáček, and Frédéric Suter

IN2P3 Computing Center / CNRS
Villeurbanne, France

HEPiX Spring Workshop

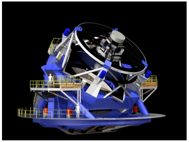
March 27, 2019

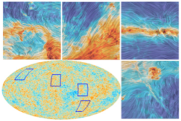
The IN2P3 Computing Center

- The French Tier-1 center for WLCG
  - also a Tier-2

⟸ 35,000 (virtual) cores
340PB of storage

80 scientific collaborations ⟹
2,500+ users





Computing Resources

Operated by Univa Grid Engine

Model	#Nodes	#vCores / Node	#vCores
Intel Xeon E5-2670 0 @ 2.60GHz	24	32	768
Intel Xeon Silver 4114 @ 2.20GHz	241	40	9,640
Intel Xeon E5-2680 v2 @ 2.80GHz	149	40	5,960
Intel Xeon E5-2680 v3 @ 2.50GHz	124	48	5,952
Intel Xeon E5-2650 v4 @ 2.20GHz	232	48	11,136
Total	**770**		**33,456**

- Parallel jobs: 512 cores without hyper-threading in 16 nodes
- GPU-based jobs: 40 K80 and 24 V100 GPUs
- Large memory jobs: 1 node with 40 cores and 1.5TB of memory

A Typical HEP Workload

- Dominated by the four LHC experiments
  - About 55% of the allocated resources
- A vast majority of sequential jobs
  - 85% sequential
    - Monte-Carlo simulations
  - 15% multi-core
    - Limited to a single node (8 or 16 cores)
    - Submitted by two groups (ATLAS and CMS)
    - Represent 50+% of the residency time
- Data-driven jobs
  - Heavily depend on storage subsystems
    - GPFS, HPSS, iRODS, dCache, XrootD

Scheduling Principle and Objectives

Principle

- Groups express pledges every year (as a computing power in HS06)
  - Well defined for LHC experiments, more approximative for small groups
- The sum of all pledges defines what CC-IN2P3 has to deliver
  - Condition the purchase of new hardware
- Each group gets a proportional share of this
  - Defines an consumption objective
  - Used by the job scheduler as a basis of its Fair-Share policy

Scheduling Principle and Objectives

Principle

- Groups express pledges every year (as a computing power in HS06)
  - Well defined for LHC experiments, more approximative for small groups
- The sum of all pledges defines what CC-IN2P3 has to deliver
  - Condition the purchase of new hardware
- Each group gets a proportional share of this
  - Defines an consumption objective
  - Used by the job scheduler as a basis of its Fair-Share policy

#1 Objective

- **Satisfy all the user group pledges**

#2 Objective

- **Maximize the utilization of the machines**

How Fair is the Fair-Share ?

At "Macro" scale

- The overall fairnees operational objective is respected
- Pledges are served
- From a 3-month to 1-year granularity

At "Micro" scale

- Operators act on scheduling
- Fix fair-share transient issues
  - Boost or block jobs/users/groups

How Fair is the Fair-Share ?

At "Macro" scale

- The overall fairnees operational objective is respected
- Pledges are served
- From a 3-month to 1-year granularity

At an intermediate scale

- What about fairness at 1-day, 1-week, or 1-month granularity?
- Is the Quality of Service the same for all our users?
  - Spoiler alert: Answer is NO!
- What can be done to improve fairness?

At "Micro" scale

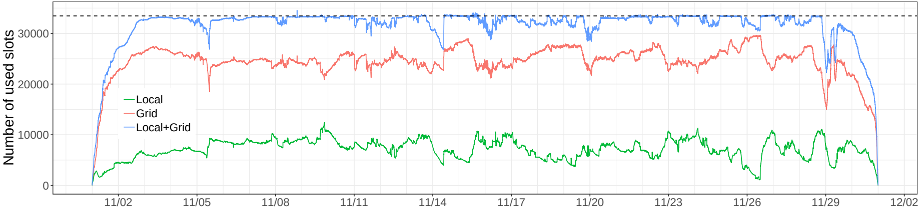
- Operators act on scheduling
- Fix fair-share transient issues
  - Boost or block jobs/users/groups

