УДК 004.75

Кязимов Дж.К.[1]***, Гасанов Х.А.***[2]
Бакинский Государственный Университет, Баку, Азербайджан
[1]dkazimov@mail.ru, [2]xalid.h@gmail.com

НЕОБХОДИМЫЕ И ДОСТАТОЧНЫЕ УСЛОВИЯ СУЩЕСТВОВАНИЯ ОПТИМАЛЬНОГО РАСПИСАНИЯ ДЛЯ ПЛАНИРОВАНИЯ ВЫЧИСЛЕНИЙ В РАСПРЕДЕЛЕННЫХ СИСТЕМАХ

Предлагается алгоритм построения оптимальных расписаний выполнения n-заданий M-идентичными процессорами в предположении, что процесс выполнения каждого задания происходит без прерывания. В рассматриваемом случае найдены необходимые и достаточные условия существования оптимального расписания.

Ключевые слова: *планирование вычислений, функции штрафа, оптимальное расписание, вычислительные ресурсы, прерывание процесса.*

1. Введение

В последнее время, вопросы, связанные с распределением ресурсов и планированием вычислений в распределенной системе, привлекают к себе внимание. Однако с появлением и активным использованием масштабируемой архитектуры многие проблемы приходится переосмысливать и взглянуть на них по-новому.

Масштабируемость наряду с надежностью, соотношением стоимость/ производительность, совместимостью и мобильностью программного обеспечения является одним из важнейших требований к современным вычислительным системам. Она подразумевает возможность пропускной способности коммуникационной подсистемы в зависимости от числа процессорных узлов, участвующих в вычислениях [1].

Изучение механизма управления ресурсами и планированием вычислений необходимо для повышения производительности распределенных систем. К ресурсам относится все, что так или иначе участвует в обработке данных: вычислительные установки, файловые системы, коммуникации, программное обеспечение, хранилища данных. Назначение систем управления – распределение программных приложений по процессорным узлам или компьютерам.

Цели, которые при этом могут преследоваться, – это увеличение реальной производительности, балансировка нагрузки процессоров и т.д. В программном обеспечении соответствующих систем, как правило, можно выделить два компонента – менеджер ресурсов и планировщик [2]. Менеджер отвечает за распределение вычислительных ресурсов, их аутентификацию, создание и миграцию процессов. Планировщик определяет очередность выполнения работ и их назначение для тех или иных ресурсов. Качество планировщика может оцениваться по двум позициям: во-первых, насколько хорошо по производительности, во-вторых, насколько эффективно в смысле накладных расходов он реагирует на запросы ресурсов со стороны потребителей. Иными словами, пользователи заинтересованы в быстром Интернете и вовсе не желают разбираться с побочными эффектами, которые сопровождают управление им.

Методы планирования процессов и прогнозирования времени выполнения программ варьируются в зависимости от приложений и постановки задачи.

Планирование вычислительного процесса в распределенных системах, как правило, выполняется составлением расписания [2].

В работе [3] рассматриваются модели управления ресурсами, а в работе [1] составление оптимального расписания для выполнения заданий с заданным сроком в распределенной системе в случае прерывания процесса выполнения заданий.

В настоящей статье определены необходимые и достаточные условия для составления оптимального расписания для планирования вычислений в распределенных системах в случае непрерывания процесса выполнения заданий. Общая постановка задачи построения расписания состоит в том, чтобы с помощью некоторого множества процессоров и дополнительных ресурсов, таких как память, библиотечные компоненты в виде программ, доступные специализированные процессоры и т.д., выполнить фиксированную систему операций.

2. Постановка задачи

Рассмотрим задачи построения оптимальных расписаний выполнения n-задания M-идентичными параллельными процессорами в предположении, что процесс выполнения каждого задания происходит без прерывания. Задание k поступает в очередь на выполнение в момент времени $d_k \geq 0$ и требует для выполнения $t_k > 0$ единиц времени. Известен момент времени $D_k \geq 0$, к которому необходимо завершить выполнение задания k.

Одной из задач такого рода является задача построения расписания, которому соответствует наименьшее значение суммы времени запаздываний в выполнении заданий относительно заданных сроков. Обозначим время завершения выполнения заданий при расписании S [1] через $\overline{t}_k$. Предполагается, что процесс выполнения каждого задания может быть начат в любой момент времени, т.е. $d_k=0$, $k=\overline{1,n}$, и процесс идет без прерываний до завершения выполнения этого задания.

