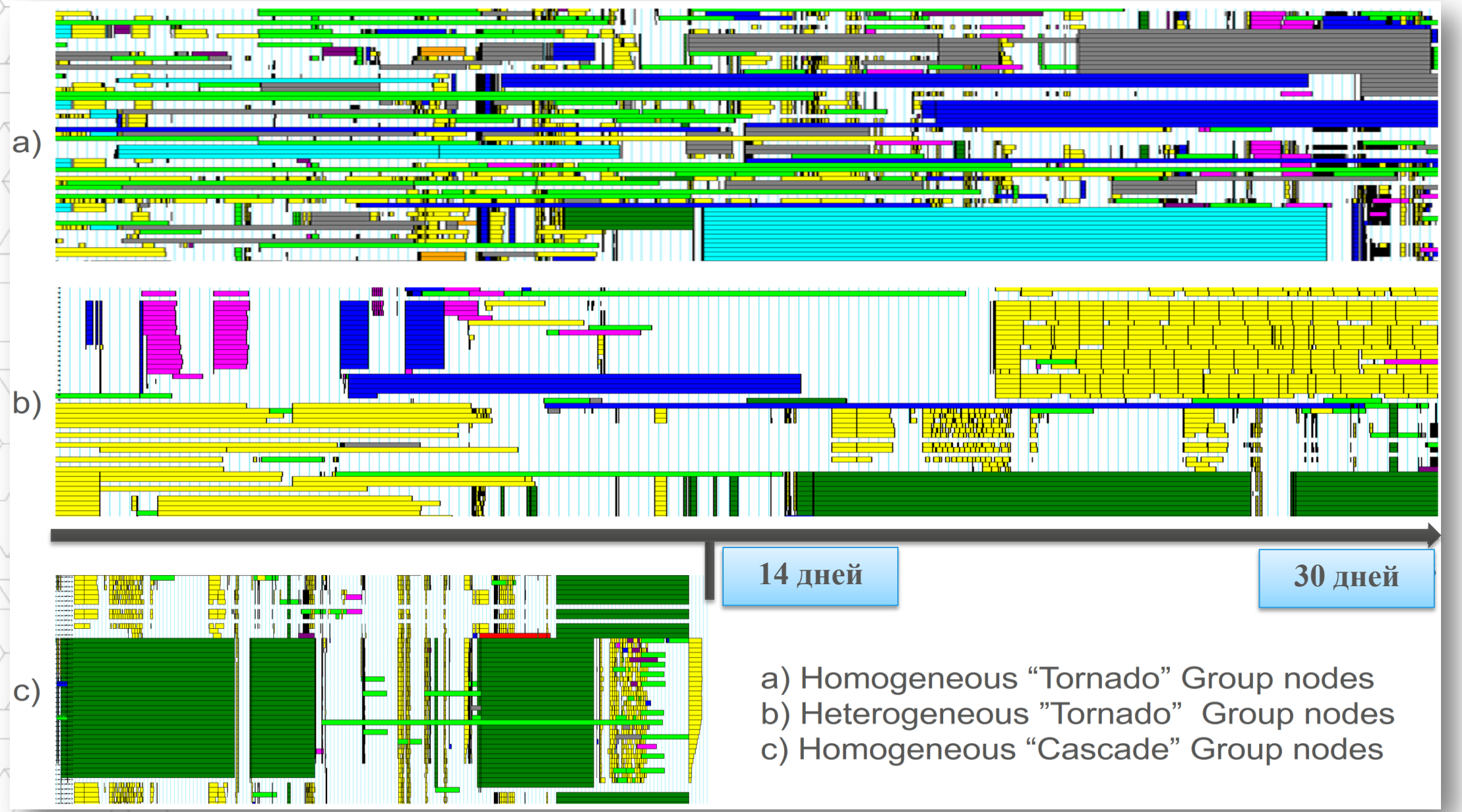
ПОЛИТЕХ
Санкт-Петербургский политехнический университет Петра Великого

