

ОБ ЭФФЕКТИВНОСТИ АЛГОРИТМОВ ПЛАНИРОВАНИЯ ЗАДАЧ УПРАВЛЕНИЯ ПОТОКАМИ ДАННЫХ ОБЛАЧНЫХ ГРИД-СИСТЕМ

В статье представлены результаты исследования с помощью симулятора облачной грид-системы разработанных алгоритмов планирования задач Backfill SDM и Backfill MDM в сравнение с существующим распространенным алгоритмом Backfill FF. Исследование показало преимущество алгоритма Backfill SDM по критериям производительности и сбалансированности. Дополнительно было проведено исследование сочетания данного алгоритма с обычным и разработанными реактивным и проактивным методами управления потоками данных в программно-конфигурируемой сети, которое показало преимущество комбинации Backfill SDM с проактивным методом. Сочетание алгоритма Backfill SDM с проактивным методом управления потоками данных может быть рекомендовано для практического внедрения в системы управления реальными вычислительных систем – кластеров и грид-систем.

Ключевые слова: планирование задач, облачные вычисления, программно-конфигурируемые сети, методы управления потоками данных.

В настоящее время в различных областях науки, техники и экономики возникают потребности в решении вычислительно трудоемких задач. Они могут быть решены с применением высокопроизводительных вычислений, для которых используют вычислительные кластеры и грид-системы.

Существующие системы управления ресурсами высокопроизводительных систем недостаточно эффективны. Как правило, они используют алгоритмы планирования, представляющие собой сочетания алгоритма Backfill [1] с одним из простых методов назначения, например, с First Fit (FF) или Best Fit [2, 3]. В среднем системы управления ресурсами обеспечивают загрузженность вычислительных ресурсов на уровне 60-80%.

Имеется возможность увеличения их эффективности за счет учета коммуникационных схем вычислительных задач и маршрутов передачи данных при планировании задач.

Действительно, если известна коммуникационная схема вычислительной задачи и текущее состояние высокопроизводительной сети, включая ее топологию и текущие задержки на портах сетевых связей, то можно назначать процессы задачи на вычислительные ядра максимально компактно. Это позволит локализовать сетевой трафик и снизить сетевую конкуренцию.

Также, если высокопроизводительная система построена на базе программно-конфигурируемой сети (ПКС) [4], то можно, зная коммуникационную схему вычислительной задачи,

проложить заранее (до запуска задачи) маршруты передачи данных между каждой парой ее взаимодействующих процессов, стремясь к снижению сетевой конкуренции. Подобный подход позволяет сократить время выполнения сетевых операций, что приводит к существенному уменьшению времени выполнения коммуникационно-интенсивных задач и увеличению загрузки вычислительных ресурсов высокопроизводительной системы.

В настоящее время отсутствуют системы управления ресурсами на основе ПКС для высокопроизводительных систем, направленные одновременно на управление маршрутами передачи данных между процессами вычислительных задач и топологическую локализацию их размещения. В этом заключается актуальность и новизна данной работы.

Для решения данной проблемы ранее автором были разработаны [5]:

1. Структурная модель распределенного ЦОД с ПКС его сегментов в виде возмещенного ориентированного мультиграфа его сегментов, каждый из которых состоит из вычислительных узлов, обычных коммутаторов, коммутаторов OpenFlow, граничных шлюзов, балансировщиков коммутаторов, контроллеров OpenFlow и хранилищ данных.

2. Модель вычислительного облака, развернутого поверх ЦОД с ПКС его сегментов. Аналогична структурной модели, отличие в наличии виртуальных устройств – вычислительных узлов, обычных коммутаторов и коммутаторов OpenFlow.

3. Модель грид-системы, функционирующей поверх вычислительного облака.

4. Модель вычислительной задачи в виде ориентированного ациклического графа ее подзадач, для каждой из которых известна коммуникационная схема выполнения.

Данные модели послужили основой разработанного симулятора облачной грид-системы, используемого для исследования алгоритмов планирования задачи и методов управления потоками данных.

