МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Нижегородский государственный университет им. Н.И. Лобачевского

Л.Г. Афраймович

УЧЕБНО-МЕТОДИЧЕСКОЕ ПОСОБИЕ ПО КУРСУ «МОДЕЛИ И МЕТОДЫ ЭФФЕКТИВНОГО ИСПОЛЬЗОВАНИЯ РАСПРЕДЕЛЕННЫХ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ» ПРИ ИЗУЧЕНИИ ТЕМЫ «ЗАДАЧИ СТАТИЧЕСКОЙ БАЛАНСИРОВКИ»

Учебно-методическое пособие

Рекомендовано методической комиссией факультета ВМК
для студентов ННГУ, обучающихся по направлению подготовки
230700 «Прикладная информатика»

Нижний Новгород
2012

УДК 519.8

Афраймович Л.Г. Учебно-методическое пособие по курсу «Модели и методы эффективного использования распределенных вычислительных систем» при изучении темы «Задачи статической балансировки»: Учебно-методическое пособие. – Нижний Новгород: Нижегородский госуниверситет, 2011. – 13 с.

Рецензент: д.т.н., профессор **Д.И. Коган**

В методических указаниях рассматриваются математические модели и методы статической балансировки загрузки распределенных вычислительных систем. Описываются задачи, возникающие при балансировке выполнения независимых задач, а также при балансировке выполнения параллельных методов конечных элементов. Пособие предназначено для студентов факультета ВМК направления подготовки «Прикладная информатика», изучающих курс «Модели и методы эффективного использования распределенных вычислительных систем».

УДК 519.8

Содержание

1. Введение

Моделирование сложных физических процессов является трудоемкой в вычислительном смысле задачей. В связи с этим актуальной является выполнение алгоритмов математического моделирования на параллельных вычислительных системах. Эффективность выполнения алгоритмов на параллельных системах определяется «разумным» распределением нагрузки между элементами системы. При этом на итоговую производительность одновременно оказывают влияние распределенная вычислительная нагрузка и коммуникационные затраты между элементами системы.

Алгоритмы, применяемые при физико-математическом моделировании, основаны на разбиение исследуемого пространства на ячейки при помощи заданной сетки и выполнение ряда вычислений по анализу (обработке) рассчитанных данных, связанных с ячейками сетки. Моделирование осуществляется итерационно – каждый шаг происходит моделирование ряда физических показателей в текущий дискретный такт времени. Каждый из физических показателей моделируется при помощи одного из двух классов алгоритмов, основанных на моделях с явными или неявными зависимостями. В случае явных зависимостей алгоритм расчета (в каждой из ячеек) основан на получении результатов расчетов в предыдущий такт времени от смежных ячеек и выполнении вычислений по анализу текущей и смежных ячеек. При неявных зависимостях, используется метод прогонок, расщепленный по направлениям – в каждом из направлений происходит анализ ячейки, затем результат передается следующей ячейке в данном направлении, далее происходит расчет очередной ячейки. И так в обе стороны по каждому из направлений.

Важным фактом является то, что вычислительная нагрузка (при явных зависимостях), связанная с обработкой различных ячеек, может отличаться (например, в неоднородных средах), более того изменяться во времени (например, по причине внутренних перемещений). Таким образом, нагрузка, будучи даже изначально сбалансированной, может со временем стать существенно разбалансированной, что приводит к потере эффективности вычислений.

При решении задачи балансировки в параллельных вычислительных системах следует выделить два подхода – динамическая и статическая балансировка. Динамическая балансировка осуществляется постоянно в каждый очередной такт дискретного времени. Основная трудность при использовании динамической балансировки в уже разработанных комплексах параллельных программных систем связана с большими затратами по доработке существующего кода программ. Поддержка динамического переназначения ячеек между вычислительными элементами параллельной системы требует, во многих случаях, внесения существенных изменений в созданные программные системы. Статическая балансировка осуществляется только в определенные моменты времени – когда вычислительная нагрузка становится существенно разбалансированной. При этом (в отличие от динамического случая) данный подход существенно более просто внедряется в уже существующие программные системы моделирования. Так как здесь в моменты разбалансировки системы останавливается работа текущего алгоритма счета, расчетные данные сбрасываются в специальные хранилища, а затем заново загружаются с учетом балансировки. При этом в применяемых на практике программных комплексах моделирования процедуры сохранения и загрузки текущего состояния часто уже предусмотрены. Таким образом, при интеграции алгоритмов статической балансировки практически не приходится вносить изменения в существующие коды программ систем.

