Курс лекций

ПАРАЛЛЕЛЬНЫЕ АЛГОРИТМЫ И МОДЕЛИ

Лекция 11

ПОСТРОЕНИЕ ОПТИМАЛЬНОЙ СИСТЕМЫ ОДИНАКОВО РАСПРЕДЕЛЕННЫХ ПРОЦЕССОВ

В МНОГОПРОЦЕССОРНЫХ ВЫЧИСЛИТЕЛЬНЫХ КОМПЛЕКСАХ

Введение (1)

Одним из важнейших требований к современным вычислительным многопроцессорным системам (МС), ВК, базам данных, маршрутизаторам и т.д. является масштабируемость (scalability). Она подразумевает способность системы увеличивать свою производительность при изменении аппаратных и программных ресурсов. В настоящее время вопросы масштабирования находятся в поле зрения как разработчиков МС, так и распределенной среды метакомпьютинга.

В данных вычислительных системах использование общего программного ресурса затрудняется из-за автономности процессорных узлов и отсутствием единой политики администрирования. Общим свойством, обеспечивающим возможность повышения производительности масштабируемых вычислительных систем, является распределенность процессов вычислений и данных с использованием принципов структурирования и конвейеризации.

Введение (2)

Все это заставляет задуматься над разработкой методов организации вычислений и распределения ресурсов, созданием эффективного аппаратного и программного обеспечения, однозначностью результатов выполнения программ, эффективным планированием вычислительных процессов [3]. В связи с этим особую актуальность приобретают задачи эффективного управления множеством процессов, имеющих доступ к общим ресурсам, в том числе и к программным.

Предложим алгоритм построения оптимальной компоновки распределенных процессов, конкурирующих за использование линейно структурированного программного ресурса, требующий не более $O(n^3)$ элементарных операций, где n – число вычислительных процессов исходной одинаково распределенной системы.

1. Основные понятия и постановка задачи (3)

Цикл является одной из функциональных конструкций как в теории алгоритмов, так и в программировании и всегда вызывал огромный интерес у исследователей, занимающихся проблемами создания высокоэффективных алгоритмов и соответствующих программных реализаций.

Интерес к циклическим конструкциям не ослабел и с появлением вычислительных систем, поддерживающих параллельную обработку. При этом появилось целое направление исследований, связанных с эффективным отображением циклических конструкций на различного рода параллельные вычислительные системы.

1. Основные понятия и постановка задачи (3)

Процесс будем рассматривать как последовательность блоков (команд, процедур) $Q_1, Q_2, ..., Q_s$, для выполнения которых используется множество процессоров (процессорных узлов, обрабатывающих устройств, интеллектуальных клиентов). При этом процесс называется *распределенным*, если все блоки или часть из них обрабатываются разными процессорами. Для ускорения выполнения процессы могут обрабатываться параллельно, взаимодействуя путем обмена информацией. Такие процессы называются *кооперативными* или *взаимодействующими* процессами.

Понятие *ресурса* используется для обозначения любых объектов вычислительной системы, которые могут быть использованы процессами для своего выполнения. *Реентерабельные (многократно используемые)* ресурсы характеризуются возможностью одновременного использования несколькими вычислительными процессами. Для параллельных систем характерной является ситуация, когда одну и ту же последовательность блоков или ее часть процессорам необходимо выполнять многократно. Такую последовательность будем называть *программным ресурсом*, а множество соответствующих процессов – *конкурирующими*.

1. Основные понятия и постановка задачи (4)

Математическая модель системы распределенной обработки конкурирующих процессов включает в себя:

s, $s \geq 2$ – число блоков линейно структурированный программного ресурса $PR = (Q_1, Q_2, \ldots, Q_s)$;

n, $n \geq 2$ – число распределенных относительно PR конкурирующих процессов;

p, $p \geq 2$ – число процессоров многопроцессорной системы;

матрицу $T_p = [t_{ij}]$ времен выполнения j–х блоков i–ми конкурирующими процессами $i = \overline{1,n}$, $j = \overline{1,s}$;

ε – время, характеризующее дополнительные системные расходы по организации структурирования и параллельного использования блоков программного ресурса PR.

