Проблемы и методы организации эффективного решения параллельных задач с учетом структур распределенных вычислительных систем

Курносов Михаил Георгиевич

mkurnosov@isp.nsc.ru

Лаборатория вычислительных систем Института физики полупроводников им. А.В. Ржанова СО РАН, Новосибирск

Кафедра вычислительных систем ГОУ ВПО “СибГУТИ”, Новосибирск

XIII Российская конференция с участием иностранных ученых “Распределенные информационные и вычислительные ресурсы” (DICR-2010) Новосибирск, Россия, 20 ноября – 3 декабря 2010 г.

Средства создания параллельных программ для распределенных вычислительных систем

Библиотеки передачи сообщений

- Библиотеки стандарта MPI: MPICH2, Open MPI
- Parallel Virtual Machine

Языки параллельного программирования для распределенных ВС

- PGAS: IBM X10, Cray Chapel
- Charm++, High-Performance Fortran, Unified Parallel C

Модель передачи сообщений (message passing)

Дифференцированный обмен
(point-to-point communication)

Коллективные обмены
(collective communication)

Трансляционный обмен
(ТО, one-to-all broadcast)

Трансляционно-циклический обмен
(ТЦО, all-to-all broadcast)

Коллекторный обмен
(КО, all-to-one broadcast)

На коллективные обмены приходится 70-80% времени всех обменов (R. Rabenseifner, 2000)

Задачи эффективной реализации параллельных программ на распределенных ВС

- **Минимизация использования в runtime-системах структур данных с пространственной сложностью выше $O(\log_2 n)$, где n – количество процессорных ядер в системе.**
  - Использование распределенных таблиц номеров процессов (например, при создании коммуникаторов в библиотеках стандарта MPI).
  - Распределенное формирование графов логических топологий (MPI Virtual Topologies).

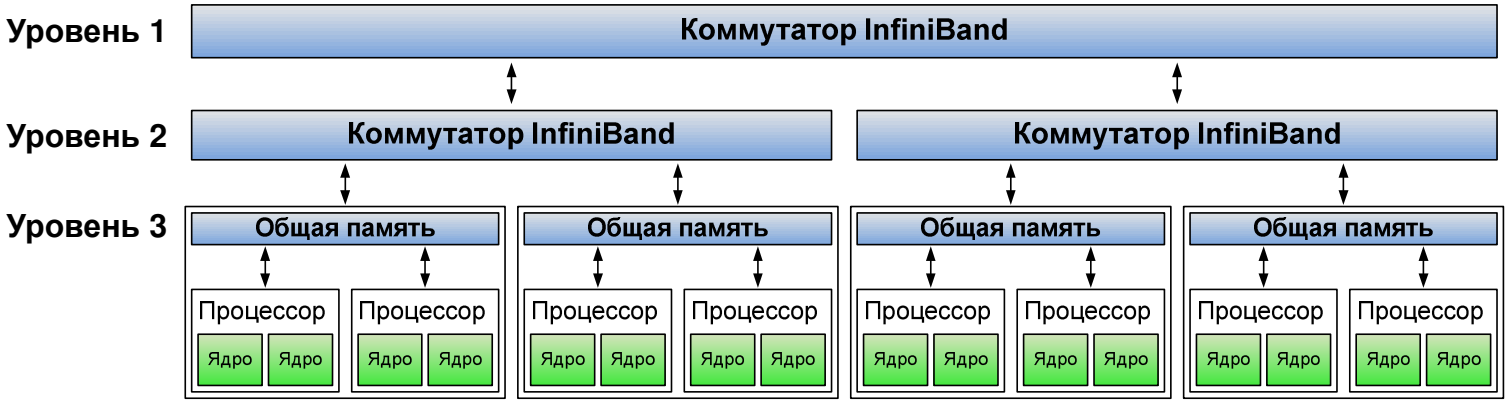
- **Обеспечение (суб)оптимального вложения (Process Mapping) в структуру распределенной ВС графа информационных обменов между ветвями параллельной программы.**
  - Применение параллельных алгоритмов вложения программ в структуры мультиархитектурных распределенных систем.

- **Реализация коллективных обменов (Collective Communication) информацией между ветвями параллельной программы с учетом структуры распределенной ВС.**
  - Организация структурно-ориентированных коллективных обменов информацией между ветвями параллельных программ.

Алгоритмы коллективных обменов информацией между ветвями параллельных программ

- **Трансляционный обмен (MPI_Bcast, MPI_Scatter):** биномиальное дерево, Scatter + Allgather.
- **Трансляционно-циклический обмен (MPI_Allgather, MPI_Alltoall, MPI_Allreduce):** кольцевой алгоритм, рекурсивное сдваивание, алгоритм Дж. Брука, алгоритм попарных обменов.
- **Коллекторный обмен (MPI_Reduce, MPI_Gather):** биномиальное дерево, Reduce-Scatter + Gather.

