

Федеральное государственное автономное образовательное учреждение
высшего образования
«Национальный исследовательский университет «Высшая школа экономики»

Технический отчет для проекта
**«HPC TaskMaster – система мониторинга эффективности задач на
суперкомпьютере»**

Оглавление

Техническое задание	3
Актуальность	3
Цели, задачи, планируемый и фактический результат	3
Обоснованность и достоверность полученных результатов	5
Реализация и внедрение результатов проекта	5
Подробное описание разработанной системы	5
Компоненты системы	5
Индикаторы	6
Теги	9
Выводы	11
Пример формирования вывода	12
Приоритет выводов	13
Уведомление пользователей о неэффективных задачах	14
Статистик эффективности работы задач пользователя	15
Алгоритмы работы системы	17
Экономическая эффективность	19
Перечень основных технических и научных результатов	20
Примеры работы ПО	20
Ход работ	22
Новизна и преимущества полученных решений	22
Заключение	23
Публикации	24

Техническое задание

Актуальность

Проблема неэффективного использования вычислительных ресурсов суперкомпьютера является одной из наиболее актуальных в суперкомпьютерной отрасли. Существует множество решений для мониторинга и отслеживания состояния вычислительных ресурсов, однако как правило, чем сложнее устроен суперкомпьютер, тем более специфичное решение необходимо реализовывать на его основе. Для суперкомпьютера «сHARISMa» НИУ ВШЭ борьба с простаивающими ресурсами и порождающими их задачами является особенно важной – количество пользователей суперкомпьютера достигает тысячи человек, а через очередь планировщика проходят сотни тысяч задач, из-за чего загрузка суперкомпьютера нередко достигает 100%. Для того, чтобы помочь администраторам суперкомпьютера и его пользователям определять неэффективные задачи, необходимо разработать экспертную систему, которая на основе собранной информации о задаче и агрегированных метрик сможет однозначно определить неэффективную задачу.

Цели, задачи, планируемый и фактический результат

Цель – разработать систему мониторинга эффективности задач «HPC TaskMaster»

Для достижения данной цели необходимо выполнить следующие задачи:

- Разработать систему мониторинга, позволяющую собирать основные показатели использования CPU, GPU, оперативной памяти и файловой системы и визуализировать их
- Разработать веб-приложение, предоставляющее разграниченный доступ к задачам суперкомпьютера для его пользователей и администраторов

- Разработать класс «индикаторы проблем», который позволит определить низкое/чрезмерное использование вычислительных ресурсов
- Разработать класс «теги», который позволит присваивать теги задачам на основании их свойств
- Разработать класс «выводы», который позволит на основании «индикаторов» и «тегов» делать окончательный вывод о том, неэффективно ли работала задача и предоставить его пользователю
- Разработать класс, позволяющий реализовывать сбор статистики эффективности работы задач пользователей для анализа из работы
- Разработать систему email оповещений пользователей
- Подготовить несколько публикаций о системе HPC TaskMaster
- Подготовить открытый репозиторий проекта для общего доступа

Планируемый результат – система мониторинга эффективности задач «HPC TaskMaster» запущена в производственную эксплуатацию на суперкомпьютере «сHARISMa», автоматически определяет неэффективные задачи, собирает по ним статистику и уведомляет пользователей об их неэффективной работе

Фактический результат – в результате работы система «HPC TaskMaster» была успешно запущена в производственную эксплуатацию. На основе результатов анализа работы задач была собрана статистика эффективности пользователей за последние месяцы, благодаря чему были проведены консультации для пользователей с наиболее неэффективными задачами, в результате чего удалось снизить неэффективную загрузку суперкомпьютера на 5%.

Обоснованность и достоверность полученных результатов

Критерии определения задачи пользователя как «неэффективной» задаются администраторами суперкомпьютера «сHARISMa» в панели администрирования системы «HPC TaskMaster».

Реализация и внедрение результатов проекта

Разработка проекта велась в закрытом GIT репозитории.

Отдельный open-source репозиторий проекта с тестовым датасетом доступен для всех желающих.

Подробное описание разработанной системы

Компоненты системы

Диаграмма компонентов системы «HPC TaskMaster» представлена на Рисунок 1:

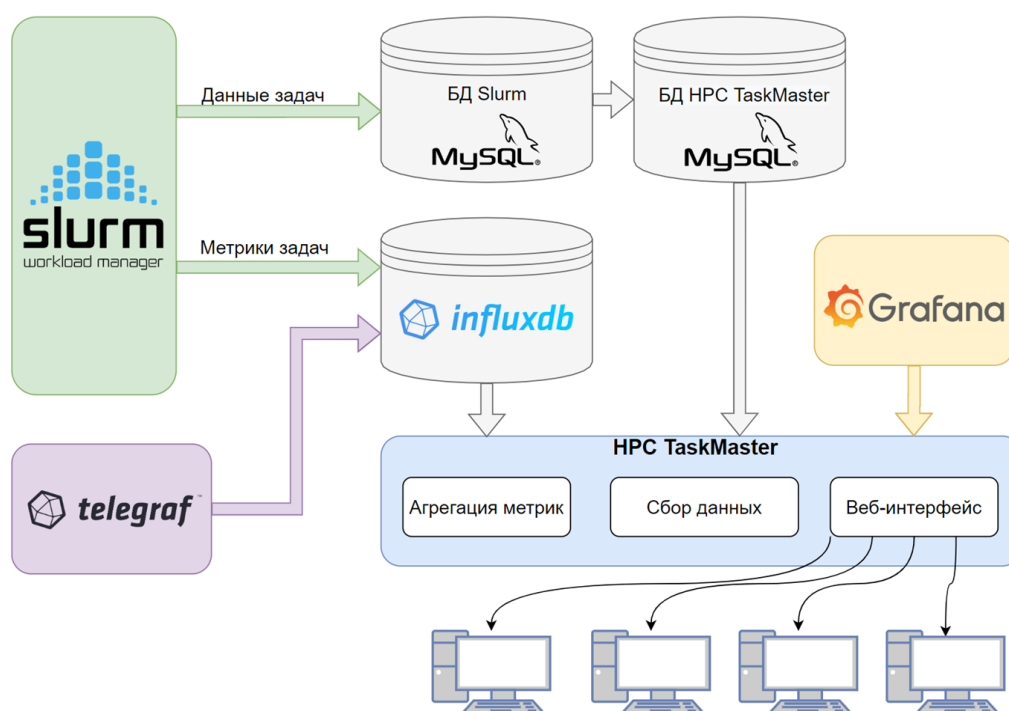


Рисунок 1 - Диаграмма компонентов системы

Используемые компоненты для работы системы:

