Высокопроизводительные параллельные вычисления

Лекция №3

Тема: Оценка коммуникационной трудоемкости параллельных алгоритмов

Козбакова Айнур Холдасовна– PhD, профессор, кафедра «Информационные системы» Факультет «Инжиниринг и информационных технологии» 01.09.2025 г.

Алгоритмы маршрутизации

Алгоритмы маршрутизации определяют путь передачи данных от процессора – источника сообщения до процессора, к которому сообщение должно быть доставлено. Среди возможных способов решения данной задачи различают:

– *оптимальные*, определяющие всегда наикратчайшие пути передачи данных, и неоптимальные *алгоритмы маршрутизации*;

– *детерминированные* и *адаптивные* методы выбора маршрутов (адаптивные алгоритмы определяют пути передачи данных в зависимости от существующей загрузки коммуникационных каналов).

К числу наиболее распространенных оптимальных алгоритмов относится класс *методов покоординатной маршрутизации* (*dimension–ordered routing*), в которых поиск путей передачи данных осуществляется поочередно для каждой размерности топологии сети коммуникации. Так, для двумерной решетки такой подход приводит к маршрутизации, при которой передача данных сначала выполняется по одному направлению (например, по горизонтали до достижения вертикали, на которой располагается процессор назначения), а затем данные передаются вдоль другого направления (данная схема известна под названием *алгоритма XY–маршрутизации*). Для гиперкуба покоординатная схема маршрутизации может состоять, например, в циклической передаче данных процессору, определяемому первой различающейся битовой позицией в номерах процессоров — того, на котором сообщение располагается в данный момент времени, и того, на который оно должно быть передано

Методы передачи данных

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 (tн) характеризует длительность подготовки сообщения для передачи, поиска маршрута в сети и т. п.;*

– *время передачи служебных данных (tс) 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 (tк). Длительность подобной передачи определяется полосой пропускания коммуникационных каналов в сети.*

К числу наиболее распространенных *методов передачи данных* относятся два основных способа коммуникации. Первый из них ориентирован на *передачу сообщений* (*метод передачи сообщений* или *МПС*) как неделимых (атомарных) блоков информации (*store–and–forward routing* или *SFR*). При таком подходе процессор, содержащий сообщение для передачи, готовит весь объем данных для передачи, определяет процессор, которому следует направить данные, и запускает операцию пересылки данных. Процессор, которому направлено сообщение, в первую очередь осуществляет прием полностью всех пересылаемых данных и только затем приступает к пересылке принятого сообщения далее по маршруту. Время пересылки данных $t_{пд}$ для метода передачи сообщения размером m байт по маршруту длиной l определяется выражением:

$$t_{пд} = t_н + (mt_к + t_с)l.$$

При достаточно длинных сообщениях временем передачи служебных данных можно пренебречь и выражение для времени передачи данных может быть записано в более простом виде:

$$t_{n\partial} = t_н + mt_к l.$$

Второй способ коммуникации основывается на представлении пересылаемых сообщений в виде блоков информации меньшего размера – пакетов, в результате чего передача данных может быть сведена к *передаче пакетов* (*метод передачи пакетов* или *МПП*). При таком методе коммуникации (*cut–through routing* или *CTR*) принимающий процессор может осуществлять пересылку данных по дальнейшему маршруту непосредственно сразу после приема очередного пакета, не дожидаясь завершения приема данных всего сообщения. Время пересылки данных при использовании метода передачи пакетов определяется выражением:

$$t_{n\partial} = t_н + mt_к + t_c l.$$

Сравнивая полученные выражения, можно заметить, что в большинстве случаев метод передачи пакетов приводит к более быстрой пересылке данных; кроме того, данный подход снижает потребность в памяти для хранения пересылаемых данных при организации приема–передачи сообщений, а для передачи пакетов могут использоваться одновременно разные коммуникационные каналы. С другой стороны, реализация пакетного метода требует разработки более сложного аппаратного и программного обеспечения сети, может увеличить накладные расходы (время подготовки и время передачи служебных данных). Кроме того, при передаче пакетов возможно возникновение конфликтных ситуаций (дедлоков).

