

УДК 681.5

СИСТЕМА МОНИТОРИНГА ТОЧЕК ПРЕДЕЛЬНОЙ НАГРУЗКИ В КЛАСТЕРНЫХ ВЫЧИСЛЕНИЯХ

К. В. ИВАНОВ

Ivanov.konstantin.89@gmail.com

ФГУП «Всероссийский научно-исследовательский институт автоматизи» (ВНИИА)

Поступила в редакцию 7 июля 2014 г.

Аннотация. Представлен прототип разрабатываемой системы мониторинга точек предельной нагрузки в кластерных вычислениях. В статье описывается подход к системе мониторинга высокопроизводительных вычислительных ресурсов как к программной компоненте, ориентированной на приложения. Описывается архитектура и общая схема работы системы. Приводятся алгоритмы поиска точек предельной нагрузки в поведении расчета.

Ключевые слова: кластер; мониторинг; вычислительные ресурсы; оптимизация.

Решение сложных научно-инженерных задач в большинстве случаев может выполняться только с помощью суперкомпьютеров [1, 2]. Современные суперкомпьютеры состоят из большого количества элементов и могут обладать довольно сложной архитектурой. При этом каждая расчетная задача обладает своим внутренним механизмом параллелизма. Вследствие различия внутренних архитектур приложений возникают ситуации, при которых суперкомпьютерные системы могут работать неэффективно [3].

Для повышения эффективности работы суперкомпьютеров необходимо выявлять вычислительные задачи, производительность которых можно увеличить. Так как эффективность работы приложения напрямую зависит от степени соответствия внутренней архитектуры приложения к предоставляемым системным ресурсам [4], следовательно, существуют два способа оптимизации работы приложений:

1) Изменение внутренней архитектуры приложения – данный метод в большинстве случаев недоступен вследствие отсутствия возможности редактирования исходного кода приложения, а в других случаях редко бывает востребован из-за возникающей негибкости вычислительной системы, под которую изменяется приложение.

2) Изменение предоставляемых расчету вычислительных ресурсов – применимость данно-

го метода в основном зависит от возможности имеющихся кластеров предоставлять различные архитектуры. Данный метод может быть использован на «универсальных вычислительных системах», которые зачастую обладают целым набором возможных архитектур вычислений.

Основными параметрами, на которые стоит обращать внимание при выборе метода оптимизации вычислений на суперкомпьютерах, являются: количество различных решателей, используемых в системе, частота изменений расчетных моделей и интенсивность появления нового инструментария для вычислений. Так, например, если имеется только один решатель со слабо изменяющейся моделью, одним из наиболее оптимальных способов повышения производительности может являться построение специфичной архитектуры суперкомпьютера для данной задачи [5].

Одной из наиболее распространенных ситуаций, при которых наблюдается потеря производительности, является нехватка одного из запрашиваемых ресурсов в системе. Состояние, при котором расчетная задача теряет производительность вследствие нехватки запрашиваемого ресурса, будем называть точкой предельной нагрузки. Для выявления таких состояний во многих вычислительных центрах используются системы мониторинга вычислительных ресурсов. Их задача состоит в сборе / регистрации, хранении и анализе ключевых (явных или косвенных) признаков/параметров описания данного объекта для вынесения суждения о поведении/состоянии данного объекта в целом.

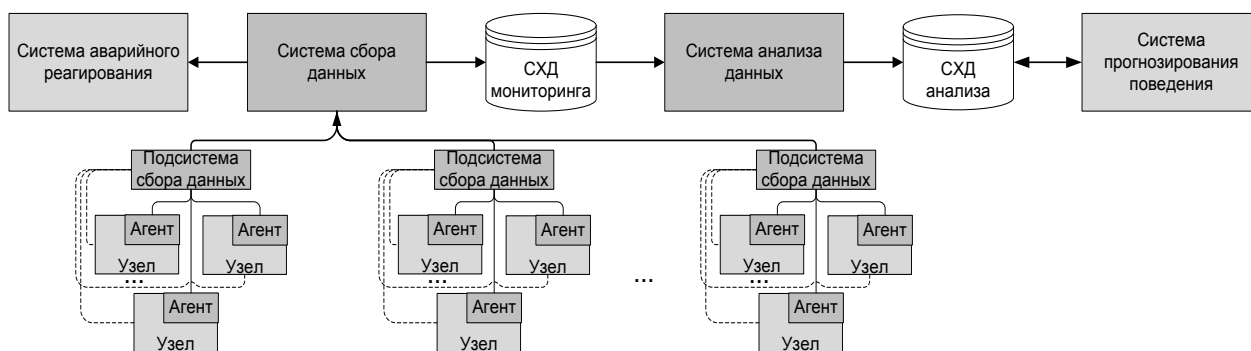


Рис. 1. Общая архитектура системы мониторинга точек предельной нагрузки

В более широком определении система мониторинга может включать в себя систему обратной связи с объектом, для поддержания его в каком-либо состоянии путем непосредственного управления [5].