Распределить задания множества A={1,2,…, n} по процессорам и указать такие последовательности их выполнения каждым процессором необходимо так, чтобы величина

$$F_\Sigma(S) = \sum_{k=1}^{n} \max(0, \overline{t_k} - D_k) \quad (2.1)$$

– была наименьшей. Функция $F_\Sigma(S)$ – называется функцией штрафа [1]. Расписание, которое удовлетворяет всем условиям рассматриваемой задачи, называется оптимальным, если ему соответствует наименьшее значение $F_\Sigma(S)$.

Ограничимся рассмотрением ситуации, когда все заданные сроки одинаковы и равны $D \geq 0$. Все рассматриваемые ниже расписания предполагаются активными.

Пусть задания множества A ={1, 2,…, n} пронумерованы в порядке неубывания значений t_k и $B_L \subseteq A$ – множество заданий, выполняемых L – м процессором, $L=\overline{1,M}$. Поиск оптимального расписания можно ограничить рассмотрением расписаний, при которых каждый процессор выполняет задания в порядке возрастания их номеров. Таким образом, искомое расписание однозначно определяется разбиением множества A на (необязательно непустые) подмножества $A_1, A_2, \ldots, A_M$ попарно не имеющие общих элементов. Если упорядоченной по возрастанию последовательностью элементов множества B_L является последовательность $(i_1, i_2, \ldots, i_{|B_L|})$, то время начала выполнения задания $\boldsymbol{i_j}$ определяется следующим образом:

$$\underline{t}_{i_j} = \sum_{k=1}^{j-1} t_{i_k}$$

а время завершения его выполнения

$$\overline{t}_{i_j} = \underline{t}_{\,i_j} + t_{i_j}$$

В следующем параграфе дается условие оптимальности расписания.

3. Условия оптимальности расписания

Разобьем множество B_L на два подмножества C_L и D_L по следующему признаку: $i_j \in C_L$, если, в противном случае $i_j \in D_L$. Положим

$$\mathrm{C} = \mathrm{U}_{L=\overline{1,\mathrm{M}}} C_{\mathrm{L}} \text{ и } \mathrm{D} = \mathrm{U}_{L=\overline{1,M}} \mathrm{D_L} \quad .$$

Теорема 3.1. Существует оптимальное расписание для выполнения n-задач M-идентичными процессорами, при котором

1. C= {1, 2,…, |C|}
2. Если |C|<n, то

$$\sum_{k \in C_L} t_k \geq D \,,$$

и D_L содержит те и только те элементы D ={|C|+1,…,n}, которые отличаются от |C| + L на величину, кратную M, $\mathrm{L} = \overline{1, M}$.

Доказательство. а) Пусть S^* – оптимальное расписание и при этом расписании L-й процессор выполняет (в порядке возрастания номеров) задания множества B_L^*, $\mathrm{C_L} = \mathrm{C_L^*}$, $\mathrm{D_L} = \mathrm{D_L^*}$, $\mathrm{L} = \overline{1,\mathrm{M}}$, $C=C^*$ и $D=D^*$.

Если $|C^*|$ =n, то справедливость утверждения теоремы очевидна. Пусть $|C^*|$<n. Если предположить, что при некотором 1 ≤ L ≤ M значение

$$\sum_{k \in C_L^*} t_k < D \,,$$

то приходим к заключению, что расписание S^* не является оптимальным, поскольку одно из требований множества D^* можно было бы выполнить L-м процессором, начиная с момента времени:

$$\sum_{k \in C_L^*} t_k < D$$

Следовательно,

$$\sum_{k \in C_L} t_k \geq D \text{ для всех } \mathrm{L} = \overline{1,\mathrm{M}}.$$

