Алгоритмы планирования менеджера ресурсов SLURM и эффективность использования суперкомпьютера «Ломоносов»

Леоненков Сергей, аспирант ВМК МГУ, НИВЦ МГУ
Жуматий С. А. к.ф.-м.н., в.н.с. НИВЦ МГУ

КАФЕДРА
суперкомпьютеров
и квантовой
информатики

SLURM

КАФЕДРА суперкомпьютеров и квантовой информатики

SLURM – это высокомасштабируемый отказоустойчивый менеджер кластеров и планировщик заданий для таких систем. Считается наиболее перспективным менеджером ресурсов. *

slurm workload manager

SLURM (v2.5.6) используется на суперкомпьютере «Ломоносов»,
SLURM (v15.08) используется на суперкомпьютере «Ломоносов-2».

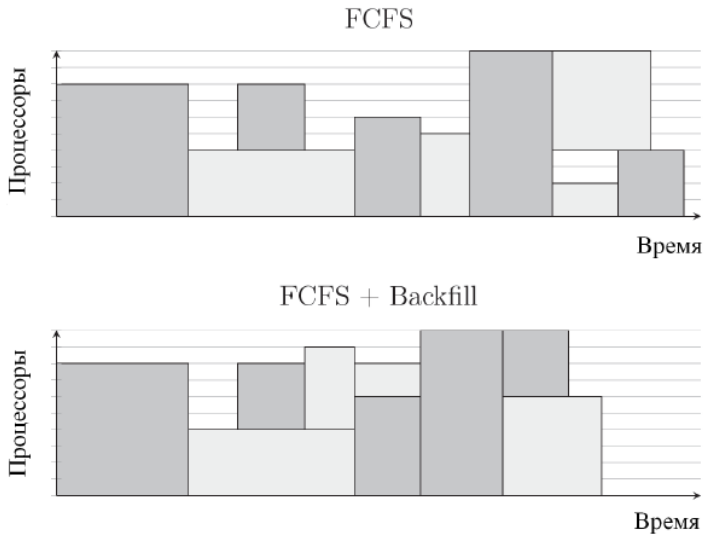
(*) Javier Cacheiro, Pablo Rey, Alvaro Simón,“Batch System Comparison Technical Report V4”, Cesga, 13/02/2014

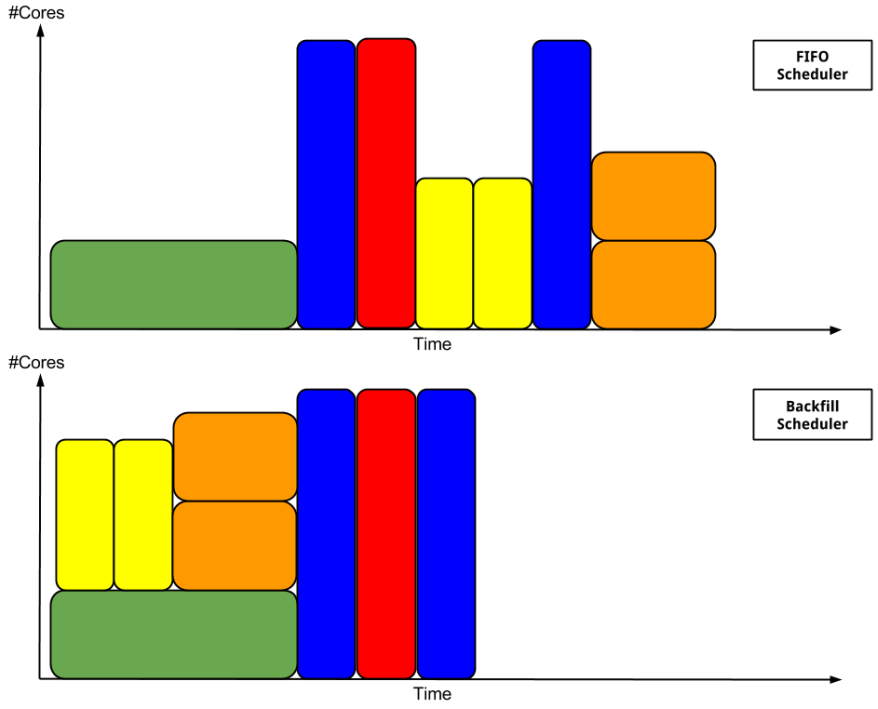
Backfill

КАФЕДРА суперкомпьютеров и квантовой информатики

В основе алгоритма Backfill лежит стандартный First Come First Served с оптимизацией «уплотнение».

