University of Basel

DMI HPC

Unlocking the Potential of HPC Systems Use for Optimizing Job Queues, Minimizing Wait Times, and Boosting Efficiency

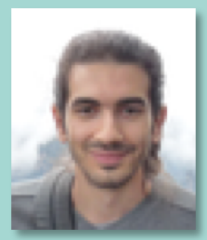
Thomas Jakobsche, Dr. Aurélien Cavelan, Dr. Rubén Cabezón, Dr. Iñaki Martínez de Ilarduya, Prof. Dr. Florina M. Ciorba

October 5, 2023 – hpc-ch forum on HPC and Data as a Service

hpc.dmi.unibas.ch | dmi-hpc@unibas.ch | @HPC_DMI_UniBas | @hpc_dmi_unibas

since 08/2015

Thomas Jakobsche



Aurélien Cavelan

Rubén Cabezón

Iñaki Martínez de Ilarduya

Florina M. Ciorba

Outline

Motivation

Problem Statement

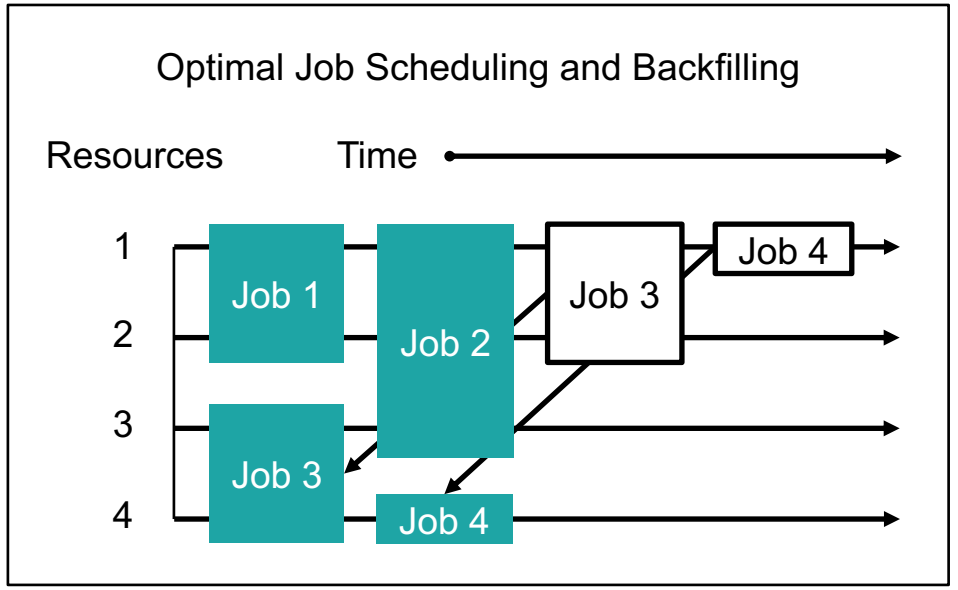
- Unnecessary job wait time caused by inaccurate resource requests and low job efficiency.