1. Планировщик задач Slurm
2. База данных MariaDB
3. База данных временных рядов InfluxDB
4. Сборщик метрик Telegraf
5. Веб-приложение для визуализации метрик Grafana
6. Python 3.5+, Django и остальные требуемые пакеты
7. База данных Redis

Основная информация о задачах пользователей собирается с планировщика задач Slurm, который выгружает ее в базу данных MySQL. Также, отдельный плагин Slurm выгружает временные ряды по конкретным задачам в базу данных временных рядов InfluxDB. Также, сбор метрик по утилизации CPU и GPU происходит при помощи демона Telegraf, установленного на каждый вычислительный узел суперкомпьютера. Telegraf позволяет собирать утилизацию по каждому конкретному ядру CPU и GPU.

Визуализация метрик происходит при помощи веб-интерфейса Grafana. За счет API, процесс создания графиков для каждой задачи полностью автоматизирован.

Индикаторы

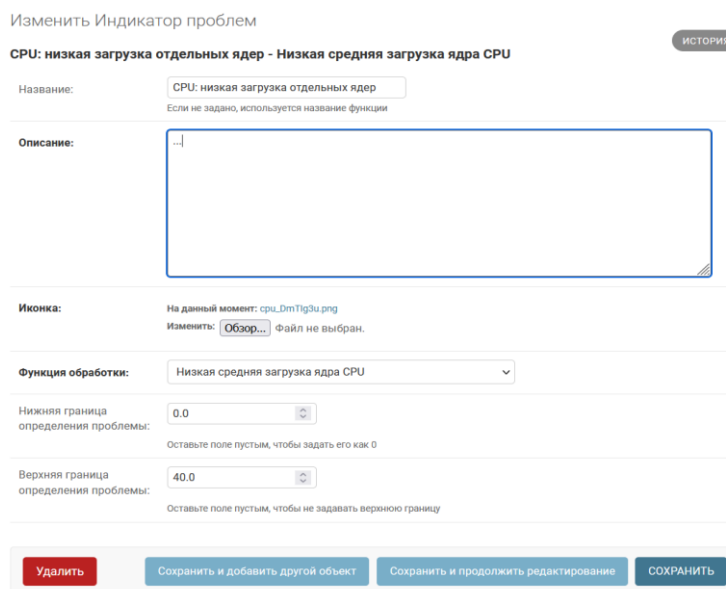
Для определения проблем с использованием ресурсов задач, в HPC TaskMaster было разработано следующий класс – индикатор. Список используемых индикаторов представлен в Таблица 1.

Таблица 1 - индикаторы системы

№	Название индикатора	Границы определения индикаторов	Тип индикатора
1	Низкая средняя загрузка ядра CPU	0-30%	Индикатор для нижней границы
2	Низкая средняя загрузка всех CPU	0-60%	Индикатор для нижней границы
3	Низкая средняя загрузка GPU	0-20%	Индикатор для нижней границы
4	Низкое средняя загрузка VRAM	0-10%	Индикатор для нижней границы
5	Низкая средняя загрузка RAM	0-1 ГБ	Индикатор для нижней границы
6	Высокая нагрузка на SSD (чтение)	350-500 МБ/с	Индикатор для верхней границы
7	Высокая нагрузка на SSD (запись)	350-500 МБ/с	Индикатор для верхней границы

- Индикаторы позволяют определить наличие проблем с ресурсами задачи;
- Администратор определяет критерии для индикаторов;
- Индикатор хранит значение уровня проблемы от 0 до 1 с двоичной точностью, который вычисляется как отношение показателя использования к разнице верхней и нижней границ;
- Нижние и верхние границы индикатора обязательные для нормирования значения индикатора. Для определенных индикаторов превышение верхней границы автоматически задает уровень индикатора равным 1;
- Существуют два типа индикаторов – индикатор верхней и нижней границ. Для индикатора нижней границы значения ниже минимальной границы будут приравнены к 0, а для индикатора верхней границы, значения выше максимальной границы будут приравнены к 1.

Пример формирования индикатора в панели администрирования системы показан на Рисунок 2.



Изменить Индикатор проблем

ИСТОРИЯ

CPU: низкая загрузка отдельных ядер - Низкая средняя загрузка ядра CPU

Название: CPU: низкая загрузка отдельных ядер
Если не задано, используется название функции

Описание: [...]

Иконка: На данный момент: cpu_DmTt3zi.png
Изменить: Обзор... Файл не выбран.

Функция обработки: Низкая средняя загрузка ядра CPU

Нижняя граница определения проблемы: 0.0
Оставьте поле пустым, чтобы задать его как 0

Верхняя граница определения проблемы: 40.0
Оставьте поле пустым, чтобы не задавать верхнюю границу

Удалить Сохранить и добавить другой объект Сохранить и продолжить редактирование СОХРАНИТЬ

Рисунок 2 - пример формирования индикатора

Преимущества индикаторов

1. Наличие индикатора сразу свидетельствует о проблеме с утилизацией ресурсов задачей;
2. Легко добавить новый индикатор в систему.

Недостатки индикаторов

1. Не всегда наличие индикатора свидетельствует о том, что задача неэффективна;
2. Вследствие пункта выше, нельзя автоматически уведомлять/снимать задачу пользователя без 100% гарантии неэффективности задачи;

Для дополнения информации, получаемой от индикаторов, был разработан новый класс – теги.