Также автором ранее были разработаны алгоритмы планирования задач, представляющие собой сочетания классического алгоритма выбора задачи из очереди Backfill и предложенных методов ее назначения на вычислительные ресурсы Summed Distance Minimization (SDM) и Maximum Distance Minimization (MDM) [6]. Ранее разработанные методы управления потоками данных для ПКС описаны в работе [6] – это реактивный метод (РМ), устанавливающий правила маршрутизации в таблицы коммутаторов OpenFlow во время выполнения вычислительных задач при появлении новых потоков данных, и проактивный метод (ПМ), устанавливающий правила до запуска задач на основе их коммуникационных схем.

Исследование алгоритмов планирования и методов управления потоками данных с помощью симулятора облачной грид-системы, является основной целью данной статьи.

Компоненты алгоритма планирования задач

Исследуемые алгоритмы планирования задач для облачной грид-системы включает в себя две компоненты:

а) алгоритм выбора следующей задачи из очереди – на основе информации о доступных вычислительных ресурсах, текущем расписании и состоянии очереди вычислительных задач выбирает следующую задачу для планирования;

б) метод назначения задач на вычислительные ресурсы – для выбранной задачи по некоторому принципу (например, топологической локальности) подбирает подходящие вычислительные узлы и назначает ее на них.

Алгоритмы планирования задач работают в тесной интеграции с методами управления потоками данных, которые маршрутизируют потоки данных между вычислительными узлами, назначенными процессам задач. Могут быть использованы как классические методы управ-

ления потоками данных, так и построенные на базе программно-конфигурируемых сетей.

Система критериев и метрик количественного оценивания эффективности работы алгоритмов планирования задач

Для адекватной оценки эффективности работы алгоритмов планирования облачной грид-системы был разработан набор из следующих метрик:

а) Время выполнения эталонного набора задач ΔT . Определяется, как разность между временем T_1 завершения всех задач набора и временем T_0 получения первой задачи:

$$\Delta T = T_1 - T_0.$$

б) Средняя загрузка вычислительных ядер высокопроизводительной системы \bar{U} . Вычисляется, как среднее арифметическое средних загрузок \bar{U}_{ij} каждого вычислительного ядра системы:

$$\bar{U} = \frac{\sum_{i=1}^n \sum_{j=1}^{C_i} \bar{U}_{ij}}{\sum_{i=1}^n C_i}.$$

Единицами измерения \bar{U} и \bar{U}_{ij} являются проценты.

в) Индекс сбалансированности загрузки ресурсов вычислительных ядер σ . Может быть вычислен по формуле:

$$\sigma = \sqrt{\frac{\sum_{i=1}^n (\bar{U}_i - \bar{U})^2}{n-1}}.$$

Данный показатель позволяет определить равномерность использования вычислительных ресурсов.

Метрики а) и б) формируют критерий производительности работы алгоритмов планирования, метрика в) – критерий сбалансированности загрузки ресурсов вычислительной системы.

Методика экспериментального исследования алгоритмов планирования задач для облачных грид-систем с помощью симулятора

Исследование разработанных алгоритмов планирования задач проводилось с помощью созданного симулятора облачной грид-системы, поддерживающего возможности симуляции вычислительного процесса и работы программ-

но-конфигурируемой сети. Для генерации потока вычислительных задач используются законы распределения случайных величин и их параметры из работы [7].

Основным достоинством применения симулятора является гибкость, позволяющая достаточно просто варьировать конфигурацию вычислительной системы, а также изменять характеристики потока вычислительных задач. Другое достоинство – экономия ресурсов реальной вычислительной системы.

Оценка эффективности совместной работы алгоритмов планирования и методов управления по описанным критериям и метрикам должна проводиться в зависимости от системной загрузки и коммуникационной интенсивности задач (доли времени коммуникаций в общем времени их выполнения). При этом должны строиться графики зависимости исследуемых метрик от величины системной загрузки, определяемой по интенсивности генерируемого модельного потока вычислительных задач. Варьируется параметр интенсивности экспоненциального распределения интервала времени между поступлением задач в очередь системы.

С целью обеспечения достоверности получаемых показателей экспериментальные исследования для одних и тех же параметров модельного потока задач и конфигурации вычислительной системы многократно повторяются, а получаемые значения метрик усредняются.

