

Молодежная научная школа

Высокопроизводительные вычисления, оптимизация и приложения

Искусство эффективного управления суперкомпьютерной системой

Воеводин Вад.В.

*Научно-исследовательский вычислительный центр
МГУ имени М.В. Ломоносова*

9 сентября 2017 г., Нижний Новгород

Современный суперкомпьютер



Современный суперкомпьютер

Пользователи
Сисадмины
Руководство

Проекты

Лицензии

Организации

Разделы

Компоненты
ПО

Компоненты
аппаратуры

Приложения

Очереди

Статусы

Пользователи

Квоты

Задачи



Суперкомпьютер «Ломоносов» (№2 в России)

Пользователи
Сисадмины
Руководство

Проекты

600

Лицензии

100

Организации

350

Разделы

10

Компоненты
ПО

100

Компоненты
аппаратуры

100 000

Приложения

600

Очереди

15

Статусы

20

Пользователи

3 000

Квоты

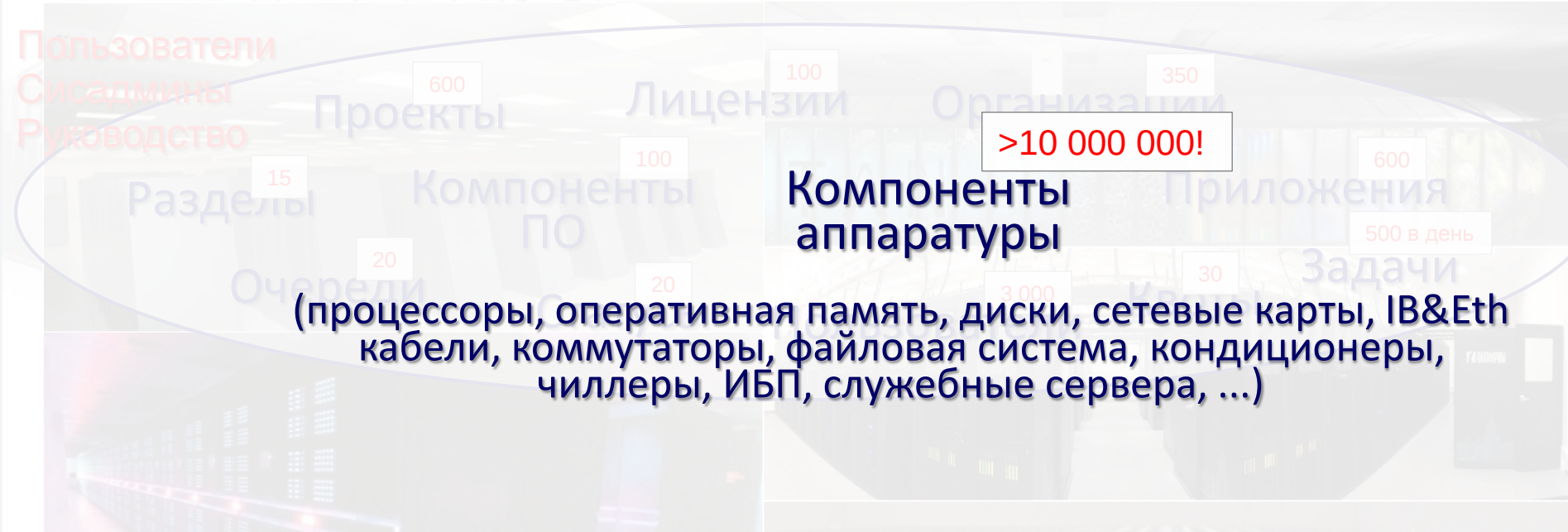
30

Задачи

1000 в день



Суперкомпьютер №1 в мире («TaihuLight», Kumaй)



Такая сложная архитектура суперкомпьютеров приводит к тому, что у нас нет полного понимания, как они работают!

А это приводит к **очень низкой эффективности** работы суперкомпьютера.

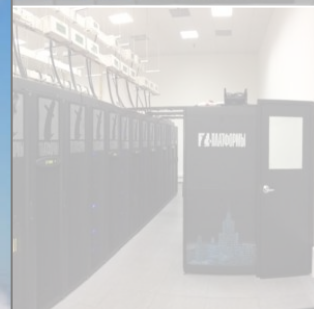
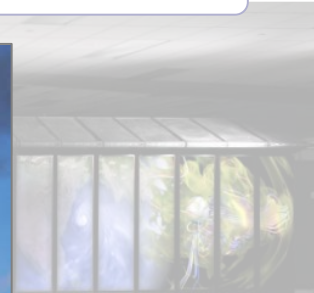
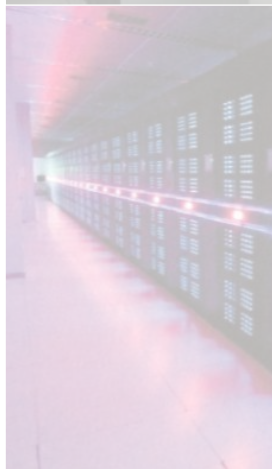
Эффективность суперкомпьютерных систем

На примере системы 1 Пфлоп/с

$1 \text{ ПФлоп} * 60 \text{ сек} * 60 \text{ мин} * 24 \text{ ч} * 365 \text{ дней} = 31,5 \text{ ЗеттаФлоп } (10^{21}) \text{ в год}$

Пиковая производительность

А на практике...



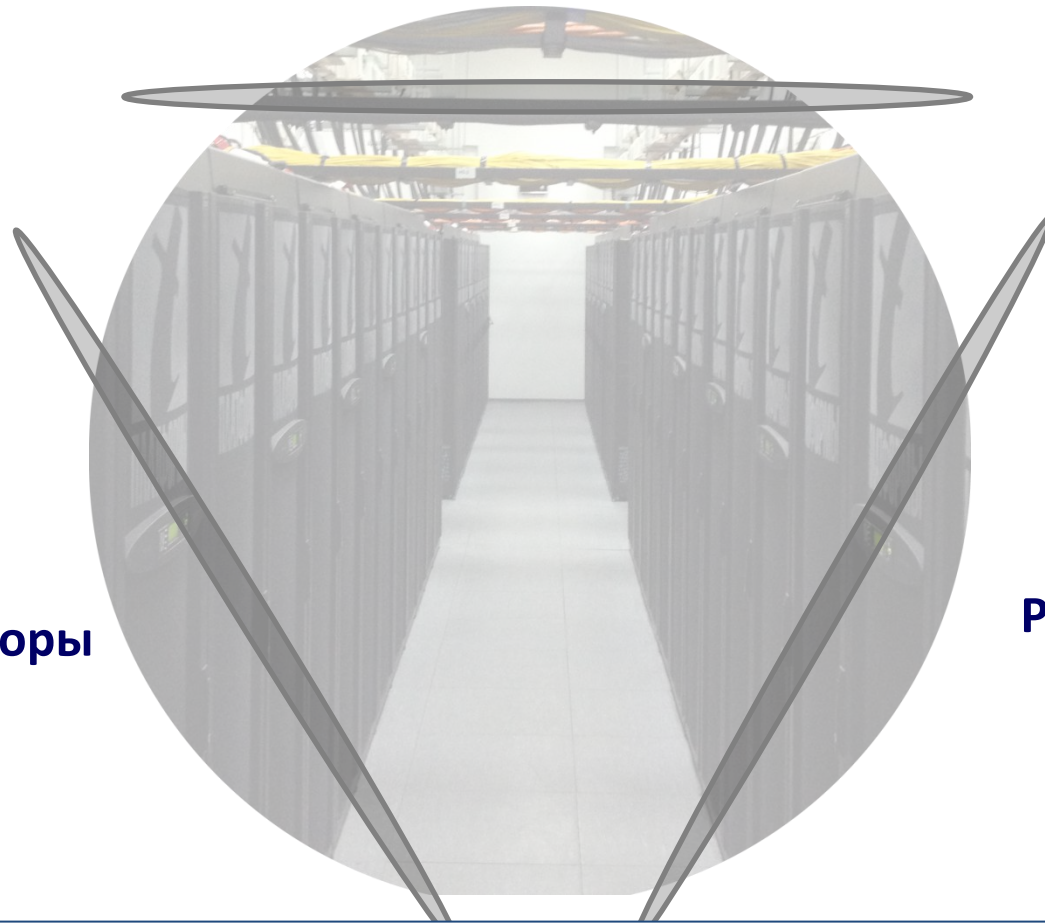
Эффективность суперкомпьютерных систем



Средняя производительность (одно ядро) суперкомпьютера “Чебышев” за 3 дня

Кому нужна эффективность?

Пользователи



Администраторы

Руководство

Пользователи, Администраторы, Руководство: преследуют разные цели, имеют разные права, принимают разные решения.

А что такое эффективность?

Пользователи – как быстро можно решить задачу

Эффективность их приложений

Администраторы – насколько эффективно используются ресурсы

Эффективность суперкомпьютера

Руководство – какова эффективность суперкомпьютерного центра,
финансовый аспект

Эффективность суперкомпьютерного центра

Пользователи, Администраторы, Руководство: преследуют разные цели, имеют разные права, принимают разные решения.

Эффективность суперкомпьютерного центра (системный уровень)



Возможных
причин потери
эффективности
очень много...

Эффективность суперкомпьютерного центра (уровень всего СКЦ)

Проекты, Пользователи, Приложения

Задачи в Очередях

Поведения Задач

Поток Задач

Стек ПО

Вычислительные компоненты

Инженерная инфраструктура

Возможных
причин потери
эффективности
очень много...

Низкая эффективность может обойтись очень дорого...

Полный контроль: стоимость задержки

Нам нужно сохранить контроль над суперкомпьютерами!

Плата за простой вычислительных ресурсов постоянно увеличивается!

Некоторые факты:

Один день суперкомпьютера «Ломоносов» (МГУ) стоит \$20 000

Один день суперкомпьютера «Titan» (ORNL, №4 в мире) стоит \$300 000

Подобная ситуация везде.

Полный контроль: стоимость задержки

Нам нужно сохранить контроль над суперкомпьютерами!

Плата за простой вычислительных ресурсов постоянно увеличивается!

Некоторые факты:

Один день суперкомпьютера «Ломоносов» (МГУ) стоит \$20 000

Один день суперкомпьютера «Titan» (ORNL, №4 в мире) стоит \$300 000

Подобная ситуация везде.

Суперкомпьютер «Ломоносов»:

Если планировщик повис, половина суперкомпьютера будет простаивать уже через 2-3 часа.

Что необходимо для эффективного управления суперкомпьютерным центром?