Классические системы мониторинга суперкомпьютеров (Ganglia, Zabbix, OVIS) [6] предоставляют функционал для получения информации о загрузке основных аппаратных составляющих вычислительной системы, но не отслеживают зависимости утилизации системы от выполняющейся задачи. Такой подход может дать лишь общие представления об использовании системы, но является малоэффективным для определения точек предельной нагрузки в расчетных приложениях.

В данной статье описывается модель системы мониторинга, которая ориентирована на пользовательские расчетные задачи. Разработанная модель позволяет найти точки предельной нагрузки, характерные как для определенной модели расчета, так и для определенного решателя или группы решателей системы. Также приводятся примеры алгоритмов автоматического поиска «узких мест» в задачах и оптимизации последующих запусков путем перераспределения ресурсов, предоставляющихся расчету.

ОБЩАЯ МОДЕЛЬ СИСТЕМЫ

Общая архитектура предлагаемой системы мониторинга точек предельной нагрузки (СМТП) (рис. 1) предполагает размещение собственных агентов на вычислительных узлах системы, которые предоставляют информацию о текущей загрузке аппаратных ресурсов узла (таких как процессор, оперативная память, сеть обмена и управления, жесткий диск и т. п.). Помимо этого, необходимы элементы централизованного опроса устройств, предоставляющих информацию об узлах, но при этом недоступные операционной системе вычислительного компо-

нента. К таким устройствам можно отнести VMC (IPMI) адаптеры и общую доступность узла.

Вся собранная информация поступает на подсистемы сбора данных, которые необходимы для снижения требований обработки базы данных, путем разделения больших расчетных сегментов на группы. Основным преимуществом данного решения является повышение масштабируемости при увеличении расчетного сегмента кластера.

Данные с подсистем мониторинга поступают в центральную систему мониторинга, которая завершает процедуру группировки данных и записывает их в базу данных. Описанные выше компоненты являются отчуждаемыми и могут работать независимо от остальной системы, предоставляя данные об использовании и загрузке узлов системы в базу данных.

Также обработанные данные проходят через систему аварийного реагирования, которая необходима для принятия экстренных решений в различных нештатных ситуациях (скачки напряжения, выход из строя инженерного оборудования и т. п.)

Система прогнозирования поведения расчетных приложений анализирует уже обработанные данные и структурирует их согласно внутренним алгоритмам. Полученные знания о шаблонах поведения задач могут помочь при оптимизации выделения ресурсов задачи как путем автоматического перераспределения ресурсов, так и указаниями по увеличению производительности. Схематично взаимодействие планировщика с системой прогнозирования показано на рис. 2.

Помимо повышения утилизации системы в целом, данная модель позволяет выявлять и структурировать различные требования расчетных приложений к вычислительной системе, что позволяет создать статистику, которая позволит сформировать требование к наиболее оптимальной архитектуре кластера.

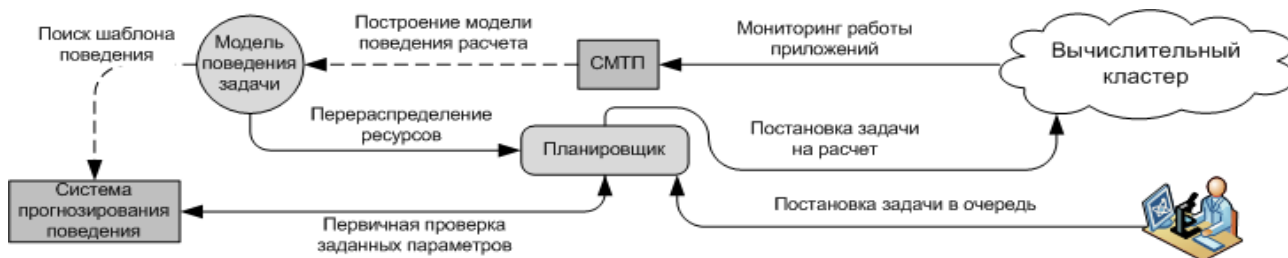


Рис. 2. Схема взаимодействия планировщика с системой прогнозирования

Однако вследствие большого разброса моделей поведения различных приложений, данный метод не всегда дает показательный результат.

После подтверждения задачи пользователем, все данные о расчете поступают в планировщик заданий. Планировщик сверяет реквизиты расчетной задачи с информацией системы прогнозирования о данном решателе на предмет нарушения статических правил оформления расчета. К статическим правилам можно отнести такие ограничения, как кратность запрашиваемых ресурсов, требования к подгружаемым модулям и т. п. При нарушении данных правил модель запрашиваемых ресурсов может автоматически корректироваться перед постановкой в очередь планировщика.

