Реализация алгоритмов коллективных обменов: временная и пространственная эффективность

Курносов Михаил Георгиевич [1,2]

Доктор технических наук, доцент

WWW: www.mkurnosov.net

[1] Заведующий Кафедрой вычислительных систем
Сибирский государственный университет телекоммуникаций и информатики, Новосибирск

[2] Лаборатория вычислительных систем
Институт физики полупроводников им. А.В. Ржанова СО РАН, Новосибирск

Тринадцатая международная азиатская школа-семинар "Проблемы оптимизации сложных систем"
в рамках международной мультиконференции IEEE SIBIRCON 2017,
ИВМиМГ СО РАН, г. Новосибирск, Россия,
18 – 22 сентября 2017 г.

СибГУТИ ИФП СО РАН

КОЛЛЕКТИВНЫЕ ОБМЕНЫ ИНФОРМАЦИЕЙ В ВС

В коллективной операции обмена участвуют все ветви программы

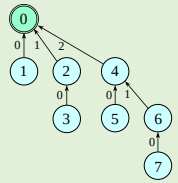
- трансляционный обмен (ТО, «один-всем», one-to-all broadcast)
- коллекторный обмен (КО, «все-одному», all-to-one broadcast)
- трансляционно-циклический обмен (ТЦО, «каждый-всем», all-to-all broadcast)

MPI 3.1
34 операции

MPICH/MVAPICH, Open MPI, Intel MPI
> 70 алгоритмов

Трансляционный обмен (One-to-all)
MPI_Bcast/ MPI_Scatter

- Алгоритм биномиального дерева (binomial tree)
- Алгоритм бинарного дерева
- Scatter + allgather
- Алгоритм плоского дерева



Коллекторный обмен (All-to-one)
MPI_Reduce/MPI_Gather

- Алгоритм биномиального дерева (binomial tree)
- Алгоритм бинарного дерева (binary tree)
- Алгоритм *k*-цепочек (*k*-chain tree)
- Конвейерный алгоритм (pipeline)
- Алгоритмы Р. Рабенсейфнера
- Плоское дерево (flat/linear tree)

Трансляционно-циклический обмен (All-to-all)
MPI_Allgather, MPI_Allreduce, MPI_Alltoall

- Алгоритм рекурсивного удваивания (recursive doubling)
- Алгоритм Дж. Брука (J. Bruck)
- Кольцевой алгоритм (ring)
- Алгоритм попарных обменов (pairwise exchange)
- Рассеивающий алгоритм (dissemination)

[*] R. Thakur, R. Rabenseifner, W. Gropp. ***Optimization of collective communication operations in MPICH*** // Int. Journal of High Performance Computing Applications. – 2005. – Vol. 19 (1). – P. 49-66.

НАПРАВЛЕНИЯ РАЗВИТИЯ КОММУНИКАЦИОННЫХ СЕТЕЙ ВС

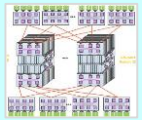
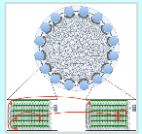
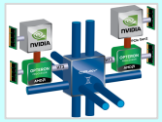
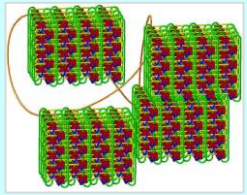
- **Увеличение производительности сети: сокращение латентности и повышение пропускной способности передачи сообщений**
  - ❑ *глубокая интеграция сетевых контроллеров в процессорные ядра* и их иерархию памяти (без использования двойного аппаратного преобразования сообщений из внутрисистемного протокола PCI Express в протокол внешнего сетевого адаптера InfiniBand, network offload)
  - ❑ построение *сетей с малым диаметром* (low-diameter network), обеспечивающих невысокую максимальную латентность и не требующие мощных сетевых контроллеров, потребляющих относительно большое количество энергии

Cray Gemini, Cray Aries, IBM PERCS, Fujitsu Tofu, Bull/Atos BXI,
SKIF 3D-torus, СМПО, Ангара, TH Express-2, МВС-Экспресс,
сети на базе открытого стандарта InfiniBand (Mellanox, Intel True Scale Fabric)

АРХИТЕКТУРНЫЕ СВОЙСТВА ВЫСОКОПРОИЗВОДИТЕЛЬНЫХ ВС

Иерархическая структура коммуникационной среды

Top500 (#49, июнь 2017)

#	Система	Rmax, PFLOPS	Rpeak / HPCG, %	Иерархические уровни коммуникационной среды ВС				
1	**Sunway TaihuLight** 10 649 600 ядер	**5 уровней**	0.38	**Switch network** *Mellanox* 40 стоек (40 960 узлов)	**Supernode network** *fully connected* 256 supernodes	**Sunway Network** *PCIe 3.0* 40 960 узлов	**Network on Chip** 4 core groups 260 ядер	**Общая память** **8 GiB DDR3** 1 MPE + 64 CPE (mesh 8x8, RISC)
2	**Tianhe-2** MilkWay-2 3 120 000 ядер	**3 уровня**	1.06	 **TH Express-2** ***fat tree*** 16 000 узлов (384 000 ядер Intel Xeon IVB + 2 736 000 Xeon Phi)			**Intel QPI** 2 x Intel Xeon 3 x Xeon Phi	**Общая память** 12 ядер Intel Xeon 57 ядер Intel Phi
3	**Piz Daint** Cray XC50 361 760 ядер	**3 уровня**	1.86	 **Cray Aries network** ***Dragonfly*** (3 уровня) 5 320 узлов (63 840 ядер Intel Xeon + 297 920 SM Tesla P100)			**Aries router** *PCIe 3.0* 4 узла	**Общая память** 12 ядер Intel Xeon 3 584 ядер Tesla P100
4	**Titan** Cray XK7 560 640 ядер	**3 уровня**	1.19	 **Cray Gemini** ***3D-тор*** 18 688 узлов (299 008 ядер Intel Xeon + 261 632 SM Tesla X20X)			**Gemini router** *HT 3.0* 2 узла	**Общая память** 16 ядер AMD Opteron 2 688 ядер Tesla K20X
5	**Sequoia** IBM BlueGene/Q 1 572 864 ядер	**2 уровня**	1.64	 ***5D-тор*** 98 304 узлов				**Общая память** **16 GiB DDR3** 16 + 2 ядер IBM PowerPC A2