Пользователи
Сисадмины
Руководство

Проекты

Лицензии

Организации

Разделы

Компоненты
ПО

Компоненты
аппаратуры

Приложения

Очереди

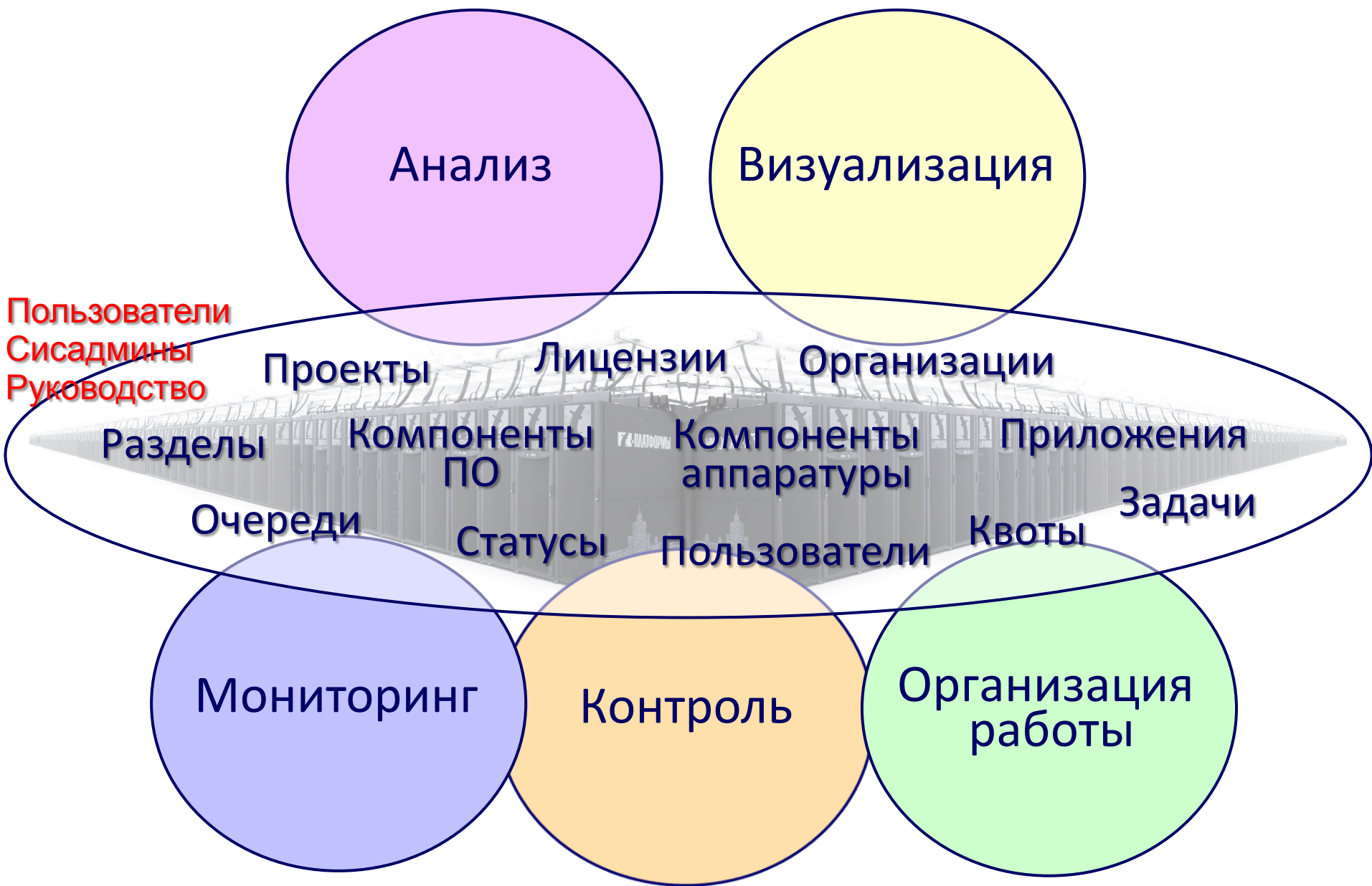
Статусы

Пользователи

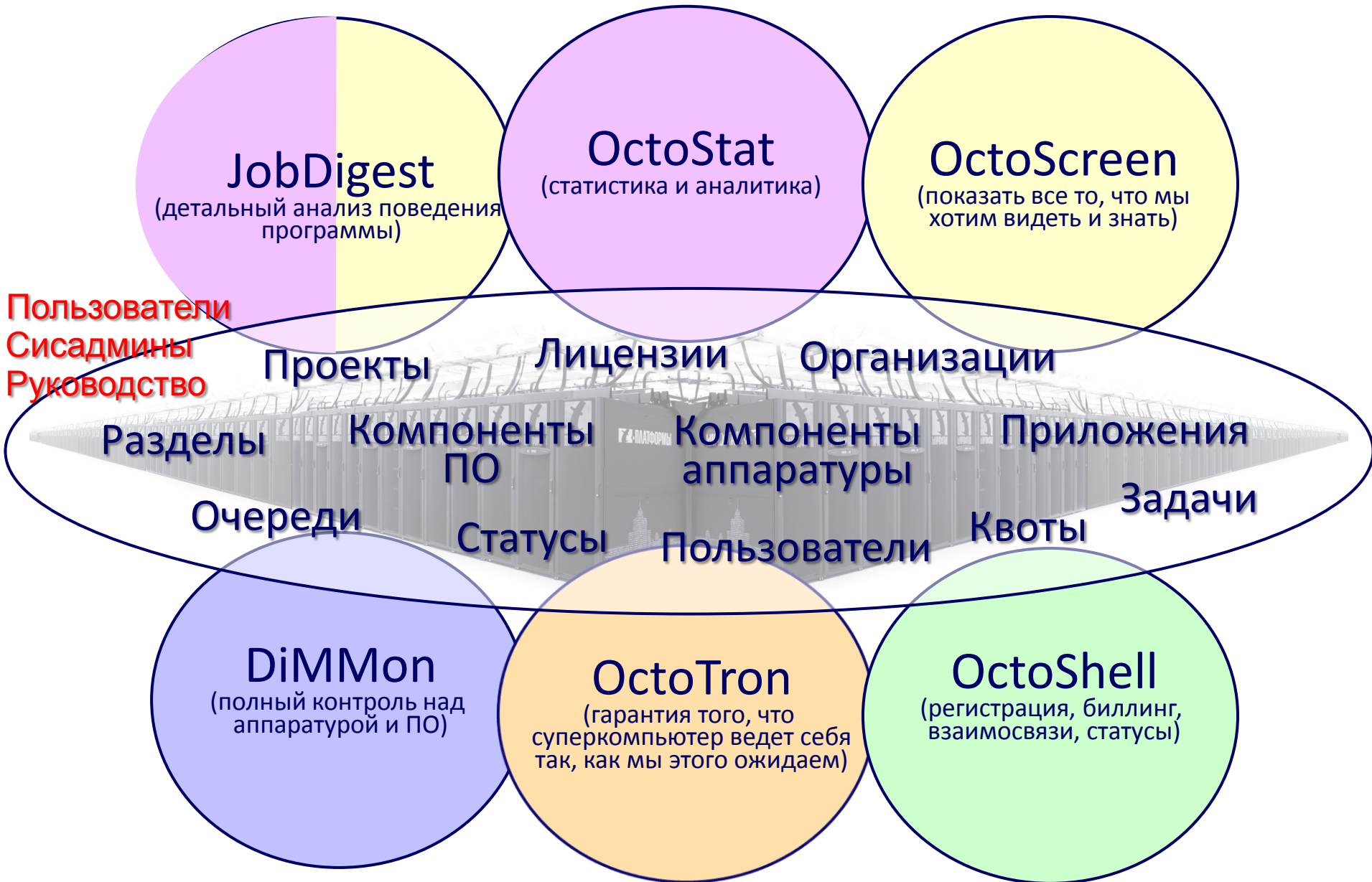
Квоты

Задачи

Что необходимо для эффективного управления суперкомпьютерным центром?



Каков наш подход?



Мониторинг

JobDigest

(детальный анализ поведения
программы)

OctoStat

(статистика и аналитика)

OctoScreen

(показать все то, что мы
хотим видеть и знать)

Цель:

Контролировать все аспекты
текущего состояния
суперкомпьютерной системы.

Организации

Компоненты
аппаратуры

Приложения

Задачи

Квоты

Пользователи

Статусы

Очереди

DiMMon

(полный контроль над
аппаратурой и ПО)

OctoTron

(гарантия того, что
суперкомпьютер ведет себя
так, как мы этого ожидаем)

OctoShell

(регистрация, биллинг,
взаимосвязи, статусы)

Пользователи
Сисадмины
Руководство

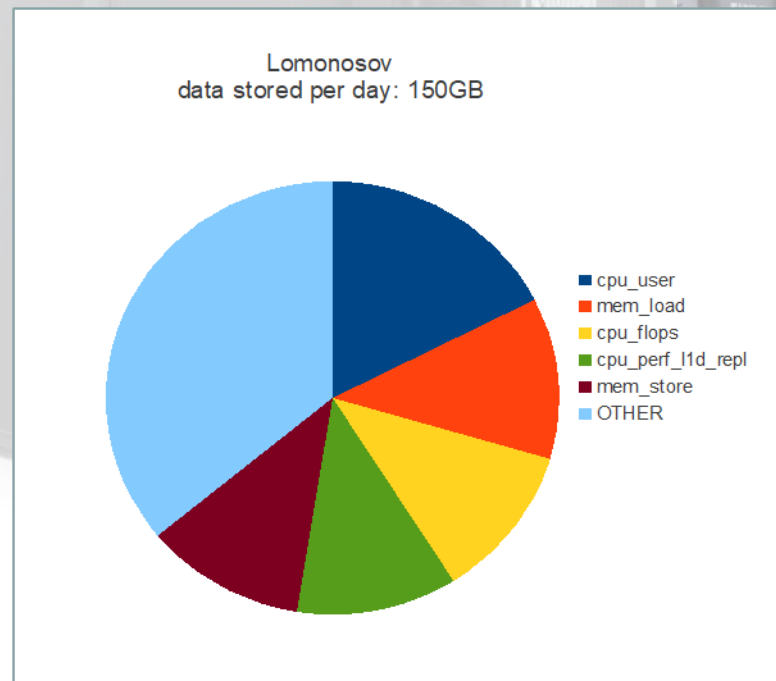
Разделы

Требования к системе мониторинга

Суперкомпьютеры будущего предъявляют серьезные требования:

- Масштабируемость: миллионы вычислительных узлов, десятки датчиков на узле.
- Низкие накладные расходы: CPU, диски, сеть (<1%).
- Простота реконфигурации и расширения функциональности.
- Частота: несколько секунд и чаще.
- Переносимость.
- Различные режимы работы.
- ...

Важно:
необходимо учитывать сложности с BigData...



Система мониторинга DiMMon

Умный подход к мониторингу:

- **анализ «на лету»:** вся важная информация должна быть извлечена до записи в БД (это упростит дальнейшую обработку и уменьшит хранимый объем данных);
- **анализ «на месте»:** данные мониторинга должны обрабатываться там же, где они были получены (сначала – обработка данных, потом – их пересылка);
- **динамическая реконфигурация:** система мониторинга должна уметь в динамике менять свою архитектуру в зависимости от текущей загрузки суперкомпьютера и текущих целей анализа.

Анализ и визуализация



Основные характеристики приложений

Динамические и интегральные характеристики:

- Загрузка процессора
 - *user load*
 - *system load, idle, iowait,...*
 - Число флоп/с
 - Работа с памятью
 - Число промахов в L1/L3 кэш-память
 - Интенсивность чтения/записи в память
 - Работа с сетью:
 - Интенсивность приема/передачи данных по Infiniband
 - Интенсивность приема/передачи пакетов по Infiniband
 - *LoadAverage*
 - Работа с GPU
- Дополнительные сведения от СУПЗ (ID, владелец, статус, очередь, команда запуска, время постановки в очередь/запуска/завершения, ...)**

Отчет JobDigest.

