Expansion of Backfill algorithm for increasing efficiency of supercomputer «Lomonosov»

Leonekov Sergei, SRCC MSU, Lomonosov Moscow State University
Zhumatiy Sergey, SRCC MSU

Department of Supercomputers and Quantum Informatics

SLURM

SLURM is a scalable, fault-tolerant cluster manager and scheduler for large computing systems. It is considered to be the most prospective resource manager. *

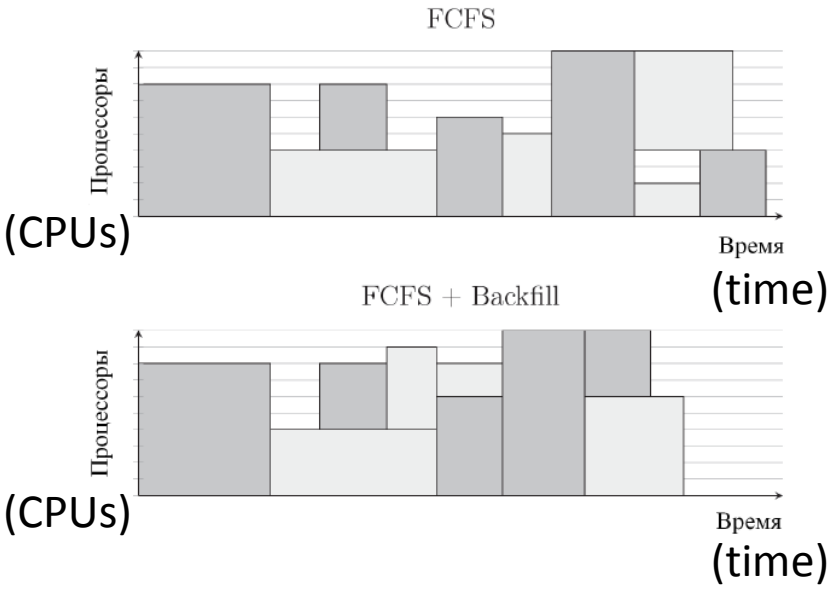
SLURM (v2.5.6) is used on supercomputer "Lomonosov".
SLURM (v15.08) is used on supercomputer "Lomonosov-2".

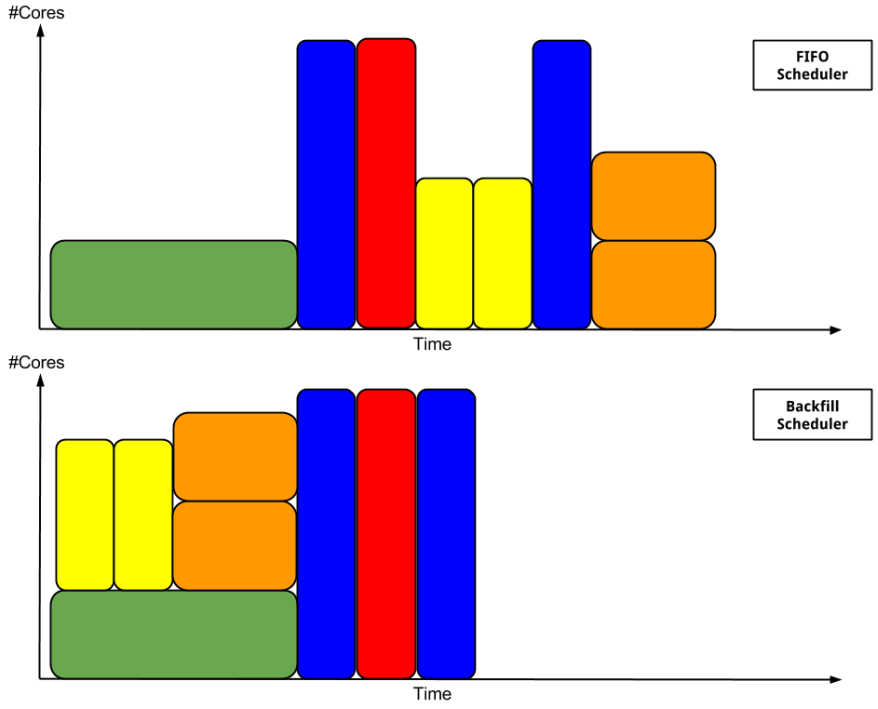
(*) Javier Cacheiro, Pablo Rey, Alvaro Simón,“Batch System Comparison Technical Report V4”, Cesga, 13/02/2014

Backfill

Scheduling algorithm Backfill is algorithm First Come First Served upgraded with "packing".

