Методы статистического анализа потока задач большого суперкомпьютерного комплекса*

А.А. Мамаева, Вад.В. Воеводин

Московский государственный университет имени М.В. Ломоносова

В настоящее время большинство суперкомпьютеров работают с низкой эффективностью – производительность многих параллельных приложений составляет менее 5% от пиковой. Для того чтобы бороться с данной проблемой, необходимо полностью контролировать текущее состояние вычислительной системы. Один из аспектов, который необходимо отслеживать, – эффективность использования суперкомпьютерных ресурсов. Для решения этой задачи в данной работе выполняется исследование особенностей, которые приводят к снижению общей эффективности функционирования вычислительной системы. Исследование проводится с помощью статистического анализа данных системного мониторинга, собираемых по всему потоку задач. На основе разработанных средств выполнен подробный анализ использования вычислительных ресурсов суперкомпьютера «Ломоносов» за период с мая 2015 года по декабрь 2016 года.

Ключевые слова: суперкомпьютер, анализ эффективности, поток задач

1. Введение

Эффективное использование суперкомпьютеров является крайне актуальной задачей в области высокопроизводительных вычислений. Однако на данный момент большинство пользователей задействуют предоставленные им ресурсы недостаточно эффективно, в результате чего большой процент вычислительных ресурсов постоянно простаивает. Поэтому необходимо разработать инструментарий, который позволил бы системному администратору и руководству суперкомпьютерного центра получать исчерпывающую картину происходящего на вычислительной установке в целом и ее компонентах, что позволит обнаруживать особенности поведения отдельных задач, пользователей или использования разделов, которые приводят к снижению эффективности работы суперкомпьютеров.

Для того чтобы решить данную проблему, необходимо, в частности, полностью контролировать поведение общего потока задач, выполняющихся на суперкомпьютере. Один из возможных методов осуществления такого контроля заключается в сборе и анализе информации о динамике выполнения всех суперкомпьютерных приложений с помощью системы мониторинга.

На основе этого метода и построено представленное в данной работе исследование. В рамках этой работы выполняется поиск, анализ и визуализация полезной информации о различных аспектах поведения общего потока задач, основанный на изучении статистики использования суперкомпьютера. Для удобного отображения и изучения полученной информации разработан web-сайт, который позволяет получить доступ ко всему разработанному функционалу.

Различные методы представления и визуализации делают возможным исследование информации с разных сторон. Они позволяют, например, сравнить поведение ресурсоемких и небольших приложений, исследовать активность отдельных пользователей и оценивать корректность использования того или иного раздела, обнаруживать пользователей, задействующих ресурсы наименее эффективно. Большой интерес также представляет динамика изменения получаемых данных. Изучение данного вопроса позволяет понять, каково

* Исследование проводится при финансовой поддержке стипендии Президента РФ (СП-1981.2016.5).

распределение загрузки суперкомпьютера с течением времени, существуют ли тенденции к повышению эффективности использования предоставляемых ресурсов и так далее.

Отметим, что данный подход не направлен на подробное изучение отдельных приложений, поскольку для этих целей уже существует множество хороших инструментов – отладчики, средства трассировки и профилирования программ и т.д.

Необходимость получения подобной информации нередко возникает и в работе других суперкомпьютерных центров, поэтому существует ряд исследований, направленных на решение схожих задач. Однако на текущий момент эта задача решена далеко не полностью. Некоторые работы изначально разработаны для применения на одном выбранном суперкомпьютере и на данный момент не являются переносимыми ([1], [2]). Другие решения из области суперкомпьютерных технологий либо обладают недостаточно богатым функционалом в рамках поставленной задачи (например, системы мониторинга Ganglia [3] и Cacti [4]), либо предоставляются на коммерческой основе (в частности, менеджер суперкомпьютерных систем Bright Manager [5]). Стоит отметить, что в данных системах анализ статистики использования ресурсов не является основной задачей; готовых переносимых инструментов, направленных непосредственно на решение подобной задачи в суперкомпьютерной области, на данный момент не найдено.

Разработанный подход был использован для подробного изучения статистических данных о функционировании суперкомпьютера «Ломоносов» [6], однако он может быть легко применен для анализа работы других вычислительных систем.

Далее в главе 2 приведено описание предложенных методов и разработанных программных средств для анализа и отображения данных по потоку задач. Глава 3 содержит описание основных результатов, полученных на суперкомпьютере «Ломоносов» с использованием разработанных средств. В последней главе описаны возможные дальнейшие направления работ по данному направлению и приведено заключение.

2. Анализ структуры потока задач

2.1 Описание предложенных методов анализа структуры потока задач

Изучение структуры потока задач основано на анализе данных от системного мониторинга. Эти данные описывают различные свойства выполнения программ: загрузку процессора, число операций чтения/записи в память в секунду, объем переданных байт в секунду, среднюю загрузку системы (loadavg) и так далее. Система мониторинга собирает эти данные от системных и аппаратных датчиков на каждом вычислительном узле суперкомпьютера: информация о работе с памятью собирается от процессорных аппаратных датчиков; данные по использованию сети получаются от сетевой карты на узле; данные по загрузке процессора и загрузке системы – от операционной системы на узле. Запуск любой задачи описывается интегральными значениями (максимум, минимум, медиана) по каждой динамической характеристике, что, по нашему мнению, позволяет получить необходимую информацию для проведения анализа всего потока задач при небольшом общем объеме данных.

