Нижегородский государственный университет им. Н.И.Лобачевского

Факультет Вычислительной математики и кибернетики

Образовательный комплекс

Введение в методы параллельного программирования

Раздел 3.

Оценка коммуникационной трудоемкости параллельных алгоритмов

Гергель В.П., профессор, д.т.н.
Кафедра математического обеспечения ЭВМ

Содержание

- Общая характеристика механизмов передачи данных
  - Алгоритмы маршрутизации
  - Методы передачи данных
- Анализ трудоемкости основных операций передачи данных
  - Передача данных между двумя процессорами сети
  - Передача данных от одного процессора всем остальным процессорам сети
  - Передача данных от всех процессоров всем процессорам сети
  - Обобщенная передача данных от одного процессора всем остальным процессорам сети
  - Обобщенная передача данных от всех процессоров всем процессорам сети
  - Циклический сдвиг
- Методы логического представления топологии коммуникационной среды
- Оценка трудоемкости операций передачи данных для кластерных систем
- Заключение

Введение

- Данный раздел посвящен вопросам анализа информационных потоков, возникающих при выполнении параллельных алгоритмов:
  - дается общая характеристика механизмов передачи данных,
  - проводится анализ трудоемкости основных операций обмена информацией,
  - рассматриваются методы логического представления структуры МВС.

Затраты на организацию взаимодействия при помощи передачи сообщений существенным образом влияют на эффективность параллельных вычислений.

Общая характеристика механизмов передачи данных…

- **Алгоритмы маршрутизации** определяют путь передачи данных от процессора-источника сообщения до процессора, к которому сообщение должно быть доставлено:
  - *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).

Общая характеристика механизмов передачи данных…

- **Алгоритмы маршрутизации**
  - *метод покоординатной маршрутизации* (*dimension-ordered routing*) – один из самых распространенных оптимальных методов маршрутизации:
    - Поиск путей передачи данных осуществляется поочередно для каждой размерности топологии сети коммуникации,
    - Для двумерной решетки: передача данных сначала выполняется по одному направлению, а затем данные передаются вдоль другого направления (*алгоритм XY-маршрутизации*),
    - Для гиперкуба: циклическая передача данных процессору, определяемому первой различающейся битовой позицией в номерах процессоров, на котором сообщение располагается в данный момент времени и на который сообщение должно быть передано.

Общая характеристика механизмов передачи данных…

Методы передачи данных…

Время передачи данных между процессорами определяет коммуникационную составляющую (*communication latency*) длительности выполнения параллельного алгоритма.

Основной набор параметров, используемый при оценке времени передачи данных, включает:

- **время начальной подготовки** ($t_н$) характеризует длительность подготовки сообщения для передачи, поиска маршрута в сети и т.п.,
- **время передачи служебных данных** (t_c) 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.

Общая характеристика механизмов передачи данных…

Методы передачи данных…

Метод передачи сообщений (*МПС*) осуществляет передачу данных как неделимых (атомарных) блоков информации (*store-and-forward routing* or *SFR*):

- 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.

Общая характеристика механизмов передачи данных…

- **Методы передачи данных…**

  Метод передачи пакетов (*МПП*) основан на представлении пересылаемых сообщений в виде блоков информации меньшего размера (*cut-through routing* or *CTR*):

  – принимающий процессор может осуществлять пересылку данных по дальнейшему маршруту непосредственно сразу после приема очередного пакета, не дожидаясь завершения приема данных всего сообщения.

Общая характеристика механизмов передачи данных…

Методы передачи данных…

Время пересылки данных $t_{пд}$ размером m байт по маршруту длиной l определяется выражением:

– Для метода передачи сообщений:

$$t_{пд} = t_н + (mt_к + t_с)l \ ,$$

при достаточно длинных сообщениях временем пересылки служебных данных можно пренебречь:

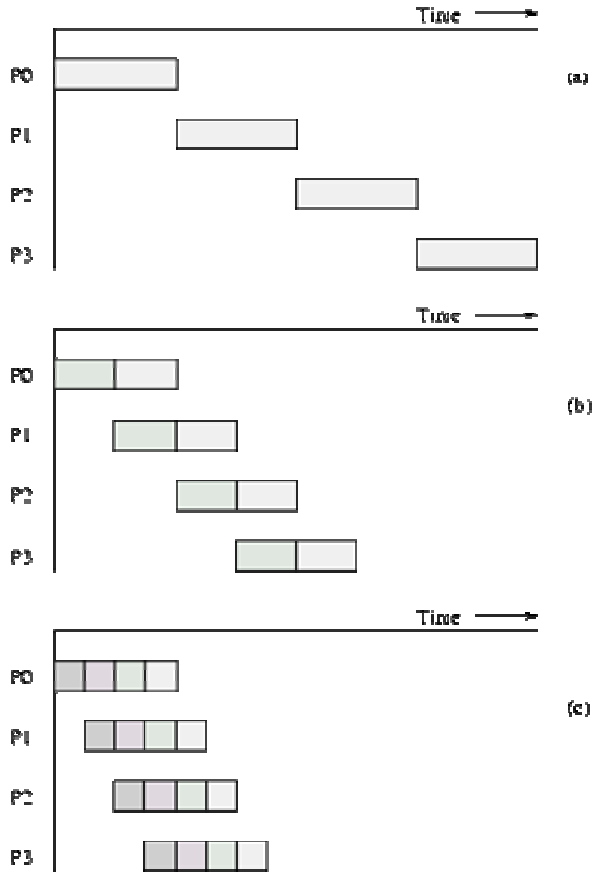
$$t_{пд} = t_н + mt_к l \ ;$$

– Для метода передачи пакетов:

$$t_{пд} = t_н + mt_к + t_с l \ .$$

Общая характеристика механизмов передачи данных…

Методы передачи данных…



- Метод передачи сообщений
- Метод пересылки пакетов (сообщение разбивается на 2 пакета)
- Метод пересылки пакетов (сообщение разбивается на 4 пакета)

Общая характеристика механизмов передачи данных

- Метод передачи пакетов (оценка применимости):
  - приводит к более быстрой пересылке данных,
  - снижает потребность в памяти для хранения пересылаемых данных для организации приема-передачи сообщений,
  - для передачи могут использоваться одновременно разные коммуникационные каналы,
  - требует разработки более сложного аппаратного и программного обеспечения сети,
  - может увечить накладные расходы (время подготовки и время передачи служебных данных),
  - при передаче пакетов возможно возникновение конфликтных ситуаций (*дедлоков*).

Анализ трудоемкости основных операций передачи данных…

- При анализе параллельных способов решения сложных научно-технических задач могут быть выделены основные коммуникационные действия, которые или наиболее широко распространены в практике, или к которым могут быть сведены многие другие процессы приема-передачи сообщений,
- 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).

Анализ трудоемкости основных операций передачи данных…

- **Передача данных между двумя процессорами сети**

  Трудоемкость данной коммуникационной операции может быть получена путем подстановки длины максимального пути в выражения для времени передачи данных при разных методах коммуникации.