Outline

- Introduction
- Analysis of the Workload(s)
  - Overall Utilization
  - Grid vs. Local Jobs
  - Origins of the Unfairness
- Reconfiguration of the Batch System
  - Redefinition of the Scheduling Queues
  - Quota Relaxation
  - Simulation Results
- Conclusion and Future Work

Overall Utilization

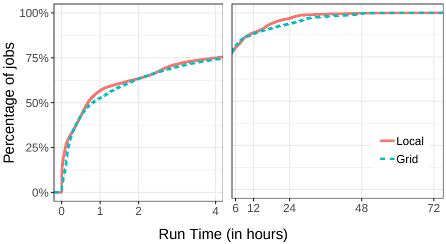
Maximize the utilization of the machines: DONE! (over 90%)



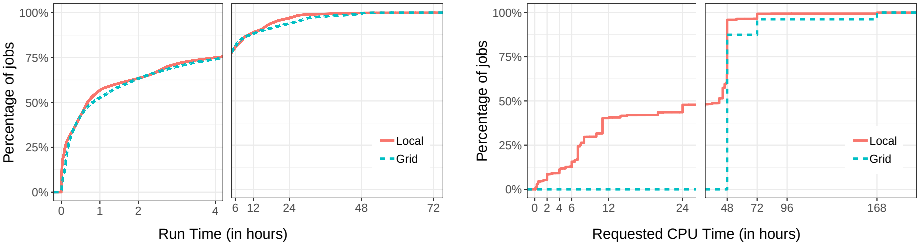
Grid vs. Local Jobs

- Grid
  - 1,495,323 jobs
  - 28% are multi-core (i.e., 8-core) jobs
  - Use 3.45 more resources
- Local
  - 1,174,078 jobs
  - 98% are sequential

