Адаптивное оценивание параметров в параллельных многопользовательских вычислительных системах[1]

А. Т. Вахитов
Санкт-Петербургский государственный университет
av38@yandex.ru

Рассматривается задача оценки параметров параллельной вычислительной системы по наблюдениям за выполнением заданий. Предложены постановки задач и возможные пути решения для оценки производительности узлов, пропускной способности каналов передачи данных, трудоемкости заданий, с учетом изменения указанных параметров с течением времени. Полученные результаты оценивания могут быть использованы алгоритмом планирования распределения заданий. На основе изложенных идей в лаборатории СПРИНТ разрабатывается сервис-брокер для распределения заданий при моделировании экспериментов адронной терапии.

Ключевые слова: распределение заданий, параллельные вычисления, линейное оценивание, стохастическая оптимизация.

1. Введение

Благодаря опыту исследовательских и учебных проектов лаборатории СПРИНТ в области параллельных вычислений в Петергофском учебно-научном комплексе СПбГУ, выполняемых уже два года при поддержке Intel, можно смело утверждать, что уникальной особенностью Грид является количество вычислительных узлов, на которых могут быть запущены пользовательские задачи. Проект Folding@Home, достигший наибольшей среди всех вычислительных проектов в мире производительности в 4.1 PFLOPS [1], обязан этим результатом трем с половиной миллионам процессоров, которые в нем участвуют. “Грид” размером в несколько компьютеров на наш взгляд не интересен ни пользователям, ни владельцам ресурсов. Поэтому свою работу, результаты которой отражены в

[1]©А. Т. Вахитов, 2008

этой статье мы сосредоточили на интеграции собственных ресурсов для работы в рамках большой Грид-системы, такой как NorduGrid. Существенными особенностями системы такого класса мы считаем также большое количество пользователей и разнообразие заданий.

Для интеграции ресурсов мы используем инструментарий Grid Programming Environment (http://gpe4gtk.sf.net), разработанный компанией Intel на основе Globus Toolkit (http://www.globus.org). GPE является простым решением для многих сложных проблем интеграции и интероперабельности. В то же время, GPE не является готовой системой, а лишь набором отдельных инструментов. В этой статье мы обсудим расширение GPE - сервис-брокер для распределения заданий с учетом изменяющихся характеристик вычислительных узлов и каналов передачи данных.

2. Общая постановка задачи

Грид-вычисления характеризуются гетерогенностью. Это свойство означает сильные различия между производительностями компьютеров и пропускными способностями сетевых соединений, задействованных в Грид-системе. Принимая это во внимание, задача распределения заданий с целью минимизации времени вычислений при имеющемся количестве ресурсов выглядит в случае Грид особенно сложной. Усугубляет ситуацию непредсказуемость поведения отдельных вычислительных узлов, неизбежно возникающая для достаточно больших Грид-систем: возможность узлов отключаться от процесса вычислений либо изменять собственную производительность в произвольные моменты времени. Задания, отправляемые пользователями для вычисления в Грид, неравномерны по сложности; их достаточно много, так что невозможно вычислить их все на одном компьютере, чтобы затем сопоставить трудоемкость вычислений и составить некоторые прогнозы для времени их вычисления на других компьютерах. Будем считать, что все коммуникации с внешними по отношению к вычислительному узлу, на котором выполняется задание, отсутствуют. Это означает, что задания не выполняют запросов к базе данных либо к другим вычислительным узлам в процессе вычислений.

Такое ограничение является существенным. Если бы мы хотели избежать его, то при планировании вычислений пришлось бы

учитывать некоторую меру близости заданий между собой, с базой данных, и сопоставлять эти величины с физической близостью вычислительных узлов и источников данных.