Существует ряд работ посвященных исследованию данной проблемы в случае, когда особенности системы не позволяют иметь общую информацию о состоянии системы (т.е. о загрузке вычислительных элементов). Подобного рода методы основаны на итеративных алгоритмах локального анализа. Обзор результатов в данной области можно найти в работах [9,11,12]. Задачи принятия оптимальных решений в случае, когда возможен сбор требуемой информации о системе, является менее изученным. Вопросы вычислительной сложности некоторых задач данного класса обсуждаются в [8]. Здесь известен ряд ε-приближенных алгоритмов решения задачи о назначении независимых подзадач на параллельные вычислительные элементы [10,13].

В данной работе предлагаются модели и методы эффективной балансировки алгоритмов физико-математического моделирования с явными зависимостями. Предлагаемые походы основаны на идее статической балансировки. В работе рассматриваются два класса задач: задачи балансировки с произвольной декомпозицией и равной трудоемкостью подзадач; задачи балансировки с регулярной декомпозицией. Предлагаются эффективные алгоритмы решения некоторых частных подзадач, основанные на методах распределения ресурсов в иерархических системах транспортного типа [1-7].

2. Балансировка с произвольной декомпозицией

2.1 Поиска оптимального сбалансированного потока

Будем предполагать, что процессоры загружены атомарными (неделимыми) задачами одинаковой трудоемкости. Также будем предполагать однородность вычислительной системы – производительности вычислительных элементов равны. Содержательно задача заключается в оптимальном равномерном перераспределении имеющейся несбалансированной загрузки процессоров.

Пусть задан граф $G=(V,A)$, $A\subseteq V^2$, V – множество параллельных процессоров, $|V|=n$, A – множество коммуникационных связей между процессорами, $|A|=m$. Пусть c_i – начальная загрузка *i*-го процессора (т.е. количество атомарных подзадач назначенных на процессор), $i\in V$. Обозначим через w среднюю загрузку системы, $w=\frac{1}{n}\cdot\sum_{i=1}^{n}c_i$. Пусть x_{ij} – количество подзадач пересылаемых из процессора i на процессор j, $(i,j)\in A$. Тогда проблема поиска сбалансированной загрузки заключается в определении величин x_{ij}, $(i,j)\in A$, называемых сбалансированным потоком, являющихся решением следующей системы линейных неравенств:

$$[w]\le c_i+\sum_{(j,i)\in A}x_{ji}-\sum_{(i,j)\in A}x_{ij}\le[w]+1,\ i\in V; \tag{1}$$

$$x_{ij}\ge 0,\ (i,j)\in A. \tag{2}$$

Задача оптимальной сбалансированной загрузки заключается в поиске сбалансированного потока с минимальными пересылками между процессорами. Данная задача ставится как задача многокритериальной оптимизации поиска набора x_{ij}, $(i,j)\in A$, удовлетворяющего системе ограничений (1),(2) и минимизирующего частные критерии оптимальности:

$$x_{ij}\to\min,\ (i,j)\in A. \tag{3}$$

В качестве схем компромисса при решении данной многокритериальной задачи предлагаются аддитивная свертка:

$$f(x)=\sum_{(i,j)}x_{ij}\to\min, \tag{4}$$

и минимаксная свертка:

$$g(x)=\max_{(i,j)\in A}x_{ij}\to\min. \tag{5}$$

Пусть f^* – оптимальное значение критерия задачи (1),(2),(4), и g^* – оптимальное значение критерия задачи (1),(2),(5). Дополнительно рассмотрим следующие ограничения:

$$\sum_{(i,j)} x_{ij} = f^*, \tag{6}$$

$$\max_{(i,j)\in A} x_{ij} = g^*. \tag{7}$$

Тогда рассмотрим схему лексикографического упорядочивания частных критериев оптимальности (4), (5). Так если критерий (4) имеет больший приоритет, то задача ставится как задача математического программирования с системой ограничений (1),(2),(6) и критерием (5). В случае большего приоритета у критерия (5) задача ставится как задача математического программирования с системой ограничений (1),(2),(7) и критерием (4).