Алгоритмы основаны на предположении об однородности каналов связи между элементарными машинами распределенной ВС



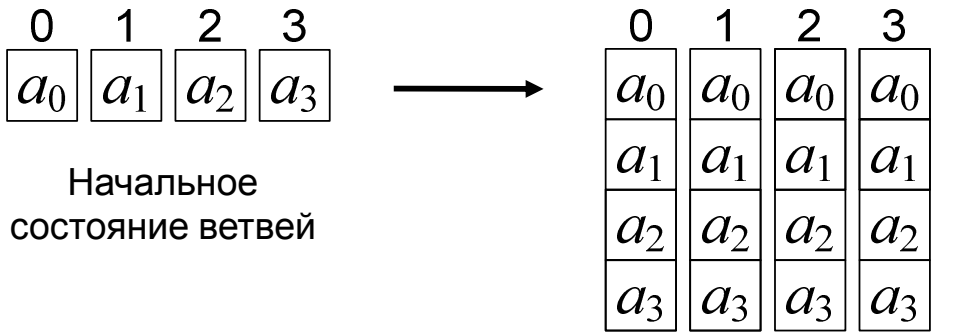
Иерархическая организация распределенных ВС

Трансляционно-циклический обмен (ТЦО) информацией между ветвями параллельных программ

Каждая ветвь $i \in \{0, 1, \ldots, n-1\}$ параллельной программы располагает локальным сообщением a_i размером m байт.

Требуется отправить сообщение a_i всем ветвям и получить сообщения от каждой.

По окончанию ТЦО все ветви должны содержать в своей памяти n сообщений $a_0, a_1, \ldots, a_{n-1}$, упорядоченных по номерам ветвей.



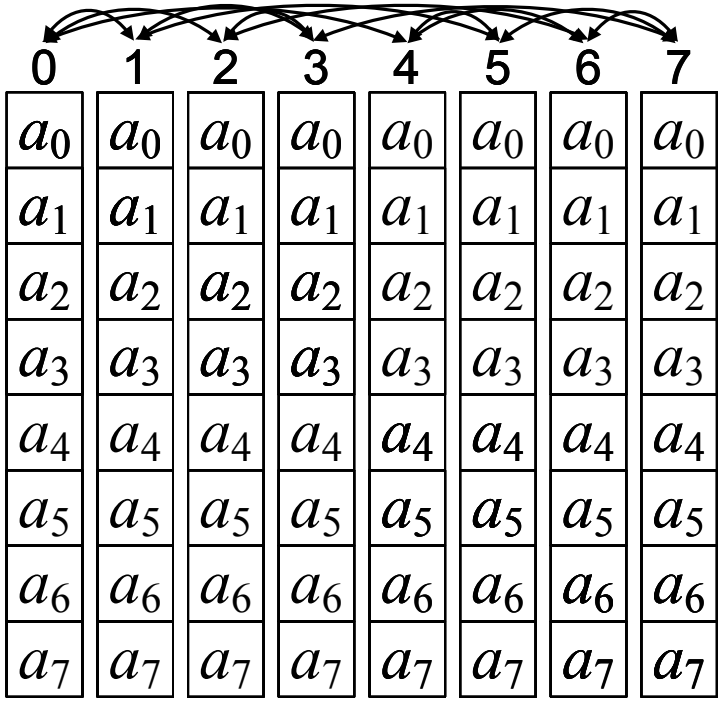
Начальное состояние ветвей

Состояние ветвей после реализации ТЦО

- MPI: `MPI_Allgather`
- Unified Parallel C: `upc_all_gather_all`

Алгоритмы реализации трансляционно-циклических обменов информацией в распределенных ВС

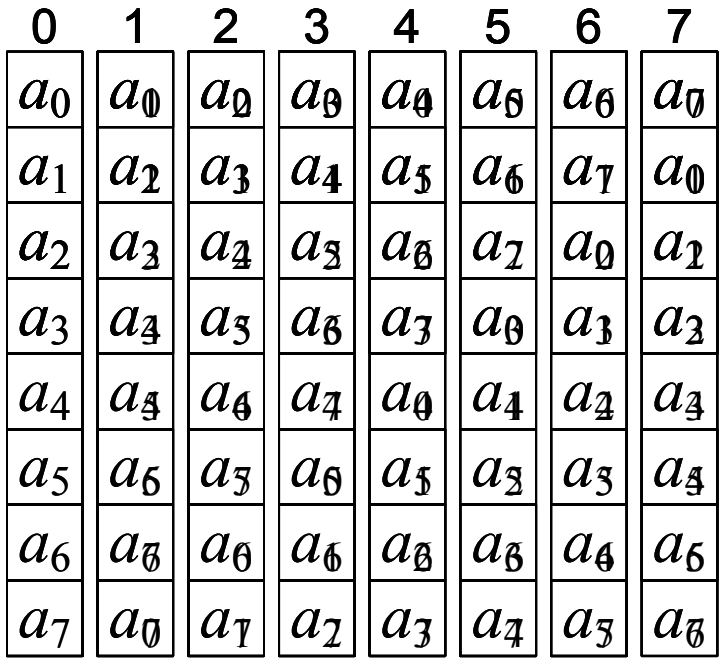
Алгоритм рекурсивного сдваивания (recursive doubling)



Количество обменов: $2\log_2 n$

Только для n равного степени двойки

Алгоритм Дж. Брука (J. Bruck et al., 1997)