ИНСТРУМЕНТАРИЙ ОРГАНИЗАЦИИ ФУНКЦИОНИРОВАНИЯ ВС

Разработка параллельных программ

MPI: MPICH, MVAPICH, Intel MPI, Open MPI; **SHMEM:** OpenSHMEM

Multihreading: OpenMP, Transactional Memory

Lightweight threading: Qthreads, MassiveThreads, Argobots

Accelerators: OpenACC, OpenCL, NVIDIA CUDA,

Оптимизирующая компиляция

(Полу)автоматическая векторизация кода (AVX, AltiVec, NEON)

PGAS

Unified Parallel C, Co-array Fortran, Cray Chapel, IBM X10

Анализ и отладка

PMPI/MPIT Intf., Vampir, Intel Tools, mpiP

Отказоустойчивость и живучесть программ

ULFM, DMTCP, BLCR, CRIU

Поддержка мультипрограммных режимов работы

TORQUE, SLURM, PBS Pro, LoadLeveler, LSF, SGE

Контроль и диагностика

Ganglia, Nagios (SNMP)

Операционная система

GNU/Linux, **LWK**: IBM CNK, Cray CNL, Kitten, ZeptoOS

АНАЛИЗ КОЛЛЕКТИВНЫХ ОПЕРАЦИЙ В РЕАЛЬНЫХ ПРИЛОЖЕНИЯХ

- Выполнен анализ промежуточных отчетов международного проекта **HPC Advisory Council** // http://hpcadvisorycouncil.com/best_practices.php

- Пакеты моделирования физико-технических процессов и природных явлений: CP2K, Quantum Espresso, VASP, DL-POLY, Amber, AMR, ANSYS, CPMD, WRF, GROMACS, LAMMPS, MILC, …

КОЛЛЕКТИВНЫЕ ОПЕРАЦИИ В РЕАЛЬНЫХ ПРИЛОЖЕНИЯХ (1)

Пакет	Предметная область	Наиболее часто вызываемые функции MPI	Функции MPI с наибольшим временем выполнения	Размеры сообщений
Abaqus	Инженерный анализ методом конечных элементов	MPI_Test, MPI_Iprobe	MPI_Test, MPI_Waitall, MPI_Bcast, MPI_Gather, MPI_Allreduce, MPI_Scatterv	64-256 байт
ABySS	Вычислительная биология	MPI_Send, MPI_Irecv, MPI_Test, MPI_Allreduce, MPI_Barrier	MPI_Send, MPI_Irecv, MPI_Test, MPI_Allreduce, MPI_Barrier	
AcuSolve	Задачи гидродинамики	MPI_Recv, MPI_Isend, MPI_Allreduce, MPI_Barrier	MPI_Allgatherv, MPI_Allgather, MPI_Comm_free,	< 4 Кбайт
Amber	Молекулярная динамика	MPI_Irecv, MPI_Isend, MPI_Waitany	MPI_Allgatherv, MPI_Allreduce, MPI_Waitall, MPI_Waitany	0-64 байт, 16-64 Кбайт
AMG2013	Алгебраический многосеточный решатель	MPI_Irecv, MPI_Isend, MPI_Allreduce	MPI_Allreduce, MPI_Allgather, MPI_Waitall	0-256 байт, 64-256 Кбайт
AMR	Моделирование на адаптивных сетках	MPI_Irecv, MPI_Send, MPI_Waitsome	MPI_Send, MPI_Allreduce, MPI_Reduce, MPI_Barrier	0-64 байт
ANSYS CFX	Вычислительная динамика жидкостей и газов	MPI_Send, MPI_Recv, MPI_Iprobe, MPI_Bcast	MPI_Recv, MPI_Bcast	< 64 Кбайт

КОЛЛЕКТИВНЫЕ ОПЕРАЦИИ В РЕАЛЬНЫХ ПРИЛОЖЕНИЯХ (2)

Пакет	Предметная область	Наиболее часто вызываемые функции MPI	Функции MPI с наибольшим временем выполнения	Размеры сообщений
ANSYS FLUENT	Вычислительная динамика жидкостей и газов	MPI_Recv, MPI_Allreduce, MPI_Bcast, MPI_Waitall	MPI_Recv, MPI_Allreduce, MPI_Bcast, MPI_Waitall	< 64 Кбайт
BQCD	Квантовая хромодинамика	MPI_Isend, MPI_Irecv, MPI_Waitall, MPI_Barrier, MPI_Allreduce	MPI_Waitall, MPI_Isend, MPI_Irecv, MPI_Barrier, MPI_Allreduce	< 500 Кбайт
CAM-SE	Моделирования климатических и погодных явлений	MPI_Waitall, MPI_Barrier, MPI_Allreduce	MPI_Waitall, MPI_Barrier, MPI_Allreduce	64-81 Кбайт, 2-3 Мбайт
COSMO	Моделирования климатических и погодных явлений	MPI_Sendrecv, MPI_Allreduce, MPI_Allgather, MPI_Gather	MPI_Sendrecv, MPI_Allreduce, MPI_Wait	< 1 Кбайт
CP2K	Атомно-молекулярное моделирование	MPI_Alltoallv, MPI_Irecv, MPI_Isend, MPI_Waitall	MPI_Alltoallv, MPI_Waitall, MPI_Reduce, MPI_Alltoall	0-256 байт, 16-256 Кбайт
CPMD	Молекулярная динамика	MPI_Bcast, MPI_Barrier, MPI_Scatter, MPI_Sendrecv	MPI_Alltoall, MPI_Bcast, MPI_Allreduce, MPI_Barrier	
DL-POLY	Молекулярная динамика	MPI_Allreduce, MPI_Scatter, MPI_Recv, MPI_Send	MPI_Allreduce, MPI_Scatter, MPI_Recv, MPI_Send	16-256 Кбайт

КОЛЛЕКТИВНЫЕ ОПЕРАЦИИ В РЕАЛЬНЫХ ПРИЛОЖЕНИЯХ (3)

