

Разработка подсистемы анализа эффективности использования вычислительных ресурсов для системы HPC TaskMaster*

П.С. Костенецкий, А.Б. Шамсутдинов, Р.А. Чулкевич, В.И. Козырев
Национальный Исследовательский Университет «Высшая Школа Экономики»

Обнаружение неэффективных задач на суперкомпьютере является одной из наиболее актуальных проблем в суперкомпьютерной отрасли, так как такие задачи могут блокировать ценные вычислительные ресурсы и приводить к общему замедлению работы всех пользователей. Существует множество критериев, по которым можно определить, что задача работает неэффективно – самым простым из них является сравнение средних показателей использования выделенных ресурсов с эталонным значением, которое считается границей определения эффективности задачи. Однако, такие критерии помогают отследить только самые простые примеры неэффективных задач. В данной статье предлагаются новые способы определения неэффективных задач, имплементированные в систему мониторинга эффективности задач *HPC TaskMaster* НИУ ВШЭ. Данные способы позволяют отслеживать запуск непараллельных программ, обнаруживать разбалансировку вычислительных ресурсов, а также определять задачи, создающие простой части выделенных им вычислительных ресурсов.

Ключевые слова: суперкомпьютер, мониторинг, эффективность.

1. Введение

Обнаружение неэффективных задач является актуальной проблемой для любых суперкомпьютерных центров. Под *неэффективными задачами* будем подразумевать такие задачи, которые используют выделяемые им вычислительные ресурсы ниже определенного эталонного уровня. Неэффективные задачи не только выполняются медленно, но и зачастую замедляют работу всех остальных пользователей суперкомпьютера, занимая ресурсы и затормаживая продвижение задач в очереди. Помимо регулярного расширения суперкомпьютера, для борьбы с нехваткой ресурсов, при условии постоянной расширяющейся базы пользователей, необходимо автоматически обнаруживать и устранять неэффективные задачи при помощи системы мониторинга, чтобы экономить дорогостоящее машинное время.

Система *HPC TaskMaster*, развернутая на суперкомпьютере НИУ ВШЭ «сHARISMa» [1], была специально разработана для обнаружения неэффективных задач [2,3], используя лучшие практики как от зарубежных [4], так и от отечественных аналогов [5–7]. Такие задачи определяются на основе анализа свойств задачи и агрегированных метрик использования ее ресурсов. На данный момент система *HPC TaskMaster* автоматически определяет наиболее распространенные проблемы в вычислительных задачах. Подсистема анализа эффективности использования вычислительных ресурсов позволяет определить, эффективно ли работала задача, используя для этого такие сущности, как индикаторы, теги и выводы, которые будут рассмотрены в данной статье.

2. Обзор системы HPC TaskMaster

Система мониторинга HPC TaskMaster была разработана в 2021 году отделом суперкомпьютерного моделирования НИУ ВШЭ [8] и постоянно расширяется. На данный момент более 400 активных пользователей имеют возможность посмотреть подробные отчеты

*Исследование выполнено с использованием суперкомпьютерного комплекса НИУ ВШЭ

о выполнении своих задач вместе с интерактивными графиками и результатами анализа. В данной главе описываются основные сущности, используемые системой для обнаружения неэффективных задач – *индикаторы*, *теги* и *выводы*, которые будут использованы в дальнейшем для внедрения новых способов обнаружения неэффективных задач.

2.1. Индикаторы

Для определения эффективности работы задачи, необходимо оценить использование выделенных для нее ресурсов. Для этой цели, было введено такое понятие как *индикатор*. Индикатор – это величина, используемая для оценки загрузки компонентов, участвующих в задаче. Индикаторы принимают значения от 0 (максимальное использование вычислительных ресурсов) до 1 (наоборот). Значение индикатора l_j вычисляется по формуле (1)

$$l_j^k = 1 - \frac{\lambda_j^k - a_j}{b_j - a_j}, \quad l_j \in [0, 1], \quad (1)$$

где a_j , b_j – заданные параметры, отвечающие за минимальное и максимальное значение j -того элемента вектора агрегированных метрик. Агрегированные метрики $\lambda_1^k, \dots, \lambda_m^k$ вычисляются для всех компонентов задачи, и включают в себя средние, минимальные и максимальные значения временных рядов.