Для подсчета среднего значения динамической характеристики по всему потоку приложений необходимо выбрать, каким образом усреднять получаемые значения. Для этого нужно определить, как учитывать влияние отдельной задачи. Запускаемые приложения сильно отличаются как по количеству используемых процессоров, так и по длительности работы, поэтому, на наш взгляд, наиболее объективной оценкой является количество затраченных ядро-часов. Например, подсчет среднего значения по характеристике X выглядит следующим образом:

$AVG = \frac{\sum_i X_i * Y_i}{\sum_i Y_i}$, где X_i – медиана характеристики X по каждой отдельной задаче, а Y_i – вес для данной задачи согласно ядро-часам. В некоторых случаях для подробного изучения потока задач предусмотрена возможность выбора минимума или максимума вместо медианы.

Далее возникает наиболее сложный вопрос – каким образом на основе полученного набора «сырых» данных по задачам получить интересную информацию о структуре потока? Нами было проведено множество экспериментов с различными методами анализа и отображения

информации. В результате этого был составлен набор таких методов, которые, по нашему мнению, представляют наибольший интерес и позволяют достаточно полно взглянуть на различные аспекты поведения потока задач. Далее перечислены шесть основных выбранных нами методов.

1. Изучение распределения значений динамических характеристик во всем потоке задач. Данное распределение отражает, какое значение является преобладающим для каждой характеристики и насколько велик разброс данных. Представляет интерес изучение по любой характеристике, поскольку позволяет обнаруживать аномальные отклонения, часто свидетельствующие о наличии проблем с эффективностью. Отметим, что здесь задача состоит в обнаружении только базовых типов аномалий; для определения более сложных случаев в НИВЦ МГУ ведется разработка подходов к поиску аномалий на основе методов машинного обучения [7].

2. Исследование корреляции динамических характеристик на основе сравнения данных по каждой задаче. Данный метод направлен на поиск взаимосвязей между различными характеристиками, что позволит лучше понимать характер выполнения задач – в частности, поможет выделять типовые шаблоны поведения задач. Поиск корреляции выполняется на основе коэффициента Пирсона [8].

3. Изучение списка пользователей с максимальными значениями (топ пользователей). Данный метод позволяет выполнять анализ как по отдельным динамическим характеристикам, так и по некоторому их подмножеству. Примером может служить поиск пользователей, одновременно активно использующих коммуникационную сеть и имеющих оптимальные значения loadavg. Другой пример – анализ низких показателей по тем же параметрам, что характеризует такие приложения как неподходящие для выполнения на суперкомпьютере. Это позволяет проанализировать, насколько эффективно и равномерно пользователи используют ресурсы относительно друг друга.

4. Сравнение различных типов приложений. Это показывает взаимосвязь между значениями динамических характеристик, количеством выделенных процессоров, длительностью работы задачи. Примером является сравнение приложений, запущенных на большом и малом числе процессоров, по количеству затраченных ими вычислительных ресурсов.

5. Сравнение показателей по разделам. Данный метод, основанный на изучении значений динамических характеристик приложений в разных разделах, позволяет сравнить, насколько эффективно (и корректно ли в целом) используется тот или иной раздел суперкомпьютера. Такое сравнение позволяет, например, определять, все ли задачи, запускаемые в специализированном разделе с графическими ускорителями, активно используют данные ускорители. На данный момент этот метод не позволил выявить существенных результатов, однако исследования по данному направлению планируется продолжить.

6. Исследование динамики изменения показателей. В данном случае выполняется сравнение результатов по любому из описанных выше пунктов за выбранные периоды времени. Это позволяет с разных сторон отследить динамику изменения работы вычислительной системы. В частности, это позволяет понять, используются ли вычислительные ресурсы более эффективно с течением времени.

Анализ указанных «срезов» общей информации о состоянии суперкомпьютера позволяет с разных сторон оценивать эффективность использования ресурсов в целом и по отдельным пользователям или задачам в отдельности, а также определять сбалансированность работы суперкомпьютера. Также подобная информация призвана помочь оценить качество применения новой политики или квоты на ресурсы, поскольку она позволяет проанализировать эффективность работы суперкомпьютера до и после ее применения.

В большинстве случаев анализ выполняется на основе средневзвешенных значений по характеристикам. Однако в некоторых случаях изучение максимумов и минимумов помогает по-новому взглянуть на данные и исследовать другие аспекты, влияющие на эффективность использования ресурсов. Диаграмма на рис. 1 отражает максимальное число промахов в кэш первого уровня по всем задачам каждого пользователя, по каждой отдельной задаче выбрана медиана. Пример показывает, что задачи трех первых пользователей заметно отличаются от других. Это не позволяет с уверенностью говорить о проблемах с эффективностью в

программах данных пользователей, однако такие большие значения сигнализируют о возможно низкой локальности обращений в память (поскольку очень часто происходят промахи в кэш-память даже первого уровня). В таком случае стоит более детально изучить данные приложения, и при необходимости рекомендуется оповестить соответствующих пользователей.