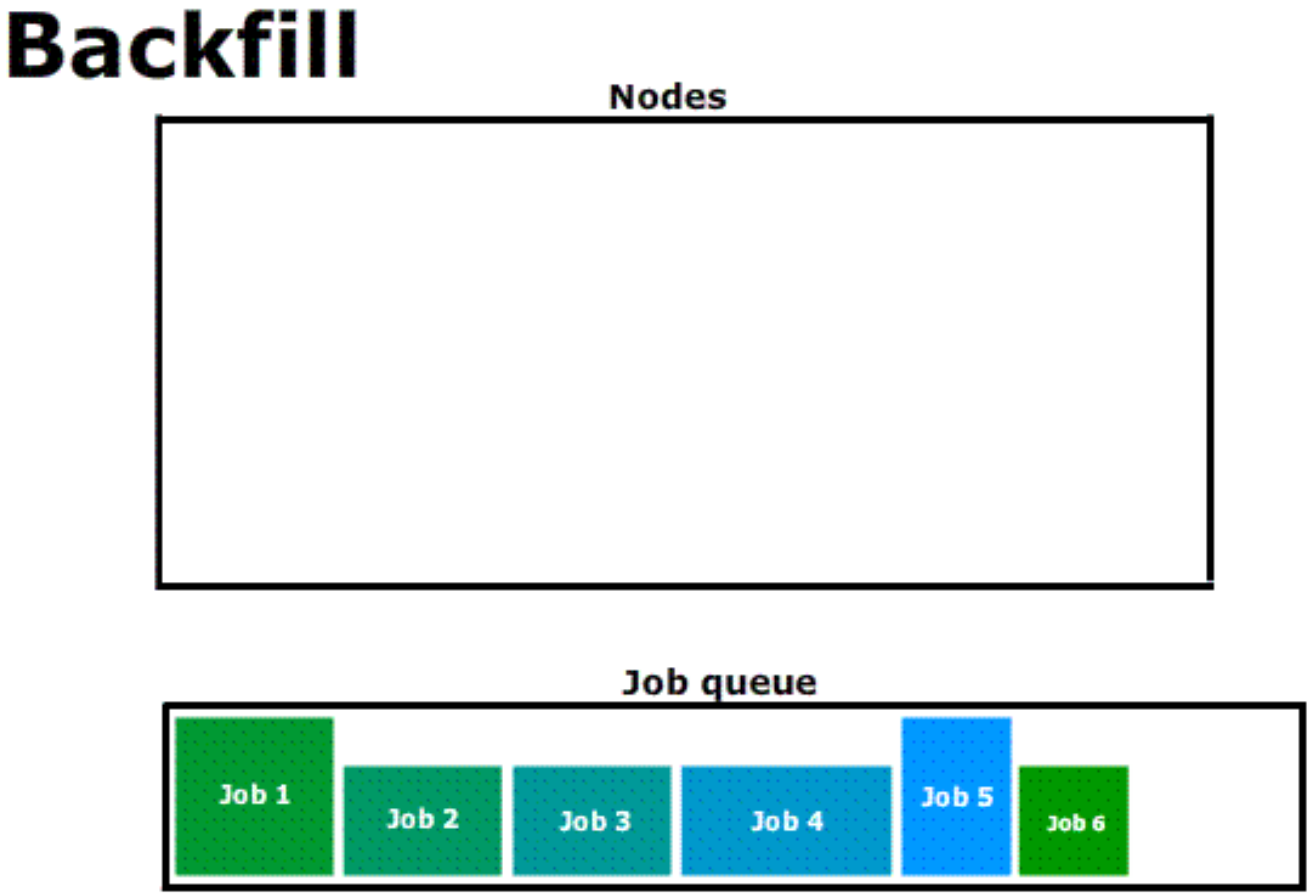
This scheduling algorithm is used in a large number of clusters and proved to be highly effective. Studies have shown that the algorithm can improve density of use of the supercomputers resources by ~ 20% and reduce the average waiting time for setting goals for performance. (*)