Теги

Теги задач – метки, передающие одно из свойств задачи. Тегами можно обозначить такие свойства, как длительность, тип задачи, обнаруженные ошибки. В отличие от индикаторов, теги не хранят в себе значение уровня. Теги нужны, чтобы делать более точные выводы о работе задачи. С помощью тегов можно с легкостью внедрять новые свойства в систему выводов.

Список тегов, используемых системой, предоставлен в Таблица 2.

Таблица 2 - теги системы

№	Название тега	Критерии назначения тега	Комментарий
1	Разбалансировка ресурсов	Неравномерное выделение ресурсов на нескольких узлах	При неправильном выделении ресурсов для задачи эти ресурсы могут быть распределены неравномерно между несколькими узлами, что приведет к снижению эффективности
2	Задача выполнялась дольше 7 дней	Длительность задачи более 7 дней	Длительные задачи требуют более пристального внимания, т. могут быть вызваны ручным выделением ресурсов или неправильно работающей программой
3	Задача завершена с ошибкой	Задача завершена с одним из аварийных статусов	Наличие аварийного статуса свидетельствует об ошибке
4	Задача запущена на 1 ядре	Для задачи выделено 1 ядро	Данный тег предназначен для фильтрации задач с 1 ядром
5	Тип задачи - salloc	Задача запущена через salloc/srun	Часто пользователи

			использующие salloc выделяют ресурсы и не используют их большую часть времени/не завершают задачу по окончанию расчетов
6	Тип задачи – Jupyter Notebook и др	Определение типа задачи по ключевым словам	Данный тег формируется по ключевым словам, поиск которых ведется в названии, в строке запуска, в исполняемом скрипте

Форма конфигурирования тега в панели администрирования системы продемонстрирована на Рисунок 3.

Изменить Тег задачи ИСТОРИЯ

Тип задачи - IO500

Отображение:

Название:
Если не задано, используется название функции

Описание:

Иконка: На данный момент: io500.jpg
Изменить: Файл не выбран.

Параметры определения тегов:

Доступные параметры для выбора

Не определены ID GPU

Максимальная длительность задачи

Минимальная длительность задачи

Задача на одном ядре

Задача завершена с ошибкой

Разбалансировка ресурсов

Избыточное использование RAM

*Перенесите
параметры
вправо,
чтобы
применить
их→*

Используемые параметры

Тип задачи

По набору этих параметров создается тег

Рисунок 3 - пример формирования тега

Так как индикаторы и теги предоставляют только общую информацию о работе задачи, дополнительно была разработана подсистема выводов, которая позволяет более точно определять, эффективно ли работала задача, используя различные параметры, в том числе показания индикаторов.

Выводы

Вывод – это финальный результат анализа задачи, предоставляемый пользователю на основе показателей имеющих индикаторов и тегов.

Принцип формирования вывода показан на Рисунок 4.

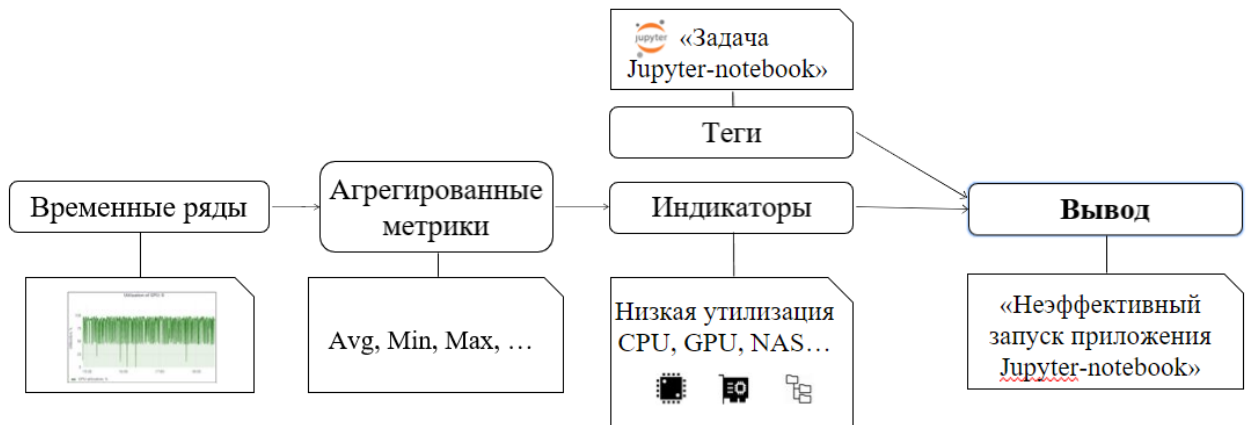


Рисунок 4 - принцип формирования выводов

- На входе есть N временных рядов (результаты мониторинга выполнения задач на кластере).
- По ним вычисляются метрики (средние значения, максимумы и минимумы и т.д.)
- Для каждой задачи получается вектор метрик
- Вектор метрик обрабатывается набором функций, каждая из которых выдает один индикатор с указанием его веса (от 0 до 1)
- Собираем все результаты функций в вектор индикаторов для задачи (например, «Малое использование памяти»).

- Далее вектор показателей поступает на вход булевым функциям. На выходе получаются *выводы о задаче* (результата анализа, например: «Неэффективный запуск приложение jupyter-notebook»)

Пример формирования вывода

На Рисунок 5 приведен пример вывода о неэффективном использовании задачей интерактивного режима запуска.