Пакет	Предметная область	Наиболее часто вызываемые функции MPI	Функции MPI с наибольшим временем выполнения	Размеры сообщений
FLOW-3D	Вычислительная аэро-, гидро- и газовая динамика	MPI_Isend, MPI_Irecv, MPI_Waitall, MPI_Allreduce	MPI_Allreduce, MPI_Waitall, MPI_Bcast	4-16 байт, 1-4 Мбайт
Graph500	Поиск в ширину в графе	MPI_Alltoallv, MPI_Allreduce, MPI_Test, MPI_Alltoall	MPI_Alltoallv, MPI_Allreduce, MPI_Test, MPI_Alltoall	< 256 Кбайт
GROMACS	Моделирование физико-химических процессов	MPI_Send, MPI_Sendrecv, MPI_Isend, MPI_Irecv, MPI_Waitall, MPI_Alltoall	MPI_Alltoall, MPI_Recv, MPI_Sendrecv, MPI_Send	4-64 Кбайт
HOOMD-blue	Молекулярная динамика	MPI_Bcast, MPI_Allreduce, MPI_Waitall	MPI_Bcast, MPI_Allreduce, MPI_Waitall	< 64 Кбайт
HPCC HPL	Решение плотной СЛАУ методом LU-декомпозиции	MPI_Iprobe, MPI_Send, MPI_Recv, MPI_Wait, MPI_Allreduce	MPI_Iprobe, MPI_Send, MPI_Recv, MPI_Wait, MPI_Allreduce	
HPCC PTRANS	Транспонирование матрицы	MPI_Sendrecv, MPI_Allreduce, MPI_Barrier	MPI_Sendrecv, MPI_Allreduce, MPI_Barrier	458 Кбайт
HPCC Random Access	Произвольный доступ к памяти	MPI_Waitany, MPI_Wait, MPI_Isend, MPI_Irecv, MPI_Alltoall, MPI_Allreduce	MPI_Waitany, MPI_Wait, MPI_Isend, MPI_Irecv, MPI_Alltoall, MPI_Allreduce	< 4 Кбайт

КОЛЛЕКТИВНЫЕ ОПЕРАЦИИ В РЕАЛЬНЫХ ПРИЛОЖЕНИЯХ (4)

Пакет	Предметная область	Наиболее часто вызываемые функции MPI	Функции MPI с наибольшим временем выполнения	Размеры сообщений
HPCC FFT	1D быстрое преобразование Фурье	MPI_Alltoall, MPI_Bcast, MPI_Allreduce	MPI_Alltoall, MPI_Bcast, MPI_Allreduce	
HPCG	Решение СЛАУ с разреженной матрицей	MPI_Wait, MPI_Allreduce, MPI_Send, MPI_Bcast	MPI_Wait, MPI_Allreduce, MPI_Send, MPI_Bcast	< 2 Кбайт
LAMMPS	Молекулярная динамика	MPI_Send, MPI_Waitany, MPI_Wait, MPI_Allreduce, MPI_Bcast	MPI_Send, MPI_Waitany, MPI_Wait, MPI_Allreduce, MPI_Bcast	< 256 Кбайт
LS-DYNA	Анализ быстротекущих процессов в задачах механики твердого и жидкого тела	MPI_Recv, MPI_Allreduce, MPI_Bcast, MPI_Alltoallv	MPI_Recv, MPI_Allreduce, MPI_Bcast, MPI_Alltoallv	< 4 Кбайт
MILC	Квантовая хромодинамика	MPI_Wait, MPI_Isend, MPI_Irecv, MPI_Allreduce	MPI_Wait, MPI_Allreduce, MPI_Isend	< 1 Кбайт
MSC Nastran	Конечно-элементный анализ	MPI_Recv, MPI_Ssend	MPI_Recv, MPI_Ssend, MPI_Barrier	< 64 байт
NAMD	Молекулярная динамика	MPI_Iprobe, MPI_Barrier,	MPI_Iprobe, MPI_Barrier, MPI_Comm_dup	< 10 Кбайт

КОЛЛЕКТИВНЫЕ ОПЕРАЦИИ В РЕАЛЬНЫХ ПРИЛОЖЕНИЯХ (5)

Пакет	Предметная область	Наиболее часто вызываемые функции MPI	Функции MPI с наибольшим временем выполнения	Размеры сообщений
NWChem	Вычислительная химия	MPI_Send, MPI_Recv, MPI_Barrier	MPI_Barrier, MPI_Recv	4-16 Кбайт
OpenAtom	Молекулярная динамики на квантовом уровне	MPI_Iprobe, MPI_Recv, MPI_Test, MPI_Isend	MPI_Iprobe, MPI_Recv, MPI_Test, and MPI_Isend MPI_Barrier	1-16 Кбайт
Open FOAM	Задачи гидродинамики	MPI_Irecv, MPI_Isend, MPI_Waitall, MPI_Allreduce	MPI_Allreduce, MPI_Waitall, MPI_Alltoallv	< 64 Кбайт
Quantum Espresso	Моделирование электронной структуры материалов	MPI_Barrier, MPI_Alltoall, MPI_Allreduce	MPI_Barrier, MPI_Alltoall, MPI_Allreduce	< 1 Мбайт
VASP	Квантово-механическое моделирование	MPI_Bcast, MPI_Allreduce, MPI_Recv, MPI_Alltoall	MPI_Alltoallv, MPI_Alltoall, MPI_Bcast	< 256 Кбайт
WRF	Моделирования климатических и погодных явлений	MPI_Bcast, MPI_Scatterv, MPI_Wait	MPI_Bcast, MPI_Scatterv, MPI_Wait	< 16 Кбайт

БАЗИСНЫЙ НАБОР КОЛЛЕКТИВНЫХ ОПЕРАЦИИ

Проведенный анализ использования функций MPI в промышленных пакетах моделирования позволяет выделить ***базисный набор коллективных операций***, оказывающих значительное влияние на масштабируемость MPI-программ:

- **трансляционно-циклические обмены** (all-to-all):
  MPI_Allreduce, MPI_Alltoallv, MPI_Alltoall, MPI_Barrier, MPI_Allgather, MPI_Allgatherv
- **трансляционные обмены** (one-to-all, all-to-one):
  MPI_Bcast, MPI_Gather, MPI_Scatter
- **дифференцированные обмены**: MPI_Isend, MPI_Irecv

ШАБЛОНЫ РЕАЛИЗАЦИИ КОЛЛЕКТИВНЫХ ОПЕРАЦИЙ

- Рассылка данных по кольцу (ring)
- Рекурсивное удваивание (recursive doubling)
- Алгоритм Брука (J. Bruck)
- Попарные обмены (pairwise exchange)
- Логическое выстраивание процессов в деревья различных видов:
  - биномиальные деревья (*k*-номиальные деревья)
  - сбалансированные *k*-арные деревья
  - плоские деревья
  - конвейеры/цепочки