ЦКП СКЦ «Политехнический»: Загрузка узлов кластера

СКЦ «Политехнический»,
ВШ искусственного интеллекта, ИКНТ

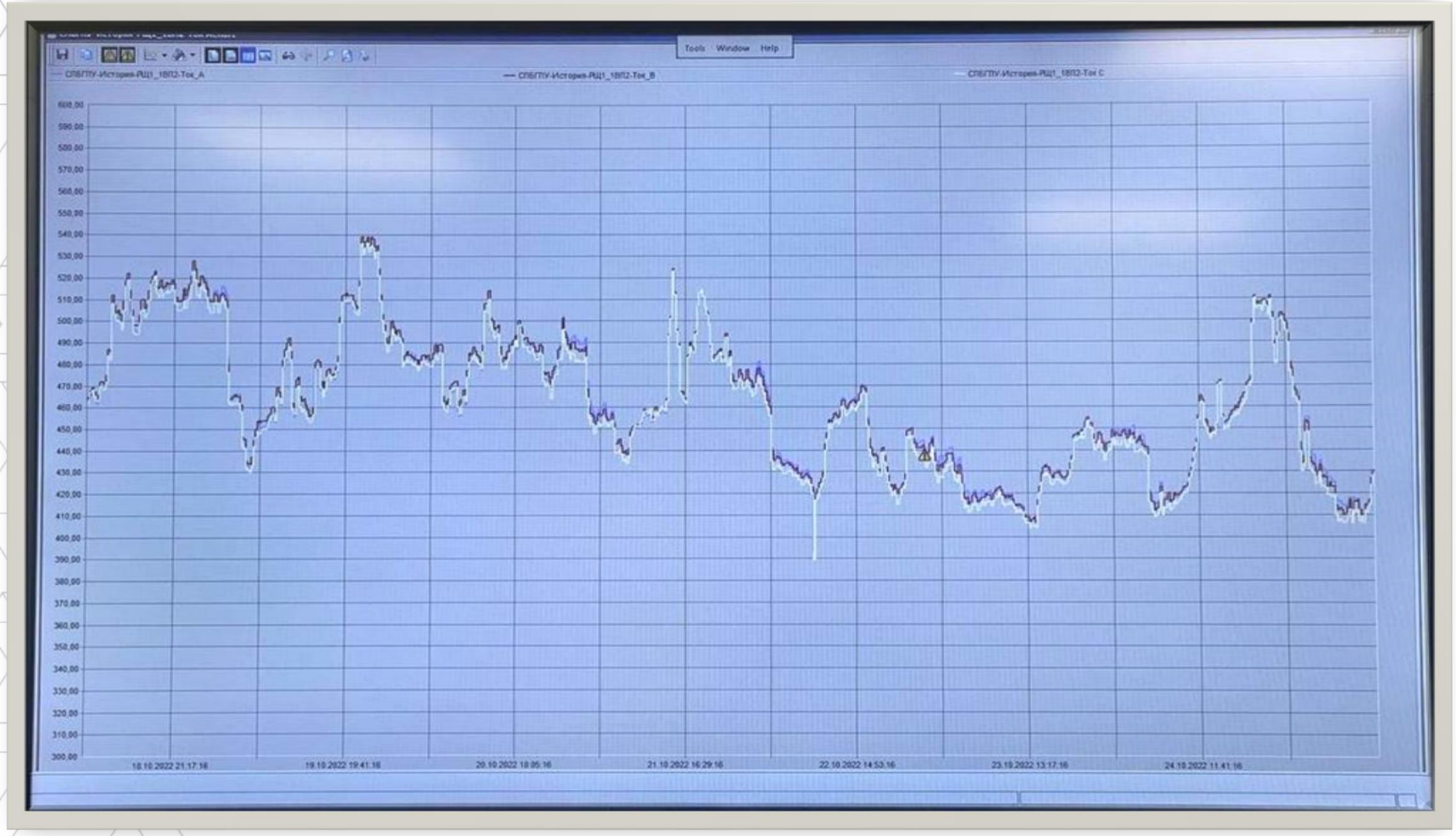
СПб
16 октября 2023 г.

Загрузка узлов кластера и резервы увеличения производительности СК



a) Homogeneous “Tornado” Group nodes
b) Heterogeneous ”Tornado” Group nodes
c) Homogeneous “Cascade” Group nodes

График потребления эл. тока вычислительными узлами СК



Точность предсказания времени приводит к повышению эффективности потребления ресурсов СК

На изображении **оранжевым миганием** показаны простои в работе одного из вычислительных кластеров (гомогенный кластер Tornado). Это значит, что в эти промежутки времени узлы не проводили вычислений и ожидали новых задач. На иллюстрации узлы заняты на 68.9%

Точность предсказания времени приводит к повышению эффективности потребления ресурсов СК

Пояснение. На самом деле, очередь из заявок на вычисления в СКЦ «Политехнический» почти никогда не пустует и, в среднем, время ожидания заявки в очереди составляет не меньше получаса. Это время ожидания зависит от того…

- сколько других заявок уже находится в очереди;
- как быстро освободится какой-либо из узлов (в свою очередь, зависит от продолжительности задач, который **на данный момент вычисляются**, короткие быстрее освободят место);
- насколько короткая задача ожидает в очереди заявок на исполнение, короткую задачу SLURM всегда сможет «втиснуть» на вычисление почти без очереди (минимально ожидая)…

Тут важно отметить, что если фактически задача маленькая, но пользователь указал для нее время, неразумно превышающее в несколько раз ожидаемое, то SLURM попросту не сможет оптимально ею заполнить пропуски в очереди, и отправит задачу в конец очереди, а узлы, которые могли бы эту задачу «обслужить» будут простаивать… Вины SLURM в этом никакой нет, и никакой самый интеллектуальный алгоритм упаковки узлов в очереди не поможет, **если он не может изменить параметры заявок пользователя**.

В простое других узлов виноват «некомпетентный» пользователь, а ведь он даже может и не задумываться о том, что страдают от этого другие, и узлы теряют драгоценное время впустую;

Задачу, у которой пользователь указал неверное количество времени, или оставил значение по умолчанию, будем называть «**ошибочной**». Если «ошибочных» задач слишком много, то очередь будет «накапливать» в себе ошибки, т.к. «ошибочные» задачи также не пропустят вперед другие задачи, даже если у других задач время оценено верно. Особенно «фатальными» являются ошибки у задач, которые запущены на более, чем одном узле.

Т. к. из-за своей «громоздкости», им тяжело найти место в очереди, то и с очень малой вероятностью планировщик (при очередных циклах перестройки очереди) сможет найти другое оптимальное место в очереди этой задаче, серьезно мешая другим задачам запуститься, и оставляя перед собой **большие простои сразу нескольких узлов**, на которых будет запущена задача. Примеры таких простоев отмечены на следующем слайде красными кругами.

Точность предсказания времени приводит к повышению эффективности потребления ресурсов СК

Иллюстрация проблемы неэффективного планирования

Рассмотрим цикл постановки заявок задач в очередь, чтобы проиллюстрировать проблему **неверных параметров задачи** и важность вмешательства «третьей стороны» в процесс постановки задач для **повышения «реальной производительности»**:

Заявка 1: 1 узел, 12 часов

12 часов

Заявка 2: 5 узлов, 2 часа

2 часа
2 часа
2 часа
2 часа
2 часа

Этап 1. Пользователи выставляют свои заявки на выполнение задач на кластере. Наиболее важными критическими для планирования являются:

- число запрошенных узлов;
- запрошенное время.

