УДК 004.75

М.Б. Ильяшенко (канд.техн.наук, доц.),
Б.А. Артюшенко (канд.техн.наук, ст.преп.)
Запорожский национальный технический университет
matviy.ilyashenko@gmail.com, bogdan.artyushenko@gmail.com

АЛГОРИТМ ОПТИМАЛЬНОГО РЕЗЕРВИРОВАНИЯ РАСПРЕДЕЛЕННЫХ ВЫЧИСЛИТЕЛЬНЫХ РЕСУРСОВ ПО КРИТЕРИЮ ЭФФЕКТИВНОГО ИСПОЛЬЗОВАНИЯ КАНАЛОВ СВЯЗИ

В работе представлен алгоритм оптимального резервирования ресурсов, позволяющий выполнять поиск областей резервирования согласно критериям по числу и максимальной загруженности использованных каналов связи. Приводится детальное описание работы алгоритма и набора предварительных условий, сужающих область поиска переборной части алгоритма.

GRID, граф, алгоритм, оптимизация, критерий

Введение

Алгоритмы управления распределенными вычислительными ресурсами, как правило, используют одну из двух возможных парадигм: централизованное управление ресурсами [1, 2] или контроль ресурсов локальными приложениями [3, 4]. Каждая парадигма имеет свои достоинства и недостатки, в частности, централизованное управление ресурсами позволяет производить глобальную оптимизацию выделения, использования и переназначения ресурсов сети, но этот подход недостаточно масштабируем как в терминах производительности, так и устойчивости сети в целом. С другой стороны, второй подход, основанный на локальных менеджерах, управляющих ресурсами одного или нескольких компьютеров, может приводить к нестабильному выделению ресурсов и не позволяет получать оптимальные решения в масштабах всей сети по такому параметру, как например, средняя загруженность ресурсов сети. Каждый подход имеет свою область применения.

Кроме того, большинство подходов к распределению ресурсов сети не учитывают необходимость совместного выделения ресурсов (co-allocation), необходимого для исполнения параллельных задача в рамках сетей GRID. Наиболее важными видами ресурсов для параллельных задач являются продуктивность процессоров, объем доступной оперативной памяти и пропускная способность или латентность каналов связи для каждого вычислительного узла сети. Алгоритм глобального резервирования распределенных вычислительных ресурсов должен поддерживать все типы ресурсов, которые могут быть зарезервированы и при этом влияют на

производительность программ. В рассмотренном алгоритме базовыми ресурсами, подлежащими резервированию, являются минимальная производительность процессов и минимальная необходимая полоса пропускания сети.

В работе использован графо-аналитический подход к решению задачи оптимального резервирования распределенных вычислительных ресурсов в сетях GRID. Недавний прогресс в развитии алгоритмов точной проверки графов на изоморфность [5-6] позволяет приступить к разработке новых алгоритмов глобальной оптимизации выделения ресурсов, способных работать с GRID сетями реальных размеров. Алгоритм оптимального резервирования распределенных вычислительных ресурсов, представленный в работе, основан на алгоритме нахождения граф-подграф изоморфизма на взвешенных графах [7] с модификациями, направленными на возможность распределения нескольких узлов параллельной задачи на один вычислительный узел сети.

Постановка задачи

Пусть даны два графа G_N и G_T. Пусть граф $G_N = (E_N, V_N, I_N, J_N)$ - представляет собой граф сети, где E_N - множество физических сетевых соединений (линий связи), V_N - множество вычислительных узлов сети (процессоров), I_N - множество весов, приписанных ребрам графа, представляющих собой пропускную способность сетевых соединений, измеренную в Mbps или Gbps и J_N - множество весов, приписанных вычислительным узлам графа сети, представляющих собой производительность процессоров, измеренную в MIPS или MFLOPS. Пусть граф $G_T = (E_T, V_T, I_T, J_T)$ - представляет собой граф параллельной задачи, для которой необходимо зарезервировать часть вычислительных ресурсов сети до начала вычислений, где E_T - множество ребер графа, соответствующих сетевым взаимодействиям между модулями параллельной программы, V_T - множество вершин, соответствующих модулям параллельной программы, I_T - множество весов, приписанных ребрам графа задачи, выражающих потребность в пропускной способности сети для данного сетевого взаимодействия, измеренную в Mbps или Gbps, J_T - множество весов, приписанных вершинная графа, соответствующих потребности модулей в вычислительной производительности процессорных узлов, измеренной в MIPS или MFLOPS.