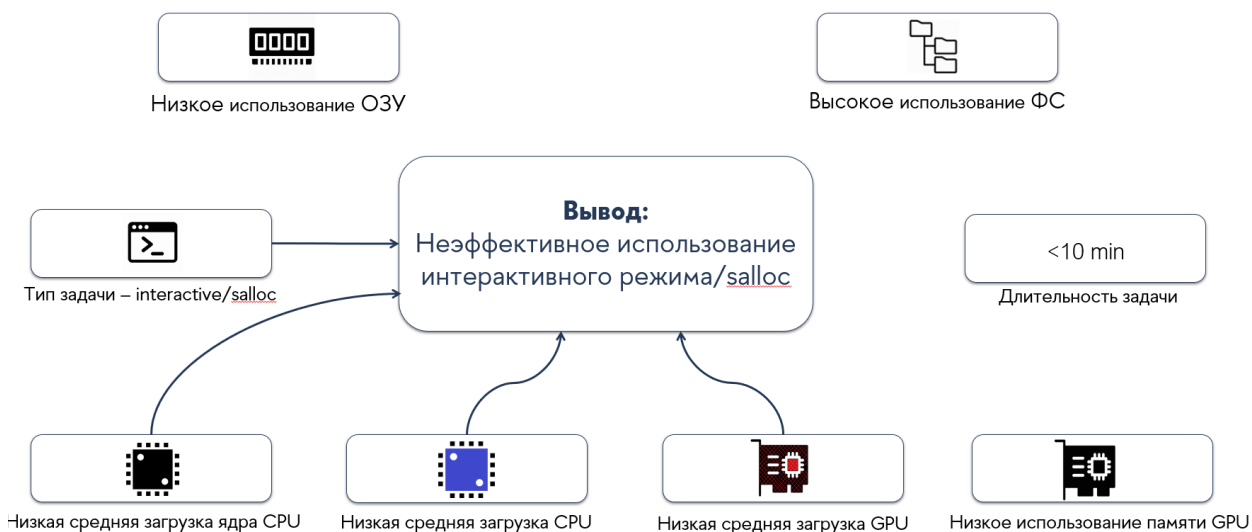


Рисунок 5 - формирование вывода

Пример формирования вывода об неэффективной задаче представлен на Рисунок 6.

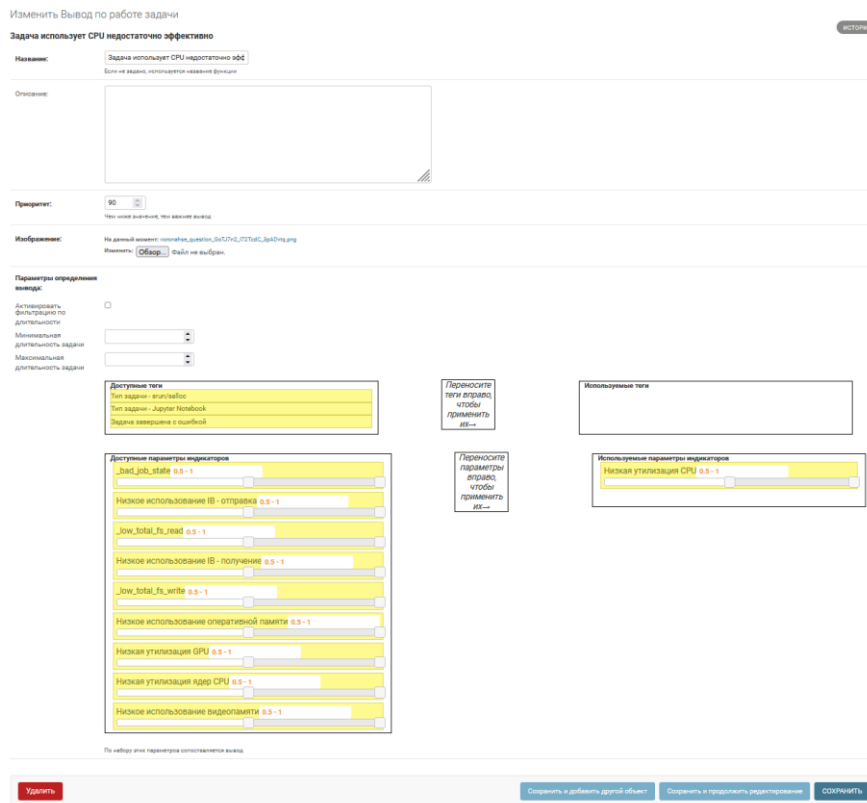


Рисунок 6 - пример формирования вывода

Приоритет выводов

В Таблица 3 приведены примеры выводов, используемые в системе HPC TaskMaster. Вывод применяется при соблюдении всех условий. Параметры выводов в таблице подобраны эмпирическим путем и настроены специально для суперкомпьютера «CHARISMA».

Таблица 3 - выводы системы

Приоритет	Название	Индикаторы, значения уровней				Теги				
		Низкое среднее использование CPU	Низкое среднее использование ядер CPU	Низкое среднее использование GPU	Низкая среднее использование видеопамати	Salloc	Jupyter Notebook	Завершена с ошибкой	Длилась менее 10 секунд	Выделено меньше ядер CPU чем GPU
11	Неэффективное использование salloc	0.9-1	0.9-1	0.9-1	-	T	F	F	F	F

22	Неэффективное использование Jupyter Notebook	0.5-1	0.5-1	0.5-1	-	F	T	F	F	F
33	Ошибка в параметрах запуска задачи	-	-	-	-	F	F	T	T	F
44	Неправильный запуск GPU задачи	-	-	0-1	0-1	F	F	F	F	T
55	Задача не использует GPU	-	-	0.9-1	0.9-1	F	F	F	F	F
66	Задача не использует CPU	0.9-1	0.9-1	-	-	F	F	F	F	F
77	Задача неэффективно использует одно или несколько ядер CPU	-	0-1	-	-	F	F	—F	—F	—F

Уведомление пользователей о неэффективных задачах

Имея автоматизированный способ определения неэффективных задач, появилась возможность реализовать автоматическое уведомление и их снятие с суперкомпьютера.

Новый модуль автоматически просматривает все активные задачи на суперкомпьютере и на основе заданных критериев в панели администрирования выполняет либо простое уведомление о неэффективной задаче, либо снимает задачу и уведомляет об этом пользователя.

Статистик эффективности работы задач пользователя

Для «HPC TaskMaster» был разработан специальный модуль сбора статистики эффективности работы задач пользователей. Данный модуль позволяет собирать и отражать статистику по следующим показателям:

- Эффективность пользователя, где отображается процент эффективных задач, процент эффективно использованных CPU и GPU;
- Количество и тип задач по месяцам. В сводке приводятся эффективные задачи и задачи, у которых были обнаружены проблема;
- Количество использованных ресурсов пользователем. В данной статистике отображается количество CPU и GPU, которые пользователь использовал эффективно и неэффективно;
- Эффективное и неэффективное использование машинного времени суперкомпьютера. Данная статистика представляется собой наибольший интерес, т. к. именно по ней можно оценивать, насколько эффективно пользователь использует выделяемые ему ресурсы;

Пример собранной статистики пользователя за год показан на Рисунок 7.