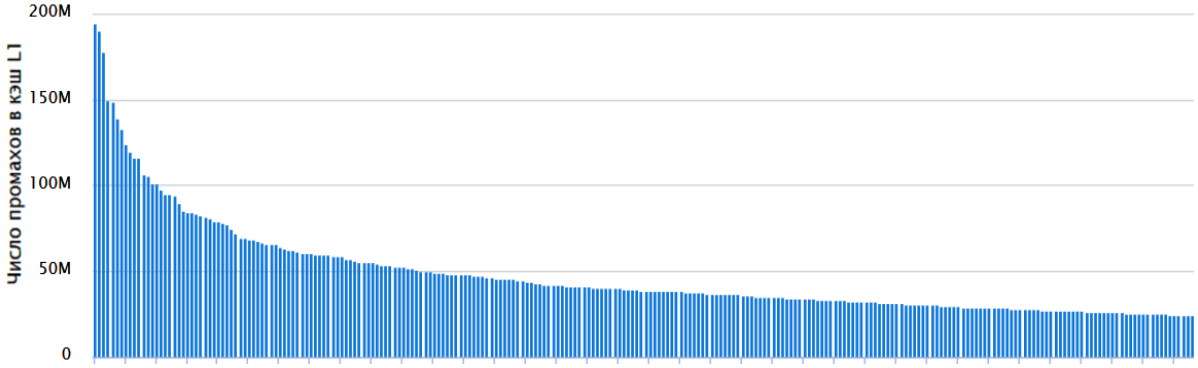

Рис. 1. Топ 250 пользователей (за все время исследования) по числу промахов в кэш первого уровня. На диаграмме столбцами представлены максимальные значения числа промахов в кэш-память первого уровня (L1) каждого пользователя, по каждой отдельной задаче выбрана медиана.

2.2 Web-интерфейс для отображения и анализа статистических данных

Все разработанные средства анализа и визуализации полученных статистических данных реализованы в рамках отдельного web-сайта, на котором отображаются основные интересующие нас аспекты поведения потока задач.

Web-интерфейс реализован с использованием языков программирования PHP и Javascript. Он предоставляет гибкую функциональность и позволяет настраивать большое количество параметров (таких, как интервал времени, за который показывать данные, характер усреднения динамических характеристик), ограничивать по значению отдельные характеристики или выбирать разделы, в которых работают пользователи. Web-сервер получает необходимые данные из базы данных MySQL, в которой хранится собранная системой мониторинга информация. Для каждой задачи сохранена информация о пользователе, времени работы, разделе запуска, выделенном числе процессоров, а также значения динамических характеристик. Как было сказано ранее, каждая характеристика представлена в базе следующими интегральными значениями: медиана, минимум и максимум.

Каждый разработанный метод анализа реализован в виде отдельной страницы на сайте. После выбора необходимых параметров из базы данных выбирается требуемая информация, которая затем обрабатывается и отображается в удобном формате. Для построения различных видов графиков и диаграмм используется библиотека Highcharts [9], предоставляющая широкий функционал.

Для подробного изучения поведения отдельных приложений приводятся ссылки на отчеты, созданные с помощью разработанного коллективом НИВЦ МГУ программного инструмента JobDigest [10]. Данный инструмент позволяет подробно анализировать поведение динамических характеристик (таких как пользовательская загрузка процессора, число операций чтения/записи из памяти в секунду, количество переданных по сети байт в секунду) во время выполнения программы. Для этого в отчете выполняется визуализация различных временных графиков по каждой характеристике, а также показывается общая информация по интегральным значениям характеристик для исследуемой задачи.

3. Результаты исследования потока задач суперкомпьютера «Ломоносов»

Исследование было проведено на данных, полученных системой мониторинга суперкомпьютера «Ломоносов» в период с 05.2015 до 12.2016. Всего было проанализировано более 150.000 задач, выполненных на данном суперкомпьютере. Проведенное исследование

позволило выявить ряд особенностей и тенденции в использовании ресурсов приложениями различного типа и размера.

В разделах 3.1 – 3.3 приведен анализ эффективности работы суперкомпьютера в целом, далее рассмотрены результаты исследования отдельных динамических характеристик.

3.1 Анализ потребления вычислительных ресурсов

Количество затраченных ядро-часов является одной из важных для рассмотрения характеристикой. Это объясняется тем, что если для приложения выделено большое число процессоров или оно работает длительный период времени, но при этом показывает низкую эффективность, то это означает, что большая часть ресурсов тратится впустую.

Первое, что было рассмотрено для общей оценки, – как изменялось с течением времени распределение ядро-часов по разделам (в случае изучения динамики изменения значений некоторой характеристики, по горизонтальной оси графика расположены месяцы запуска приложений). Диаграмма на рис. 2 показывает, что самыми востребованными для вычислений являются узлы с двумя четырехъядерными процессорами. Это вполне закономерно, поскольку наибольший раздел в суперкомпьютере «Ломоносов» построен на процессорах данного типа. На долю вычислений на графических ядрах приходится в среднем около 15-25%. Разница в пропорциях связана в первую очередь с изменением числа узлов в разделах (вследствие периодического отключения части узлов и затем повторного введения в счетное поле), поскольку доступные вычислительные ресурсы в течение всего рассматриваемого временного интервала были постоянно загружены.