Грид-система, которую мы будем рассматривать, состоит из двухуровневой иерархии. Можно называть ее “сегментом” Грид. Согласно определению [2], сегмент Грид обладает следующими особенностями:

1. Недостаток централизованного контроля (остутствие централизованного управления политиками, у каждого ресурса свой администратор)
2. Географическая глобальная распределенность (пространственно удаленные узлы связаны через сети общего доступа, например Интернет)
3. Согласование международных политик администрирования, управления и контроля (необходимо установить общие стандарты управления и контроля для нескольких стран или организаций)

Сегмент в нашей модели будет состоять из набора узлов. Каждый узел может быть как отдельной рабочей станцией, так и кластером. Важно, что вычислительный узел характеризуется внутренней гомогенностью. Будем называть систему распределения заданий в Грид брокером, так как основной ее функцией является посредничество между отправителем общей задачи и владельцами вычислительных ресурсов. В Грид может быть несколько сервисов-брокеров, и их функционирование может не быть согласованным. Например, один брокер будет глобальным для определенного сегмента, другой - специализированным по какому-то классу задач или группе пользователей, которой разрешено запускать через него задачи. Брокер распределяет задания в условиях неполной информации относительно занятости вычислительных ресурсов.

В составе брокера мы рамссматриваем три различные компоненты: 1) база даннных о выполненных заданиях, 2) анализатор, 3) планировщик.

База данных о выполненных заданиях хранит информацию о приложениях и входных параметтрах, с которыми они когда-либо

запускались в Грид-системе. Ее типичной функцией является определение, запускалось ли уже текущее задание когда-либо, и если да, то каков его уникальный идентификатор. Сложность определения идентичности заданий достаточно высока. На вход приложения, запускаемого в Грид, зачастую передаются многомегабайтные файлы. Каким образом определить, что два запуска одного приложения с различными входными данными можно рассматривать, как запуски одного и того же задания?

Анализатор занимается определением параметров Грид-системы, связанных с производительностью узлов, пропускной способностью каналов данных между узлами и с трудоемкостью каждого задания. Эта статья будет далее посвящена именно задачам и алгоритмам работы анализатора.

Планировщик занимается собственно составлением плана выполнения заданий. Имея оценки производительности, пропускной способности каналов данных и сложности каждого задания, можно воспользоваться одной из известных эвристик для планирования выполнения заданий [3,4]. В в общем случае составление оптимального расписания является NP-полной задачей. Нужно иметь в виду тонкости планирования, возникающие, когда вычислительный узел является параллельной системой. В этом случае нужно посылать на него блоки из заданий, которые будут одновременно выполняться. Важной задачей является определение числа процессоров. Для этого может быть предложена специальная эвристика, опирающаяся на оценки относительной трудоемкости различных заданий, которые будут рассмотрены далее.

В рамках Globus Toolkit самой последней стабильной версии 4.2.1 существует метапланировщик GridWay, как раз предназначенный для распределения заданий локальным планировщикам отдельных вычислительных узлов, имеющий адаптеры к PBS, Sun Grid Engine, Torque [5,6]. GridWay пользуется рядом эвристик, среди которых есть и распределение согласно накопленной статистике и истории выплонения заданий. Однако мы считаем целесообразным выделение модуля построения оценок параметров вычислительной системы на основе истории выполнения заданий в отдельную независимую компоненту, результатами работы которой может пользоваться планировщик в различных эвристиках.

В этой статье мы подробнее рассмотрим алгоритм оценивания

неизвестных параметров вычислительной системы. Как было сказано выше, эта задача распадается на три отдельных проблемы, которые и будут рассмотрены в следующих разделах.