Заявка 3: 3 узла, 16 часов

16 часов
16 часов
16 часов

Заявка 4: 1 узел, 4 часа

4 часа

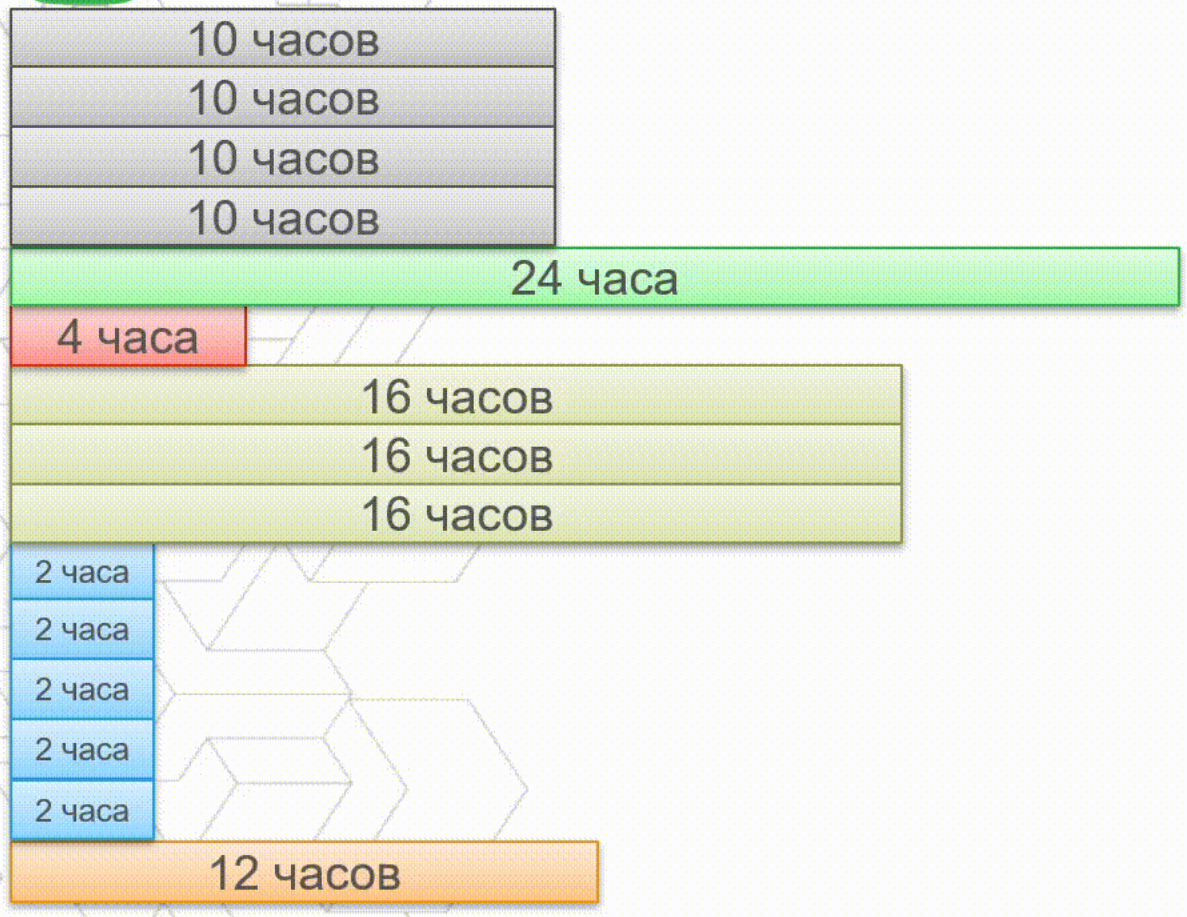
Заявка 5: 1 узел, 24 часа

24 часа

Заявка 6: 4 узла, 10 часов

10 часов
10 часов
10 часов
10 часов

Иллюстрация распределения заявок по узлам кластера



<u>Этап 2.</u> SLURM распределяет заявки из очереди, назначая им номера узлов и время начала выполнения. Еще неизвестно насколько пользователи ошиблись во времени.

Иллюстрация простоев узлов кластера после неэффективного планирования

Этап 3. После распределения заявок на стадии планирования остались «дыры» в очереди задач, **узлы неизбежно будут простаивать**. Задачи еще не были запущены.