(*) David Jackson, Quinn Snell, Mark Clement “Core Algorithms of the Maui Scheduler”, Brigham Young University, Provo, Utah

Backfill example



MSU SRCC

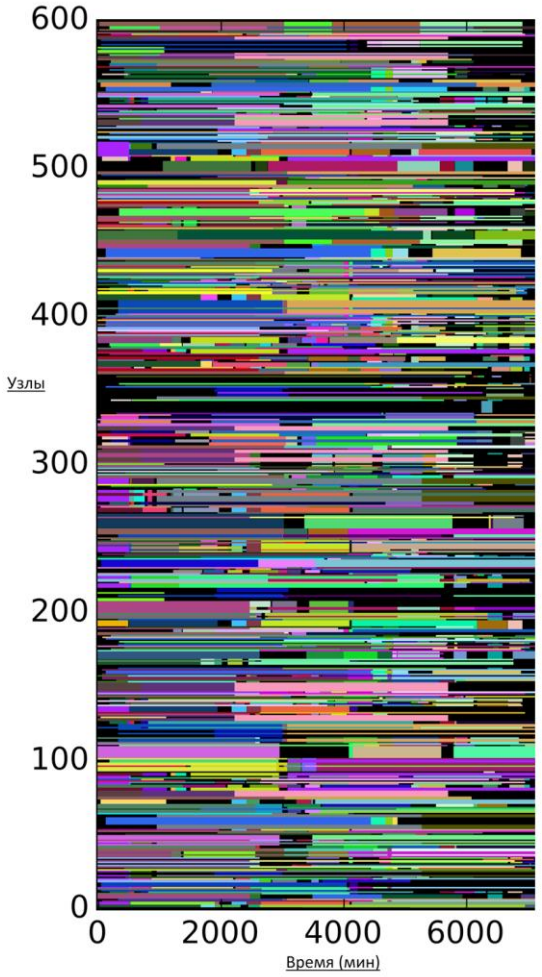
Supercomputer "Lomonosov"

- is ranked in the 2nd place in the Top50;
- has more than 400 "active" accounts during a year;
- runs 200-400 tasks every day.

Supercomputer "Lomonosov-2" is ranked in the 36th place in the TOP500.

If the priority “transparency” of each individual task is increased it will be able to use SLURM more intelligently.