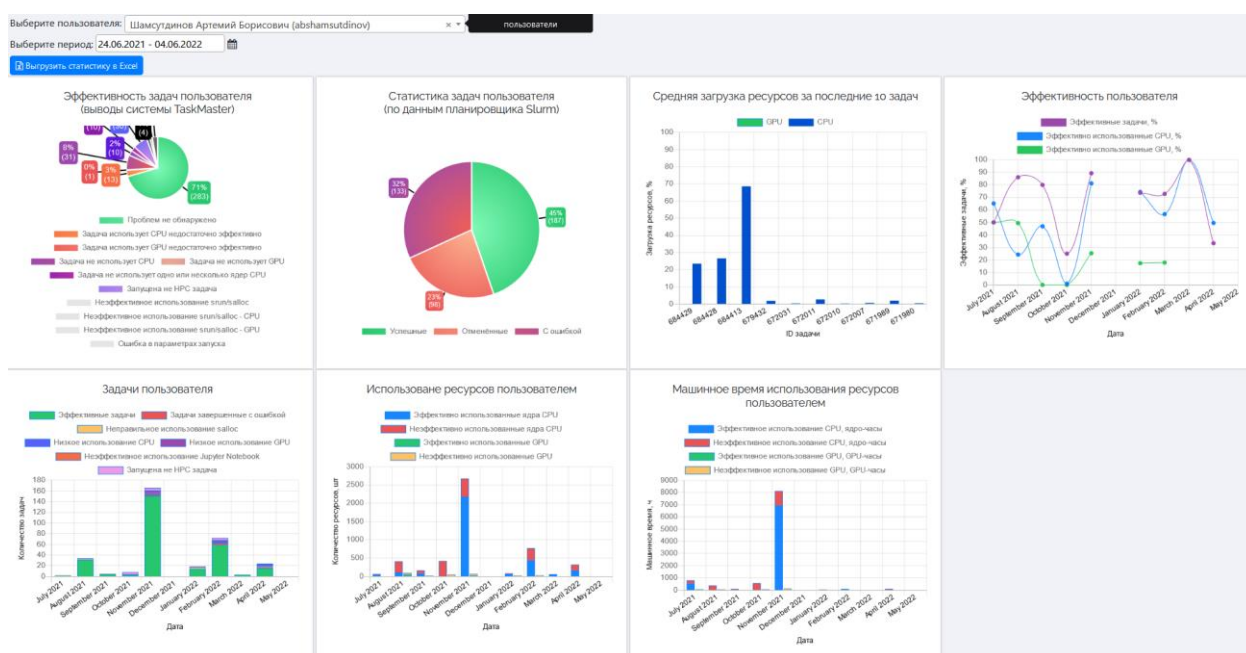


Рисунок 7 - пример статистики эффективности работы пользователя

Данный модуль является ценным инструментом при консультациях с пользователем суперкомпьютера – на основе данных можно быстро сделать выводы:

- с каким типом задач у пользователя возникают наибольшие трудности;
- как много машинных часов пользователь занимает и сколько из них простаивает;
- как активно он использует ресурсы суперкомпьютера;
- какой прослеживается тренд в использовании суперкомпьютера – стал ли пользователь эффективнее использовать ресурсы после консультации или нет;

Для реализации сбора статистики был разработан новый класс под названием «UserStat». Данный класс хранит в себе агрегированные показатели эффективности пользователя за один календарный месяц:

- Общее количество задач
- Количество эффективных задач (задач, для которых нет вывода о проблеме)
- Количество эффективных задач в %
- Количество задач с ошибкой в параметрах запуска
- Количество задач с неправильным использованием srun/salloc
- Количество задач с низким использованием CPU
- Количество задач с низким использованием GPU
- Количество задач с неэффективным использованием Jupyter Notebook
- Количество непараллельных задач
- Количество задач с неопределенным типом проблемы
- Количество использованных ядер CPU

- Количество неэффективно использованных ядер CPU
- Количество эффективно использованных ядер CPU в %
- Время использования ядер CPU
- Время неэффективного использования ядер CPU в секундах
- Эффективно использованное время CPU в %
- Количество использованных ядер GPU
- Количество неэффективно использованных ядер GPU
- Количество эффективно использованных ядер GPU в %
- Время использования ядер GPU
- Время неэффективного использования ядер GPU в секундах
- Эффективно использованное время GPU в %

За счет наличия агрегированных полей, которые хранят готовые значения, обращения к БД занимают намного меньше времени и отрисовка графиков происходит намного быстрее, т. к. системе не приходится каждый раз проводить расчеты.

Алгоритмы работы системы

Основной алгоритм работы системы предоставлен на Рисунок 8.

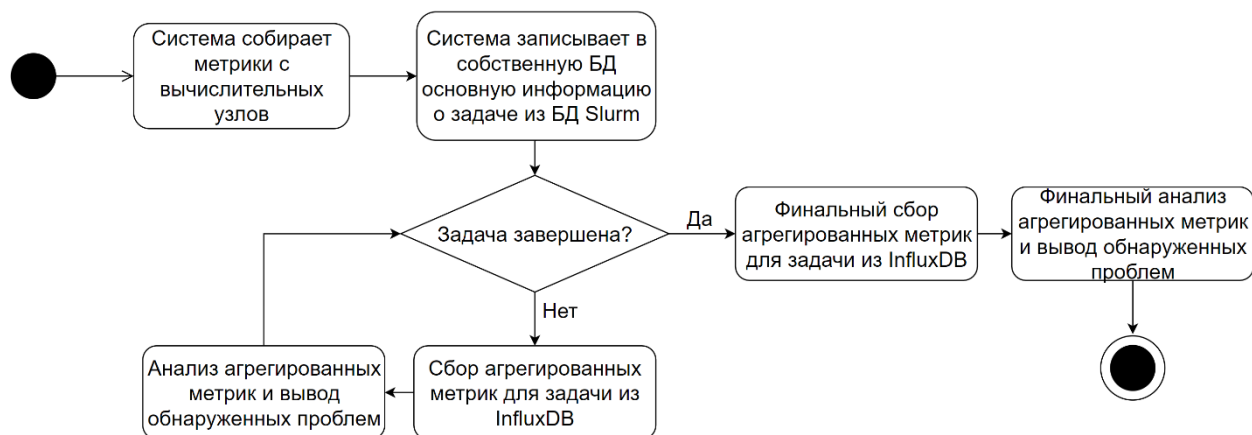


Рисунок 8 - алгоритм работы системы

Автоматическое создание графиков для задачи в Grafana происходит по следующему алгоритму, представленному на Рисунок 9.