Анализ эффективности отдельных приложений

job id	1314965	cpu_flops	240582000.0	cpu_flops	2086480000.0	cpu_flops	36.07
account		cpu_idle	47.5283	cpu_idle	100.0	cpu_idle	0.0
command	ffxp	cpu_iowait	0.0178268	cpu_iowait	86.0	cpu_iowait	0.0
workdir	unknown	cpu_irq	0.000436416	cpu_irq	1.0	cpu_irq	0.0
slurm-out	unknown/slurm-1314965.out	cpu_nice	0.0	cpu_nice	0.0	cpu_nice	0.0
partition	gpu	cpu_perf_l1d_repl	2661700.0	cpu_perf_l1d_repl	28549900.0	cpu_perf_l1d_repl	35.91
timelimit(min)	4320	cpu_soft_irq	0.32768	cpu_soft_irq	6.0	cpu_soft_irq	0.0
submit	2016-07-09 01:18:40 (1468016320)	cpu_system	1.21602	cpu_system	100.0	cpu_system	0.0
start	2016-07-12 22:45:07 (1468352707)	cpu_user	28.4004	cpu_user	100.0	cpu_user	0.0
end	2016-07-15 22:46:10 (1468611970)	gpu_load	0.0	gpu_load	0.0	gpu_load	0.0
duration (minutes)	4321	gpu_mem_load	0.0	gpu_mem_load	0.0	gpu_mem_load	0.0
number of cores	168	gpu_mem_usage	10658800.0	gpu_mem_usage	10858500.0	gpu_mem_usage	10334200.0
number of nodes	21	ib_rcv_data	4700.33	ib_rcv_data	16399600.0	ib_rcv_data	0.0
number of cores * hours	12098.94	ib_rcv_pckts	15.3161	ib_rcv_pckts	608.17	ib_rcv_pckts	0.0
node6-153-[02-03],node6-154-[06-07],node6-161-[03-05],node6-162-11,node6-164-16,node6-165-[01-03,15],node6-170-[07-12],node6-172-[12-13]		ib_xmit_data	1879.58	ib_xmit_data	129574.0	ib_xmit_data	0.0
		ib_xmit_pckts	8.23421	ib_xmit_pckts	600.87	ib_xmit_pckts	0.0
		llc_miss	102829.0	llc_miss	1668000.0	llc_miss	4.69
		loadavg	4.8232	loadavg	14.88	loadavg	0.0
		mem_load	377220000.0	mem_load	1639010000.0	mem_load	8287.4
		mem_store	149185000.0	mem_store	634774000.0	mem_store	2581.3

auto_low_gpu_load ✖

auto_avg_CPU_idle TOO HIGH ✖

auto_avg_IB_packages_num_LOW ✖

auto_job_start_script_CUSTOM ✖

partition_GPU ✖

status_TIMEOUT ✖

auto_avg_IB_speed_LOW ✖

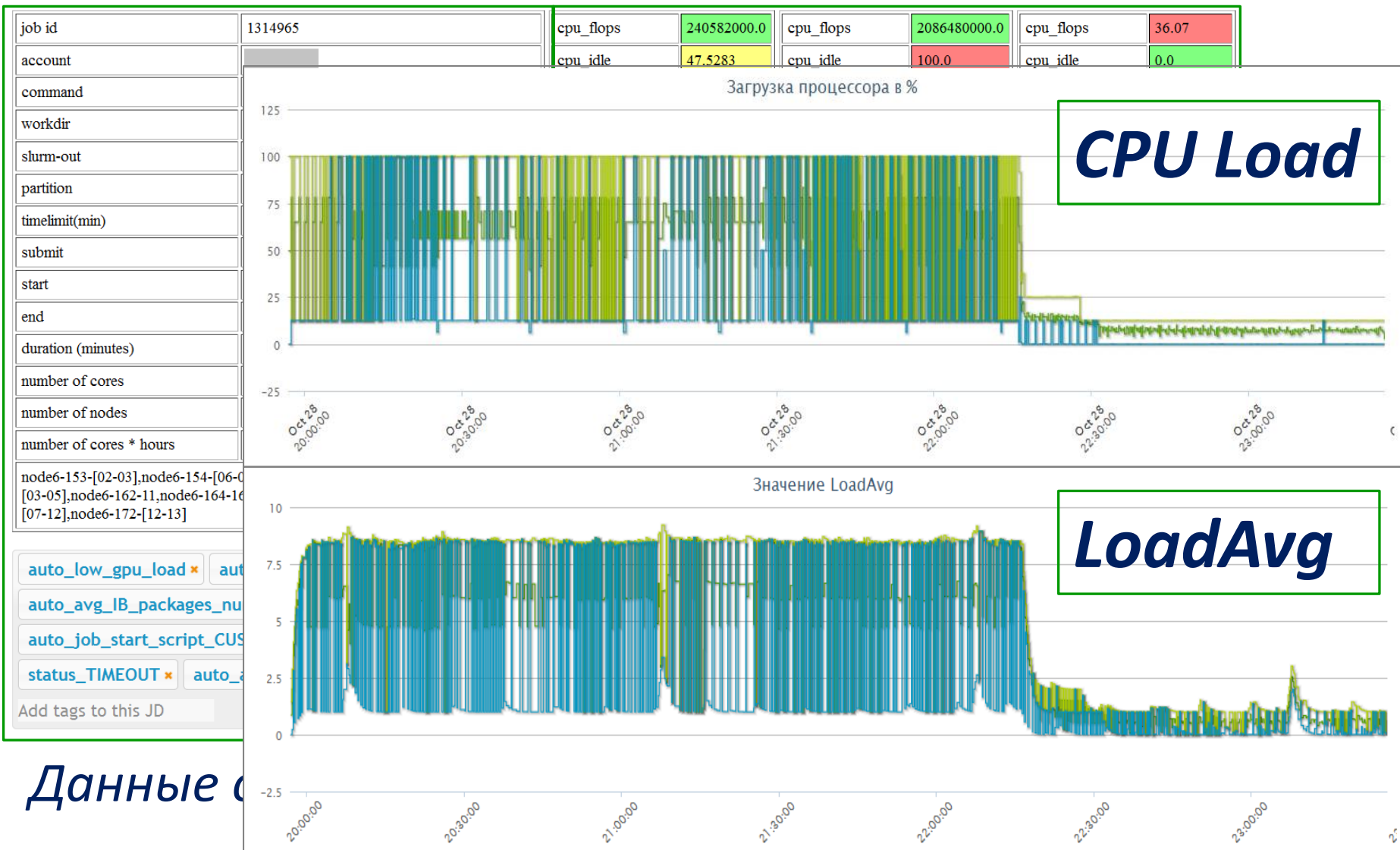
Add tags to this JD

Интегральные значения
по задаче

Данные от СУПЗ

Отчет JobDigest.

Анализ эффективности отдельных приложений



Динамика может рассказать многое...

Информация о задаче № regular-1404302831-496016 пользователя

expand all

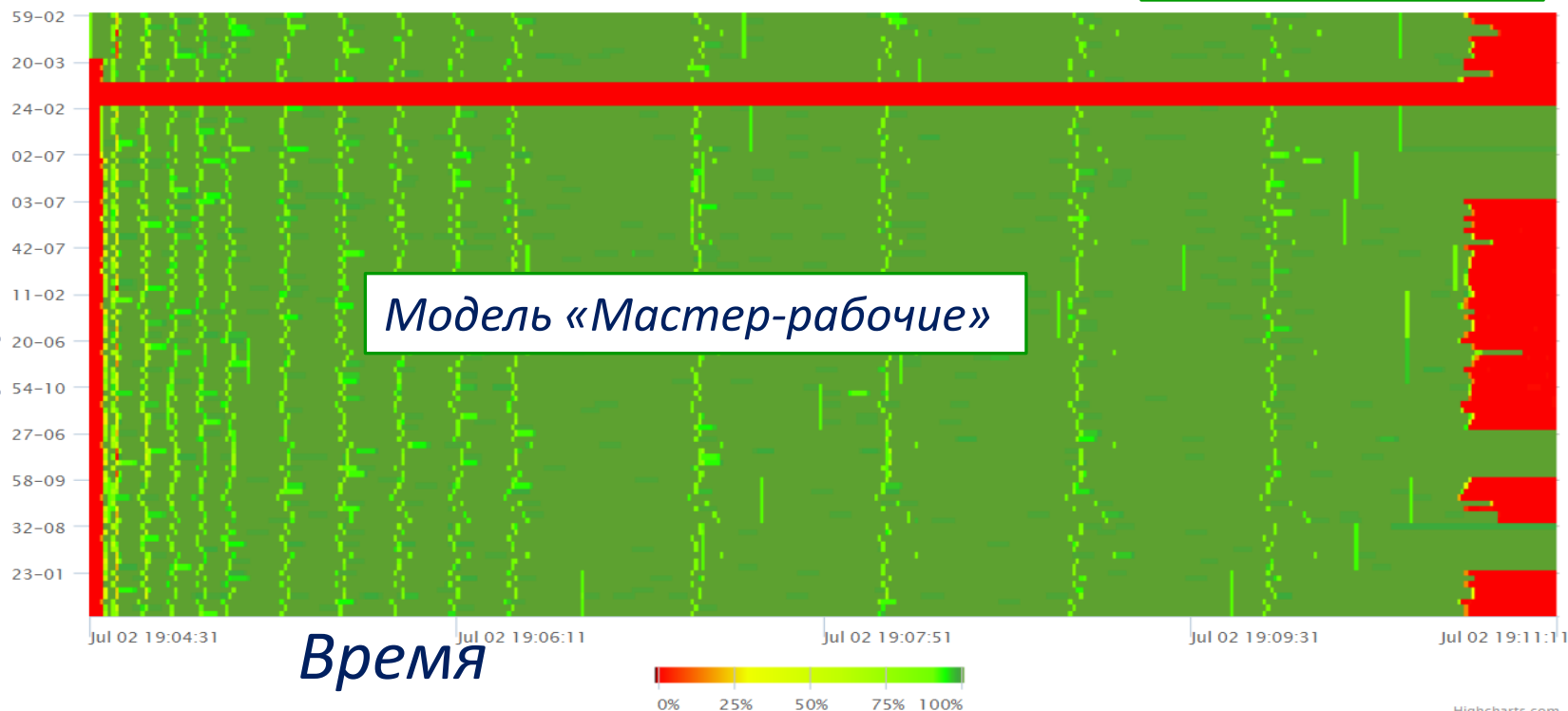
Информация о запуске

Строка запуска: ./2dxy_m01_p08.ex
Число ядер: 100
Номера узлов: node-02-07,node-03-07,node-11-02,node-20-03,node-20-06,node-23-01,node-24-02,node-27-06,node-32-08,node-42-07,node-54-10,node-58-09,node-59-02
Дата постановки в очередь: Wed, 02 Jul 2014 16:07:11 (1404302831)
Дата запуска: Wed, 02 Jul 2014 19:04:29 (1404313469)
Дата окончания счета: Wed, 02 Jul 2014 19:46:13 (1404315973)
Время счета: 0 days 0 hours 41 minutes 44 seconds
Время ожидания: 0 days 2 hours 57 minutes 18 seconds

