Анализ потока задач на параллельной вычислительной системе

А.С. Князев

В статье описывается архитектура программного комплекса для мониторинга и исследования программно-аппаратной среды вычислительной системы. Предлагается набор характеристик для мониторинга и рассматривается его информативность для администратора и пользователя вычислительной системы. Приводится сравнение результатов апробации системы на кластерах НИВЦ МГУ за 2007 и 2008 года.

1. Введение

Потенциал современных вычислительных систем очень велик, но достижение максимальной эффективности их работы невозможно без наблюдения администратором системы за потоком задач. И чем сложнее система, чем больше процессоров и ядер она использует, тем сложнее понимать её поведение, определять узкие места и причины снижения производительности [1].

Для наблюдения за поведением системы администратор должен, во-первых, иметь возможность собирать количественные данные о работе системы, и, во-вторых, уметь их анализировать. В современных вычислительных системах большинство приложений, узлов, сетевых и даже инфраструктурных устройств (кондиционеры, ИБП) предоставляют большое количество доступных для наблюдения характеристик. К сожалению, у существующих распространённых средств мониторинга есть недостатки, которые существенно затрудняют сбор и анализ данных.

Все они ориентированы на решение первой задачи – сбор информации. Это относится и к системам мониторинга систем общего типа, таким как Zabbix [6] и Nagios [7], и к системам наблюдения за вычислительными кластерами: Ganglia [3], SuperMon [4], PARMON [5]. Они предоставляют администратору возможность оценить состояние вычислительной системы на текущий момент, но не хранят подробную историю состояний в прошлом.

Анализ поведения системы необходим как пользователю, так и администратору. Пользователь, анализируя выполнение своих программ, определяет эффективность их работы, проблемы в выполнении, адекватность реализации алгоритма применительно к данной вычислительной системе.

Администратору необходимо, во-первых, уметь определять «нормальное», «штатное» поведение системы, и, во-вторых, в случае отклонения, находить его причину. Существующие средства предлагают задавать «штатное» поведение установлением границ допустимых значений параметров вычислительной системы. Этого не достаточно для описания всех нештатных ситуаций, и определение этих границ может быть очень непростым для каждой системы.

Анализ накопленных данных и потока задач помогает администратору в определении того, насколько полно и качественно используется его вычислительная система. Для больших систем хранение всех собираемых данных с максимальной точностью нецелесообразно, но и простого набора агрегатных характеристик, таких как максимум/минимум/среднее, недостаточно для анализа нештатных ситуаций в прошлом. Один из подходов к хранению данных истории рассматривается в статье [2].

В суперкомпьютерном центре МГУ давно ведутся исследования эффективности параллельных программ и вычислительных систем, см. например [8], [9]. Настоящая работа описывает развитие существующих систем мониторинга.

2. Система SVA

Система мониторинга ParCon [10], которая используется в настоящее время, обеспечивает режим наблюдения «на лету» за состоянием узлов вычислительного кластера. Автором разра-

ботана новая система SVA, которая развивает ParCon, решая задачи хранения исторических данных о поведении кластера, их визуализации и анализа.

SVA представляет поведение системы в виде потоков событий, источником которых является вычислительный узел. Каждое событие состоит из даты и времени, типа события (*характеристики*) и значения. Таким образом, при P узлах и N типах событий, в каждый момент времени система мониторинга сохраняет $P \times N$ событий, при типичных значениях $P > 100, N > 10$. Поведение одного узла представляет собой *траекторию* в *фазовом пространстве характеристик*, поведение задачи – траекториями узлов, на которых она считается. Формально, используя обозначения из [13], траектория описывается как набор функций $m_i(t)$ для каждого из P узлов, описывающих состояние узла в момент времени t:

$$(m_1(t), m_2(t), \ldots, m_N(t))_p \quad p = 1,2,\ldots,P$$

Пусть R_i - множество значений $m_i(t)$, тогда фазовое пространство характеристик обозначается $M = R_1 \times R_2 \times \ldots \times R_N$.

