

## Раздел II. Математическое и программное обеспечение суперкомпьютеров

УДК 004.051

А.С. Антонов, А.М. Теплов

### ИСПОЛЬЗОВАНИЕ ДАННЫХ СИСТЕМНОГО МОНИТОРИНГА ДЛЯ ОПРЕДЕЛЕНИЯ ФАКТОРОВ, УМЕНЬШАЮЩИХ МАСШТАБИРУЕМОСТЬ ПРИЛОЖЕНИЯ\*

*Описан подход к поиску причин, ограничивающих масштабируемость параллельных программ, основанный на использовании данных системного мониторинга. С помощью изложенного подхода можно получать данные как о наличии причин, связанных с аппаратными характеристиками вычислительной системы, так и о степени их влияния на эффективность выполнения программы. Изложены результаты проведенных исследований по применению данных системного мониторинга работы параллельных приложений для определения таких факторов, влияющих на масштабируемость приложений. Рассмотрен набор тестовых и пользовательских приложений и отражение в данных системного мониторинга факторов, влияющих на масштабируемость. Для выявления влияния этих факторов применялся визуальный анализ данных системного мониторинга и методы кластеризации этих данных. Для получения данных использовался инструмент построения отчетов о работе программы Job Digest, после чего данные анализировались с помощью анализатора WEKA. Результаты проведенных исследований позволяют сделать вывод о применимости предложенного подхода при исследовании программ для анализа влияния большого числа аппаратных факторов, ограничивающих масштабируемость параллельных программ.*

*Масштабируемость; эффективность; системный мониторинг; анализ эффективности; кластеризация данных; параллельное программирование.*

A.S. Antonov, A.M. Teplov

### USE OF SYSTEM MONITORING DATA TO DETERMINE FACTORS REDUCING APPLICATIONS SCALABILITY

*The paper describes an approach to finding the cause of limiting the scalability of parallel programs based on the use of data monitoring system. Using the approach described here can be prepared as data of a reason associated with the hardware of the computing system, and their impact on the program execution efficiency. The results of studies on the use of data monitoring system works parallel applications for the determination of the factors affecting the scalability of applications. Consider a set of test and custom applications, and reflected in the data system monitoring factors that affect scalability. To determine the effect of these factors is to use visual data analysis system monitoring and clustering methods of the data. To obtain the data used tool to generate reports on the program's Job Digest, after which the data were analyzed using the analyzer WEKA. The results of these studies suggest the applicability of the proposed approach for research programs to analyze the impact of a large number of hardware factors limiting the scalability of parallel programs.*

*Scalability; efficiency; system monitoring; efficiency analysis; data clustering; parallel programming.*

---

\* Работа выполнена при поддержке гранта РФФИ 13-07-00790 а.

**Введение.** На эффективность и масштабируемость параллельного приложения влияют различные факторы. К ним можно отнести как факторы, связанные с алгоритмом работы приложения и особенностями его реализации, так и факторы, связанные со свойствами программно-аппаратной среды, в которой приложение выполняется.

*Масштабируемость* параллельного приложения называется свойство параллельной программы, характеризующее зависимость изменения динамических характеристик его работы от параметров запуска этого приложения. *Эффективность* работы параллельного приложения связана и с другими динамическими характеристиками его работы и определяется отношением реальных значений показателей этих характеристик к максимальным теоретическим показателям. В силу взаимосвязи между понятиями эффективности и масштабируемости параллельного приложения, влияние на масштабируемость будут оказывать, в том числе, те факторы, которые оказывают влияние и на эффективность выполнения. При анализе масштабируемости важно знать, каким образом меняется их влияние при изменении параметров запуска.

В данной статье кратко рассмотрены примеры того, как факторы, влияющие на масштабируемость приложения, отражены в результатах системного мониторинга, собранных при работе приложения. Результаты мониторинга считаются с аппаратных датчиков и не требуют модификации кода программы и, следовательно, не влияют на работу самого приложения.

**1. Инструментарий, примененный при исследовании.** Результаты работы получены с помощью инструментария WEKA [1] и системы мониторинга работы приложений Job Digest [2].

Система Job Digest разработана для анализа приложений, выполняемых на суперкомпьютерах СКИФ МГУ «Чебышев» и «Ломоносов» [3,4], установленных в Научно-исследовательском вычислительном центре МГУ. Она использует результаты мониторинга для визуализации состояния вычислительной системы и измерения показаний различных аппаратных датчиков во время выполнения приложения. В процессе исследования система мониторинга Job Digest [3] получает информацию с множества аппаратных счетчиков. Эти данные счетчиков со всех узлов суперкомпьютера поступают в общее хранилище. Данные из хранилища ассоциируются с ходом работы приложений, выполняемых на суперкомпьютере. Система очень гибкая и имеет большое количество настроек.

Инструментарий WEKA является свободно распространяемым программным пакетом с открытым исходным кодом и применяется для анализа последовательностей векторов, содержащих в себе какую-либо информацию. WEKA содержит в себе большое число алгоритмов интеллектуального анализа данных и обеспечивает возможность визуализации при анализе информации. С помощью этого инструментария данные системы Job Digest преобразуются в вектор значений различных счетчиков так, чтобы в одном векторе значения датчиков соответствовали одному и тому же временному интервалу. Значения от каждого датчика в формируемом векторе занимают позиции так, чтобы в каждом полученном векторе в одной и той же позиции находились данные от одного и того же датчика.