1. Основные понятия и постановка задачи (5)

Определение. Распределенная система n взаимодействующих конкурирующих процессов называется **неоднородной**, если времена выполнения блоков PR зависят от объемов обрабатываемых данных и/или их структуры, т.е. разные для разных процессов.

Определение. Система взаимодействующих конкурирующих процессов называется **одинаково распределенной**, если времена t_{ij} выполнения блоков Q_j, $j=\overline{1,s}$, программного ресурса PR каждым из i–х процессов совпадают и равны t_i для всех $i=\overline{1,n}$, т.е. справедлива цепочка равенств $t_{i1}=t_{i2}=\ldots=t_{is}=t_i$ для всех $i=\overline{1,n}$.

1. Основные понятия и постановка задачи (6)

Будем считать, что взаимодействие процессов, процессоров и блоков линейно структурированного программного ресурса подчинено следующим условиям:

1) ни один из блоков PR не может обрабатываться одновременно более чем одним процессором;

2) ни один из процессоров не может обрабатывать одновременно более одного блока;

3) обработка каждого блока осуществляется без прерываний;

4) распределение блоков программного ресурса по процессорам МС для каждого из процессов осуществляется циклически по правилу: блок с номером $j = kp + i$, $j = \overline{1,s}$, $i = \overline{1,p}$, $k \geq 0$, распределяется на процессор с номером i. Кроме того, введем дополнительные условия, которые определяют режимы взаимодействия процессов, процессоров и блоков PR:

5) отсутствуют простои процессоров при условии готовности блоков, а также невыполнение блоков при наличии процессоров;

6) для каждого из n процессов момент завершения выполнения j–го блока на i–м процессоре совпадает с моментом начала выполнения следующего $(j+1)$–го блока на $(i+1)$–м процессоре, $i = \overline{1, p-1}$, $j = \overline{1, s-1}$; 7) для каждого из блоков структурированного программного ресурса момент завершения его выполнения l–м процессом совпадает с моментом начала его выполнения $(l+1)$–м процессом на том же процессоре, $l = \overline{1, n-1}$.

1. Основные понятия и постановка задачи (7)

Условия 1–5 определяют **асинхронный режим** взаимодействия процессоров, процессов и блоков, который предполагает отсутствие простоев процессоров МС при условии готовности блоков, а также невыполнение блоков при наличии процессоров.

Если к условиям 1–4 добавить условие 6, то получим **первый синхронный режим**, обеспечивающий непрерывное выполнение блоков программного ресурса внутри каждого из вычислительных процессов.

Второй синхронный режим, определяемый условиями 1–4, 7, обеспечивает непрерывное выполнение каждого блока всеми процессами.

1. Основные понятия и постановка задачи (8)

Ранее было доказано, что для одинаково распределенных систем конкурирующих процессов минимальное общее время для всех трех базовых режимов в случае **неограниченного параллелизма** $(s \le p)$, и в случае **ограниченного параллелизма** $(s > p)$, но если $T_\varepsilon^n \le pt_{\max}^\varepsilon$, определяется по формуле:

$$T(p,n,s,\varepsilon) = T_\varepsilon^n + (s-1)t_{\max}^\varepsilon, \qquad (1)$$