CPU Load

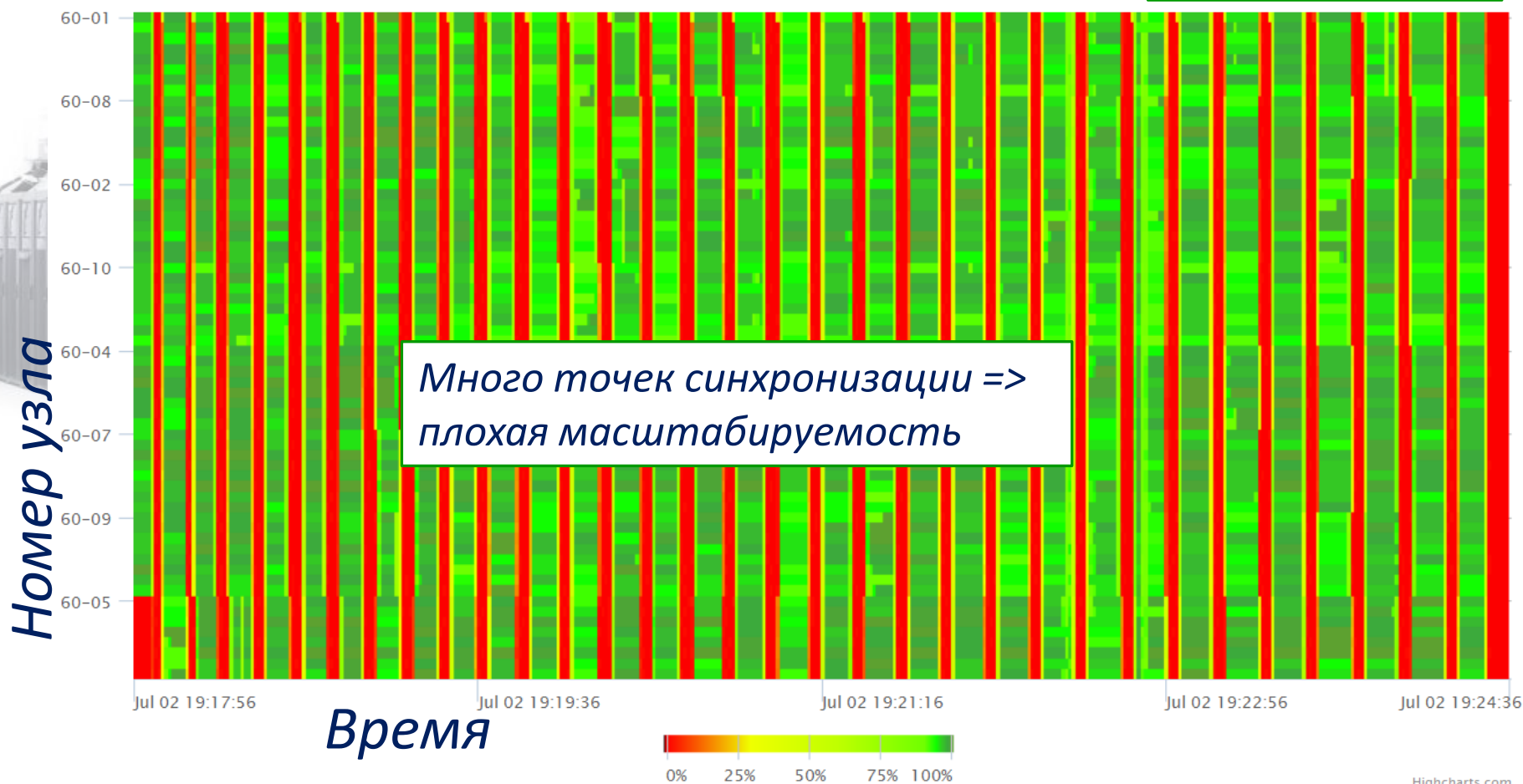
Модель «Мастер-работчие»

Номер узла



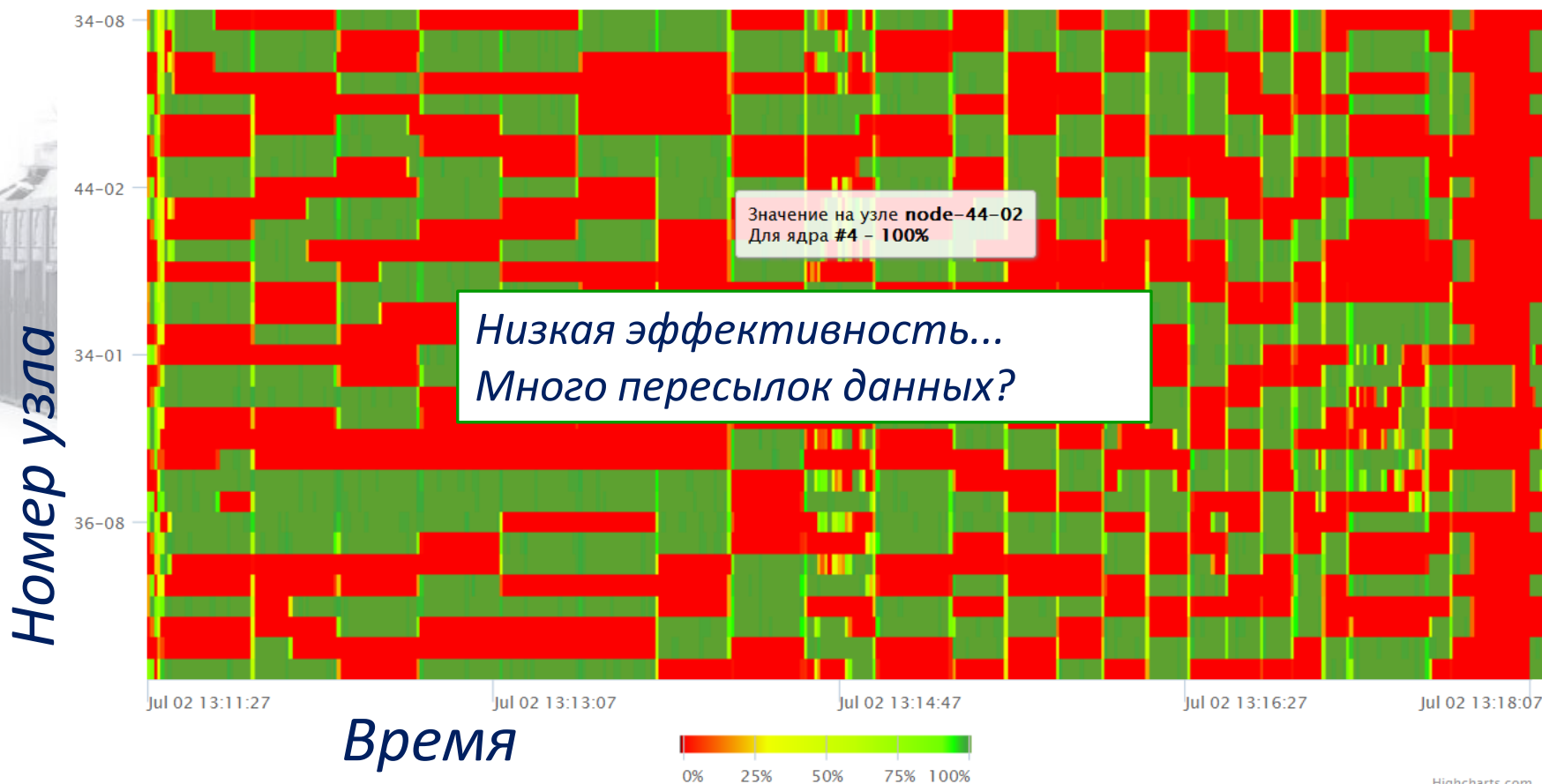
Динамика может рассказать многое...

CPU Load



Динамика может рассказать многое...

CPU Load



Анализ

JobDigest

(детальный анализ поведения программы)

OctoStat

(статистика и аналитика)

OctoScreen

(показать все то, что мы хотим видеть и знать)

Цель:

Умный анализ и сбор статистики по общему потоку задач

Организации

Приложения

Задачи

Квоты

Пользователи

Статусы

Очереди

DiMMon

(полный контроль над аппаратурой и ПО)

OctoTron

(гарантия того, что суперкомпьютер ведет себя так, как мы этого ожидаем)

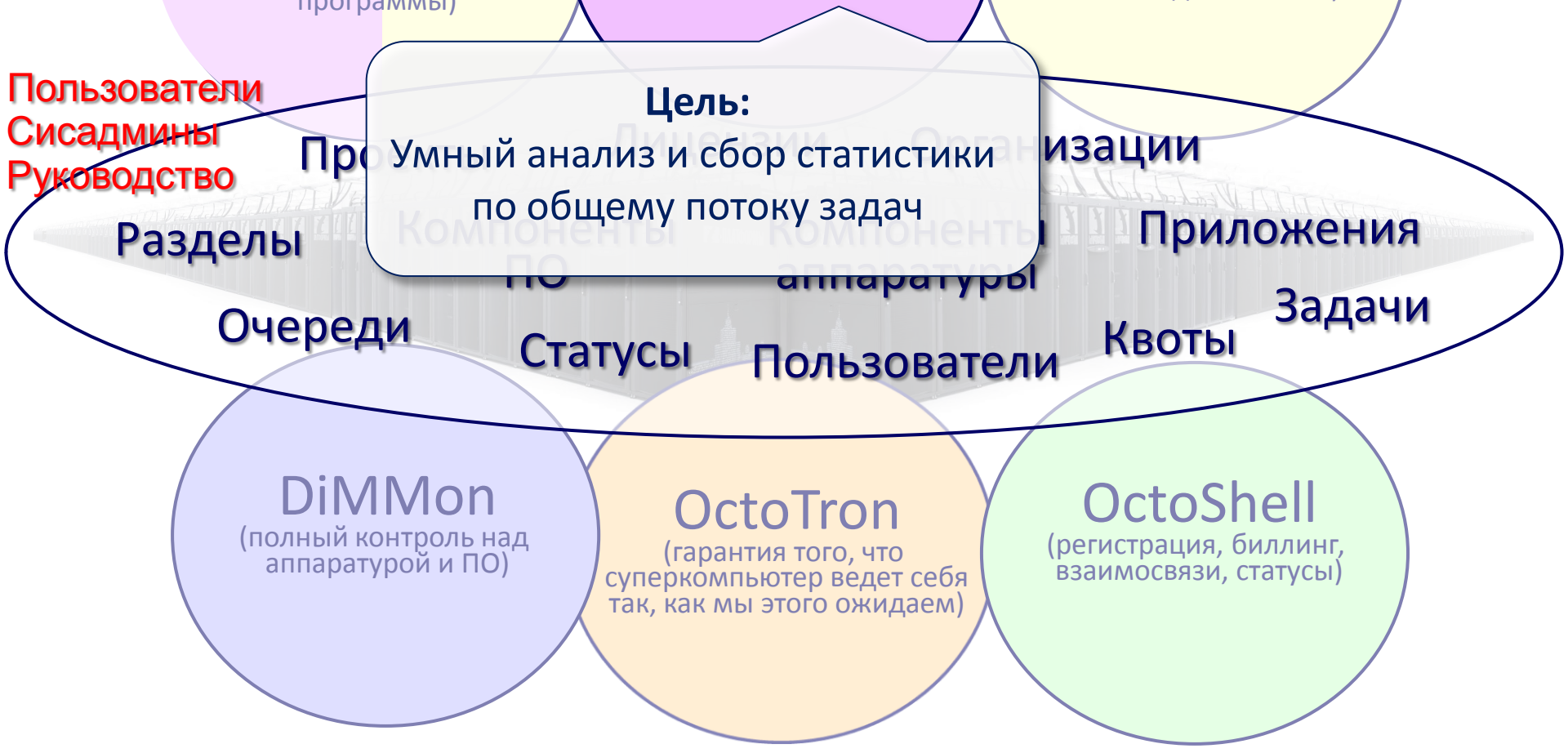
OctoShell

(регистрация, биллинг, взаимосвязи, статусы)

Пользователи
Сисадмины
Руководство

Проект

Разделы



Изучение общего потока задач

- Эффективность работы суперкомпьютеров низка...
(JobDigest – хорошо, но только когда мы уже знаем, что задачу нужно внимательно изучить)



- Но пользователи зачастую об этом не подозревают
(администраторы тоже не всегда в курсе...)