Индикатор определяется путем задания нижней и верхней границы, их совокупность используется для нормирования значения индикатора. Существуют два типа индикаторов – индикатор нижней и верхней границ. *Индикатор нижней границы* принимает максимальное значение (единица) при значении показателя равным нижней границе. *Индикатор верхней границы* принимает максимальное значение (единица) при значении показателя равным верхней границе. Данные два типа индикатора необходимы для оценки различных показателей, например, эффективная вычислительная задача должна активно использовать оперативную память (здесь используется индикатор нижней границы), но если задача слишком часто обращается к дисковой системе хранения данных, то ее эффективность падает (в этом случае используется индикатор верхней границы). Список ранее разработанных индикаторов системы HPC TaskMaster представлен в работе [2].

2.2. Теги

Теги задачи – это 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. Теги позволяют легко внедрять новые свойства в систему выводов.

Примеры тегов приведены в таблице 1.

Таблица 1. Теги системы HPC TaskMaster

№ пп.	Название тега	Критерии назначения тега
1	Задача выполнялась дольше 7 дней	Длительность задачи более 7 дней
2	Задача завершена с ошибкой	Задача завершена с одним из аварийных статусов
3	Задача запущена на 1 ядре	Для задачи выделено 1 ядро
4	Тип задачи – salloc	Задача запущена через salloc/srun
5	Тип задачи – Jupyter Notebook и др.	Определение типа задачи по ключевым словам

2.3. Выводы

Вывод об эффективности задачи – это финальный результат анализа задачи, предоставляемый пользователю на основе показателей имеющихся индикаторов и тегов. Принцип формирования вывода показан на рисунке 1.



Рис. 1. Принцип формирования выводов

Рассмотрим его подробнее:

- на входе есть N временных рядов (результаты мониторинга выполнения задачи на вычислительном 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);
- все результаты функций собираются 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 приложения jupyter-notebook»).

Новые индикаторы и теги, о которых пойдет речь в этой статье, будут использованы, чтобы дополнить уже имеющиеся выводы, сделать их более точными и надежными.

3. Обнаружение неэффективных задач

В данной главе предлагаются способы для обнаружения двух новых типов неэффективных задач – непараллельных вычислительных задач, запущенных в параллельном режиме, а также задач, вызывающих простой вычислительных ресурсов. Выводы об эффективности таких задач будут производиться как при помощи индикаторов, в которых будут храниться значения уровня проявления проблемы, так и при помощи тегов.

Временные ряды, соответствующие использованию вычислительных ресурсов пользовательскими задачами, хранятся в базе данных InfluxDB. Данные собираются при помощи плагина Telegraf, установленного на всех вычислительных узлах каждые 10 секунд. Далее данные агрегируют: посчитывается среднее значение за каждую астрономическую минуту. Таким образом, количество точек во всех временных рядах вычислительной задачи становится одинаковым, более того у всех временных рядов задачи используется одна и та же временная шкала. Это позволяет в дальнейшем проводить сравнительный анализ временных рядов.

3.1. Индикатор низкой корреляции использования вычислительных ресурсов

Как правило, причиной появления в потоке задач суперкомпьютера непараллельных задач являются либо ошибки, допущенные пользователем при постановке задачи в очередь планировщика, либо запуск изначально непараллельной задачи на несколько ядер процессора или даже на GPU. Ключевым критерием, по которому можно определить, является ли задача непараллельной – неравномерное использование выделенных ресурсов. Если посмотреть на графики использования ресурсов у непараллельной задачи, то можно заметить, что некоторые графики будут сильно отличаться друг от друга по причине того, что разные ядра CPU или GPU выполняют за этот период времени разные задачи. Соответственно, чтобы определить непараллельную задачу, необходимо сравнить временные ряды вычислительных ресурсов между собой и выявить, какие из них выбиваются из «общей картины». Стоит отметить, что временные ряды CPU и GPU сравниваются отдельно, и для них будут созданы свои индикаторы.

Сравнивая индикаторы с аналогичными сущностями в других системах анализа задач, как например JobDigest [5], можно выделить наличие у индикаторов системы HPC TaskMaster значений. Значения позволяют более гибко настраивать итоговый вывод о работе задачи.

Пример графиков использования CPU непараллельной задачей, запущенной на двух ядрах, приведен на рисунке 2. В данном примере видно, что загрузка переключалась с одного ядра на другое, а не использовала параллелизм. В случае с параллельной задачей, графики загрузки были бы визуальнo схожи.