Для помощи администратору в определении «штатного» режима работы – то есть режима, в котором все выделенные задаче ресурсы доступны и используются максимально эффективно, предлагаются два подхода. SVA, накопив историю поведения кластера и исходя из того, что большую часть времени кластер работает в штатном режиме, устанавливает доверительные интервалы для траекторий задач в пространстве M. При выходе траектории из интервалов, SVA помечает этот участок как возможно неэффективный. Администратор может уточнять оценки, просматривая исторические данные и указывая SVA на участки неэффективной работы. Возможно и выполнение вручную оптимизированных задач и задание их траекторий как примера заведомо эффективного выполнения.

Помимо этого, используются методы статистического анализа. В простейшем случае это определение корреляций и скрытых переменных: при анализе корреляции сравнивается её значение для штатных режимов и текущего, скрытые переменные позволяет снизить размерность пространства M, исключив некоторые типы событий из анализа. Динамический поиск проекций (dynamic statistical projection pursuit, [12]) является более сложным методом анализа: человек обладает хорошими способностями визуального анализа, но пространство M слишком многомерное для непосредственного отображения. Метод автоматически ищет проекции M в пространство 3 измерений для визуального отображения, выбирая отображаемые m_i таким образом, чтобы удовлетворить критерию максимальной интересности проекции. В качестве критерия обычно используется отклонение от нормального распределения.

Хранение данных позволяет работать как с визуальным представлением истории вычислительной системы, так и с числовыми данными. Объём истории событий на кластере СКИФ МГУ «Чебышев» за один день может достигать 50 гигабайт, для уменьшения этого объёма используется следующее: для штатных режимов работы сохраняются агрегатные характеристики потока, выбранные администратором (например, среднее, экстремумы, дисперсия), а для нештатных режимов данные сохраняются с высокой точностью. Производится фильтрация данных по относительному изменению значений: если $\left|\frac{m_i(t) - m_i(t+1)}{m_i(t)}\right| < \Delta$, то для обоих моментов времени сохраняется значение $m_i(t)$. Для ускорения анализа и уменьшения размера рабочего набора, данные старше некоторого порога могут перемещаться в архив.

Значения характеристик могут быть числовыми, логическими и строковыми. Их сбор не подразумевает внесения изменений в пользовательские программы, но это может быть полезно для более детального изучения программы. Для этого пользователь может сам модифицировать программу, или же может применяться инструментация исходного кода компилятором, как, например, в работе [11].

Набор наблюдаемых характеристик определяется администратором системы. Чем больше характеристик мы наблюдаем, тем больше наше знание о работе системы, но и тем больше хранимые данные и тем больше накладные расходы мониторинга. На кластере СКИФ МГУ «Че-

бышев» $P >> N$ и добавление каждой новой характеристики особенно сильно увеличивает размер пространства M. Основные наблюдаемые характеристики приведены в таблице 1.

Таблица 1. Основные наблюдаемые характеристики.

Характеристика	**Единица измерения**
Использование ЦП (cpu_user, cpu_system, cpu_idle)	%
Работа с файлом подкачки	страниц в секунду
Чтение/запись на локальный диск	мегабайт в секунду
Чтение/запись на удалённое сетевое хранилище	мегабайт в секунду
Использование системной сети	пакетов в секунду
Использование вспомогательной сети	пакетов в секунду
Показания датчиков температуры	градус Цельсия
Частота вращения вентиляторов	оборотов в секунду

Эти характеристики наблюдаются непосредственно на вычислительном узле кластера. Для описания поведения задачи, выполняющейся на нескольких узлах, показательными являются отклонения характеристик по узлам, корреляция характеристик на различных узлах, средние и пиковые значения. При анализе задач одного пользователя администратор наблюдает за типичными проблемами, возникающими у его задач, за их эволюцией за время работы пользователя на кластере.

3. Реализация системы

Система SVA реализована с использованием языков Java и Perl и рассчитана на эксплуатацию в ОС семейства GNU/Linux. Архитектура системы изображена на рисунке 1. Для сбора характеристик вычислительного узла используется модуль Antmon, входящий в состав программного комплекса ParCon [10].