**2. Факторы, влияющие на масштабируемость параллельных программ.** По результатам проведенных исследований были выделены несколько групп факторов, в большей или меньшей степени влияющих на масштабируемость программ.

К первой группе отнесены факторы, связанные с работой коммуникационной сети, как одного из наиболее важных компонентов современного суперкомпьютера. Посредством коммуникационной сети происходит взаимодействие процессов,

и потому факторы, влияющие на эффективность этого взаимодействия, будут влиять и на масштабируемость параллельной программы. При увеличении числа взаимодействующих процессов накладные расходы на организацию сетевого взаимодействия будут увеличиваться и тем самым оказывать влияние на масштабируемость. Это такие факторы, как латентность сети, пропускная способность, топология коммуникационной сети и т.п.

Ко второй группе отнесены факторы, связанные с компонентами вычислительного узла суперкомпьютера. Это факторы, зависящие от объема и характеристик всей иерархии памяти узла. При увеличении числа процессов на узле каждому процессу достается меньший объем быстрой памяти. Выход же за границы быстрой памяти может привести к резким изменениям эффективности работы.

Наконец, к третьей группе отнесены факторы, связанные с характеристиками примененного варианта алгоритма и исследуемой параллельной программы. К таким факторам можно отнести предел декомпозиции данных для вычислителя или же дисбаланс нагрузки на вычислители и другие особенности, связанные с алгоритмическим устройством программы.

**3. Методика проведения экспериментов.** Для выявления факторов, влияющих на масштабируемость приложений [5] при исследовании результатов системного мониторинга [6–8], анализировались как реальные приложения пользователей суперкомпьютерного комплекса, работающие в общей очереди задач, так и специально написанные для этого тесты.

Для моделирования факторов, связанных с работой коммуникационной сети, использована система специально сконструированных тестов. Данные тесты не проводят никаких вычислений, а занимаются только отправкой и приемом сообщений. Это необходимо, чтобы максимально увеличить нагрузку на коммуникационную сеть и минимизировать влияние вычислений на общую картину, получаемую на основе данных мониторинга.

Система тестов коммуникационной сети состояла из трех типов приложений, которые по разной схеме пересылают сообщения между участвующими процессами.

Тест «master» работает следующим образом: один главный процесс рассылает сообщения всем остальным процессам приложения. Параметрами запуска регулируются длина сообщения и число циклов рассылки. В данном тесте основная нагрузка приходится на рассылающий процесс, и поэтому для его высокой производительности важны свойства коммуникационной сети. При высокой загрузке сообщениями малой длины ограничивающим масштабируемость фактором станет латентность сети.

Тест «snake» работает следующим образом: на первом шаге все процессы с чётным номером пересылают сообщение своим соседям с номером, на единицу большим. На втором шаге процессы с нечётным номером пересылают сообщение соседям с номером, также большим на единицу. Последний процесс пересылает сообщения нулевому. В этом тесте большая часть процессов обменивается с соседними по номерам процессами. При запуске по несколько процессов на узле значительная часть пересылок будет приходиться на пересылки внутри узла, поэтому коммуникационная сеть будет задействована в меньшей степени. Физическое размещение процессов по вычислительным узлам может зависеть от настроек системы очередей конкретного суперкомпьютера, но, как правило, при плотном размещении процессов по узлам (на каждое ядро узла по одному процессу) на одном узле размещаются соседние процессы.

Тест «butterfly» работает следующим образом: каждый процесс с номером меньше половины общего числа процессов на первом шаге отправляет сообщение процессу с симметричным номером с конца (нулевой процесс посылает последне-

му, второй предпоследнему и т.д.). Таким образом, при обычном размещении процессов по вычислительным узлам, бóльшая часть сообщений направлена не на соседние процессы, а на процессы, находящиеся физически на разных вычислительных узлах. Поэтому большая часть сообщений проходит не внутри узла, а по коммуникационной сети.

Этот набор тестов позволяет оценить отражение в результатах мониторинга основных свойств коммуникационной сети, влияющих на масштабируемость приложения.

Для выявления факторов, связанных с эффективным использованием компонентов вычислительного узла суперкомпьютера, применялись приложения, по-разному использующие иерархию памяти, в том числе такие, как реализация блочного перемножения матриц и тест Linpack.

Для получения результатов системы мониторинга применялись отчеты о работе приложения Job Digest. Данные отчета анализировались на предмет нахождения корреляций между различными графиками системных датчиков. Анализировались как численные показатели датчиков, так и динамика изменения их значений.

Для анализа зависимостей между различными показателями данные Job Digest преобразовывались в представление данных для анализатора WEKA. Для этого был разработан скрипт, преобразующий несколько временных рядов по различным датчикам в один временной ряд, объединяющий в вектор все показания от датчиков мониторинга.

