**Сравнительный анализ эффективности библиотек динамического выделения памяти**

И. А. Киреев, М. Г. Курносов[[1]](#footnote-2)

Проведено экспериментальное исследование алгоритмов глобальной редукции стандарта MPI на вычислительном кластере с сетью связи Gigabit Ethernet. Определены оптимальные алгоритмы глобальной редукции. Эксперименты проведены на вычислительном кластере с сетью связи Gigabit Ethernet.

*Ключевые слова*: глобальная редукция, allreduce, MPI, вычислительные системы.

1. Введение

Библиотеки динамического выделения памяти — инструменты для управления динамическим выделением памяти в соответствии с задачами приложения, в котором используются. Библиотеки динамического выделения памяти предоставляют эффективные алгоритмы контроля памяти, а также функции управления блоками памяти, её фрагментации для оптимизации производительности. Одними наиболее распространёнными библиотеками динамического выделения памяти являются: *ptmalloc2*, *jemalloc*, *tcmalloc*.

Цель исследования — выполнить сравнительный анализ эффективности библиотек динамического выделения памяти: *ptmalloc2*, *jemalloc*, *tcmalloc*.

2. Библиотеки динамического выделения памяти (различия?)

Основными пользовательскими функциями в библиотеках динамического выделения памяти являются: функция выделения (резервирования) памяти, функция освобождения памяти. В общем случае функция malloc выделения памяти резервирует блок памяти размера *n* и возвращает указатель на *void,* который указывает на начало блока зарезервированной памяти.

void \*malloc(size\_t n)

Основной функцией для очистки памяти является функция free*,* которая принимает указатель *ptr* на начало выделенной памяти и освобождает занимаемый блок

void free(void \*ptr)

Рассматриваемые библиотеки динамического выделения памяти имеет собственные механизмы её контроля. Каждая из исследуемых библиотек использует механизм выделения блоков памяти для каждого потока, что уменьшает вероятность возникновения конкуренции за общие ресурсы. Из рассматриваемых библиотек динамического выделения памяти, стандартной для большинства систем на базе Linux является *ptmalloc2*.

3. Результаты экспериментов

Эксперименты выполнены на вычислительном кластере из 10 узлов. Каждый узел содержит два процессора Intel Xeon Quad (4 ядра). Сеть связи Gigabit Ethernet. Операционная система GNU/Linux, библиотека Open MPI 5.0.0. В качестве теста производительности использован тест OSU MPI Micro-Benchmarks 5.8. За время выполнения бралась максимальная латентность (max latency).

На рис. 1 показано времени выполнения алгоритмов операции MPI\_Allreduce от размера сообщения. Алгоритм *basic\_linear* показывает худшую производительность, так как время его выполнения линейно зависит от числа *p* процессов. Алгоритм *nonoverlapping* показывает наименьшее время выполнения для сообщений, размер которых не превышает 8 Кбайт. Для сообщений больше 8 Кбайт рекомендуется использовать алгоритм *ring*.

|  |  |
| --- | --- |
| а) | б) |
| в) | г) |

Рис. 1. Время выполнения алгоритмов MPI\_Allreduce (80 процессов, тип данных MPI\_FLOAT, бинарная операция MPI\_SUM): а) алгоритм basic\_linear; б) алгоритм nonoverlapping; в) алгоритм recursive\_doubling; г) алгоритм ring

На рис. 2 показано время выполнения алгоритма *segmented\_ring* операции MPI\_Allreduce от размера сообщения для различных размеров *S* сегмента. Алгоритм разбивает исходное сообщение на сегменты по *S* байт и передает их по кольцу. Для сообщений от 64 Кбайт до 256 Кбайт рекомендуется использовать размер сегмента от 16 Кбайт до 64 Кбайт. При б*о*льших размерах сообщений целесообразно выбирать *S* = 64 Кбайт.

Для размеров сообщений меньших 8 Кбайт рекомендуется использовать алгоритм *nonoverlapping*. Для сообщений, размер которых превышает 8 Кбайт, наиболее эффективен алгоритм *segmented\_ring.*

|  |  |
| --- | --- |
|  |  |
|  | |

Рис. 2. Время выполнения алгоритма *segmented\_ring* для операции MPI\_Allreduce  
(80 процессов – 10 узлов по 8 ядер, тип MPI\_FLOAT, бинарная операция MPI\_SUM)

**4. Заключение**

Проведено экспериментальное исследование алгоритмов глобальной редукции стандарта MPI на вычислительном кластере с сетью связи стандарта Gigabit Ethernet и библиотекой Open MPI 5.0.0. На тестах OSU Mirco-Benchmarks показано, что при выполнении глобальной редукции для сообщений меньше 8 Кбайт рекомендуется использовать алгоритм *nonoverlapping* (Reduce + Bcast). В случае сообщений больше 8 Кбайт, наиболее эффективен алгоритм *segmented\_ring,* который выполняет автоматическую сегментацию сообщения.

**Литература**

1. MPI-4.0 Standard // URL: <https://www.mpi-forum.org/docs/mpi-4.0/mpi40-report.pdf> (дата обращения: 15.03.2022).
2. Open MPI // URL: <https://www.open-mpi.org/> (дата обращения: 15.03.2022).
3. OSU Micro-Benchmarks // URL: <https://mvapich.cse.ohio-state.edu/benchmarks/> (дата обращения 15.03.2022)
4. *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.
5. *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.

**Киреев Илья Анатольевич**

студент группы ИВ-121 СибГУТИ (630102, Новосибирск, ул. Кирова, 86), тел. (383) 269-83-82, e-mail: ilya\_kireev\_2004[@bk.](mailto:rmashkalev054@gmail.com)ru.

**Курносов Михаил Георгиевич**

д.т.н., профессор, профессор кафедры вычислительных систем Сибирского государственного университета телекоммуникаций и информатики (630102, г. Новосибирск, ул. Кирова, 86), тел. (383) 269-83-82, e-mail: [mkurnosov@sibguti.ru](mailto:mkurnosov@sibguti.ru); старший научный сотрудник федерального государственного бюджетного учреждения науки Института физики полупроводников им. А.В. Ржанова Сибирского отделения Российской академии наук (630090, г. Новосибиpск, пp. Ак. Лавpентьева, 13), тел. (383) 330-56-26, e-mail: mkurnosov@isp.nsc.ru.

**Experimental Study of Allreduce Algorithms on a Multicore Cluster**

**R. Mashkalev, M. Kurnosov**

An experimental study of global reduction algorithms on a computing cluster has been carried out. Optimal algorithms for global reduction are determined. The obtained theoretical estimates are supported by the results of experiments on a computing cluster with a Gigabit Ethernet communication network.

*Keywords*: global reduction, allreduce, MPI, computing systems.

1. Исследования выполнены в рамках госзадания ИФП СО РАН (0242-2021-0011). [↑](#footnote-ref-2)