



Московский государственный университет имени М.В.Ломоносова  
Факультет вычислительной математики и кибернетики  
Кафедра суперкомпьютеров и квантовой информатики

# **Переносимая система визуализации данных о задачах, выполняемых на суперкомпьютере**

ВЫПОЛНИЛ:  
ЛАТЫПОВ ШАМИЛЬ ИЛЬДАРОВИЧ

НАУЧНЫЙ РУКОВОДИТЕЛЬ:  
ВЕД.Н.С. НИВЦ МГУ  
К.Ф.-М.Н. АЛЕКСАНДР СЕРГЕЕВИЧ АНТОНОВ

# Введение

Суперкомпьютеры играют ключевую роль в современных исследованиях, однако оптимизация и эффективность приложений являются вызовом. На данный момент существует инструмент JobDigest для анализа работы всех или отдельно взятых задач, выполняемых на суперкомпьютере Lomonosov-2. Для отображения данных используется три основных типа представления:

1. Таблица Lomonosov-2 Task Table, отображающая информацию по всем задачам с суперкомпьютера Ломоносов-2.
2. Показатели по каждой задаче по отдельности с детальными графиками с показателями датчиков.
3. Отрисовка тепловых карт по каждому сенсору для всех задач по отдельности.

# Постановка задачи

Цель: разработка и реализация переносимой системы визуализации данных о задачах, выполняемых на суперкомпьютере, для упрощения анализа и оптимизации процессов, связанных с выполнением задач.

Для достижения поставленной цели необходимо выполнить следующие задачи:

1. Изучение баз данных MongoDB и InfluxDB, используемых для хранения метаданных и показателей датчиков соответственно.
2. Разработка представления данных в виде таблицы, агрегирующей всю необходимую информацию о задачах, выполняемых на суперкомпьютере, наподобие общей таблицы Lomonosov-2 Task Table из JobDigest.
3. Разработка панелей и таблиц в Grafana, отображающих данные отдельно по задачам.
4. Реализация способов экспорта данных из таблиц без потери информации.

# Grafana

Программная система для визуализации и мониторинга данных, ориентированная на данные систем ИТ-мониторинга.

В работе Grafana применяется для разработки переносимых панелей, отображающих данные по задачам.



# Базы данных

## InfluxDB

Высокопроизводительная база данных для управления временными рядами, специально разработанная для работы с метриками и событиями.

Для разработки переносимой системы эта база данных используется для отбора показателей датчиков по задачам и узлам, а также передачи этих значений в панели и таблицы в Grafana.

## MongoDB

Документоориентированная система управления базами данных, которая предоставляет высокую гибкость и масштабируемость. MongoDB хранит данные в виде гибких документов в формате BSON (бинарный JSON), что считается стандартным расширением текстового файла для хранения данных.

При реализации решения отображения данных MongoDB используется как хранилище метаданных по программам, выполняемых на суперкомпьютере Ломоносов-2.

## **Данные по всем задачам**

- Получение данных из баз данных.
- Применение функций преобразований к таблице для изменения отображаемых значений.
- Настройка цветовой разметки ячеек в таблице.
- Оптимизация запросов к базам данных для ускорения работы и чтения данных.

Таблица имеет название Lomonosov-2 Task Table

# Применённые параметры в таблице Lomonosov-2 task table:

1. Пересчёт значений показателей: используется операция умножения числовых значений времени начала и конца работы программы на 1000 для перевода времени из секунд в микросекунды, а также деление продолжительности работы на 60 для перевода из секунд в минуты.
2. Конвертирование типов данных: строковое значение JobID, которое считывается из базы данных InfluxDB, конвертируется в числовое значение (необходимо для преобразования в п.3).
3. Объединение таблиц от двух баз данных в одну по JobID через функцию Merge, которая работает как JOIN-объединение для реляционных баз данных.
4. «Организация» полей:
  - Расставление столбцов в нужном порядке;
  - Скрытие лишних столбцов (\_id, t\_start, t\_end, Time);
  - Переименование столбцов t\_start\*1000 в t\_start и t\_end\*1000 в t\_end (столбцы образуются после п.1 с применением функции пересчёта значений)
5. Функция фильтрации данных: удаляет пустые строки, в которых столбцы с метаданными не имеют значений.