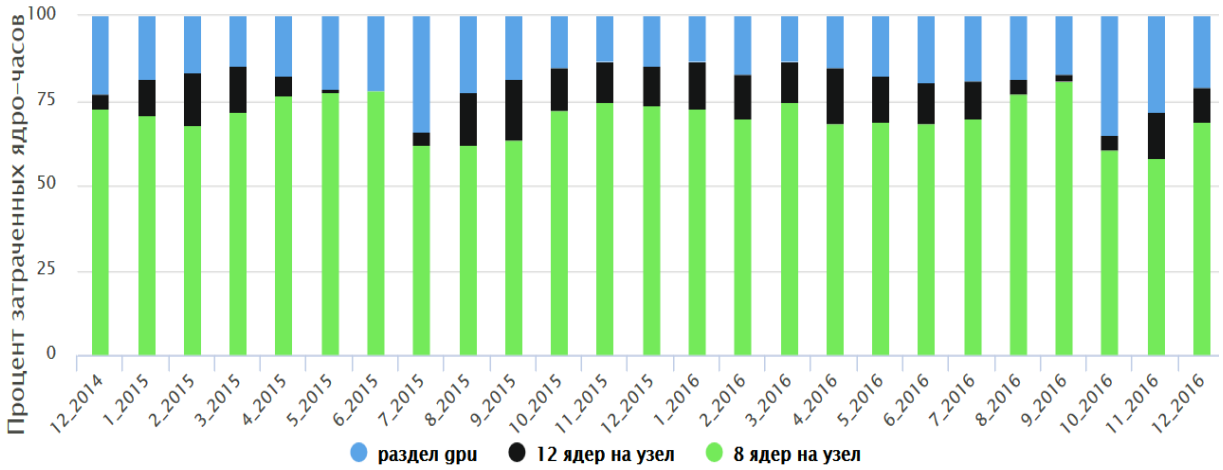

Рис. 2. График распределения затрат ядро-часов между разделами с разным числом узлов в зависимости от месяца запуска приложений (по горизонтальной оси).

Особое внимание нужно уделять эффективности работы больших по количеству выделенных ядер задач, так от момента запуска и до завершения они занимают большее количество ресурсов.

Сравнение значений затраченных ядро-часов больших (использующих более 512 ядер) и остальных задач в зависимости от месяца запуска задачи приведено на рис. 3. Из рисунка видно, что доля больших задач меняется существенно, но в целом достаточно велика – в среднем около 40% от общего потребления ядро-часов.

При этом анализ статистики показывает, что 90% всех затраченных ядро-часов за все время приходится на первые 12.04% задач из топа по затраченным ядро-часам. Динамика изменения этой характеристики отражена на рис. 4. Для сравнения, аналогичные исследования для суперкомпьютера Blue Waters [1] выявили значения 3% и 7% на различных типах узлов.

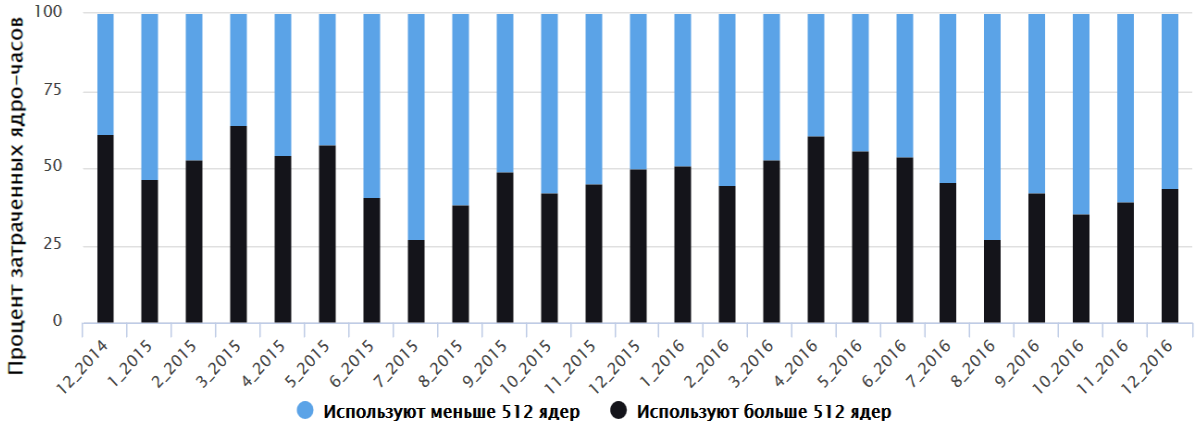

Рис. 3. Распределение затраченных ядро-часов между различными по количеству выделенных ядер задачами, в зависимости от месяца запуска приложений (по горизонтальной оси).

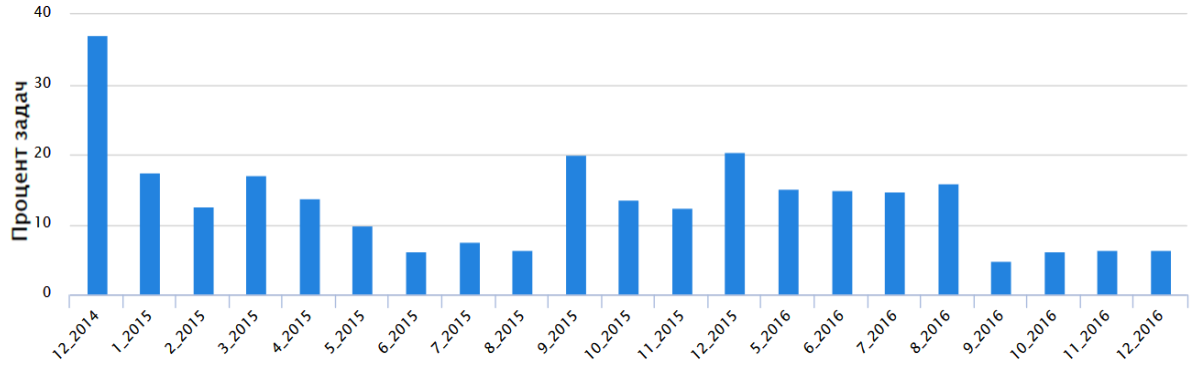

Рис. 4. Диаграмма, отражающая какой процент задач из топа по затраченных ядро-часом составляет 90% от суммарного числа ядро-часов в зависимости от месяца запуска приложений (по горизонтальной оси).

