РЕАЛИЗАЦИЯ АЛГОРИТМОВ КОЛЛЕКТИВНЫХ ОБМЕНОВ: ВРЕМЕННАЯ И ПРОСТРАНСТВЕННАЯ ЭФФЕКТИВНОСТЬ

М. Г. Курносов

Сибирский государственный университет телекоммуникаций и информатики

УДК 004.272

Рассмотрены аспекты реализации алгоритмов коллективных обменов стандарта MPI. Отражены результаты анализа частоты использования функций стандарта MPI в популярных пакетах суперкомпьютерного моделирования. Выделен базисный набор коллективных операций, оказывающих значительное влияние на масштабируемость MPI-программ. Описаны основные подходы к реализации алгоритмов коллективных операций с позиций оптимизации времени их выполнения, сложности по памяти и энергоэффективности.

Ключевые слова: коллективных обменов, MPI, PGAS, параллельное программирование, вычислительные системы.

Введение. Производительность элементарных машин (ЭМ) и коммуникационной сети вычислительной системы (ВС) является основным фактором определяющим время реализации параллельных алгоритмов решения прикладных задач пользователей [1-6].

Архитектура ВС и их компонентов развивается с учетом информационно-вычислительных характеристик основных методов параллельного решения сложных задач (принцип ко-дизайна, co-design) и учетом достигнутых промышленностью ограничений на технологии производства микросхем. Анализ текущего опыта и международных инициатив (USA DOE ECP, ECI, EESI, BDEC, Japan FLAGSHIP 2020, China NSC Project) по построению систем эксафлопсной производительности позволяет предопределить основные архитектурные свойства перспективных ВС [7-9]:

– парадигма формирования элементарных машин – мультиархитектура, многопроцессорные SMP/NUMA-узлы;

– функциональная структура и топология – иерархические с глубокой интеграцией в процессорные ядра средств доступа к коммуникационной сети;

– большемасштабность и масштабируемость.

Потенциальные возможности по параллельной обработке информации увеличиваются на всех уровнях иерархии системы: растёт количество ЭМ и число процессоров в них, число ядер в процессорах, а также количество параллельно работающих скалярных и векторных АЛУ в процессорных ядрах.

Работа в области коммуникационных сетей ведется по трем основным направлениям:

– сокращение потребления энергии коммуникационным оборудованием (коммутаторами, маршрутизаторами и сетевыми контроллерами ЭМ);

– увеличение производительности сети: сокращение латентности и повышение пропускной способности передачи сообщений;

– оптимизация совокупной стоимости владения коммуникационной сетью в расчете на одну ЭМ системы и коммуникационный порт.

Работа выполнена при частичной финансовой поддержке РФФИ (код проекта 15-07-00653).

Для сокращение латентности и повышения пропускной способности сети реализуется глубокая интеграция сетевых контроллеров в процессорные ядра и их иерархию памяти, в частности без использования двойного аппаратного преобразования сообщений из внутрисистемного протокола PCI Express в протокол внешнего сетевого адаптера InfiniBand. Строятся сети с малым диаметром (low-diameter network), обеспечивающие невысокую максимальную латентность и не требующие мощных сетевых контроллеров, потребляющих относительно большое количество энергии.

Часть сетевого стека MPI/PGAS/SHMEM реализуется аппаратно в сетевых контроллерах, что позволяет при должной поддержки со стороны системного программного обеспечения разгрузить процессорные ядра от коммуникационных задач (network offload). Фактически сетевой контролер ЭМ – это специализированный коммуникационный процессор с памятью фиксированного размера. Текущие реализации сетевых контроллеров специализированных коммуникационных сетей (Cray Gemini, Cray Aries, IBM PERCS, Fujitsu Tofu, Bull/Atos BXI, SKIF 3D-torus, СМПО, Ангара, ТН Express-2, МВС-Экспресс) [11-14] и сетей на базе открытого стандарта InfiniBand (Mellanox, Intel True Scale Fabric), как правило, аппаратно реализуют примитивы удалённого прямого чтения и записи памяти ЭМ системы (RDMA – Remote Direct Memory Access), а также фиксированный набор атомарных операций. На базе этих примитивов, контролеры могу аппаратно реализовывать и коллективные (глобальные) операции информационных обменов между ЭМ системы.