Исследования показали, что алгоритм позволяет повысить плотность использования ресурсов суперкомпьютеров на примерно 20% и уменьшить среднее время ожидания задачами постановки на исполнение. (*)





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

«Ломоносов»

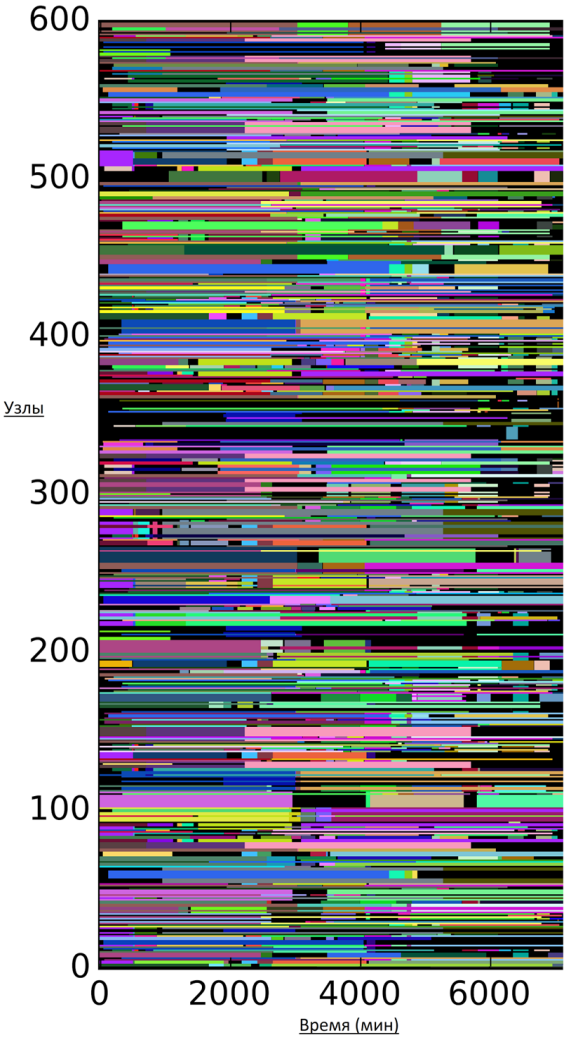
КАФЕДРА суперкомпьютеров и квантовой информатики

На суперкомпьютере «Ломоносов» используется стандартный планировщик и алгоритм Backfill.

Вычисление приоритетов в архитектуре системы SLURM на суперкомпьютере «Ломоносов» отдано специальному плагину, который идет в стандартной комплектации менеджера ресурсов, Multifactor Priority Plugin.

Прозрачность системы приоритетов.

~~Priority = 4294901717~~



Часть очереди Regular4

Метрики эффективности

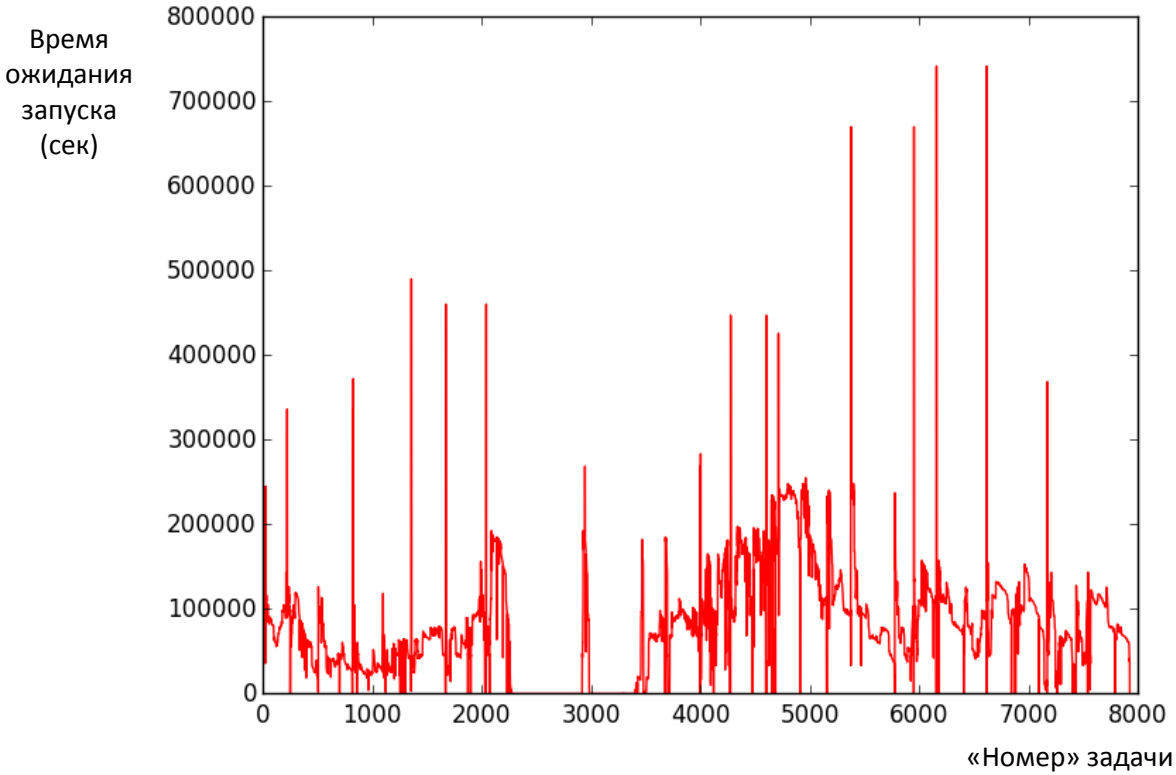
Предлагаемый набор метрик:

1) Утилизация процессорного времени;
2) Скорость старта первой задачи пользователей с момента запуска;
3) Отношение количества запущенных задач к единице времени («Скорострельность»);
4) Количество пользователей, обслуженных в единицу времени;
5) Суммарное время до предсказанного старта каждой задачи в очереди;
6) Отношение runtime к estimates; аналог 5);
7) Среднее время постановки задачи на исполнение;
8) Суммарное время простоя узлов очереди («Потерянные процессоро-часы»); аналог 1);
9) Скорость прохождения задач пользователей на запуск (Среднее суммарное время от постановки в очередь до запуска заданий каждого пользователя/ или же отдельно по заданиям);
10)Скорости запуска типов задач (маленькие/большие/средние);

Сколько ждать в очереди?

КАФЕДРА суперкомпьютеров и квантовой информатики

В среднем задачи стояли в очереди более 22 часов!

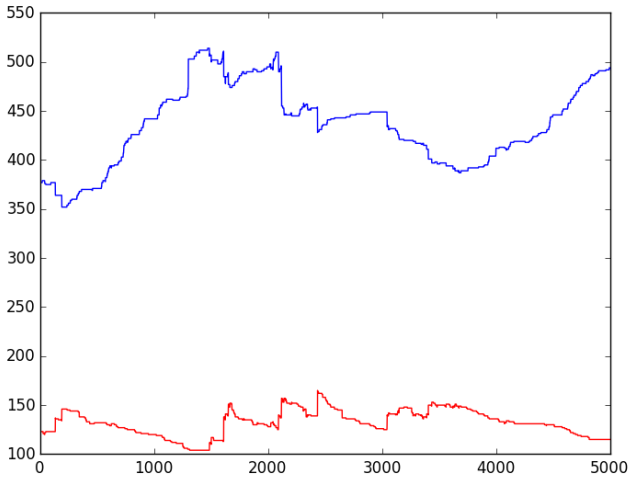


Загруженность системы

КАФЕДРА суперкомпьютеров и квантовой информатики

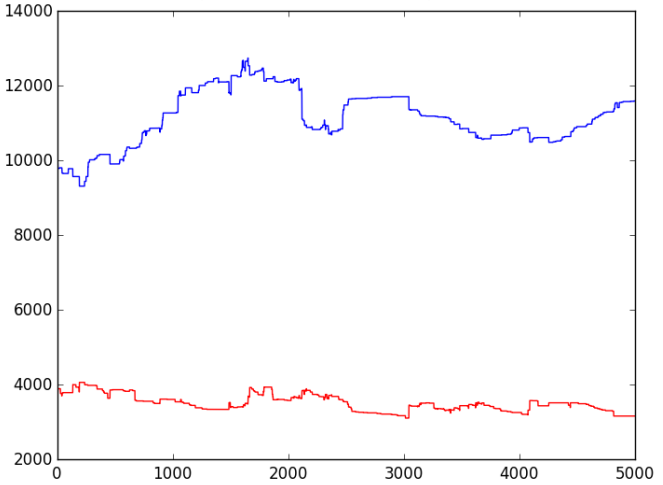
Задач в очереди

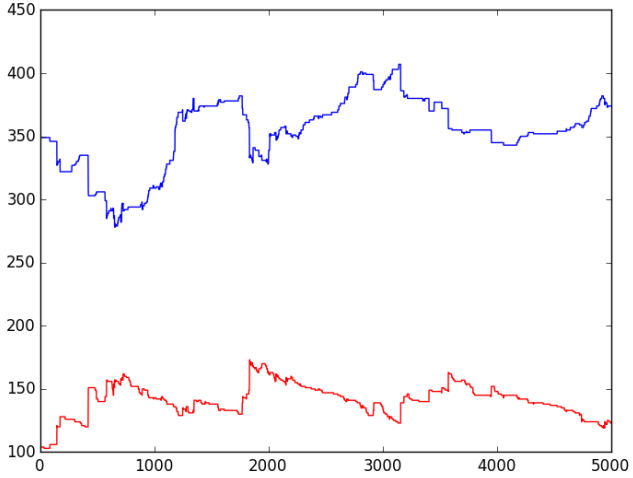
Кол-во задач (шт.)