- Необходимо изучать эффективность общего потока задач
(важно учитывать разные аспекты эффективности)

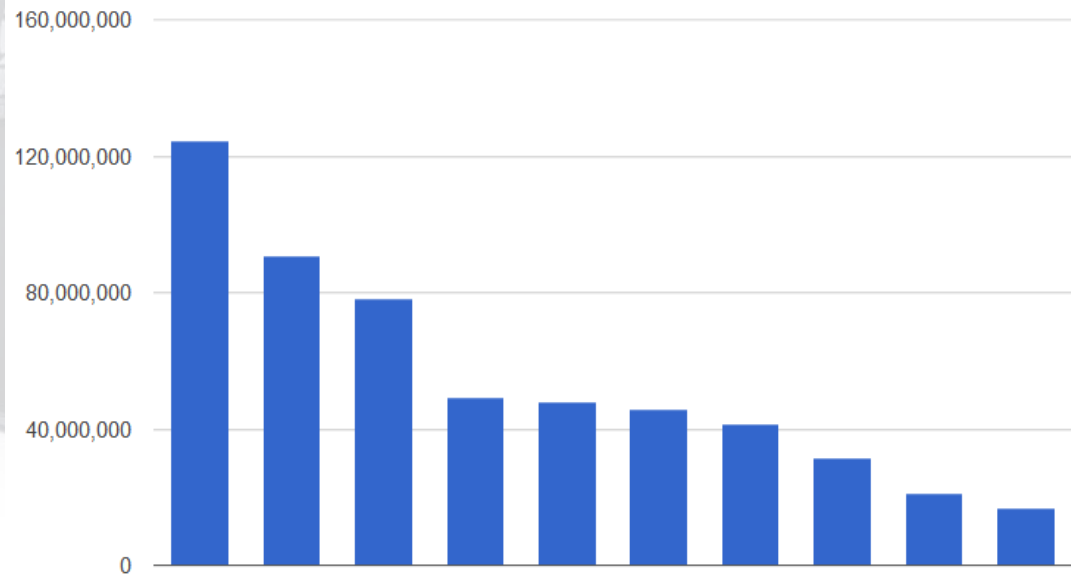


• OctoStat:

- Ежедневная статистика
- Поиск аномалий

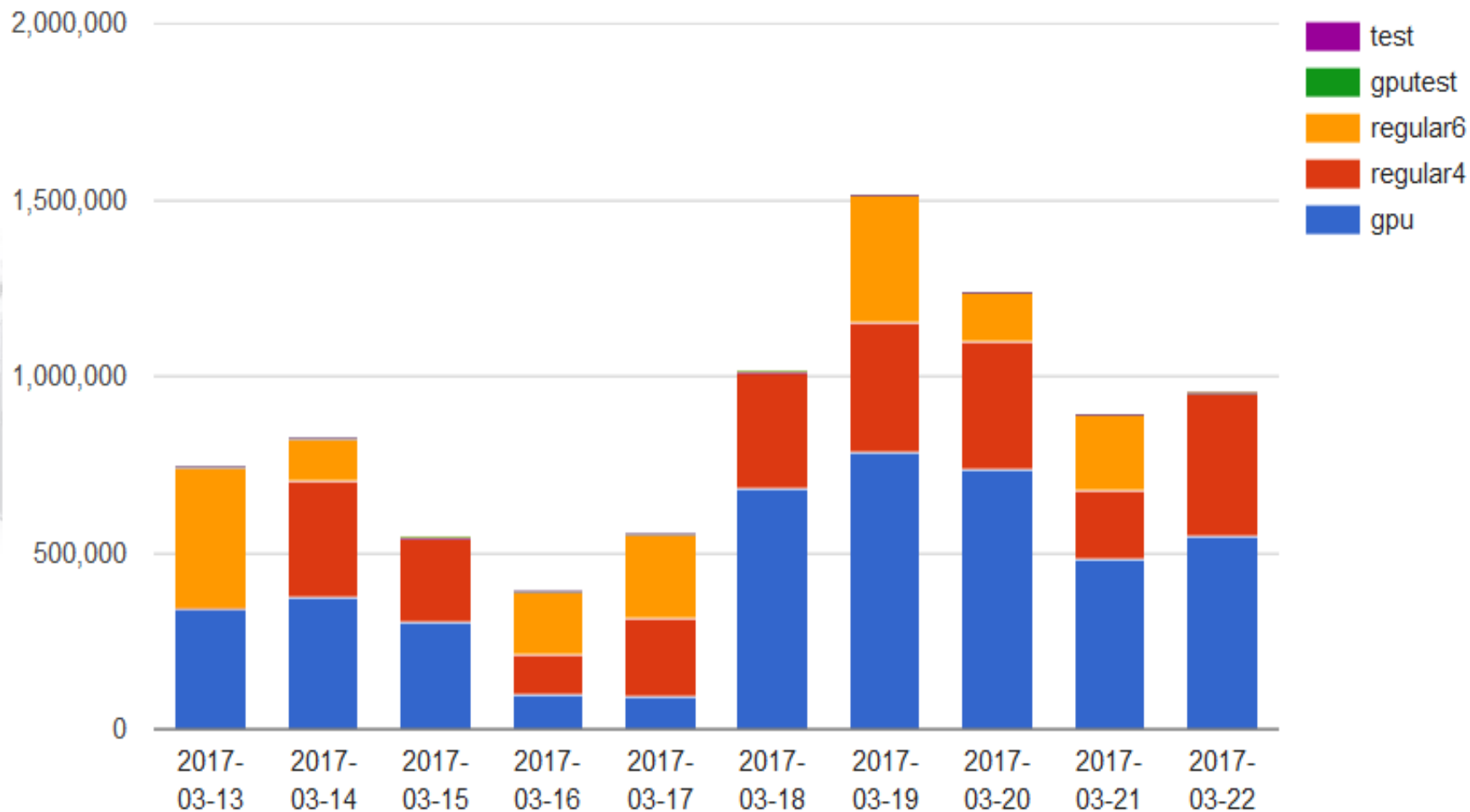
OctoStat. Ежедневная статистика

- Цель – анализ использования ресурсов
- Изучаем эффективность:
 - общей загрузки суперкомпьютера
 - стратегий планирования очередей
 - политик и квот



Топ пользователей по затраченным процессорно-часам

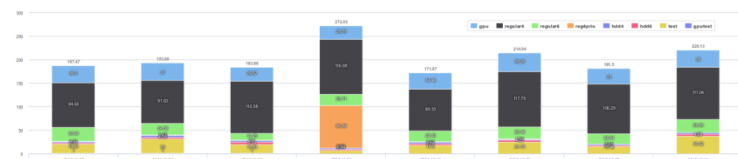
Ежедневная статистика. Примеры



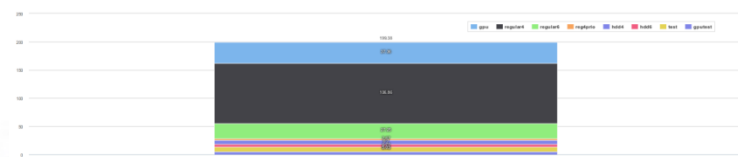
Статистика по времени ожидания в очередях

Ежедневная статистика. Примеры

3.1 Average running task count on Lomonosov by days



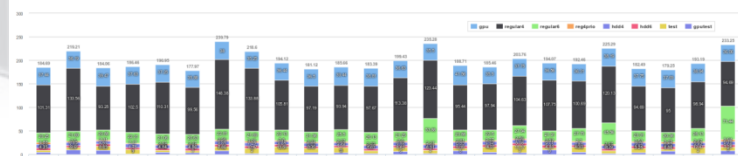
3.2 Average running task count on Lomonosov by months



3.3 Average running task count on Lomonosov by weekdays



3.4 Average running task count on Lomonosov by hours



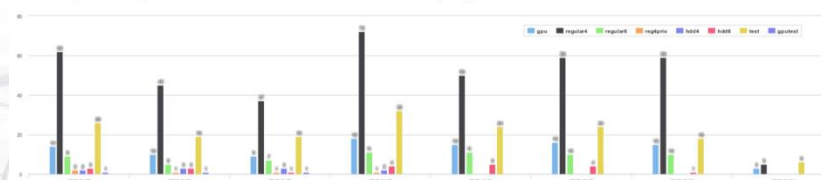
3.13 Distribution of user tasks count by runtime(x axis is in minutes), only tasks shorter than 1 hour



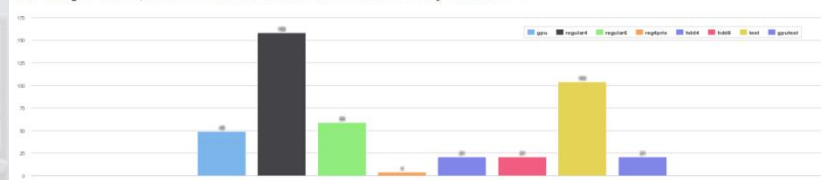
3.14 Distribution of user tasks count by runtime(x axis is in hours), only tasks longer than 1 hour



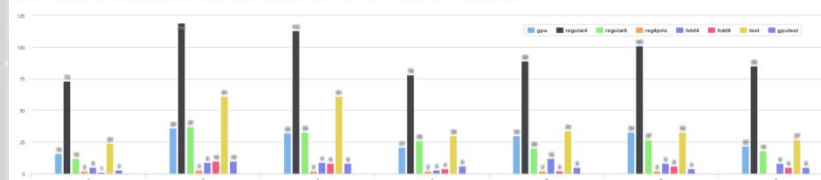
4.1 Unique users, who submitted a task on Lomonosov by days



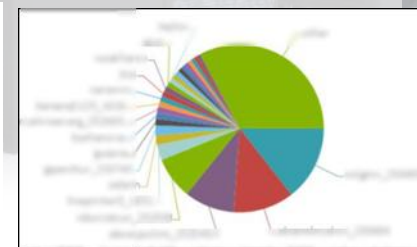
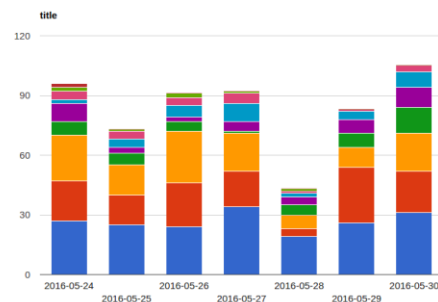
4.2 Unique users, who submitted a task on Lomonosov by months



