Л7. Оценка коммуникационной трудоемкости параллельных алгоритмов. Принципы разработки параллельных методов

1. Гергель В. П. Теория и практика параллельных вычислений. – М.: Интернет-Университет Информационных Технологий; БИНОМ. Лаборатория Базовых Знаний, 2007. –с. 71..89, 92-107.

2. Hockney R. The communication challenge for MPP: Intel Paragon and Meiko CS-2 // Parallel Computing. 1994. 20 (3). P. 389 -
398.

1. Коммуникационные затраты

Время передачи данных между процессорами определяет *коммуникационную составляющую* (*communication latency*) длительности выполнения параллельного алгоритма в многопроцессорной вычислительной системе. Основной набор параметров, описывающих время передачи данных, состоит из следующего ряда величин:

- *время начальной подготовки* ($t_н$) характеризует длительность подготовки сообщения для передачи, поиска маршрута в сети и т. п.;
- *время передачи служебных данных* ($t_с$) между двумя соседними процессорами (т. е. для процессоров, между которыми имеется физический канал передачи данных). К служебным данным может относиться заголовок сообщения, блок данных для обнаружения ошибок передачи и т. п.;
- *время передачи одного слова данных* по одному каналу передачи данных ($t_к$). Длительность подобной передачи определяется полосой пропускания коммуникационных каналов в сети.

2. Структура кластерных систем

Для кластерных вычислительных систем широко применяемым способом построения коммуникационной среды является использование концентраторов (hub) или коммуникаторов (switch). При этом топология сети кластера представляет собой полный граф, в котором, однако, имеются определенные ограничения на одновременность выполнения коммуникационных операций. При использовании концентраторов передача данных в каждый текущий момент может выполняться только между двумя процессорными узлами; коммуникаторы могут обеспечивать взаимодействие нескольких непересекающихся пар процессоров. Другое часто применяемое решение состоит в использовании метода передачи пакетов (часто реализуемого на основе стека протоколов TCP/IP) в качестве основного способа выполнения коммуникационных операций.

3. Оценка трудоемкости операции коммуникации

При указанных выше условиях трудоемкость операции коммуникации между процессорными узлами может быть оценена в соответствии с выражением (модель А):

$$t_{пд}(m) = t_н + m^* t_к + t_с;$$

Здесь подготовки данных *tн* предполагается постоянным (не зависящим от объема передаваемых данных), время передачи служебных данных *tc* не зависит от количества передаваемых пакетов. Эти предположения не в полной мере соответствуют действительности, и временные оценки, получаемые в результате использования модели, могут не обладать необходимой точностью.

4. Расширенная модель *B*

В рамках новой расширенной модели трудоемкость передачи данных между двумя процессорами определяется в соответствии со следующими выражениями (модель B):

$$t_{пд} = \begin{cases} t_{нач_0} + m \cdot t_{нач_1} + (m + V_c) \cdot t_к, & n = 1 \\ t_{нач_0} + (V_{пах} - V_c) \cdot t_{нач_1} + (m + V_c \cdot n) \cdot t_к, & n > 1 \end{cases},$$

5. Пояснения к модели ***B***

где $n=\lceil m/(V_{max}-V_c)\rceil$ есть количество пакетов, на которое разбивается передаваемое сообщение, величина V_{max} определяет максимальный размер пакета, который может быть доставлен в сети (по умолчанию для операционной системы MS Windows в сети Fast Ethernet V_{max} =1500 байт), а V_c есть объем служебных данных в каждом из пересылаемых пакетов (для протокола TCP/IP, ОС Windows 2000 и сети Fast Ethernet V_c=78 байт). Поясним также, что в приведенных соотношениях константа $t_{нач_0}$ характеризует аппаратную составляющую латентности и зависит от параметров используемого сетевого оборудования, значение $t_{нач_1}$ задает время подготовки одного байта данных для передачи по сети.