В терминах определений, введенных выше, задача резервирования распределенных вычислительных ресурсов формализируется следующим образом. Необходимо найти подстановку $\varphi : V_T \to V_N$, такую, что для каждой пары вершин (v_i, v_j), если $(v_i, v_j) \in E_T \Rightarrow (\varphi(v_i), \varphi(v_j)) \in E_N$ и для $(i_i, i_j) \in I_T \le (\varphi(i_i), \varphi(i_j)) \in I_N$, и $j_i \in J_T \le \varphi(j_j) \in J_N$.

Задача оптимизации резервирования ресурсов по критерию

минимального количества задействованных вычислительных узлов сети означает, что среди всех $\varphi: V_T \to V_N$ выбирается та подстановка, при которой $|V_N| \to \min$.

Алгоритм резервирования ресурсов

Структура алгоритма

Как следует из постановки задачи, проблема схожа с задачей нахождения граф-подграф изоморфизма на взвешенных графах, стой лишь разницей, что допускается совмещение нескольких вершин графа задачи с одной и той же вершиной графа сети. Веса, приписанные вершинам и ребрам графа сети, являются ограничениями «сверху», налагаемыми на возможное решение.

Описание алгоритма удобно производить в терминах поиска в пространстве состояний. Каждому состоянию соответствует частичная подстановка $\varphi(s)$, содержащая лишь часть вершин графов, которые уже были совмещены.

Алгоритм состоит из предварительной и основной части.

Предварительная часть алгоритма

Основной задачей предварительной части алгоритма является выполнение всех проверок, которые основаны на данных известных до начала работы алгоритма и не базируются на информации, о уже совмещенных вершинах частичной подстановки $\varphi(s)$. На их основе формируется матрица возможных совмещений. Так же в предварительной части алгоритма производится сортировка вершин графов.

Центральным элементом предварительной части алгоритма является матрица возможных совмещений. Это бинарная таблица $M_{i,j}$, каждая ячейка которой хранит агрегированное значение о возможности, либо не возможности, совмещения вершин $V_{N,i}$ и $V_{T,j}$. Значения матрицы формируются на основании предварительных проверок и могут принимать значения «истина» или «ложь», в зависимости от того перспективно или нет совмещение соответствующих вершин в основной части алгоритма на основании проверок, выполненных в предварительной части алгоритма.

В представленном алгоритме реализовано несколько базовых проверок, формирующих матрицу возможных совмещений $M_{i,j}$. Если результат проверки условия «истина», то значение матрицы, соответствующее проверяемым вершинам $V_{N,i}$ и $V_{T,j}$, устанавливается равным «ложь».

Пусть $|V_X|$ - степень вершины V_X. Тогда условие, основанное на сравнении степеней вершин, выглядит следующим образом:

$$|V_{N,i}| < |V_{T,j}| \Rightarrow M = false$$

Условие, основанное на сравнении весов, приписанных вершинам графов:

$$J_{N,i} < J_{T,j} \Rightarrow M_{i,j} = false$$

Пусть Vin_X - подмножество, содержащее все вершины, связанные с вершиной V_X входящими ребрами, тогда условие, основанное на этом определении:

$$\left|Vin_{N,i}\right| < \left|Vin_{T,j}\right| \Rightarrow M = false$$

Пусть Iin_X - подмножество весов, приписанных ребрам, входящим в вершину V_X. Условие, основанное на значении Iin_X:

$$\sum Iin_{N,i} < \sum Iin_{T,j} \Rightarrow M_{i,j} = false$$

Пусть $Vout_X$ - подмножество, содержащее все вершины, связанные с вершиной V_X исходящими ребрами, тогда условие, основанное на этом определении:

$$\left|Vout_{N,i}\right| < \left|Vout_{T,j}\right| \Rightarrow M = false$$

Пусть $Iout_X$ - подмножество весов, приписанных ребрам, исходящим из вершины V_X. Условие, основанное на значении $Iout_X$:

$$\sum Iout_{N,i} < \sum Iout_{T,j} \Rightarrow M_{i,j} = false$$

Приведенный выше список простейших условий, не требующих сложных вычислений, тем не менее, часто приводит к значительному сокращению области поиска основной, переборной, части алгоритма. Но в дополнение к приведенному списку, предлагается использование условий, основанных на волновом разложении графов, более подробно описанном в работе [8]. Метод волнового разложения графов формирует много структурной информации о графе, которая может быть использована для построения более эффективных условий ограничения области поиска, используемых в предварительной части алгоритма. Наиболее важной частью волнового разложения графов, используемой в данном алгоритме, является подграф окружения вершины. Это подграф, состоящий из вершин, находящихся на расстоянии k ребер от исходной, где k - параметр, определяющий как много соседних вершин будет задействовано при формирования подграфа окружения. Другим важным параметром волнового разложения, использованным в работе, является число вершин, вошедших в каждую волну разложения графа, начиная с заданной вершины.

Пусть $W_{X,Y,k}$ - множество вершин, вошедших в волновое разложение графа X, начиная с вершины Y с максимальным реберным расстоянием до вершин k и пусть $\left|W_{X,Y,k}\right|$ - число вершин в подмножестве $W_{X,Y,k}$. Условие, основанное на этом определении:

$$\left|W_{N,i,k}\right| < \left|W_{T,j,k}\right| \Rightarrow M = false,$$

где $k = 1..\left|W_{T,j}\right|$

Пусть $U_{X,Y,k}$ - множество весов, соответствующих вершинам множества $W_{X,Y,k}$:

$\sum U_{N,i,k} < \sum U_{T,j,k} \Rightarrow M = false$,

где $k = 1..|W_{T,j}|$

Пусть $Q_{X,Y,k}$ - подмножество ребер, которые вошли в волновое разложение графа X, начиная с вершины Y с максимальным расстоянием до вершин в k ребер и пусть $|Q_{X,Y,k}|$ - количество ребер в подмножестве $Q_{X,Y,k}$. Тогда формируется условие:

$|Q_{N,i,k}| < |Q_{T,j,k}| \Rightarrow M = false$,

где $k = 1..|Q_{T,j}|$

Пусть $R_{X,Y,k}$ - множество весов, приписанных ребрам, вошедшим в подмножество $Q_{X,Y,k}$:

$\sum R_{N,i,k} < \sum R_{T,j,k} \Rightarrow M = false$,

где $k = 1..|Q_{T,j}|$

Еще одним важным действием, выполняемым в рамках предварительной части алгоритма, является сортировка вершин графов. Целью является перестановка вершин графов таким образом, чтобы получить более сильное ограничивающее условие, основанное на частичных подстановках, получаемых в процессе работы основной, переборной, части алгоритма. Идея основана на том, чтобы вначале совмещать вершины, имеющие больше внутренних реберных связей, и, как результат, формирующих более сильное ограничивающее условие переборной части. В программе сортируются только вершины графа G_T, описывающего требования параллельной задачи к распределенным ресурсам, в то время, как порядок следования вершин графа сети G_N остается без изменений, т.к. соответствие вершин графа G_N вершинам графа G_T будет устанавливаться уже в переборной части алгоритма.

Пусть $T_{T,i}$ - число ребер, инцидентных вершинам, имеющим индекс меньше, чем i и пусть $P_{T,i} = \sum M_j$ - число возможных совмещений, доступных для вершины $V_{T,i}$ графа G_T. Тогда порядок следования вершин графа G_T определяется следующими выражениями:

$V_{T,i} \leftrightarrow V_{T,j}$, где $T_{T,k} = \min(T_{T,j})$ для $j = (i+1)..|V_T|$

В случае, если $T_{T,i} = T_{T,k}$ применяется другое условие:

$V_{T,i} \leftrightarrow V_{T,j}$, где $P_{T,k} = \min(P_{T,i}, P_{T,j})$

Переборная часть алгоритма

Переборная часть алгоритма объединяет в себе все действия, направленные на поиск полной подстановки φ, являющейся решением задачи. Алгоритм представлен рекурсивной функцией поиска в пространстве состояний, которая на каждом шаге вложенности генерирует новую частичную постановку $\varphi_{i+1}(s)$ из предыдущей частичной подстановки $\varphi_i(s)$,

путем добавления одной вершины в частичную подстановку.