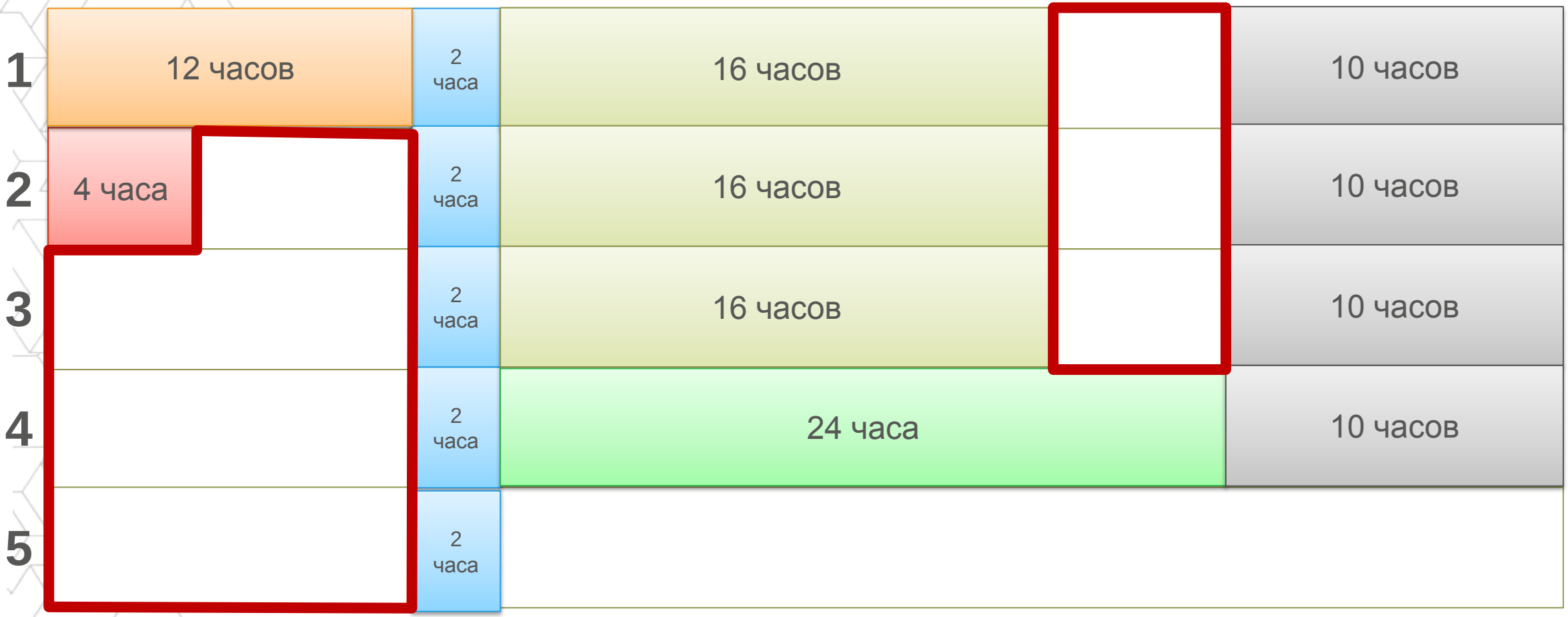
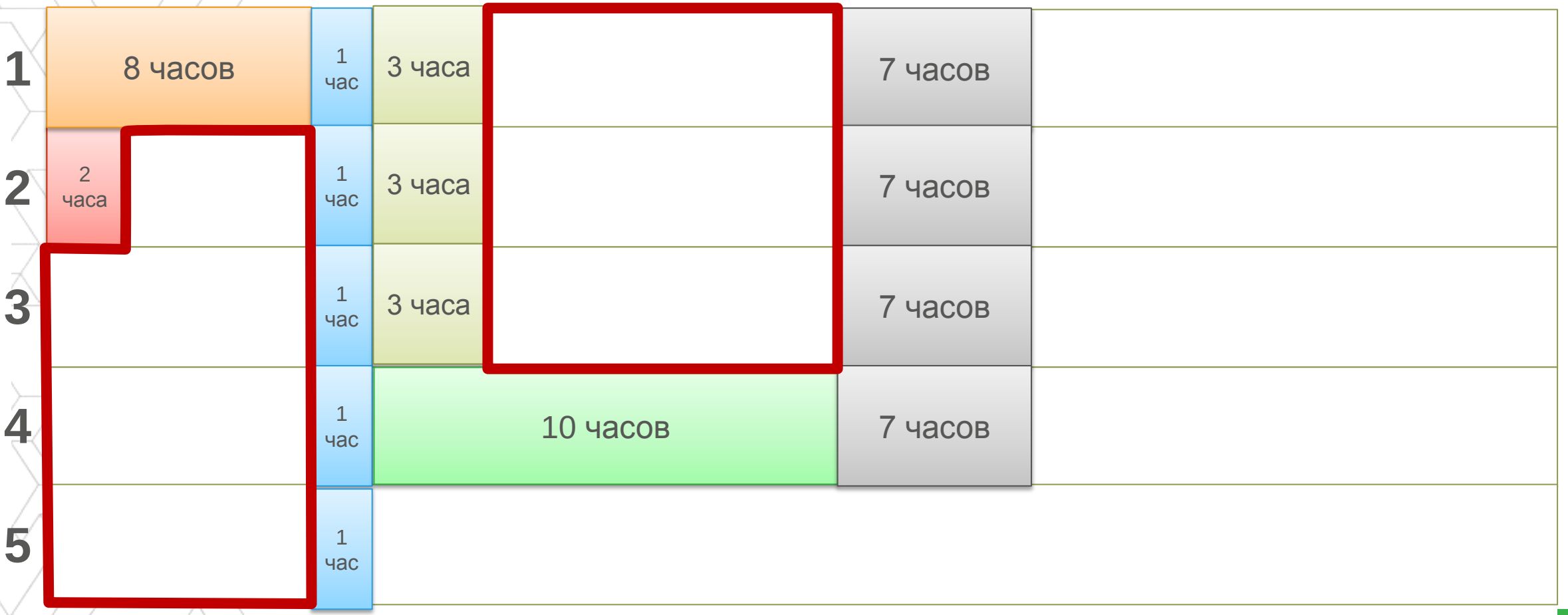


Иллюстрация перестройки очереди во время выполнения задач

Этап 4. Задачи запущены и очередь динамически перестраивается после завершения очередной задачи, но это не приносит существенного улучшения ситуации. В большинстве случаев, **реальное время выполнения задач меньше**, заданного пользователем, поэтому важно изменять параметры запуска пользователя **до начала вычислений**, чтобы избежать простоев в кластере и предотвратить аварийное завершение задачи там, где это возможно.