3. Оценивание трудоемкости заданий

Трудоемкостью заданий мы будем называть относительную меру сложности заданий друг по отношению к другу. Для того, чтобы обеспечить корректность задания этой величины, будем считать, что в рамках Грид выделен некоторый эталонный компьютер с номером 0, на котором и сравнивается трудоемкость отдельных заданий, и за меру сложности возьмем просто продолжительность выполнения задания на указанном компьютере. Сложность состоит в том, что не все задания могут быть выполнены на этом компьютере в силу большого их разнообразия. Будем обозначать известное время вычисления задания $t_i(k)$, где i – номер компьютера, k – номер задания. Каждое измерение времени выполнения неизбежно сопряжено с неопределенностью, которая выражается в том, что от наблюдения к наблюдению значение $t_i(k)$ изменяется. Обозначим $t_i(k)^* = Et_i(k)$ — математическое ожидание времени выполнения задания, $\hat{t}_i(k)$ — оценку времени выполнения задания k на компьютере i. Оценка может строиться как исходя из измерений $t_i(k)$, так и, при их отсутствии, исходя из других имеющихся сведений.

Обратимся к выполнению заданий на эталонном компьютере. Для определения трудоемкости заданий енобходимо построить вектор $\hat{t}_0 = (\hat{t}_0(1), ..., \hat{t}_0(N))^T$, где N - число всех возможных заданий в рассматриваемой нами Грид-системе.Рассмотрим вероятностное пространство, образованное минимальной сигма-алгеброй, содержащей все события вида “на узле i было рассчитано задание k”, с некоторыми неизвестными вероятностями. Рассмотрим функцию $F(\hat{t}_0(k), t_0(i), t_j(i), t_j(k)) = |\hat{t}_0(k) - \frac{t_0(i)}{t_j(i)} t_j(k)|^2$, соответствующую событию “на узле j и на узле 0 выполнено задание i, а также на узле j выполнено задание k”. По данным измерений, имеющимся после того, как произошло такое событие, требуется определить оценку времени выполнения задания k на эталонном компьютере 0. Поставим задачу

$$E_{k,i,j}F = E_{k,i,j}|\hat{t}_0(k) - \frac{t_0(i)}{t_j(i)} t_j(k)|^2 \to \min_{\hat{t}_0 = (\hat{t}_0(1), ..., \hat{t}_0(N))^T}, \qquad (1)$$

Таким образом, формулировка в виде минимизации математического ожидания означает, что чем чаще в Грид обрабатывается некоторое задание, тем точнее оценка его трудоемкости. Чем чаще задание обрабатывается на конкретном узле, что может быть вызвано наличием на этом узле уникального программного обеспечения, тем точнее оценка трудоемкости задания соответствует его выполнению на этом узле. Задача (1) является задачей линейной фильтрации сигнала вида $y_n = \varphi_n\theta + v_n$, где по наблюдениям y_n требуется построить оценку вектора θ при известных φ_n и статистической природе помехи v_n. Предположение о статистической природе помехи кажется нам в этом случае вполне разумным. Такая задача решается с помощью фильтра Калмана-Бьюси.

4. Оценивание производительности вычислительных узлов

Грид отличается гетерогенностью, что проявляется в разнообразии вычислительных узлов. В состав сегмента Грид могут входить отдельные рабочие станции, кластеры или суперкомпьютеры. Производительность узлов, таким образом, может отличаться во много раз. Узлы могут быть как параллельными, так и последовательными. Будем считать, что для параллельных узлов в состав планировщика входит эвристика, которая отправляет блоки заданий, которые узел выполняет одновременно, и в то же время путем усреднения времени обработки блока по числу заданий либо более сложным способом можно получить по результатам обработки блока время выполнения каждого задания. Воспользуемся обозначениями предыдущего раздела и будем считать, что $t_i(k)$ это время выполнения задания k на компьютере i. Производительность узла будем оценивать относительно эталонного компьютера, также определенного в предыдущем разделе. Рассмотрим следующую модель времени выполнения задания:

$$t_i(k) = (\mu_{ik} + w_{ik})t_0(k) + c_{ik} + v, \tag{2}$$

где μ_{ik} - коэффициент производительности вычислительного узла, w_{ik} - неопределенность, связанная с особенностями конкретного задания, c_{ik} - постоянное время, необходимое на инициацию и завершение процесса, v - погрешность измерения, связанная со влиянием сторонних факторов. Ясно, что в такой модели $Ew_{ik} = 0$. О

математическом ожидании v нельзя говорить того же, так как в v включается задержка, вызванная системными сервисами и другими приложениями, работающими на вычислительном узле, но не относящимися к переданному брокером узлу заданию. Кроме того, сюда же включается систематическая ошибка, возникающая при определении времени выполнения отдельного задания на параллельном вычислительном узле.