Начальное состояние $\varphi_0(s)=0$. На каждом шаге функция перебирает все вершины графа G_N, которые имеют пометку «истина» в строке матрицы возможных совмещений, соответствующей вершине $V_{T,i}$, используемой на текущем уровне вложенности алгоритма i. Для каждой вершины $V_{N,j}$, алгоритм производит несколько проверок, имеющих целью удостовериться, что все необходимые условия для включения новой вершины в частичную подстановку $\varphi_{i+1}(s)$ выполнены. Далее приведен полный список условий, используемых в переборной части алгоритма с более подробным описанием.

Наиболее важным является условие, основанное на значение соответствующего элемента матрицы возможных совмещений M:

$M_{i,j}=true$

Пусть $T_{N,i}$ - число ребер, связывающих вершину $V_{N,i}$ с как минимум одной вершиной из вошедших в частичную подстановку $\varphi_i(s)$:

$T_{N,i}>T_{N,j}$

Условие, основанное на множестве ребер, входящих в текущую частичную подстановку:

$(v_i,v_k)=(\varphi_i(v_i),\varphi_i(v_k))$,

где $k=1..i$ (1)

Условие, основанное на весах этих ребер:

$(i_i,i_k)\geq(\varphi_i(i_i),\varphi_i(i_k))$,

где $k=1..i$ (2)

Если все условия, описанные выше, истинны, то очередная пара вершин $(V_{T,i},V_{N,j})$ будет добавлена в частичную подстановку $\varphi_i(s)$ и будет сформирована новая частичная подстановка $\varphi_{i+1}(s)$.

Перебор вариантов производится методом поиска в глубину.

Поиск оптимальной области резервирования

Для поиска оптимальной области резервирования используется особенность реализации алгоритма, заключающийся в том, что перебор возможных подстановок $\varphi(s)$ можно продолжать, после нахождения первой полной подстановки и перебрать все полные подстановки, которых может быть несколько. Из всех полных подстановок, которые будут найдены алгоритмом, выбирается та, что соответствует критерию оптимальности.

Оптимальное резервирование ресурсов по критерию минимального числа используемых каналов связи (сетевых соединений) основано на применении алгоритма к реберным графам (G_N^R и G_T^R), полученным из исходного графа сети G_N и исходного графа задачи G_T.

Реберный граф (Line (edge) graph) - для заданного графа G граф L(G), вершинами которого служат ребра графа G и две вершины смежны в L(G) тогда и только тогда, когда соответствующие ребра смежны в G [9].

Фактически, путем преобразования исходных графов к реберным графам, задача совмещения вершин преобразуется к задаче совмещения ребер исходных графов. Веса, приписанные вершинам исходных графов, теперь приписываются соответствующим исходным вершинам ребрам, а веса приписанные ребрам исходных графов, приписываются соответствующим им вершинам реберных графов. В дальнейшем к реберным графам применяется описанный выше алгоритм, модифицированный для выбора из всех возможных подстановок той, что соответствует критерию оптимальности.

Пусть подстановка $\varphi(s)$ ставит в соответствие вершинам реберного графа задачи G_T^R часть вершин реберного графа сети $G_N^R : \varphi(s) : V_T^R \to V_N^{R'} \in V_T^R$.

Оптимальная область резервирования по критерию числа задействованных сетевых соединений определяется следующим условием, налагаемым на полную подстановку:

$\varphi(s) : V_T^R \to V_N^{R'} \in V_N^R$, где $\left| V_N^{R'} \right| \to \min$

Практический смысл оптимизации области резервирования по числу задействованных сетевых соединений заключается в том, что используется минимально возможное число физических линий связи, что позволяет экономить на прокладке дополнительных линий связи или использовать их для совершенно других задач. Так же это упрощает администрирование, позволяя упростить маршрутизацию и давая возможность передавать минимально необходимое, но достаточное число каналов связи в пользование или аренду.

Оптимальная область резервирования по критерию максимальной загруженности каналов связи сети определяется следующим условием:

$\varphi(s) : V_T^R \to V_N^{R'} \in V_N^R$,

при этом $\sum \left(J_N^{R'} - J_N^R \right) \to \min$,

где $J_N^{R'}$ - множество весов, приписанных вершинам реберного графа сети $V_N^{R'}$, задействованным в полной подстановке.