б) Выберем наибольший элемент $j \in C^*$ и наименьший элемент $i \in D^*$. Предположим, что i<j. Если t_i=t_j то задания эквивалентны и их можно поменять местами. Пусть t_i<t_j. Поскольку каждый процессор выполняет задания в порядке возрастания их номеров, то задания i и j выполняются разными процессорами. Пусть, для определенности, задание i выполняется процессором L, а задание j – процессором H. Очевидно, можно полагать, что $\underline{t}_j < D$, $\overline{t}_j \geq D$, $\underline{t}_i \geq D$ и не существует $k \in D_L$ с $\underline{t}_k < \underline{t}_i$. Рассмотрим новое расписание S_1, отличающееся от расписания S^* тем, что задание i выполняется процессором H непосредственно перед заданием j. Суммарный штраф при этом изменяется на величину

$$t_i \times (|D_H^*| - |D_L^*| + 2) + \max\left(\underline{t}_j + t_i - D, 0\right) - (\overline{t}_i - D).$$

Если $|D_L^*| \geq |D_H^*| + 2$, то это величина неположительная. Если $|D_L| = |D^*_н| + 1$, то это величина также неположительная при $\underline{t}_j + t_i \leq \mathrm{D}$.

Если не имеет места ни одна из указанных ситуаций, то рассмотрим расписание S_2, отличающееся от исходного оптимального расписания S^* тем, что на процессоре L вместо задания ***i*** выполняется задание j, а на процессоре H вместо задания j выполняется задание ***i***. Суммарный штраф при этом изменится на величину $\left(t_j - t_i\right) \times |D^*_L| - |D^*_H| + \max(\underline{t}_j + t_i - D, 0) - (\bar{t}_j - D)$. Если $|D^*_L| \leq |D^*_н|$ то эта величина неположительная. Она неположительная и в случае, когда $|D^*_L| = |D^*_н|+1$ при условии, что $\underline{t}_j + t_i > D$

Таким образом, в любом случае существует расписание, которому соответствует суммарный штраф, не превосходящий по величине суммарный штраф, соответствующий оптимальному расписанию S^*. Множество C, определяемое этим расписанием, отличается от C^* только тем, что оно содержит элемент ***i*** и, возможно, не содержит элемент j.

Повторяя аналогичные рассуждения конечное число раз, приходим к заключению, что существует оптимальное расписание, удовлетворяющее условию 1) теоремы. Если |C|<n, то в этом расписании

$$\sum_{k\in C_L} t_k \geq D \text{ для всех } L = \overline{1, M}.$$

Не нарушая общности, можно полагать, что

$$D \leq \sum_{k\in C_L} t_k \leq \cdots \leq \sum_{k\in C_M} t_k.$$

Таким образом, существует оптимальное расписание, при котором задание $|C^*|+1$ выполняется первым процессором, задание $|C^*|+2$ – вторым процессором и т.д., задание $|C^*|+M$ – M-м процессором, задание $|C^*|+M+1$ – первым процессором и т.д., пока не будет исчерпано множество $D^*=\{|C^*|+1,\ldots, n\}$. Предполагается, естественно, что $|C^*|<n$. *Теорема доказана***.**

Обозначим через S(p) класс расписаний, удовлетворяющих условию теоремы 3.1 и дополнительному условию |C|=p, p – натуральное число. Оптимальное расписание S^*, очевидно, принадлежит хотя бы одному из классов S(p) при некотором $p=p^*$.

Число различных непустых классов S(p) не превышает M. Действительно, пусть $\underline{p}$ наименьшее значение p, при котором $S(P) \neq \emptyset$. Если $\underline{p} = n$, то S(p) – единственный непустой класс. Если $\underline{p} < n$, то по определению класса $S(\underline{p})$ значения

$$Z_L = \sum_{k\in C_L} t_k - D \geq 0, \qquad L = \overline{1, M}$$

для любого расписания из этого класса и. следовательно,

$$\sum_{L=1}^{M} Z_L + \sum_{i=1}^{n} t_{|C|+i} < \sum_{i=1}^{M} t_{|C|+i}$$

только при h≤M-1. Обозначим через $\bar{p}$ наибольшее значение p, при котором $S(p) \neq \emptyset$. Таким образом, $\bar{p} - p \leq M - 1$. Необходимо отметить, что при некоторых $\underline{p} < p < \bar{p}$. Может оказаться $S(p) =\emptyset$.

Найдем достаточные условия, при которых расписанию $S\in S(p)$ соответствует наименьший суммарный штраф среди всех расписаний, принадлежащих классу S(p),

$\underline{p} < p < \overline{p}$. Обозначим через q(p) число, равное остатку от деления (n-p) на M при n-p>0 и равное M в противном случае.