Интересно отметить, что первый 1% задач из топа затраченных ядро-часов составляет 33.177% всех ядро-часов за все время. Получается, что 1 из 12 процентов, указанных выше, затрачивает 33 из 90 процентов ядро-часов. Данное значение варьируется от 16% до 61% в зависимости от месяца запуска.

При изучении результатов работы отдельных пользователей, стало известно, что самые большие суммарные значения за все время имеют 2 пользователя, затратившие 6.881% и 6.213% всех ядро-часов (табл. 1). Это является высоким показателем, так как общее число рассмотренных за это время пользователей более 700. Значение достаточно быстро убывает: пользователи со значением менее 1% расположены, начиная с 22 места, менее 0.1% – со 154 места.

Таблица 1. Топ10 пользователей по количеству затраченных ядро-часов за все время исследования.

	Количество ядро-часов	Процент от всех ядро-часов	Количество задач
1	19015825	6.88	1419
2	17169402	6.21	3850
3	10994678	3.98	290
4	10031963	3.63	433
5	8189509	2.96	1694
6	6808331	2.46	3721
7	5690386	2.06	287
8	5312657	1.92	760
9	4922911	1.78	435
10	4530435	1.64	187

3.2 Изучение средней загрузки системы (Load Average)

Значение load average показывает среднее число процессов, готовых для выполнения за определенный период времени (в данном случае за последнюю минуту). Эта характеристика является одним из возможных показателей того, насколько хорошо задача подходит для вычисления на суперкомпьютере и в частности для работы в конкретном разделе. Если значение меньше числа доступных на узле ядер, то это может означать недостаточную загруженность вычислительных ресурсов.

График распределения значений loadavg узлов с 8 и 12 ядрами (рис. 5) показывает, что наиболее стабильно пользователи загружают восьмиядерные узлы – среднее значение loadavg в этом случае постоянно около 8, что в целом является хорошим показателем. Двенадцатиядерные узлы используются менее эффективно: loadavg не достигает числа доступных ядер ни в один из месяцев. Loadavg узлов с графическими процессорами с середины 2016 года имеет тенденцию к плавному увеличению значения.

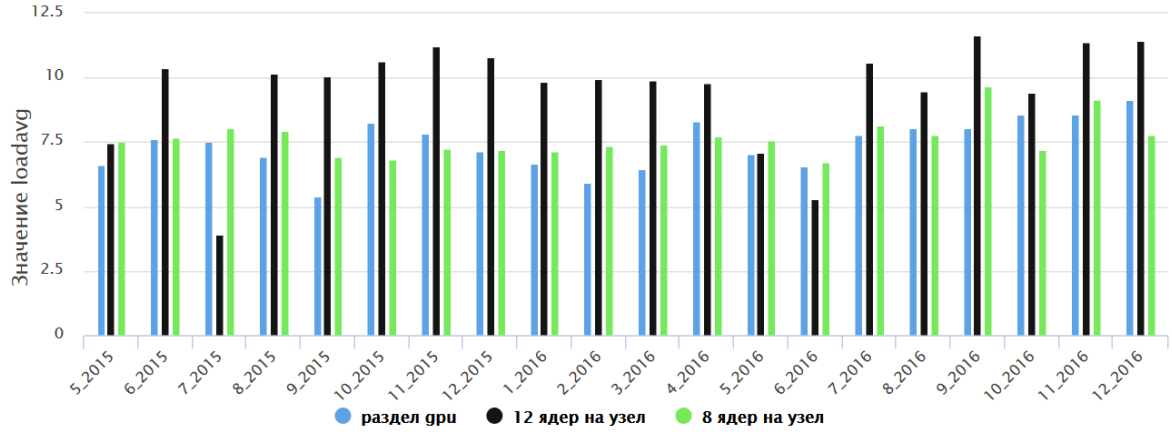

Рис. 5. График распределения значения loadavg для различных типов узлов в зависимости от месяца запуска приложений (по горизонтальной оси).

Достаточно высокий процент – 11.78% (87 из 739) – рассмотренных пользователей имеют средневзвешенное значение loadavg < 1 по всем их задачам. Но при этом они суммарно затрачивают только 0.48% всех ядро-часов, а значит их влияние в целом невелико. Однако, на работу четырех пользователей из них (табл. 2), наиболее активных в плане потребления ядро-часов (> 100000), необходимо обратить внимание, и понять причины неэффективности их приложений. В частности, среднее значение loadavg у первого пользователя (0.0705) очень низко – в абсолютном большинстве случаев подобное значение говорит о некорректной работе программы, при этом в данном случае это значение является средним по всем задачам пользователя. Необходимо более детальное изучение запусков данного пользователя на основе других инструментов анализа эффективности отдельных приложений.

Таблица 2. Топ4 пользователей со средним loadavg < 1 и с числом ядро-часов > 100 000.

	Количество ядро-часов	Среднее значение loadavg
1	394839	0.07
2	325393	0.26
3	148221	0.61
4	180754	0.75

3.3 Анализ наиболее ресурсоемких задач

Исследование поведения ресурсоемких задач требует отдельного детального рассмотрения, поскольку эффективность их выполнения может очень сильно влиять на общую эффективность суперкомпьютера в целом. В данной работе к такому классу задач было решено относить приложения, которые демонстрируют активное использование коммуникационной сети и

значение loadavg, близкое к оптимальному (loadavg = a * (число ядер на узле), где 0.9 < a < 2), при этом будем рассматривать средневзвешенное значение loadavg отдельно для каждого типа узлов. Необходимо определить, какое использование коммуникационной сети можно считать активным. В этом исследовании было решено сравнивать среднее число полученных байт в секунду (ib_rcv_data) каждой задачи со значением k*max(ib_rcv_data), где максимум берется по всем задачам, а k – некоторый подобранный коэффициент. Таким образом, в отличие от выбора фиксированной константы учитывается то, насколько высоких показателей использования коммуникационной сети удается достичь реальным пользователям суперкомпьютера.