Анализ основных методов параллельного моделирования физико-технических процессов и природных явлений, а также параллельного решения информационно-вычислительных задач показывает, что для их реализации используется две схемы информационных обменов: дифференцированные обмены (индивидуальные) между парой процессов и коллективные обмены (глобальные, групповые), в которых участвуют все процессы параллельной программы.

Реализация указанных схем обменов возложена на коммуникационную сеть вычислительной системы и системы параллельного программирования: стандарт MPI, runtime-системы языков параллельного программирования семейства PGAS (IBM X10, Cray Chapel), Unified Parallel C.

В данной работе рассматриваются различные аспекты реализации алгоритмов коллективных обменов в ВС. Среди основных критериев их оптимизации выделены: время выполнения обменов, время выполнения вычислений в ходе коллективной операции и объем памяти, необходимый каждому процессу.

Приведены результаты анализа статистики использования коллективных операций стандарта MPI в популярных пакетах моделирования физико-технических процессов и природных явлений: CP2K, Quantum Espresso, VASP, DL-POLY, Amber, AMR, ANSYS, CPMD, WRF, GROMACS, LAMMPS, MILC. Выбор данных пакетов обусловлен их активным использованием на текущих большемасштабных ВС. Динамические характеристики указанных пакетов и схемы реализуемых ими обменов целесообразно принять во внимание при разработке инструментальных средств параллельного программирования перспективных ВС. В результате анализа пакетов сформировано базисное подмножество коллективных операций, требующих эффективной реализации в большемасштабных ВС.

Представлены основные подходы к реализации алгоритмов коллективных операций на базе примитивов дифференцированных обменов.

1. Операции коллективных обменов. Среди основных схем информационных обменов значительное место по частоте использования и приходящемуся на них суммарному времени выполнения занимают коллективные операции обмена информацией (групповые, глобальные, collective communications). Эффективная реализация коллективах операций в значительной степени влияет на масштабируемость параллельных алгоритмов и программ [15].

Стандарт MPI 3.1 включает 17 коллективных операции, каждая их них представлена в двух версиях – блокирующей и неблокирующей (non-blocking collective). Неблокирующие версии коллективных операций введены в стандарта MPI 3.0 и ориентированы на оптимизацию совмещения вычисления с коллективных обменом. В табл. 1 приведены результаты анализа частоты использования функций стандарта MPI в популярных пакетах суперкомпьютерного моделирования. Результаты получены путем анализа отчетов международного проекта HPC Advisory Council [16]. Функции перечислены в порядке убывания частоты вызовов и суммарного времени их выполнения. Для ряда пакетов приведены значения размеров сообщений, характерные для тестовых наборов данных, на которых выполнялось профилирование.

Таблица 1 – Частота использование MPI-функций в пакетах суперкомпьютерного моделирования и тестах производительности

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

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
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 системы линейных уравнений 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
HPCC FFT	1D быстрое преобразование Фурье	MPI_Alltoall, MPI_Bcast, MPI_Allreduce	MPI_Alltoall, MPI_Bcast, MPI_Allreduce	
HPCG	Решение системы линейных уравнений 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
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

Проведенный анализ использования функций MPI в промышленных пакетах моделирования позволяет выделить базисный набор 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.

Разработка эффективных алгоритмов реализации указанных операций на большемасштабных ВС является актуальной задачей.

2. Подходы к реализации алгоритмов коллективных обменов. Основная часть алгоритмов коллективных обменов реализуются на базе примитивов двусторонних обменов (send/recv) и основана на ряде фиксированных методов: рассылка данных по кольцу (ring), рекурсивное удваивание сообщения (recursive doubling) и расстояния между взаимодействующими процессами, рекурсивное деление сообщения пополам (recursive halving), алгоритм Брука (J. Bruck), попарные обмены (pairwise exchange) и методы, упорядочивающие процессы в деревья различных видов: биномиальные деревья, сбалансированные k-арные деревья, плоские деревья, конвейеры/цепочки [18].