~~Priority = 4294901717~~

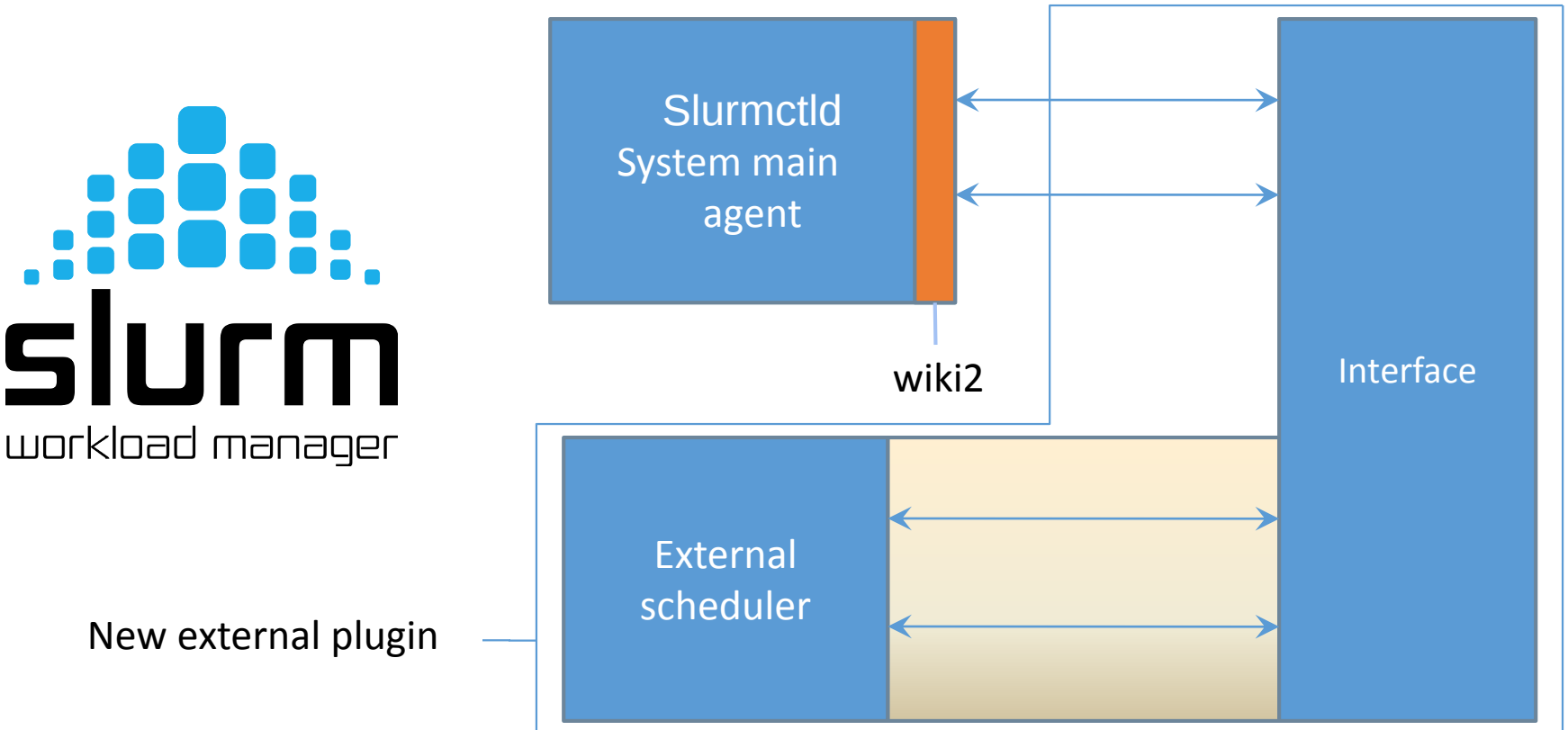


Queue Regular4

Development goals

The main goals are

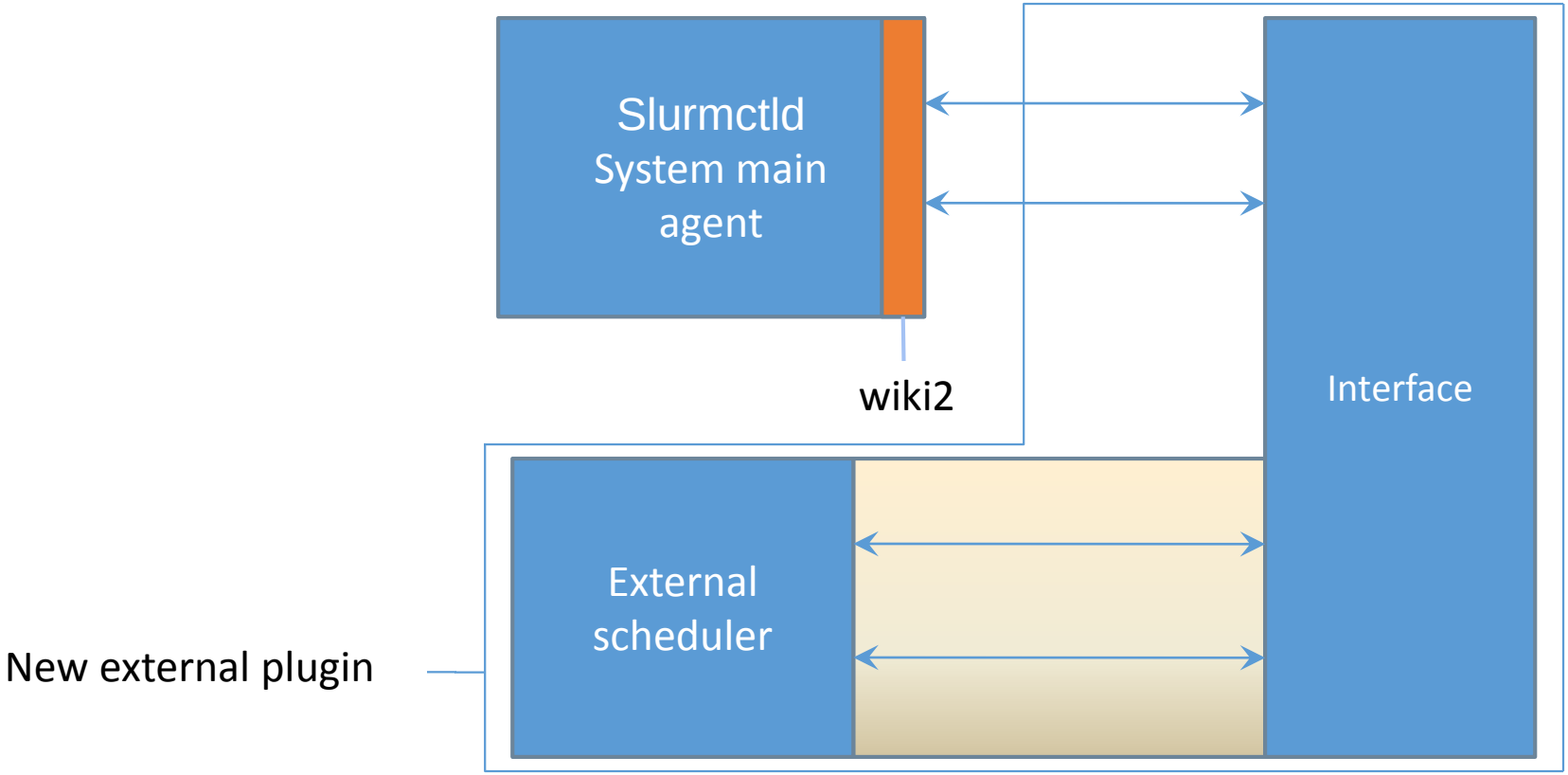
- To increase number of the users resourced at any time, "transparency" of priorities and ease of administration tasks queue using resource manager SLURM;
- To develop a planner with all of the features on the basis of SLURM.



External scheduler architecture

Types of SLURM/wiki2 messages: CANCELJOB, GETJOBS, JOBMODIFY, NOTIFYJOB, STARTJOB, INITIALIZE, REQUEUEJOB, SUSPENDJOB, RESUMEJOB, SIGNALJOB.

Two types of SLURM/wiki2 events: tasks state change AND slurmctld status change.