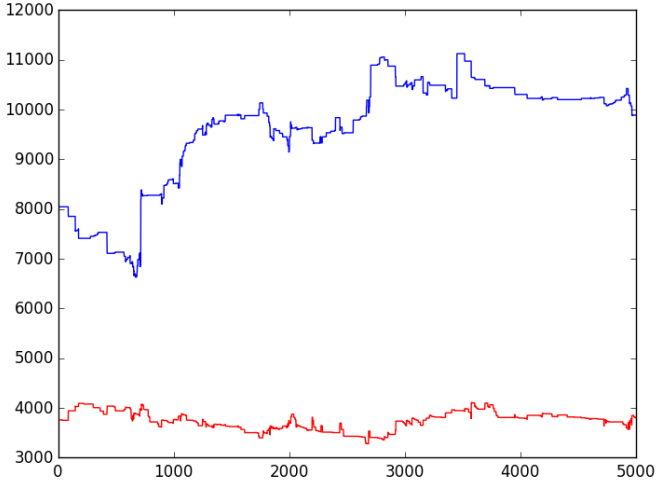
Суммарное кол-во запрошенных узлов

Кол-во задач (шт.)





Время (сек)



Время (сек)

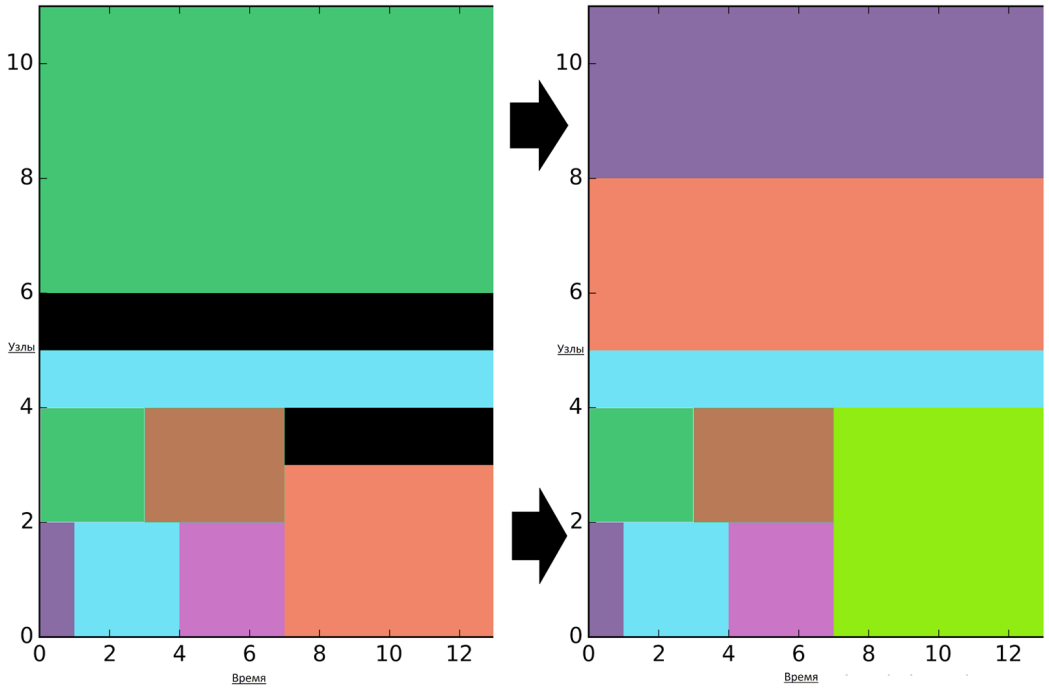
SLURM optimizations

КАФЕДРА суперкомпьютеров и квантовой информатики

Для оптимизации работы планировщика SLURM были внесены следующие дополнения:

- Учет и контроль процессоро-часов, запрошенных каждым пользователем;
- Создание прозрачной системы приоритетов с возможностью настройки в режиме реального времени;
- Реализация возможности использования узлов из разных разделов для пользователей с определенным приоритетом;
- Добавление квот по времени: определённое число процессоро-часов в неделю/месяц/год;
- Реализация возможности использования разных алгоритмов планирования в разных очередях.