а в остальных случаях общее время выполнения n одинаково распределенных процессов, использующих структурированный на s блоков программный ресурс PR, в вычислительной среде с p процессорами при асинхронном режиме и в режиме непрерывного выполнения каждого блока всеми процессами, составляет величину:

$$T(p,n,s,\varepsilon) = \begin{cases} kT_\varepsilon^n + (p-1)t_{\max}^\varepsilon, & при\ \ s = kp,\ \ k > 1,\ \ T_\varepsilon^n > pt_{\max}^\varepsilon, \\ (k+1)T_\varepsilon^n + (r-1)t_{\max}^\varepsilon, & при\ \ s = kp + r,\ \ k \ge 1,\ 1 \le r < p,\ \ T_\varepsilon^n > pt_{\max}^\varepsilon, \end{cases} \qquad (2)$$

где $T_\varepsilon^n = \sum_{i=1}^{n} t_i^\varepsilon$ – суммарное время выполнения каждого из блоков Q_j всеми n процессами с учетом накладных расходов ε, $t_{\max}^\varepsilon = \max\limits_{1 \le i \le n} t_i^\varepsilon$, $t_i^\varepsilon = t_i + \varepsilon$, $i = \overline{1,n}$.

1. Основные понятия и постановка задачи (9)

С понятием множества одинаково распределенных конкурирующих процессов свяжем понятие линейной упаковки.

Определение. Пусть $M = \{m_1, m_2, \ldots, m_n\}$ – конечное упорядоченное множество предметов. Разбиение множества M на l непересекающихся подмножеств $M_1, M_2, \ldots, M_l$ такое, что каждое подмножество есть объединение последовательных элементов множества M, будем называть **линейной упаковкой** множества M ранга l.

Так как времена выполнения блоков программного ресурса каждым из процессов совпадают $t_{i1} = t_{i2} = \ldots = t_{is} = t_i$, $i = \overline{1,n}$, то элементами множества M будем рассматривать последовательность, например, первых блоков $(Q_{11}, Q_{21}, \ldots, Q_{n1})$ структурированного программного ресурса одинаково распределенных процессов, которую обозначим $(q_1, q_2, \ldots, q_n)$. В этом случае линейная упаковка множества M получается объединением блоков q_i, $i = \overline{1,n}$, последовательных процессов в один программный блок. Линейную упаковку блоков q_i, $i = \overline{1,n}$, которая приведет к уменьшению количества процессов в МС, будем называть **линейной компоновкой** и обозначать LC_l.

1. Основные понятия и постановка задачи (10)

Обозначим через K множество всевозможных компоновок блоков системы одинаково распределенных процессов, конкурирующих за использование программного ресурса PR, а через K_l множество компоновок ранга l, $l=\overline{1,n}$. Отметим, что компоновкой ранга n является исходная одинаково распределенная система $LC_n=(q_1,\ q_2,\ \ldots,\ q_n)$, а ранга 1 – компоновка блоков в один программный блок $LC_1=(q_1\cup q_2\cup\ldots\cup q_n)$. Нетрудно подсчитать, что $|K|=2^{n-1}$, $|K_l|=C_{n-1}^{l-1}=\dfrac{(n-1)!}{(l-1)!(n-l)!}$.

Пусть $LC_l=(q_1',\ q_2',\ \ldots,\ q_l')$ – линейная компоновка блоков.

Обозначим:

? $t(q_i')=\sum\limits_{q\in q_i'}t(q)$ – время выполнения i–го элемента компоновки LC_l, $i=\overline{1,l}$;

? $t(LC_l)=(t(q_1'),\ t(q_2'),\ \ldots,\ t(q_l'))$ – последовательность времен выполнения блоков q_i', $i=\overline{1,l}$;

? $t_{\max}(LC_l)=\max\limits_{1\le i\le l}\{t(q_i')\}$ – время выполнения максимального блока компоновки LC_l;

? $t_{\min}=\min\{t_{\max}(LC_l)\ |\ LC_l\in K_l\}$.

1. Основные понятия и постановка задачи (11)

Задача оптимальной компоновки блоков $(q_1, q_2, ..., q_n)$ множества одинаково распределенных конкурирующих процессов, состоит в том, чтобы при заданных $p \geq 2$, $n \geq 2$, $s \geq 2$, $\varepsilon > 0$, найти такую линейную компоновку LC_l исходной одинаково распределенной системы, при которой достигается минимум функционалов (1) и (2).