При выборе k = 0.1 (что соответствует на данный момент около 300МБ в секунду) получаем список всего из 8 пользователей (табл. 3), что говорит о том, что малое число пользователей использует коммуникационную сеть настолько интенсивно. Первого пользователя можно не рассматривать, поскольку он затратил очень мало ресурсов (всего 3.44 ядро-часов). Пользователи под номерами 2, 3, 7 и 8 показывают хорошую загрузку вычислительных ядер, в то время как пользователи 4 и 5, использующие оба типа узлов, имеют недостаточное значение loadavg для двенадцатиядерных узлов.

Таблица 3. Топ8 пользователей по значению ib_rcv_data (число полученных байт в секунду)

	Количество ядро-часов	Средневзвешенное ib_rcv_data	Средневзвешенное loadavg
1	3	550280900	8 ядер на узел: 4.06
2	62031	512380610	8 ядер на узел: 14.39
3	123024	446031732	8 ядер на узел: 7.78
4	298155	439596771	Все: 7.62 8 ядер на узел: 7.57 12 ядер на узел: 9.56
5	2217416	397876826	Все: 7.64 8 ядер на узел: 7.64 12 ядер на узел: 10.46
6	1595911	346495393	8 ядер на узел: 5.13
7	13678	330604838	8 ядер на узел: 7.90
8	4417	314238042	8 ядер на узел: 7.99

При выборе же k = 0.03 получаем уже список из 120 пользователей, удовлетворяющих такому ограничению значения ib_rcv_data. При этом 39 имеют неоптимальное средневзвешенное loadavg хотя бы для одного из типов узлов по количеству ядер (только у одного пользователя значение больше оптимального, у других – меньше), что сигнализирует о возможной недостаточной загрузке вычислительных ядер 1/3 частью пользователей для данного значения коэффициента k.

Отметим, что подобный анализ призван лишь выявить возможные проблемы с эффективностью и сделать некоторые предположения. Полноценное исследование, которое позволит однозначно говорить о низкой эффективности, должно проводиться с помощью других инструментов, таких как, например, система JobDigest. Это находится вне рамок данного исследования, направленного на изучение потока задач в целом.

3.4 Обнаружение аномальных значений характеристик

Анализ списков пользователей с максимальными значениями по характеристикам во многих случаях позволил выявить наличие небольшого числа пользователей (как правило, от 1 до 5), которые имеют значительно выделяющиеся из всего ряда значения. Наибольшее число пиков и их наибольшее отклонение от среднего значения можно наблюдать в случае, когда из всех задач пользователя выбирается приложение с максимальным значением, а не рассчитывается средневзвешенное значение по всем задачам, так как последний вариант подавляет выбросы. Такие случаи требуют более детального анализа, поскольку сигнализируют об аномальном поведении программ, что зачастую говорит о проблемах с эффективностью.

Также может быть интересным наблюдение максимума не только по всем задачам пользователя, но и внутри отдельных задач. Такие случаи могут свидетельствовать о сбое в работе системных датчиков или системного программного обеспечения, или же об аномально высокой активности приложений. Первые случаи необходимо отслеживать, поскольку их устранение позволяет повышать точность используемых методов и подходов. Выявленным примером такого сбоя является максимальное по задачам значение cpu_user_load = 1451 (эта динамическая характеристика показывает загрузку процессора задачей пользователя и измеряется в процентах, соответственно ее значения должны принадлежать интервалу [0, 100]). Второй тип случаев нужно уметь обнаруживать, поскольку это также зачастую свидетельствует о проблемах с эффективностью.

На рис. 6 представлен пример построения топа пользователей по значению load average на узле, когда внутри задачи и по всем задачам пользователя взяты максимумы. При рассмотрении всех задач пользователя, показанного на диаграмме первым столбцом, было найдено 24 задачи со значением более 100, из них 8 – с loadavg более 1000. В каждом из этих запусков такое высокое значение максимума удерживалось от 1 часа до 3, за этот же период времени среднее значение характеристики было аналогичным.

Оптимальное значение loadavg должно быть примерно равно числу доступных ядер на узле. Таким образом, была найдены задачи, в которых число активных процессов в десятки, а то и сотни раз превышает оптимальное значение. Это очевидно является аномальным поведением и говорит о серьезном снижении эфективности выполнения, поскольку приводит к очень существенным накладным расходам.

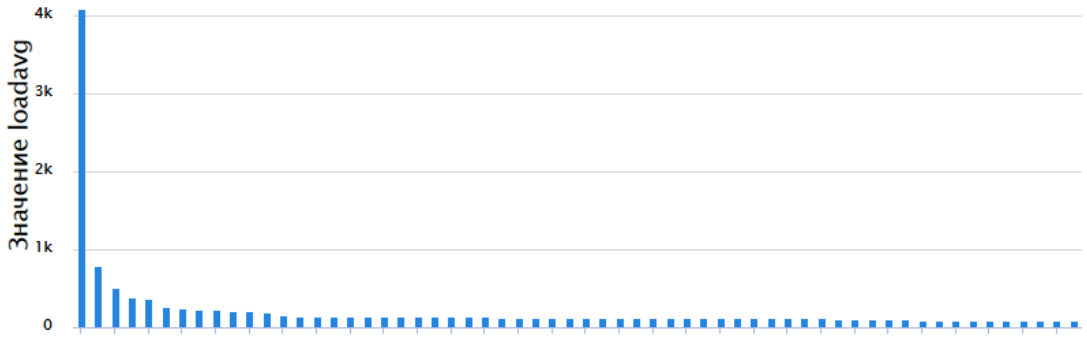

Рис. 6. Начало топа всех пользователей по значению loadavg за все время исследования, столбцами представлены значения характеристики для каждого пользователя.