РЕАЛИЗАЦИЯ КОЛЛЕКТИВНЫХ ОПЕРАЦИИ

#	Операция	MVAPICH2 2.3a	Intel MPI 2017 Update 2	Open MPI 2.1.0
1	MPI_BARRIER	1. Dissemination algorithm [1] 2. Dissemination SMP 2L 3. Pairwise exchange with recursive doubling [2] 4. Pairwise exchange with RDBL SMP (2L)	1. Dissemination 2. Recursive doubling 3. Topology aware dissemination 4. Topology aware recursive doubling 5. Binominal gather + scatter 6. Topology aware binominal gather + scatter 7. Topology aware SHM-based flat 8. Topology aware SHM-based Knomial 9. Topology aware SHM-based Knary	1. BASE: Double ring 2. BASE: Recursive doubling 3. BASE: Recursive doubling up & down 4. BASE: Bruck 5. BASE: Linear 6. BASIC: Tree 7. BASIC: Allreduce 8. HCOLL: Barrier 9. Portals4: h-cube top & bottom 10. SM: Tree
2	MPI_IBARRIER	Dissemination algorithm	Dissemination algorithm	1. HCOLL: Ibarrier 2. Portals4: Ibarrier 3. LIBNBC: Dissemination algorithm
3	MPI_BCAST	1. SMP 2L Binomial tree (inter, intra) 2. Binomial tree 3. Scatter (bmtree) + allgather (rdbl) 4. Scatter (bmtree) + allgather (ring) 5. Allgather (ring) + overlap with Shmem bcast 6. Bcast knomial 7. IB mcast 8. Pipelined bcast	1. Binomial 2. Recursive doubling 3. Ring 4. Topology aware binomial 5. Topology aware recursive doubling 6. Topology aware ring 7. Shumilin's 8. Knomial 9. Topology aware SHM-based flat 10. Topology aware SHM-based Knomial 11. Topology aware SHM-based Knary	1. base: Binary tree 2. base: Pipeline 3. base: k-chain 4. base: Binomial tree 5. base: Split binary tree 6. base: Linear 7. basic: bmtree 8. FCA 9. HCOLL 10. Portals4 11. sm: mcb tree
4	MPI_IBCAST	1. Binomial tree 2. Scatter (bmtree) + allgather (rdbl) 3. Scatter (bmtree) + allgather (ring) 4. SMP 2L Binomial tree (inter, intra)	1. Binomial 2. Recursive doubling 3. Ring 4. Knomial	1. Libnbc: binomial 2. Libnbc: linear 3. Libnbc: chain

MPI 3.1 -- 34 операции

MVAPICH, Open MPI, Intel MPI
> 70 алгоритмов

РЕАЛИЗАЦИЯ КОЛЛЕКТИВНЫХ ОПЕРАЦИИ

#	Операция	MVAPICH2 2.3a	Intel MPI 2017 Update 2	Open MPI 2.1.0
5	MPI_GATHER	1. Binomial tree 2. Linear 3. SMP 2L	1. Binomial 2. Topology aware binomial 3. Shumilin's 4. Binomial with segmentation	1. base: binomial tree 2. basic: linear
6	MPI_IGATHER	1. Binomial tree	1. Binomial 2. Knomial	1. libnbc: linear
7	MPI_GATHERV	1. Linear	1. Linear 2. Topology aware linear 3. Knomial	1. basic: linear 2. hcol
8	MPI_IGATHERV	1. Linear	1. Linear	1. libnbc: linear
9	MPI_SCATTER	1. Binomial tree 2. Mcast 3. Linear 4. SMP 2L Bmtree 5. SMP 2L Linear	1. Binomial 2. Topology aware binomial 3. Shumilin's	1. base: binomial 2. basic: linear 3. Portals4: linear
10	MPI_ISCATTER	1. Binomial tree	1. Binomial 2. Knomial	1. libnbc: linear
11	MPI_SCATTERV	1. Linear	1. Linear 2. Topology aware linear	1. basic: linear
12	MPI_ISCATTERV	1. Linear	1. Linear	1. libnbc: linear
13	MPI_ALLGATHER	1. Bruck	1. Recursive doubling	1. base: bruck

РЕАЛИЗАЦИЯ КОЛЛЕКТИВНЫХ ОПЕРАЦИИ

#	Операция	MVAPICH2 2.3a	Intel MPI 2017 Update 2	Open MPI 2.1.0
13	MPI_ALLGATHER	1. Bruck 2. Ring 3. Recursive doubling	1. Recursive doubling 2. Bruck's 3. Ring 4. Topology aware Gatherv + Bcast 5. Knomial	1. base: bruck 2. base: rdbl 3. base: ring 4. base: neighbor exchange 5. base: linear 6. fca: 7. hcoll: 8. Portals4:
14	MPI_IALLGATHER	1. Bruck 2. Ring 3. Recursive doubling	1. Bruck’s 2. Ring 3. Recursive doubling	1. libnbc:
15	MPI_ALLGATHERV	1. Bruck 2. Recursive doubling 3. Ring	1. Recursive doubling 2. Bruck's 3. Ring 4. Topology aware Gatherv + Bcast	1. base: bruck 2. base: ring 3. base: neib. exch 4. base: gatherv + bcast 5. basic: alltoallv 6. hcol:
16	MPI_IALLGATHERV	1. Bruck 2. Recursive doubling 3. Ring	1. Recursive doubling 2. Bruck’s 3. Ring	1. libnbc: linear 2. hcol:
17	MPI_ALLTOALL	1. Bruck 2. Linear isend+irecv 3. Pairwise exch.	1. Bruck's 2. Isend/Irecv + waitall 3. Pair wise exchange 4. Plum's	1. base: linear 2. base: pairwise exch. 3. base: bruck 4. base: linear block 5. basic: linear 6. hcol:

РЕАЛИЗАЦИЯ КОЛЛЕКТИВНЫХ ОПЕРАЦИИ

#	Операция	MVAPICH2 2.3a	Intel MPI 2017 Update 2	Open MPI 2.1.0
18	MPI_IALLTOALL	1. Linear isend+irecv	1. Bruck’s 2. Isend/Irecv + Waitall 3. Pairwise exchange	1. libnbc: linear
19	MPI_ALLTOALLV	1. Linear isend+irecv	1. Isend/Irecv + waitall 2. Plum's	1. base: linear 2. base: pairwise exch. 3. base: linear block 4. basic: linear 6. hcol:
20	MPI_IALLTOALLV	1. Linear isend+irecv	1. Isend/Irecv + Waitall	1. libnbc: linear 2. libnbc: pairwise exch. 3. hcol:
21	MPI_ALLTOALLW	1. Linear isend+irecv	1. Isend/Irecv + waitall	1. basic: linear
22	MPI_IALLTOALLW	1. Linear isend+irecv	1. Isend/Irecv + Waitall	1. libnbc: linear 2. libnbc: pairwise exch.
23	MPI_ALLREDUCE	1. Recursive doubling 2. Rabenseifner’s algorithm 3. SMP 2L (reduce, allreduce, bcast) 4. Mellanox SHARP (contig) 5. SMP 2L (reduce, allreduce, bcast) 6. HW Mulitcast: reduce + mcast	1. Recursive doubling 2. Rabenseifner's 3. Reduce + Bcast 4. Topology aware Reduce + Bcast 5. Binomial gather + scatter 6. Topology aware binominal gather + scatter 7. Shumilin's ring 8. Ring 9. Knomial 10. Topology aware SHM-based flat 11. Topology aware SHM-based Knomial 12. Topology aware SHM-based Knary	1. BASE: recursive doubling 2. BASE: ring 3. BASE: ring segmented 4. BASE: linear 5. BASE: reduce + bcast 6. HCOLL: allreduce 7. Portals4: kary tree top & bottom 8. SM: reduce + bcast