Такую компоновку будем называть **оптимальной**.

2. Свойства оптимальных компоновок и вспомогательные результаты (12)

Для решения поставленной задачи потребуются следующие результаты.

Теорема 1. Если LC_l – оптимальная линейная компоновка одинаково распределенной системы, то компоновка LC_l', такая, что $t_{\max}(LC_l') = t_{\min}$, также является оптимальной.

Доказательство. Из определения $t_{\min}$ следует, что

$$t_{\max}(LC_l) = t_{\min}. \tag{3}$$

Покажем, что при выполнении условий теоремы компоновка LC_l' оптимальная, т.е. при заданных ε, p, s

$$T(p, LC_l, s, \varepsilon) = T(p, LC_l', s, \varepsilon). \tag{4}$$

При $s = p$ для оптимальной компоновки LC_l $t_{\max}(LC_l) = t_{\min}$. Следовательно, теорема выполняется.

2. Свойства оптимальных компоновок и вспомогательные результаты (13)

Пусть $s > p$, $s = kp + r$, $1 \le r \le p$. Рассмотрим возможные случаи:

1) $T_\varepsilon^n \le p(t_{\min} + \varepsilon)$, $T_\varepsilon^n \le p(t_{\max}(LC_l) + \varepsilon)$, тогда из (1) и условия оптимальности LC_l следует:

$$T(p, LC_l', s, \varepsilon) - T(p, LC_l, s, \varepsilon) = T_\varepsilon^n + (s-1)(t_{\min} + \varepsilon) - T_\varepsilon^n - (s-1)(t_{\max}(LC_l) + \varepsilon) =$$
$$= (s-1)(t_{\min} - t_{\max}(LC_l)) \ge 0 .$$

Так как $s \ge 2$, то в силу (3) следует, что $t_{\max}(LC_l) = t_{\min}$ и (4) выполняется.

2) $T_\tau^s > p(t_{\min} + \varepsilon)$, $T_\tau^s > p(t_{\max}(LC_l) + \varepsilon)$, тогда из (2)

$$T(p, LC_l', s, \varepsilon) - T(p, LC_l, s, \varepsilon) = (k+1)T_\varepsilon^n + (r-1)(t_{\min} + \varepsilon) - (k+1)T_\varepsilon^n - (r-1)(t_{\max}(LC_l)$$

$$= (r-1)(t_{\min} - t_{\max}(LC_l)) \ge 0 .$$

Отсюда в силу (3) следует (4).

2. Свойства оптимальных компоновок и вспомогательные результаты (14)

3) $T_\varepsilon^n > p(t_{\min} + \varepsilon)$, $T_\varepsilon^n \le p(t_{\max}(LC_l) + \varepsilon)$, тогда

$$T(p, LC_l^{'}, s, \varepsilon) - T(p, LC_l, s, \varepsilon) = (k+1)T_\varepsilon^n + (r-1)(t_{\min} + \varepsilon) - T_\varepsilon^n - (s-1)(t_{\max}(LC_l) + \varepsilon) =$$

$$= (k+1)T_\varepsilon^n + (r-1)(t_{\min} + \varepsilon) - (kp + r - 1)(t_{\max}(LC_l) + \varepsilon) =$$

$$= kT_\varepsilon^n - kpt_{\max}(LC_l) + (r-1)(t_{\min} + \varepsilon) - (r-1)(t_{\max}(LC_l) + \varepsilon) =$$

$$= k(T_\tau^s - pt_{\max}(LC_l)) + (r-1)(t_{\min} - t_{\max}(LC_l)) \ge 0,$$

что возможно лишь при $r = 1$ и $T_\varepsilon^n = p(t_{\max}(LC_l) + \varepsilon)$. Отсюда следует справедливость (4).