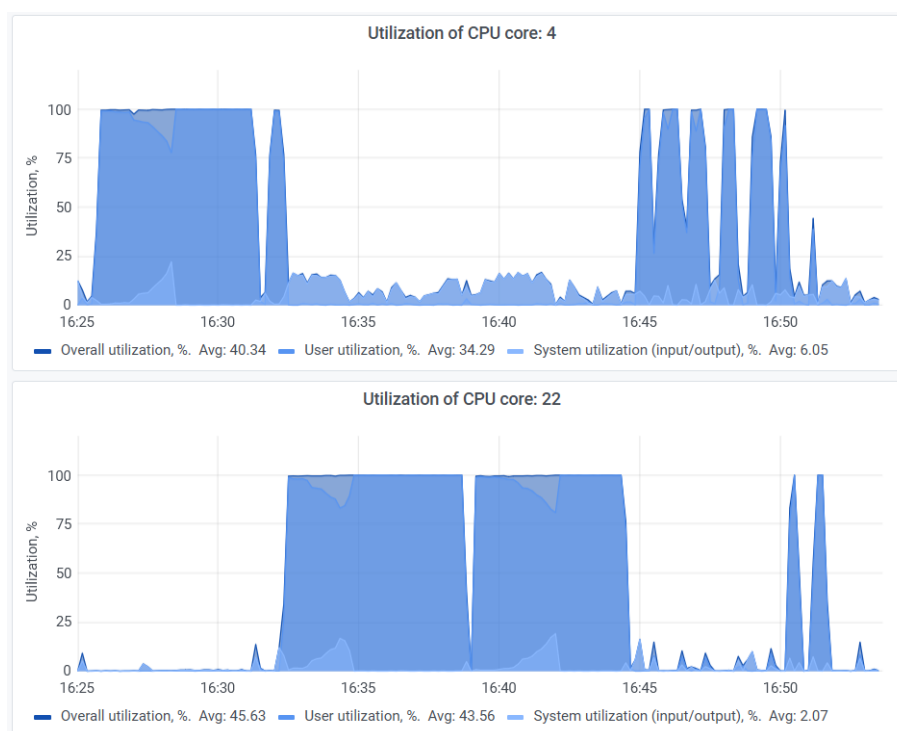


Рис. 2. Пример: непараллельная задача запущена на двух ядрах CPU

Так как исследуемые значения временных рядов обычно не соответствуют нормальному закону распределения (предварительно временные ряды проверяются на нормальность распределения по критерию Шапиро-Уилка), в качестве способа проводить сравнение временных рядов был выбран подсчет корреляции Спирмена [9]. Коэффициент корреляции Спирмена применяется для исследования меры линейной связи между случайными вели-

чинами, измеренных в метрических шкалах на одной и той же выборке. Он позволяет определить, насколько пропорциональна изменчивость двух переменных. Для подсчета корреляция Спирмена используется встроенная функция библиотеки pandas, которая возвращает значение коэффициента корреляция в пределах от -1 до 1 . В приведенном на рисунке 2 примере корреляция Спирмена между временными рядами ядер составляет -0.75 , т.е. временные ряды практически полностью не совпадают друг с другом.

К недостаткам использования корреляции Спирмена при сравнении временных рядов использования можно отнести то, что при наличии большого количества шумов (в первую очередь это касается графиков загрузки CPU) корреляция может быть слабой, несмотря на визуальную схожесть графиков. По этой причине будем учитывать только отрицательные коэффициенты корреляции. Коэффициент корреляция вычисляется для всех пар временных рядов. Значение уровня индикатора подсчитывается как отношение количества пар временных рядов с отрицательной корреляцией менее -0.5 к общему количеству всех пар. Таким образом, для рассмотренной задачи, выполняющейся на двух ядрах центрального процессора, значение уровня индикатора для каждого ядра CPU будет приравнено к 1.0 .

В данном примере индикатор свидетельствует об обнаружении задачи, не поддерживающей параллелизм, но по ошибке, запущенной на нескольких ядрах. Такие задачи произвольно перемещаются операционной системой между свободными ядрами CPU, а реальная загрузка создается только на одно из ядер.