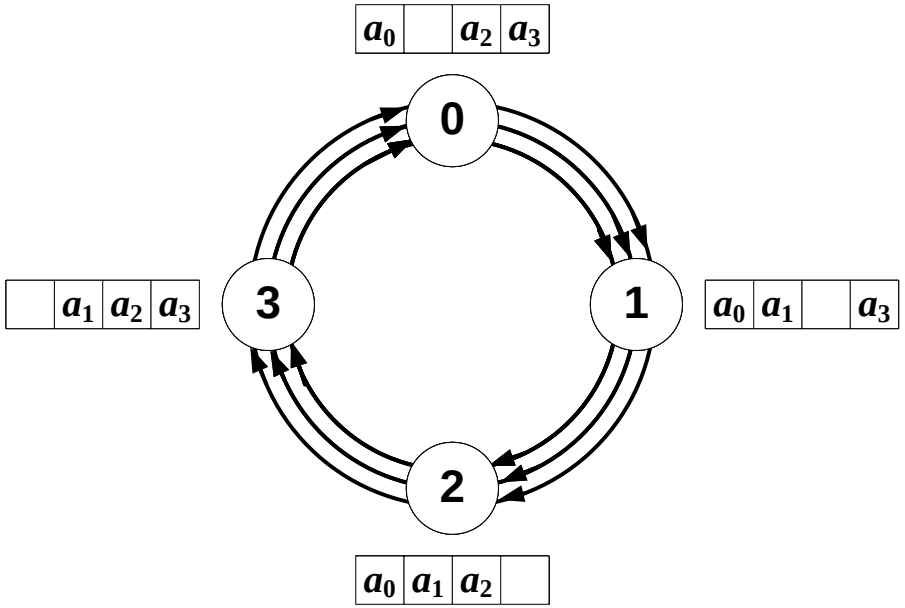
РЕАЛИЗАЦИЯ КОЛЛЕКТИВНЫХ ОПЕРАЦИИ

#	Операция	MVAPICH2 2.3a	Intel MPI 2017 Update 2	Open MPI 2.1.0
24	MPI_IALLREDUCE	1. Recursive doubling 2. Rabenseifner’s algorithm 3. SMP 2L (reduce, allreduce, bcast)	1. Recursive doubling 2. Rabenseifner’s 3. Reduce + Bcast 4. Ring (patarasuk) 5. Knomial 6. Binomial	1. HCOLL: allreduce 2. libnbc: binomial tree 3. libnbc: ring 4. libnbc: linear 5. Portals4: kary tree top
25	MPI_REDUCE	1. Binomial 2. Rabensifner’s 3. SMP 2L: intra, inter 4. Knomial 5. Shmem 6. Zcpy	1. Shumilin's 2. Binomial 3. Topology aware Shumilin's 4. Topology aware binomial 5. Rabenseifner's 6. Topology aware Rabenseifner's 7. Knomial 8. Topology aware SHM-based flat 9. Topology aware SHM-based Knomial 10. Topology aware SHM-based Knary 11. Topology aware SHM-based binomial	1. BASE: kchain 2. BASE: pipeline 3. BASE: binary tree 4. BASE: binomial tree 5. BASE: in order binary 6. BASE: linear 7. SM: reduce 8. FCA: 9. HCOLL: 10. Portals4: 11. SM: reduce
26	MPI_IREDUCE	1. Binomial 2. Rabensifner’s	1. Rabenseifner’s 2. Binomial 3. Knomial	1. libnbc: binomial 2. libnbc: kchain 3. libnbc: linear
27	MPI_REDUCE_SCATTER_BLOCK	1. Traff’s butterfly 2. Rec. halving + rec. doubling 3. Recursive doubling 4. Pairwise exch.	--	1. base: reduce + scatter
28	MPI_IREDUCE_SCATTER_BLOCK	1. Traff’s butterfly 2. Rec. halving + rec. doubling 3. Recursive doubling	--	1. libnbc: pairwise exch

РЕАЛИЗАЦИЯ КОЛЛЕКТИВНЫХ ОПЕРАЦИИ

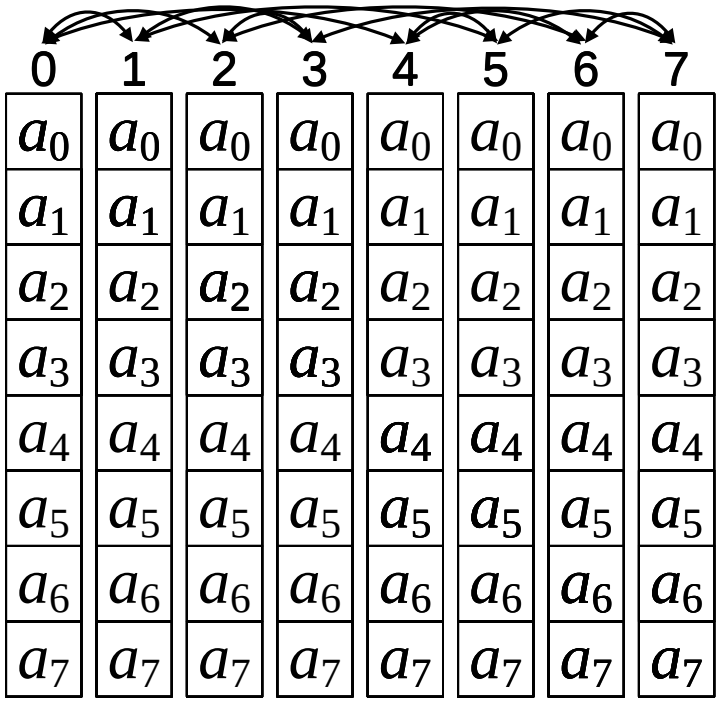
#	Операция	MVAPICH2 2.3a	Intel MPI 2017 Update 2	Open MPI 2.1.0
28	MPI_IREDUCE_SCATTER_BLOCK	1. Traff's butterfly 2. Rec. halving + rec. doubling 3. Recursive doubling 4. Pairwise exch.	--	1. libnbc: pairwise exch
29	MPI_REDUCE_SCATTER	1. Traff's butterfly 2. Rec. halving + rec. doubling 3. Recursive doubling 4. Pairwise exch.	1. Recursive halving 2. Pairwise exchange 3. Recursive doubling 4. Reduce + Scatterv 5. Topology aware Reduce + Scatterv	1. base: reduce + scatterv 2. base: rec. halving 3. base: ring 4. basic: rec. halving
30	MPI_IREDUCE_SCATTER	1. Traff's butterfly 2. Rec. halving + rec. doubling 3. Recursive doubling 4. Pairwise exch.	1. Recursive halving 2. Pairwise 3. Recursive doubling	1. libnbc: pairwise exch
31	MPI_SCAN	1. Recursive doubling	1. Partial results gathering 2. Topology aware partial results gathering	1. basic: pipeline (linear)
32	MPI_ISCAN	1. Recursive doubling	1. Recursive doubling	1. libnbc: pipeline (linear)
33	MPI_EXSCAN	1. Recursive doubling 2. SMP 2L	1. Partial results gathering 2. Partial results gathering regarding layout of processes	1. basic: pipeline (linear)
34	MPI_IEXSCAN	1. Recursive doubling	1. Recursive doubling	1. libnbc: pipeline (linear)