При разработке и реализации алгоритмов коллективных операций стремятся достигнуть (суб)оптимальных значений одного или нескольких показателей эффективности:

1. Время выполнения алгоритма коллективной операции.

2. Сложность по памяти – объем дополнительной памяти, необходимый алгоритму для реализации информационных обменов и вычислений.

3. Энергоэффективность – потребление энергии вычислительной системой (элементарными машинами и коммуникационным оборудованием), при реализации алгоритма коллективной операции.

При разработке *оптимальных по времени выполнения* алгоритмов возникает необходимость уточнения понятия «время работы алгоритма». Здесь рассматривается реальное время выполнения реализованного алгоритма на заданной ВС, либо – оценка времени выполнения алгоритма в модели параллельных вычислений: Хокни, LogP/LogGP, BSP и др. [17, 18]. Кроме этого, могут уточняться границы эффективности алгоритмов для сообщений различных размеров. Например, время выполнения операции MPI_Allgather кольцевым алгоритмом и алгоритмом рекурсивного удваивания в модели Хокни есть:

$$T_{ring} = (p - 1)\alpha + (p - 1) / p \cdot m \cdot \beta,$$
$$T_{rdbl} = \log_2(p)\alpha + (p - 1) / p \cdot m \cdot \beta,$$

где α – латентность канала связи, β – время передачи одного байта сообщения, m – размер сообщения в байтах, p – число процессов. Из приведённых выражений видно, что алгоритма рекурсивного удваивания характеризуется меньшей латентностью и может быть рекомендован для сообщения небольших размеров.

Сложность алгоритма коллективной операции по памяти отражает использование дополнительной памяти для выполнения глобальной операции. Значительная часть алгоритмов создают дополнительный буфер для приема промежуточных сообщений. Кроме этого, ряд операции типа all-to-all оперируют информационными массивами, длина которых равна числу p процессов в программе (числу ЭМ в системе). Создание временных копий таких массивов (сообщений) может отразится на объеме доступной памяти для параллельной программы пользователей, а учитывая рост числа p ЭМ в перспективных ВС, может ограничить применимость алгоритмов объемом оперативной памяти на ЭМ системы.

Введенные в стандарта MPI 3.0 неблокирующие версии коллективных операций требует для своей реализации и выполнения предварительного создания расписания обменов каждого алгоритма – упорядоченного списка с описанием дифференцированного обмена для каждого шага. Такие списки могут требовать порядка $O(p)$ ячеек памяти и потенциально ограничивают применимость таких алгоритмов в большемасштабных ВС.

Специализированные сетевые контролеры, допускающую выгрузку на них части коммуникационных операций MPI/PGAS/SHMEM (network offload), имеют ограниченный объем памяти. Поэтому для ВС на базе таких коммуникационных сетей требуется разработка алгоритмов, характеризующихся относительно низкой сложностью по памяти – порядка $O(\log(p))$ и ниже.

Энергоэффективность алгоритмов оценивается эмпирически или в моделях, с учетом структуры обменов. Основана идея – минимизировать количество используемых вычислительных и коммуникационных устройств (сетевых контроллеров и транзитных коммутаторов/маршрутизаторов), задействованных в обмене.

На практике разработать алгоритм доставляющий оптимальное значение всем указанным критериям проблематично. Поэтому один из критериев фиксируется как основной, а остальные записываются в виде ограничений.

Заключение. Время реализации коллективных операций информационных обменов MPI/PGAS/SHMEM в значительной степени влияет на масштабируемость параллельных алгоритмов и программ. Проведенный анализ использования функций MPI в промышленных пакетах моделирования CP2K, Quantum Espresso, VASP, DL-POLY, Amber, AMR, ANSYS, CPMD, WRF, GROMACS, LAMMPS, MILC показал, что определяющее значение на масштабируемость MPI-программ оказывает выделенный базисный набор операций: MPI_Allreduce, MPI_Alltoallv, MPI_Alltoall, MPI_Barrier, MPI_Allgather, MPI_Allgatherv; MPI_Bcast, MPI_Gather, MPI_Scatter.