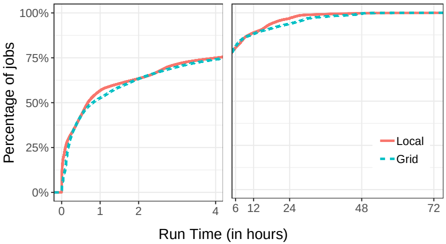
Grid vs. Local Jobs

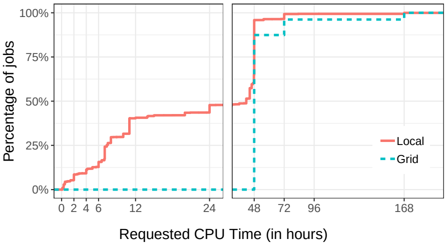


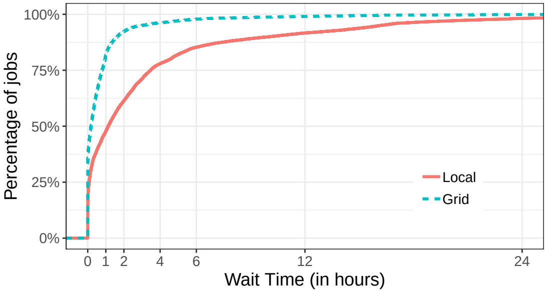
Grid vs. Local Jobs



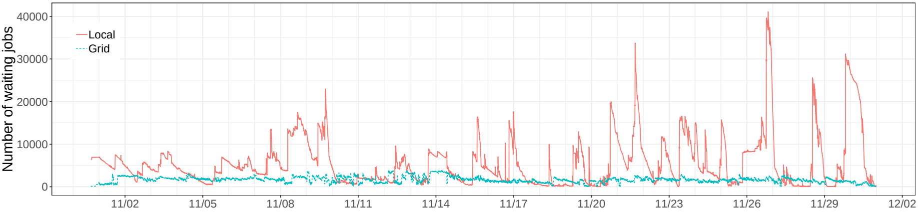
Grid vs. Local Jobs



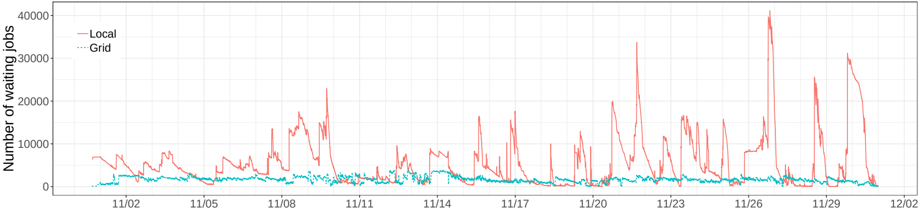


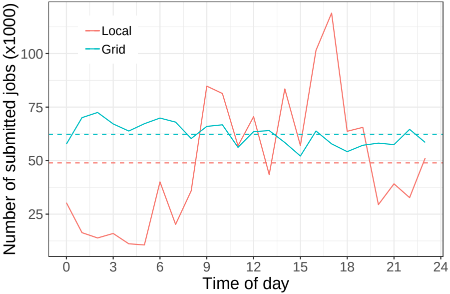


Origins of the Unfairness

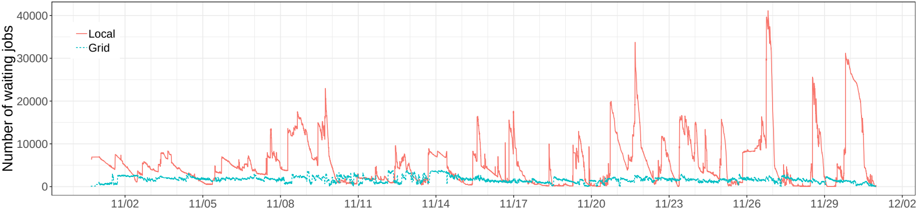


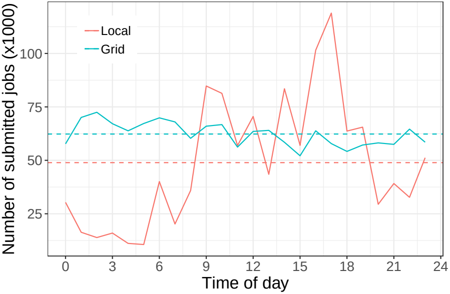
Origins of the Unfairness

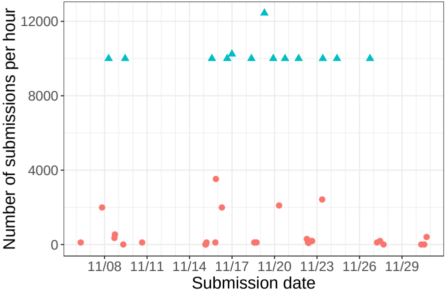




Origins of the Unfairness







- And also share-related priorities and stringent quotas

Outline

- Introduction
- Analysis of the Workload(s)
  - Overall Utilization
  - Grid vs. Local Jobs
  - Origins of the Unfairness
- Reconfiguration of the Batch System
  - Redefinition of the Scheduling Queues
  - Quota Relaxation
  - Simulation Results
- Conclusion and Future Work

Redefinition of the Scheduling Queues

Queue name	CPU Time	Time	Memory	File Size	Cores
mc-long	48h	58h	3.6G	30G	33,568
mc-huge	72h	86h	8G	30G	9,040
mc-longlasting	202h	226h	3G	30G	19,800
long	48h	58h	4G	30G	33,568
huge	72h	86h	10G	110G	10,418
longlasting	168h	192h	4G	30G	3,931

- Sequential vs. Multi-core
  - But Multi-core = Grid $\rightsquigarrow$ even higher priority
- Walltime not considered at all
- No "Resource pools"

Redefinition of the Scheduling Queues

Queue name	CPU Time	Time	Memory	File Size	Cores
mc-long	48h	58h	3.6G	30G	33,568
mc-huge	72h	86h	8G	30G	9,040
mc-longlasting	202h	226h	3G	30G	19,800
long	48h	58h	4G	30G	33,568
huge	72h	86h	10G	110G	10,418
longlasting	168h	192h	4G	30G	3,931

- Sequential vs. Multi-core
  - But Multi-core = Grid $\rightsquigarrow$ even higher priority
- Walltime not considered at all
- No "Resource pools"