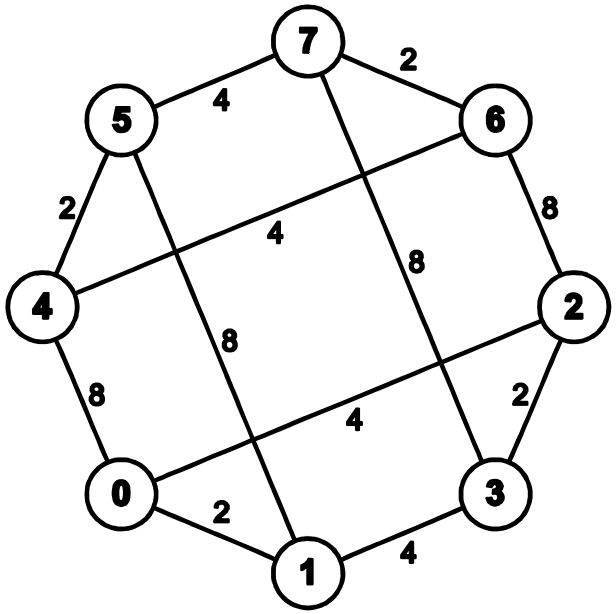


Количество обменов: $2\lceil \log_2 n \rceil$

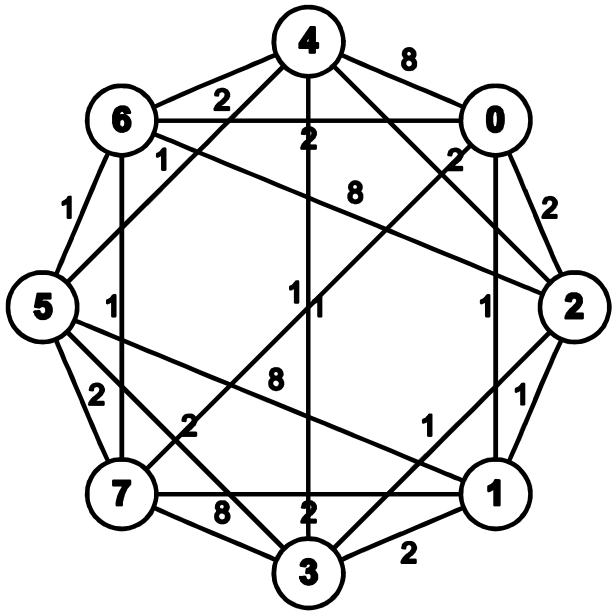
На шаге k ветвь i взаимодействует с ветвями $(i - 2^k + n) \bmod n$ и $(i + 2^k) \bmod n$

На каждом шаге размер передаваемого блока удваивается: $m, 2m, 4m$

Информационные графы алгоритмов реализации трансляционно-циклических обменов



Граф алгоритма
рекурсивного сдваивания, m = 1



Граф алгоритма
Дж. Брука, m = 1

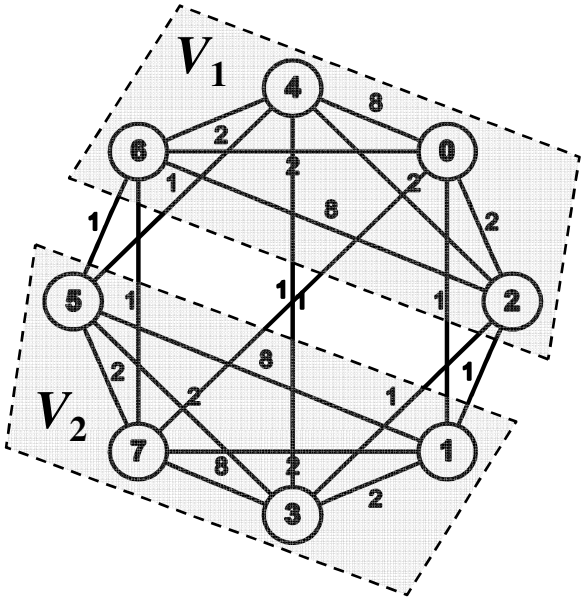
Вес d_{ij} ребра отражает объем данных переданных по нему при реализации алгоритма

Метод оптимизации алгоритмов трансляционно-циклических обменов в иерархических распределенных ВС

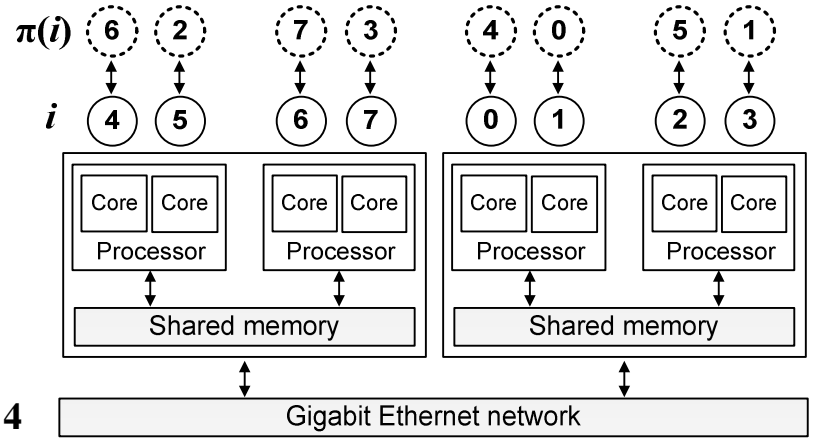
1. **Формируется взвешенный граф** $G = (V, E)$, $|V| = n$ алгоритма реализации ТЦО для $m = 1$.

2. **Строится разбиение графа на h непересекающихся подмножеств** $V_1, V_2, \ldots, V_h$ с целью минимизации суммарного веса рёбер, инцидентных различным подмножествам разбиения.

   Количество элементов в подмножестве V_u должно быть равно заданному числу s_u, $u = 1, 2, \ldots, h$.



3. **Строится отображение** $\pi(i)$ – ветвям с номерами из подмножества V_u назначаются номера ветвей, распределенных на элементарные машины вычислительного узла q_u.



$h = 2$

$s_1 = s_2 = 4$

Алгоритмы реализации трансляционно-циклических обменов в иерархических распределенных ВС

Алгоритм recursive doubling exch.

1. Ветвь i передает свое сообщение a_i ветви $\pi^{-1}(i)$, принимает от ветви $\pi(i)$ сообщение и делает его начальным.