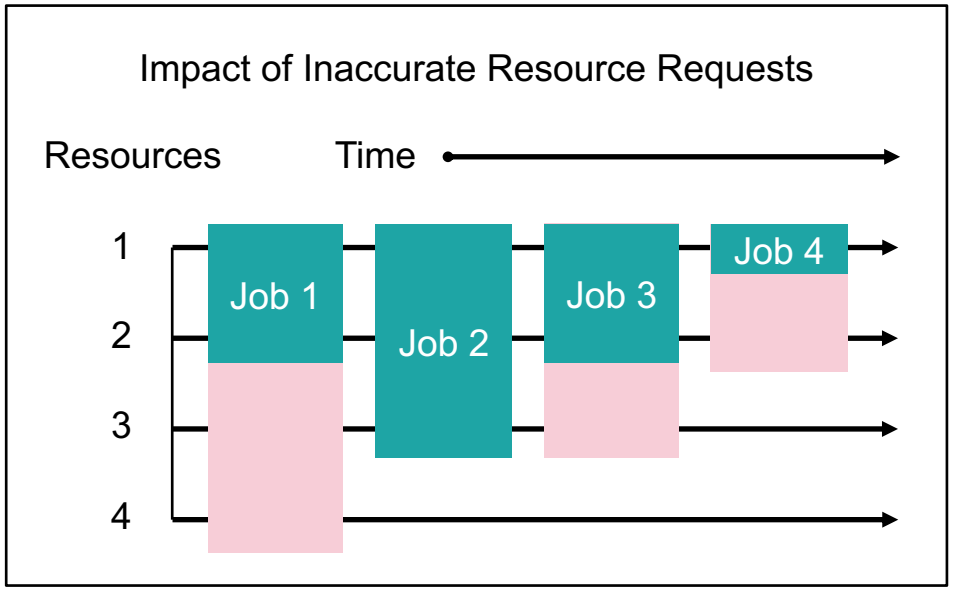
Challenge

- A job's wait time is not always avoidable even on fully utilized systems – sometimes resources are busy.

Analysis Goal

- Understand and avoid "unnecessary" job wait time caused by inaccurate resource requests.
- Improve system utilization and efficiency.





Dataset and Job Queues

Timespan	Jan – Aug 2023
#Accounts	~150
#Users	~500
#Jobs	~55 million
CPU Hours	~20 million
GPU Hours	~100 thousand

Queues	%Jobs	%CPUh	%Backfill	%Wait Time
30min	75.8%	12.3%	77.2%	34.1%
6hours	22.0%	49.3%	52.1%	43.6%
1day	1.9%	20.9%	30.4%	21.1%
1week	0.1%	12.6%	27.9%	1.0%
2weeks	>0.1%	0.8%	36.0%	>0.1%

Shorter queues offer access to more resources.

Outline

Relation between Job Resource Requests, Wait Time, and Backfilling

- We investigate positive and negative correlations for the top 10 users by wait time.

➢ More requested resources correlate with higher wait times and less backfilling.

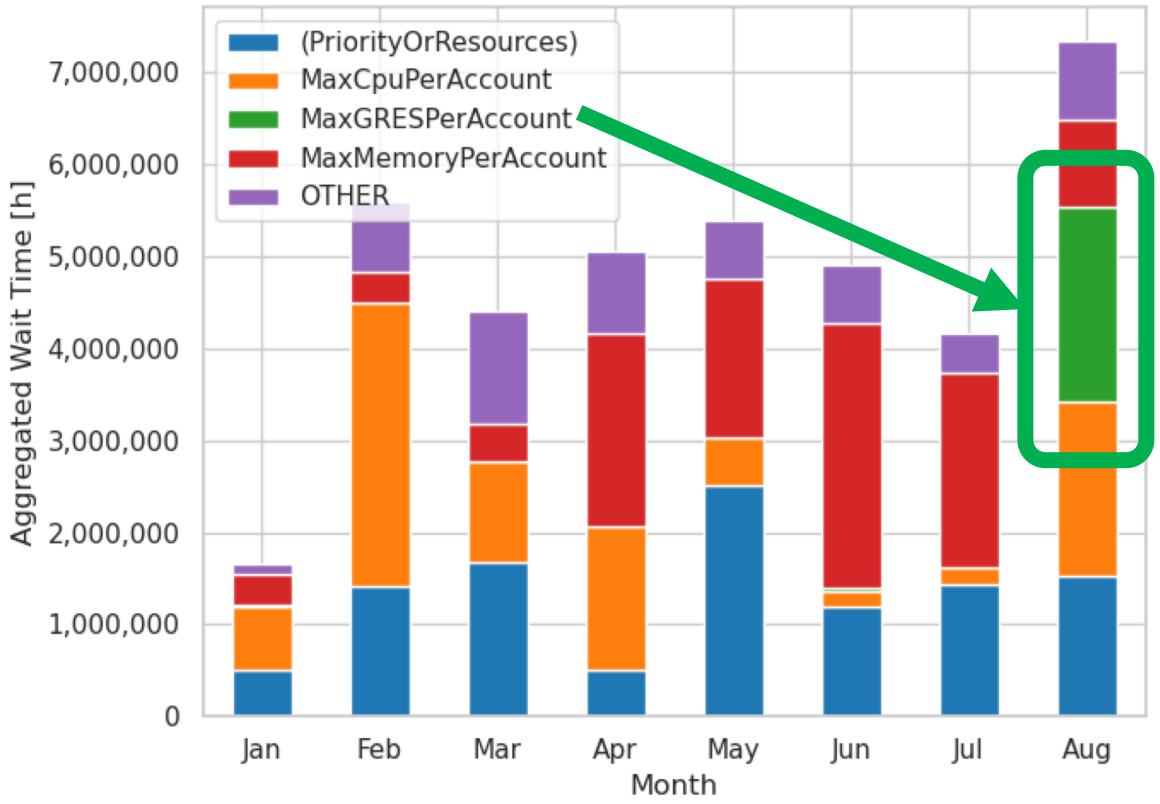
➢ Requested memory plays a higher role than requested CPUs, even higher than Timelimit.