Результаты исследования

Объектами исследования являются алгоритмы планирования задач – Backfill FF, Backfill SDM и Backfill MDM, а также методы управления потоками данных – ОМ (обычный метод), РМ (реактивный метод), ПМ (проактивный метод).

В симуляторе была выбрана конфигурация исследуемой моделируемой системы, которая включает в себя 2 сегмента, каждый из которых включает в себя 4 коммутатора OpenFlow, 8 вычислительных узлов (32 Гб оперативной памяти, 4 вычислительных ядра) и контроллер OpenFlow. К каждому коммутатору подключено два вычислительных узла, коммутаторы связаны друг с другом по принципу кольца. Контроллер OpenFlow подключен к каждому коммутатору напрямую. Сегменты связаны между собой сетью Интернет.

Симуляция проводилась на двух эталонных наборах задач: вычислительно-интенсивном (доля коммуникаций не более 30%) и ком-

муникационно-интенсивном (доля коммуникаций не менее 70%).

На первом шаге исследования использовался обычный метод управления потоками данных в сочетании с исследуемыми алгоритмами планирования. Полученные результаты для коммуникационно-интенсивного набора задач приведены на рисунках 1–3.

Рисунок 1 иллюстрирует зависимость времени выполнения коммуникационно-интенсивного набора задач ΔT от величины системной загрузки. Худшее время обеспечивает алгоритм Backfill FF, наилучшие результаты у Backfill SDM, чуть хуже проявил себя алгоритм Backfill MDM. Backfill SDM по отношению к алгоритму Backfill FF обеспечивает снижение времени выполнения коммуникационно-интенсивного набора задач на 4–6%.

На рисунке 2 приведена зависимость средней загруженности вычислительных ядер \bar{U} от величины системной загрузки для коммуникационно-интенсивного набора задач. Лучшее из всех по данному показателю проявил себя алгоритм Backfill SDM (обеспечивает увеличение загруженности вычислительных ядер до 4,5% по сравнению с Backfill FF), несколько хуже результаты у Backfill MDM. Значительное отставание у Backfill FF.

Следует заключить, что по критерию производительности для коммуникационно-интенсивного набора задач при обычном методе управления сетью наилучшим алгоритмом является разработанный Backfill SDM.

По критерию сбалансированности загрузки вычислительных ресурсов оба алгоритма Backfill SDM и Backfill MDM показали хорошие результаты, несколько хуже результаты у Backfill FF (рисунок 3).

Также исследование выявило, что для вычислительно-интенсивного набора задач результаты работы алгоритмов планирования Backfill SDM и Backfill MDM очень близки и обеспечивают уменьшение времени выполнения эталонного набора задач по сравнению с Backfill FF до 1,5%. Это связано с тем, отсутствует выигрыш от топологически локализованного размещения задач, которые большую часть времени своего исполнения тратят на вычисления, а не на обмен информацией между своими процессами. В данном сценарии разработанные алгоритмы Backfill SDM и Backfill MDM также продемонстрировали хороший уровень сбалансированности загрузки вычислительных ресурсов.

ным и проактивным методами управления потоками данных в сети.

На рисунке 4 приведены графики зависимости времени выполнения коммуникационно-интенсивного набора задач ΔT от величины системной загрузки для различных методов управления потоками данными.

По графикам видно, что использование РМ приводит к увеличению времени выполнения эталонного набора задач до 1,2%. Это связано с тем, что правила маршрутизации устанавливаются в коммутаторы OpenFlow

во время выполнения вычислительных задач, что приводит к дополнительным накладным расходам. ПМ, напротив, устанавливает правила маршрутизации в коммутаторы OpenFlow до запуска вычислительных задач, поэтому он обеспечивает снижение времени выполнения эталонного набора задач до 1,5%. Схожие результаты получаются по остальным показателям системы метрик и критериев для коммуникационно-интенсивного набора задач. Таким образом, для коммуникационно-интенсивного набора задач лидером является