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	$t_н + mt_к \log_2 p$	$t_н + mt_к + t_c \log_2 p$

Анализ трудоемкости основных операций передачи данных…

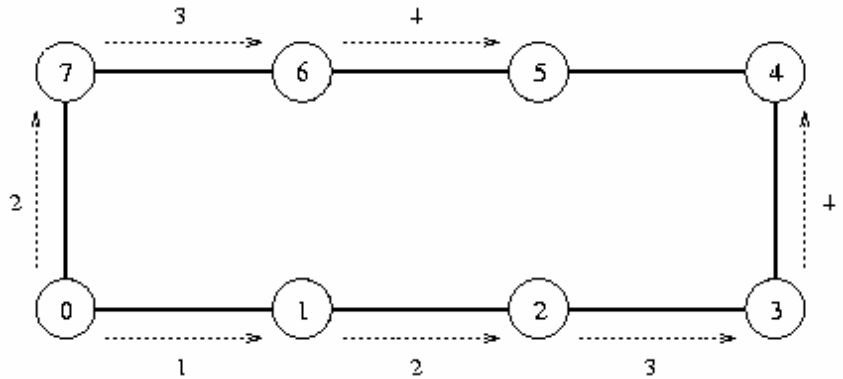
- **Передача данных от одного процессора всем остальным процессорам сети…**

  Операция передачи данных (одного и того же сообщения) от одного процессора всем остальным процессорам сети (*one-to-all broadcast* or *single-node broadcast*) является одним из наиболее часто выполняемых коммуникационных действий; двойственная операция передачи – прием на одном процессоре сообщений от всех остальных процессоров сети (*single-node accumulation*).

Анализ трудоемкости основных операций передачи данных…

- **Передача данных от одного процессора всем остальным процессорам сети (*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:

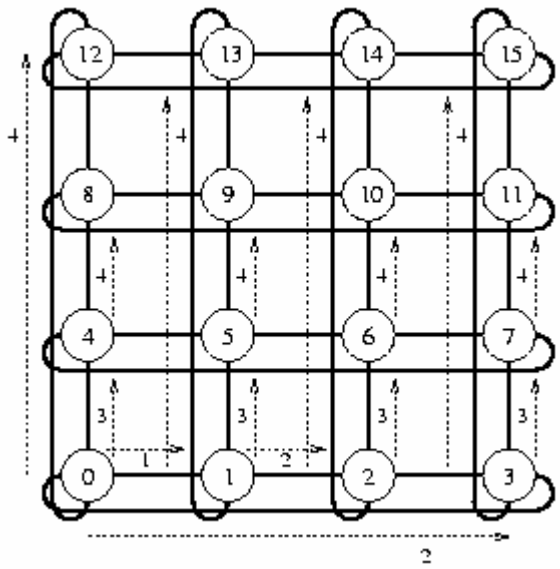


  Трудоемкость выполнения операции рассылки в этом случае будет определяться соотношением:

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

Анализ трудоемкости основных операций передачи данных…

- **Передача данных от одного процессора всем остальным процессорам сети (*передача сообщений*)…**



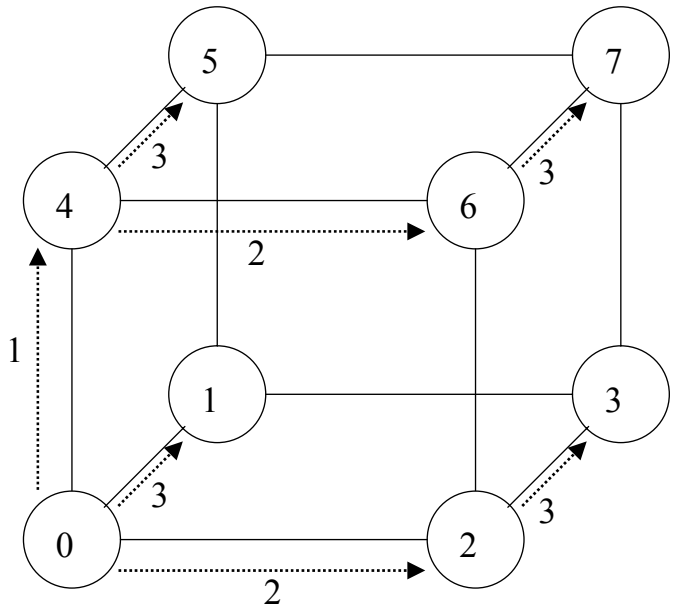
Для топологии типа **решетки-тора** рассылка может быть выполнена в виде двухэтапной процедуры. На первом этапе организуется передача сообщения всем процессорам сети, располагающимся на той же горизонтали решетки, что и процессор-инициатор передачи; на втором этапе процессоры, получившие копию данных на первом этапе, рассылают сообщения по своим соответствующим вертикалям.

Длительности операции рассылки в соответствии с описанным алгоритмом определяется соотношением:

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

Анализ трудоемкости основных операций передачи данных…

- **Передача данных от одного процессора всем остальным процессорам сети (*передача сообщений*)**



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 – в результате после первого этапа имеется два процессора, имеющих копию пересылаемых данных. 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

$$t_{пд} = (t_н + mt_к)\log_2 p$$

Анализ трудоемкости основных операций передачи данных…

- **Передача данных от одного процессора всем остальным процессорам сети (*передача пакетов*)…**

Для топологии типа **кольца** 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, находящиеся на расстоянии *p/4* и т.д.

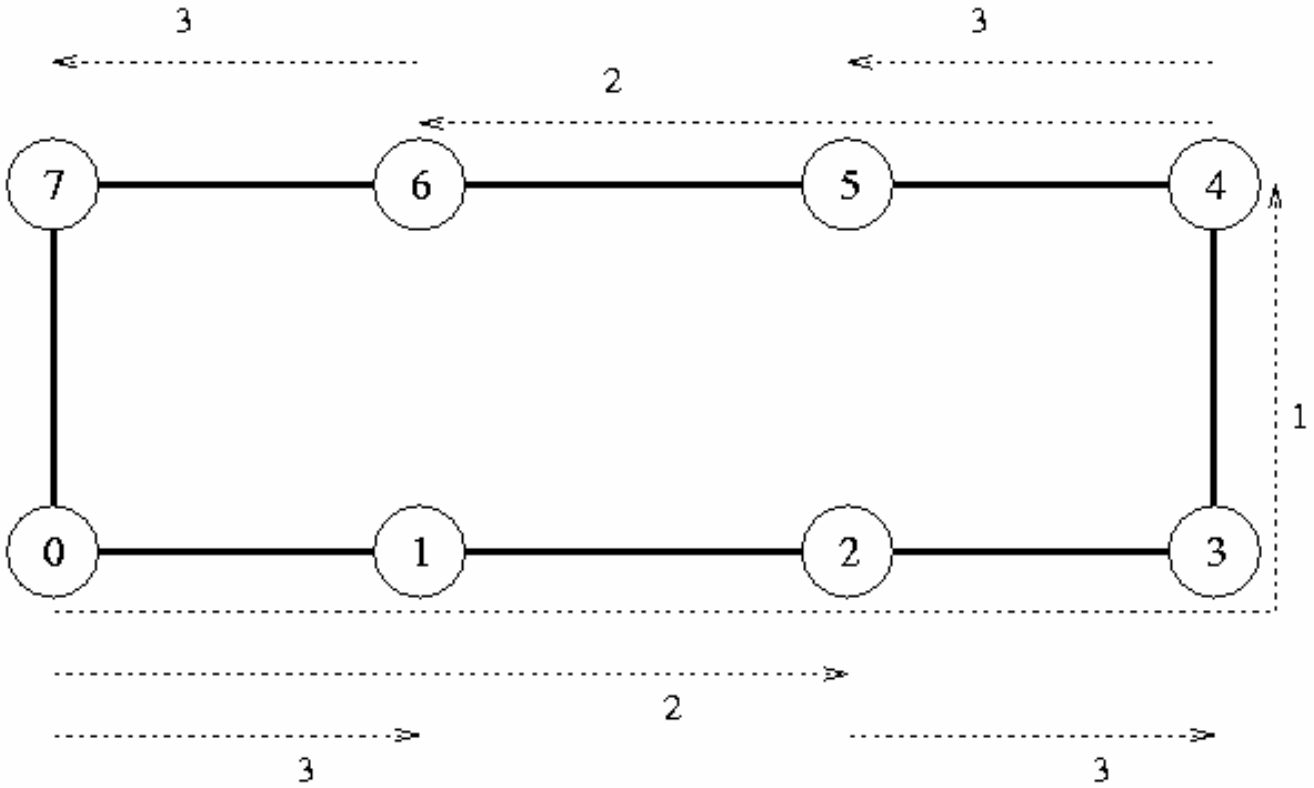
Трудоемкость выполнения операции рассылки при таком методе передачи данных определяется соотношением:

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

Анализ трудоемкости основных операций передачи данных…

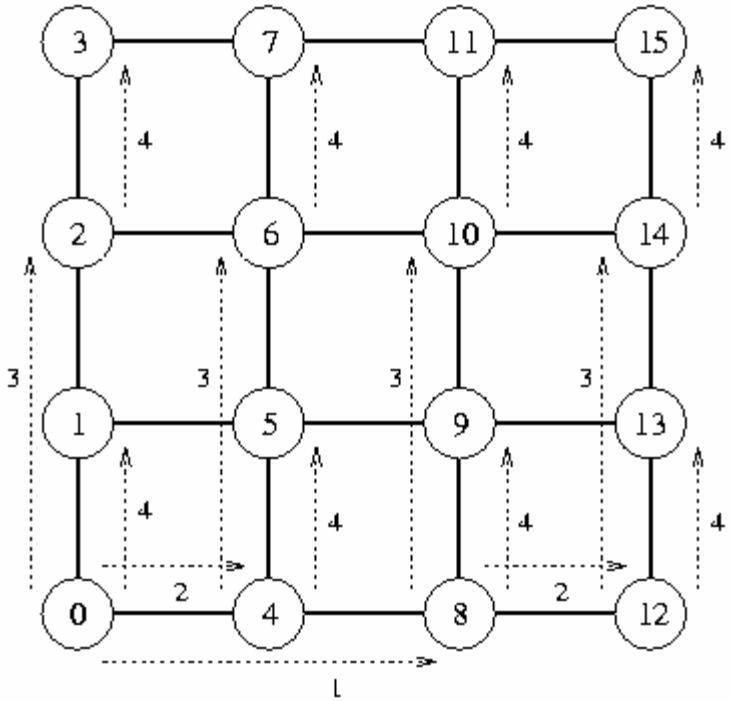
- **Передача данных от одного процессора всем остальным процессорам сети (*передача пакетов*)…**

  Топология типа кольца



Анализ трудоемкости основных операций передачи данных…

- **Передача данных от одного процессора всем остальным процессорам сети (*передача пакетов*)**



Для топологии типа **решетки-тора** 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.

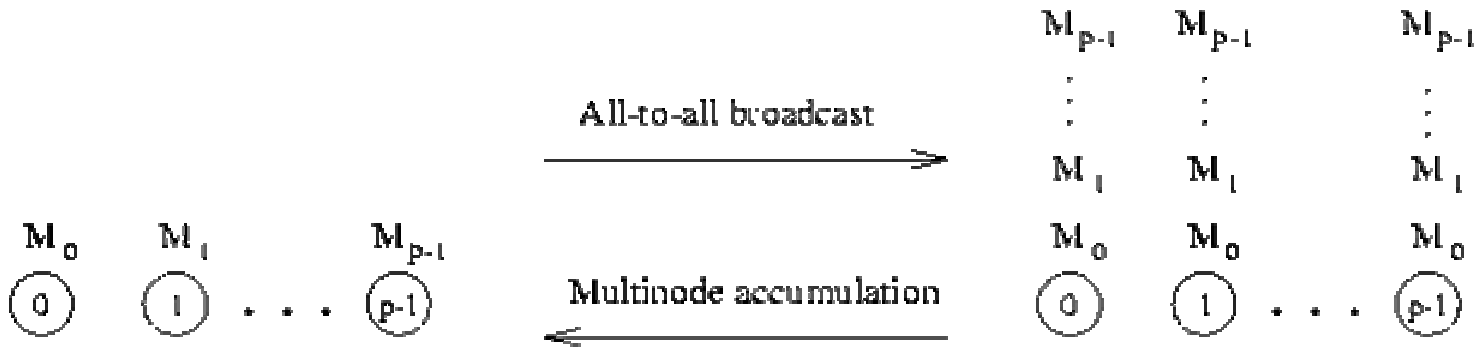
Получаемый в результате такого обобщения алгоритм рассылки характеризуется следующим соотношением для оценки времени выполнения:

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

Анализ трудоемкости основных операций передачи данных…

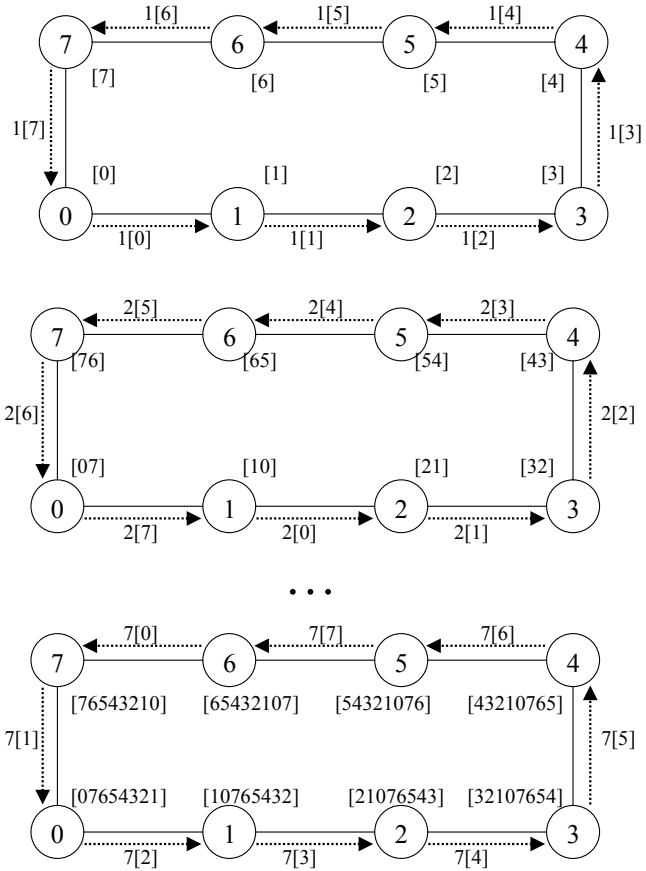
- **Передача данных от всех процессоров всем процессорам сети…**

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



Анализ трудоемкости основных операций передачи данных…

- **Передача данных от всех процессоров всем процессорам сети (*передача сообщений*)…**



Для **кольцевой топологии** каждый процессор может инициировать рассылку своего сообщения одновременно (в каком-либо выбранном направлении по кольцу). В любой момент времени каждый процессор выполняет прием и передачу данных; завершение операции множественной рассылки произойдет через (*p-1*) цикл передачи данных.