Модуль `ant_agent` отвечает за сбор характеристик на вычислительном узле. Для минимизации накладных расходов на сбор, он написан на языке C. Собранные характеристики он отправляет модулю сбора `ant_mon`, для передачи данных используется собственный двоичный протокол поверх UDP. Модули `ant_mon` могут быть иерархически организованы в виде дерева для уменьшения нагрузки на узел и оптимизации использования сети. `Ant_mon` отправляет все данные модулю SVA, который сохраняет данные в СУБД PostgreSQL, предоставляет веб-интерфейс к историческим данным и производит анализ.

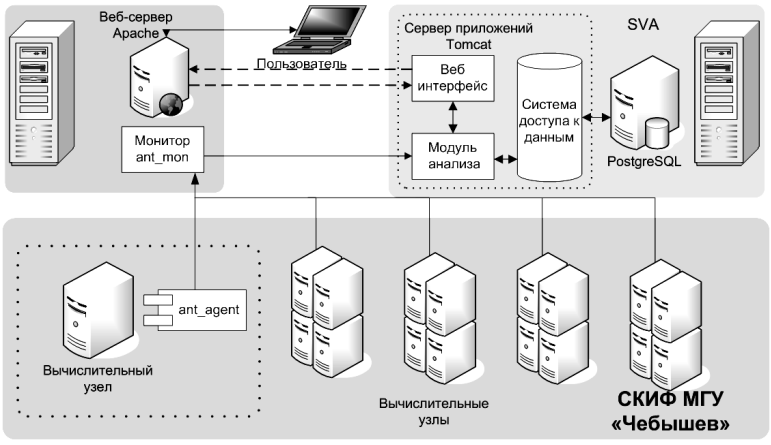

Рисунок 1. Архитектура SVA

4. Анализ и визуализация потока задач

Система SVA собирала данные о задачах на кластерах НИВЦ МГУ с 2007 года. Рассматривается кластер Ant и новый кластер СКИФ МГУ «Чебышев».

Таблица 2. Вычислительные кластеры.

	Ant	**СКИФ МГУ «Чебышев»**
Количество узлов	80	625
Количество процессоров	160	1250
Количество ядер	160	5000
Процессор	AMD Opteron 248	Intel Xeon E5472
Тактовая частота, ГГц	2.2	3.0
R_{max}	0.704 TFlop/s	60 TFlop/s
R_{peak} (Linpack)	0.512 TFlop/s	47 TFlop/s
Продолжительность наблюдения, дней	520	96
Количество задач	41500	96
Задачи, использовавшие в сумме 99% процессоро-часов	20000	5500

Рост производительности составил почти два порядка, тем интереснее сравнить использование кластеров пользователями (таблица 2). На новом кластере задачи, использующие в сумме 99% процессоро-часов, составляют треть от всех задач, а на кластере Ant почти половину. Это значит, что не все задачи успешно масштабируется на новый кластер.

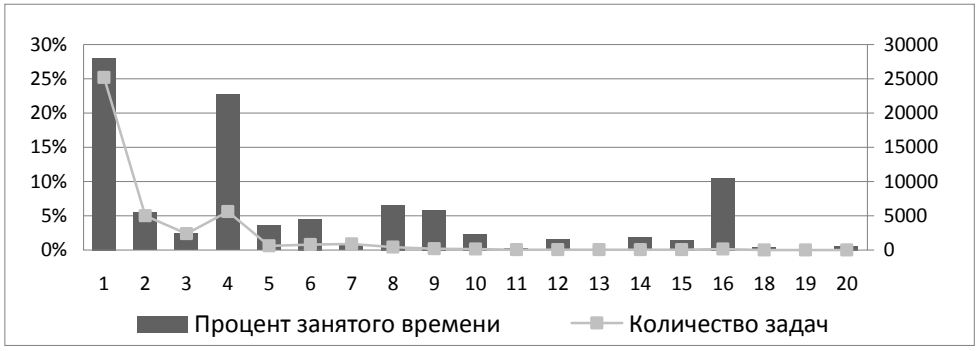

Рисунок 2. Распределение задач по количеству процессоров, кластер Ant

Рисунок 3. Распределение задач по количеству ядер, кластер СКИФ МГУ «Чебышев»

Изменилось и распределение задач по количеству используемых процессоров. На рисунке 3 для наглядности отображены только те значения количества ядер, которые использовали более 1% общего времени счёта на кластере.