Во время выполнения расчета на вычислительном кластере система мониторинга собирает данные об утилизации аппаратных ресурсов задачей. На основе собранных данных строится модель поведения задачи, которая анализируется системой прогнозирования поведения на предмет наличия точек предельной нагрузки в ближайшем времени. Если получаемая модель, согласно алгоритму поиска соответствия, совпадает с шаблоном поведения, имеющим точки предельной нагрузки в ближайшем времени, то система мониторинга отправляет сигнал для планировщика о возможном неэффективном использовании ресурсов вычислителя.

На основе полученной информации от системы мониторинга точек предельной нагрузки планировщик заданий может перераспределить ресурсы для проблемной задачи в соответствии с используемой политикой планирования.

В данной работе были рассмотрены типовые случаи достижения точек предельной нагрузки в пользовательских расчетах, проводимых в области гидродинамики и теплогидравлики.

АНАЛИЗ ПРИЛОЖЕНИЙ

Сбор и анализ приложений проводились на вычислительном кластере ФГУП «ВНИИА», который обладает необходимой спецификой

расчетов для успешной оптимизации распределения вычислительных ресурсов (имеется небольшое количество используемых решателей, расчетные модели изменяются нечасто, многие из используемых решателей обладают сильной спецификой архитектуры распараллеливания).

При мониторинге точек предельной нагрузки расчетного приложения следует обращать внимание на все возможные ресурсы, которые может использовать расчет, такие как:

- Центральный процессор. При этом необходимо анализировать данные на уровне ядра, ведь неравномерная загрузка ядер является одним из признаков неэффективного использования ресурсов.
- Графические ускорители (если используются).
- Сети обмена.
- Сети управления. Данная метрика необходима из-за возможности использования сети управления вместо сети обмена вследствие неверно заданных параметров распараллеливания приложения.
- Система хранения данных.
- Оперативная память.

По собранным данным об использовании аппаратных ресурсов строится модель поведения задачи, которая имеет привязку к используемому решателю, модели расчета (данные о которой указываются пользователем) и пользователю системы. Визуализированная модель поведения конкретной задачи показана на рис. 3.



Рис. 3. График использования аппаратных ресурсов расчетной задачей (гидродинамика)

Для каждой расчетной задачи собирается подробная статистика, которая в дальнейшем классифицируется на группы подобия. Используемый алгоритм группировки элементов основан на поочередном сравнении точек каждого графиков и сравнении с задаваемым критерием подобия. Данный алгоритм является простым, но требовательным к ресурсам, что в рамках предоставленных вычислительных ресурсов и количества данных для анализа являлось наиболее оптимальным решением.

После анализа выполнения задач системой формировалось множество возможных моделей поведения $A = \{x_1, x_2, \dots, x_n\}$, при этом каждой модели поведения x_i ставились в соответствие параметры запуска задачи y_i . Точки предельной нагрузки можно определить как превалирующую загрузку одного из аппаратных ресурсов в условиях невысокого использования других ресурсов. Для каждого типа точек предельной нагрузки имеются определенные изменения конфигурации, позволяющие оптимизировать работу приложения (например, при сильной загрузке сети обмена – уменьшение количества узлов; при использовании большого количества оперативной памяти – увеличение количества узлов распараллеливания). Таким образом, все множество моделей делится на два подмножества: $B = \{y_i, \dots\}$ – подмножество моделей поведения с возможной оптимизацией и $C = \{x_2, \dots\}$ – подмножество моделей поведения с высокой производительностью. При этом существует множество переходов $G = B \Rightarrow C$, которое является сопоставлением наилучшему выделению ресурсов для конкретной задачи, что позволяет на основе знаний о поведении задачи оптимизировать утилизацию ресурсов вычислительного поля.

Основными достоинствами системы являются механизмы автоматизации выделения ресурсов, которые показывают большой прирост утилизации вычислительных ресурсов за счет выявления ошибок некорректно заданных требований к расчету на задачах со строгими правилами распараллеливания. К таким задачам можно отнести как целые решатели системы, которые могут выставлять специфичные требования к количеству расчетных ядер, так и задачи с определенными типами расчетных моделей.

Данные типы задач легко выявляются системой мониторинга на первоначальных этапах расчета из-за видимого простоя одного или нескольких ядер системы, и СМТП перезапускает задачу на количестве ядер, корректно использующихся в предыдущем расчете, что позволяет

сильно повысить утилизацию системных ресурсов.

