

HPC TaskMaster – система мониторинга эффективности задач суперкомпьютера

П.С. Костенецкий, А.Б. Шамсутдинов, Р.А. Чулкевич, В.И. Козырев

Национальный исследовательский университет "Высшая школа экономики"

HPC TaskMaster – система мониторинга суперкомпьютера *cHARISMa* НИУ ВШЭ, которая автоматически распознает неэффективные задачи пользователей, позволяя существенно экономить дорогостоящее машинное время. Пользователи могут просматривать отчеты о выполнении своей задачи вместе с интерактивными графиками. Данная система построена на базе открытого программного обеспечения, что позволяет установить ее на других вычислительных кластерах.

Ключевые слова: суперкомпьютер, мониторинг, эффективность

1. Введение

Система мониторинга эффективности задач является важным инструментом для обнаружения некорректно запущенных вычислений, влекущих за собой недостаточно эффективное использование ресурсов суперкомпьютера. В данной работе описывается новая система мониторинга эффективности задач *HPC TaskMaster*, разработанная в Национальном исследовательском университете «Высшая школа экономики» для суперкомпьютера *cHARISMa* (Computer of HSE for Artificial Intelligence and Supercomputer Modelling). Данная система предназначена для мониторинга задач пользователей суперкомпьютера и обнаружения неэффективно используемых ресурсов.

Разработанная система позволяет пользователям просматривать отчеты о выполнении их задач вместе с интерактивными графиками выполнения, а также автоматически определять задачи, которые работали неэффективно. Имея доступ к результатам анализа своих задач, в дальнейшем пользователи могут запускать свои задачи более эффективно, что позволит существенно экономить машинное время суперкомпьютера.

Среди ошибок пользователей суперкомпьютера, наиболее распространенными являются:

- выделение недостаточного или избыточного объема ресурсов для задачи;
- запуск на CPU задач, поддерживающих вычисления на GPU;
- запуск непараллельной задачи на нескольких ядрах CPU или графических процессоров;
- выделение мощностей вычислительного узла без запуска расчетов.

Ключевой особенностью суперкомпьютера НИУ ВШЭ является то, каким образом он выделяет ресурсы для задач пользователей. Вместо выделения целого вычислительного узла для одной задачи, пользователю дается определенное количество ядер процессора и GPU, тем самым на вычислительном узле могут выполняться сразу несколько десятков задач, оптимизируя тем самым использование ресурсов суперкомпьютера. Из-за описанной выше особенности *cHARISMa* уже разработанные системы мониторинга задач, такие как отечественная система *JobDigest* [1] или зарубежная *Savio* [2] не подходят по главному критерию: данные системы предназначены для работы в режиме, когда для задачи пользователя выделяется целый вычислительный узел. По этой причине для *cHARISMa* была разработана собственная система мониторинга задач, с использованием открытого программного обеспечения [3].

2. Описание суперкомпьютера sHARISMa

Высокопроизводительный вычислительный кластер *sHARISMa* занимает 6 место по производительности в списке ТОП-50 суперкомпьютеров СНГ по состоянию на 2021 год. Ресурсы вычислительного кластера предназначены для поддержки проведения фундаментальных научных исследований и организации учебного процесса, а также для выполнения научных и научно-практических проектов, требующих использования суперкомпьютерных систем. На данный момент, более 500 пользователей работают на суперкомпьютере по 193 научным и учебным проектам НИУ ВШЭ.

Таблица 1. Технические характеристики суперкомпьютера «sHARISMa»

Количество узлов/CPU/ядер/GPU	40/80/1816/116
Модель процессоров	Intel Xeon Gold 6248R, 6240R, 6152
Модель GPU	NVIDIA Tesla V100 32 GB NVLink
Оперативная память кластера	34 ТБ
Параллельная система хранения данных	СХД на базе Lustre 840 ТБ
Тип вычислительной сети	InfiniBand EDR (2x100 Гбит/с)
Тип управляющей сети	Gigabit Ethernet
Пиковая производительность	1 Петафлопс
Производительность в LINPACK	653,7 Терафлопс
Операционная система	Linux Centos 7.6

Подробные характеристики суперкомпьютера описаны в статье [4]. В 2021 планируется расширение суперкомпьютера НИУ ВШЭ новыми типами вычислительных узлов с 8 графическими картами NVIDIA Tesla A100 80GB SXM в каждом, что позволит существенно увеличить количество проводимых научных исследований мирового уровня.

3. Проектирование системы

Для проектирования системы мониторинга эффективности задач были определены следующие функциональные требования.