Pearson Linear Correlation Coefficient (-1 to 1)		
	Avg. Wait Time	Backfilling
Avg. Timelimit	0.58	-0.69
Avg. ReqCPUS	0.45	-0.63
Avg. ReqMem	0.70	-0.87

Wait Reasons per Month

- Slurm retains a wait reason per job other than Priority or Resources, in that case it stores "*None*".
- MaxCpuPerAccount and MaxMemoryPerAccount are limits per account and per queue.
- MaxGRESPerAccount is a GPU limit also per account and per queue.

➢ GPU limit wait time happened mostly in August 2023!



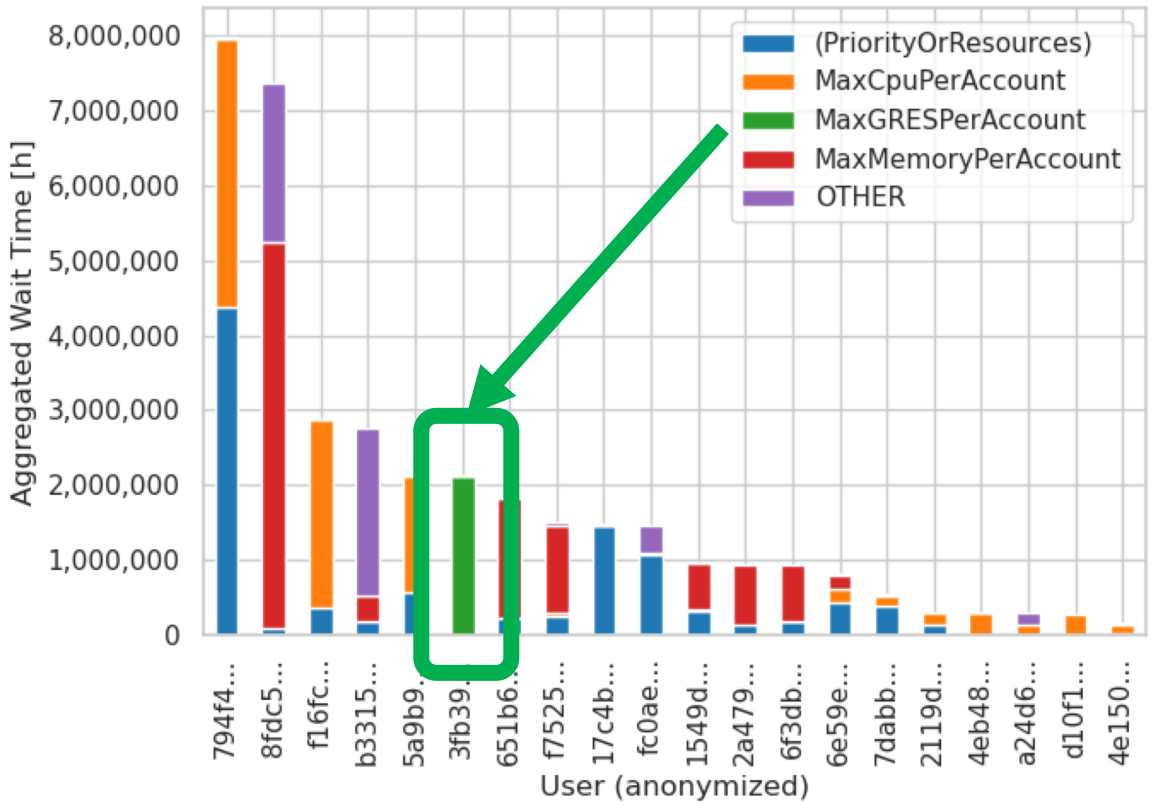
Wait Reasons by User

- The top ten users aggregate
  ~80% of the total wait time.

- Users wait either due to CPU limits
  or Memory limits, but not both.

The GPU wait case becomes more interesting:

- now we know it only affects one of the top users.