2. На шаге $k = 0, 1, \ldots, \log_2 n - 1$ ветви i и $\pi^{-1}(i \oplus 2^k)$ обмениваются ранее принятыми 2^k сообщениями.

Алгоритм recursive doubling reorder

1. Ветвь i сдвигает свое сообщение a_i из позиции i в позицию $\pi(i)$.

2. На шаге $k = 0, 1, \ldots, \log_2 n - 1$ ветви i и $\pi^{-1}(i \oplus 2^k)$ обмениваются ранее принятыми 2^k сообщениями.

3. Сообщение из позиции $j = 0, 1, \ldots, n - 1$ переносится в позицию $\pi^{-1}(j)$.

Алгоритм Bruck exch.

1. Ветвь i передает свое сообщение a_i ветви $\pi^{-1}(i)$, принимает от ветви $\pi(i)$ сообщение и делает его начальным.

2. На шаге $k = 0, 1, \ldots, \lceil \log_2 n \rceil - 1$ ветвь i передает все принятые сообщения ветви $\pi^{-1}((i' - 2^k + n) \bmod n)$ и принимает сообщения от ветви $\pi^{-1}((i' + 2^k) \bmod n)$, где $i' = \pi(i)$.

3. Каждая ветвь циклически сдвигает сообщения вниз на i' позиций.

Алгоритм Bruck reorder

1. На шаге $k = 0, 1, \ldots, \lceil \log_2 n \rceil - 1$ ветвь i передает все принятые сообщения ветви $\pi^{-1}((i' - 2^k + n) \bmod n)$ и принимает сообщения от ветви $\pi^{-1}((i' + 2^k) \bmod n)$, где $i' = \pi(i)$.

2. Сообщение в позиции $j = 0, 1, \ldots, n - 1$ переставляется в позицию $\pi^{-1}((j + i') \bmod n)$.

Метод оптимизации трансляционно-циклических обменов информацией в иерархических распределенных ВС

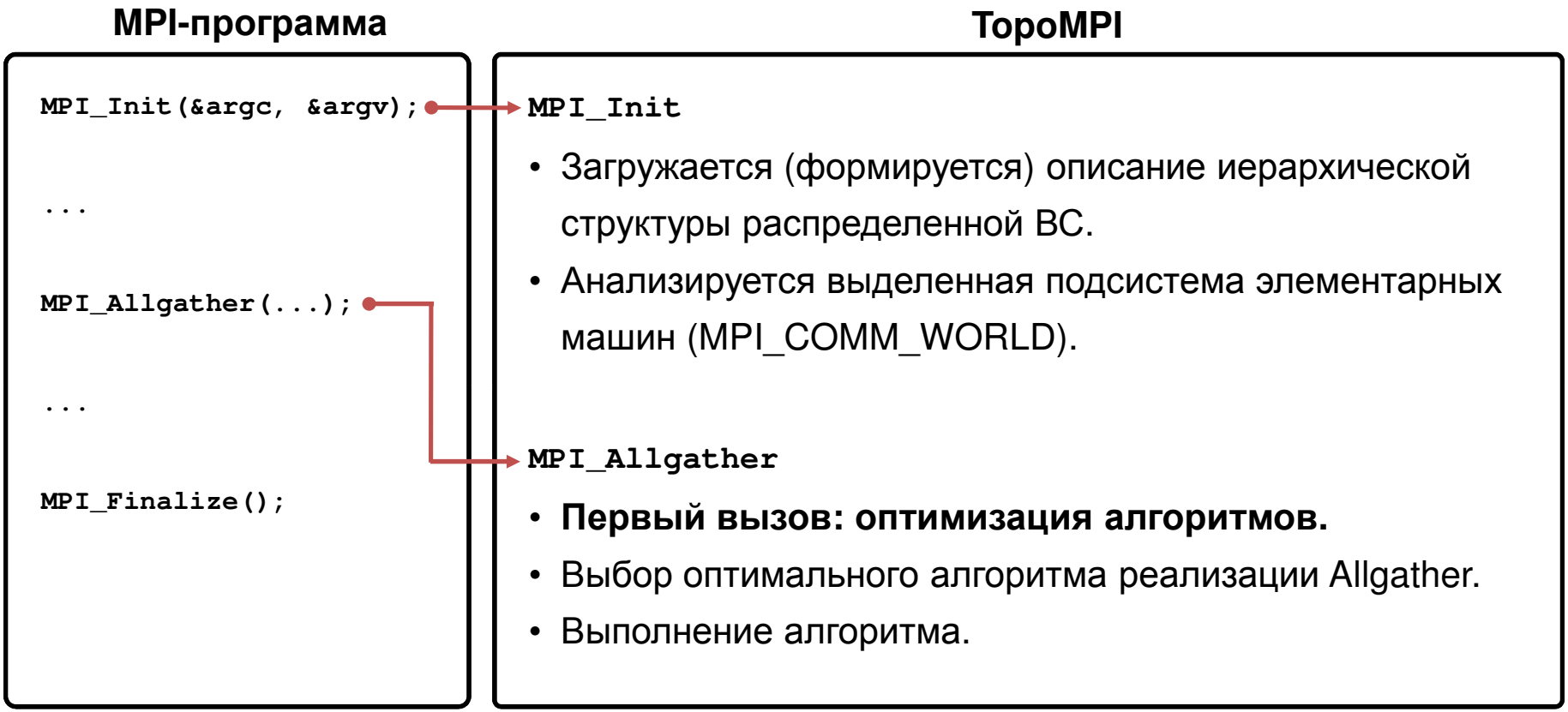
- Формирование, разбиение графа и построение отображений $\pi(i)$ и $\pi^{-1}(i)$ осуществляется **единожды** при первом вызове коллективной операции.