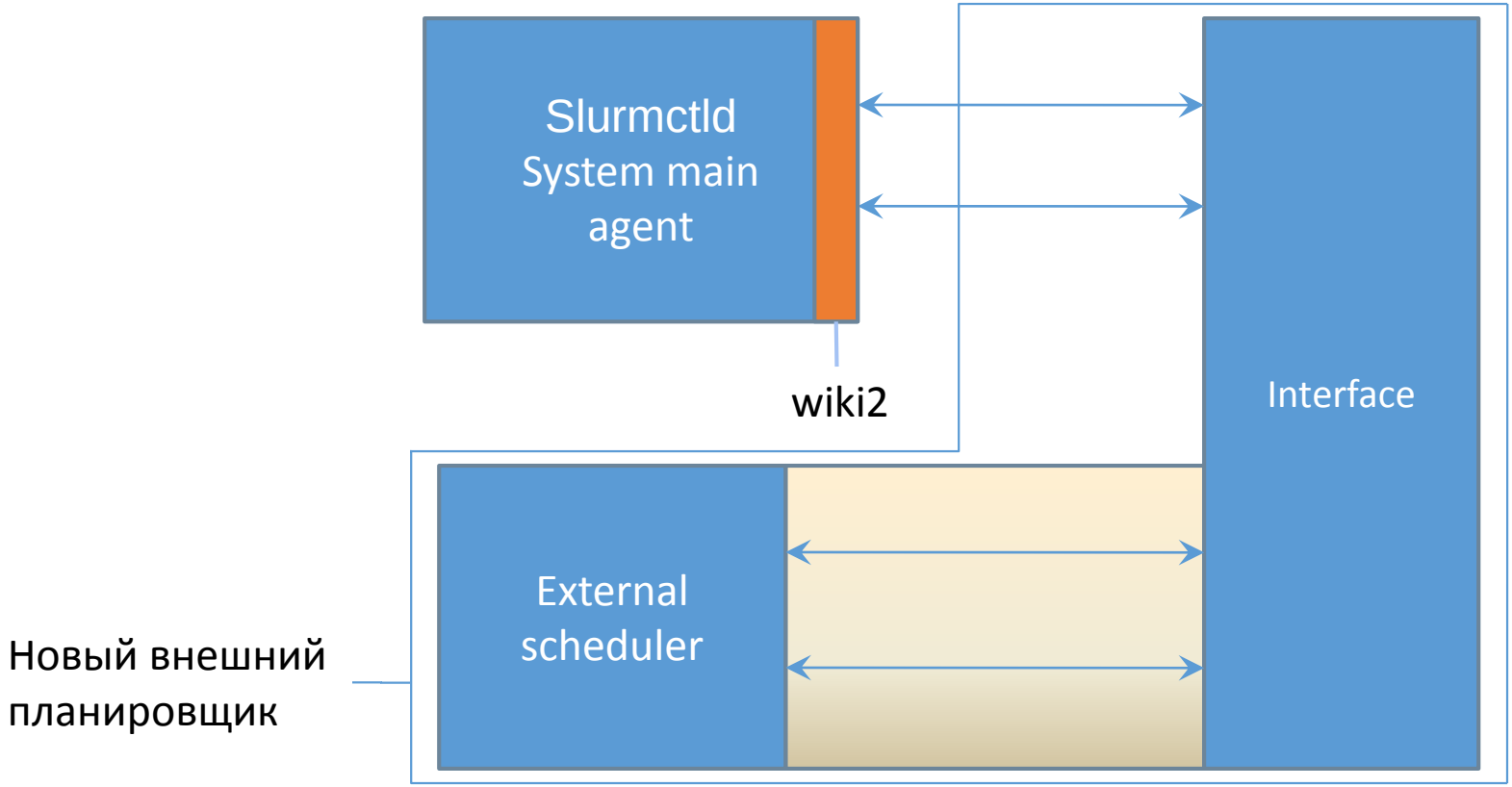
Slurmctld
System main
agent



Внешний планировщик

Типы сообщений SLURM/wiki2: CANCELJOB, GETJOBS, JOBMODIFY, NOTIFYJOB, STARTJOB, INITIALIZE, REQUEUEJOB, SUSPENDJOB, RESUMEJOB, SIGNALJOB.

Два типы событий SLURM/wiki2: tasks_state_change AND slurmctld_status_change.



Внешний планировщик

Решение легко расширяемо благодаря использованию примитивов SLURM и полнота информации, которую передает менеджер ресурсов.