47.1	0.99	0	11.1	0	7.27
None	None	None	None	None	None
49.9	0	6450346	12.1	117768	231174
None	None	None	None	None	None
None	None	None	None	None	None
None	None	None	None	None	None
None	None	None	None	None	None
None	None	None	None	None	None
49.2	0	4153536	12.0	66163	223613
49.8	0	25869238	12.0	0	0.628
49.8	0	35152825	12.0	0	0.659
None	None	None	None	None	None
4.57	70.0	0	1.00	1001000	1007564

Рис. 1. Цветовая разметка по показателям датчиков

## Цветовая разметка

Цветовая кодировка для наглядного представления различных показателей и упрощения восприятия информации.

<u>1554908</u>	RUNNING	<u>gavrilov</u>
<u>1554907</u>	COMPLETED	<u>ermakov</u>
<u>1554906</u>	FAILED	<u>wasabiko</u>
<u>1554905</u>	COMPLETED	<u>ermakov</u>
<u>1554904</u>	CANCELLED	<u>gavrilov</u>
<u>1554903</u>	COMPLETED	<u>ermakov</u>
<u>1554902</u>	CANCELLED	<u>gavrilov</u>
<u>1554901</u>	COMPLETED	<u>gavrilov</u>

Рис. 2. Визуализация статуса задачи

1. Разметка по показателям датчиков, которая в зависимости от числовых значений окрашивает ячейки в 3 цвета: зелёный, жёлтый и красный (рис. 1).
2. Отображение определённого цвета в зависимости от статуса задачи (рис. 2).



# Данные из InfluxDB

Чтение данных происходит с помощью языка запросов InfluxQL, похожий на SQL. В таблице Lomonosov-2 Task Table используется запрос, который с помощью вложенных функций отбирает средние показатели датчиков, вычисляет среднее арифметическое и группирует данные по ID задач.

Было проведено много экспериментов с разными видами запросов, и результаты показали, что запрос в данном виде (рис. 3) имеет наибольшую эффективность для отбора требуемых данных в нужной форме.

```
SELECT
    mean("cpu_avg") as cpu_user,
    mean("loadavg") as loadavg,
    mean("gpu_load") as gpu_load,
    mean("ib_rcv_data_mpi") as ib_rcv_data_mpi,
    mean("lustre_read_bytes") as lustre_read_bytes,
    mean("lustre_write_bytes") as lustre_write_bytes
FROM
    (SELECT "avg" as cpu_avg FROM data WHERE "sensor" = 'cpu_user'),
    (SELECT "avg" as loadavg FROM data WHERE "sensor" = 'loadavg'),
    (SELECT "avg" as gpu_load FROM data WHERE "sensor" = 'gpu_load'),
    (SELECT "avg" as ib_rcv_data_mpi FROM data WHERE "sensor" = 'ib_rcv_data_mpi'),
    (SELECT "avg" as lustre_read_bytes FROM data WHERE "sensor" = 'lustre_read_bytes'),
    (SELECT "avg" as lustre_write_bytes FROM data WHERE "sensor" = 'lustre_write_bytes')
WHERE $timeFilter AND "jobid" <> "
GROUP BY "jobid"
```