Теорема доказана.

2. Свойства оптимальных компоновок и вспомогательные результаты (15)

Теорема 2. Если для компоновок LC_l и LC_{l-1}, $l>2$, $t_{\max}(LC_l)=t_{\max}(LC_{l-1})$, то

$$T(p,LC_l,s,\varepsilon)>T(p,LC_{l-1},s,\varepsilon).$$

Из теоремы 1 следует, что если для каждого ранга $l=2,\ldots,n$, можно эффективно строить линейную компоновку LC_l блоков систем одинаково распределенных процессов с наименьшим максимальным элементом среди компоновок этого ранга $(t_{\max}(LC_l)=t_{\min})$, то «эффективно» будет решена исходная задача, поскольку в этом случае оптимальную компоновку необходимо будет выбирать из $(n-1)$ компоновки.

Очевидно также, что наименьший максимальный элемент среди компоновок ранга l с убыванием l не убывает, т.е.

$$t_{\min}(LC_{l_1})\geq t_{\min}(LC_{l_2}),\ 1<l_1<l_2\leq n, \qquad (5)$$

что позволяет при решении задачи оптимальной компоновки исключить из рассмотрения компоновку в один программный блок.

С практической точки зрения является естественным предположение

$$\varepsilon\leq t_i,\ i=\overline{1,n}, \qquad (6)$$

что позволяет при решении задачи оптимальной компоновки исключить из рассмотрения компоновку в один программный блок.

2. Свойства оптимальных компоновок и вспомогательные результаты (16)

Наряду с исходной задачей рассмотрим следующую оптимизационную задачу «линейной упаковки в контейнеры».

Для заданных предметов конечного упорядоченного множества $M = \{m_1,\ m_2,\ \dots,\ m_n\}$ и соответствующей последовательности их размеров $v(m_1),\ v(m_2),\ \dots,\ v(m_n)$, $v(m_i) > 0$, $i = \overline{1,n}$, числа $B > 0$ – вместимости контейнера, $B \geq \max\limits_{1 \leq i \leq n}\{v(m_i)\}$, требуется найти такую линейную упаковку множества M, чтобы размер каждого элемента упаковки $v(M_l)$ не превосходил B и l было наименьшим.

В общем случае, т.е. когда отсутствует условие линейности упаковки, эта задача является NP–трудной в сильном смысле, поскольку при $v(m_i) \in (0,1)$, $i = \overline{1,n}$, $B = 1$, дает классическую оптимизационную задачу упаковки в контейнеры. Условие линейности упаковки, связанное с задачей оптимальной компоновки блоков одинаково распределенных систем, существенно упрощает ее решение.

2. Свойства оптимальных компоновок и вспомогательные результаты (17)

Задача линейной упаковки в контейнеры эффективно решается с помощью следующего LF–алгоритма (last–fit).

1) Первый предмет m_1 загружается в первый контейнер, а остальные предметы – в порядке возрастания их номеров.

2) Предмет m_i, $i = \overline{2,n}$, загружается в последний контейнер из числа частично упакованных, если сумма помещенных в него предметов не превосходит $B - m_i$, в противном случае он загружается в следующий пустой контейнер.

Оптимальность линейной упаковки, которую строит LF–алгоритм, легко доказывается методом от противного. LF–алгоритм требует не более $3n$ элементарных операций и является составной частью алгоритма решения исходной задачи оптимальной компоновки.

3. Алгоритм построения оптимальной компоновки (18)

Пусть $P_n = (t_1,\ t_2,\ \ldots,\ t_n)$ – последовательность времен выполнения каждого из блоков q_i, $i = \overline{1,n}$, всеми n процессами, $n \geq 3$, $p \geq 2$ – число процессоров, ε – время, характеризующее дополнительные системные расходы, $\varepsilon \leq t_i$, $i = \overline{1,n}$.