- **Все шаги метода эффективно реализуемы в параллельном виде** на подсистеме элементарных машин, выделенной для реализации программы.

- В каждой ветви **достаточно хранить информацию об отображении смежных с ними ветвей**. Для алгоритмов рекурсивного сдваивания и Дж. Брука достаточно хранить порядка $O(\log_2 n)$ байт.

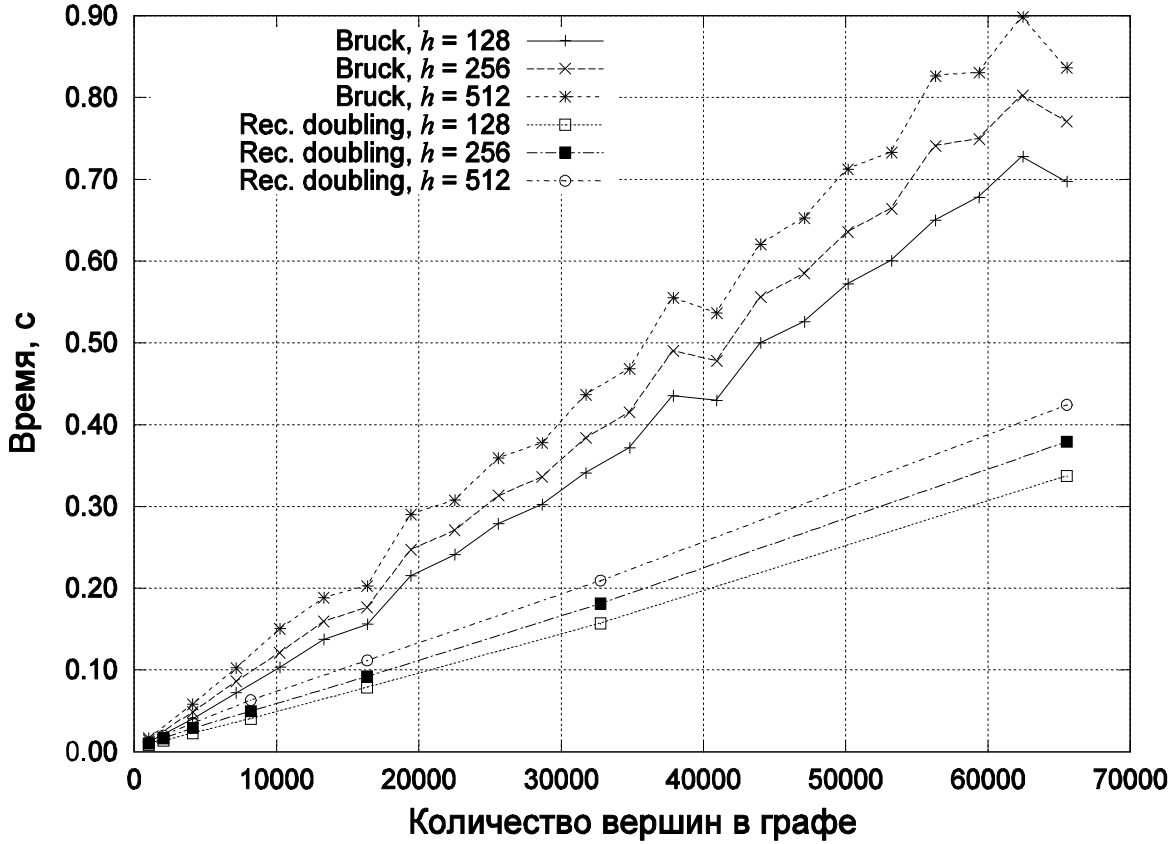
Предложенный метод реализации ТЦО применим в большемасштабных распределенных ВС