Рис. 3. Запрос данных на языке InfluxQL

# Общая таблица по задачам

Lomonosov-2 task table														
jobid	state	account	t_start	t_end	num_cores	duration (min)	partition	cpu_user	gpu_load	ib_rcv_data_mpi	loadavg	lustre_read_bytes	lustre_write_bytes	
1558023	TIMEOUT	artemglova_250883	2023-04-27 12:56:46.0...	2023-04-27 13:02:13	70	5.45	test	None	None	None	None	None	None	
1558022	COMPLETED	shrpv_2209	2023-04-27 12:55:46.0...	2023-04-27 12:55:52	14	0.100	test	None	None	None	None	None	None	
1558021	CANCELLED	artemglova_250883	2023-04-27 12:54:46.0...	2023-04-27 12:55:38	70	0.867	test	31.4	0	9234592	1.64	0	16.0	
1558020	CANCELLED	chertov_old	2023-04-27 12:54:46.0...	2023-04-27 12:56:17	210	1.52	test	None	None	None	None	None	None	
1558019	FAILED	artemglova_250883	2023-04-27 12:50:35.0...	2023-04-27 12:50:42	70	0.117	test	None	None	None	None	None	None	
1558018	FAILED	artemglova_250883	2023-04-27 12:49:01.0...	2023-04-27 12:49:11	70	0.167	test	None	None	None	None	None	None	
1558017	FAILED	artemglova_250883	2023-04-27 12:47:58.0...	2023-04-27 12:48:15	70	0.283	test	None	None	None	None	None	None	
1558016	CANCELLED	artemglova_250883	2023-04-27 12:46:45.0...	2023-04-27 12:47:24	70	0.650	test	None	None	None	None	None	None	
1558015	CANCELLED	artemglova_250883	2023-04-27 12:45:45.0...	2023-04-27 12:46:44	70	0.983	test	86.4	0	33245882	9.32	83810	43513	
1558014	FAILED	chertov_old	2023-04-27 12:44:19.0...	2023-04-27 12:44:35	210	0.267	test	None	None	None	None	None	None	
1558013	CANCELLED	artemglova_250883	2023-04-27 12:43:16.0...	2023-04-27 12:43:39	70	0.383	test	None	None	None	None	None	None	
1558011	COMPLETED	mazalevaolya_2237	2023-04-27 13:47:34.0...	2023-04-27 13:53:31	14	5.95	compute_prio	None	None	None	None	None	None	
1558009	FAILED	artemglova_250883	2023-04-27 12:38:34.0...	2023-04-27 12:38:42	70	0.133	test	0.283	0	2853	1.11	21.7	30.3	
1558008	FAILED	chertov_old	2023-04-27 12:37:44.0...	2023-04-27 12:38:01	210	0.283	test	None	None	None	None	None	None	
1558007	FAILED	artemglova_250883	2023-04-27 12:37:33.0...	2023-04-27 12:37:42	70	0.150	test	0.0693	0	3.00	1.11	0	16.0	
1558006	FAILED	artemglova_250883	2023-04-27 12:36:29.0...	2023-04-27 12:36:39	98	0.167	test	None	None	None	None	None	None	
1558005	COMPLETED	shrpv_2209	2023-04-27 12:35:58.0...	2023-04-27 12:36:05	14	0.117	test	None	None	None	None	None	None	
1558004	FAILED	artemglova_250883	2023-04-27 12:35:26.0...	2023-04-27 12:35:35	98	0.150	test	None	None	None	None	None	None	
1558003	FAILED	artemglova_250883	2023-04-27 12:34:44.0...	2023-04-27 12:34:58	70	0.233	test	None	None	None	None	None	None	
1558002	COMPLETED	shrpv_2209	2023-04-27 12:32:50.0...	2023-04-27 12:32:57	14	0.117	test	0.00416	0	0	0.0293	0	14.1	
1558001	FAILED	shrov_2209	2023-04-27 12:29:42.0...	2023-04-27 12:29:46	14	0.0667	test	None	None	None	None	None	None	

Рис. 4. Общая таблица по задачам

# Данные по каждой задаче или отдельному узлу

В работе реализованы специальные панели и таблицы, которые отображают данные по конкретной задаче или отдельному узлу.

Всего 4 дополнительные панели:

1. Панель с таблицей, представляющей из себя список узлов.
2. Панель с детальными графиками с значениями датчиков на конкретный узел за заданный промежуток времени, а также список задач, выполненных на этом узле.
3. Панель с одной таблицей, отображающей список задач, для которых доступен подробный просмотр данных.
4. Панель с несколькими таблицами, демонстрирующих время начала и конца времени работы программы, все метаданные по задаче, графики с показателями всех датчиков по времени, список затрагиваемых узлов, а также минимальные, средние и максимальные значения датчиков за всё время выполнения программы.

# Переносимость панелей и таблиц

## Экспорт:

1. Открыть панель, которую необходимо перенести.
2. В меню настроек панели выбрать JSON представление.
3. Скопировать содержимое JSON модели.
4. Сохранить конфигурацию в отдельный файл с расширением .json, либо сохранить текст с конфигурацией

## Импорт:

1. Открыть в меню Grafana меню для импорта таблиц.
2. Загрузить файл с расширением .json или вставить json-код в поле для ввода текста.
3. Завершить процесс импорта, нажав по соответствующей кнопке Import.

# Результаты

- На основе системы Grafana разработана переносимая система визуализации данных.
- Реализованы панели, содержащие таблицы с данными как каждой задачи по отдельности, так и всех задач.
- Разработаны и протестированы методы экспорта и импорта панелей в Grafana, обеспечивающие возможность сохранения и обмена конфигурациями таблиц.

Таким образом, удалось разработать и реализовать переносимую систему визуализации данных о задачах, выполняемых на суперкомпьютере Ломоносов-2. Благодаря этому, пользователи смогут анализировать различные показатели о выполняемых на суперкомпьютере программах с помощью удобного и гибкого инструмента визуализации данных.