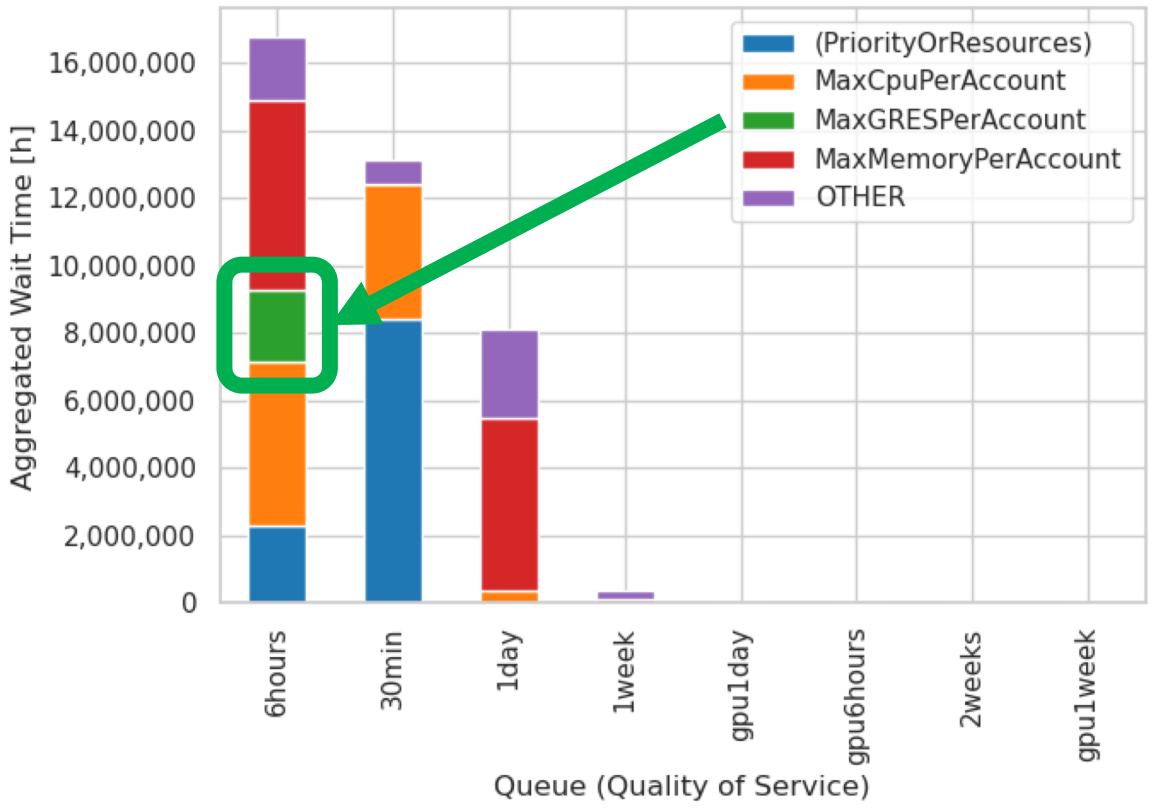


Wait Reasons per Queue

- 6hours queue experiences most of the wait time.
- 30min queue has almost no Memory limit wait time.
- 1day queue has almost no CPU limit wait time.

Conclusion to the GPU limit wait case:

- Jobs were submitted to the “normal” 6hours queue (limited to 16 GPUs) instead of the dedicated gpu6hours queue (access to 40 GPUs).
- The GPU limit wait case is a good example of unnecessary wait time due to misconfiguration.



Outline

Potential to Improve Appropriate Queue Submissions

- We consider submission to a shorter queue possible if the **job execution time * 2 <= Timelimit** of the shorter queue (100% buffer/overestimation).