3.2. Тег неравномерно распределенных ресурсов

При запуске параллельной задачи на нескольких узлах опытный пользователь вычислительного кластера старается равномерно распределить нагрузку между узлами, что означает, что на каждый узел выделено одинаковое количество ядер CPU и GPU. Новый тег «Неравномерно распределенные ресурсы» позволяет сразу определить задачи, чьи ресурсы были неправильно распределены между выделенными узлами. Опыт показал, что практически всегда такие задачи имеют проблемы с использованием выделенных вычислительных ресурсов.

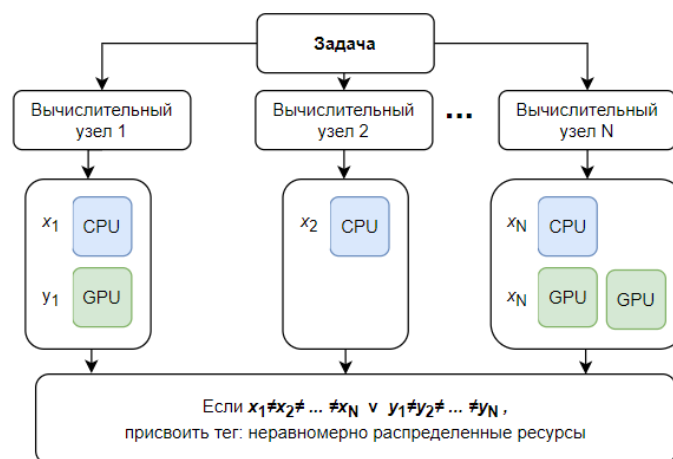


Рис. 3. Принцип формирования тега неравномерно распределенных ресурсов

3.3. Тег нечетного количества CPU и GPU

Из-за особенностей работы параллельных программ, задачи с четным количеством CPU и GPU как правило выполняются быстрее по сравнению с нечетным количеством. Например, в тесте производительности LINPACK, задача на 16 ядер даст лучший результат, чем задача на 17 ядер, за счет распределения ядер в виде матрицы размером 4×4 против мат-

рицы 1×17 . По этой причине, для задач с нечетным количеством ядер (не считая задач на одном ядре) был создан отдельный тег.

3.4. Тег неделимости нацело количества ядер CPU и GPU

Еще одним способом обнаружения некорректно запущенной параллельной задачи, является подсчет отношения количества ядер CPU к количеству GPU. Как правило, при запуске параллельной задачи, на каждый GPU выделяется одинаковое количество ядер CPU. Если же подсчитанное отношение не является целым числом, то для такой задачи присваивается тег «Неделимость нацело количества ядер CPU и GPU», что свидетельствует о потенциальной проблеме с работой задачи.

3.5. Индикатор простоя выделенных задаче ресурсов

Зачастую, причиной простоя ресурсов является запуск задач пользователя в интерактивном режиме при помощи команды планировщика Slurm «salloc», когда пользователь допускает слишком большие временные промежутки между запусками расчетов или забывает завершить задачу после их окончания. Система *HPC TaskMaster* определяет большинство задач такого типа за счет тега «salloc», определяющего тип задачи, и индикаторов для средних значений использования CPU и GPU. Однако, у данного подхода был недостаток, из-за которого часть задач уходила от внимания системы – так как подсчитываются только средние значения CPU и GPU за время выполнения задачи, а затем сравниваются с пороговыми значениями, то при наличии как высокой загрузки, так и небольших промежутков простаивания, среднее значение может быть в пределах нормы.

Для определения такого типа задач снова проводится работа с временными рядами задачи. Для каждого временного ряда ядер CPU и GPU проводится подсчет значений, меньше порогового, после чего доля простаивания подсчитывается как отношение этих значений к общему количеству точек временного ряда.

На основе полученных результатов формируется индикатор. Пример задачи с простаивающим ресурсом приведен на рисунке 4.