В WEKA данные атрибутов анализировались на предмет характерных зависимостей между показаниями различных датчиков в некоторый момент времени. Для этого рассматривались диаграммы зависимостей между показаниями различных датчиков.

Также для собранных данных состояния всех датчиков применялся алгоритм кластеризации. С его помощью определялось, какую часть времени работы программы она провела в некотором состоянии.

**4. Результаты анализа проведенных экспериментов. Пропускная способность коммуникационной сети и топология.** В системах с распределённой памятью пропускная способность коммуникационной сети может стать ограничивающим фактором масштабируемости приложения в случае, если процессы обмениваются между собой очень большими объёмами данных. Тогда пропускная способность будет определять, какое время будет расходоваться на пересылку каждого сообщения. Таким образом, чем больше процессов будут пересылать такие сообщения, тем выше будет нагрузка на коммутаторы, соединяющие вычислительные узлы.

Рассмотрим запуск теста «butterfly» с параметрами запуска, соответствующими коротким сообщениям различной длины (1, 1 000 000, 100 000 000).

На графиках скорости передачи данных по сети Infiniband (рис. 1,а) видна низкая интенсивность использования сети со скоростью около 100 МБ/с при длине сообщения 1. При увеличении длины сообщения до 1 000 000 (рис. 1,б) виден рост интенсивности использования сети до 900 МБ/с. Разница между максимальным и минимальным значениями становится больше. Средние значения передачи и получения данных попеременно находятся возле максимального и минимального значений. Такое поведение может указывать на достижение предела пропускной способности отдельных узлов, что вызывает задержку передачи и приёма большого числа сообщений большой длины.

В третьем эксперименте с тестом «butterfly» длина сообщения увеличена до 100 000 000 (рис. 1,в). Максимальное значение скорости передачи и приёма данных находится на уровне 1100 МБ/с, что превышает значение в предыдущем эксперименте (900 МБ/с). Однако минимальное значение скорости передачи и приёма

данных при очень длинных сообщениях находится на уровне 650 МБ/с. Это значительно меньше значения в предыдущем эксперименте (850 МБ/с), которое не намного меньше максимального значения. Схема пересылок осталась прежней, но разрыв между максимальным и минимальным значениями скорости передачи и приёма данных по сети сильно увеличился. Минимальное значение скорости передачи данных стало меньше, чем при эксперименте с меньшим размером сообщения. Сильно возросли отклонения значений средней величины скорости передачи и приёма данных, которые имеют ступенчатую структуру.

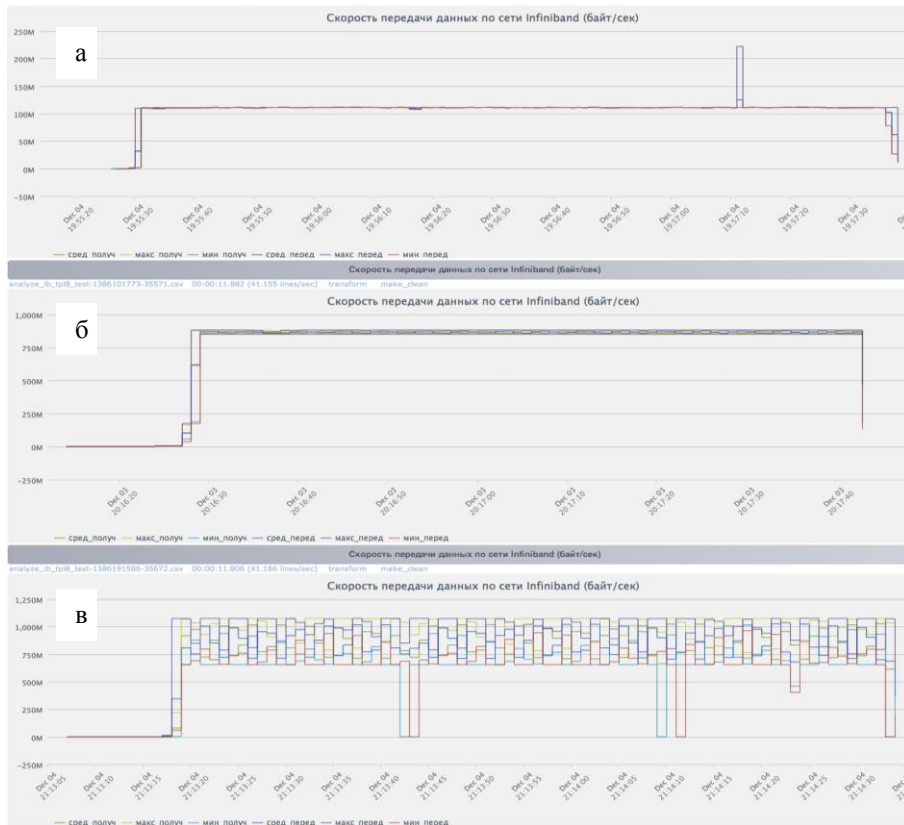


Рис. 1. Скорость передачи данных по сети Infiniband (а – длина 1, б – длина 1 000 000, в – длина 1 000 000), тест «butterfly»