6. Преимущества модели *B*

Величина латентности увеличивается линейно в зависимости от объема передаваемых данных:

$$t_н = t_{нач_0} + v \cdot t_{нач_1}$$

При этом предполагается, что подготовка данных для передачи второго и всех последующих пакетов может быть совмещена с пересылкой по сети предшествующих пакетов и латентность, тем самым, не может превышать величины:

$$t_н = t_{нач_0} + (V_{max} - V_c) \cdot t_{нач_1}.$$

Учитывать увеличение объема передаваемых данных при росте числа пересылаемых пакетов за счет добавления служебной информации (заголовков пакетов):

$$(m + V_c \cdot n) \cdot t_к$$

7. Модель *C*

Для практического применения перечисленных моделей необходимо выполнить оценку значений параметров используемых соотношений. В этом отношении полезным может оказаться использование и более простых способов вычисления временных затрат на передачу данных — одной из известных схем подобного вида является подход, в котором трудоемкость операции коммуникации между двумя процессорными узлами кластера оценивается в соответствии с это моделью ***C***, предложенной Хокни (the Hockney model):

$$t_{пд}(m) = t_н + m t_к$$

,Обозначения, которые приняты в работе Хокни [2]:

$$t_{пд}(m) = \alpha + m/\beta,$$

где α есть *латентность* сети передачи данных (т. е. $\alpha = t_н$), а β обозначает *пропускную способность* сети (т. е. $\beta = R = 1/t_к$).

8. Платформа экспериментальной проверки

Для проверки адекватности рассмотренных моделей реальным процессам передачи данных приведем результаты выполненных экспериментов в сети многопроцессорного кластера Нижегородского университета (компьютеры IBM PC Pentium 4 1300 МГц и сеть Fast Etherrnet). При проведении экспериментов для реализации коммуникационных операций использовалась библиотека MPI.

9. Оценка параметров моделей

Часть экспериментов была выполнена для оценки параметров моделей: • значение латентности tн для моделей A и C определялось как время передачи сообщения нулевой длины;
• величина пропускной способности R (tк=(1/R))оценивалась максимальным значением скорости передачи данных;
• значения величин $t_{нач_0}$ и $t_{нач_1}$ оценивались при помощи линейной аппроксимации времен передачи сообщений размера от 0 до V_{max}.