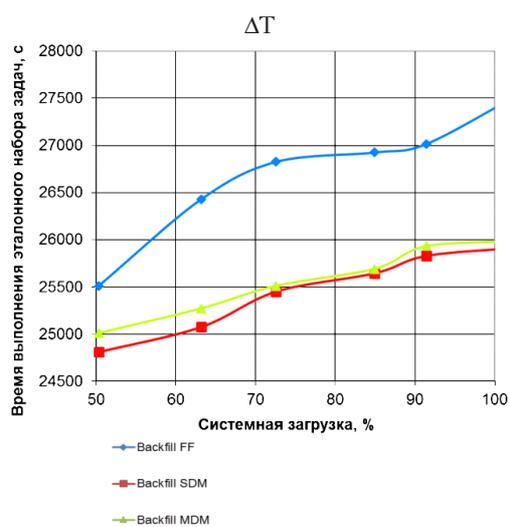


Рисунок 1. Зависимость времени выполнения коммуникационно-интенсивного набора задач ΔT от величины системной загрузки

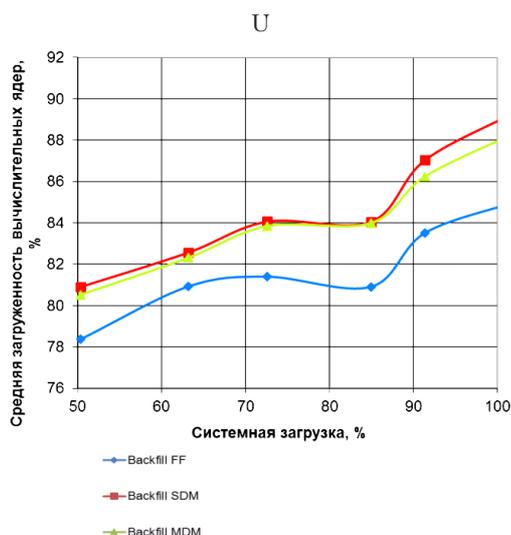


Рисунок 2. Зависимость средней загруженности вычислительных ядер \bar{U} от величины системной загрузки для коммуникационно-интенсивного набора задач

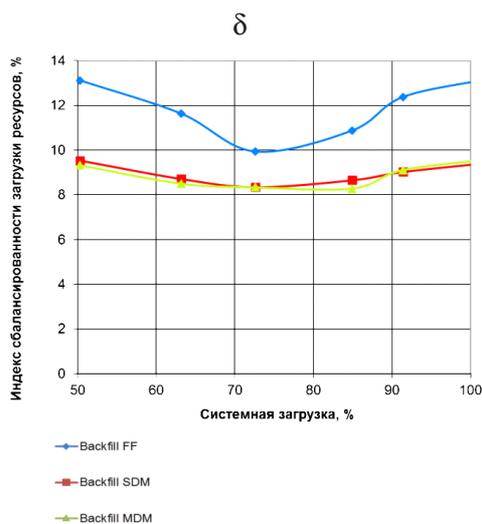


Рисунок 3. Зависимость индекса сбалансированности загрузки ресурсов σ от величины системной загрузки для коммуникационно-интенсивного набора задач

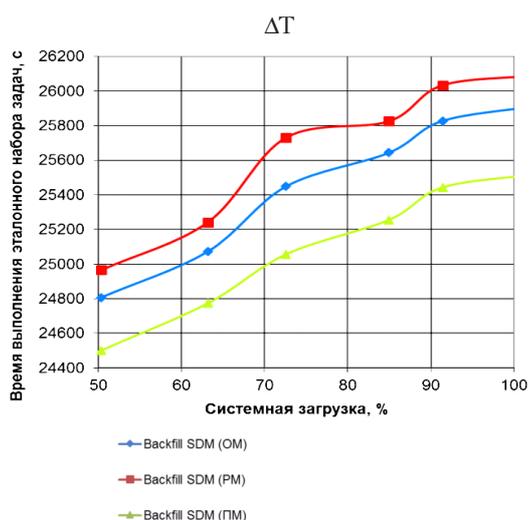


Рисунок 4. Зависимость времени выполнения вычислительно-интенсивного набора задач ΔT от величины системной загрузки для различных методов управления потоками данными

Обсуждение полученных результатов

Разработан алгоритм планирования Backfill SDM, который в сочетании с обычным методом управления потоками данных обеспечивает снижения времени выполнения коммуникационно-интенсивного и вычислительно-интенсивного наборов задач по сравнению с алгоритмом Backfill FF до 4–6% и 1,5% соответственно, что говорит о его высокой производительности. Также данный алгоритм показал высокий уровень сбалансированности загрузки вычислительных ресурсов.