Практический смысл оптимизации по критерию максимальной загруженности каналов связи сети в наиболее полном использовании имеющихся линий передачи данных. Если такая оптимизация производится в масштабах всей сети для составного графа, содержащего все задачи, выполняемые в сети, то, как следствие, получается глобальная оптимизация распределения зарезервированных областей, если же критерий оптимальности используется при добавлении каждой следующей задачи в сеть, то оптимизация зарезервированных областей производится жадным алгоритмом, который так же дает хорошие результаты оптимизации, без необходимости увеличения размерности задачи до масштабов всей сети.

Выводы

В работе представлен переборный алгоритм оптимизации резервирования распределенных вычислительных ресурсов по критериям числа и максимальной загруженности задействованных каналов связи. В алгоритме использованы составные условия, ограничивающие область поиска переборной части алгоритма, в том числе, основанные на волновом разложении графов, значительно сокращающие время поиска оптимального решения. Для оптимизации области резервирования по критериям эффективного использования каналов связи, предложен метод решения задачи оптимизации на реберных графах.

Дальнейшие усилия будут направлены на совершенствование алгоритма для решения задач оптимизации в сверхбольших сетях, посредством разбиения задачи оптимизации на несколько подзадач, меньшей размерности.

Литература

1. Foster I., Roy A., Sander V. A quality of service architecture that combines resource reservation and application adaptation. // Proceedings of the 8th International Workshop on Quality of Service (IWQOS). – Pittsburgh, PA. – June 2000. – pp. 181-188.
2. Sander V. A Metacomputer Architecture Based on Cooperative Resource Man-agement, // Proceedings of High Performance Computing and Networking Europe 1997 (HPCN 1997). – Wien. – April (1997). – pp. 28-30.
3. Berman F., Wolski R., Figueira S., Schopf J., Shao G. Application level scheduling on distributed heterogeneous networks. // Proceedings of Supercomputing 1996. – 1996, 39
4. Wandan Zeng, Guiran Chang, Dengke Zhang, Xiuying Zheng. "G-RSVPM: A Grid Resource Reservation Model" // First International Conference on Semantics, Knowledge and Grid (SKG'05). – Guilin Guanxi, China – 2005. – p. 79.
5. Bunke H., Vento M. Benchmarking of graph matching algorithms. // Proceedings of the 2nd Workshop on Graph-based Representations. – Haindorf. – 1999. – pp. 109–114.
6. Foggia P., Sansone C., Vento M. A performance comparison of five algorithms for graph isomorphism. // Proc. of the 3rd IAPR TC-15 Workshop on Graph-based Representations in Pattern Recognition. – Italy. – 2001. – pp. 188–199.
7. Ильяшенко М.Б. Алгоритм нахождения граф-подграф изоморфизма для взвешенных графов и его применение / М.Б. Ильяшенко // Радиоэлектроника, Информатика, Управление. – 2007. – №1. – С. 62-68.
8. Пинчук В.П. Табличные инварианты на графах и их применение / В.П. Пинчук // Кибернетика и системный анализ. – 2001. – №4. – С.33-45.
9. Лекции по. теории графов / Емеличев В. А., Мельников О. И., Сарванов В. И., Тышкевич Р. И. — М.: Наука, 1990. — 384 с.

Надійшла до редколегії: 25.10.2010 Рецензент: канд.техн.наук, доц. Зеленьова І.Я.

Ільяшенко М.Б., Артюшенко Б. А.
Запорізький національний технічний університет

Алгоритм оптимального резервування розподілених обчислювальних ресурсів за критерієм ефективного використання каналів зв'язку. В роботі запропоновано

алгоритм оптимального резервування обчислювальних ресурсів, що здатен виконувати пошук областей резервування, згідно критеріям оптимальності по числу та максимальному завантаженню задіяних каналів зв'язку. Наводиться детальний опис алгоритму та набору попередніх умов, що звужують область пошуку у переборній частині алгоритму.
GRID, граф, алгоритм, оптимізація, критерій

Il'yashenko M., Artushenko B.
Zaporizhzhya National Technical University

Algorithm for Optimal Distributed Resources Reservation Using Effective Load of Communicational Channels Criteria. This paper presents an algorithm for optimal distributed resources reservation that allows finding reservation areas according to minimal amount of used computational nodes optimization criteria and maximal productivity usage of equipped communicational channels. Paper contains detailed algorithm description, including set of preliminary conditions that reduce computational complexity of enumerating part of the algorithm.
GRID, graph, algorithm, optimization, criterion