Библиотека коммуникационных функций TopoMPI



http://topompi.cpct.sibsutis.ru

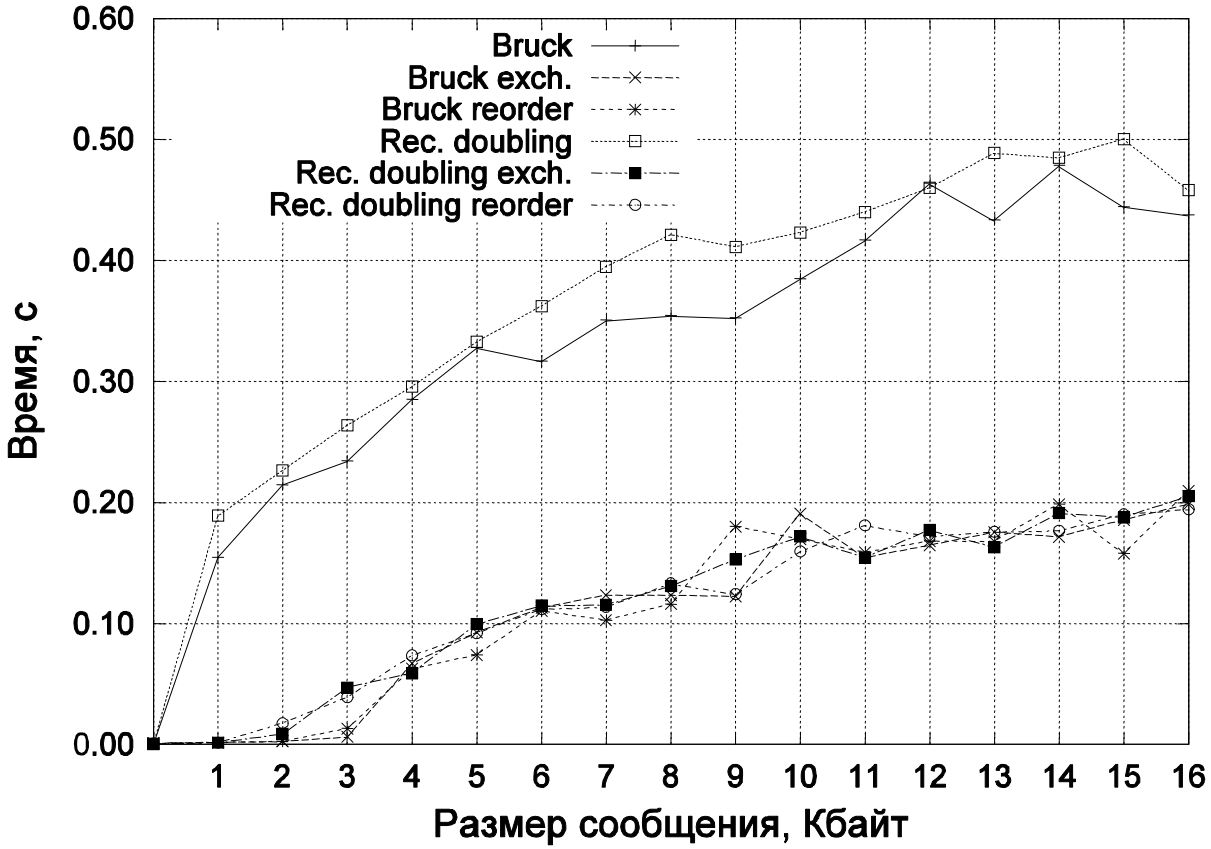
Результаты экспериментов



Время разбиения информационных графов алгоритмов на h подмножеств и формирования отображений на процессоре Intel Xeon E5420 (2.5 GHz)

Время разбиения графа алгоритма Дж. Брука с количеством вершин 1048576 на 131072 подмножеств: 32,73 с.

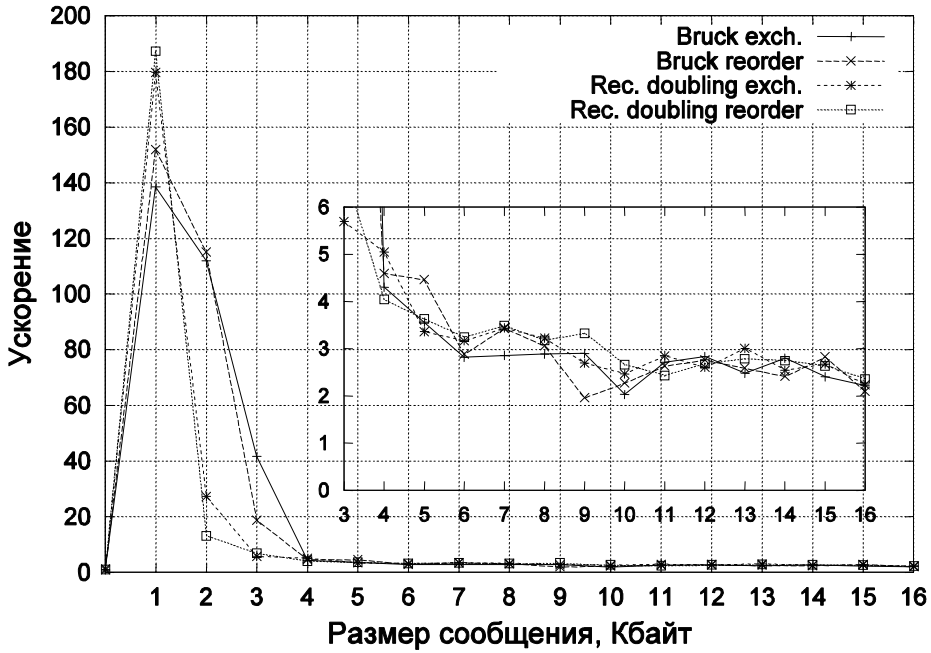
Результаты экспериментов



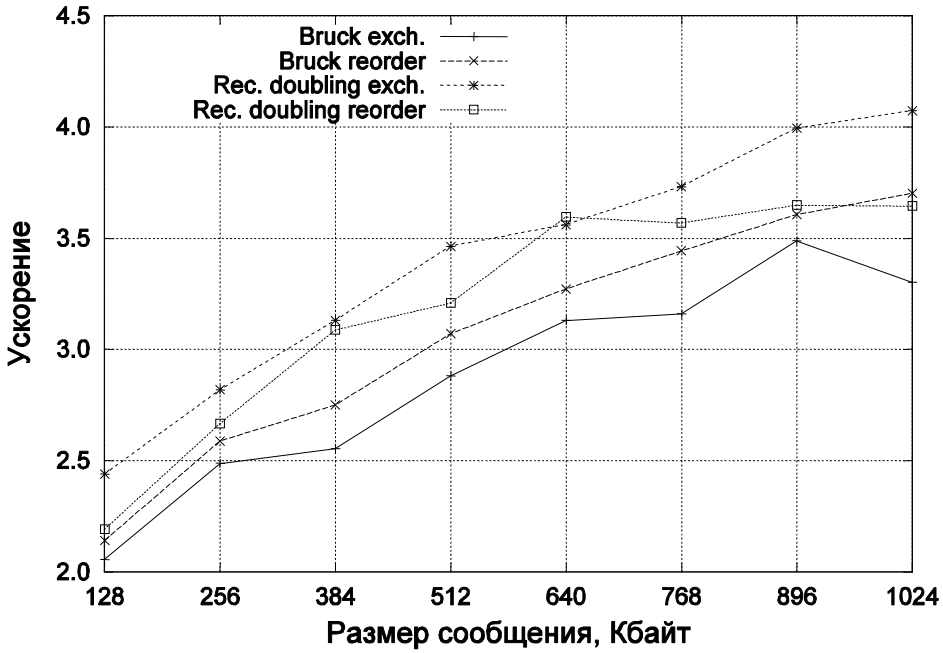
Время реализации ТЦО с учетом времени оптимизации алгоритмов:
8 двухпроцессорных узлов по 8 процессорных ядер; сеть связи Gigabit Ethernet