КОЛЬЦЕВОЙ АЛГОРИТМ (RING)



Каждая ветвь выполняет 2(n – 1) обменов

АЛГОРИТМ РЕКУРСИВНОГО УДВАИВАНИЯ (RECURSIVE DOUBLING)

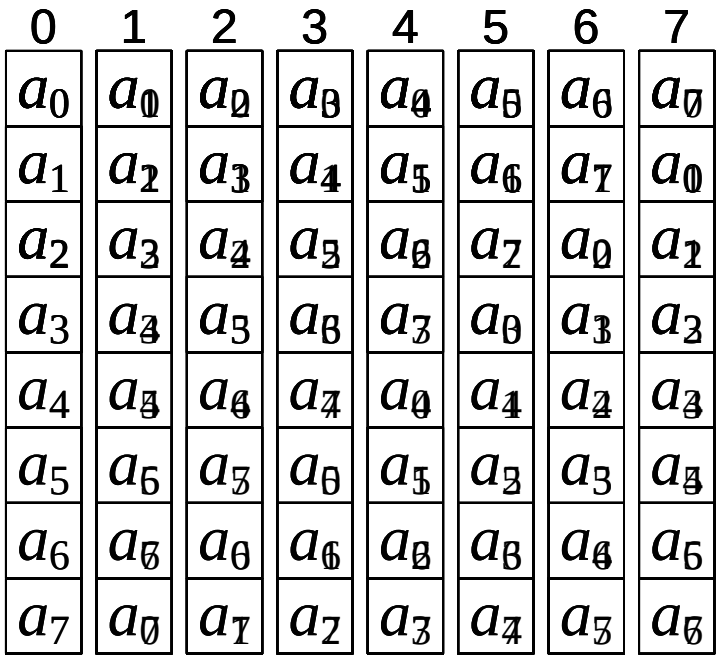


Количество обменов: $2\log_2 n$

Только для n равного степени двойки

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

АЛГОРИТМ ДЖ. БРУКА (J. BRUCK ET AL., 1997)

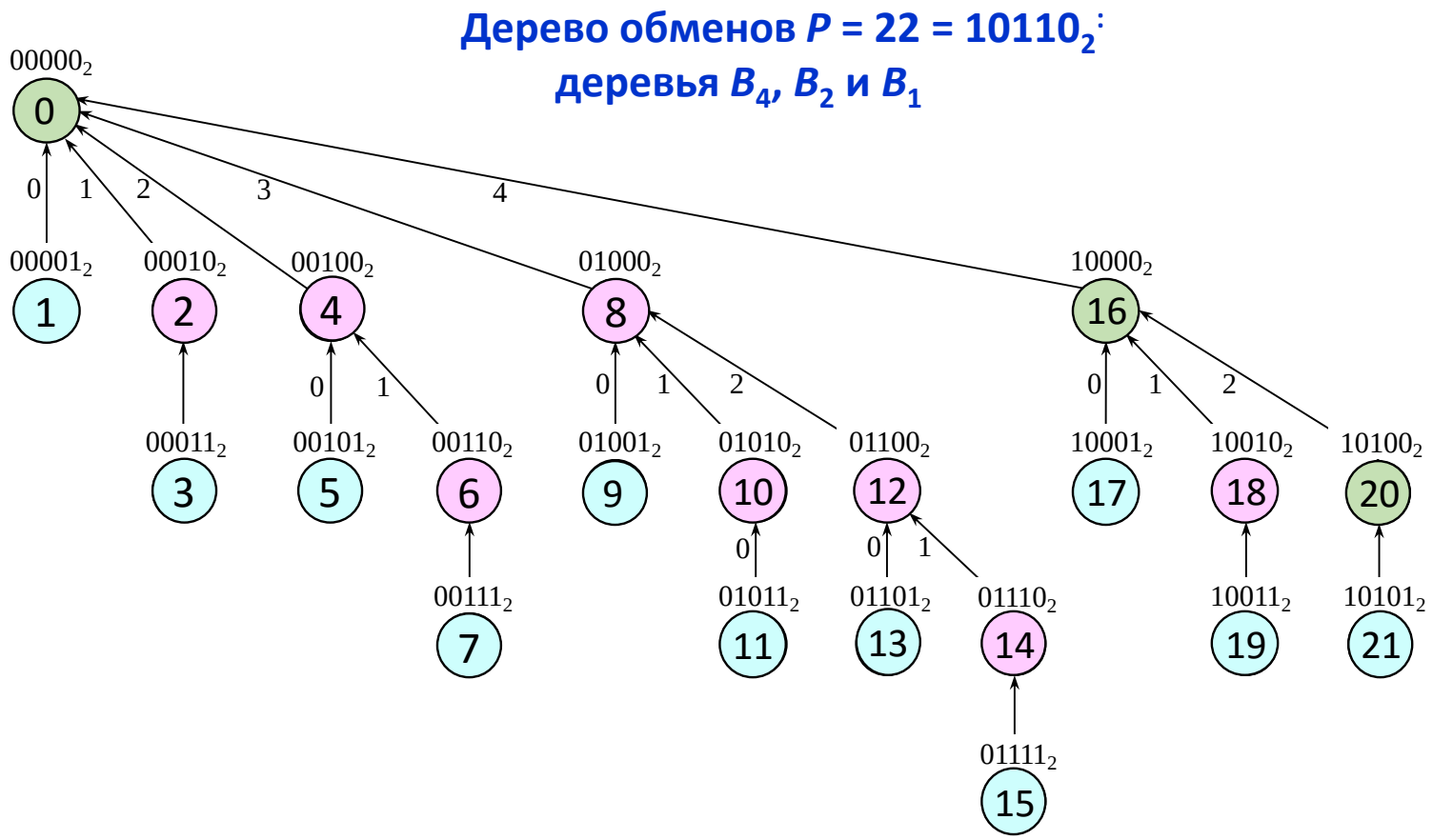


Количество обменов: $2\lceil \log_2 n \rceil$

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

АЛГОРИТМ БИНОМИАЛЬНОГО ДЕРЕВА (BINOMIAL TREE)