Общее повышение производительности вычислительной системы зависит от качества задач и глубины понимания работы кластера пользователем, вследствие чего оценка производительности системы мониторинга точек предельной нагрузки является специфичной для каждого вычислительного центра. Но в любом случае СМТП помогает системным администраторам в первичном анализе запрашиваемых ресурсов и обнаружении мест предельной нагрузки на вычислительной системе.

ИТОГИ И ПЛАНЫ

В дальнейшем планируется дополнить СМТП более интеллектуальными алгоритмами классификации поведения расчетных задач для более раннего распознавания шаблонов модели и раннего реагирования на возможное появление точек предельной нагрузки на вычислительных кластерах. Также планируется добавить механизмы мониторинга инфраструктуры вычислителя с целью рассмотрения зависимости внешних параметров функционирования системы на поведение выполняемых расчетов. Полученные данные позволят уменьшить затраты на поддержание рабочего состояния кластеров без потери производительности путем нахождения баланса между проводимыми расчетами и инфраструктурными показателями системы.

СПИСОК ЛИТЕРАТУРЫ

1. Новиков А. Б., Петунин С. А. Влияние специализированных алгоритмов планирования заданий на эффективность использования вычислительных ресурсов в частных случаях // Супервычисления и математическое моделирование: тр. XIII междунар. сем. РФЯЦ-ВНИИЭФ, 2011. С. 213–215. [A. B. Novikov, S. A. Petunin. "Influence of specialized algorithms for scheduling efficient use of computing resources in particular cases," (In Russian), in *Proc. XIII Int. Seminar "Supercomputing and mathematical modeling"*, pp. 213-215, VNIIEF, 2011.]
2. Воеводин В. В., Воеводин Вл. В. Параллельные вычисления. СПб.: БХВ-Петербург, 2002. 608 с. [V. V. Voevodin, V. V. Voevodin, *Parallel computing*, (In Russian) St. Petersburg: BHV-Petersburg, 2002.]
3. Иванов К. В. Система мониторинга с прогнозированием ошибок // Параллельные вычислительные технологии (ПАВТ'2013). Челябинск, 2013. С. 592. [K. V. Ivanov, "Monitoring system with forecasting errors," (In Russian), in *PAVT'2013*, p. 592, Chlyabinsk, 2013.]
4. Ivanov K. V., Novikov A. B., Petunin S. A. Management of HPC clusters: development and maintenance // CSIT'2013: Proc. of the Workshop on Computer Science and Information Technologies (Vienna–Budapest–Bratislava, September 15–21, 2013). Vol. 2. P. 43–46. [K. V. Ivanov, A. B. Novikov, S. A. Petunin, "Management of HPC clusters: development and

maintenance," in Proc. of the Workshop on Computer Science and Information Technologies (CSIT' 2013), Vienna–Budapest–Bratislava, September 15–21, 2013, Vol.2, pp. 43–46.]

5. Адинец А. В., Брызгалов П. А., Воеводин В. В., Жуматий С. А., Никитенко Д. А., Стефанов К. С. JobDigest – подход к исследованию динамических свойств задач на суперкомпьютерных системах // Вестник УГАТУ. 2013. Т. 17, № 2 (55). С. 131–137. [A. V. Adinec, et al., "JobDigest - approach to the study of the dynamic properties of the tasks on the supercomputer system," (In Russian), *Vestnik UGATU*, vol. 17, no. 2 (55), pp. 131-137, 2013.]

6. OVIS-3 User's Guide / J. Brandt, et al. M.: Sandia National Laboratories Report, SAND2010-7109, 2010. [J. Brandt, et al., *OVIS-3 User's Guide*. M.: Sandia National Laboratories Report, SAND2010-7109, 2010.]

ОБ АВТОРЕ

ИВАНОВ Константин Владимирович, асп. Дипл. сист. программист (МИФИ, 2010). Готовит дис. о системе мониторинга для высокопроизв. выч. кластеров.

METADATA

Title: Monitoring points ultimate load of the cluster calculations.

Authors: K. V. Ivanov.

Affiliation:

All-Russia Research Institute of Automatics (VNIIA), Russia.

Email: ivanov.konstantin.89@gmail.com

Language: Russian.

Source: Vestnik UGATU (scientific journal of Ufa State Aviation Technical University), vol. 18, no. 3 (64), pp. 290-294, 2014. ISSN 2225-2789 (Online), ISSN 1992-6502 (Print).

Abstract: This paper presents a prototype system developed for monitoring the load limit points in cluster computing. The paper describes an approach to the monitoring system high performance s-productivity computing resources as system-oriented applications. The architecture and general scheme of the system are described. The algorithms of search points in the behavior of the limit load calculation.

Key words: cluster; monitoring; computing resources; optimization.

About the author:

IVANOV, Konstantin Vladimirovich, Postgrad. (PhD) Student, All-Russia Research Institute of Automatics (VNIIA). System Programmer (MePHI, 2010).