➢ The majority of the jobs from 6hours, 1day, 1week, and 2weeks queues could have been submitted to shorter queues.

➢ We know that submission to shorter queues is desirable, due to more access to resources, more backfilling and lower job wait times.

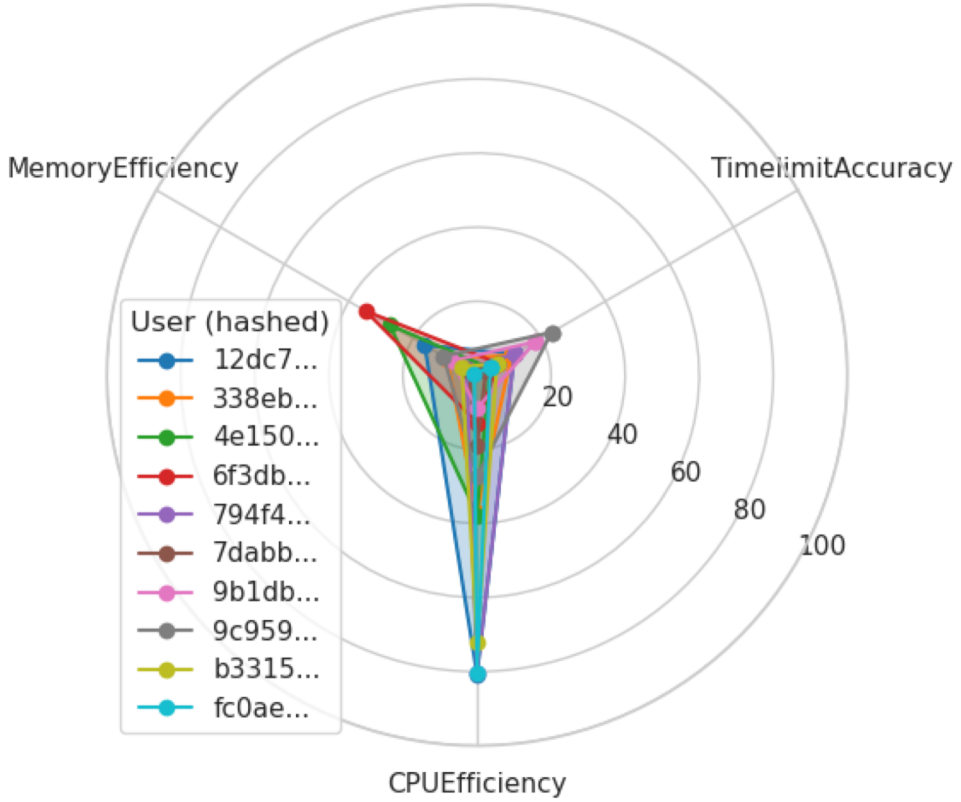
Submission Queue	Number of Jobs	% of Jobs where Submission to Shorter Queue was Possible
30min	39’820’630	already shortest queue
6hours	11’545’374	80.7%
1day	1’000’464	94.2%
1week	54’341	93.0%
2weeks	8’26	90.1%

Potential for Appropriate Resource Requests

- The plot shows the top 10 users by CPUh.