Разработанные методы решения рассматриваемых задач поиска сбалансированного потока основаны на методах распределения ресурсов в иерархических системах (см. [1-7]) и имеют следующую трудоемкость:

Задача	**Вычислительная сложность**
поиск сбалансированного потока, система (1),(2)	$O(mn)$
поиск оптимального сбалансированного потока с аддитивной сверткой, задача (1),(2),(4)	$O(mn^2)$
поиск оптимального сбалансированного потока с минимаксной сверткой, задача (1),(2),(5)	$O(mn \log C)$ (*)
поиск оптимального сбалансированного потока с лексикографическим упорядочиванием (1),(2),(7),(4)	$O(mn \log C + mn^2)$
поиск оптимального сбалансированного потока с лексикографическим упорядочиванием (1),(2),(6),(5)	$O(mn^2 \log C)$

(*) $C = \sum_{i=1}^{n} c_i$.

2.2. Построение расписания при реализации сбалансированного потока

Будем предполагать, что параллельная вычислительная система функционирует в дискретном времени. И пусть известен некоторый сбалансированный поток. Тогда для данного потока требуется построить расписание, определяющее в какие такты времени и как должны быть сделаны пересылки между процессорами, чтобы реализовать данный поток.

Пусть x_{ij}^t – количество подзадач пересылаемых из процессора i в процессор j в такт времени t, $(i,j)\in A$, $t \in N$. Тогда T-расписанием для потока $x_{ij}, (i,j) \in A$, будем называть набор переменных x_{ij}^t, $(i,j) \in A$, $t = \overline{1,T}$, удовлетворяющий следующей системе ограничений:

$$x_{ij} = \sum_{t=1}^{T} x_{ij}^t\ ,\ (i,j) \in A; \tag{8}$$

$$c_i + \sum_{t=1}^{k-1} \sum_{(j,i)\in A} x_{ji}^t \geq \sum_{t=1}^{k} \sum_{(i,j)} x_{ij}^t\ ,\ i \in V\ ,\ t = \overline{1,k}\ . \tag{9}$$

Здесь ограничения (8) означают, что за T тактов планирования будет реализован весь объем потока. Ограничения (9) означают, что в каждый из дискретных тактов времени процессор может переслать подзадач не больше, чем он обладает на данный момент.

Были разработаны эффективные алгоритмы решения следующих задач, связанных с проблемой построения расписания для сбалансированных потоков:

Задача	**Вычислительная сложность**
для заданной начальной загрузки определить существует ли сбалансированный поток, для которого существует *T*-расписания (для фиксированного значения *T*)	$O(mnT^2)$
для заданной начальной загрузки определить сбалансированный поток, для которого существует *T*-расписание с минимально возможным значением *T*	$O(mn^3 \log n)$
на множестве всех сбалансированных потоков, для которых существует *T*-расписание, определить оптимальный поток с аддитивной сверткой (для фиксированного значения *T*)	$O(mn^2T^3)$
на множестве всех оптимальных сбалансированных потоков с аддитивной сверткой определить поток, для которого существует *T*-расписание с минимально возможным значением *T*	$O(mn^5 \log n)$

Разработанные алгоритмы основаны на методах распределения ресурсов в иерархических системах (см. [1-7]).

3. Балансировка с регулярной декомпозицией

3.1 Исходные параметры

Пусть $A = \{a_{ij} \mid i = \overline{1, N^{(1)}}, j = \overline{1, N^{(2)}}\}$ – множество двухмерных прямоугольных регулярных ячеек сетки; $\gamma(a)$ – объем вычислений в ячейки a, $a \in A$; λ – объем передаваемых данных между ячейками сетки. Через $B = \{b_i \mid i = \overline{1, M}\}$ обозначим множество вычислительных элементов параллельной системы. Пусть $g(b)$ – время, требуемое элементу b для выполнения единицы объема вычислений, $b \in B$; $h(b', b'')$ – время, требуемое для передачи единицы объема данных из элемента b' в элемент b'', при этом $h(b', b'') = 0$, если $b' = b''$, $b', b'' \in B$.