Система приоритетов может быть изменена «на ходу».

Новый алгоритм планирования может быть добавлен без внесения любых изменений в SLURM. Более того, внешний планировщик имеет стандартный набор примитивов, что позволяет легко добавлять и исправлять алгоритмы планирования

Данный:

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

...

На выходе:

- «новая» конфигурация очереди, запуск задач.

“Прозрачность” приоритетов

- Multifactor plugin:

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

“Прозрачность приоритетов”:

- Единая укороченная шкала для пользовательских приоритетов;
- Использование ежегодных опросов пользователей;
- Учет ежегодной оценки работы пользователя;
- Приоритеты по специальным группам (студенты, исследовательские группы и т.д.)

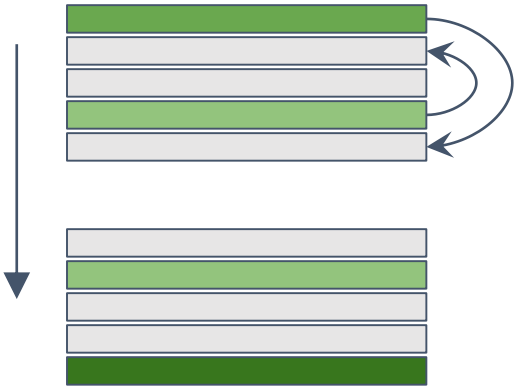
Предлагаемые оптимизации

КАФЕДРА суперкомпьютеров и квантовой информатики

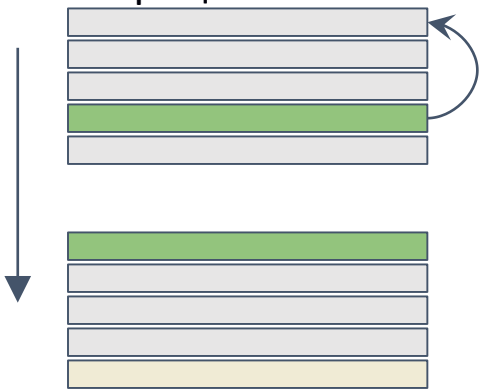
Стандартный цикл планировщика (пример для одной очереди):

1. Создание очереди;
2. Сортировка по приоритетам;
3. Удаление «лишних» задач (CPU-limit);
4. Загрузка информации из таблицы «Ускорений»;
5. «Ускорение» задач;
6. Распределение свободных узлов;
7. «Упаковка», если требуется;
8. Запуск заданий;
9. Обновление таблицы «Ускорений» и очереди;
10. Возвращаемся на шаг 2.

«Ускорение» на каждом шаге:



Операция «Boost»:



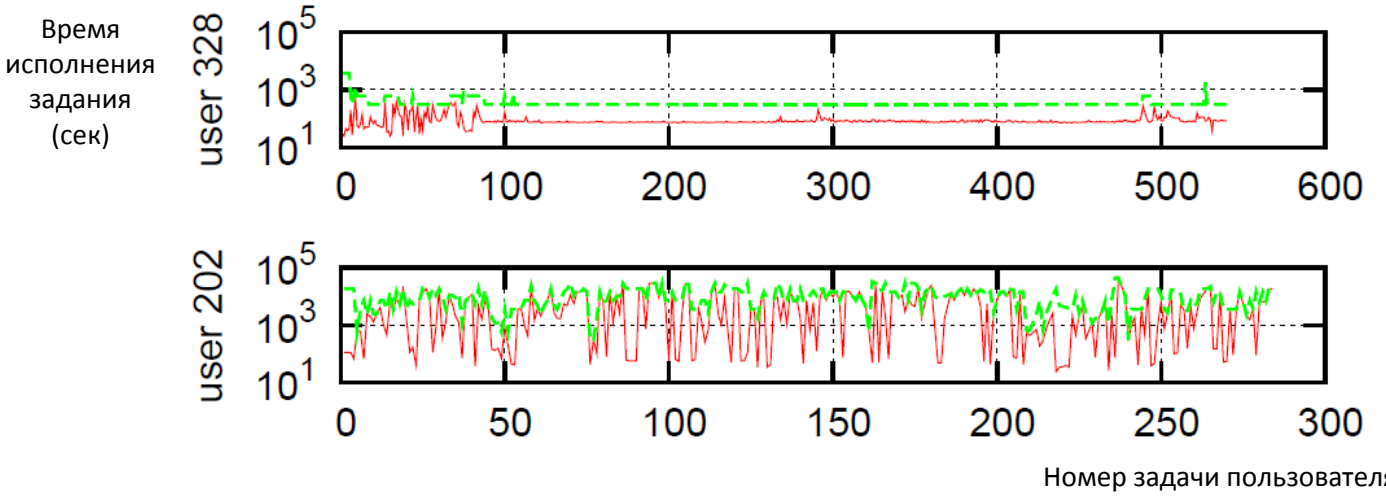
Пример таблицы «Ускорений»:

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

Сколько ждать в очереди?