На кластере СКИФ МГУ «Чебышев» SVA на текущий момент не собирает информацию по всем характеристиками узла, поэтому нет возможности сравнить её с архивом истории по кластеру Ant. Запуск SVA в полном объёме планируется на март 2009г.

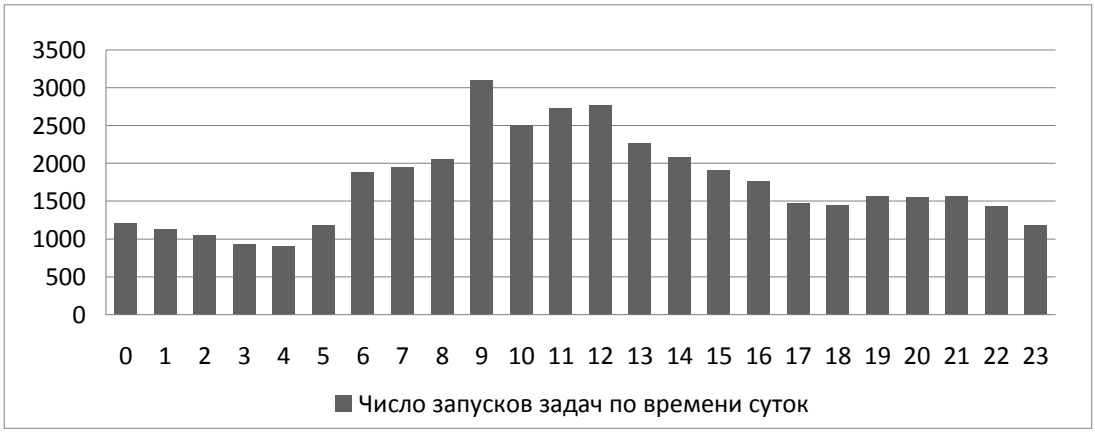

Рисунок 4. Число запусков задач по времени суток

На рисунке 4 изображена одна из характеристик потока задач на вычислительной системе – распределение запусков задач по времени суток, для кластера Ant.

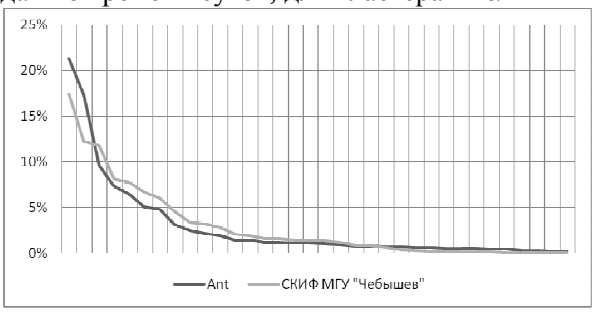

Рисунок 5.Использование пользователями процессорного времени

Распределение процессоро-часов (в %) по пользователям показано на рисунке 5. Вид зависимости почти совпал для обоих кластеров. Так, 4 пользователя с максимальным количеством процессоро-часов используют 50% всего времени, 10 пользователей – 80%, 15 пользователей – 90%.

За 2 года работы SVA на кластере Ant, были обнаружены некоторые характерные случаи неэффективного выполнения программ. На рисунке 6 изображены две характеристики вычислительного узла, использование центрального процессора (`cpu`) и работа с файлом подкачки(`pgp_out`). Данные программы превышают доступный объём ОЗУ, и процессор значительную часть времени простаивает.