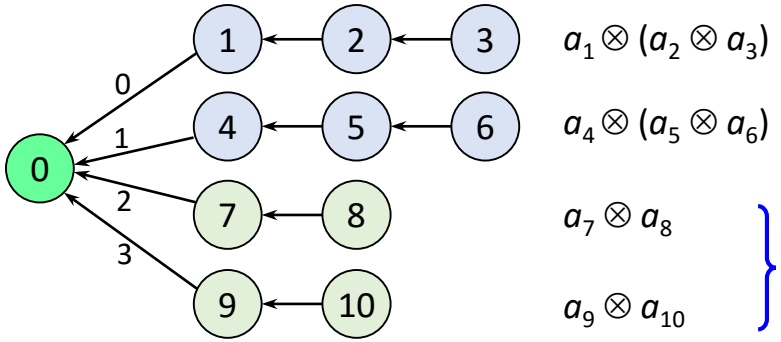
- Дерево обменов – совокупность *биномиальных деревьев*, степени которых определяются номерами значащих битов в *P*



АЛГОРИТМ *K* ПАРАЛЛЕЛЬНЫХ ЦЕПОЧЕК (*K*-CHAIN TREE)

- Процессы выстраиваются в *k* цепочек (конвейеров, pipeline) и передают результаты частичных редукций корню – процессу 0
- Остаток (*P* – 1) % *k* процессов распределяется по первым (*P* – 1) % *k* цепочкам – в каждую добавляется по одному процессу [*]

P = 11
k = 4



$a_1 \otimes (a_2 \otimes a_3)$

$a_4 \otimes (a_5 \otimes a_6)$

$a_7 \otimes a_8$

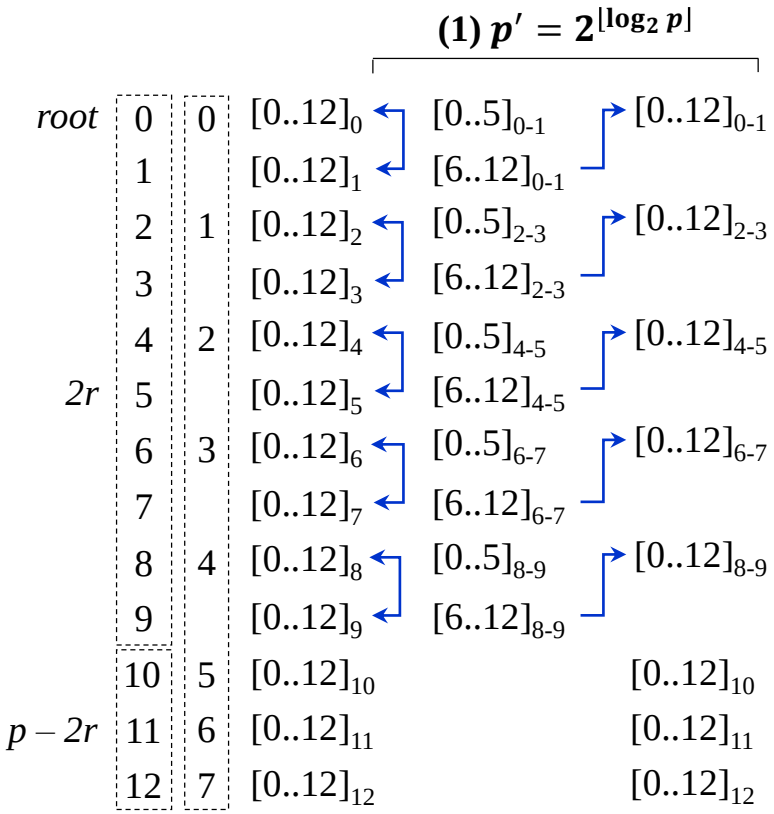
$a_9 \otimes a_{10}$

Число ***длинных*** цепочек: (*P* – 1) % *k*
Число процессов в длинных цепочках: ⌊(*P* – 1) / *k*⌋ + 1

Число ***коротких*** цепочек: *k* – (*P* – 1) % *k*
Число процессов в коротких цепочках: ⌊(*P* – 1) / *k*⌋