3.2 Варьируемые параметры

Пусть $z^{(1)}+1$, $z^{(2)}+1$ – количество строк и столбцов декомпозиции двухмерной сетки; $y_i^{(1)}$ ($y_j^{(2)}$) – номер строки (столбца) сетки, с которого начинается *i*-ый (*j*-ый) столбец (строка) декомпозиции сетки, $i = \overline{1, z^{(1)} + 1}$ ($j = \overline{1, z^{(2)} + 1}$). Обозначим через x_{ij} – вычислительный элемент, который обрабатывает прямоугольную область сетки между *i*-ой и (*i*+1)-ой строками и между *j*-ым и (*j*+1)-ым столбцами, $i = \overline{1, z^{(1)}}$, $j = \overline{1, z^{(2)}}$.

3.3 Исходные параметры

Общая математическая проблема регулярной декомпозиции 2-х мерной сетки описывается следующей системой ограничений:

$$z^{(1)} \cdot z^{(2)} \le M; \tag{10}$$

$$1 = y_1^{(t)} < y_2^{(t)} < \ldots < y_{z^{(t)}+1}^{(t)} = N^{(t)} + 1, \ t = \overline{1,2}; \tag{11}$$

$$x_{ij} \in B, \ i = \overline{1, z^{(1)}}, \ j = \overline{1, z^{(2)}}; \tag{12}$$

$$x_{i_1 j_1} \ne x_{i_2 j_2}, \text{ где } i_1, i_2 = \overline{1, z^{(1)}}, \ j_1, j_2 = \overline{1, z^{(2)}} \text{ и } (i_1, j_1) \ne (i_2, j_2). \tag{13}$$

$$z^{(1)}, z^{(2)} \ge 1, \ z^{(1)}, z^{(2)} \in \mathrm{N}; \tag{14}$$

$$y_k^{(t)} \in \mathrm{N}, \ k = \overline{1, z^{(t)}}, t = \overline{1,2}. \tag{15}$$

Здесь условия (10) представляют собой требование того, что количество вычислительных элементов, требуемых для реализации декомпозиции не может превосходить количество элементов всей параллельной вычислительной системы; условия (11) – номера строк и столбцов должны определять регулярную декомпозицию сетки размера $N^{(1)} \times N^{(2)}$ на $z^{(1)} \times z^{(2)}$ прямоугольных подобластей; условия (12) – каждой из подобластей сетки ставится в соответствие вычислительный элемент системы; условие (13) – вычислительные элементы, соответствующие различным подобластям должны быть различными; условия (14) и (15) – естественные ограничения на переменные.

3.4 Критерии оптимальности

Критерии оптимальности, определяющие условия сбалансированности декомпозиции связаны с минимизацией времени, требуемого для пересылки данных между вычислительными элементами (внешние затраты) и времени требуемого для выполнения обсчета подобластей соответствующих вычислительным элементам (внутренние затраты). Внешние и внутренние затраты определяются через функции $T_{внут}$ и $T_{внеш}$, соответственно, и связаны со следующими критериями оптимальности:

$$T_{внут}(z^{(1)}, z^{(2)}, y, x) = \max_{\substack{i=\overline{1,z^{(1)}},\\ j=\overline{1,z^{(2)}}}} \left(g(x_{ij}) \cdot \sum_{u=y_i^{(1)}}^{y_{i+1}^{(1)}-1} \sum_{v=y_j^{(2)}}^{y_{j+1}^{(2)}-1} \gamma(a_{uv}) \right) \to \min; \qquad (16)$$

$$\begin{aligned} T_{внеш}(z^{(1)}, z^{(2)}, y, x) = \max\{ & \qquad (17)\\ & \max_{i=\overline{1,z^{(1)}}} \max_{j=\overline{1,z^{(2)}-1}} h(x_{ij}, x_{ij+1})(y_{i+1}^{(1)} - y_i^{(1)})\lambda, \\ & \max_{j=\overline{1,z^{(2)}}} \max_{i=\overline{1,z^{(1)}-1}} h(x_{ij}, x_{i+1j})(y_{j+1}^{(2)} - y_j^{(2)})\lambda \} \to \min. \end{aligned}$$

Таким образом, проблема балансировки 2-х мерной сетки при расчете некоторого физического показателя по явной схеме ставится как задача двухкритериальной оптимизации (10-(17).