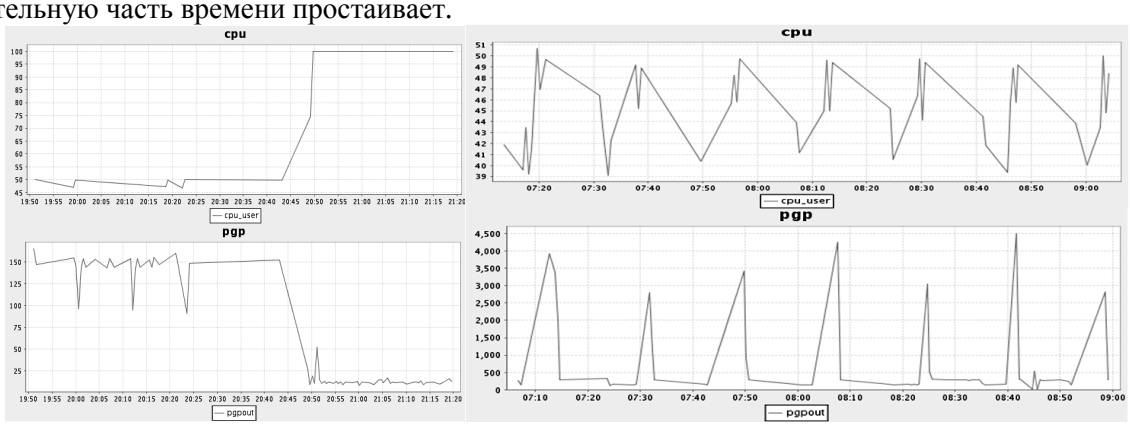

Рисунок 6. Примеры неэффективного выполнения программы

5. Заключение

С ростом вычислительных систем становится более сложной и задача определения эффективности их работы. Для решения этой задачи предложен программный комплекс SVA, обес-

печивающий хранение, визуализацию и анализ данных. Администратору вычислительной системы комплекс SVA помогает определить неэффективности в использовании, пользователю – приспособленность его задач к данной вычислительной системе, оценить эффект от изменений алгоритма, реализации программы или размерности задачи.

Приоритетами развития SVA является обеспечение масштабируемости и запуск на кластере СКИФ МГУ «Чебышев» в полном режиме в качестве постоянного средства; увеличение числа алгоритмов автоматического анализа эффективности работа узла, задачи и кластера в целом; предоставление пользователям отчётов по результатам их работы на кластере; создание модельного пакета задач для сертификации вычислительных систем.

6. Литература

1. Robert W. Wisniewski, Bryan Rosenburg. Efficient, Unified, and Scalable Performance Monitoring for Multiprocessor Operating Systems. SC2003, November 15-21, 2003, Phoenix, Arizona, USA.
2. Evan Hoke, Jimeng Sun, John D. Strunk, Gregory R. Ganger, Christos Faloutsos. InteMon: Continuous Mining of Sensor Data in Large-scale Self-* Infrastructures. ACM SIGOPS Operating Systems Review. Vol 40 Issue 3 P. 38-44. July, 2006. ACM Press.
3. Sacerdoti, F.D. Katz, M.J. Massie, M.L. Culler, D.E. Wide area cluster monitoring with Ganglia. IEEE International Conference on Cluster Computing, 2003. Proceedings. P. 289-298.
4. M. J. Sottile and R. Minnich. Supermon: A high-speed cluster monitoring system. IEEE International Conference on Cluster Computing, 2002. Proceedings. P. 39–46.
5. PARMON: A Monitoring System for Clusters of Computers [http://www.gridbus.org/~raj/parmon/]
6. Zabbix [http://www.zabbix.com]
7. Nagios [http://www.nagios.org]
8. А.Н. Андреев, А.С. Антонов, Вл.В. Воеводин, С.А. Жуматий Комплексный подход к анализу эффективности программ для параллельных вычислительных систем Высокопроизводительные вычисления и их приложения. Труды научной конференции, пос. Черноголовка, 2000, Изд-во МГУ. С. 18-20.
9. С.А. Жуматий. Система анализа производительности параллельных программ на кластерных установках. Вычислительны методы и программирование. 2005. Раздел 2. С. 57-64.
10. Воеводин Вл.В., Жуматий С.А. "Вычислительное дело и кластерные системы" М.: Изд-во МГУ, 2007. 150 с.
11. Kevin A. Huck et al. Capturing Performance Knowledge for Automated Analysis. SC2008, Austin, Texas, 2008. Proceedings.
12. Jeffrey S. Vetter, Daniel A. Reed. Managing Performance Analysis with Dynamic Statistical Projection Pursuit. SC1999, November, 1999, Portland, Oregon. Proceedings.
13. D.A. Reed, O.Y. Nickolayev, P.C. Roth. Real-Time Statistical Clustering for Event Trace Reduction. International Journal of High Performance Computing Applications, Vol. 11, No. 2, P. 144-159.