3.5 Корреляция динамических характеристик

Для изучения взаимосвязи каждой пары отдельных динамических характеристик были использованы коэффициент корреляции Пирсона и коэффициент его значимости по t-критерию Стьюдента [8]. Значение коэффициентов корреляции несколько различны по разделам, что показано в таблице 4. Динамические характеристики, приведенные в таблице: cache_1/cache_3 – количество промахов в кэш-память первого/третьего уровня в секунду; cpu_user – процент времени, в течение которого ядро занято выполнением пользовательской задачи; mem_store/mem_load – количество выполненных микроопераций чтения/записи в секунду; loadavg – среднее число процессов, готовых для выполнения за последнюю минуту; ib_recv – количество принятых байт (data) или пакетов (pckts) в секунду; ib_xmit – количество посланных байт или пакетов в секунду. Пустые клетки в таблице соответствуют коэффициентам корреляции, которые являются незначимыми по критерию Стьюдента, их значения не рассматриваются.

В целом полученные результаты подтверждают предположение о том, что сильно зависимыми являются только пары характеристик, связанные с передачей данных по коммуникационной сети (ib_*), а также чтение/запись из/в память (mem_store и mem_load). Другие характеристики либо слабо коррелируют, либо зависимость полностью отсутствует.

Стоит отметить, что такой подход позволяет оценивать только линейную зависимость между характеристиками. В дальнейшем, вероятно, будут использованы другие метрики для определения более сложной зависимости.

Таблица 4. Пары динамических характеристик, коэффициент корреляции Пирсона для которых не менее 0.6 хотя бы в одном из разделов (regular6, test, gpu_p, reg4prio, hdd4 – разделы суперкомпьютера «Ломоносов»).

		Все		regular6		test		gpu_p		reg4prio		hdd4	
Характеристика 1	Характеристика 2	Число задач	Коэффициент корреляции Пирсона	Число задач	Коэффициент корреляции Пирсона	Число задач	Коэффициент корреляции Пирсона	Число задач	Коэффициент корреляции Пирсона	Число задач	Коэффициент корреляции Пирсона	Число задач	Коэффициент корреляции Пирсона
cache_1	cache_3	81217	0.402	5023	0.488	27668	0.383	138	0.973	1820	0.522	2341	0.476
mem_load	cpu_user	81215	0.663	5024	0.737	27662	0.669	138	0.908	1821	0.638	2341	0.630
mem_load	mem_store	81211	0.841	5024	0.870	27661	0.847	138	0.712	1821	0.693	2341	0.836
mem_load	loadavg	81206	0.444	5024	0.532	27660	0.450		–	1820	0.562	2341	0.662
mem_load	cpu_flops	19447	0.378	772	0.540	10420	0.390		–		–	711	0.392
mem_store	loadavg	81205	0.423	5024	0.556	27659	0.459		–	1820	0.355	2341	0.621
cpu_user	mem_store	81211	0.630	5024	0.728	27661	0.685	138	0.768	1821	0.409	2341	0.583
cpu_user	loadavg	81685	0.578	5025	0.729	27676	0.551		–	1845	0.852	2374	0.863
ib_rcv_data	ib_rcv_pckts	81530	0.733	5024	0.830	27677	0.727	143	0.940	1843	0.883	2365	0.510
ib_rcv_data	ib_xmit_data	81561	0.985	5038	0.997	27685	0.977	143	0.873	1843	0.991	2366	0.894
ib_rcv_data	ib_xmit_pckts	81525	0.733	5024	0.830	27674	0.727	143	0.940	1843	0.883	2365	0.511
ib_rcv_pckts	ib_xmit_data	81530	0.738	5024	0.830	27677	0.738	143	0.952	1843	0.877	2365	0.529
ib_rcv_pckts	ib_xmit_pckts	81525	0.999	5024	0.999	27674	0.999	143	0.999	1843	0.999	2365	0.999
ib_xmit_data	ib_xmit_pckts	81525	0.739	5024	0.830	27674	0.739	143	0.952	1843	0.877	2365	0.529

4. Заключение

В рамках данного исследования разработаны методы для проведения анализа статистических данных по работе структуры потока задач, выполняемых на суперкомпьютере. Эти методы основаны на изучении данных системного мониторинга, которые предоставляют интегральные значения различных динамических характеристик по каждой задаче из данного потока. Был реализован набор методов, направленных на анализ и отображение различных аспектов поведения потока задач, таких как распределение ресурсов между различными типами приложений, динамика изменения эффективности использования вычислительных ядер и коммуникационной сети, аномально высокие значения динамических характеристик в работе приложений некоторых пользователей.

Разработанные методы были применены на практике для анализа потока задач суперкомпьютера «Ломоносов». Были изучены данные по общему потоку задач примерно за 1.5 года его работы, что позволило выявить некоторые интересные особенности. В частности, было проведено исследование наиболее ресурсоемких приложений, обнаружены задачи с аномальным поведением, а также проанализировано распределение затраченных ядро-часов за весь интервал.