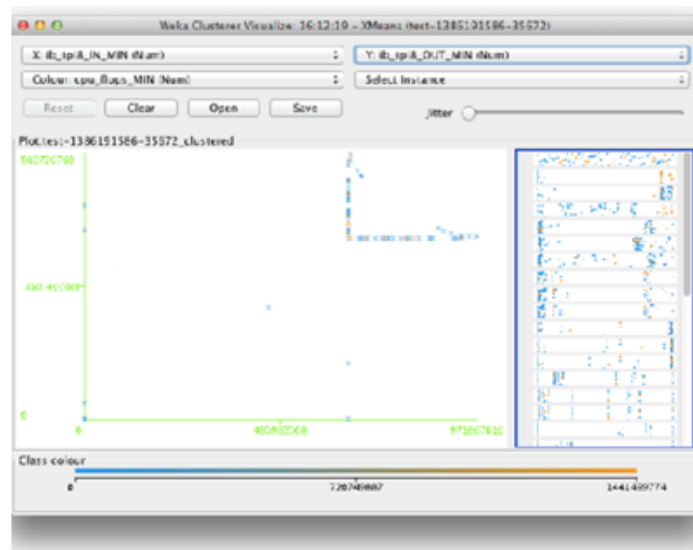


Рис. 2. Скорость передачи данных по сети Infiniband (длина 1 000 000), тест «snake»

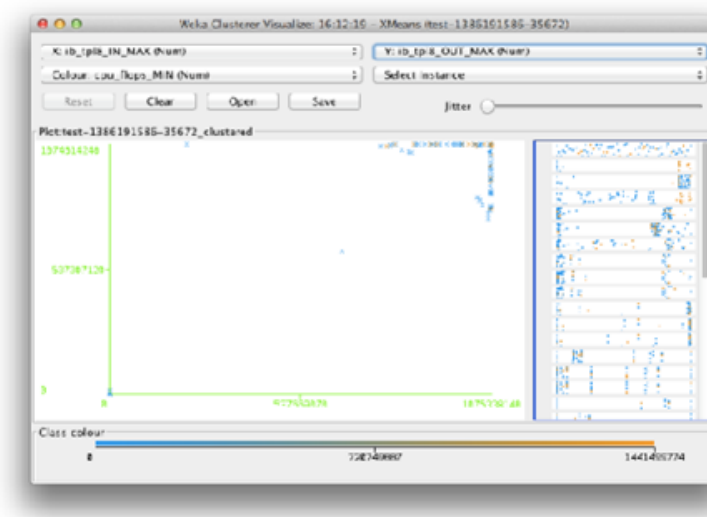
Структура графика скорости передачи данных по сети Infiniband указывает на то, что предел пропускной способности сети передачи данных определяет время выполнения операции обмена сообщениями.

Если преобразовать данные Job Digest в представление анализатора WEKA и рассмотреть зависимости атрибутов минимальной, максимальной и средней скорости передачи сообщений от соответствующих значений скорости приема сообщений для сообщений длиной 1 000 000 и сообщений длиной 100 000 000, то станет понятно, что предел скорости передачи действительно достигнут.

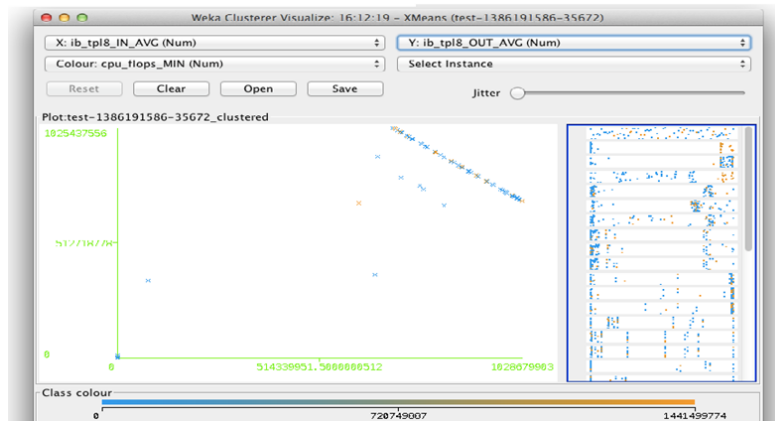
По графикам (рис. 3,а) можно сделать вывод, что максимальная и минимальная скорости передачи и приема данных при использовании очень длинных сообщений ограничены соответственно сверху и снизу константным значением, а изменения максимального и минимального значений, как приема, так и передачи сообщений, не влияют друг на друга.



а



б



6

Рис. 3. Зависимость атрибутов минимальной (а), максимальной (б) и средней (в) скоростей входящей передачи данных от скорости получения данных по сети Infiniband для теста «butterfly» с сообщениями длиной 100 000 000

Среднее значение демонстрирует обратную зависимость между скоростью приема и скоростью передачи данных, которая в сумме остается константной (рис. 3,б). Это говорит о том, что при использовании очень длинных сообщений предел скорости передачи данных достигнут. А это соответственно влияет на масштабируемость такого приложения.