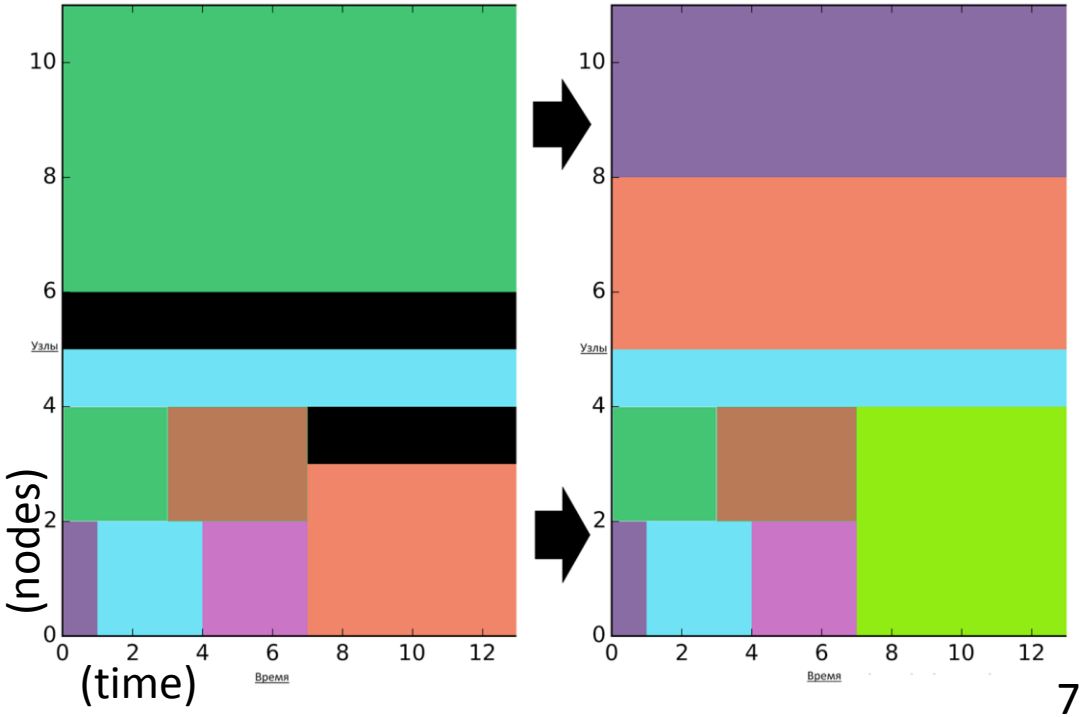


SLURM optimizations

New external scheduler functionality includes all the features of the built-in Backfill planning algorithm and introduces several new features:

1. The number of processor-hours requested by each user is taken into account;
2. There is a transparent priority system that can be adjusted in real time;
3. Users with a certain level of priority can use nodes from different sections;
4. Various time quotas can be set: a certain number of CPU hours per week/month/year;
5. Different planning algorithms can be used in different queues;

Experiments	Data (Acceleration)	
	Delay (user)	Delay (job)
Backfill	0,998	1
Backfill + CPU-h limit	1	0,907



“Transparency” of priorities

- Multifactor plugin:

Job_priority = (PriorityWeightAge) * (age_factor) + (PriorityWeightFairshare) * (fairshare_factor) + (PriorityWeightJobSize) * (job_size_factor) + (PriorityWeightPartition) * (partition_factor) + (PriorityWeightQOS) * (QOS_factor); // Priority = 4294901717

“Transparency”:

- takes a single limited scale for all users;
- uses data of every year users survey;
- uses “year work mark” weight;
- arranges priorities by special groups (students, university scientists and so on).

Scalable external scheduler

The solution is easily expandable thanks to its software primitives and completeness of the information that external scheduler receive from SLURM.

Priority system can be adjusted in real time.

New scheduling algorithm can be added without making any changes in SLURM core functions and plugins. Moreover, external scheduler interface is standardized and simplified as much as possible, making it easy to add and edit a new scheduling techniques.

Data:

- job info
- queue info
- SLURM state
- cluster state

...

Output:

- modified queue state after each step of external scheduling

New Backfill-based algorithm

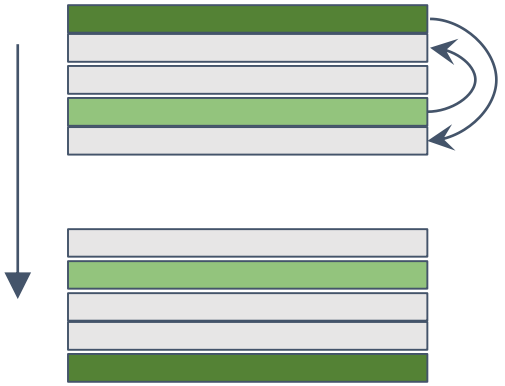
Standard scheduler cycle (single partition example) looks like:

1. Create a queue for the partition;
2. Sort tasks by priority;
3. Remove tasks (which is not startable) (CPU-limit);
4. Load data from "Acceleration" table;
5. "Accelerate" jobs from the table to the specified number of positions;
6. Find nodes for running jobs;
7. "Pack", if required;
8. Run tasks;
9. Update queue and node statuses and "Acceleration" table;
10. Return to Step 2.