Предполагаемые дальнейшие работы по данной тематике направлены на решение нескольких задач. Часть работ будет направлена на расширение предоставляемой функциональности за счет разработки новых методов анализа статистических данных, что поможет оценить другие аспекты поведения потока задач. Предполагается автоматизировать проведение предлагаемого анализа, что позволит оперативно оповещать администраторов системы о наиболее важных полученных результатах. Также планируется применить разработанные методы и на других вычислительных системах.

Литература

1. Jones M. D. et al. Workload Analysis of Blue Waters //arXiv preprint arXiv:1703.00924. – 2017.
2. Evans T. et al. Comprehensive resource use monitoring for hpc systems with tacc stats //Proceedings of the First International Workshop on HPC User Support Tools. – IEEE Press, 2014. – С. 13-21.

3. Massie M. L., Chun B. N., Culler D. E. The ganglia distributed monitoring system: design, implementation, and experience //Parallel Computing. – 2004. – Т. 30. – №. 7. – С. 817-840.
4. Система мониторинга Cacti. URL: http://www.cacti.net/documentation.php (дата обращения: 14.04.2017)
5. Bright manager – программное обеспечение для управления кластерными системами. URL: http://www.brightcomputing.com/product-offerings/bright-cluster-manager-for-hpc (дата обращения: 14.04.2017)
6. Практика суперкомпьютера "Ломоносов" / В. В. Воеводин, С. А. Жуматий, С. И. Соболев и др. // Открытые системы. СУБД. — 2012. — № 7. — С. 36–39.
7. Data mining method for anomaly detection in the supercomputer task flow / V. Voevodin, V. Voevodin, Д. Шайхисламов, D. Nikitenko // NUMERICAL COMPUTATIONS: THEORY AND ALGORITHMS (NUMTA–2016): Proceedings of the 2nd International Conference “Numerical Computations: Theory and Algorithms”. — Vol. 1776 of AIP Conference Proceedings. — 2016. — P. 090015–1–090015–4. [DOI]
8. Коэффициент корреляции Пирсона и t-критерий Стьюдента. URL: https://en.wikipedia.org/wiki/Pearson_correlation_coefficient (дата обращения: 10.04.2017)
9. Библиотека для визуализации Highcharts. URL: http://www.highcharts.com/products/highcharts (дата обращения: 10.04.2017)
10. Job digest: an approach to dynamic analysis of job characteristics on supercomputers / A. V. Adinets, P. A. Bryzgalov, V. V. Voevodin et al. // Вычислительные методы и программирование: Новые вычислительные технологии (Электронный научный журнал). — 2012. — Vol. 13. — P. 160–166.

Methods for statistical analysis of large supercomputer job flow

A.A. Mamaeva, Vad.V. Voevodin

Lomonosov Moscow State University

Most of modern supercomputers are being used very inefficiently – many parallel applications is less than 5% of peak performance. To deal with this problem, we need to fully control the current state of computing system. One of the aspects that we need to monitor – the efficiency of supercomputer resource usage, which is the subject of this work. To solve this problem, in this research we study the features that lead to a decrease in the overall supercomputer functioning efficiency. The analysis is performed using statistical analysis of system monitoring data collected throughout the job flow. A detailed analysis of the use of “Lomonosov” supercomputer computing resources was carried out with the help of the developed tools during the period from May 2015 to December 2016.

Keywords: supercomputer, efficiency analysis, job flow

References

1. Jones M. D. et al. Workload Analysis of Blue Waters //arXiv preprint arXiv:1703.00924. – 2017.
2. Evans T. et al. Comprehensive resource use monitoring for hpc systems with tacc stats //Proceedings of the First International Workshop on HPC User Support Tools. – IEEE Press, 2014. – С. 13-21.

3. Massie M. L., Chun B. N., Culler D. E. The ganglia distributed monitoring system: design, implementation, and experience //Parallel Computing. – 2004. – Т. 30. – №. 7. – С. 817-840.
4. Monitoring system Cacti. URL: http://www.cacti.net/documentation.php (accessed: 14.04.2017)
5. Bright manager – software for managing cluster systems. URL: http://www.brightcomputing.com/product-offerings/bright-cluster-manager-for-hpc (accessed: 14.04.2017)
6. Practice of Lomonosov Supercomputer / Vl. V. Voevodin, S. A. Zhumatii, S. I. Sobolev, et al. // Otkrytye sistemy. SUBD [Open systems. DBMS]. — 2012. — № 7. — P. 36–39. (in Russian)
7. Data mining method for anomaly detection in the supercomputer task flow / V. Voevodin, V. Voevodin, D. Shaykhislamov, D. Nikitenko // NUMERICAL COMPUTATIONS: THEORY AND ALGORITHMS (NUMTA–2016): Proceedings of the 2nd International Conference “Numerical Computations: Theory and Algorithms”. — Vol. 1776 of AIP Conference Proceedings. — 2016. — P. 090015–1–090015–4. [DOI]
8. Pearson correlation coefficient and Student's t-distribution: https://en.wikipedia.org/wiki/Pearson_correlation_coefficient (accessed: 10.04.2017)
9. Library for visualization Highcharts. URL: http://www.highcharts.com/products/highcharts (accessed: 10.04.2017)
10. Job digest: an approach to dynamic analysis of job characteristics on supercomputers / A. V. Adinets, P. A. Bryzgalov, V. V. Voevodin et al. // Numerical Methods and Programming: Vychislitel’nye Metody i Programmirovanie (Scientific on-line open access journal). — 2012. — Vol. 13. — P. 160–166.