Длительность выполнения операции рассылки оценивается соотношением:

$$t_{пд} = (t_н + mt_к)(p-1)$$

Анализ трудоемкости основных операций передачи данных…

- **Передача данных от всех процессоров всем процессорам сети (*передача сообщений*)…**

  Для топологии типа **решетки-тора** множественная рассылка сообщений может быть выполнена при помощи алгоритма, получаемого обобщением способа передачи данных для кольцевой структуры сети:

  – 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

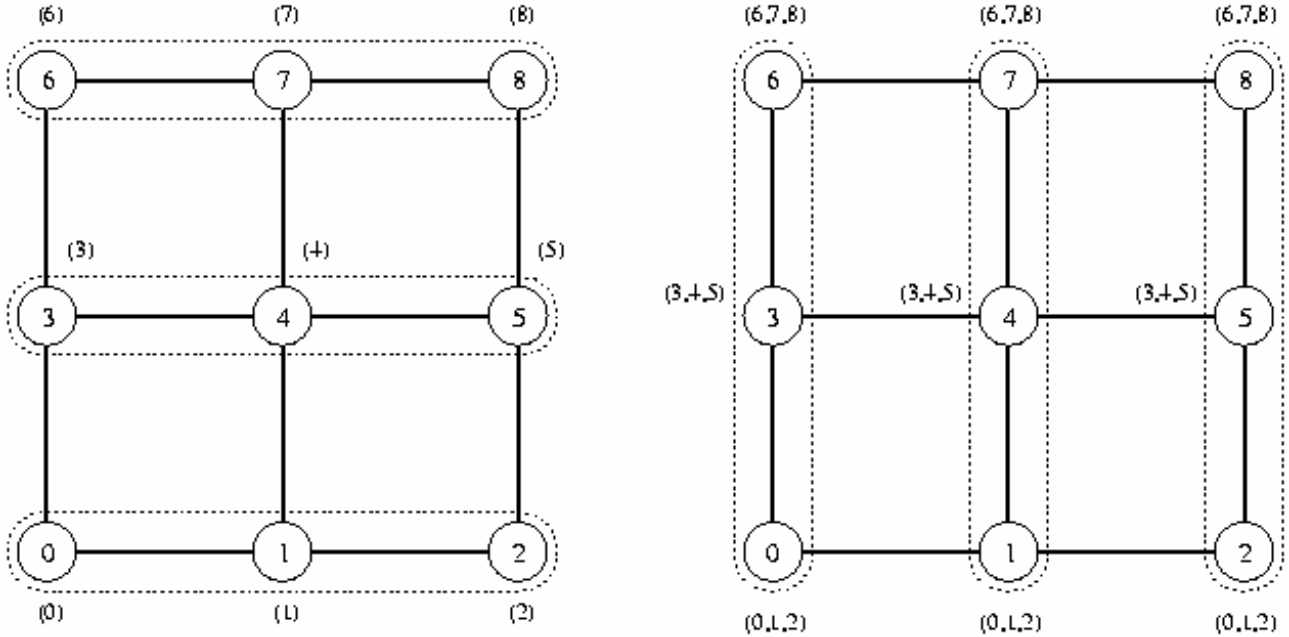
  $$t'_{пд} = (t_н + mt_к)(\sqrt{p} - 1)$$

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

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

Анализ трудоемкости основных операций передачи данных…

- **Передача данных от всех процессоров всем процессорам сети (*передача сообщений*)…**



Решетка-тор - общая длительность операции рассылки определяется соотношением

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

Анализ трудоемкости основных операций передачи данных…

- **Передача данных от всех процессоров всем процессорам сети (*передача сообщений*)**

  Для **гиперкуба** алгоритм обобщенной множественной рассылки сообщений может быть получен путем обобщения способа выполнения операции для топологии типа решетки:

  – На каждом этапе i, $1 \le i \le N$, выполнения алгоритма функционируют все процессоры сети, которые обмениваются своими данными со своими соседями по l размерности и формируют объединенные сообщения,

  – При организации взаимодействия двух соседей канал связи между ними рассматривается как связующий элемент двух равных по размеру подгиперкубов исходного гиперкуба, и каждый процессор пары посылает другому процессору только те сообщения, что предназначены для процессоров соседнего подгиперкуба.

  – Время операции рассылки может быть получено при помощи выражения:

$$t_{пд} = (t_н + \frac{1}{2} m p t_к) \log_2 p$$

Анализ трудоемкости основных операций передачи данных…

- **Передача данных от всех процессоров всем процессорам сети (*передача пакетов*)…**

  Применение более эффективного метода передачи данных для кольцевой структуры и топологии типа решетки-тора 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 времени для передачи по одной и той линии передачи имеется несколько ожидающих пересылки пакетов данных). Перегрузка каналов приводит к задержкам при пересылках данных, что и не позволяет проявиться всем преимуществам метода передачи пакетов.

Анализ трудоемкости основных операций передачи данных…

- **Передача данных от всех процессоров всем процессорам сети (*операция редукция*)…**

  Широко распространенный пример операции множественной рассылки - задача редукции (*reduction*) или, другими словами, процедура выполнения той или иной обработки данных, получаемых на каждом процессоре в ходе множественной рассылки (например, проблема вычисления суммы значений, находящихся на разных процессорах, и рассылки полученной суммы по всем процессорам сети).

Анализ трудоемкости основных операций передачи данных…

- **Передача данных от всех процессоров всем процессорам сети (*операция редукции*)**

  Способы решения задачи редукции могут состоять в следующем:

  – **непосредственный подход**: выполнение операции множественной рассылки и последующая обработка данных на каждом процессоре,
  – **более эффективный алгоритм**: операция одиночного приема данных на отдельном процессоре, выполнение на этом процессоре действий по обработке данных, и рассылка полученного результата обработки всем процессорам сети,
  – **наилучший способ** - совмещение процедуры множественной рассылки и действий по обработке данных, когда каждый процессор сразу же после приема очередного сообщения реализует требуемую обработку полученных данных. При этом время решения задачи (при топологии сети в виде гиперкуба и размере сообщения *m=1*):

$$t_{пд} = (t_н + t_к)\log_2 p$$

Анализ трудоемкости основных операций передачи данных…

- **Передача данных от всех процессоров всем процессорам сети**

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

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

  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).

Анализ трудоемкости основных операций передачи данных…

- **Обобщенная передача данных от одного процессора всем остальным процессорам сети…**

  Общий случай передачи данных от одного процессора всем остальным процессорам сети состоит в том, что все рассылаемые сообщения являются различными (*one-to-all personalized communication* or *single-node scatter*).

  Двойственная операция передачи для данного типа взаимодействия процессоров – обобщенный прием сообщений (*single-node gather*) на одном процессоре от всех остальных процессоров сети (отличие данной операции от ранее рассмотренной процедуры сборки данных на одном процессоре (*single-node accumulation*) состоит в том, что обобщенная операция сборки не предполагает какого-либо взаимодействия сообщений (типа редукции) в процессе передачи данных).