Среднее время оптимизации алгоритмов составило 480 мкс и 300 мкс, соответственно,
для алгоритмов Bruck exch., Bruck reorder и recursive doubling exch., recursive doubling reorder

Результаты экспериментов



а

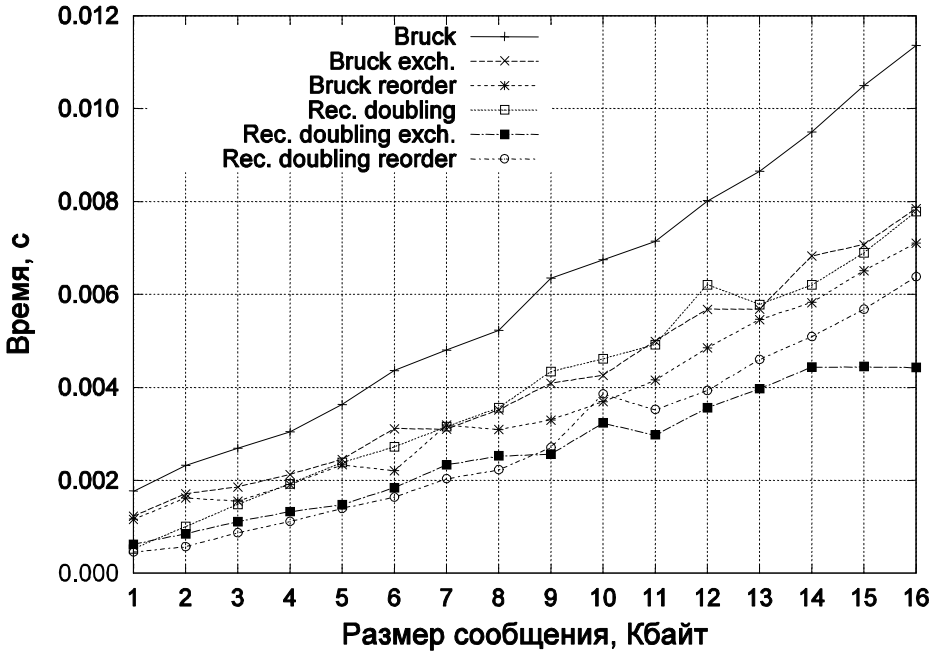


б

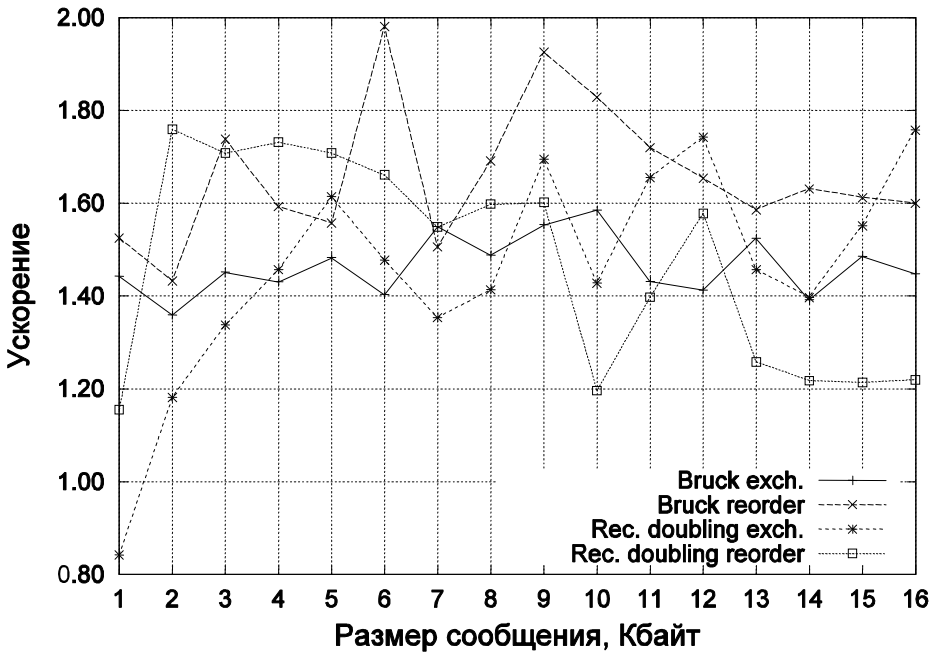
Ускорение алгоритмов на кластерной ВС:
8 двухпроцессорных узлов по 8 процессорных ядер; сеть связи Gigabit Ethernet

а – ускорение алгоритмов на “коротких” сообщениях;
б – ускорение алгоритмов на “длинных” сообщениях