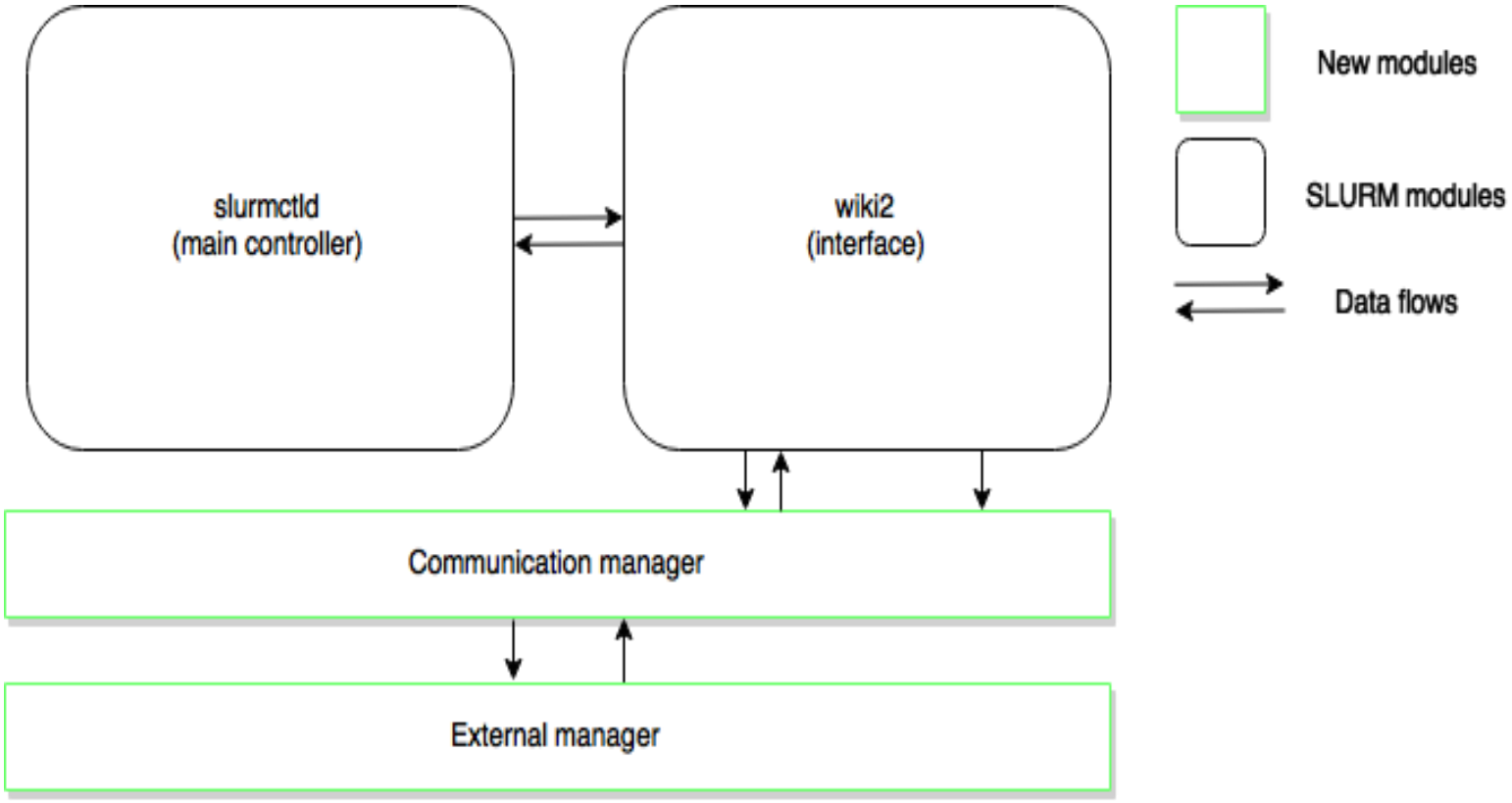
КАФЕДРА суперкомпьютеров и квантовой информатики

Эксперименты	Data (Acceleration)	
	Delay (user)	Delay (job)
Backfill	0,998	1
Backfill + Optim.	1	0,907



Заключение

КАФЕДРА суперкомпьютеров и квантовой информатики



Спасибо за внимание!