➢ We observe high potential for these users to improve TimelimitAccuracy and MemoryEfficiency.

➢ The problem with overestimating memory
  - the job waits longer itself
  - the job blocks memory for other jobs
  - this makes job scheduling difficult
  - ultimately degrades the service for every job



Outline

Appropriate Queue Recommendation

- Reporting TimelimitAccuracy for individual jobs may not be the best approach to support users (see tools: sacct, jobstats, reportseff, etc.).
- We identify working directories that are the origin of overestimated jobs.
- Users potentially benefit and adapt to more coarse grained queue recommendations.

WorkDir (hashed)	#Jobs	Max Exec. Time	Queue Chosen by User	Recommended Queue
e711e...	101’077	0:03:33	6hours	30min
b9239...	48’833	0:01:18	6hours	30min
3b843...	25’686	1:28:59	1day	6hours
...	...	...	...	...

Detecting Problem Cases that Aggregate Specific Wait Time

- It can be an indication of misconfiguration if all jobs from a specific working directory are waiting only for one reason.
- Indeed we see a return of the GPU limit wait case! 99% of the wait time of all jobs from this working directory are due to GPU limits. Which could be avoided by submitting to the appropriate GPU queue.
- This form of reason-based analysis can support the detection of misconfiguration and problem cases.

WorkDir (hashed)	#Jobs	Wait Time [h]	Top Reason for 99% of Wait Time
a48a2...	13’608	2’107’958	MaxGRESPerAccount
5bc88...	8’206	290’445	MaxCpuPerAccount
d29c0...	21’125	222’892	MaxMemoryPerAccount
...	...	...	...

Outline

Open Questions and Future Work

Open Questions

- What are the reasons for overestimation?
  a) Fear of timeouts?
  b) Misconfiguration/application crashes?
  c) “Mindlessness”?

- How to incentivize users to submit more accurate resource requests?
  a) Command-line tool support?
  b) Include Accuracy/Efficiency into Job Priority?

Future Work

- Fine tune Queue CPU and Memory limits.

- Investigate the impact of overestimation on blocking nodes for other jobs.

- Develop user support tools to automatically detect
  a) misconfigured jobs (GPU jobs on CPU nodes),
  b) bulk submissions to inappropriate queues,
  c) aggregation of specific wait time by reason.

Take-Away Messages

Backfilling works as intended

Dataset and Job Queues

Timespan	Jan – Aug 2023
#Accounts	~150
#Users	~500
#Jobs	~55 million
CPU Hours	~20 million
GPU Hours	~100 thousand

Queues	%Jobs	%CPUh	%Backfill	%Wait Time
30min	75.8%	12.3%	77.2%	34.1%
6hours	22.0%	49.3%	52.1%	43.6%
1day	1.9%	20.9%	30.4%	21.1%
1week	0.1%	12.6%	27.9%	1.0%
2weeks	>0.1%	0.8%	36.0%	>0.1%

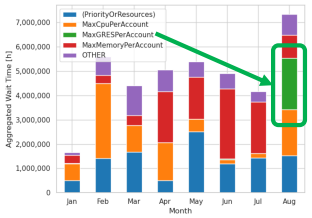
Shorter queues offer access to more resources.

Different job wait reasons

Wait Reasons per Month

- Slurm retains a wait reason per job other than Priority or Resources, in that case it stores "*None*".
- MaxCpuPerAccount and MaxMemoryPerAccount are limits per account and per queue.
- MaxGRESPerAccount is a GPU limit also per account and per queue.

