

Методы и средства балансировки нагрузки в неоднородных вычислительных системах

И.А. Алексеев, В.А. Егунов, С.В. Панюлайтис, А.А. Чекушкин

Волгоградский государственный технический университет

Аннотация: В настоящее время наблюдается стремительное развитие вычислительных систем, в том числе систем, включающих в свой состав устройства различного типа. Такие системы называют неоднородными или гетерогенными. В состав подобной системы помимо традиционных процессоров CPU могут входить графические GPU, программируемые вентильные матрицы, другие устройства. Одной из ярко выраженных проблем подобных систем является сложность распределения вычислительной нагрузки между узлами вычислительной системы. В данной работе описываются типы балансировки и методы распределения вычислительной нагрузки в неоднородных вычислительных системах. Описывается пример балансировки нагрузки в реальной гетерогенной вычислительной системе.

Ключевые слова: неоднородная система, гетерогенная система, балансировка нагрузки, распределение нагрузки, вычислительный кластер.

Введение. Настоящее время характеризуется высоким темпом развития информационных наук и роста потребности в вычислительных ресурсах. Стремительное повышение объёма обрабатываемой информации, усложнение приложений и используемых в них алгоритмов влечёт за собой увеличение нагрузки на электронно-вычислительные машины. При этом производительности вычислений, обеспечиваемой одной машиной, часто оказывается недостаточно. Всё это послужило толчком к развитию вычислительных систем – совокупности аппаратных и программных средств, образующих единую среду, предназначенную для решения объёмных и сложных задач. Вычислительные системы строятся на базе многих вычислительных элементов. Если вычислительная система состоит из вычислительных устройств различного типа, то она считается неоднородной (гетерогенной) [1]. В качестве элементов гетерогенной системы помимо традиционных процессоров (CPU) используются графические процессоры

(GPU), программируемые пользователем вентильные матрицы (FPGA) и интегральные схемы специального назначения (ASIC).

Гетерогенные вычислительные системы зачастую обладают потенциально большей вычислительной мощностью, чем однородные и позволяют решать задачи за меньший промежуток времени. Однако для решения задач на гетерогенных системах часто необходимо выполнить адаптацию алгоритмов решения задач. Затрачиваемые на адаптацию алгоритмов под гетерогенные системы ресурсы оправдывают себя, так как значительно возрастает эффективность вычислительной системы [2].

Целью настоящей работы является анализ современных алгоритмов балансировки нагрузки в неоднородных параллельных вычислительных системах. Также в работе приводится описание результатов, полученных в итоге выполнения работ по балансировке вычислительной нагрузки на кластере Волгоградского государственного технического университета.

Актуальность темы. Появление вычислительных систем к увеличению вычислительных мощностей, доступных в рамках одной среды, в которой выполняется пользовательское приложение. В свою очередь при переходе на вычислительные системы появился ряд проблем, не возникающих в рамках нераспределённой архитектуры. Одной из таких ярко выраженных проблем является необходимость балансировки нагрузки между узлами (элементами) вычислительной системы.

Появление данной проблемы привело к возникновению алгоритмов и методов балансировки нагрузки. Проблема балансировки вычислительной нагрузки распределённого приложения возникает по ряду причин:

- структура распределенного приложения неоднородна, различные логические процессы требуют различных вычислительных мощностей;

- структура вычислительного комплекса (например, кластера), также неоднородна, т.е. разные вычислительные узлы обладают разной производительностью;

- структура взаимодействия между узлами неоднородна, т.к. линии связи, соединяющие узлы, могут иметь различные характеристики пропускной способности.

Балансировка нагрузки предполагает решение двух задач. Первая – равномерно распределить нагрузку вычислительных узлов. Вторая – распределить нагрузку на каждом узле между разнородными вычислительными элементами. Кроме того, балансировка предполагает перенос части вычислений с наиболее загруженных вычислительных узлов на менее загруженные узлы, в связи с чем может меняться эффективность вычислительной системы в целом. Неверная балансировка может привести к перегрузке одного из узлов или к неэффективному использованию его ресурсов. Эффективное предоставление и планирование ресурсов обеспечат их доступность по требованию, сохранение высокой производительности при использовании ресурсов в условиях высокой / низкой нагрузки.

В настоящее время гетерогенные вычислительные системы достаточно широко распространены [3-5]. Примером такой системы может быть вычислительный кластер при Волгоградском государственном техническом университете. Характеристика данного вычислительного кластера будет представлена далее.

Типы балансировки вычислительной нагрузки. Различают статические и динамические методы балансировки нагрузки в вычислительных системах [6].