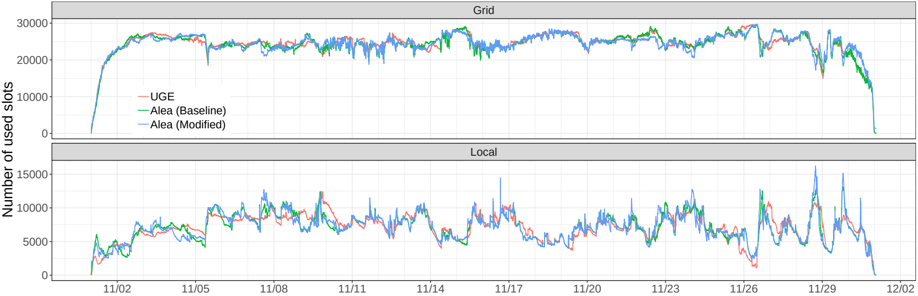
Queue name	CPU Time	Time	Memory	File Size	Cores
local-short	6h	7h	4G	30G	20,000
local-medium	24h	28h	4G	30G	15,000
local-long	48h	58h	4G	30G	10,000
grid	48h	58h	3.6G	30G	25,000
huge	72h	86h	10G	110G	10,000
longlasting	202h	226h	3G	30G	5,000

Quota Relaxation

- Existing large quota $\rightsquigarrow$ Harmless jobs
- Classify local jobs according to the fraction of resources they can use
  - 0-5%
  - 5-10%
  - 10+%
- Conservative relaxation
  - 0-5% $\rightsquigarrow$ increase by 5%
  - 5-10% $\rightsquigarrow$ increase by 10%
  - 10+% $\rightsquigarrow$ increase by 20%
- Extreme relaxation
  - a.k.a. make your storage admin crazy
  - 0-5% $\rightsquigarrow$ increase by 100%
  - 5-10% $\rightsquigarrow$ increase by 200%
  - 10+% $\rightsquigarrow$ increase by 300%

Simulation Results

- Replay the entire workload in simulation
- Rely on the Alea job scheduling simulator
  - Models the algorithms, queues, quotas, ...
- Have to first check that the simulation captures the main trends of the original schedule $\rightsquigarrow$ Baseline version



Simulation Results

Redefinition of the scheduling queues

			Percentiles			
Workload	**Scenario**	**Average**	**50th**	**75th**	**90th**	**Maximum**
Grid	Baseline	1h 10m	0s	8m 18s	1h 18m	15d 21h 54m
	Modified	1h 45m	0s	14m	2h 2m	14d 4h 33m
Local	Baseline	2h 3m	4m 30s	1h 40m	6h 40m	11d 21h 41m
	Modified	1h 58m	8s	1h 10m	6h 20m	4d 19h 6m

Simulation Results

Redefinition of the scheduling queues

Workload	Scenario	Average	Percentiles 50^{th}	Percentiles 75^{th}	Percentiles 90^{th}	Maximum
Grid	Baseline	1h 10m	0s	8m 18s	1h 18m	15d 21h 54m
	Modified	1h 45m	0s	14m	2h 2m	14d 4h 33m
Local	Baseline	2h 3m	4m 30s	1h 40m	6h 40m	11d 21h 41m
	Modified	1h 58m	8s	1h 10m	6h 20m	4d 19h 6m

Quota Relaxation

Workload	Scenario	Average	Percentiles 50^{th}	Percentiles 75^{th}	Percentiles 90^{th}	Maximum
Grid	Conservative	1h 53m	0s	16m	2h 21m	13d 15h 21m
	Extreme	1h 57m	4s	17m 41s	2h 47m	14d 4h 41m
Local	Conservative	1h 39m	2s	45m 40s	5h 8m	3d 16h 58m
	Extreme	1h 14m	1s	21m 55s	2h 30m	3d 23h 11m

Outline

- Introduction
- Analysis of the Workload(s)
  - Overall Utilization
  - Grid vs. Local Jobs
  - Origins of the Unfairness
- Reconfiguration of the Batch System
  - Redefinition of the Scheduling Queues
  - Quota Relaxation
  - Simulation Results
- Conclusion and Future Work

Conclusion

- Batch systems are complex
  - Many configuration parameters
- Have to know understand your workload
- Study different options
  - Redefine queues
  - Leverage job duration
  - Relax quotas
- Leverage simulation to assess the impact of modifications
  - It's a production system, disruption is forbidden

Future Work

- LHC grid jobs are always there
  - 5,000 slots for ATLAS
  - 4,000 slots for CMS
  - 2,500 slots for LHCb

$\leadsto$ More than 30% of the resources

- What-if these jobs where submitted to a HTCondor pool instead of UGE?
  - Handle HTC with a HTC tool
  - Leave the batch to batch users
  - Two systems to manage
  - Can we still guarantee a 90+% utilization?