Анализ трудоемкости основных операций передачи данных…

- **Обобщенная передача данных от одного процессора всем остальным процессорам сети…**

  Трудоемкость операции обобщенной рассылки сопоставима со сложностью выполнения процедуры множественной передачи данных. Процессор-инициатор рассылки посылает каждому процессору сети сообщение размера m и, тем самым, нижняя оценка длительности выполнения операции характеризуется величиной

$$mt_к(p-1)$$

Анализ трудоемкости основных операций передачи данных…

- **Обобщенная передача данных от одного процессора всем остальным процессорам сети (*передача сообщений*)**

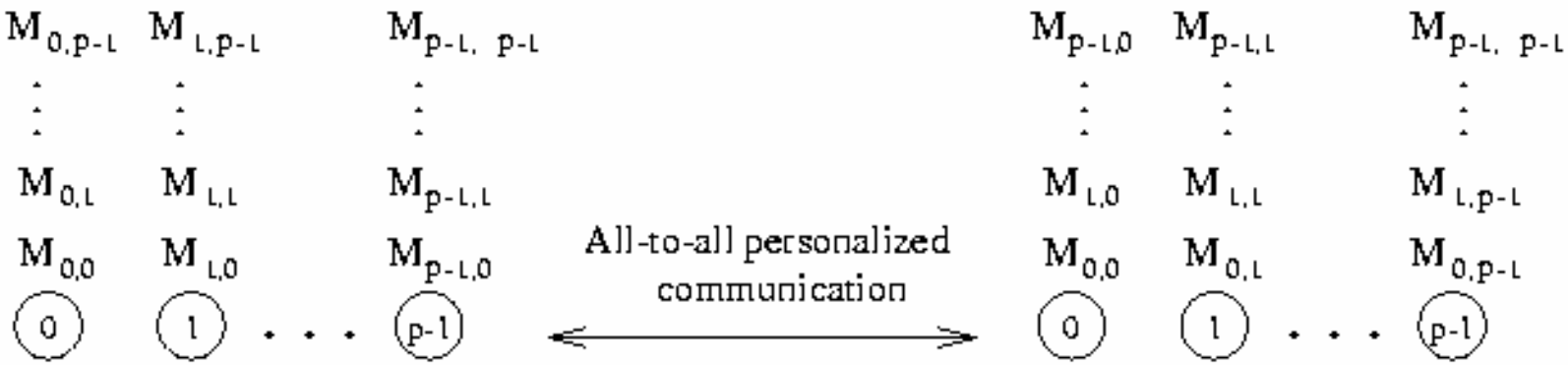
  Гиперкуб - процессор-инициатор рассылки передает половину своих сообщений одному из своих соседей (например, по первой размерности) – в результате, исходный гиперкуб становится разделенным на два гиперкуба половинного размера, в каждом из которых содержится ровно половина исходных данных. Далее действия по рассылке сообщений могут быть повторены и общее количество повторений определяется исходной размерностью гиперкуба.

$$t_{пд} = t_н \log_2 p + m t_к (p-1)$$

Анализ трудоемкости основных операций передачи данных…

- **Обобщенная передача данных от всех процессоров всем процессорам сети…**

  Обобщенная передача данных от всех процессоров всем процессорам сети (*total exchange*) представляет собой наиболее общий случай коммуникационных действий. Необходимость в выполнении подобных операций возникает в параллельных алгоритмах быстрого преобразования Фурье, транспонирования матриц и др.



Анализ трудоемкости основных операций передачи данных…

- **Обобщенная передача данных от всех процессоров всем процессорам сети (*передача сообщений*)…**

  ***Кольцевая топология*.**

  Каждый процессор производит передачу всех своих исходных сообщений своему соседу (в каком-либо выбранном направлении по кольцу). Далее процессоры осуществляют прием направленных к ним данных, затем среди принятой информации выбирают свои сообщения, после чего выполняет дальнейшую рассылку оставшейся части данных.

  Длительность выполнения подобного набора передач данных оценивается при помощи выражения:

$$t_{пд} = (t_н + \frac{1}{2} m p t_к)(p-1)$$

Анализ трудоемкости основных операций передачи данных…

- **Обобщенная передача данных от всех процессоров всем процессорам сети (*передача сообщений*)…**

  ***Решетка-тор*.**

  На первом этапе организуется передача сообщений раздельно по всем процессорам сети, располагающимся на одних и тех же горизонталях решетки (каждому процессору по горизонтали передаются только те исходные сообщения, что должны быть направлены процессорам соответствующей вертикали решетки); после завершения этапа на каждом процессоре собираются p сообщений, предназначенных для рассылки по одной из вертикалей решетки. На втором этапе рассылка данных выполняется по процессорам сети, образующим вертикали решетки.

  Общая длительность всех операций рассылок определяется соотношением $t_{пд} = (2t_н + mpt_к)(\sqrt{p} - 1)$

Анализ трудоемкости основных операций передачи данных…

- **Обобщенная передача данных от всех процессоров всем процессорам сети (*передача сообщений*)…**

  ***Гиперкуб*.**

  На каждом этапе i, $1 \le i \le N$, выполнения алгоритма функционируют все процессоры сети, которые обмениваются своими данными со своими соседями по i размерности и формируют объединенные сообщения. При организации взаимодействия двух соседей канал связи между ними рассматривается как связующий элемент двух равных по размеру подгиперкубов исходного гиперкуба, и каждый процессор пары посылает другому процессору только те сообщения, что предназначены для процессоров соседнего подгиперкуба.

  Время операции рассылки может быть получено при помощи выражения:

$$t_{пд} = (t_н + \frac{1}{2} m p t_к) \log_2 p$$

Анализ трудоемкости основных операций передачи данных…

- **Обобщенная передача данных от всех процессоров всем процессорам сети (передача пакетов)**

  Применение метода передачи пакетов не приводит к улучшению временных характеристик для операции обобщенной множественной рассылки.

  Гиперкуб. В этом случае рассылка может быть выполнена за *p-1* последовательных итераций. На каждой итерации все процессоры разбиваются на взаимодействующие пары процессоров, причем это разбиение на пары может быть выполнено таким образом, чтобы передаваемые между разными парами сообщения не использовали одни и те же пути передачи данных.

  Как результат, общая длительность операции обобщенной рассылки может быть определена в соответствии с выражением:

$$t_{пд} = (t_н + mt_к)(p-1) + \frac{1}{2} t_c p \log_2 p$$

Анализ трудоемкости основных операций передачи данных…

- **Циклический сдвиг…**

  Частный случай обобщенной множественной рассылки есть процедура перестановки (*permutation*), представляющая собой операцию перераспределения информации между процессорами сети, в которой каждый процессор передает сообщение определенному неким способом другому процессору сети. Конкретный вариант перестановки – *циклический q-сдвиг* (*cirlular q-shift*), при котором каждый процессор i, $1 \le i \le N$, передает данные процессору с номером $(i + q) \bmod p$. Подобная операция сдвига используется, например, при организации матричных вычислений.

Анализ трудоемкости основных операций передачи данных…

- **Циклический сдвиг (*передача сообщений*)…**

  ***Решетка-тор*.**

  Пусть процессоры перенумерованы по строкам решетки от *0* до *p-1*. На первом этапе организуется циклический сдвиг с шагом $q \bmod \sqrt{p}$ по каждой строке в отдельности (если при реализации такого сдвига сообщения передаются через правые границы строк, то после выполнения каждой такой передачи необходимо осуществить компенсационный сдвиг вверх на 1 для процессоров первого столбца решетки). На втором этапе реализуется циклический сдвиг вверх с шагом $\lfloor q / \sqrt{p} \rfloor$ для каждого столбца решетки.