Этот вывод подтверждает также то, что при рассмотрении той же зависимости для атрибутов того же теста с использованием сообщений длиной 1 000 000 характер зависимости сильно отличается. Средняя скорость передачи данных почти не отличается от максимальной и минимальной и меняется в малых пределах, но пропорционально – с увеличением скорости приема увеличивается и скорость передачи данных (рис. 3,в).

Если же рассмотреть различие между запуском тестов «butterfly» и «snake» (рис. 1,б и 2), то проявляется влияние топологии расположения процессоров. В этих двух экспериментах общее число одновременно пересылаемых сообщений и их длина одинаковы, но интенсивность использования сети отличается почти в 4 раза (около 950 МБ/с для теста «butterfly» и около 230 МБ/с для теста «snake» при одинаковой длине сообщения 1 000 000).

Большинство сообщений теста «snake» было доставлено через внутренние каналы связи, так как соседние процессы расположены физически рядом. Поэтому время на передачу данных оказалось значительно меньшим. Действительно, при такой схеме передачи информации на каждом узле вычислительного комплекса через внешние каналы связи по коммуникационной сети Infiniband будут обмениваться только два процесса на узле, а именно, те процессы, которые физически расположены на разных узлах с отправителем и получателем. В случае же теста «butterfly» (см. рис. 1) все процессы на узле будут использовать Infiniband, так как все их получатели и отправители будут являться удалёнными адресатами. Важно, что в первом и втором случаях объём переданных данных оставался одинаковым, потому что в обоих тестах было передано одинаковое число сообщений одинаковой длины, отличалась только схема их пересылок. Это и оказало сильное влияние на время выполнения тестов.

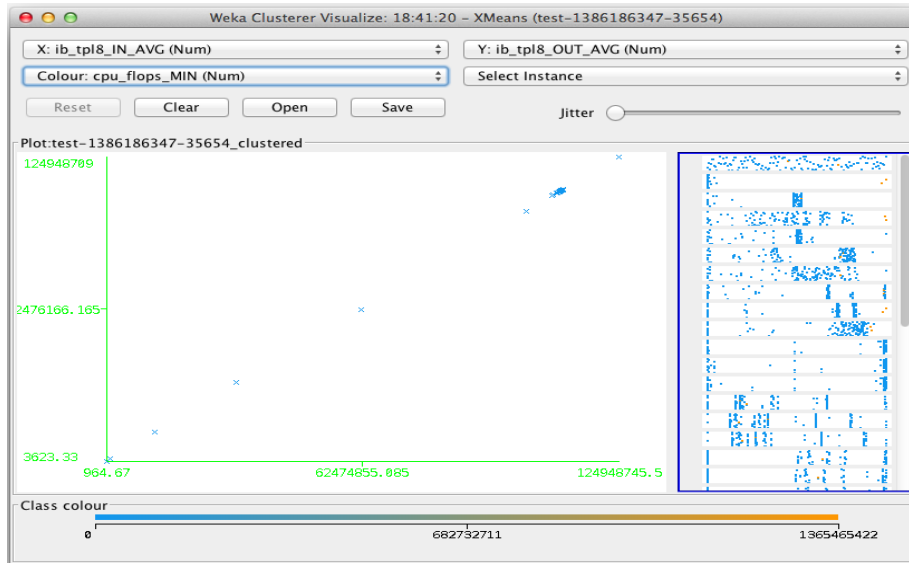


Рис. 4. Зависимость атрибутов средней скорости входящей передачи данных от скорости получения данных по сети Infiniband для теста «butterfly» с сообщениями длиной 1 000 000

Таким образом, на графиках Job Digest можно увидеть ограничения, накладываемые на масштабируемость приложения, связанные с характеристиками и топологией коммуникационной сети и схемы пересылок данных в приложении.

**Использование при работе жёсткого диска.** Поскольку время доступа к жёсткому диску сравнительно велико, его интенсивное использование может стать фактором, сдерживающим масштабируемость приложения. Наиболее сильно это будет проявляться, если процессы используют частые обмены с дисками очень небольшими порциями данных. При этом время на передачу данных может значительно превышать время их записи на диск.

Рассмотрим пример графика системы Job Digest работы теста Linpack. В тесте был определён такой размер данных, что они не должны помещаться в оперативную память вычислительных узлов. Тест был запущен в разделе, содержащем локальные диски, поэтому задача должна активно использовать дисковую память и своп.

На графике количества операций с плавающей точкой (рис. 5,а) видна общая низкая производительность приложения. Только на нескольких участках производительность повышается. Это подтверждает предположение о том, что общая эффективность работы мала.

На графике скорости обмена данными с диском (рис. 5,б) видна активность на протяжении всей работы теста. Такая дисковая активность может свидетельствовать либо о том, что используется запись в файл, либо о том, что данные подкачиваются из свопа.

График интенсивности работы со своп-памятью (рис. 5,в) подтверждает предположение о том, что на протяжении всей работы приложения используется своп-память.