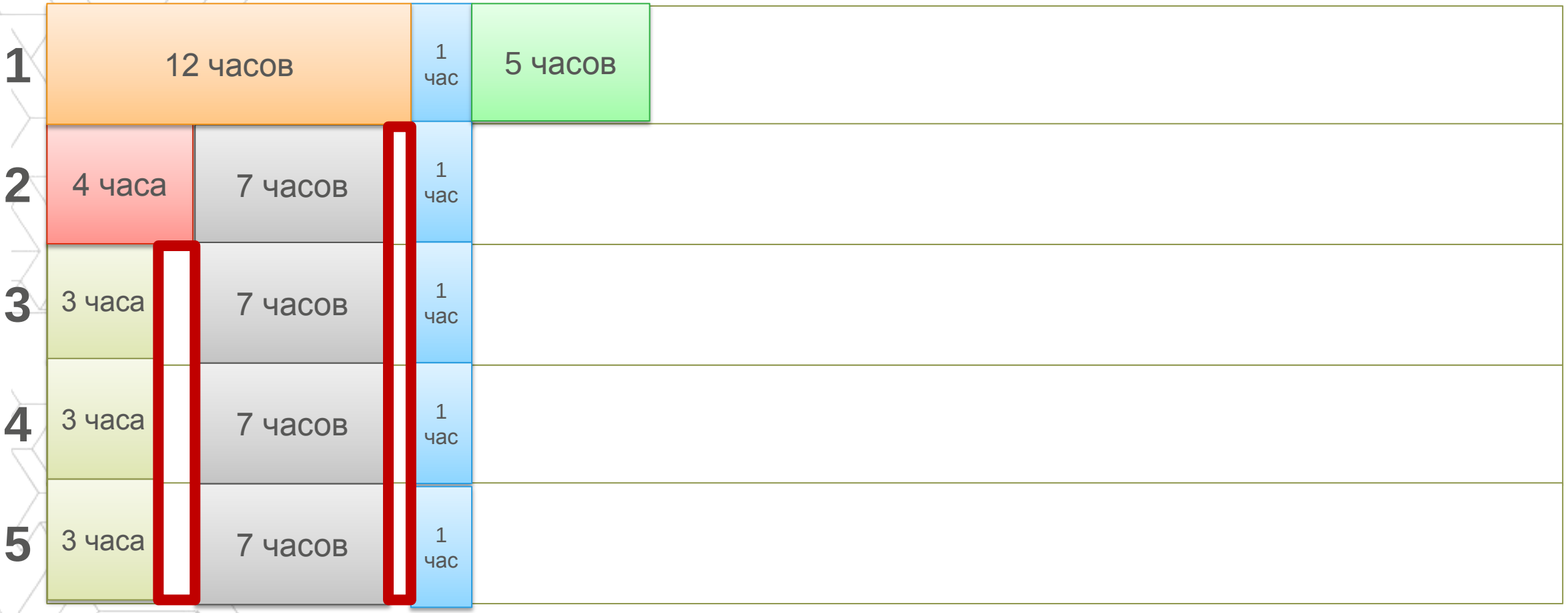
ФАКТИЧЕСКАЯ ЗАГРУЗКА УЗЛОВ БЕЗ ВМЕШАТЕЛЬСТВА «ИНТЕЛЛЕКТУАЛЬНОГО ДИСПЕТЧЕРА»

Этап 5. После запуска, задачи отработали фактически за меньшее время. Благодаря динамическому пересчету очереди, SLURM удалось немного уменьшить один из простоев узлов, но увеличилась длительность второго простоя. **Прогноза времени исполнения** и изменений заявок пользователей **не производилось**. Если предсказать параметры задачи до начала их исполнения, то простои можно избежать.

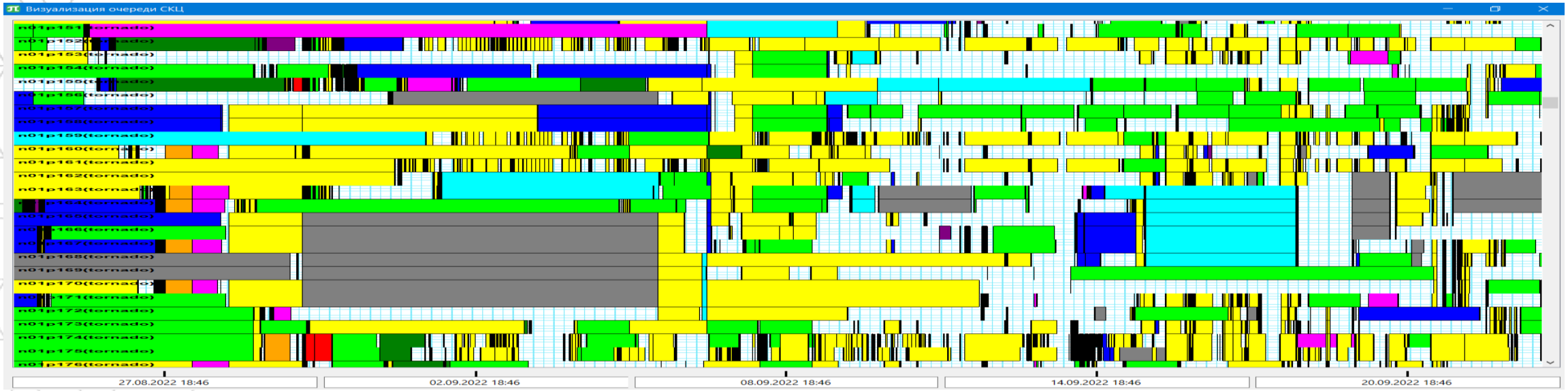


Результат прогноза алгоритма и изменения параметров задач

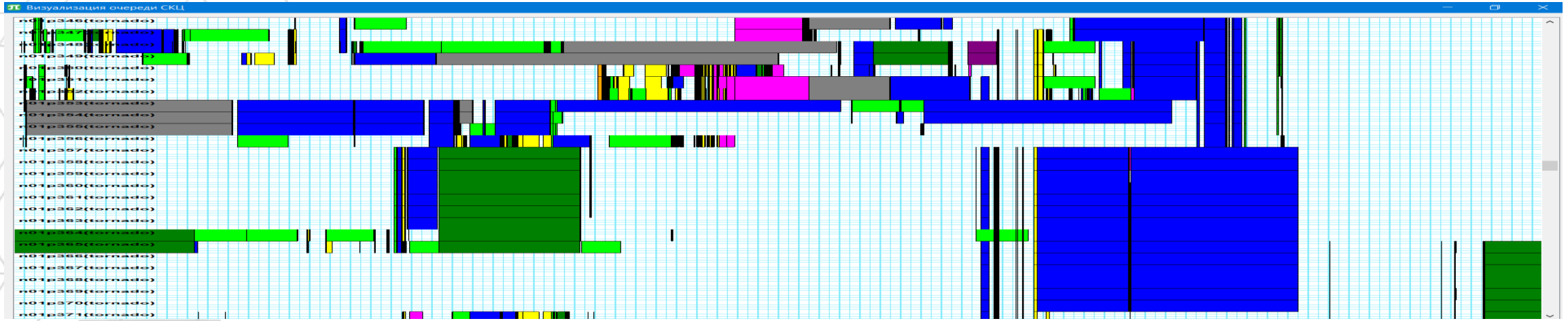
Оптимизированная очередь. Предполагаемая очередь, при которой был произведен прогноз времени исполнения и изменены параметры заявок пользователей до начала вычислений. За меньшее время удалось вычислить большее число задач. Реальная производительность кластера существенно повышена. Это достигнуто не только за счет более «плотной» упаковки задач в очередь, но и ввиду уменьшения числа возможных ошибок.