  Общая длительность всех операций рассылок определяется соотношением

$$t_{пд} = (t_н + mt_к)(2\lfloor \sqrt{p} / 2 \rfloor + 1)$$

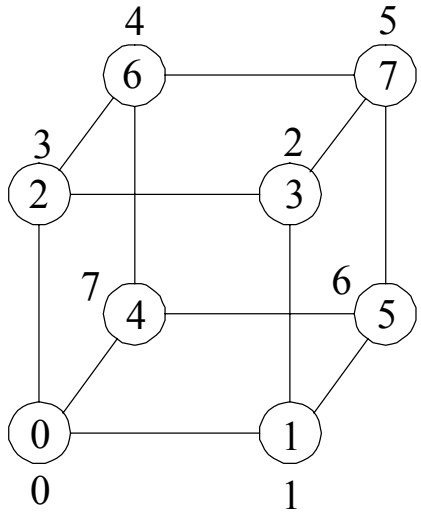
Анализ трудоемкости основных операций передачи данных…

- **Циклический сдвиг (*передача сообщений*)…**

  ***Гиперкуб*.**

  Алгоритм циклического сдвига может быть получен путем логического представления топологии гиперкуба в виде кольцевой структуры.

  Необходимое соответствие может быть получено, например, при помощи известного кода Грея, который можно использовать для определения процессоров гиперкуба, соответствующих конкретным вершинам кольца.



Анализ трудоемкости основных операций передачи данных…

- **Циклический сдвиг (*передача сообщений*)**

  ***Гиперкуб*.**

  Представим величину сдвига q в виде двоичного кода. Количество ненулевых позиций кода определяет количество этапов в схеме реализации операции циклического сдвига.

  На каждом этапе выполняется операция сдвига с величиной шага, определяемой наиболее старшей ненулевой позицией значения q (например, при исходной величине сдвига $q=5=101_2$, на первом этапе выполняется сдвиг с шагом 4, на втором этапе шаг сдвига равен 1). Выполнение каждого этапа (кроме сдвига с шагом 1) состоит в передаче данных по пути, включающему две линии связи.

  Как результат, верхняя оценка для длительности выполнения операции циклического сдвига определяется соотношением:

$$t_{пд} = (t_н + mt_к)(2\log_2 p - 1)$$

Анализ трудоемкости основных операций передачи данных

- **Циклический сдвиг (*передача пакетов*)**

  Использование пересылки пакетов может повысить эффективность выполнения операции циклического сдвига для топологии **гиперкуб**.

  Реализация всех необходимых коммуникационных действий в этом случае может быть обеспечена путем отправления каждым процессором всех пересылаемых данных непосредственно процессорам назначения.

  Использование метода покоординатной маршрутизации позволит избежать коллизий при использовании линий передачи данных.

  Длина наибольшего пути при такой рассылке данных определяется как $log_2 p - \gamma(p)$, где $\gamma(p)$ есть наибольшее целое значение j такое, что 2^j есть делитель величины сдвига q.

  Длительность операции циклического сдвига может быть определена при помощи выражения:

$$t_{пд} = t_н + mt_к + t_c(\log_2 p - \gamma(q))$$

Методы логического представления топологии коммуникационной среды…

- Ряд алгоритмов передачи данных допускает более простое изложение при использовании вполне определенных топологий сети межпроцессорных соединений
- Многие методы коммуникации могут быть получены при помощи того или иного логического представления исследуемой топологии.

*Важным моментом при организации параллельных вычислений является возможность **логического представления** разнообразных топологий на основе имеющихся (физических) межпроцессорных структур*

Методы логического представления топологии коммуникационной среды…

- Способы логического представления (отображения) топологий характеризуются следующими тремя основными характеристиками:
  - **уплотнение дуг** (*congestion*), выражаемое как максимальное количество дуг логической топологии, отображаемых в одну линию передачи физической топологии,
  - **удлинение дуг** (*dilation*), определяемое как путь максимальной длины физической топологии, на который отображаемая дуга логической топологии,
  - **увеличение вершин** (*expansion*), вычисляемое как отношение количества вершин в логической и физической топологиях.

Методы логического представления топологии коммуникационной среды…

- **Представление кольцевой топологии в виде гиперкуба…**

  Установление соответствия между кольцевой топологией и гиперкубом может быть выполнено при помощи *двоичного рефлексивного кода Грея* G(i,N) (*binary reflected Gray code*)

$$G(0,1)=0, \quad G(1,1)=1, \quad G(i,s+1)=\begin{cases} G(i,s), & i<2^s, \\ 2^s+G(2^{s+1}-1-i,s), & i\geq 2^s, \end{cases}$$

  i задает номер значения в коде Грея, а *N* есть длина этого кода.

Методы логического представления топологии коммуникационной среды…

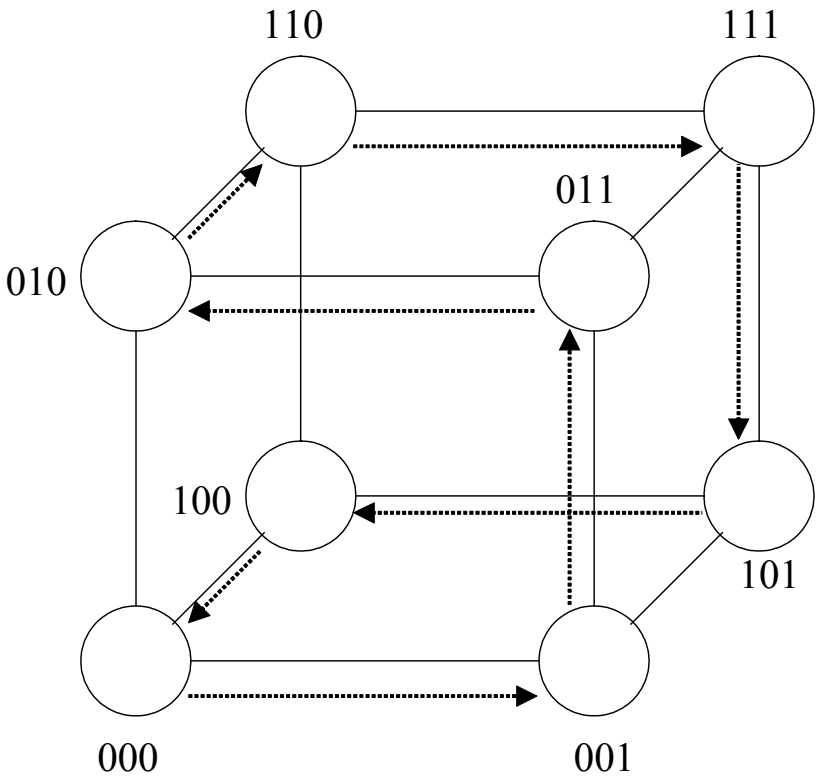
- **Представление кольцевой топологии в виде гиперкуба…**

  Отображение кольцевой топологии на гиперкуб для сети из *p=8* процессоров:

Код Грея для N=1	Код Грея для N=2	Код Грея для N=3	Номера процессоров	
			гиперкуба	кольца
0	0 0	0 0 0	0	0
1	0 1	0 0 1	1	1
	1 1	0 1 1	3	2
	1 0	0 1 0	2	3
		1 1 0	6	4
		1 1 1	7	5
		1 0 1	5	6
		1 0 0	4	7

Методы логического представления топологии коммуникационной среды…

- **Представление кольцевой топологии в виде гиперкуба…**



Важным свойством кода Грея является тот факт, что соседние значения *G(i,N)* и *G(i+1,N)* имеют только одну различающуюся битовую позицию. Как результат, соседние вершины в кольцевой топологии отображаются на соседние процессоры в гиперкубе.