4.3 Unique users, who submitted a task on Lomonosov by weekdays



Unique users, who completed a task in every queue



Ежедневная статистика. Классы задач

	Вчера	За прошлую неделю	За прошлый месяц	С начала года
Всего задач	778	4 738	23 892	202 892
«Идеальные» (loadavg оптимален и высокая активность по сети)	108	157	1 815	15 192
«Необщительные» (задача на неск. узлов, но почти нет активности по сети)	24	139	330	3 424
«Последовательные» (задачи на 1 ядро)	6	39	141	1 179
«1-узловые» (задачи на 1 узел)	47	881	2 774	19 956
«Подозрительные» (loadavg и загрузка процессора очень малы)	3	84	129	138

Ломоносов-2, статистика от 20 июля 2017 года

OctoStat. Поиск аномалий

- *Цель – обнаружить в общем потоке задачи с необычным поведением*
- *Изучаем эффективность приложений*
- *Критерии необычного поведения:*
 - *точно не определены*
 - *будут отличаться для различных суперкомпьютеров*
- *Решение – применение методов машинного обучения*

Классификация

- *Основа – данные системного мониторинга*
 - *Те же данные, что и в JobDigest*
 - *По сути – временные ряды по каждой характеристике по каждой задаче*



- *Каждую задачу хотим отнести к одному из 3 классов:*
 - *Нормальная*
 - *Аномальная*
 - *Подозрительная*

Выбор способа классификации задач

- 1. Изучать интегральные характеристики по всей задаче*
- 2. Выделять и анализировать интервалы*
- 3. Анализировать непосредственно временные ряды*



Выбор способа классификации

ib_rev_data	4700.33	ib_rev_data	16399600.0	ib_rev_data	0.0
ib_rev_pkts	15.3161	ib_rev_pkts	608.17	ib_rev_pkts	0.0
ib_xmit_data	1879.58	ib_xmit_data	129574.0	ib_xmit_data	0.0
ib_xmit_pkts	8.23421	ib_xmit_pkts	600.87	ib_xmit_pkts	0.0

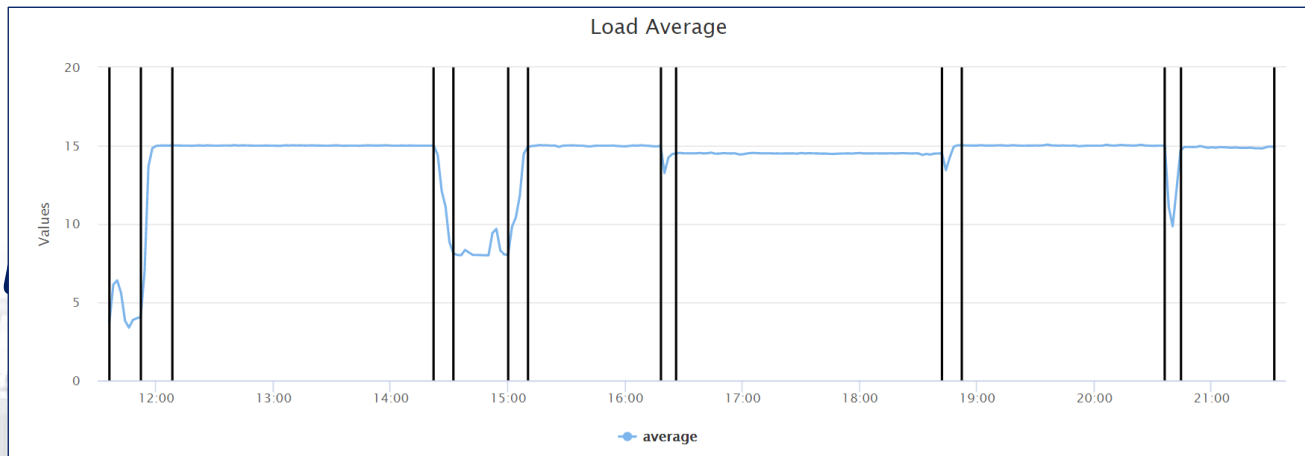
1. Изучать интегральные характеристики по всей задаче

- Пробовали раньше
- Точность неплохая
- Но: недостаточно информации о причинах аномального поведения!

2. Выделять и анализировать интервалы

3. Анализировать непосредственно временные ряды

Выбор способа классификации



1. Изучать

2. Выделять и анализировать интервалы

- Разбиваем задачу на временные интервалы
- В рамках интервала поведение может быть точно описано интегральными характеристиками
- На основе классифицированных интервалов принимается решение о всей задаче

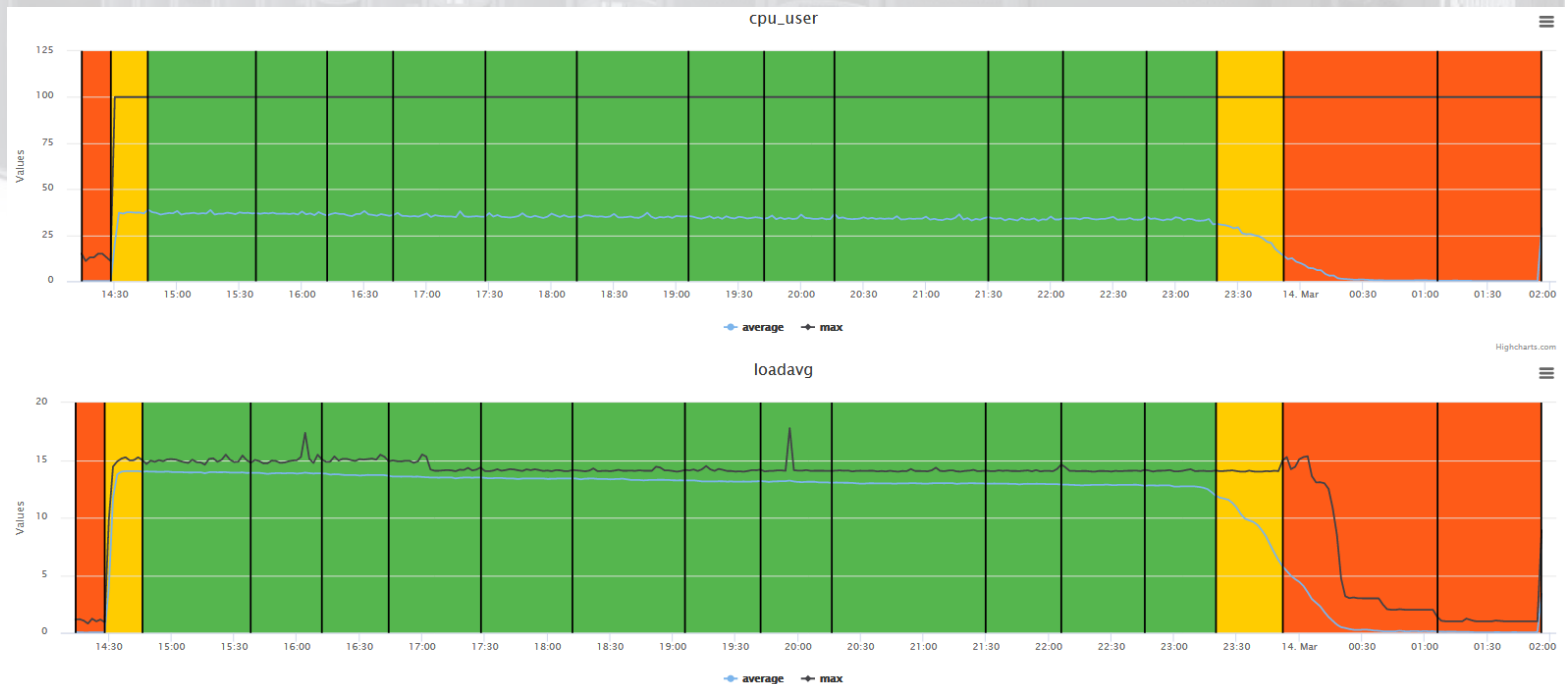
3. Анализировать непосредственно временные ряды

Выбор способа классификации

1. Изучать интегральные характеристики по всей задаче
2. Выделять и анализировать интервалы
3. Анализировать непосредственно временные ряды
 - Методы вроде ARIMA, LSTM
 - В планах на будущее

Классификация на основе интервалов

- Используем классификатор на основе *Random Forest*
- Обучение на 270 интервалах (115 реальных задач суперкомпьютера «Ломоносов-2»)



Что удалось обнаружить

- Постоянно работает на суперкомпьютере «Ломоносов-2»
 - Поток задач анализируется каждые 30 минут
 - Ежедневный отчет отправляется на почту администраторам
- Собрана статистика с 15 по 22 марта 2017
 - В среднем за день на «Ломоносов-2» выполняется 500-1000 задач
 - Классифицировано: 1862
- Найдено:
 - Аномальных задач: 95
 - Подозрительных: 244

Итого обнаружено:

~30.000 ядро-часов явно аномального поведения!

(задача зависла, некорректные входные данные, сбой инфраструктуры, ...)

~720.000 ядро-часов работы с низкой эффективностью

(неэффективная реализация, особенности алгоритма, различные сбои, ...)

Подобная ситуация везде...

Контроль

JobDigest

(детальный анализ поведения программы)

OctoStat

(статистика и аналитика)

OctoScreen

(показать все то, что мы хотим видеть и знать)

Цель:

Выявлять нештатные ситуации (любые отклонения от ожидаемого нормального поведения суперкомпьютера).

DiMMon

(полный контроль над аппаратурой и ПО)

OctoTron

(гарантия того, что суперкомпьютер ведет себя так, как мы этого ожидаем)

OctoShell

(регистрация, биллинг, взаимосвязи, статусы)