Результаты экспериментов



а



б

Ускорение алгоритмов на кластерной ВС:
8 двухпроцессорных узлов по 8 процессорных ядер; сеть связи InfiniBand 4x DDR

а – время реализации ТЦО “коротких” сообщений;
б – ускорение алгоритмов на “длинных” сообщениях

Заключение

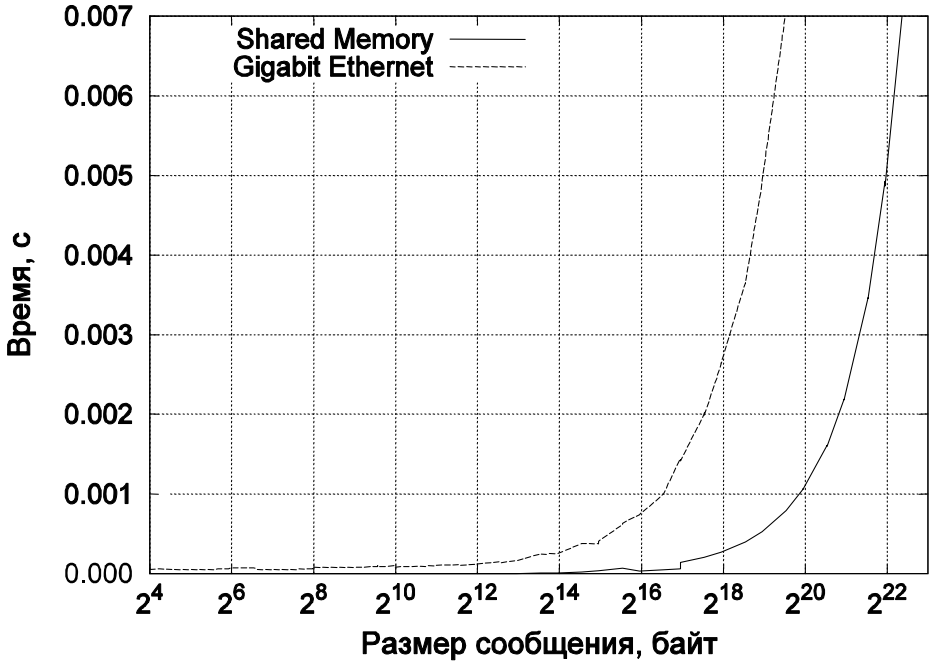
- Разработанный метод применим для всех типов коллективных обменов.
- Время затрачиваемое на построение отображений ветвей не значительно и компенсируется многократным обращением в программе к коллективным операциям информационных обменов.

Направление дальнейших работ:

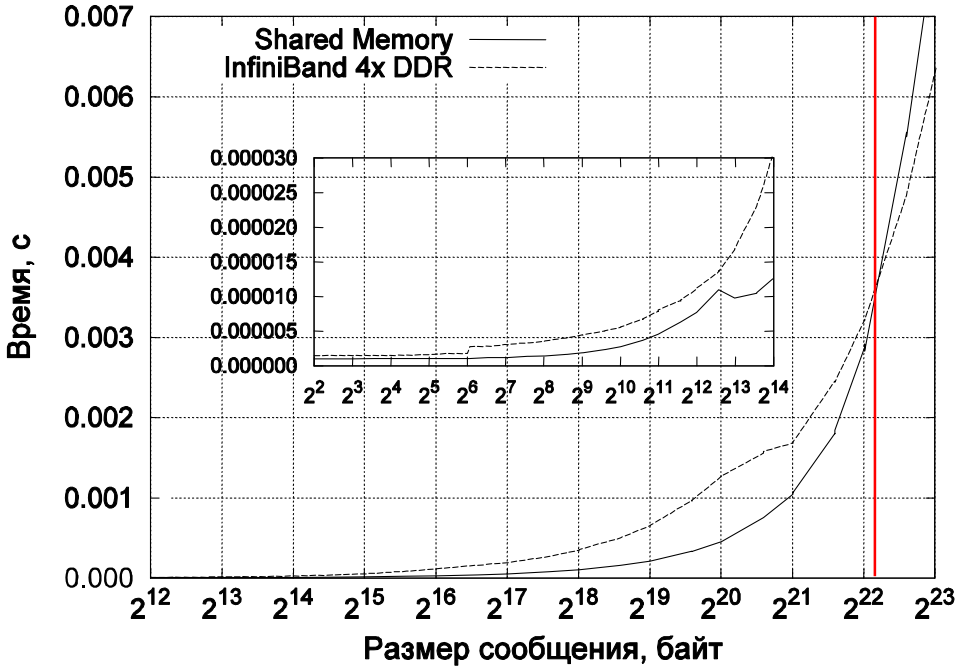
- Развитие структурно-ориентированных методов реализации коллективных операций информационных обменов, не требующих предварительного вложения графов алгоритмов в структуры распределенных ВС.
- Разработка методов динамического выбора оптимального алгоритма реализации коллективной операции для заданного сообщения и подсистемы элементарных машин.
- Создание децентрализованных алгоритмов вложения параллельных программ в большемасштабные иерархические распределенные ВС.
- Оптимизация выполнения параллельной программы на основе данных её предыдущего выполнения (Profile-Guided Optimization - PGO).

Спасибо за внимание!

Иерархическая организация распределенных ВС



а



б

Время передачи сообщений через сеть межузловых связей и общую память узла:

а – кластер ЦПВТ ГОУ ВПО “СибГУТИ” (Gigabit Ethernet)

б – кластер ИВЦ ГОУ ВПО “НГУ” (InfiniBand 4x DDR)