Методы логического представления топологии коммуникационной среды

- **Представление топологии решетки в виде гиперкуба**

  Отображение топологии решетки на гиперкуб может быть выполнено в рамках подхода, использованного для кольцевой структуры сети. Тогда для отображения решетки на гиперкуб размерности $N=r+s$ можно принять правило, что элементу решетки с координатами (i,j), будет соответствовать процессор гиперкуба с номером

$$G(i,r)\,||\,G(j,s),$$

  где операция $||$ означает конкатенацию кодов Грея.

Оценка трудоемкости операций передачи данных для кластерных систем…

- Для кластерных вычислительных систем одним из широко применяемых способов построения коммуникационной среды является использование концентраторов (*hub*) или переключателей (*switch*).
- В этих случаях топология сети кластера представляет собой *полный граф*, в котором имеются определенные ограничения на одновременность выполнения коммуникационных операций:
  - При использовании концентраторов передача данных в каждый текущий момент времени может выполняться только между двумя процессорными узлами,
  - Переключатели могут обеспечивать одновременное взаимодействие нескольких непересекающихся пар процессоров.
- В качестве основного способа выполнения коммуникационных операций используется *метод передачи пакетов* (как правило, на основе протокола TCP/IP).

Оценка трудоемкости операций передачи данных для кластерных систем…

- Трудоемкость операции коммуникации между двумя процессорными узлами может быть оценена в соответствии с выражением (***модель А***):

$$t_{пд}(m) = t_н + m * t_k + t_c$$

- Замечания:
  - время подготовки данных предполагается постоянным (не зависящим от объема передаваемых данных),
  - время передачи служебных данных 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.

Оценка трудоемкости операций передачи данных для кластерных систем…

- Уточнение модели (***модель B***):

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

$n=[m/(V_{max}-V_c)]$ - количество пакетов, на которое разбивается передаваемое сообщение,

V_{max} - максимальный размер пакета, который может быть доставлен в сети,

V_c - объем служебных данных в каждом из пересылаемых пакетов,

$t_{нач0}$ - аппаратная составляющая латентности,

$t_{нач1}$ - время подготовки одного байта данных для передачи по сети.

Латентность, тем самым, не превышает величины:

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

Оценка трудоемкости операций передачи данных для кластерных систем…

- Для практического применения перечисленных моделей необходимо выполнить оценку значений параметров используемых соотношений
- Более простой способ вычисления временных затрат на передачу данных – *модель Хокни* (*Hockney*), в которой трудоемкость операции коммуникации между двумя процессорными узлами кластера оценивается в соответствии с выражением (***модель С***):

$$t_{пд}(m) = t_н + m t_к$$

Оценка трудоемкости операций передачи данных для кластерных систем…

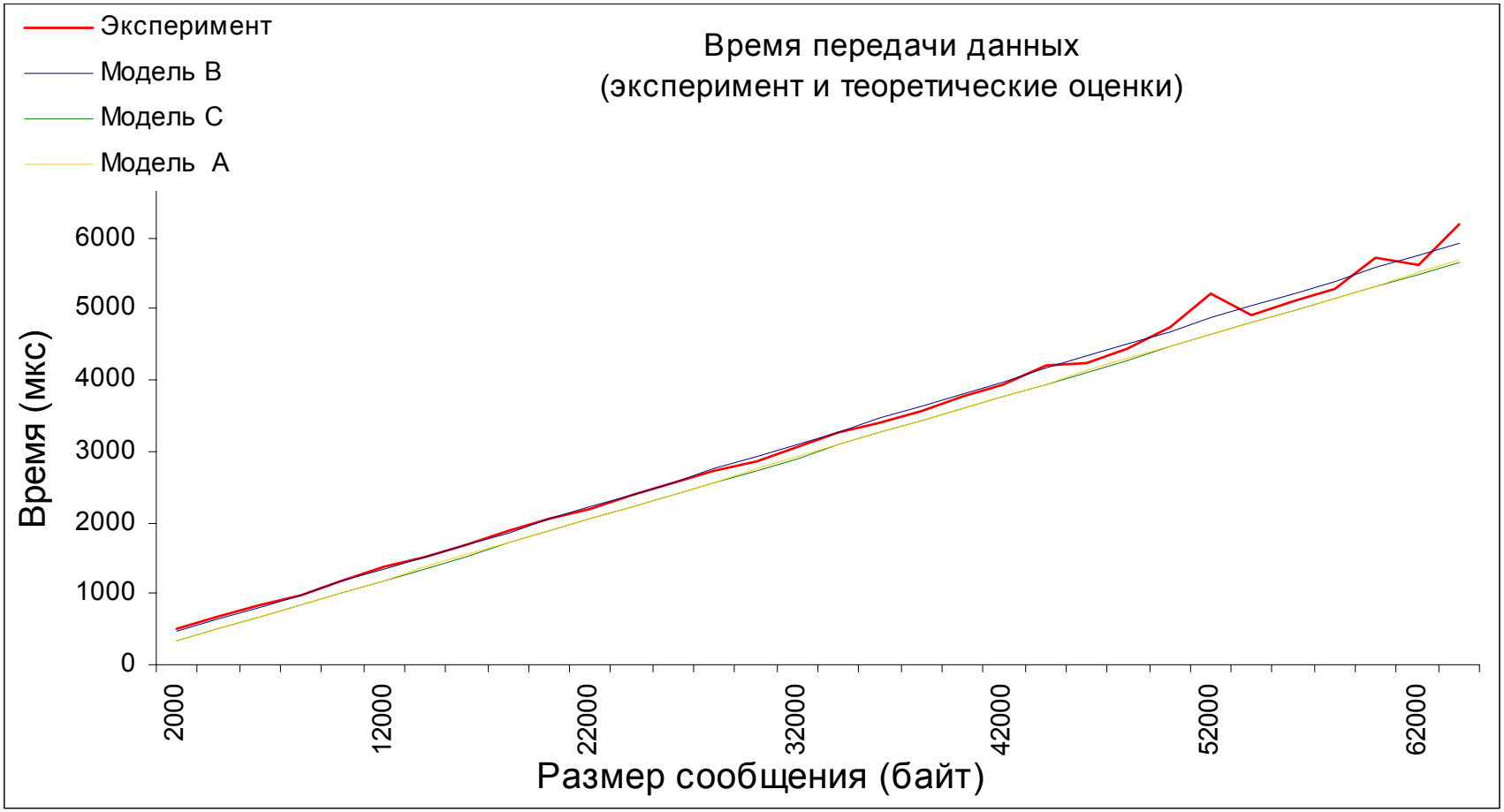
- Описание вычислительных экспериментов
  - Эксперименты выполнялись в сети многопроцессорного кластера Нижегородского университета (компьютеры IBM PC Pentium 4 1300 Мгц и сеть Fast Etherrnet). При проведении экспериментов для реализации коммуникационных операций использовалась библиотека MPI,
  - Значение латентности для моделей *A* и *C* определялось как время передачи сообщения нулевой длины,
  - Величина пропускной способности *R* устанавливалась максимально наблюденной в ходе экспериментов скорости передачи данных, т.е.

$$R = \max_{m} \quad (t_{пд}(m)/m)$$

    и полагалось $t_к = 1/R$,
  - значения величин $t_{нач0}$ и $t_{нач1}$ оценивались при помощи линейной аппроксимации времен передачи сообщений размера от 0 до V_{max}.