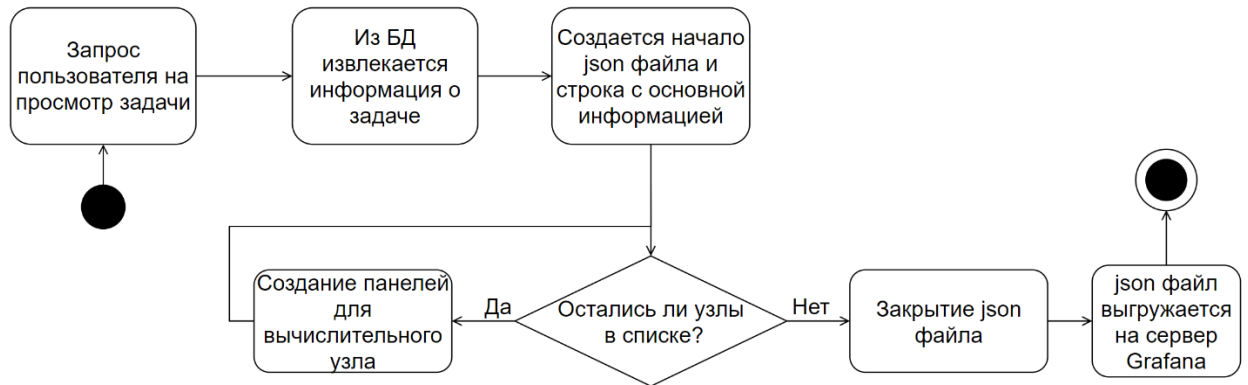


Рисунок 9 - алгоритм создания графиков в Grafana

Алгоритм формирования вывода представлен на Рисунок 10.

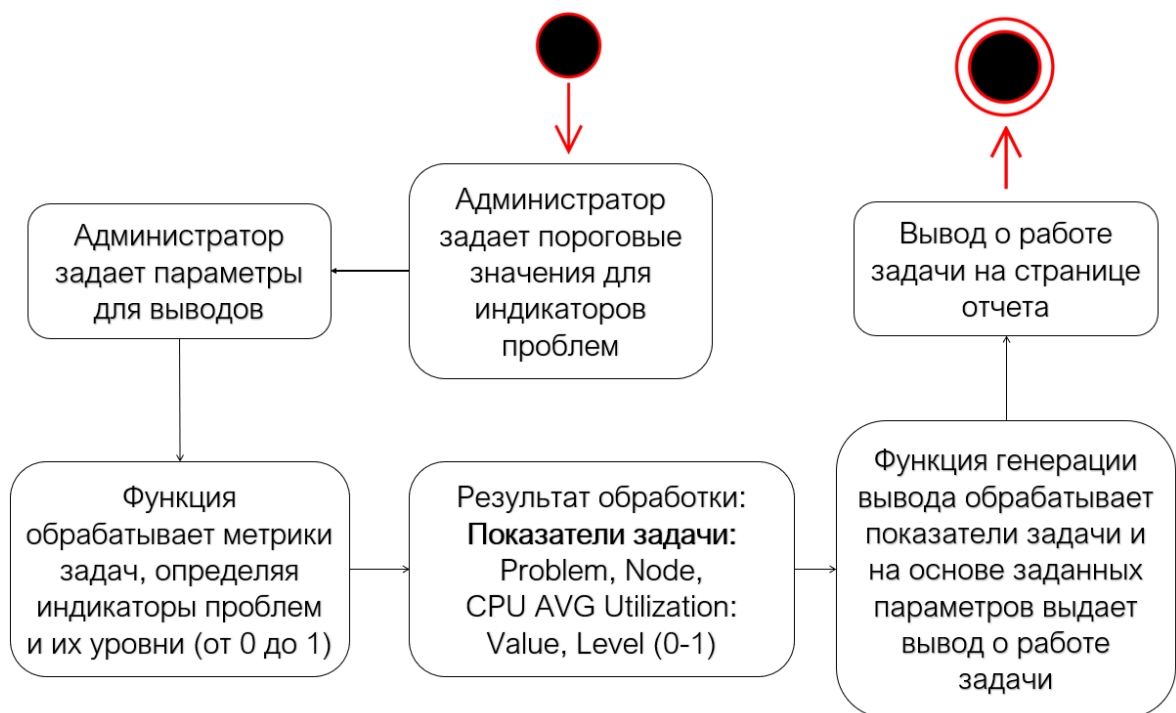


Рисунок 10 - алгоритм формирования вывода

Алгоритм автоматических email уведомлений о неэффективных задачах представлен на Рисунок 11.