1. Система должна собирать следующие данные для каждой задачи:
 - утилизация конкретных ядер CPU, выделенных для задачи;
 - утилизация GPU, на которых выполняется задача;
 - утилизация видеопамати GPU;
 - энергопотребление GPU;
 - утилизация оперативной памяти, создаваемая задачей;
 - использование сети *InfiniBand*;
2. Система должна анализировать собранные данные и на их основе определять, эффективно ли работала задача;

3. Система должна предоставлять пользователям суперкомпьютера доступ к списку выполненных задач, отчет об их выполнении и графики их выполнения при помощи веб-приложения;

Опираясь на приведенные выше функциональные требования, был проведен анализ работающих инструментов для мониторинга на *cHARISMa*, которые могут быть использованы для новой системы. Для запуска задач на суперкомпьютере используется планировщик задач *Slurm*. Данные о задачах *Slurm* сохраняются двумя способами: основные данные записываются в реляционную базу данных *MySQL* при помощи фоновой программы *slurm database*, а метрики задач записываются в базу данных временных рядов *InfluxDB* при помощи плагина *acct gather*. Данный плагин собирает такие данные, как утилизация CPU, утилизация памяти, использование сети *InfiniBand*, при этом он сохраняет метрики для каждой отдельной задачи.

Для сбора недостающих показателей утилизации конкретных ядер центральных процессоров и графических процессоров был установлен фоновый процесс *Telegraf* [5], который имеет встроенные плагины для сбора метрик утилизации ядер центральных и графических процессоров. Таким образом, имея ID центральных процессоров и графических процессоров, назначенных для задачи, система может собрать метрики для задействованных компонентов, вследствие чего может проанализировать эффективность выполнения задачи.

Собранные метрики при помощи *Telegraf* и *acct gather* сохраняются в БД *InfluxDB* [6].

В качестве инструмента для визуализации графиков на суперкомпьютере *cHARISMa* установлена *Grafana* [7]. *Grafana* предоставляет большие возможности для настройки и оформления графиков, а также имеет поддержку создания их при помощи API. Данный API позволяет автоматизировать создание графиков для каждой задачи.

Преимуществом использования комбинации *Telegraf*, *InfluxDB* и *Grafana* является возможность установки и настройки данных инструментов на любом суперкомпьютере. Помимо этого, данные инструменты делают систему мониторинга достаточно гибкой. При необходимости, можно собирать дополнительные данные, используя встроенные плагины *Telegraf* или разработав свои.

Проанализировав существующие инструменты для мониторинга, можно обозначить следующие нефункциональные требования для системы.

1. Сбор метрик должен осуществляться при помощи фоновой программы *Telegraf* и плагина *Slurm acct gather*;
2. Метрики должны храниться в базе данных временных рядов *InfluxDB*;
3. Визуализация метрик должна происходить при помощи веб-интерфейса *Grafana*;
4. Основные данные о задачах должны собираться из БД *Slurm*;
5. Программный код системы мониторинга должен быть разработан на языке Python.

Диаграмма компонентов системы *HPC TaskMater* представлена на рис. 1.

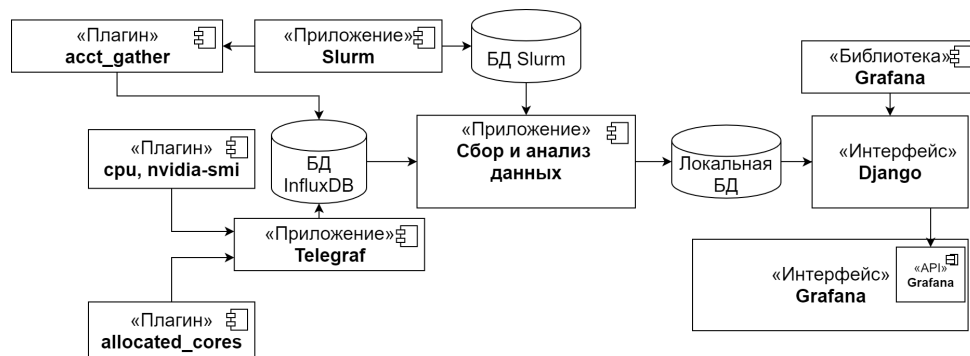


Рис. 1. Диаграмма компонентов системы

4. Алгоритм работы системы

В данном разделе приведены основные алгоритмы работы системы *HPC TaskMaster*.

4.1. Алгоритм сбора и анализа статистики системы

Алгоритм работы системы мониторинга эффективности задач представлен на рис. 2.