Оценка трудоемкости операций передачи данных для кластерных систем…

- Результаты вычислительных экспериментов…



Оценка трудоемкости операций передачи данных для кластерных систем…

- Результаты вычислительных экспериментов

Объем сообщения (байт)	Время передачи (мкс)	Погрешность теоретической оценки времени передачи данных, в %		
		Модель A	Модель B	Модель C
2000	495	33.45%	7.93%	34.80%
10000	1184	13.91%	1.70%	14.48%
20000	2055	8.44%	0.44%	8.77%
30000	2874	4.53%	-1.87%	4.76%
40000	3758	4.04%	-1.38%	4.22%
50000	4749	5.91%	1.21%	6.05%
60000	5730	6.97%	2.73%	7.09%

Оценка трудоемкости операций передачи данных для кластерных систем

- Как можно заметить по результатам проведенных экспериментов, оценки трудоемкости операций передачи данных по модели *B* имеют меньшую погрешность
- Вместе с этим важно отметить, что для предварительного анализа временных затрат на выполнение коммуникационных операций точности модели *C* может оказаться достаточно. Кроме того, данная модель носит наиболее простой вид среди всех рассмотренных моделей.
- С учетом последнего обстоятельства, далее во всех последующих разделах для оценки трудоемкости операций передачи данных будет применяться именно модель *C* (модель Хокни); при этом для модели будет использоваться форма записи

$$t_{пд}(m) = \alpha + m / \beta,$$

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

Заключение…

- Представлена общая характеристика алгоритмов маршрутизации и методов передачи данных. Для подробного рассмотрения выделены метод передачи сообщений (*store-and-forward routing*) и метод передачи пакетов (*cut-through routing*), для которых определены оценки времени выполнения коммуникационных операций.
- Определены основные типы операций передачи данных, выполняемых в ходе параллельных вычислений. Для всех операций рассмотрены алгоритмы их выполнения на примере топологий кольца, решетки и гиперкуба. Приведены оценки их временной трудоемкости как для метода передачи сообщений, так и для метода передачи пакетов.

Заключение

- Рассмотрены *методы логического представления топологий* на основе конкретных (физических) межпроцессорных структур.
- Проведен анализ моделей, при помощи которых могут быть получены оценки времени выполнения операций передачи данных для кластерных вычислительных систем. По результатам сравнения для дальнейшего использования при оценке временной трудоемкости коммуникационных операций выбрана наиболее простая модель - *модель Хокни*.

Вопросы для обсуждения

- Оценка разных методов передачи данных
- Возможные типовые операции передачи данных
- Полезность использования логических топологий
- Достаточность рассмотренного множества типовых операций передачи

Темы заданий для самостоятельной работы

- Разработайте алгоритмы выполнения основных операций передачи данных для топологии сети в виде 3-мерной решетки.
- Разработайте алгоритмы выполнения основных операций передачи данных для топологии сети в виде двоичного дерева.
- Примените модель *B* из подраздела 3.4 для оценки временной сложности операций передачи данных. Сравните получаемые показатели.
- Примените модель *C* из подраздела 3.4 для оценки временной сложности операций передачи данных. Сравните получаемые показатели.
- Разработайте алгоритмы логического представления двоичного дерева для различных физических топологий сети.

Литература…

- **Гергель, В.П., Стронгин, Р.Г.** (2001). Основы параллельных вычислений для многопроцессорных вычислительных систем. - Н.Новгород, ННГУ (2 изд., 2003).
- **Andrews**, G. R. (2000). Foundations of Multithreaded, Parallel, and Distributed Programming.. – Reading, MA: Addison-Wesley (русский перевод Эндрюс Г.Р. Основы многопоточного, параллельного и распределенного программирования. – М.: Издательский дом "Вильямс", 2003).
- **Hockney**, R. W., Jesshope, C.R. (1988). Parallel Computers 2. Architecture, Programming and Algorithms. - Adam Hilger, Bristol and Philadelphia. (русский перевод 1 издания: Хокни Р., Джессхоуп К. Параллельные ЭВМ. Архитектура, программирование и алгоритмы. - М.: Радио и связь, 1986)

Литература

- **Culler**, D., Singh, J.P., Gupta, A. (1998) Parallel Computer Architecture: A Hardware/Software Approach. - Morgan Kaufmann.
- **Kumar** V., Grama, A., Gupta, A., Karypis, G. (1994). Introduction to Parallel Computing. - The Benjamin/Cummings Publishing Company, Inc. (2nd edn., 2003)
- **Quinn**, M. J. (2004). Parallel Programming in C with MPI and OpenMP. – New York, NY: McGraw-Hill.
- **Skillicorn**, D.B., Talia, D. (1998). Models and languages for parallel computation. – ACM Computing surveys, 30, 2.

Следующая тема

- **Параллельное программирование на основе MPI**

Авторский коллектив

Гергель В.П., профессор, д.т.н., руководитель

Гришагин В.А., доцент, к.ф.м.н.

Абросимова О.Н., ассистент (раздел 10)

Лабутин Д.Ю., ассистент (система ПараЛаб)

Курылев А.Л., ассистент (лабораторные работы 4, 5)

Сысоев А.В., ассистент (раздел 1)

Гергель А.В., аспирант (раздел 12, лабораторная работа 6)

Лабутина А.А., аспирант (разделы 7,8,9, лабораторные работы 1, 2, 3, система ПараЛаб)

Сенин А.В., аспирант (раздел 11, лабораторные работы по Microsoft Compute Cluster)

Ливерко С.В. (система ПараЛаб)

О проекте

Целью проекта является создание образовательного комплекса "Многопроцессорные вычислительные системы и параллельное программирование", обеспечивающий рассмотрение вопросов параллельных вычислений, предусматриваемых рекомендациями Computing Curricula 2001 Международных организаций IEEE-CS и ACM. Данный образовательный комплекс может быть использован для обучения на начальном этапе подготовки специалистов в области информатики, вычислительной техники и информационных технологий.

Образовательный комплекс включает **учебный курс "Введение в методы параллельного программирования" и лабораторный практикум "Методы и технологии разработки параллельных программ"**, что позволяет органично сочетать фундаментальное образование в области программирования и практическое обучение методам разработки масштабного программного обеспечения для решения сложных вычислительно-трудоемких задач на высокопроизводительных вычислительных системах.

Проект выполнялся в Нижегородском государственном университете им. Н.И. Лобачевского на кафедре математического обеспечения ЭВМ факультета вычислительной математики и кибернетики (http://www.software.unn.ac.ru). Выполнение проекта осуществлялось при поддержке компании Microsoft.