Пользователи
Сисадмины
Руководство

Проекты

Лицензии

Разделы

Компоненты
ПО

Очереди

Статусы

Приложения

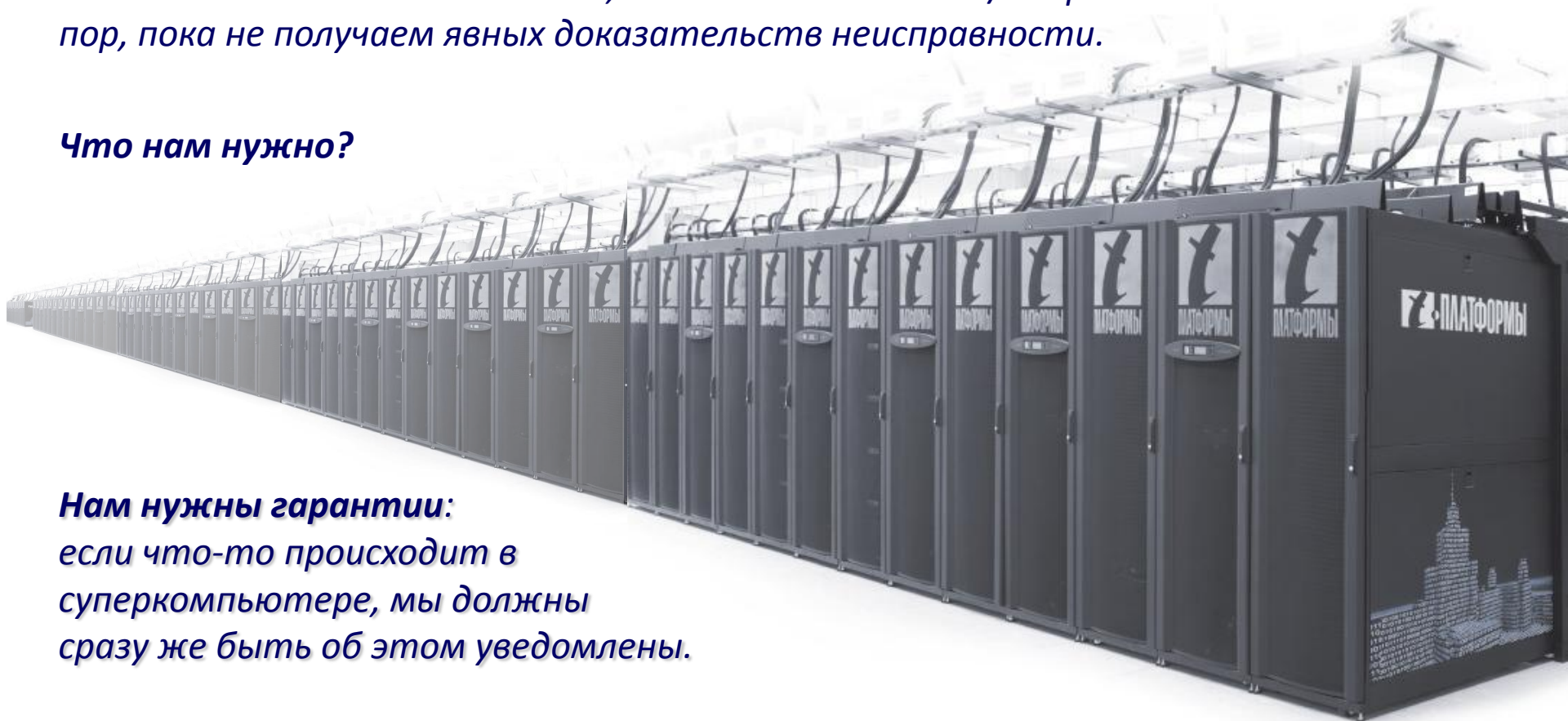
Задачи

Octotron: автономная жизнь суперкомпьютеров

Что сейчас? Мы лишь надеемся, что компонента HW/SW работает до тех пор, пока не получаем явных доказательств неисправности.

Что нам нужно?

*Нам нужны гарантии:
если что-то происходит в
суперкомпьютере, мы должны
сразу же быть об этом уведомлены.*



Octotron: автономная жизнь суперкомпьютеров

Что сейчас? Мы лишь надеемся, что компонента HW/SW работает до тех пор, пока не получаем явных доказательств неисправности.

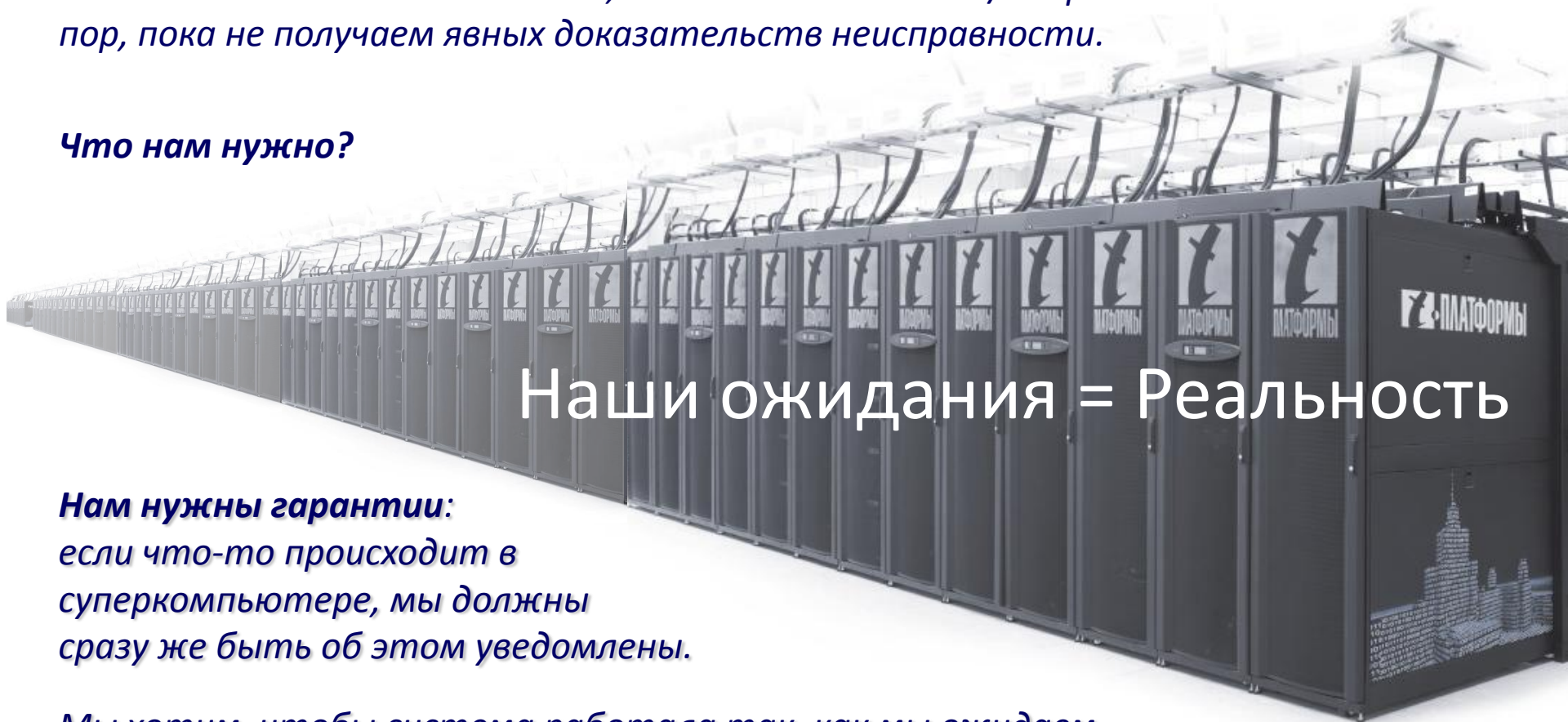
Что нам нужно?

Наши ожидания = Реальность

Нам нужны гарантии:

если что-то происходит в суперкомпьютере, мы должны сразу же быть об этом уведомлены.

Мы хотим, чтобы система работала так, как мы ожидаем.



Octotron: автономная жизнь суперкомпьютеров

*Если возникло несоответствие между нашими ожиданиями и реальным поведением суперкомпьютера, мы должны немедленно об этом узнать.
Но...*

*Суперкомпьютер огромен, мы сами больше не в состоянии его контролировать полностью.
Но...*

Суперкомпьютер может это делать сам (вместо нас), нужно только ему объяснить, что значит «наши ожидания».

Octotron: автономная жизнь суперкомпьютеров



Суперкомпьютер должен быть в состоянии выполнять самодиагностику и принимать решения полностью автономно.

Более того:

Чем больше суперкомпьютер, тем более автономным он должен быть...

Octotron: автономная жизнь суперкомпьютеров

Как мы это делаем?

- *Тотальный мониторинг над компонентами HW&SW, над инженерной инфраструктурой;*
- *Для гарантии того, что “наши ожидания = реальности”:*
 - *формальная модель суперкомпьютера (модель – это граф),*
 - *набор правил + набор реакций*

как основа для Автономной жизни и контроля над суперкомпьютерами МГУ:

- “Ломоносов”, 1.7 Pflops, 12K CPUs, 2K GPU:

116K+ вершин, 332K+ ребер, 2.4M+ атрибутов.

Разворачивание системы, обнаружение ошибок, критических и аварийных ситуаций, отключение минимального объема оборудования, самодиагностика, обеспечение преемственности и т.п. – все это должно делаться согласно модели суперкомпьютера.

Octotron: автономная жизнь суперкомпьютеров

Как мы это делаем?

- *Тотальный мониторинг над компонентами HW&SW, над инженерной инфраструктурой;*
- *Для гарантии того, что “наши ожидания = реальности”:*
 - *формальная модель суперкомпьютера (модель – это граф),*
 - *набор правил (какие нештатные ситуации ловить) + набор реакций (как на них реагировать)*

как основа для Автономной жизни и контроля над суперкомпьютерами МГУ:

*- “Ломоносов”, 1.7 Pfllops, 12K CPUs, 2K GPU:
116K+ вершин, 332K+ ребер, 2.4M+ атрибутов.*

Разворачивание системы, обнаружение ошибок, критических и аварийных ситуаций, отключение минимального объема оборудования, самодиагностика, обеспечение преемственности и т.п. – все это должно делаться согласно модели суперкомпьютера.

Примеры нештатных ситуаций

Простые случаи:

- Аппаратные сбои (диски, память, сетевые порты).
- Программные сбои (требуемые сервисы не работают).
- Сетевые сбои (оборудование не видны).
- Опасное состояние оборудования (перегрев, кончается свободное место).
- ...

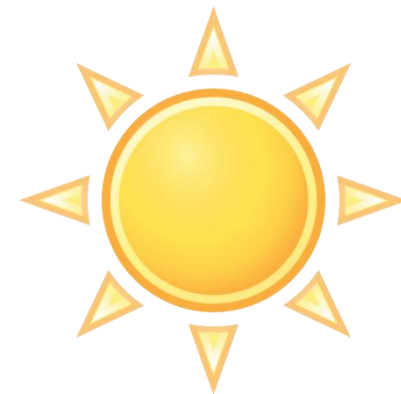
Более сложные случаи (проверка набора условий):

- Превышение заданного отношения сработавших датчиков (пример: проверка температуры в горячем коридоре).
- Несоответствие режимов работы компонент (пример: разные режимы на портах одного кабеля).
- Ошибка видна только при анализе общего состояния (пример: мало задач во всех очередях).