Замечание 1. Общая проблема балансировки возникает при балансировке методик математического моделирования сложных физических процессов не по одному, как представлено в рассмотренной задаче, а по ряду показателей. Однако и общая задача может быть сведены к задаче балансировки расчета одного показателя.

Замечание 2. Предложенная постановка задачи балансировки для случая 2-х мерной сетки, естественным образом обобщается на случай 3-х мерной сетки. Представленная математическая модель может быть обобщена и на случай нерегулярной декомпозиции.

4. Заключение

В заключение отметим ряд открытых проблем в области статической балансировки.

Среди открытых проблем балансировки с произвольной декомпозицией следует отметить следующие. Задача сбалансированное перераспределение загрузки в случае, когда подзадачи могут иметь разную трудоемкость. Можно показать, что данная задача оказывается в общем случае NP-трудной. Таким образом, интерес здесь представляет поиск частных полиноминально разрешимых подзадач. А также поиск эффективных ε-приближенных алгоритмов решения задачи в общей постановки. Задача построения расписания для сбалансированного потока при непрерывной модели времени.

Задачи балансировки с регулярной декомпозицией являются менее исследованными. К настоящему моменту были построены эвристические алгоритмы решения поставленных задач данного класса, проводятся более детальные исследования данной проблематики. Предметом бедующих исследований также является случай неявных зависимостей.

Литература

1. Афраймович Л.Г. Потоковые алгоритмы исследования совместности иерархических систем распределения ресурсов с ограничениями // Вестник Нижегородского государственного университета. Математическое моделирование и оптимальное управление. 2006. 2(31). – С. 129-138.
2. Афраймович Л.Г., Прилуцкий М.Х. Многоиндексные задачи распределения ресурсов в иерархических системах // Автоматика и телемеханика. 2006. №6. – С. 194-205.
3. Афраймович Л. Г., Прилуцкий М. Х. Многопродуктовые потоки в древовидных сетях // Известия РАН. Теория и системы управления. 2008. №.2. – С. 57-63.
4. Афраймович Л.Г., Прилуцкий М.Х. Поиск потока в несовместных транспортных сетях // Управление большими системами. Выпуск 24. М.: ИПУ РАН. 2009. С.147-168.
5. Прилуцкий М.Х. Многокритериальное распределение однородного ресурса в иерархических системах // Автоматика и телемеханика. 1996. №2. – С. 24-29.
6. Прилуцкий М.Х. Распределение однородного ресурса в иерархических системах древовидной структуры // Труды международной конференции «Идентификация систем и задачи управления SICPRO'2000». – М.: Институт проблем управления им. В.А. Трапезникова РАН. 2000. – С. 2038-2049.
7. Прилуцкий М. Х. Многокритериальные многоиндексные задачи объёмно-календарного планирования // Известия РАН. Теория и системы управления. 2007. № 1. – С. 78-82.
8. Brucker P., Jurisch B., Krämer A. Complexity of scheduling problems with multi-purpose machines // Annals of Operations Research. V. 70. 1997. – P. 57-73.
9. Deikmann R., Frommer A., Monien B. Efficient schemes for nearest neighbor load balancing // Parallel Computing. 25. 1999. – P. 789-812.
10. Gairing M., Lucking T., Mavronicolas M., Monien B. Computing Nash Equilibria for Scheduling on Restricted Parallel Links // Proc. 36th Annual ACM Sympos. Theory Comput. 2004. – P. 613-622.

11. Isanders P. On the Efficiency of Nearest Neighbor Load Balancing for Random Load // Proc. Parcella ’96: Parallel Processing by Cellular Automata and Arrays. 1996. – P. 120-127.
12. Lucking T., Monien B., Rode M. On the Problem of Scheduling Flows on Distributed Networks // Proceedings of the 27th International Symposium on Mathematical Foundations of Computer Science. 2002. – P. 495-505.
13. Zsuzsanna M. On scheduling problems with parallel multi-purpose machines // TR-2005-02. 2005.