Теорема 3.2. Расписание S∈S(p) является оптимальным в S(p), если при этом расписании величина

$$Z(p) = \sum_{i=1}^{q(p)} \max(0, Z_i)$$

достигает наименьшего значения.

Доказательство. Если p=n, то *Z(p)=F∑(S)* Пусть p<n и, следовательно, $Z_L \geq 0$ для всех $L = \overline{1, M}$. Не нарушая общности, будем предполагать, что $Z_1 \leq Z_2 \leq \ldots \leq Z_M$.

Суммарный штраф $F_\Sigma(S)$, соответствующий любому расписанию S∈S(p), можно представить в виде

$$F_\Sigma(S) = \sum_{L=1}^{M} |D_L| Z_L + \overline{F_\Sigma},$$

где величина $\overline{F_\Sigma}$ не зависит от S∈S(p).

По условию $|D_1| = |D_2| = \ldots |D_{q(p)}| = |D_{q(p)+1}| + 1 \ldots = |D_M| + 1$ и

$$\sum_{L=1}^{m} Z_L = const$$

для всех S∈S(p). Следовательно, $F_\Sigma(S)$ достигает наименьшего значения, если величина $\sum_{L=1}^{q(p)} Z_L$ минимальна, что и требовалось доказать.

Таким образом, при построении оптимального расписания S^* можно воспользоваться следующей вычислительной схемой. Сначала необходимо определить значение $\underline{p}$. Затем, используя теоремы 3.1, 3.2, построить оптимальное в S(p) расписание S_p для $p=\underline{p}, \underline{p}+1, \ldots, \overline{p}$.

Среди полученных расписаний, число которых не превосходит M, выбрать то, которому соответствует наименьший суммарный штраф. Это расписание является искомым.

4. Выводы

Получены необходимые и достаточные условия существования оптимального расписания для планирования вычислений в распределенных системах в случае непрерывания процесса выполнения заданий. При таких условиях дается алгоритм построения оптимального расписания.

Литература

1. Кязимов Дж.К., Гасанов Х.А. Оптимальные расписания для выполнения заданий с заданным сроком в распределенных системах. Известия НАНА, том XXXI, №3, 2011, стр.3–8.
2. Топорков В.В. Модели распределенных вычислений. М.: Физматлит, 2004, стр.320.
3. Топорков В.В. Опорные планы согласованного выделения ресурсов при организации распределенных вычислений на масштабируемых системах. Программирование, №3, 2008, стр. 50–64.
4. Теория расписаний и вычислительных машин / Под ред. Э.Г.Коффмона – М.: Наука, 1984. стр.334.

UOT 004.75

Kazımov Cavanşir K.[1], Həsənov Xalid A.[2]
Bakı Dövlət Universiteti,Bakı, Azərbaycan
[1]dkazimov@mail.ru, [2]xalid.h@gmail.com

Paylanmış sistemlərdə hesablanmanın planlaşdırılması üçün optimal cədvəlin varlığının zəruri və kafi şərtləri

n məsələnin M –eyni tipli prosessorda yerinə yetirilməsi üçün optimal cədvəlin qurulması alqoritmi təklif edilir və nəzərdə tutulur ki, hər bir məsələnin yerinə yetirilməsi kəsilməz baş verir. Qeyd edilən hal üçün optimal cədvəlin varlığının zəruri və kafi şərtləri təklif edilib.

Açar sözlər: *hesablanmanın planlanması, cərimə funksiyası, optimal cədvəl, hesablama resursları, prosesin kəsilməsi.*

Javanshir K. Kazimov[1], Khalid A. Hasanov [2]
Baku State University, Baku, Azerbaijan
[1]dkazimov@mail.ru, [2]xalid.h@gmail.com

Necessary and sufficient conditions for existence of the optimum schedule for planning computing in distributed systems

An algorithm for construction of optimum schedules of execution of n tasks in н-identical processors have been offered in the assumption that execution process of each task occurs without interruption. Necessary and sufficient conditions for existence of the optimum schedule in this case have been proposed.

Key words: *computing planning, penalty functions, an optimum schedule, computing resources, interruption of the process.*