Для поиска коэффициента производительности μ_{ik} можно искать решение уравнения вида (2), считая $x = (\mu_{ik}, c_{ik})^T$ переменной. Фактически, это уравнение задает линейную модель наблюдений вида

$$y_n = \varphi^T \theta_n + v_n,$$

где $\varphi^T = (t_0(k), 1)$, $\theta_n = (\mu_{ik}, c_{ik})^T$. Задача оценивания процесса θ_n, в нашем случае представленного в виде $\theta_n = \theta_{n-1} + (w_{ik}, 0)^T$, решается при условии статистической природы v с помощью фильтра Винера-Колмогорова. Для его использования необходимо также знание спектральных плотностей и ковариаций процессов θ_n и y_n. Недостатком применения фильтра Винера-Колмогорова в этой задаче может быть фактическое отсутствие статистической природы у v. Для того, чтобы справиться с этой сложностью, можно применять метод поиска решения уравнения (2), основанный на идеях рандомизации пробной последовательности, изложенных подробно в [7]. Предложенные в этой работе О. Н. Граничиным алгоритмы оценивания методом стохастической аппроксимации и рандомизированным методом наименьших квадратов одинаково пригодны в описанной нами задаче. Роль рандомизированной последовательности играет в нашем случае последовательность заданий. На практике для успешной работы метода необходимо достаточное разнообразие потока заданий, чего несложно добиться в условиях многопользовательского сегмента Грид. Сложность может возникнуть в обеспечении симметричности распределения пробного возмущения. Также, подход на основе рандомизации требует вмешательства в политику планирования, что выходит за рамки предложенной модели организации сервиса-брокера.

5. Оценивание пропускной способности каналов передачи данных

Задача состоит в построении оценки времени передачи файла произвольного (разумного) объема между двумя узлами A и B на основании известной истории пересылки файлов между этими узлами. Обозначим как $t_1^{AB}(size_1), ..., t_n^{AB}(size_n)$ известные наблюдения, где t_i^{AB} - время пересылки файла объемом $size_i$. Пусть требуется построить функцию $\theta(size, \{t_i^{AB}\})$, оценивающую время пересылки из A в B файла объемом $size$.

Задача осложняется тем фактом, что чаще всего распределенные вычисления организуются не на основе специально для этого проведенных каналов передачи данных, а используют существующие сети, к примеру Интернет. Тогда пропускная способность сети с точки зрения сервиса-брокера может динамически меняться при возникновении нагрузки со стороны других пользователей сети. Эти изменения необходимо отслеживать и корректировать оценки.

Алгоритмы оценивания пропускной способности каналов (bandwidth estimation) делятся на два типа. Первые используют статистические модели передачи данных в сети. Примером может служить известная модель Delphi [8]. Однако, она использует модель с одним возможным критическим участком в канале передачи данных, в то время как в реальности их чаще всего бывает несколько [9]. Второй класс алгоритмов основан на следующей простой эвристике. Пусть происходит постепенное увеличение количества передаваемых данных в канале. Когда наступает предел насыщения канала, дальнейшие пакеты становятся в очередь и скорость передачи резко падает. Определив этот момент, можно с какой-то точностью говорить о пропускной способности канала [10]. К недостаткам обоих методов относится тот факт, что они оба плохо работают при условии сторонней нагрузки. Также, известно, что при пересылке данных между достаточно географически удаленными пунктами отдельные пакеты могут идти по различным маршрутам, число которых заранее не известно [11]. В этой ситуации легко можно представить себе пример, когда алгоритм на основе эвристики будет ошибаться.