Статическая балансировка выполняется перед запуском приложения на вычислительной системе и предполагает, что структура системы в процессе выполнения задачи не меняется. При обеспечении отказоустойчивости узлов

и гарантированном качественном соединении между узлами плюсом данного типа балансировки является возможность запоминания результатов балансировки и последующее их применение (в т.ч. в рамках генетических алгоритмов балансировки). Однако использование статической балансировки в ряде случаев может не дать ожидаемого эффекта из-за того, что в ходе выполнения задачи вычислительная среда может измениться. Например, один или несколько вычислительных узлов могут выйти из строя; вычислительные узлы, на которых осуществлялось накопление статистики, в момент запуска могут быть заняты другими вычислениями и пр. Кроме того применение статической балансировки приводит к относительно большим накладным расходам из-за постоянного мониторинга имеющихся в наличии свободных ресурсов, а также задач, обрабатываемых вычислительной системой. Всё это требует дополнительных аппаратных средств.

Вторым типом балансировки, в противовес статической, является динамический подход к распределению нагрузки [7]. В отличие от статических методов, когда в системе задано только начальное распределение задач между узлами системы, в динамической системе задачи могут добавляться в процессе работы системы.

Данный тип балансировки учитывает изменчивость вычислительной системы во времени. Распределение и разделение нагрузки меняется во времени, обновляется в ходе выполнения пользовательского приложения и задачи распределяются в соответствии с новым, более оптимальным планом.

Использование на практике динамической балансировки рационально в ситуации, когда время, затрачиваемое на реализацию алгоритмов балансировки существенно меньше времени выполнения задачи.

Методы распределения нагрузки. Для балансировки нагрузки в гетерогенных вычислительных системах можно выделить три основных

метода: централизованная, распределённая и иерархическая балансировка [8-10].

В централизованной технике балансировки вычислительной нагрузки всё распределение и решение о планировании принимается одним узлом. Этот узел отвечает за хранение базы знаний и может применять статический или динамический подход для балансировки нагрузки. Такой прием при своей относительной простоте может создать большие накладные расходы на центральном узле. Также сеть больше не является отказоустойчивой, так как сбой центрального узла может привести к нарушению в работе всей вычислительной сети и восстановление работы центрального узла может потребовать определенного времени.

В методе распределенной балансировки нагрузки нет единого узла, отвечающего за подготовку ресурсов или выполнение задачи распределения ресурсов. Вместо этого несколько узлов контролируют сеть, чтобы выполнить точное распределение нагрузки. Алгоритмы, реализующие данный метод, используют теорию очередей.

Иерархическая балансировка нагрузки включает в себе различные уровни сети, участвующие в принятии решений. Она работает в режиме master / slave. Фактически подобный метод можно проиллюстрировать древовидной структурой, в которой каждый узел в дереве находится под наблюдением своего родительского узла.

Каждый из описанных методов имеет свои недостатки и сильные стороны. Наиболее гибким является распределённый метод, так как он обеспечивает наибольшую отказоустойчивость, при которой выполнение задачи балансировки нагрузки ложится на все узлы системы равномерно. Однако этот метод нецелесообразно применять для систем небольшого размера. Для них наилучшим решением будет централизованный метод в силу простоты своей реализации.

Балансировка нагрузки в реальных гетерогенных вычислительных системах. В качестве примера рассмотрим использование систем очередей в реальных задачах на гетерогенном кластере Волгоградского государственного технического университета.

На настоящий момент (октябрь 2020 года) кластер ВолгГТУ имеет следующую конфигурацию.

Таблица № 1.

Состав кластера ВолгГТУ

Количество узлов	ЦПУ	ОЗУ	Сопроцессоры
7	Xeon E5-2650v3 (x2)	128 Гб	-
2			Xeon Phi 31S1P (x2)
1			Nvidia GTX 1080Ti
1			Nvidia P100
8	Xeon Phi 7210	16 Гб (GDDR5) + 196 Гб (DDR4)	-
3	Xeon E5-2650v4 (x2)	256 Гб	-
1	Xeon(R) Gold-6130 (x2)	256 Гб	Nvidia V100 (x2)
1	Xeon E5-2660 (x2)	128 Гб	Xeon Phi 5110P
1	Xeon E5-2660 (x2)	128 Гб	Nvidia K20C

Для обеспечения сетевого взаимодействия между вычислительными узлами используется сетевое оборудование, имеющее достаточно высокую пропускную способность. В кластере ВолгГТУ используются сети Infiniband с пропускной способностью 54 Гбит/с и Ethernet 1 Гбит/с. С целью организации эффективного доступа к ресурсам кластера и балансировки вычислительной нагрузки на кластере был внедрен менеджер кластера SLURM.