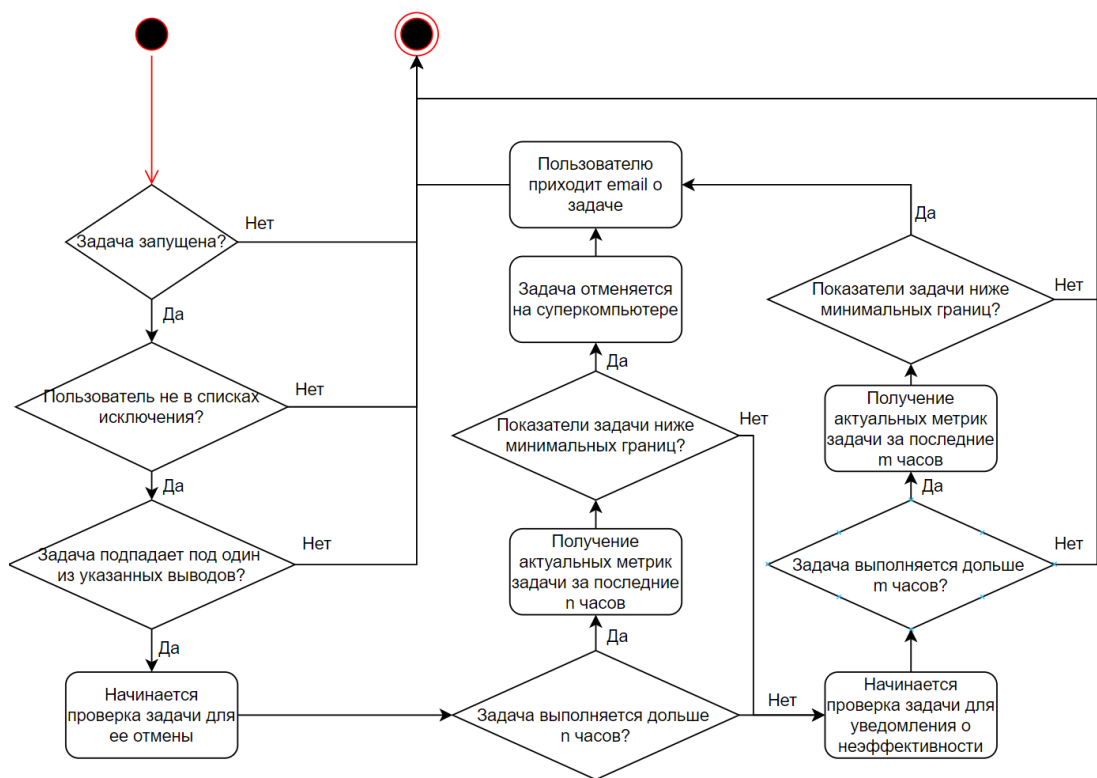


Рисунок 11 - алгоритм работы email уведомлений

Экономическая эффективность

Благодаря разработке данной подсистемы была собрана статистика эффективности задач пользователей суперкомпьютера за последний квартал. Были проведены консультации с пользователями для помощи с правильным запуском задач. За счет данных консультаций удалось сократить объем неэффективных вычислений более чем на 100000 машинных часов, таким образом на 3,5% больше вычислительных ресурсов освободилось для пользователей суперкомпьютера. Дальнейшие консультации позволят уменьшить количество неэффективных вычислений ориентировочно еще на 3%. Таким образом, система без существенных вложений позволяет экономить дорогостоящее машинное время суперкомпьютера, освобождая его для приоритетных научных исследований НИУ ВШЭ.

Перечень основных технических и научных результатов

- 1) Подсистема анализа эффективности задач внедрена в систему HPC TaskMaster на суперкомпьютере sHARISMa и запущена в производственную эксплуатацию.
- 2) Собрана статистика эффективности более чем по 500 пользователям суперкомпьютера.
- 3) Опубликовано три научные статьи, посвященные системе «HPC TaskMaster», две из них на английском в журнале уровня Q2 Scopus.

Примеры работы ПО

Начальная страница личного кабинета пользователя суперкомпьютера представлена Рисунком 12.

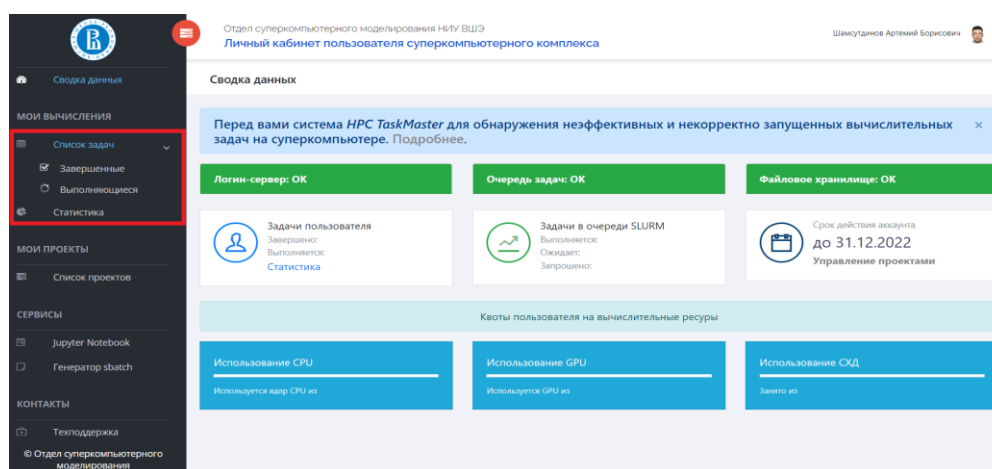


Рисунок 12 - начальная страница личного кабинета

Пример неэффективной задачи пользователя представлен на Рисунок 13 – пользователь выделил ресурсы при помощи команды srun/salloc, но не использовал их