Варианты реакций на нештатные ситуации

- *Запись в лог (Json формат)*
- *Оповещение*
 - *Эл. почта*
 - *SMS*
- *Другие реакции:*
 - *Отключение оборудования*
 - *Вывод вычислительных узлов из счетного поля*
 - *Перезапуск агентов мониторинга*

Осготрон на практике (жара 10.06.15-19.06.15)



Опасность:

- 1089 — узел недоступен
- 63 — некорректный режим работы кондиционера
- 21 — высокая температура чиллера

Предупреждение:

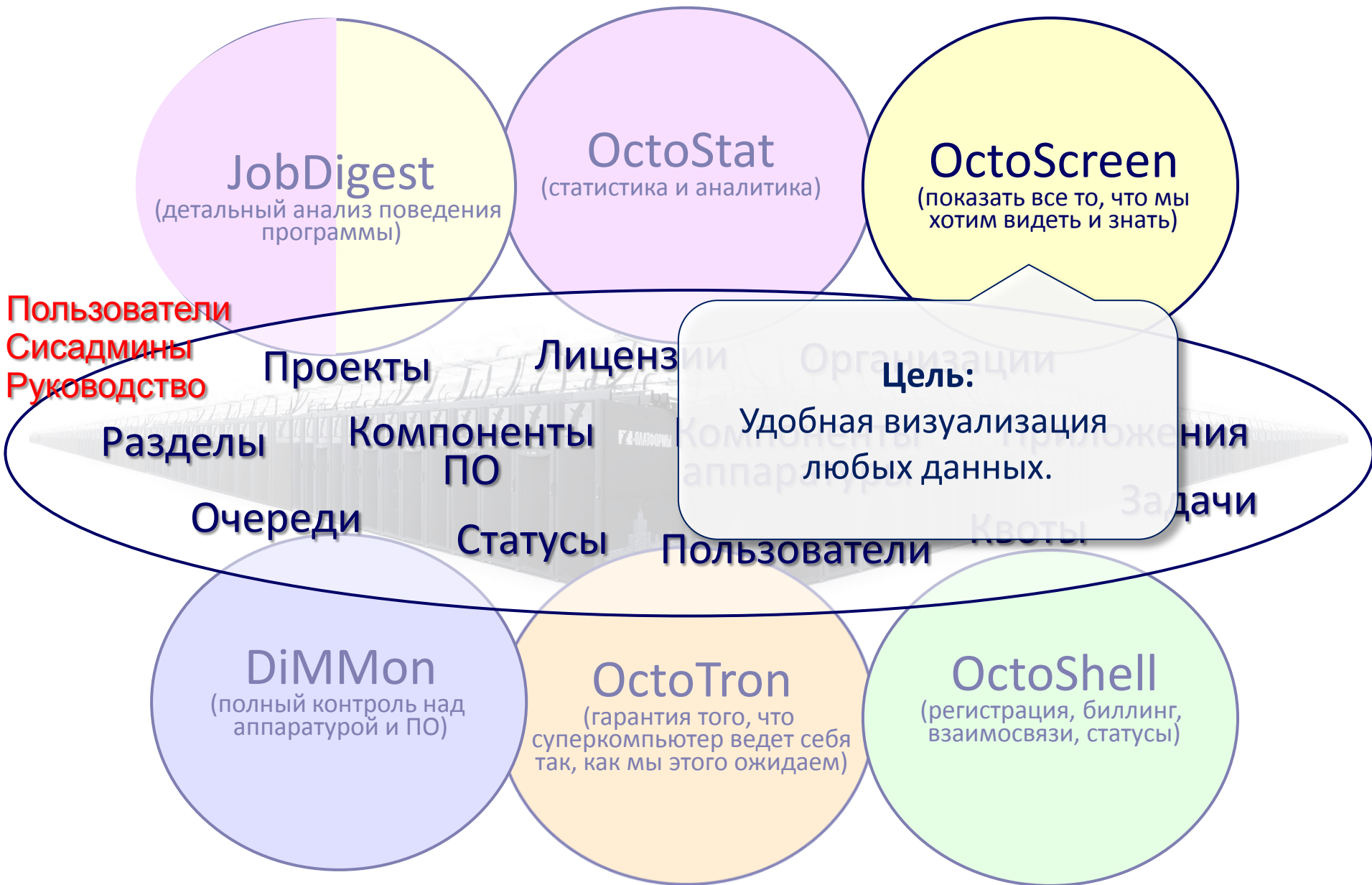
- 474 — высокая температура в горячем коридоре
- 157 — высокая температура в холодном коридоре
- 84 — высокая температура жидкости в кондиционере
- 29 — не работает порт Ethernet
- 9 — сбой Infiniband-карты

Информация:

- 103 — большое значение LoadAVG на узле
- 74 — процесс-зомби на узле
- 38 — высокий уровень порождения процессов на узле



Визуализация

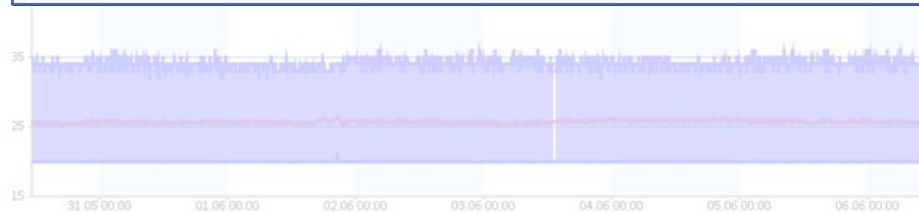
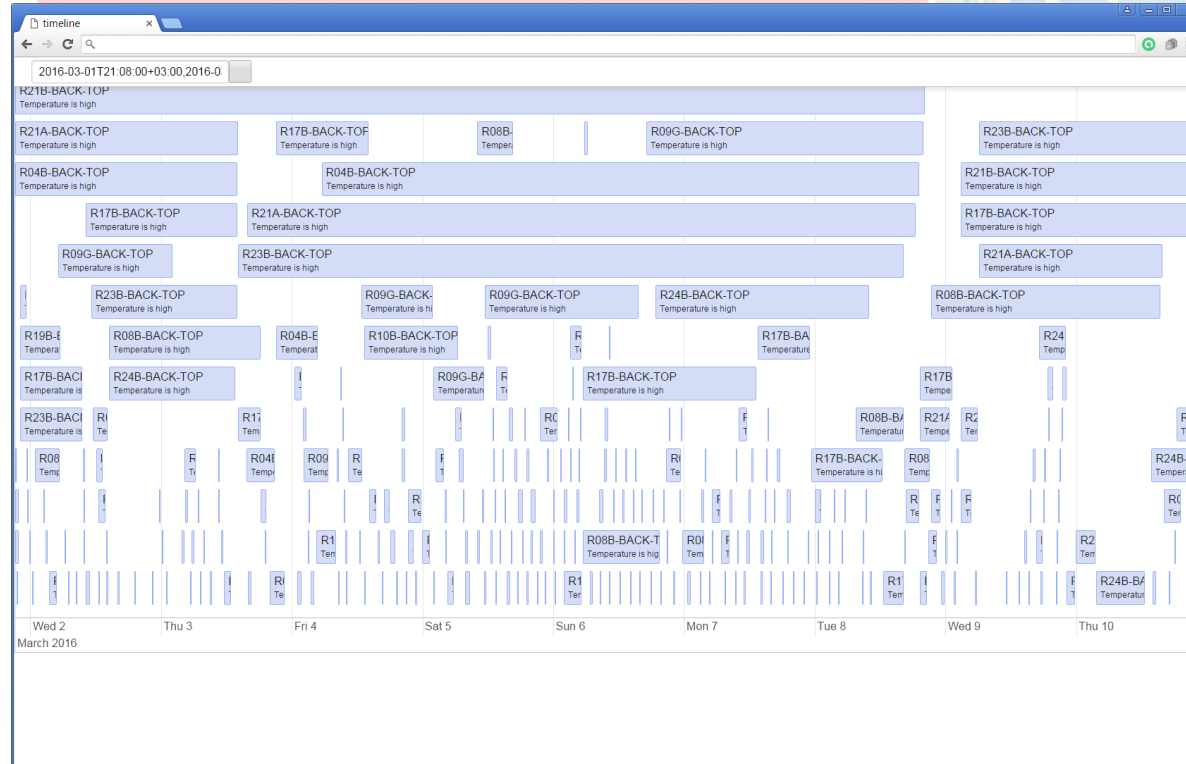
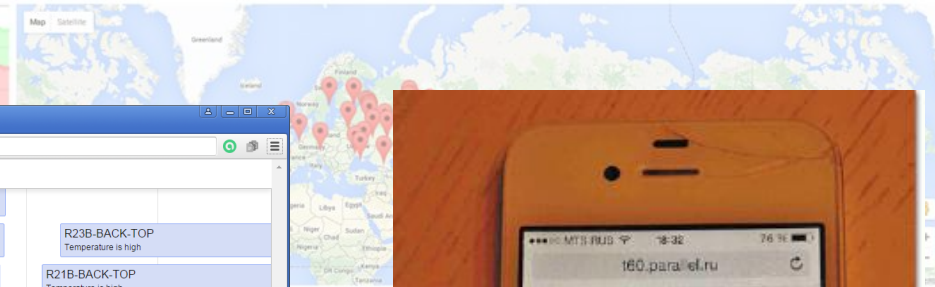


OctoScreen. Примеры

Число задач в очереди и на счёте



Участники расчётов - распределение по городам

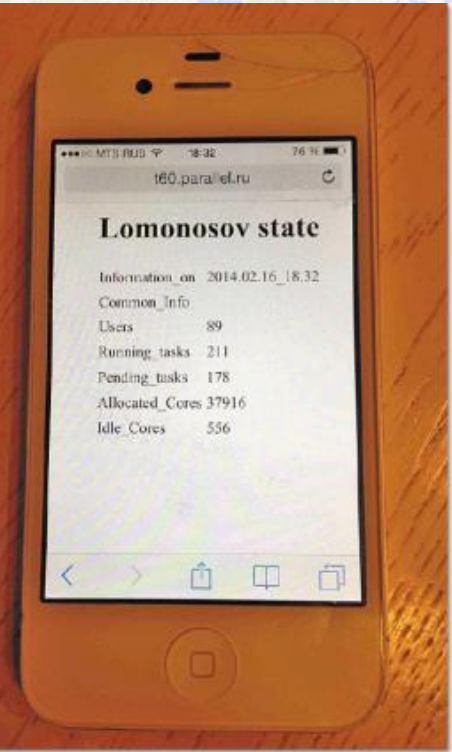


Среднее Границы

CanvasJS.com



R08	R09	R10	R11	R12	R13	R14	R15	R16	R17	R18	R19	R20	R21	R22	R23	R24	R25	R26	R27	R28	R29	R30	R31	R32	R33	R34	R35	R36	R37	R38	R39	R40	R41	R42	R43	R44	R45	R46	R47	R48	R49	R50	R51	R52	R53	R54	R55	R56	R57	R58	R59	R60	R61	R62	R63	R64	R65	R66	R67	R68	R69	R70	R71	R72	R73	R74	R75	R76	R77	R78	R79	R80	R81	R82	R83	R84	R85	R86	R87	R88	R89	R90	R91	R92	R93	R94	R95	R96	R97	R98	R99	R100
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	------

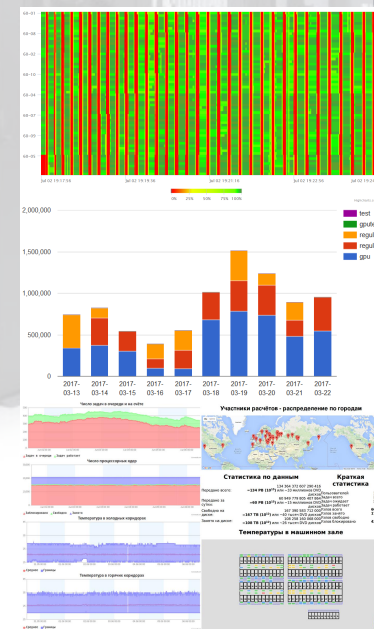


Организация работы СКЦ



OctoShell. Основные функции

- Организация Проектов (отдельных исследований), в рамках которых происходит:
 - выделение ресурсов
 - создание и контроль учетных записей
- Предоставление доступа
- Поддержка пользователей
 - решение вопросов пользователей
 - справка, рассылки и уведомления
- Обслуживание оборудования
- Предполагается интеграция Octoshell со всеми системами эффективного управления



Молодежная научная школа

Высокопроизводительные вычисления, оптимизация и приложения

Спасибо!

Вопросы?

*Воеводин Вад.В.
vadim@parallel.ru*

9 сентября 2017 г., Нижний Новгород