До установки менеджера SLURM балансировка нагрузки в вычислительном кластере осуществлялась фактически «вручную». Причем часто этим занимались не только администраторы кластера, но и рядовые

пользователи. Они запускали свои задачи на произвольно выбранных узлах, что часто приводило к крайне несбалансированной нагрузке на узлы кластера. Установка SLURM в корне изменила ситуацию, использование кластера стало гораздо более эффективным.

Литература

1. Четверушкин Б.Н. Высокопроизводительные многопроцессорные вычислительные системы // Вестник РАН. – 2002. – Т. 72, № 9. – С. 786- 794.

2. Ластовецкий А.Л., Калинов А.Я., Ледовских И.Н., Арапов Д.М., Посыпкин Н. А. Язык и система программирования для высокопроизводительных параллельных вычислений на неоднородных сетях // Программирование. – 2000. – №4. – С. 55–80.

3. Кулакович А.Ю., Баранов Е.Ю. Оценка зависимостей времени работы алгоритма для восстановления расфокусированных изображений, выполняемого на CPU и GPU // Инженерный вестник Дона, 2019. - № 1. – URL: ivdon.ru/ru/magazine/archive/n1y2019/5518

4. Быков Д.В., Неретин А.Д. Прогнозирование производительности при реализации алгоритмов генерации случайных последовательностей больших размерностей на реконфигурируемых архитектурах с сопроцессорами // Инженерный вестник Дона, 2014. - № 2. – URL: ivdon.ru/ru/magazine/archive/n2y2014/2414

5. Копысов С. П., Кузьмин И. М., Недожогин Н. С., Новиков А. К. Параллельные алгоритмы формирования и решения системы дополнения Шура на графических ускорителях // Ученые записки Казанского ун-та. Серия Физико-матем. науки. – 2012. – Т. 154, Кн. 3. – С. 202–215.

6. Юрич, М. Ю. Анализ систем и методов балансировки нагрузки вычислительных систем // Научный вестник Черновецкого университета, 2010, Т.1, вып.1. – С. 14-19.

7. Копысов С.П. Динамическая балансировка нагрузки для параллельного распределенного МДО // Методы и средства обработки информации / Под ред. Королева Л.Н. – М.: МГУ, 2003. – С. 222–227.

8. Kopysov S. P., Novikov A. K. Parallel adaptive mesh refinement with load balancing on heterogeneous cluster // Algorithms and Tools for Parallel Computing on Heterogeneous Clusters: Nova Science Publishers, 2007. – Pp. 45–53.

9. Bhandarkar M., Kale L. V., de Sturler E., Hoeflinger J. Adaptive Load Balancing for MPI Programs // Lecture Notes in Computer Science. – 2001. – № 2074. – pp. 108–117.

10. Dikken L., Linden F., Vasseur J., Sloot P. DynamicPVM : Dynamic load balancing on parallel systems // Lecture Notes in Computer Science. – 1994. – No 797. – pp. 273–277.

References

1. Chetverushkin B.N. Vestnik RAN. 2002. V.72. №9. Pp.786 - 794.

2. Lastoveckij A.L., Kalinov A.YA., Ledovskih I.N., Arapov D.M., Posypkin N. A. Programirovanie. 2000. №4. Pp. 55–80.

3. Kulakovich A.YU., Baranov E.YU. Inzhenernyj vestnik Dona, 2019. № 1. URL: ivdon.ru/ru/magazine/archive/n1y2019/5518

4. Bykov D.V., Neretin A.D. Inzhenernyj vestnik Dona, 2014. № 2. URL: ivdon.ru/ru/magazine/archive/n2y2014/2414

5. Kopysov S. P., Kuz'min I. M., Nedozhogin N. S., Novikov A. K. Uchenye zapiski Kazanskogo un-ta. Seriya Fiziko-matem. nauki. 2012. V. 154, №. 3. Pp. 202–215.

6. Yurich, M. Yu. Nauchny`j vestnik Chernoveczkogo universiteta. 2010, V.1, r.1. Pp.14-19.

7. Kopysov S.P. Dinamicheskaya balansirovka nagruzki dlya parallel`nogo raspredelenного MDO [Dynamic load balancing for parallel distributed MDO].



Metody` i sredstva obrabotki informacii. Pod red. Koroleva L.N. M: MGU, 2003. Pp.222-227.

8. Копысов С. П., Новиков А. К. Algorithms and Tools for Parallel Computing on Heterogeneous Clusters: Nova Science Publishers, 2007. Pp. 45–53.

9. Bhandarkar M., Kale L. V., de Sturler E., Hoeflinger J. Lecture Notes in Computer Science. 2001. № 2074. Pp. 108–117.

10. Dikken L., Linden F., Vesseur J., Sloot P. Dynamic PVM: Dynamic Lecture Notes in Computer Science. 1994. № 797. Pp. 273–277.