➢ GPU limit wait time happened mostly in August 2023!



Appropriate queue submission

Potential to Improve Appropriate Queue Submissions

- We consider submission to a shorter queue possible if the **job execution time * 2 <= Timelimit** of the shorter queue (100% buffer/overestimation).

➢ The majority of the jobs from 6hours, 1day, 1week, and 2weeks queues could have been submitted to shorter queues.

➢ We know that submission to shorter queues is desirable, due to more access to resources, more backfilling and lower job wait times.

Submission Queue	Number of Jobs	% of Jobs where Submission to Shorter Queue was Possible
30min	39'820'630	already shortest queue
6hours	11'545'374	80.7%
1day	1'000'464	94.2%
1week	54'341	93.0%
2weeks	8'26	90.1%

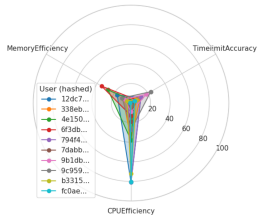
Low resource request accuracy

Potential for Appropriate Resource Requests

- The plot shows the top 10 users by CPUh.

➢ We observe high potential for these users to improve TimelimitAccuracy and MemoryEfficiency.

➢ The problem with overestimating memory
- the job waits longer itself
- the job blocks memory for other jobs
- this makes job scheduling difficult
- ultimately degrades the service for every job



Automatically detect problem cases

Detecting Problem Cases that Aggregate Specific Wait Time

➢ It can be an indication of misconfiguration if all jobs from a specific working directory are waiting only for one reason.

➢ Indeed we see a return of the GPU limit wait case! 99% of the wait time of all jobs from this working directory are due to GPU limits. Which could be avoided by submitting to the appropriate GPU queue.

➢ This form of reason-based analysis can support the detection of misconfiguration and problem cases.

WorkDir (hashed)	#Jobs	Wait Time [h]	Top Reason for 99% of Wait Time
a48a2...	13'608	2'107'958	MaxGRESPerAccount
5bc88...	8'206	290'445	MaxCpuPerAccount
d29c0...	21'125	222'892	MaxMemoryPerAccount
...	...	...	...

Opportunity to improve scheduling

Motivation

Problem Statement

➢ Unnecessary job wait time caused by inaccurate resource requests and low job efficiency.

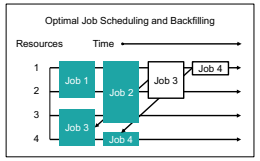
Challenge

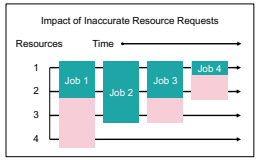
➢ A job's wait time is not always avoidable even on fully utilized systems – sometimes resources are busy.

Analysis Goal

➢ Understand and avoid "unnecessary" job wait time caused by inaccurate resource requests.

➢ Improve system utilization and efficiency.





University of Basel

DMI HPC

Unlocking the Potential of HPC Systems Use for Optimizing Job Queues, Minimizing Wait Times, and Boosting Efficiency

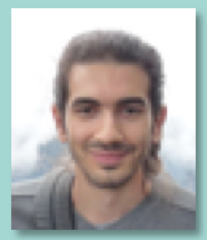
Thomas Jakobsche, Dr. Aurélien Cavelan, Dr. Rubén Cabezón, Dr. Iñaki Martínez de Ilarduya, Prof. Dr. Florina M. Ciorba

October 5, 2023 – hpc-ch forum on HPC and Data as a Service

hpc.dmi.unibas.ch | dmi-hpc@unibas.ch | @HPC_DMI_UniBas | @hpc_dmi_unibas

since 08/2015

Thomas Jakobsche



Aurélien Cavelan

Rubén Cabezón

Iñaki Martínez de Ilarduya

Florina M. Ciorba