Высокая загрузка СКЦ кластера Tornado на интервале месяца



76.2 % узлов занято



38.6 % узлов занято

«РЕВОЛЮЦИОННЫЙ ПЕРЕДЕЛ» 2023- ПЕРЕХОД ОТ ЦИФРОВЫХ АВТОМАТОВ К ЯЗЫКОВЫМ **ТРАНСФОРМЕРАМ-ИНТЕРПРЕТАТОРАМ**

«В начале было Слово…»
Евангелия от Иоанна

«Все есть число»
Пифагор
570-490 до н.э.

Использование ИИ

Вычисления **чисел**

Вычисление **смыслов**

Эра
механических автоматов, исполняющих **один алгоритм, вычисления**

Эра
электронных автоматов, вычисляющих числа с помощью программ-алгоритмов

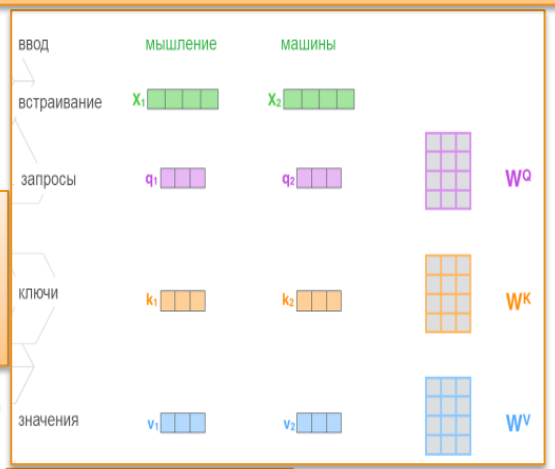
Эра
«интеллектуальных» вычислительных платформ «трансформер-интерпретатор»

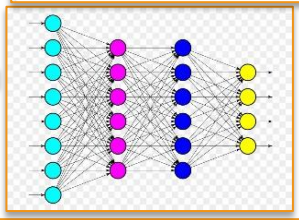
Алгоритм вычисления записанный **человеком** на языке «понятным» компьютерам

Алгоритм записанный на естественном языке, понятном человеку

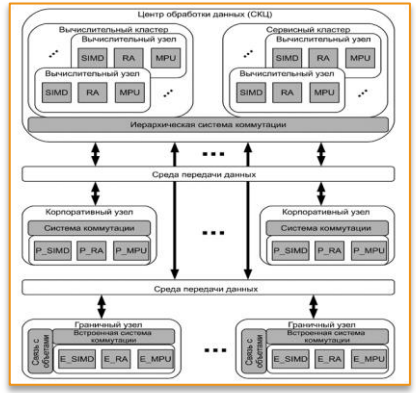
It from (Qu)bit
Дж. Уильер
1990
(все физические сущности в своей основе являются информационно-теоретическими)