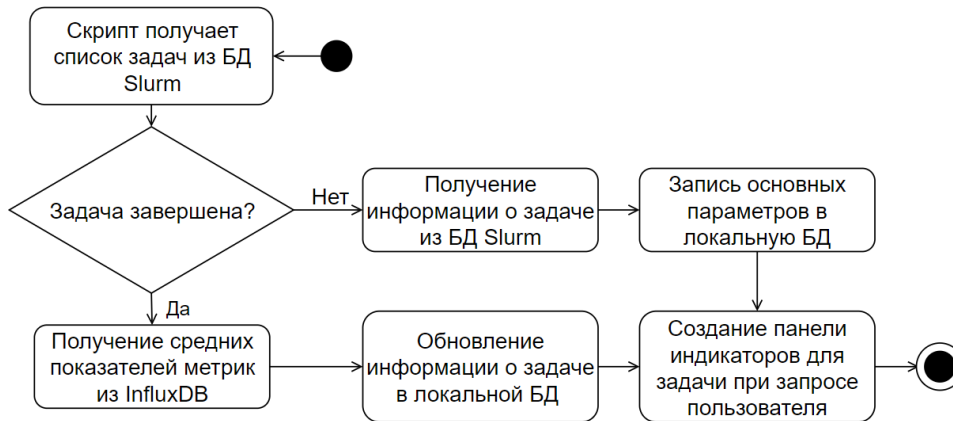


Рис. 2. Алгоритм работы системы мониторинга эффективности задач

Система мониторинга эффективности задач работает по следующему принципу:

- основной скрипт системы мониторинга эффективности задач собирает список запущенных задач из базы данных Slurm;
- если задача запущена и не занесена в новую базу данных, то информация о ней записывается в новую базу данных;
- если задача выполнена, то для нее собираются средние показатели таких значений, как утилизация центральных и графических процессоров, сети *InfiniBand*, после чего информация о задаче обновляется в новой базе данных;
- на основе имеющихся данных о задаче, при запросе пользователя для нее автоматически строится панель индикаторов в Grafana.

4.2. Алгоритм работы автоматической генерации панели индикаторов в Grafana

Для автоматического создания панели индикаторов для каждой задачи в *Grafana*, был разработан специальный алгоритм, представленный на рис. 3.

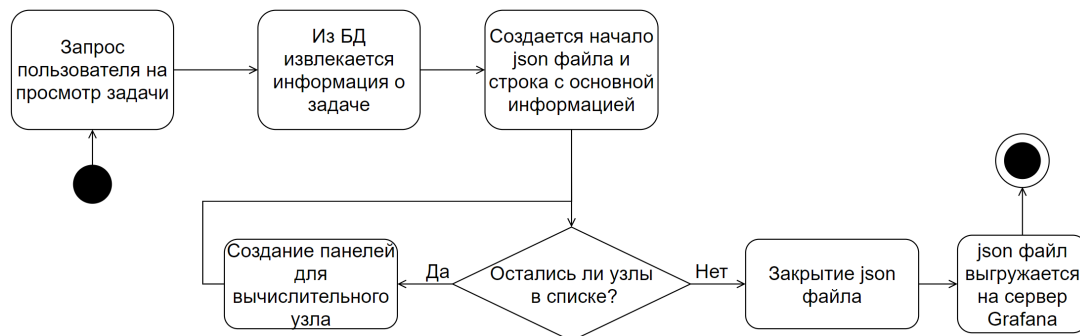


Рис. 3. Алгоритм работы автоматической генерации панели индикаторов

Генерация панели индикаторов в Grafana работает по следующему принципу:

- пользователь в системе выбирает свою задачу и запускает процесс создания панели индикаторов;
- из новой базы данных извлекается необходимая информация о задаче: ID задачи, список узлов, ID центральных и графических процессоров;
- генерируется начало json файла для панели индикаторов;
- алгоритм обходит каждый вычислительный узел в списке и создает для него панели с графиками из готовых json шаблонов;
- когда узлы заканчиваются, json файл закрывается и выгружается на сервер *Grafana*.

В дальнейшем, созданная панель индикаторов используется для отображения на сайте системы при помощи технологии *iframe*, где пользователь может в интерактивном режиме ознакомиться с графиками, просматривая показатели за определенный промежуток времени. Также, система создает панель индикаторов как для выполненной, так и для работающей задачи, тем самым пользователь имеет возможность наблюдать работу своей задачи в реальном времени.