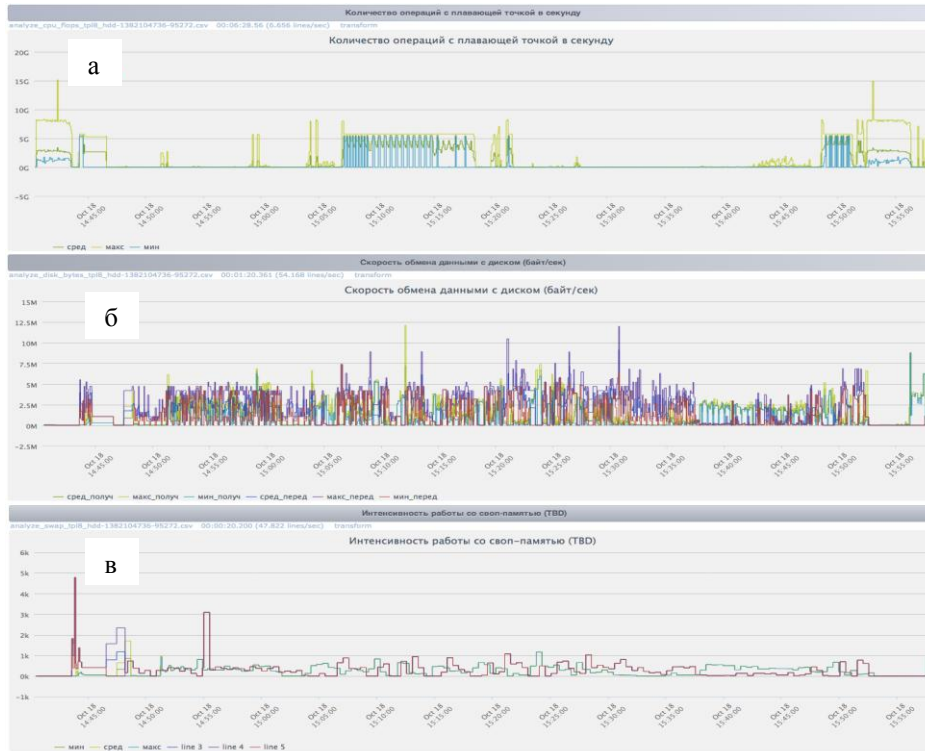


Рис. 5. а – количество операций с плавающей точкой в секунду; б – скорость обмена данными с диском; в – интенсивность работы со своп-памятью. Тест Linpack

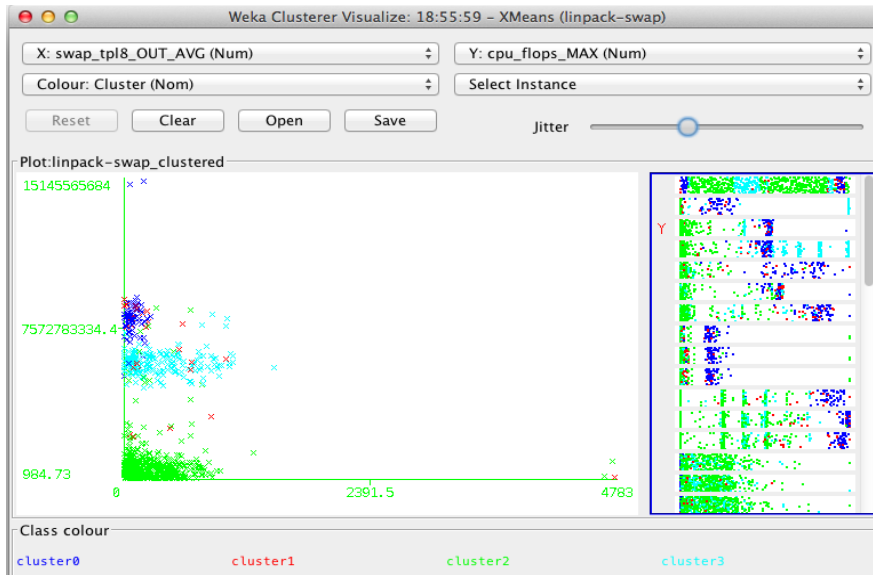


Рис. 6. Зависимость активности своп-памяти в приложении (ось X) от производительности (ось Y). Цветами обозначены выделенные классы состояний работы приложения

Если преобразовать данные Job Digest в представление WEKA и проанализировать их, то увидим, что активность своп-памяти связана с производительностью работы приложения.

На графике зависимости использования своп-памяти от производительности (рис. 6) показана визуализация результата алгоритма кластеризации состояний системы во время работы. На нем видно, что большую часть времени (67 % – cluster2) приложение имело низкую производительность и ненулевую активность своп-памяти. Значительно меньшую часть времени (21 % – cluster3) приложение имело среднюю производительность при активном обмене с диском. И самую высокую производительность имело только в 9 % (cluster0) состояний при отсутствии обменов с диском.

Таким образом, на графиках Job Digest можно увидеть ограничения, накладываемые на масштабируемость приложения пределом объёма оперативной памяти.

**Предел декомпозиции данных.** Рассмотрим графики количества операций с плавающей точкой двух запусков одного и того же приложения на одном и том же числе процессоров, но с разным размером задачи.

Было осуществлено два запуска перемножения матриц размером 1024x1024 и 512x512.