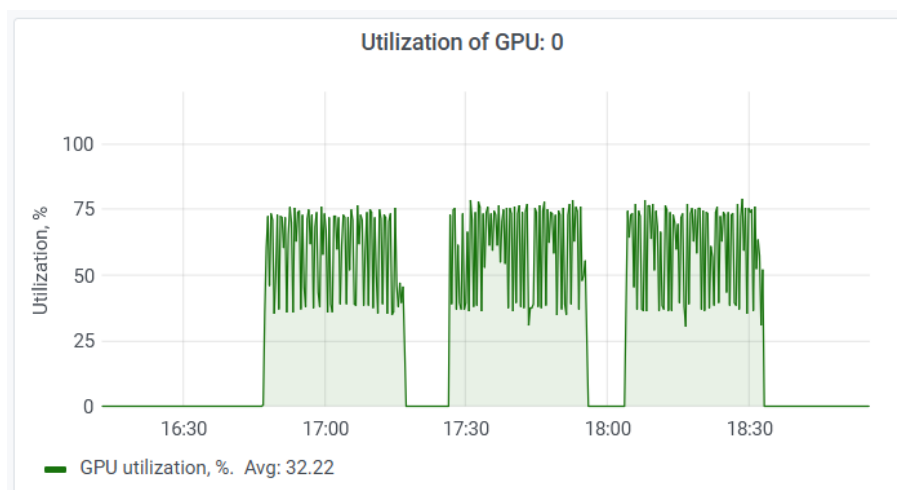


Рис. 4. Пример: GPU простаивает половину времени

Для данной задачи значение уровня индикатора составит 0.21, при заданном минимальном уровне простаивания в 30% и максимальном уровне 100%.

Индикатор простаивания используемых ресурсов в первую очередь будет использоваться для выводов неэффективных задач, запущенных при помощи команды «salloc», где проблема простаивания ресурсов наиболее актуальна.

4. Заключение

Благодаря разработке новых тегов и индикаторов, в системе *HPC TaskMaster* расширился спектр определяемых типов неэффективных задач. Повышение точности формирования выводов о задаче на основе новых индикаторов и тегов позволяет проводить более качественную работу по консультированию пользователей суперкомпьютера и обнаруживать неэффективные запуски. За 2022 год благодаря системе удалось высвободить 7% высокопроизводительных вычислительных ресурсов. В качестве дальнейших планов развития системы планируется внедрение индикаторов, использующих математические методы поиска аномалий временных рядов, а также искусственные нейронные сети.

Система «*HPC TaskMaster*» развернута на суперкомпьютерном комплексе НИУ ВШЭ «сHARISMa» и доступна для всех его пользователей. Система имеет открытый исходный код, доступна в репозитории [10] и может использоваться в других суперкомпьютерных центрах.

Литература

1. Kostenetskiy P.S., Chulkevich R.A., Kozyrev V.I. HPC Resources of the Higher School of Economics // Journal of Physics: Conference Series. 2021.
2. Kostenetskiy P., et al. HPC TaskMaster – Task Efficiency Monitoring System for the Supercomputer Center // Commun. Comput. Inf. Sci. 2022. Vol. 1618 CCIS. P. 17–29.
3. Костенецкий П.С., Козырев В.И., Чулкевич Р.А., Шамсутдинов А.Б. HPC TaskMaster – система мониторинга эффективности задач суперкомпьютера // Суперкомпьютерные дни в России: Труды международной конференции (27-28 сентября 2021 г., Москва). М.: МАКС Пресс, 2021.
4. 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.
5. Nikitenko D. и др. JobDigest – Detailed System Monitoring-Based Supercomputer Application Behavior Analysis // Communications in Computer and Information Science. Springer Verlag, 2017. P. 516–529.
6. Nikitenko D.A., Voevodin V.V., Zhumatiy S.A. Deep analysis of job state statistics on Lomonosov-2 supercomputer // Supercomput. Front. Innov. 2018. Vol. 5, no. 2.
7. Voevodin V.V., et al. Administration, Monitoring and Analysis of Supercomputers in Russia: a Survey of 10 HPC Centers // Supercomput. Front. Innov. 2021. Vol. 8, no. 3.
8. Костенецкий П.С., Шамсутдинов А.Б., Чулкевич Р.А., Козырев В.И. Свидетельство о государственной регистрации программы для ЭВМ «HPC TaskMaster – система мониторинга эффективности задач суперкомпьютера» № 2022682037 от 18.11.2022, правообладатель: Федеральное государственное автономное образовательное учреждение высшего образования «Национальный исследовательский университет «Высшая школа экономики»
9. Гржибовский А.М. Корреляционный анализ // Экология человека. 2008. № 9. URL: <https://cyberleninka.ru/article/n/korrelyatsionnyy-analiz> (дата обращения: 27.01.2023).
10. Open Source / HPC TaskMaster · GitLab [Электронный ресурс]. URL: <https://git.hpc.hse.ru/open-source/hpc-taskmaster> (дата обращения: 13.12.2021).