Анализ трудоемкости основных операций передачи данных.

При всем разнообразии выполняемых операций передачи данных при параллельных способах решения сложных научно–технических задач, определенные процедуры взаимодействия процессоров сети могут быть отнесены к числу основных коммуникационных действий, либо наиболее широко распространенных в практике параллельных вычислений, либо тех, к которым могут быть сведены многие другие процессы приема–передачи сообщений. Важно отметить также, что в рамках подобного базового набора для большинства операций коммуникации существуют процедуры, обратные по действию исходным операциям (так, например, операции передачи данных от одного процессора всем имеющимся процессорам сети соответствует операция приема в одном процессоре сообщений от всех остальных процессоров). Как результат, рассмотрение коммуникационных процедур целесообразно выполнять попарно, поскольку во многих случаях алгоритмы выполнения прямой и обратной операций могут быть получены исходя из одинаковых предпосылок.

Рассмотрение основных операций передачи данных осуществим на примере таких топологий сети, как кольцо, двумерная решетка и гиперкуб. Для двумерной решетки будет предполагаться также, что между граничными процессорами в строках и столбцах решетки имеются каналы передачи данных (т.е. топология сети представляет собой тор). Как и ранее, величина m будет означать размер сообщения в словах, значение p определяет количество процессоров в сети, а переменная N задает размерность топологии гиперкуба.

2.3 Передача данных между двумя процессорами сети

Трудоемкость данной коммуникационной операции может быть получена путем подстановки длины максимального пути (диаметра сети) в выражения для времени передачи данных при разных методах коммуникации таб.2.1.

Таблица 2.1. Время передачи данных между двумя процессорами

Топология	Передача сообщений	Передача пакетов
Кольцо	$t_н + mt_к\lfloor p/2 \rfloor$	$t_н + mt_к + t_c\lfloor p/2 \rfloor$

Решетка–тор	$t_н + 2mt_к\lfloor\sqrt{p}/2\rfloor$	$t_н + mt_к + 2t_c\lfloor\sqrt{p}/2\rfloor$
Гиперкуб	$t_н + mt_к \log_2 p$	$t_н + mt_к + t_c \log_2 p$

Передача данных от одного процессора всем остальным процессорам сети.

Операция передачи данных (одного и того же сообщения) от одного процессора всем остальным процессорам сети (*one–to–all broadcast* или *single–node broadcast*) является одним из наиболее часто выполняемых коммуникационных действий. Двойственная ей операция – прием на одном процессоре сообщений от всех остальных процессоров сети (*single–node accumulation*). Подобные операции используются, в частности, при реализации матрично–векторного умножения, решении систем линейных уравнений методом Гаусса, решении задачи поиска кратчайших путей и др.

Простейший способ реализации операции рассылки состоит в ее выполнении как последовательности попарных взаимодействий процессоров сети. Однако при таком подходе большая часть пересылок является избыточной и возможно применение более эффективных алгоритмов коммуникации. Изложение материала будет проводиться сначала для метода передачи сообщений, затем – для пакетного способа передачи данных.

Передача сообщений.

Для кольцевой топологии процессор – источник рассылки может инициировать передачу данных сразу двум своим соседям, которые, в свою очередь, приняв сообщение, организуют пересылку далее по кольцу. Трудоемкость выполнения операции рассылки в этом случае будет определяться соотношением:

$$t_{n\partial} = (t_н + mt_к)\lceil p/2\rceil.$$

Для топологии типа решетка–тор алгоритм рассылки может быть получен из способа передачи данных, примененного для кольцевой структуры сети. Так, рассылка может быть выполнена в виде двухэтапной процедуры. На первом этапе организуется передача сообщения всем процессорам сети, располагающимся на той же горизонтали решетки, что и процессор – инициатор передачи. На втором этапе процессоры, получившие копию данных на первом этапе, рассылают сообщения по своим соответствующим вертикалям. Оценка длительности операции рассылки в соответствии с описанным алгоритмом определяется соотношением:

$$t_{n\partial} = 2(t_н + mt_к)\lceil\sqrt{p}/2\rceil.$$

Для гиперкуба рассылка может быть выполнена в ходе N–этапной процедуры передачи данных. На первом этапе процессор–источник сообщения передает данные одному из своих соседей (например, по первой размерности) – в результате после первого этапа есть два процессора, имеющих копию пересылаемых данных (данный результат можно интерпретировать также как разбиение исходного гиперкуба на два таких одинаковых по размеру гиперкуба размерности N–1, что каждый из них имеет копию исходного сообщения). На втором этапе два процессора, задействованные на первом этапе, пересылают сообщение своим соседям по второй размерности и т.д. В результате такой рассылки время операции оценивается при помощи выражения:

$$t_{n\partial} = (t_н + mt_к)\log_2 p.$$

Сравнивая полученные выражения для длительности выполнения операции рассылки, можно отметить, что наилучшие показатели имеет топология типа гиперкуб;

более того, можно показать, что данный результат является наилучшим для выбранного способа коммуникации с помощью передачи сообщений.

Передача пакетов.

Для топологии типа кольцо алгоритм рассылки может быть получен путем логического представления кольцевой структуры сети в виде гиперкуба. В результате на этапе рассылки процессор – источник сообщения передает данные процессору, находящемуся на расстоянии p/2 от исходного процессора. Далее, на втором этапе оба процессора, уже имеющие рассылаемые данные после первого этапа, передают сообщения процессорам, находящимся на расстоянии p/4, и т.д. Трудоемкость выполнения операции рассылки при таком *методе передачи данных* определяется соотношением:

$$t_{n\partial} = \sum_{i=1}^{\log_2 p} (t_н + mt_к + t_c p/2^i) = (t_н + mt_к)\log_2 p + t_c(p-1)$$

как и ранее, при достаточно больших сообщениях временем передачи служебных данных можно пренебречь.

Для топологии типа решетка–тор алгоритм рассылки может быть получен из способа передачи данных, примененного для кольцевой структуры сети, в соответствии с тем же способом обобщения, что и в случае использования метода передачи сообщений. Получаемый в результате такого обобщения алгоритм рассылки характеризуется следующим соотношением для оценки времени выполнения:

$$t_{n\partial} = (t_н + mt_к)\log_2 p + 2t_c(\sqrt{p} - 1).$$

Для гиперкуба алгоритм рассылки (и, соответственно, временные оценки длительности выполнения) при передаче пакетов не отличается от варианта для метода передачи сообщений.

Передача данных от всех процессоров всем процессорам сети.

Операция передачи данных от всех процессоров всем процессорам сети (*all–to–all broadcast* или *multinode broadcast*) является естественным обобщением одиночной операции рассылки, двойственная ей операция – *прием сообщений на каждом процессоре от всех процессоров сети* (*multinode accumulation*). Подобные операции широко используются, например, при реализации матричных вычислений.

Возможный способ реализации операции множественной рассылки состоит в выполнении соответствующего набора операций одиночной рассылки. Однако такой подход не является оптимальным для многих топологий сети, поскольку часть необходимых операций одиночной рассылки потенциально может быть выполнена параллельно. Как и ранее, материал будет рассматриваться раздельно для разных *методов передачи данных*.

Передача сообщений.

Для кольцевой топологии каждый процессор может инициировать рассылку своего сообщения одновременно (в каком–либо выбранном направлении по кольцу). В любой момент каждый процессор выполняет прием и передачу данных, завершение операции множественной рассылки произойдет через p–1 цикл передачи данных. Длительность выполнения операции рассылки оценивается соотношением:

$$t_{n\partial} = (t_н + mt_к)(p-1).$$

Для топологии типа решетка–тор множественная рассылка сообщений может быть выполнена при помощи алгоритма, получаемого обобщением способа передачи данных для кольцевой структуры сети. Схема обобщения состоит в следующем. На первом этапе организуется передача сообщений раздельно по всем процессорам сети, располагающимся на одних и тех же горизонталях решетки (в результате на каждом процессоре одной и той же горизонтали формируются укрупненные сообщения размера $m\sqrt{p}$, объединяющие все сообщения горизонтали). Время выполнения этапа:

$$t'_{n\partial} = (t_н + mt_к)(\sqrt{p} - 1).$$

На втором этапе рассылка данных выполняется по процессорам сети, образующим вертикали решетки. Длительность этого этапа:

$$t''_{n\partial} = (t_н + m\sqrt{p}t_к)(\sqrt{p} - 1).$$

Общая длительность операции рассылки определяется соотношением:

$$t_{n\partial} = 2t_н(\sqrt{p} - 1) + mt_к(p - 1).$$

Для гиперкуба алгоритм множественной рассылки сообщений может быть получен путем обобщения ранее описанного способа передачи данных для топологии типа решетки на размерность гиперкуба N. В результате такого обобщения схема коммуникации состоит в следующем. На каждом этапе i, $1 \le i \le N$, выполнения алгоритма функционируют все процессоры сети, которые обмениваются своими данными со своими соседями по i–ой размерности и формируют объединенные сообщения. Время операции рассылки может быть получено при помощи выражения:

$$t_{n\partial} = \sum_{i=1}^{\log_2 p} (t_н + 2^{i-1}mt_к) = t_н \log_2 p + mt_к(p - 1).$$

Передача пакетов.

Применение более эффективного для кольцевой структуры и топологии типа решетка–тор метода передачи данных не приводит к какому–либо улучшению времени выполнения операции множественной рассылки, поскольку обобщение алгоритмов выполнения операции одиночной рассылки на случай множественной рассылки приводит к перегрузке каналов передачи данных (т.е. к существованию ситуаций, когда в один и тот же момент для передачи по одной и той же линии имеется несколько ожидающих пересылки пакетов данных). Перегрузка каналов приводит к задержкам при пересылках данных, что и не позволяет проявиться всем преимуществам метода передачи пакетов.

Широко распространенным примером операции множественной рассылки является *задача редукции* (*reduction*), которая определяется в общем виде как процедура выполнения той или иной обработки данных, получаемых на каждом процессоре в ходе множественной рассылки (в качестве примера такой задачи может быть рассмотрена проблема вычисления суммы значений, находящихся на разных процессорах, и рассылки полученной суммы по всем процессорам сети). Способы решения *задачи редукции* могут состоять в следующем:

– непосредственный подход заключается в выполнении операции множественной рассылки и последующей затем обработке данных на каждом процессоре в отдельности;

– более эффективный алгоритм может быть получен в результате применения операции одиночного приема данных на отдельном процессоре, выполнения на этом процессоре действий по обработке данных и рассылки полученного результата обработки всем процессорам сети;

– наилучший же способ решения *задачи редукции* состоит в совмещении процедуры множественной рассылки и действий по обработке данных, когда каждый процессор сразу же после приема очередного сообщения реализует требуемую обработку полученных данных (например, выполняет сложение полученного значения с имеющейся на процессоре частичной суммой). Время решения *задачи редукции* при таком алгоритме реализации в случае, например, когда размер пересылаемых данных имеет единичную длину (m=1) и топология сети имеет структуру гиперкуба, определяется выражением:

$$t_{n\partial} = (t_н + t_к)\log_2 p.$$

Другим типовым примером использования операции множественной рассылки является задача нахождения частных сумм последовательности значений S_i (в англоязычной литературе эта задача известна под названием *prefix sum problem*)

$$S_k = \sum_{i=1}^{k} x_i, \quad 1kp$$

будем предполагать, что количество значений совпадает с количеством процессоров, значение x_i располагается на i–м процессоре и результат S_k должен получаться на процессоре с номером *k*.

Алгоритм решения данной задачи также может быть получен при помощи конкретизации общего способа выполнения множественной операции рассылки, когда процессор выполняет суммирование полученного значения (но только в том случае, если процессор – отправитель значения имеет меньший номер, чем процессор–получатель).