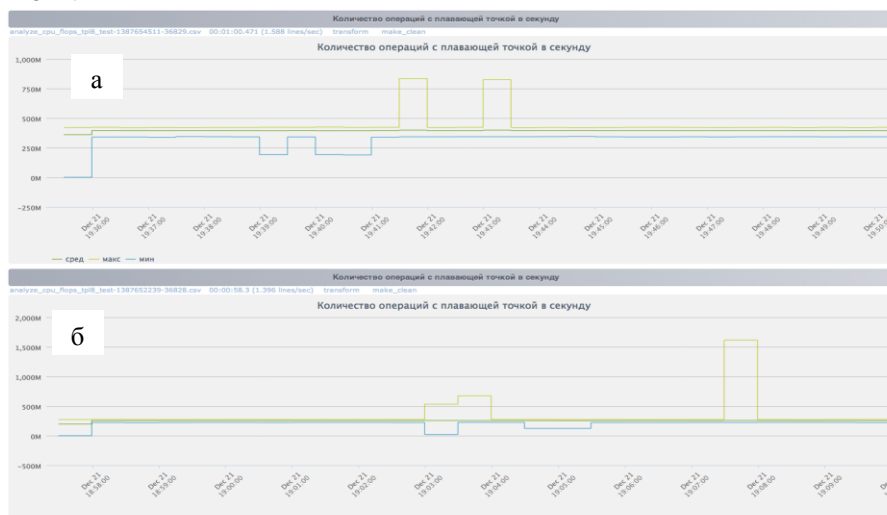


Рис. 7. Среднее количество операций с плавающей точкой в секунду, перемножение матриц (а – матрицы размером 1024x1024; б – матрицы размером 512x512)

При запуске перемножения матриц размером 1024x1024 среднее количество операций с плавающей точкой в секунду (рис. 7,а) принимает значения выше 420 Мфлопс.

При запуске того же приложения, но с размером матрицы 512x512 (рис. 7,б) средняя и максимальная производительности не превышают значения 270 Мфлопс.

Если размер задачи не позволяет обеспечить большой объём вычислений на каждом процессе, то наступит момент, когда на организацию вычислений будет уходить время, сравнимое со временем, затраченным на сами вычисления. Таким образом, затраты на организацию работы параллельного приложения станут сравнимы с затратами на выполняемую полезную работу. То есть этот фактор также ограничивает масштабируемость приложения.

При уменьшении размера задачи ожидаемо было бы увеличение производительности, если данные хорошо укладываются в кэш-память и к ним быстрый доступ. Но если данных уже слишком мало, то производительность будет падать с уменьшением объема вычислений.

Это подтверждает предположение о том, что размер матрицы в данном случае слишком мал. Поэтому производительность значительно уменьшилась.

Таким образом, на графиках Job Digest можно увидеть достижение предела декомпозиции данных, ограничивающего масштабируемость.

**Заключение.** Рассмотренные результаты экспериментов подтверждают, что в результатах системного мониторинга могут проявляться различные факторы, влияющие на масштабируемость параллельных программ. К ним относятся факторы, относящиеся к влиянию программно-аппаратной среды, свойств алгоритма и другие. Представленные результаты экспериментов демонстрируют некоторые наиболее яркие проявления факторов, ограничивающих масштабируемость приложений. Информация от Job Digest, дополненная зависимостью значений одних атрибутов от других, полученной в WEKA, позволяет сделать выводы о степени влияния различных факторов на масштабируемость как тестов, так и реальных приложений.

#### БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Weka 3: Data Mining Software in Java. <http://www.cs.waikato.ac.nz/ml/weka/>.
2. Адинец А.В., Брызгалов П.А., Воеводин Вад. Вл., Жуматий С.А., Никитенко Д.А., Стефанов К.С. Job Digest-подход к исследованию динамических свойств задач на суперкомпьютерных системах // Вестник Уфимского гос. авиационного техн. ун-та. – 2013. – Т. 17, № 2(55). – С. 131-137.
3. Воеводин Вл.В., Жуматий С.А., Соболев С.И., Антонов А.С., Брызгалов П.И., Никитенко Д.А., Стефанов К.С., Воеводин В.В. Практика суперкомпьютера «Ломоносов» // Открытые системы. – 2012. – № 7. – С. 36-39.
4. Антонов А.С. СКИФ МГУ – основа Суперкомпьютерного комплекса Московского университета // Вторая Международная научная конференция «Суперкомпьютерные системы и их применение» (SSA'2008): доклады конференции (27–29.10.2008 г., Минск). – Минск: ОИПИ НАН Беларуси, 2008. – С. 7-10.
5. Антонов А.С., Теплов А.М. Исследование масштабируемости программ с использованием инструментов анализа параллельных приложений на примере модели атмосферы NH3D // Вестник Южно-Уральского государственного ун-та. Сер. Вычислительная математика и информатика. – 2013. – Т. 2, № 1. – С. 5-16.
6. Антонов А.С., Жуматий С.А., Никитенко Д.А., Стефанов К.С., Теплов А.М., Швец П.А. Исследование динамических характеристик потока задач суперкомпьютерной системы // Вычислительные методы и программирование: Новые вычислительные технологии (Электронный научный журн.). – 2013. – Т. 14. – С. 104-108.
7. Никитенко Д.А., Стефанов К.С. Исследование эффективности параллельных программ по данным мониторинга // Вычислительные методы и программирование: Новые вычислительные технологии (Электронный научный журн.). – 2012. – Т. 13. – С. 97-102.
8. Никитенко Д.А. Комплексный анализ производительности суперкомпьютерных систем, основанный на данных системного мониторинга // Вычислительные методы и программирование: Новые вычислительные технологии (Электронный научный журн.). – 2014. – Т. 15. – С. 85-97.