Алгоритм построения оптимальной линейной компоновки блоков состоит из следующих этапов.

1) Строим массив из $\dfrac{n(n+1)}{2} - 1$ чисел x_{ij}, $i = \overline{2,n}$, $j = \overline{1,i}$, по правилу:

$$\begin{gathered}
x_{nj} = t_j,\ \ j = \overline{1,n}, \\
x_{n-1,j} = x_{nj} + t_{j+1},\ \ j = \overline{1,n-1}, \\
\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots, \\
x_{n-k,j} = x_{n-k+1,j} + t_{j+k},\ \ j = \overline{1,n-k}, \\
\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots, \\
x_{2j} = x_{3j} + t_{j+n-2},\ \ j = 1,2.
\end{gathered} \tag{7}$$

Здесь числа $x_{ij} = t_j + t_{j+1} + \ldots + t_{j+n-i}$ представляют собой длительности всевозможных линейных компоновок блоков.

3. Алгоритм построения оптимальной компоновки (19)

2) Упорядочиваем числа x_{ij} по возрастанию с одновременным удалением избыточных одинаковых элементов и элементов $x_{ij} < \max\limits_{1 \le j \le n}\{t_j\}$. В результате получим возрастающую последовательность чисел $v_1 < v_2 < \ldots < v_k$, для которой $v_1 \le \max\limits_{1 \le j \le n}\{t_j\}$, $n - 1 \le k < \dfrac{n(n+1)}{2} - 1$.

3) Полагаем $T_0 = T(p,n,s,\varepsilon)$, $P_0 = P_n$, $l_0 = n$, $i = 1$.

4) Принимая вместимость B равной v_i, $i = \overline{1,k}$, к исходному множеству одинаково распределенных конкурирующих процессов применяем LF–алгоритм линейной упаковки. Пусть l_i – ранг полученной компоновки блоков $(q_1,\ q_2,\ \ldots,\ q_n)$.

5) Если $l_i = l_{i-1}$, то полученную компоновку LC_i не принимаем в рассмотрение, вычисляем $i = i + 1$ и переходим к п.4.

6) Вычисляем значение $T_i = T(p, LC_{l_i}, s, \varepsilon)$. Если $T_i < T_0$, то полагаем $T_0 = T_i$, $P_0 = P_t(LC_{l_i})$, иначе T_0 и P_0 оставляем без изменений.

7) Если $l_i > 2$, то вычисляем $i = i + 1$ и переходим к п.4, иначе $l_i = 2$. Алгоритм заканчивает работу.

После окончания работы алгоритма T_0 будет давать минимальное значение функционалов (1), (2), P_0 – оптимальную компоновку. Правильность его работы следует из теорем 1, 2 и соотношений (4), (5).

Приведенный алгоритм требует не более $O(n^3)$ элементарных операций, поскольку на первом этапе для построения массива чисел x_{ij}, $i = \overline{2,n}$, $j = \overline{1,i}$, требуется $O(n^2)$ элементарных операций, на втором, используя быстрые алгоритмы сортировки – $O(n^2 \log_2 n)$, на этапе 4 в цикле по v_i – не более $O(n^3)$.

3. Алгоритм построения оптимальной компоновки (20)

Пример. Пусть $P_9 = (5,2,10,3,5,5,2,5,3)$ – последовательность времен выполнения блоков q_i, $i = \overline{1,9}$, $p = 3$ – число процессоров, $s = 5$ – число блоков линейно структурированного программного ресурса, $\varepsilon = 1$ – дополнительные системные расходы на каждый блок, связанные со структурированием и конвейеризацией.