X-входные данные и описание заданий





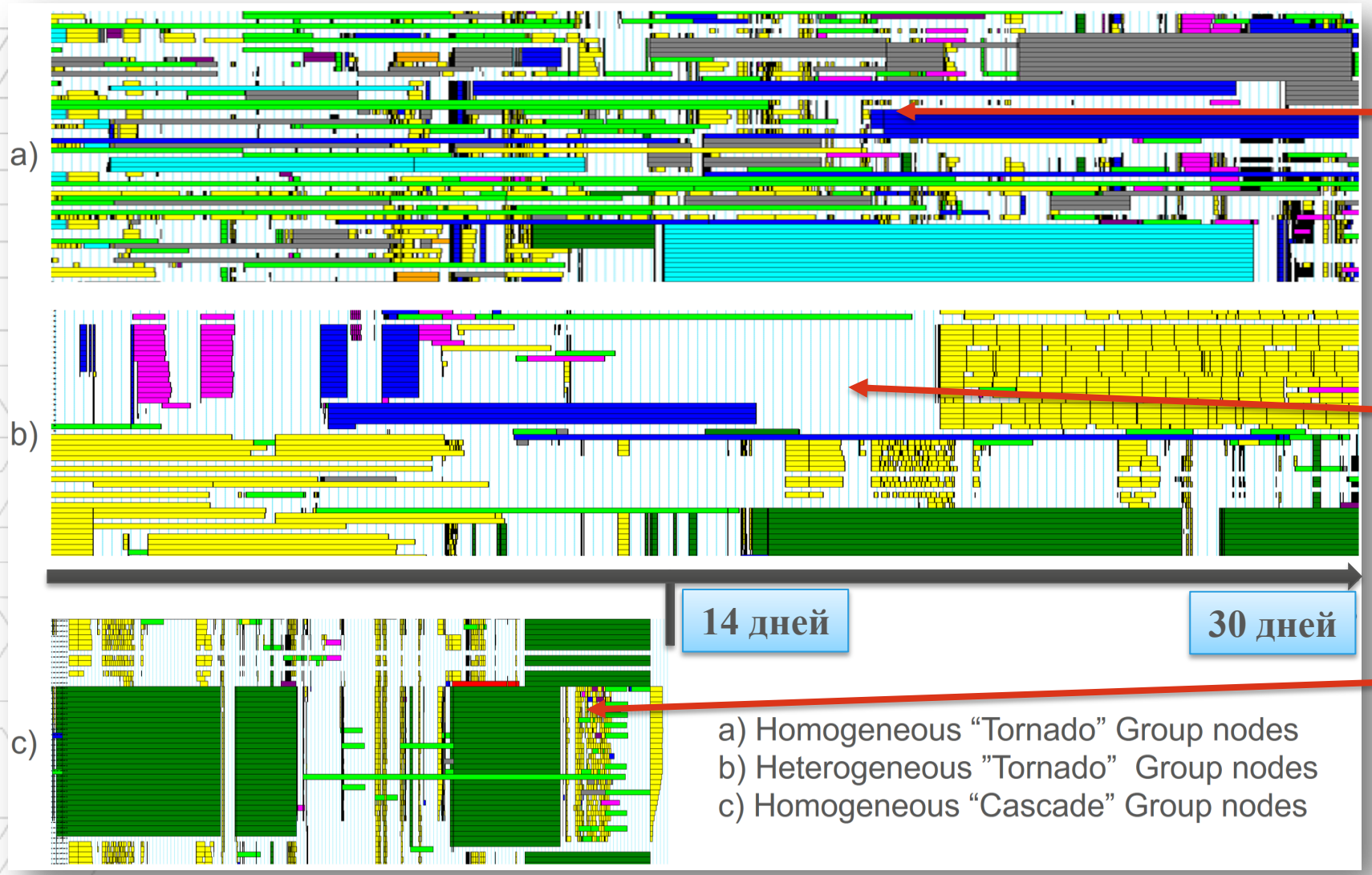
Y – выходные данные - результаты



Описание процессов на «языке данных» (case-based)

Описание процессов на «языке алгоритмов» (model-based)

Загрузка узлов кластера и резервы увеличения производительности СК



Не более 60% загруженности кластера

Не более 40% загруженности кластера

Не более 50% загруженности кластера

Потребляемые узлы, шт.

На графиках цветными блоками отмечены различные задачи и их потребление вычислительных узлов в штуках (по вертикали) и вычислительного времени (по горизонтали)

Точность предсказания времени приводит к повышению эффективности потребления ресурсов СК

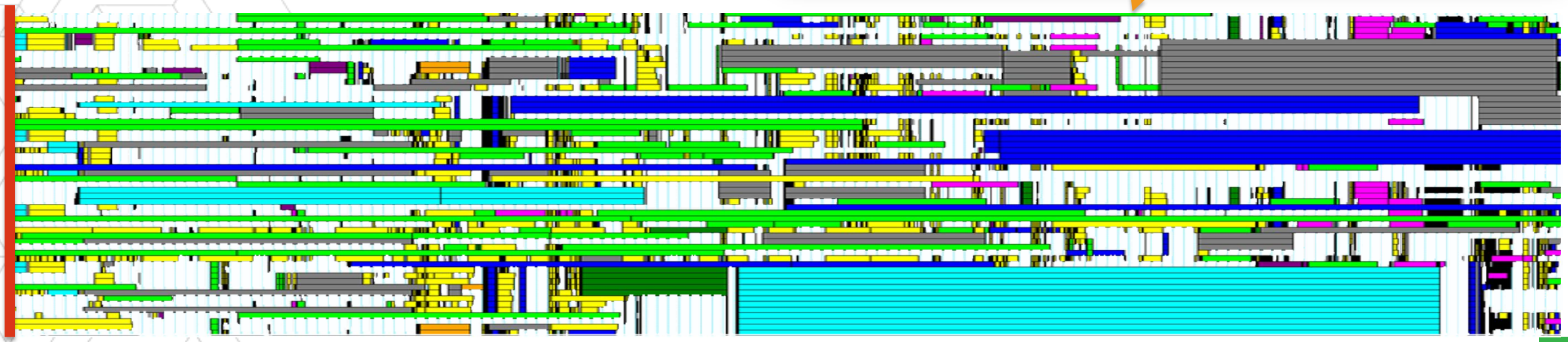
Вследствие **отсутствия точного предсказания пользователем** планировщик не может заполнить «пустоты» в очереди перед крупными задачами, для которых было запрошено большое количество узло-часов.

Обучение идет **успешно**, если удается **минимизировать «пропуски»** в очереди задач. Это приводит к сглаживанию кривой потребления эл. тока узлами и приближение ее к уровню максимума.

Физической интерпретацией повышения эффективности является максимизация интеграла (площади под графиком) функции потребления тока.