5. Обнаружение неэффективных задач при помощи системы

Анализ эффективности происходит после выполнения задачи, когда система подсчитывает и сравнивает утилизацию с эталонными значениями. На данный момент, в системе реализовано распознавание следующих типов проблем:

- низкая утилизация как минимум одного из ядер центрального процессора или графического процессора во время выполнения задачи;
- низкая утилизация ядер центрального процессора при высокой утилизации графических процессоров;
- низкая утилизация графических процессоров при высокой утилизации ядер центрального процессора;
- низкое использование сети *InfiniBand* при использовании нескольких узлов.

В системе выводится список задач с проблемами. При клике на задачу пользователь попадает на панель индикаторов, где видны все графики загрузки, созданные для конкретно данной задачи и описание найденных проблем. Администратор системы имеет возможность задать эталонные значения, на основе которых система будет делать вывод, работает ли задача неэффективно. В списке задач пользователь может просмотреть отчет о выполнении своей задачи, а также графики для нее. Пример отчета о выполнении задачи приведен на рис. 4.

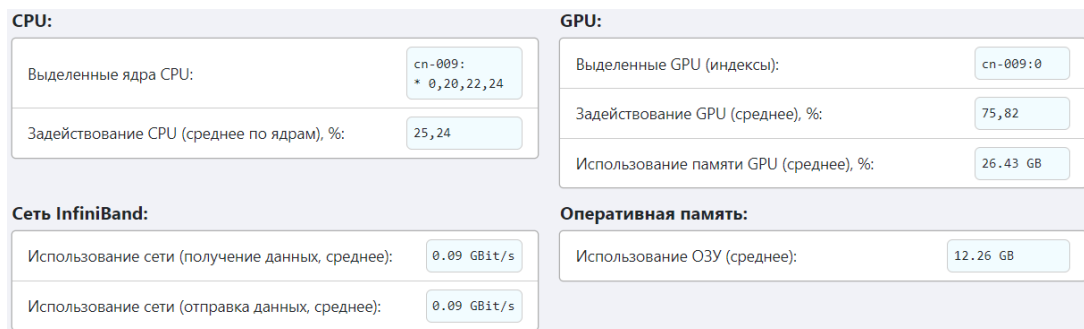


Рис. 4. Фрагмент отчета о выполнении задачи

Пример панели с графиками задачи приведен на рис. 5.



Рис. 5. Графики утилизации вычислительных ресурсов задач

6. Заключение

Разработанная система мониторинга эффективности задач *HPC TaskMaster* позволяет пользователям суперкомпьютера просматривать информацию о своих задачах, а также обнаруживает случаи, когда выделенные задачей ресурсы были использованы неэффективно. Благодаря использованию в качестве подсистем открытого программного обеспечения, система отличается гибкостью, и может быть внедрена на различные суперкомпьютеры.

В качестве дальнейшего развития системы планируется: 1) расширить список отслеживаемых проблем в задачах пользователей, 2) внедрить анализа временных рядов при помощи математических методов или путем использования нейронных сетей, 3) подготовить систему для публикации в открытом репозитории.

Исследование выполнено с использованием суперкомпьютерного комплекса *sHARISMa* НИУ ВШЭ [4].

Литература

1. Nikitenko D. и др. JobDigest – Detailed System Monitoring-Based Supercomputer Application Behavior Analysis // Communications in Computer and Information Science. Springer Verlag, 2017. Т. 793. С. 516–529.
2. Chan N. A resource utilization analytics platform using grafana and telegraf for the Savio supercluster // ACM International Conference Proceeding Series. Association for Computing Machinery, 2019.
3. Шамсутдинов А.Б., Костенецкий П.С. Разработка системы мониторинга эффективности задач на суперкомпьютере sHARISMa // Параллельные вычислительные технологии ПаВТ'2021, г. Волгоград.
4. Kostenetskiy P.S., Chulkevich R.A., Kozyrev V.I. HPC Resources of the Higher School of Economics // Journal of Physics: Conference Series. 2021. Т. 1740, № 1.
5. Rattanatamrong P. и др. Overhead study of telegraf as a real-time monitoring agent // JCSSE 2020 - 17th International Joint Conference on Computer Science and Software Engineering. Institute of Electrical and Electronics Engineers Inc., 2020. С. 42–46.
6. Fadhel M., Sekerinski E., Yao S. A comparison of time series databases for storing water quality data // Advances in Intelligent Systems and Computing. 2019. Т. 909.
7. Beermann T. и др. Implementation of ATLAS Distributed Computing monitoring dashboards using InfluxDB and Grafana // EPJ Web Conf. EDP Sciences, 2020. Т. 245. С. 03031.