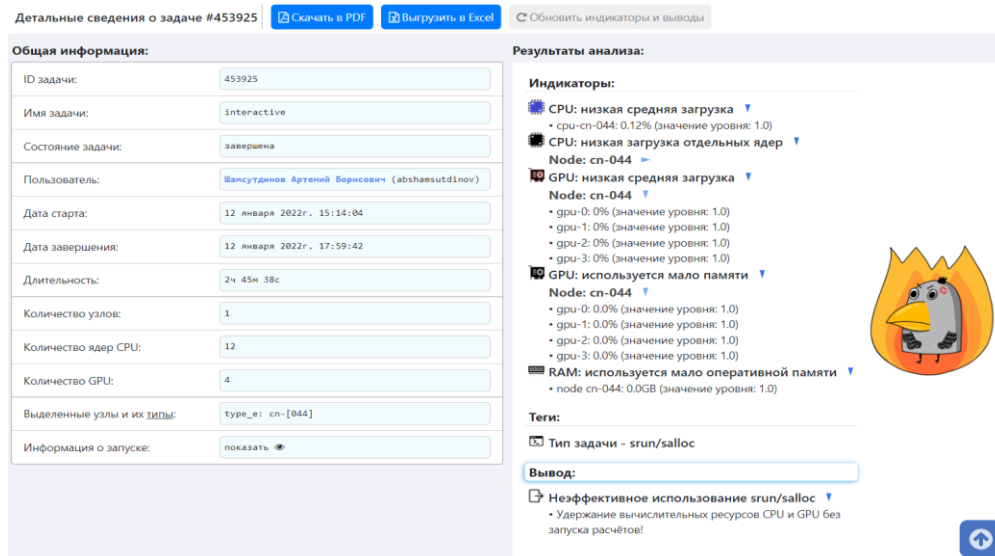


Рисунок 13 - пример неэффективной задачи пользователя

Пример построенных графиков для неэффективной задачи пользователя представлен на Рисунок 14.



Рисунок 14 - пример графиков для неэффективной задачи

Ход работ

1. Литературный обзор схожих решений по анализу эффективности задач на суперкомпьютере
2. Настроена система мониторинга с использованием инструментов, таких как Telegraf, InfluxDB, Grafana
3. Разработка первой версии системы «HPC TaskMaster» с классом «индикаторов»
4. Разработка класса «тегов» для определения свойств задач
5. Разработка класса «выводов» для формирования выводов о работе задач
6. Подготовка публикации для международной суперкомпьютерной конференции PaVT 2022 на английском языке
7. Подготовка открытого репозитория системы «HPC TaskMaster» с документацией и тестовыми данными
8. Разработка системы сбора статистики об эффективности работы задач пользователей
9. Разработка системы email оповещений пользователей о запуске ими неэффективных задач

Новизна и преимущества полученных решений

- 1) Разработанная система определения неэффективных задач написана с использованием открытого программного обеспечения и может быть развернута на других суперкомпьютерах
- 2) Критерии для определения неэффективных задач задаются администратором, а не программистом, что дает возможность быстро создать новый тип «выводов» для определения новых неэффективных задач
- 3) Система анализа эффективности задач позволяет создавать комплексные выводы для точного определения конкретной неэффективной задачи, за счет множества доступных критериев.

Дополнительные критерии всегда могут быть добавлены с минимальным изменением кода программы

- 4) Реализован подсчет неэффективно использованных ресурсов в машиночасах, что позволяет точно определить, сколько ресурсов занял пользователь
- 5) Разработанная система статистики позволяет оценить эффективность пользователя за любой промежуток времени и оказать ему своевременную консультацию

Заключение

В результате проекта была разработана система мониторинга эффективности задач «HPC TaskMaster» для суперкомпьютера НИУ ВШЭ «CHARISMa». Разработанная система позволила оценить эффективность использования центральных процессоров, графических ускорителей и твердотельных накопителей. Пользователям суперкомпьютера стал доступен полноценный отчет о работе каждой их задаче, с развернутыми выводами об их работе.

Были написана статья на английском языке для конференций Параллельные Вычислительные Технологии 2022, посвященная системе «HPC TaskMaster».

Автоматическое уведомление пользователей о неэффективных задачах и автоматическая отмена неэффективных задач позволяет расчистить очередь суперкомпьютера, тем самым предоставив занятые ресурсы другим пользователям и ускорив общую работу на кластере.

Благодаря работе системы и проведенным за счет нее консультациям пользователей, за 2022 год удалось повысить эффективность использования центральных процессоров на 5%, а графических ускорителей – на 10%. Статистика эффективности использования ресурсов представлена на Рисунок 15.

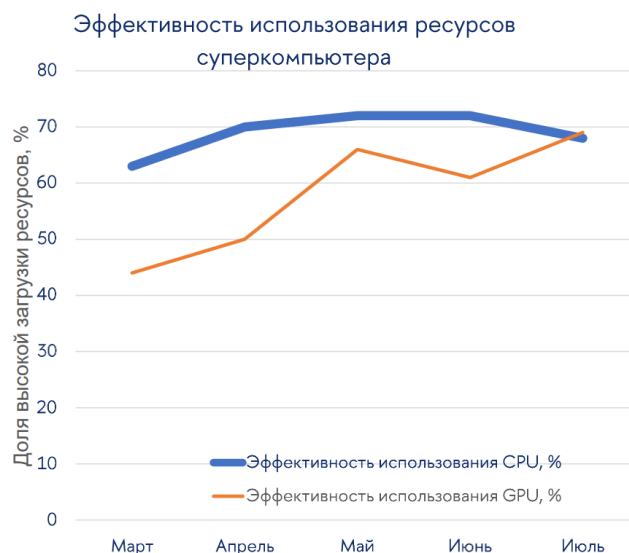


Рисунок 15 - статистика эффективности использования ресурсов по месяцам

Публикации

1. Костенецкий П. С., Козырев В. И., Чулкевич Р. А., Шамсутдинов А. Б. [HPC TaskMaster – система мониторинга эффективности задач суперкомпьютера](#) // В кн.: Суперкомпьютерные дни в России: Труды международной конференции (27-28 сентября 2021 г., Москва). М. : МАКС Пресс, 2021.
2. Voevodin, V. V., Chulkevich, R. A., Kostenetskiy, P. S., Kozyrev, V. I., Maliutin, A. K., Nikitenko, D. A., Rykovanov, S. G., Shamsutdinov, A. B., Shkandybin, Y. N., & Zhumatiy, S. A. (2021). Administration, Monitoring and Analysis of Supercomputers in Russia: a Survey of 10 HPC Centers. *Supercomputing Frontiers and Innovations*, 8(3), 82–103. <https://doi.org/10.14529/jsfi210305>
3. Kostenetskiy P., Chulkevich R., Kozyrev V., Shamsutdinov A. [HPC TaskMaster - Task Efficiency Monitoring System for the Supercomputer Center](#) // *Communications in Computer and Information Science*. 2022