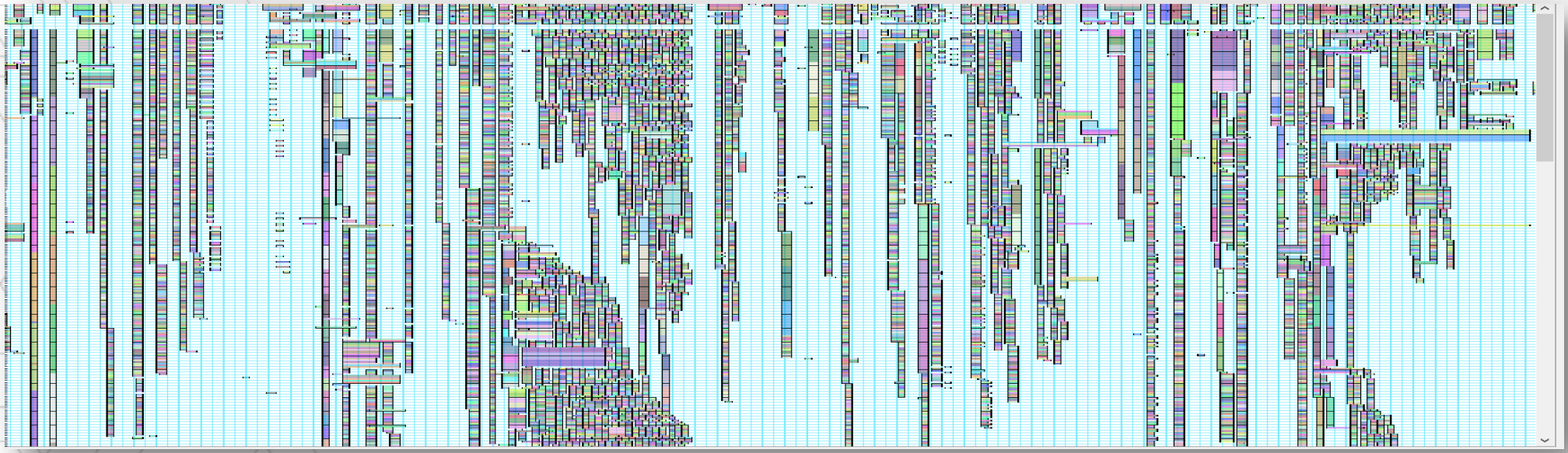
Использование ресурсов СК, близкое к пиковой производительности – цель алгоритмов предсказания времени

Резервы повышения реальной производительности СК, в среднем не менее 40%



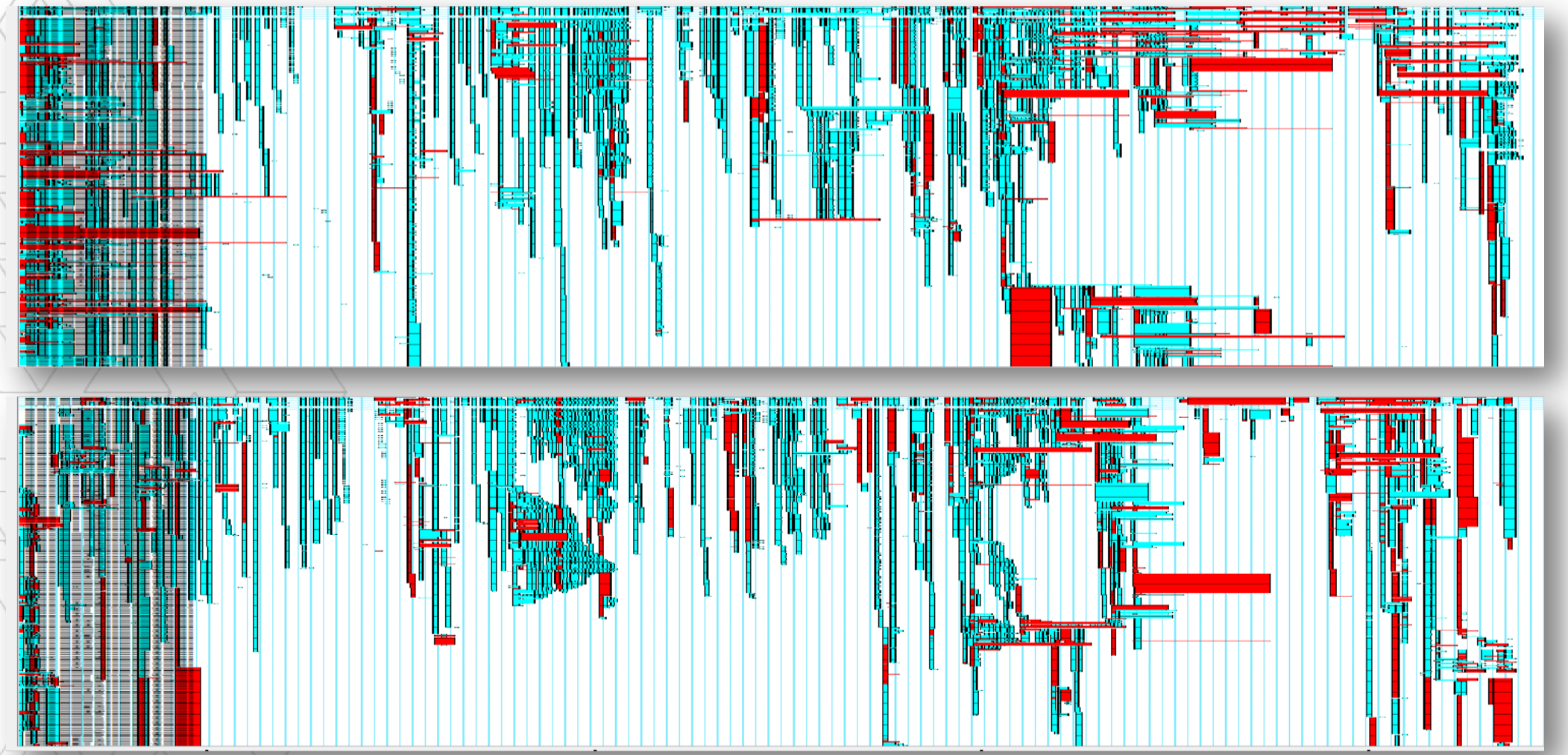
Сравнение загрузки узлов кластера

Без применения ИИ: между задачами замечаются частые пустые зазоры, **менее эффективное использование ресурсов**



С применением ИИ: задачи расположены «плотнее» друг к другу, **оставляя ресурсы для других задач**

Доли задач, снятых с вычисления диспетчером



Без применения ИИ: ***1.1%***

Красным отмечены неуспешные задачи

С применением ИИ: ***1.8%***

Важно отметить, что проблематично отследить понижение этого критерия, т.к. если выделить в прошлом неуспешной задаче больше времени, то трудно оценить помогло ей это или нет, т.к. неизвестно ее реальное времени вычисления (проблема симуляции). По этой причине удается только зафиксировать повышение неуспешных задач.