TCP является протоколом с обратной связью. Сеанс связи про-

исходит следующим образом: отправитель посылает пакеты с данными, а получатель периодически высылает в ответ информацию о доступном “окне”, то есть свободном объеме буфера получателя, который еще можно занять пакетами отправителя. При неизменном качестве передачи пакетов размер “окна” постоянно растчт, сначала экспоненциально, затем линейно. При ухудшении скорости передачи пакетов размер “окна” уменьшается [12].

Мы будем считать, что есть некоторая установившаяся скорость передачи пакетов в рамках канала. Устройство механизмов определения этой скорости (детали протокола TCP) нас более интересовать не будут. Оно, конечно, имеет свою специфику, которая проявляется при условии наличия одного объемного потока данных, однако в каналах общего назначения со сторонними эффектами проявления таких эффектов минимальны [12]. Будем считать, что пропускная способность канала передачи данных представляется функцией $t_{AB}(size) = t_{AB} \cdot size + c_{AB} + w$, возвращающей время передачи файла при полной занятости канала, где $size$ - размер файла, а w - неопределенность, обусловленная работой адаптационного механизма TCP, а также передачей по каналу посторонних данных.

Рассмотрим следующую модель передачи данных в канале:

$$t_{send} = t_{AB} size + c_{AB} + w, \tag{3}$$

где t_{send} - время передачи данных, t_{AB} - пропускная способность канала, $size$ - размер передаваемого файла, c_{AB} - постоянное время, необходимое на открытие соединения и пр., w - помеха.

Задача оценивания параметров t_{AB}, c_{AB} является также задачей фильтрации. При этом мы можем предположить статистическую природу помехи w и воспользоваться фильтром Калмана-Бьюси. Впрочем, здесь справедливы и изложенные выше соображения о рандомизации пробной последовательности.

С другой стороны, фильтрация Калмана-Бьюси, возможно, не сможет обеспечить состоятельность оценок в реальности, с учетом особенных свойств распределения вероятностей размеров передаваемых файлов $size$. Пусть имеется сигма-алгебра, порожденная элементарными событиями вида “переслан файл объема $size$”. У каждого такого события имеется своя вероятность, зависящая от запус-

каемых задач в системе. Система параллельных вычислений с большим количеством пользователей и простым интерфейсом добавления новых приложений обладает существенным разнообразием возможных вычислительных заданий. Учитывая этот факт, можно предположить, что распределение вероятностей, заданное на такой сигма-алгебре, будет являться распределением Парето [13]. Это распределение характерно для размеров веб-страниц или файлов на жестком диске. Его особенностью является "размытость" принимаемых случайной величиной значений. Зачастую у подобных распределений отсутствуют конечные центральные моменты степени выше чем $1+\epsilon$ для некоторого $\epsilon > 0$. В этом случае, следует рассматривать задачу вида:

$$E_{size}|t_{send} - \hat{t}_{AB}size - \hat{c}_{AB}|^{1+\epsilon} \to \min_{\hat{c}_{AB},\hat{t}_{AB}} . \tag{4}$$

где $\epsilon > 0$. Показатель степени под математическим ожиданием $1+\epsilon$ обусловлен тем, что для распределения Парето может не быть конечного центрального момента степени 2, а значит - аналогичная задача, но с показателем 2, потеряет смысл. В этом случае, опираясь на полученный для алгоритма стохастической оптимизации с пробным рандомизированным возмущением на входе результат [14] о состоятельности оценок для распределений, имеющих конечные центральные моменты степени не выше $1+\epsilon$, $\epsilon > 0$, можно получить результат о применимости рандомизированного алгоритма в случае линейной регрессии для такого типа задач, который будет теоретическим обоснованием использования рандомизированного на входе алгоритма для оценивания пропускной способности каналов передачи данных. Для применения алгоритма с пробным рандомизированным возмущением на входе требуется точно знать значение $Esize$. На практике можно использовать вместо точного значения его состоятельную (или сильносостоятельную) оценку.