Поскольку в этом случае $5 = s > p = 3$ и суммарное время выполнения каждого из блоков Q_j всеми n процессами с учетом накладных расходов ε

$$T_\varepsilon^n = \sum_{i=1}^{n} t_i^\varepsilon = \sum_{i=1}^{9} t_i + n\varepsilon = 40 + 9 = 49 > pt_{\max}^\varepsilon = 3(10+1) = 33,$$

то общее время выполнения $n = 9$ одинаково распределенных процессов, использующих структурированный на $s = 5$ блоков программный ресурс PR, в вычислительной среде с $p = 3$ процессорами при асинхронном режиме и в режиме непрерывного выполнения каждого блока всеми процессами, согласно второй формуле в (2) составит величину

$$T(3,9,5,1) = (k+1)T_\varepsilon^n + (r-1)t_{\max}^\varepsilon = 2 \cdot 49 + 1 \cdot 11 = 109 \text{ единиц времени.}$$

Найдем линейную компоновку LC_l исходной одинаково распределенной системы, при которой достигается минимум функционалов (1) и (2) с помощью приведенного выше алгоритма.

3. Алгоритм построения оптимальной компоновки (21)

Строим массив из $\frac{9(9+1)}{2} - 1 = 44$ чисел x_{ij}, $i = \overline{2,9}$, $j = \overline{1,i}$, согласно правилу (7). На рис.1 приведена схема формирования этих чисел. Они представляют собой сумму чисел, стоящих у основания соответствующего треугольника.

i	*j*								
	1	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*
2	37	35							
3	32	32	33						
4	30	27	30	23					
5	25	25	25	20	20				
6	20	20	23	15	17	15			
7	17	15	**18**	13	12	12	10		
8	7	12	13	8	10	7	7	8	
9	5	2	10	3	5	5	2	5	3

Рис.1

Упорядочивая x_{ij} по возрастанию с одновременным удалением избыточных одинаковых элементов и элементов $x_{ij} < 10$, получим следующую возрастающую последовательность чисел:

$$(10,\ 12,\ 13,\ 15,\ 17,\ 20,\ 23,\ 25,\ 27,\ 30,\ 32,\ 33,\ 35,\ 37). \qquad (8)$$

3. Алгоритм построения оптимальной компоновки (22)

Принимая последовательно вместимость B равной значениям элементов последовательности (8), к исходной системе одинаково распределенных конкурирующих процессов применяем LF–алгоритм до тех пор, пока он не даст компоновку ранга 2. Вычисления, проводимые на каждом шаге, приведены в табл.1, где прочерк означает, что нет необходимости вычислять соответствующие значения $T(p, LC_i, s, \varepsilon)$.

Как видно из табл.1, оптимальной компоновкой будет

$$LC_5 = (Q_1 \cup Q_2,\ Q_3,\ Q_4 \cup Q_5,\ Q_6 \cup Q_7,\ Q_8 \cup Q_9),$$

для которой $P_t(LC_5) = (7{,}10{,}8{,}7{,}8)$ и $T(3{,}5{,}5{,}1) = 101$.

Таким образом, общее время выполнения исходной одинаково распределенной системы улучшено за счет оптимальной компоновки на $109 - 101 = 8$ единиц времени или на $\dfrac{8}{109} \cdot 100\% \approx 7{,}34\%$.

3. Алгоритм построения оптимальной компоновки (23)

Таблица 1

B	$P_t(LC_{l_i})$	l_i	$T(p, LC_{l_i}, s, \varepsilon)$
10	**7,10,8,7,8**	**5**	$T_1^5 = 40 + 5 = 45 > 33 = 3(10+1) = pt_{10}^1$, тогда $T(3,5,5,1) = (1+1)T_1^5 + (2-1)t_{10}^1 = 2 \cdot 45 + 1 \cdot 11 = 101$
12	7,10,8,12,3	5	–
13	7,13,12,8	4	$T_1^4 = 40 + 4 = 44 > 42 = 3(13+1) = pt_{13}^1$, тогда $T(3,4,5,1) = (1+1)T_1^4 + (2-1)t_{13}^1 = 2 \cdot 44 + 1 \cdot 14 = 102$
15	7,13,12,8	4	–
17	17,15,8	3	$T_1^3 = 40 + 3 = 43 < 54 = 3(17+1) = pt_{17}^1$, тогда $T(3,3,5,1) = T_1^3 + (5-1)t_{17}^1 = 43 + 4 \cdot 18 = 115$
20	20,17,3	3	–
23	20,17,3	3	–
25	25,15	2	$T_1^2 = 40 + 2 = 42 < 78 = 3(25+1) = pt_{25}^1$, тогда $T(3,2,5,1) = T_1^2 + (5-1)t_{25}^1 = 42 + 4 \cdot 26 = 146$