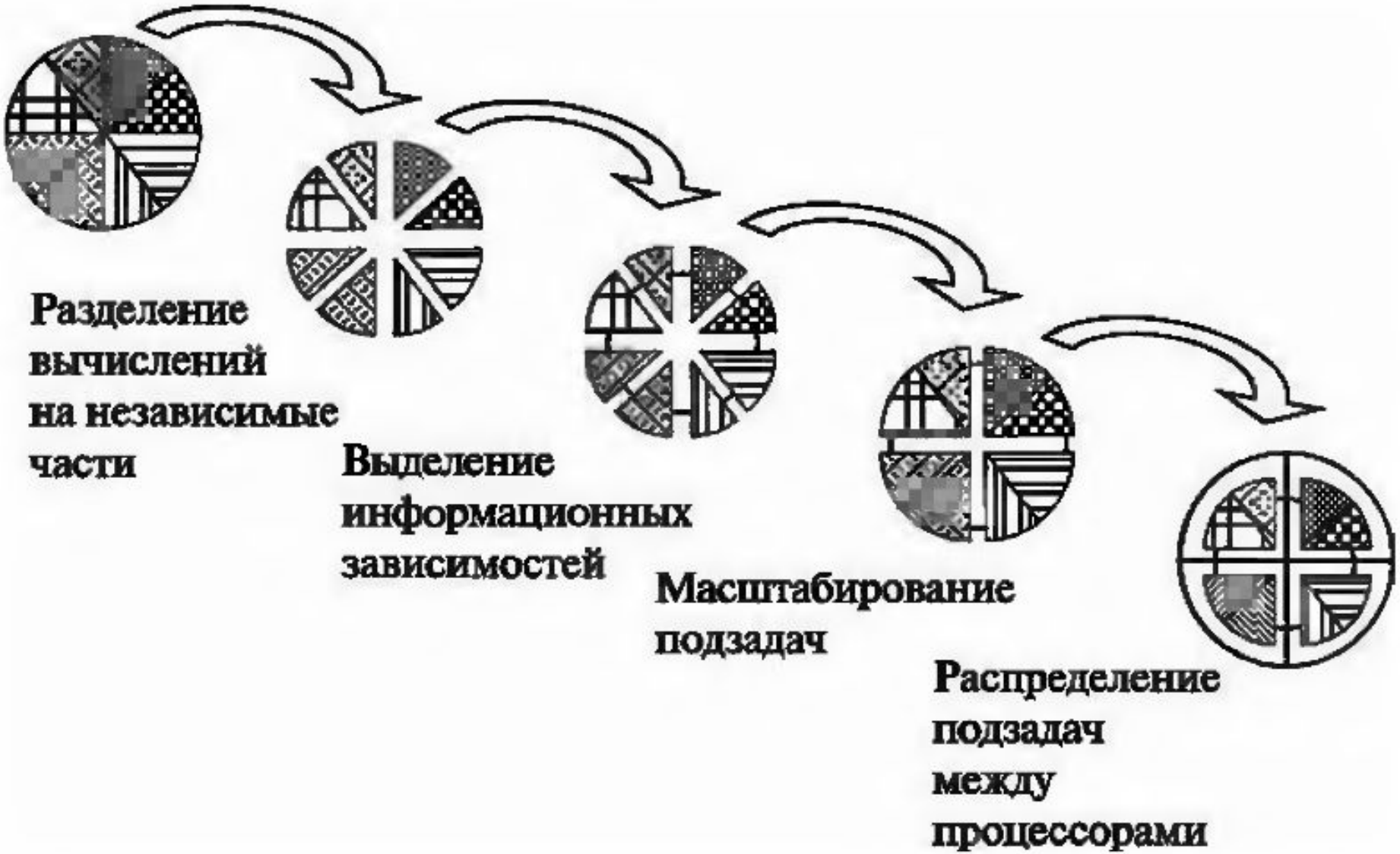
10. Методика эксперимента

В ходе экспериментов осуществлялась передача данных между двумя узлами кластера, размер передаваемых сообщений варьировался от 0 до 8 Мб. Для получения более точных оценок выполнение каждой операции осуществлялось многократно (более 100 000 раз), после чего полученные результаты усреднялись. Для иллюстрации ниже приведен результат одного эксперимента, при проведении которого размер передаваемых сообщений изменялся от 2000 до 60 000 байт.

11. Результаты вычислительных экспериментов

Объем сообщения (байт)	Время передачи (мкс)	Погрешность теоретической оценки времени передачи данных, %		
		Модель A	Модель B	Модель C
2000	495	33,45	7,93	34,80
10 000	1184	13,91	1,70	14,48
20 000	2055	8,44	0,44	8,77
30 000	2874	4,53	-1,87	4,76
40 000	3758	4,04	-1,38	4,22
50 000	4749	5,91	1,21	6,05
60 000	5730	6,97	2,73	7,09

12. Принципы разработки параллельных алгоритмов



13. Разделение вычислений на независимые части

Выбор способа разделения вычислений на независимые части основывается на анализе вычислительной схемы решения исходной задачи. Требования, которым должен удовлетворять выбираемый подход, обычно состоят в обеспечении равного объема вычислений в выделяемых подзадачах и минимума информационных зависимостей между этими подзадачами (при равных условиях нужно отдавать предпочтение редким операциям передачи сообщений большего размера по сравнению с частыми пересылками данных небольшого объема).

14. Проверка разделения вычислений на независимые части

Для оценки корректности этапа разделения вычислений на независимые части можно воспользоваться контрольным списком вопросов:
- выполненная декомпозиция не увеличивает объем вычислений и необходимый объем памяти?
- возможна ли при выбранном способе декомпозиции равномерная загрузка всех имеющихся процессоров?
- достаточно ли выделенных частей процесса вычислений для эффективной загрузки имеющихся процессоров (с учетом возможности увеличения их количества)?

15. Определение информационных зависимостей

При наличии вычислительной схемы решения задачи после выделения базовых подзадач определение информационных зависимостей между ними обычно не вызывает больших затруднений. При этом, следует отметить, что на самом деле этапы выделения подзадач и информационных зависимостей достаточно сложно поддаются разделению. Выделение подзадач должно происходить с учетом возникающих информационных связей, после анализа объема и частоты необходимых информационных обменов между подзадачами может потребоваться повторение этапа разделения вычислений.