Другой разработанный алгоритм Backfill MDM по результатам исследования продемонстрировал по сравнению с Backfill SDM более худшие результаты и не может быть рекомендован для практического использования.

Сочетание алгоритма планирования Backfill SDM с проактивным методом управления потоками данных позволяет достичь дополнительно снижения времени выполнения коммуникационно-интенсивного и вычислительно-интенсивного наборов задач по сравнению с обычным методом до 1,5% и 0,5% соответственно.

Выводы

С помощью симулятора облачной грид-системы было проведено исследование разработанных алгоритмов планирования задач Backfill SDM и Backfill MDM в сравнение с существующим распространенным алгоритмом Backfill FF. Исследование показало преимущество алгоритма Backfill SDM по критериям производительности и сбалансированности.

Дополнительно было проведено исследование сочетания данного алгоритма с обычным и разработанными реактивным и проактивным методами управления потоками данных в программно-конфигурируемой сети, которое показало преимущество комбинации Backfill SDM с проактивным методом.

Сочетание алгоритма Backfill SDM с проактивным методом управления потоками данных может быть рекомендовано для практического внедрения в системы управления реальных вычислительных систем – кластеров и грид-систем.

24.12.2013

Исследования выполнены при поддержке Министерства образования и науки Российской Федерации в рамках проекта №07.514.11.4153 ФЦП «Исследования и разработки по приоритетным направлениям развития научно-технологического комплекса России на 2007–2013 годы» и РФФИ (проекты №12-07-31089 и №13-07-97046).

Список литературы:

1. Feitelson, D. Utilization and predictability in scheduling the IBM SP2 with backfilling / D. Feitelson, A. Weil // Proceedings of the First Merged International Parallel Processing Symposium and Symposium on Parallel and Distributed Processing. – 1998. – С. 542-546.
2. Коваленко, В.Н. Использование алгоритма Backfill в грид / В.Н. Коваленко, Д.А. Семячкин // Труды международной конференции «Распределенные вычисления и Грид-технологии в науке и образовании». – 2004. – с. 139-144.
3. Полежаев, П.Н. Симулятор вычислительного кластера и его управляющей системы, используемый для исследования алгоритмов планирования задач // Вестник ЮрГУ. Серия «Математическое моделирование и программирование», 2010. – №35(211), вып. 6. – С. 79 – 90.
4. McKeown N., Anderson T., Balakrishnan H., Parulkar G., Peterson L., Rexford J., Shenker S., Turner J.; Openflow: enabling innovation in campus networks // ACM SIGCOMM Computer Communication Review, 2008, vol. 38, p. 69-74.
5. Полежаев П.Н. Математическая модель распределенного вычислительного центра обработки данных с программно-конфигурируемыми сетями его сегментов // Вестник «Оренбургского государственного университета», 2013. – 5(154) – С. 198-204.
6. Полежаев П.Н., Ушаков Ю.А., Шухман А.Е. Система управления ресурсами для высокопроизводительных вычислений, основанная на использовании программно-конфигурируемой сети // Системы управления и информационные технологии, №4(54), 2013. – С. 65-69
7. Lublin, U. The workload on parallel supercomputers: modeling the characteristics of rigid job [Текст] / U. Lublin, G. Feitelson // Journal of Parallel and Distributed Computing archive. – 2003. -Vol. 63, issue 11. – с. 542 – 546

Сведение об авторе:

Полежаев Петр Николаевич, преподаватель кафедры компьютерной безопасности и математического обеспечения информационных систем математического факультета

Оренбургского государственного университета

460018, г. Оренбург, пр. Победы, 13, ауд. 20520, тел.: (3532) 372534, e-mail: peter.polezhaev@mail.ru