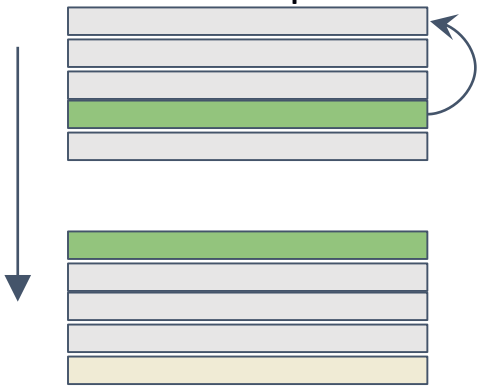
“Acceleration” table example:

JobID	User	Speed	Count	SpeedUp
1242	stud	0.3	0	2
…	…	…	…	…
…	…	…	…	…

“Acceleration” on each step:



“Boost” operation:

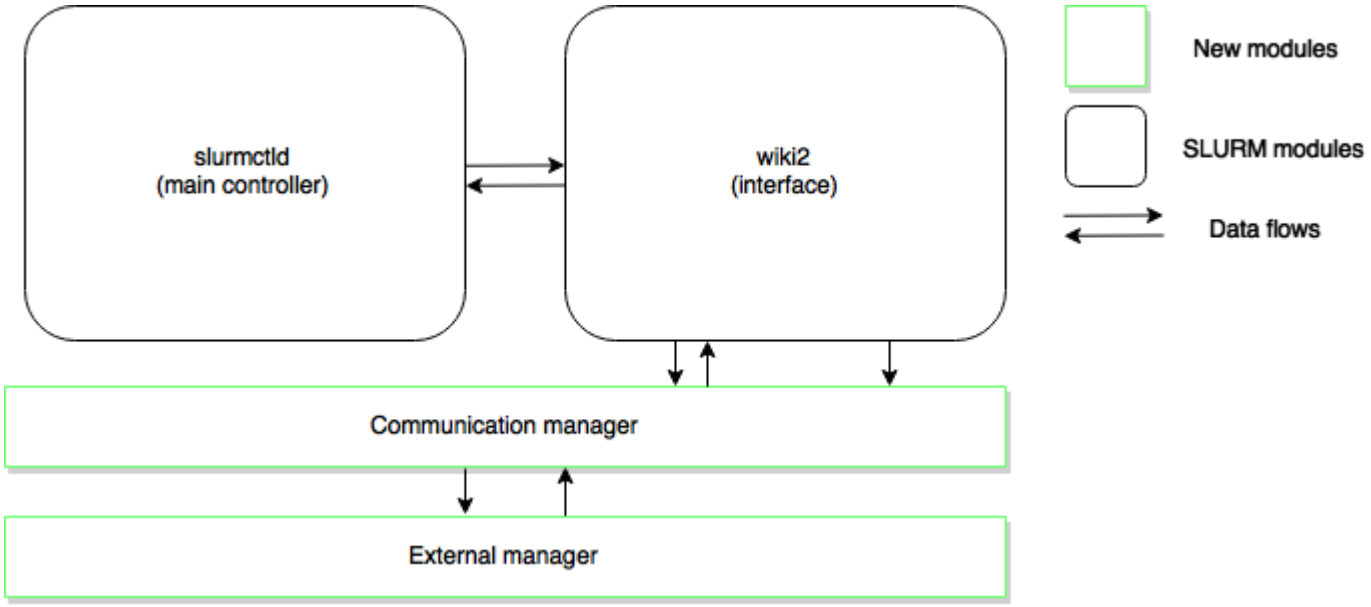


Conclusion

An external scheduler was implemented:

- based on SLURM primitives;
- scalable;
- portable;
- different planning algorithms can be used in different queues;
- simplified priority system;

The implemented system was tested on both “Lomonosov” and “Lomonosov-2” and ready for deployment.



Thank you for your attention!

Leonenkov Sergei
Lomonosov Moscow State University
Scientific Research Computing Centre
leonenkovs@gmail.com

Publications and presentations:

1. S. N. Leonenkov, S. A. Zhumatiy "Expanding the functionality of SLURM resource manager", XVI International Supercomputer Conference "Scientific service in the Internet: A variety of supercomputing worlds", 9/26/2014.
2. Poster presentation at the conference "Parallel Computing Technologies (PaVT) 2015", 01/04/2015.
3. S. N. Leonenkov, S. A. Zhumatiy "Introducing new backfill-based scheduler for SLURM resource manager", YSC-2015, Greece

Leonenkov Sergey
Lomonosov Moscow State University
Scientific Research Computing Centre
leonenkovs@gmail.com