6. Заключение

Методы оценивания неизвестных параметров параллельной вычислительной системы, рассмотренные в этой статье, будут использованы при разработке нового сервиса-брокера для отправки заданий в систему распределенных вычислений ПУНК СПбГУ, включающую кластеры математико-механического и физического фа-

культетов, отдельные рабочие станции и, возможно, выход в другие сегменты Грид NorduGrid.

Рассмотренные задачи могут решаться различными методами, часть из которых хорошо известно (фильтры Винера-Колмогорова и Калмана-Бьюси), часть являются довольно новыми (рандомизированные фильтры). Применение последних требует отдельного теоретического обоснования, но позволит ослабить предположения о статистическом характере неопределенности, что является, на взгляд автора, важным.

Список литературы

[1] *Folding@Home*. Client statistics. http://fah-web.stanford.edu/cgi-bin/main.py?qtype=osstats

[2] *Шошмина И. В., Зароченцев А. К., Иванов А. С., Феофилов А. Г.* Использование Grid-технологий для крупномасштабных научных экспериментов. Ч. I. Введение в Grid-технологии с примерами практических занятий на базе ARC NorduGrid. СПб, октябрь 2006.

[3] *Munir E. U., Li J., Shi S., Zou Z., Yang D.* MaxStd: A task scheduling heuristic for heterogeneous computing environment // Information Technology Journal. 2008.

[4] *Xiaoshan H., Sun X. H., von Laszewski G.* QoS guided min-min heuristic for grid task scheduling // Journal of Computer Science and Technology. Vol. 14. No 4. 2003. PP. 442–451.

[5] Документация Globus 4.2.1 *http://www.globus.org/toolkit/docs/4.2/*

[6] *Tannenbaum T., Wright D., Miller K., and Livny M.* Condor - A distributed job scheduler // Beowulf Cluster Computing with Linux. The MIT Press. 2002.

[7] *Granichin O.N.* Linear regression and filtering under nonstandard assumptions (Arbitrary noise) // IEEE Trans. on AC. Vol. 49. oct. 2004. PP. 1830-1835.

[8] *Ribeiro V., Coates M., Riedi R., Sarvotham S., Hendricks B., Baraniuk R.* Multifractal cross-traffic estimation // Proc. of ITC Specialist Seminar on IP Traffic Measurement. Sept. 2000.

[9] *Ribeiro V. J., Riedi R. H., Baraniuk R. G., Navratil J., Cottrell J.* PathChirp: Efficient available bandwidth estimation for network paths // In Proc. of The Conf. on Passive and Active Measurements (PAM). April 2003.

[10] *Jain M., Dovrolis C.* End-to-end available bandwidth: measurement methodology, dynamics, and relation with TCP throughput // Proc. ACM SIGCOMM. 2002.

[11] *Paxson V.* End-to-End Internet packet dynamics // IEEE/ACM Trans. on Networking. Vol. 7. No 3. Jun 1999. PP. 277 - 292.

[12] *Joo Y., Ribeiro V.,Feldmann A.,Gilbert A.C., Willinger W.* TCP/IP traffic dynamics and network performance: A lesson in workload modeling, flow control, and tracedriven simulations // ACM SIGCOMM. Vol. 31. No 2. 2001. PP. 25–37.

[13] *Mitzenmacher M.* Dynamic models for file sizes and double pareto distributions // Internet Mathematics. Vol. 1. No 3. 2004. PP. 305–333.

[14] *Вахитов А. Т., Граничин О. Н., Сысоев С. С.* Точность оценивания рандомизированного алгоритма стохастической оптимизации // Автоматика и телемеханика. 2006. № 4. С. 86–96.