Леоненков Сергей
leonenkovs@gmail.com

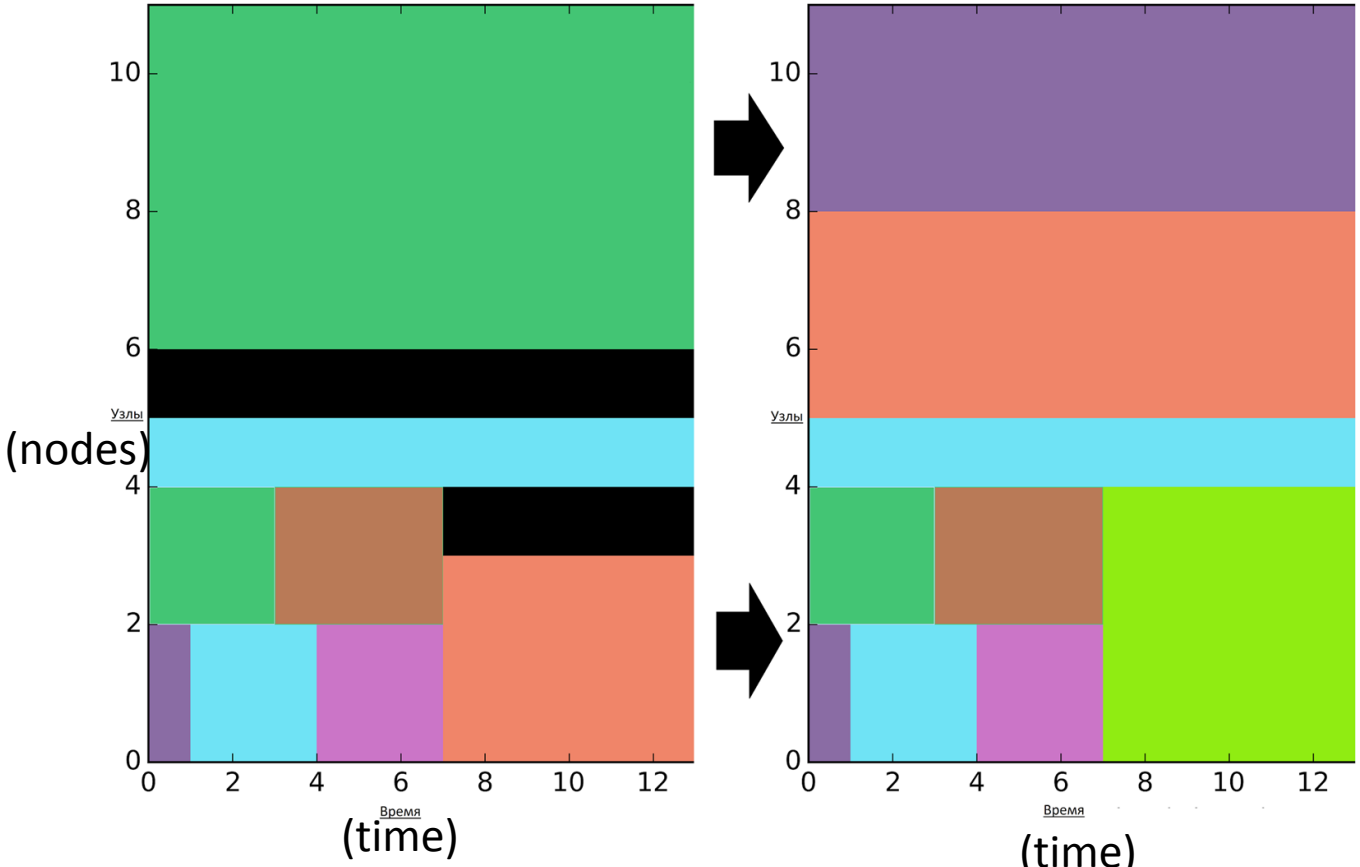
Publications and presentations:

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

CPU hours limit

Пример использования лимита процессорочасов.



Типичная ситуация

New Backfill-based algorithm

Стандартный цикл работы планировщика на примере одной партиции выглядит так:

1. Создание очереди заданий для партиции;
2. Сортировка заданий по приоритету;
3. Удаление задач, которые не представляется возможным запустить (CPU-limit);
4. Загрузка данных из таблицы «ускорений»;
5. «Ускорение» заданий из таблицы на указанное кол-во позиций;
6. Нахождение узлов для запуска заданий;
7. «Уплотнение», если требуется;
8. Запуск заданий;
9. Обновление очереди, статусов задач и узлов, таблицы «ускорений»;
10. Возврат к пункту 2.