#### REFERENCES

1. Weka 3: Data Mining Software in Java. Available at: <http://www.cs.waikato.ac.nz/ml/weka/>.
2. Adinets A.V., Bryzgalov P.A., Voevodin Vad. Vl., Zhumatiy S.A., Nikitenko D.A., Stefanov K.S. Job Digest-approach to the study of the dynamic properties task on supercomputing systems], *Vestnik Ufimskogo gos. aviatsionnogo tekhn. un-ta* [Bulletin of the Ufa state aviation technical University], 2013, Vol. 17, No. 2(55), pp. 131-137.

3. Voevodin V.I., Zhumatiy S.A., Sobolev S.I., Antonov A.S., Bryzgalov P.I., Nikitenko D.A., Stefanov K.S., Voevodin V.V. Praktika superkomp'yutera «Lomonosov» [Practice supercomputer "Lomonosov"], Otkrytye sistemy [Open systems], 2012, No. 7, pp. 36-39.
4. Antonov A.S. SKIF MGU – osnova Superkomp'yuternogo kompleksa Moskovskogo universiteta [SKIF MSU - based Supercomputer of the University of Moscow], Vtoraya Mezhdunarodnaya nauchnaya konferentsiya «Superkomp'yuternye sistemy i ikh primeneniye» (SSA'2008): doklady konferentsii (27-29.10.2008 g., Minsk) [Second international scientific conference "Supercomputer systems and applications" (SSA'2008): report of the conference (27-29.10.2008, Minsk)]. Minsk: OIPI NAN Belarusi, 2008, pp. 7-10.
5. Antonov A.S., Teplov A.M. Issledovanie masshtabiruемости programm s ispol'zovaniem instrumentov analiza parallel'nykh prilozheniy na primere modeli atmosfery NH3D [Study the scalability of programs using tools of analysis of parallel applications on the example of a model atmosphere NH3D], Vestnik Yuzhno-Ural'skogo gosudarstvennogo un-ta. Ser. Vychislitel'naya matematika i informatika [Bulletin of the South Ural state universitetet. A series of Computational mathematics and Informatics], 2013, Vol. 2, No. 1, pp. 5-16.
6. Antonov A.S., Zhumatiy S.A., Nikitenko D.A., Stefanov K.S., Teplov A.M., Shvets P.A. Issledovanie dinamicheskikh kharakteristik potoka zadach superkomp'yuternoy sistemy [Investigation of the dynamic characteristics of the task flow supercomputer system], Vychislitel'nye metody i programmirovaniye: Novye vychislitel'nye tekhnologii (Elektronnyy nauchnyy zhurn.) [Computational methods and programming: New computational technologies (Electronic scientific journal)], 2013, Vol. 14, pp. 104-108.
7. Nikitenko D.A., Stefanov K.S. Issledovanie effektivnosti parallel'nykh programm po dannym monitoringa [Study of the efficiency of parallel programs according to the monitoring], Vychislitel'nye metody i programmirovaniye: Novye vychislitel'nye tekhnologii (Elektronnyy nauchnyy zhurn.) [Computational methods and programming: New computational technologies (Electronic scientific journal)], 2012, Vol. 13, pp. 97-102.
8. Nikitenko D.A. Kompleksnyy analiz proizvoditel'nosti superkomp'yuternykh sistem, osnovannyy na dannykh sistemnogo monitoring [A comprehensive analysis of performance supercomputer systems, based on the data monitoring system], Vychislitel'nye metody i programmirovaniye: Novye vychislitel'nye tekhnologii (Elektronnyy nauchnyy zhurn.) [Computational methods and programming: New computational technologies (Electronic scientific journal)], 2014, Vol. 15, pp. 85-97.

Статью рекомендовал к опубликованию д.ф.-м.н., профессор Вл.В. Воеводин.

**Антонов Александр Сергеевич** – Научно-исследовательский Вычислительный центр МГУ им. М.В. Ломоносова; e-mail: asa@parallel.ru; 119234, Москва, Ленинские горы, 1, стр. 4.; тел. +74959392347; к.ф.-м.н.; в.н.с.

**Теплов Алексей Михайлович** – e-mail: alex-teplov@yandex.ru; тел.: +79267658534; м.н.с.

**Antonov Aleksander Sergeevich** – Research Computing Center Lomonosov Moscow State University; e-mail: asa@parallel.ru; 1, Leninskie gory street, building 4, Moscow, 119234, Russia; phone: +74959392347; cand. of phis.-math. sc.; leading researcher.

**Teplov Aleksey Mikhailovich** – e-mail: alex-teplov@yandex.ru.ru; phone: +79267658534; junior researcher.