3. Алгоритм построения оптимальной компоновки (24)

Правильность работы предложенного алгоритма подтверждают результаты, полученные в других научных работах, где показано, что оптимальную одинаково распределенную систему следует искать среди стационарных одинаково распределенных систем.

Доказана следующая теорема.

3. Алгоритм построения оптимальной компоновки (25)

Теорема. Для того, чтобы эффективная одинаково распределенная система конкурирующих процессов в случае ограниченного параллелизма в асинхронном и втором синхронном режимах была оптимальной при заданных $p \geq 2$, T_ε^n, $\varepsilon > 0$, необходимо и достаточно, чтобы она была стационарной и число процессов n_0 в системе равнялось одному из чисел:

$$1)\ \left[\left[\sqrt{\frac{(p-1)T_\varepsilon^n}{k\varepsilon}}\right], \left[\sqrt{\frac{(p-1)T_\varepsilon^n}{k\varepsilon}}\right]+1\right] \cap [2,n], \text{ при } s = kp,\ k > 1,$$

$$2)\ \left[\left[\sqrt{\frac{(r-1)T_\varepsilon^n}{(k+1)\varepsilon}}\right], \left[\sqrt{\frac{(r-1)T_\varepsilon^n}{(k+1)\varepsilon}}\right]+1\right] \cap [2,n], \text{ при}$$

$s = kp + r,\ k \geq 1,\ 1 \leq r < p,$

в котором функция $\overline{\Delta}_\varepsilon(x) = (s-1)T^n\left(1 - \dfrac{1}{x}\right) - (x + s - 1)\varepsilon$, $x \geq 1$, достигает наибольшего значения, где $[z]$ – наибольшее целое, не превосходящее z, n – заданное число.

По данным примера число процессов $n_0 = \left[\sqrt{\dfrac{(r-1)T_\varepsilon^n}{(k+1)\varepsilon}}\right] + 1 = \left[\sqrt{\dfrac{40}{2}}\right] + 1 = [4{,}47] + 1 = 5$, что подтверждает ранг полученной оптимальной компоновки блоков.

Список литературы

□

СПИСОК ЛИТЕРАТУРЫ

1. Воеводин В.В., Воеводин Вл.В. Параллельные вычисления / В.В. Воеводин. – СПб.: БХВ–Петербург, 2002. – 608 с.
2. Коваленко Н.С., Самаль С.А. Вычислительные методы реализации интеллектуальных моделей сложных систем / Н.С. Коваленко. – Минск: Беларуская навука, 2004. – 166 с.
3. Топорков В.В. Модели распределенных вычислений / В.В. Топорков. – М.: Физматлит, 2004. – 320 с.
4. Иванников В.П., Коваленко Н.С., Метельский В.М. О минимальном времени реализации распределенных конкурирующих процессов в синхронных режимах. // Программирование. – 2000. – №5. – С. 44–52.
5. Капитонова Ю.В., Коваленко Н.С., Павлов П.А. Оптимальность систем одинаково распределенных конкурирующих процессов // Кибернетика и системный анализ. – 2005. – № 6. – С. 3–10.
6. Павлов П.А. Эффективность распределенных вычислений в масштабируемых системах // Научно–технические ведомости СПбГПУ. – 2010. – №1. – С. 83–89.

Спасибо за внимание