Разработка эффективных алгоритмов реализации указанных операций для большемасштабных ВС является актуальной задачей. При этом основными критериями оптимизации алгоритмов коллективных обменов являются: время выполнения алгоритма, сложность по памяти и энергоэффективность.

Список литературы

1. Евреинов Э.В., Хорошевский В.Г. Однородные вычислительные системы. – Новосибирск: Наука, 1978. – 318 с.

2. Каляев А.В. Многопроцессорные системы с программируемой. – Москва: Радио и связь, 1984. – 240 с.

3. Миренков Н.Н. Параллельное программирование для многопроцессорных вычислительных систем. – Москва: Радио и связь, 1989. – 320 с.

4. Вальковский В.А., Малышкин В.Э. Синтез параллельных программ и систем на вычислительных моделях. – Новосибирск: Наука, 1988. – 129 с.

5. Корнеев В.В. Параллельные вычислительные системы. – Москва: Нолидж, 1999. – 320 с.

6. Монахов О.Г., Монахова Э.А. Параллельные системы с распределенной памятью: структуры и организация взаимодействий. – Новосибирск: ИВМиМГ СО РАН, 2000. – 242 с.

7. Воеводин В.В., Воеводин Вл.В. Параллельные вычисления. – СПб.: БХВ-Петербург, 2002. – 608 с.

8. Хорошевский В.Г. Распределенные вычислительные системы с программируемой структурой // Вестник СибГУТИ. – 2010. – № 2. – С. 3-41.

9. Степаненко С.А. Мультипроцессорные среды суперЭВМ. Масштабирование эффективности. – М.: ФИЗМАТЛИТ, 2016. – 312 с.

10. Dongarra J., Beckman P., Moore T. et al. International Exascale Software Project Roadmap // The International Journal of High Performance Computing Applications. – 2011. – Vol. 25, Issue 1. – pp. 3-60.

11. Alverson R., Roweth D., Kaplan L. The Gemini System Interconnect // International Symposium on High Performance Interconnects. – 2010. – pp. 83–87.

12. Eisley N., Heidelberger P., Senger R. The IBM Blue Gene/Q interconnection network and message unit // International Conference for High Performance Computing, Networking, Storage and Analysis. – 2011. – pp. 1–10.

13. Левин В.К., Четверушкин Б.Н., Елизаров Г.С., Горбунов В.С., Лацис А.О., Корнеев В.В., Соколов А.А., Андрюшин Д.В., Климов Ю.А. Коммуникационная сетьМВС-Экспресс // Информационные технологии и вычислительные системы. – 2014. – № 1. – С. 10–24.

14. Симонов А.С., Макагон Д.В., Жабин И.А., Щербак А.Н., Сыромятников Е.Л., Поляков Д.А. Первое поколение высокоскоростной коммуникационной сети «Ангара» // Наукоемкие технологии. – 2014. – Т. 15. No 1. – С. 21–28.

15. Balaji P. MPI on Millions of Cores / Balaji P., Buntinas D., Goodell D., Gropp W., Hoefler T., Kumar S., Lusk E., Thakur R., Traff J. // Parallel Processing Letters. – 2011. – Vol. 21, Issue 1. – P. 45-60.

16. HPC Advisory Council Best Practices // URL: http://hpcadvisorycouncil.com/best_practices.php

17. Hoefler T., Moor D. Energy, Memory, and Runtime Tradeoffs for Implementing Collective Communication Operations // Journal of Supercomputing Frontiers and Innovations. 2014. Vol 1, No. 2, P. 58-75.

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

Курносов Михаил Георгиевич – д-р техн. наук, доцент, заведующий кафедрой Сибирского государственного университета телекоммуникаций и информатики; 630102, Новосибирск, e-mail: mkurnosov@gmail.com