16. Анализ информационных зависимостей

Предпочтительные формы информационного взаимодействия выделены подчеркиванием: • локальные и глобальные схемы передачи данных — для локальных схем передачи данных в каждый момент времени выполняются только между небольшим числом подзадач, в глобальных операциях принимают участие все подзадачи;
• структурные и произвольные способы взаимодействия — структурные взаимодействия по стандартным схемам коммуникации, для произвольных структур схема не носит характера однородности;
• статические или динамические схемы передачи данных — для статических схем моменты и участники информационного взаимодействия фиксируются на этапах проектирования и разработки параллельных программ, динамические определяются в ходе выполняемых вычислений;
• синхронные и асинхронные способы взаимодействия — для синхронных способов операции передачи данных выполняются только при готовности всех участников взаимодействия и завершаются только после полного окончания всех коммуникационных действий, при асинхронном выполнении операций это не обязательно.

17. Оценка выделения информационных зависимостей

Для оценки правильности этапа выделения информационных зависимостей следует воспользоваться контрольным списком вопросов:
- соответствует ли вычислительная сложность подзадач интенсивности их информационных взаимодействий?
- является ли одинаковой интенсивность информационных взаимодействий для разных подзадач?
- является ли схема информационного взаимодействия локальной?
- не препятствует ли выявленная информационная зависимость параллельному решению подзадач?

18. Масштабирование набора подзадач

Масштабирование схемы параллельных вычислений проводится в случае, если количество подзадач отличается от числа используемых процессоров. Для сокращения количества подзадач выполняется укрупнение (агрегация) вычислений. Применяемые здесь правила совпадают с рекомендациями начального этапа выделения подзадач: определяемые подзадачи, как и ранее, должны иметь одинаковую вычислительную сложность, а объем и интенсивность информационных взаимодействий между подзадачами должны оставаться минимальным. Первыми претендентами на объединение являются подзадачи с высокой степенью информационной взаимозависимости.
При недостаточном количестве имеющихся подзадач для загрузки всех доступных процессоров выполняется детализация (декомпозиция) вычислений. Выполнение этапа масштабирования вычислений должно свестись к разработке правил агрегации и декомпозиции подзадач, которые должны параметрически зависеть от числа процессоров, применяемых для вычислений.

19. Оценка масштабирования

Список контрольных вопросов для оценки правильности этапа масштабирования, выглядит следующим образом:

- не ухудшится ли локальность вычислений после масштабирования имеющегося набора подзадач?
- имеют ли подзадачи после масштабирования одинаковую вычислительную и коммуникационную сложность?
- соответствует ли количество задач числу имеющихся процессоров?
- зависят ли параметрически правила масштабирования от количества процессоров?

20. Распределение подзадач между процессорами

Управление распределением нагрузки для процессоров возможно только для вычислительных систем с распределенной памятью, для мультипроцессоров (систем с общей памятью) распределение нагрузки обычно выполняется операционной системой автоматически. Данный этап распределения подзадач между процессорами является избыточным, если количество подзадач совпадает с числом имеющихся процессоров. Основной показатель успешности выполнения данного этапа —эффективность использования процессоров. Пути достижения хороших результатов остаются прежними: необходимо обеспечить равномерное распределение вычислительной нагрузки между процессорами и минимизировать количество сообщений, передаваемых между ними. Точно так же как и на предшествующих этапах проектирования, оптимальное решение проблемы распределения подзадач между процессорами основывается на анализе информационной связности графа «подзадачи — сообщения».

21. Оценка этапа распределения подзадач

Перечень контрольных вопросов для проверки этапа распределения подзадач состоит в следующем: • не приводит ли распределение нескольких задач на один процессор к росту дополнительных вычислительных затрат?

• существует ли необходимость динамической балансировки вычислений?

• не является ли процессор-менеджер «узким» местом при использовании схемы «менеджер — исполнитель»?