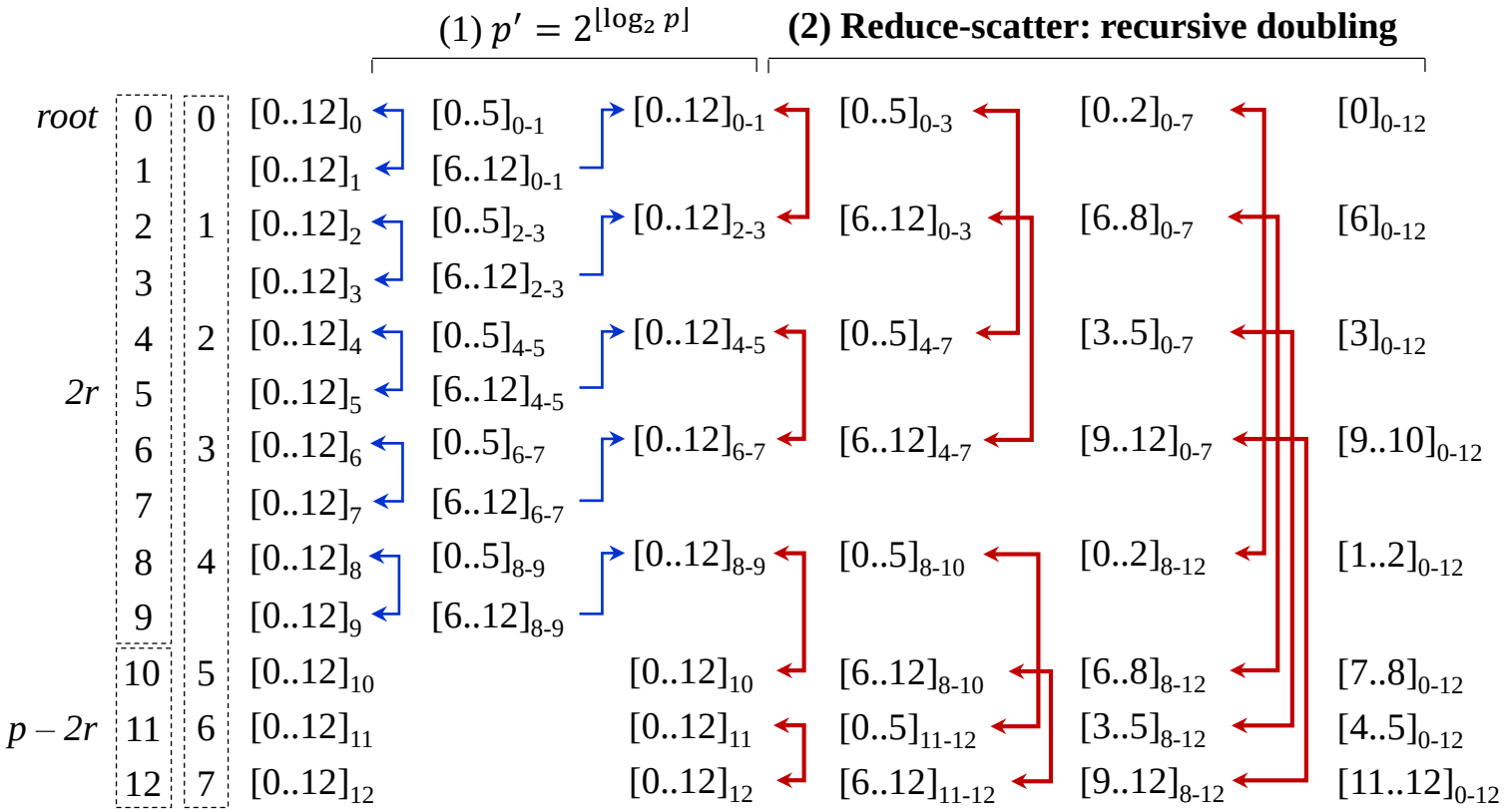
[*] Fagg G., Pjesivac-Grbovic J., Bosilca G., Dongarra J., Jeannot E. ***Flexible collective communication tuning architecture applied to Open MPI*** // Proc. of Euro PVM/MPI, 2006. – P. 1-10.

АЛГОРИТМ Р. РАБЕНСЕЙФНЕРА (MPI_REDUCE)



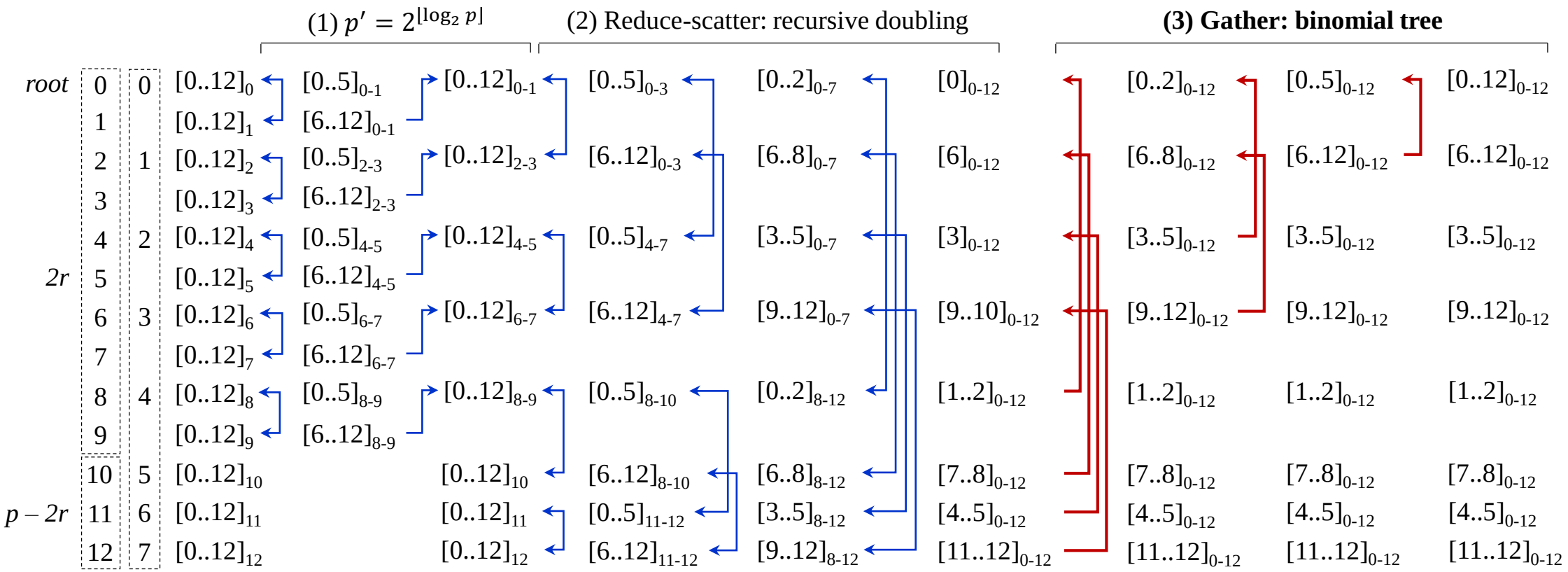
(1) Переход к числу процессов, равному степени 2

АЛГОРИТМ Р. РАБЕНСЕЙФНЕРА (MPI_REDUCE)



(2) Рекурсивное удваивание между активными процессами, число которых равно степени 2

АЛГОРИТМ Р. РАБЕНСЕЙФНЕРА (MPI_REDUCE)



(3) Прием результатов в процессе 0 (биномиальное дерево)

Спасибо за внимание!

Курносов Михаил Георгиевич [1,2]

Доктор технических наук, доцент

WWW: www.mkurnosov.net

[1] Заведующий Кафедрой вычислительных систем
Сибирский государственный университет телекоммуникаций и информатики, Новосибирск

[2] Лаборатория вычислительных систем
Институт физики полупроводников им. А.В. Ржанова СО РАН, Новосибирск

Тринадцатая международная азиатская школа-семинар "Проблемы оптимизации сложных систем"
в рамках международной мультиконференции IEEE SIBIRCON 2017,
ИВМиМГ СО РАН, г. Новосибирск, Россия,
18 – 22 сентября 2017 г.