Методы реализации односторонних обменов стандарта MPI

Аненков Александр Дмитриевич

Сибирский государственный университет телекоммуникаций и информатики

E-mail: alex.anenkov@outlook.com

Ежегодная российская научно-техническая конференция

«Обработка информации и математическое моделирование»

г. Новосибирск, 2017

Стандарт MPI

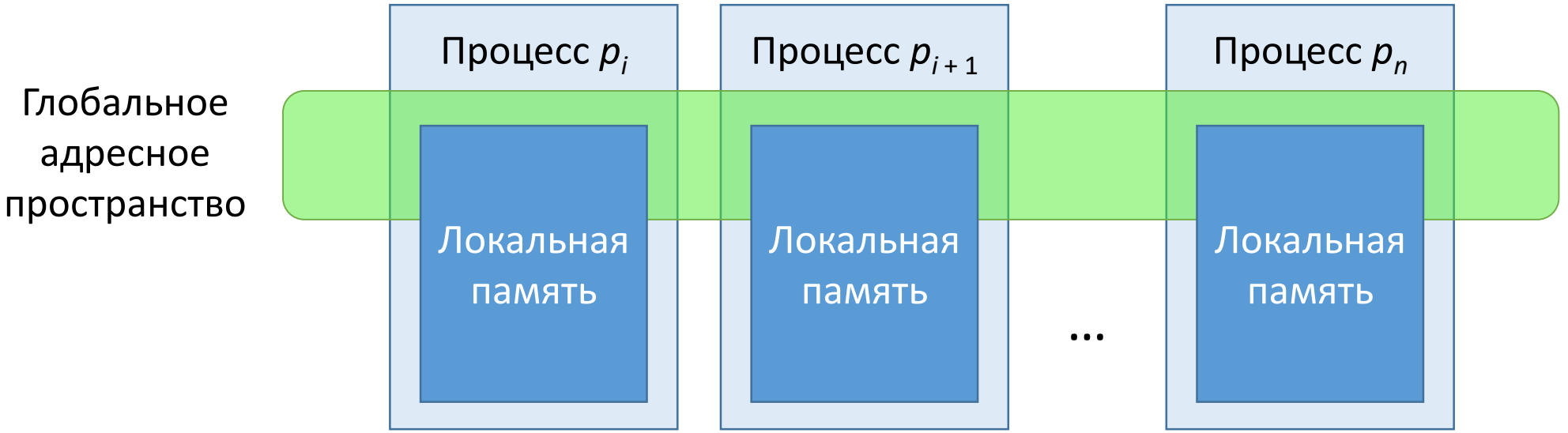
- **Message Passing Interface (MPI)** – это стандарт на программный интерфейс коммуникационных библиотек для создания параллельных программ в модели передачи сообщений.
- Стандарт определяет интерфейс для языков программирования C и Fortran.
- Стандарт де-факто для систем с распределенной памятью.
- Обеспечивает переносимость программ на уровне исходного кода между разными ВС (Cray, IBM, NEC, Fujitsu, ...).

http://www.mpi-forum.org/

http://mpi-forum.org/docs/mpi-3.1/mpi31-report.pdf

Односторонние обмены

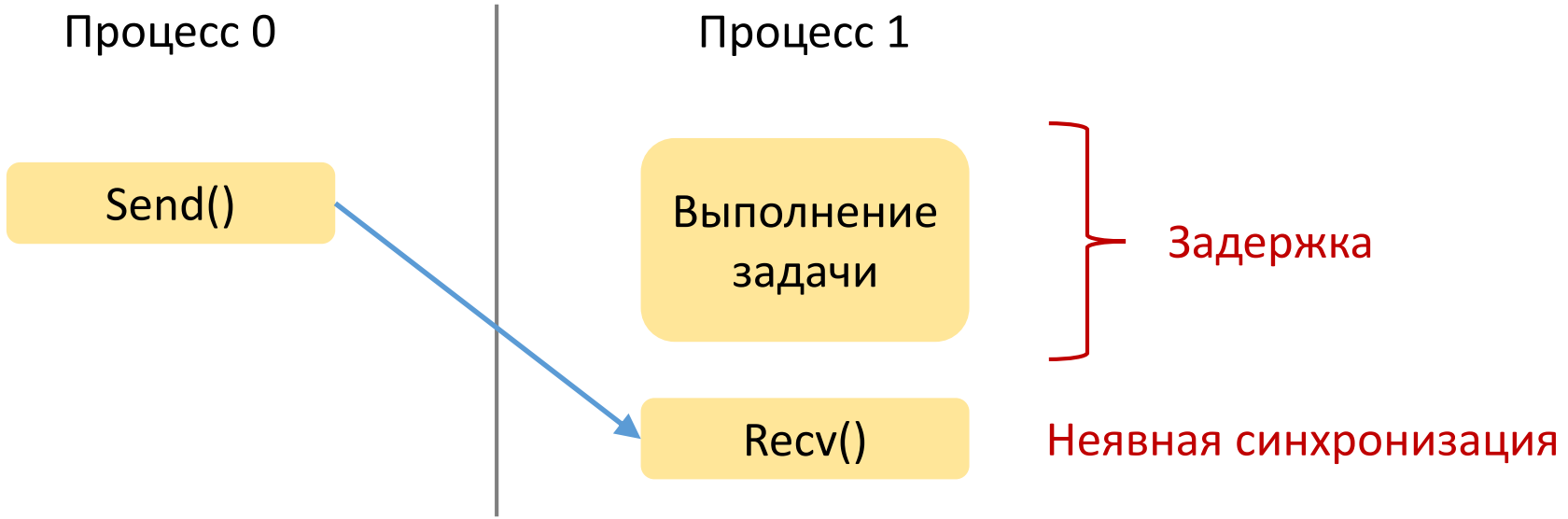
Имеется вычислительная система в которой выполняется p_n процессов. Каждый процесс выделяет часть своей локальной памяти под глобальное адресное пространство, благодаря чему любой процесс может осуществлять доступ к памяти другого процесса (RMA – Remote Memory Access).



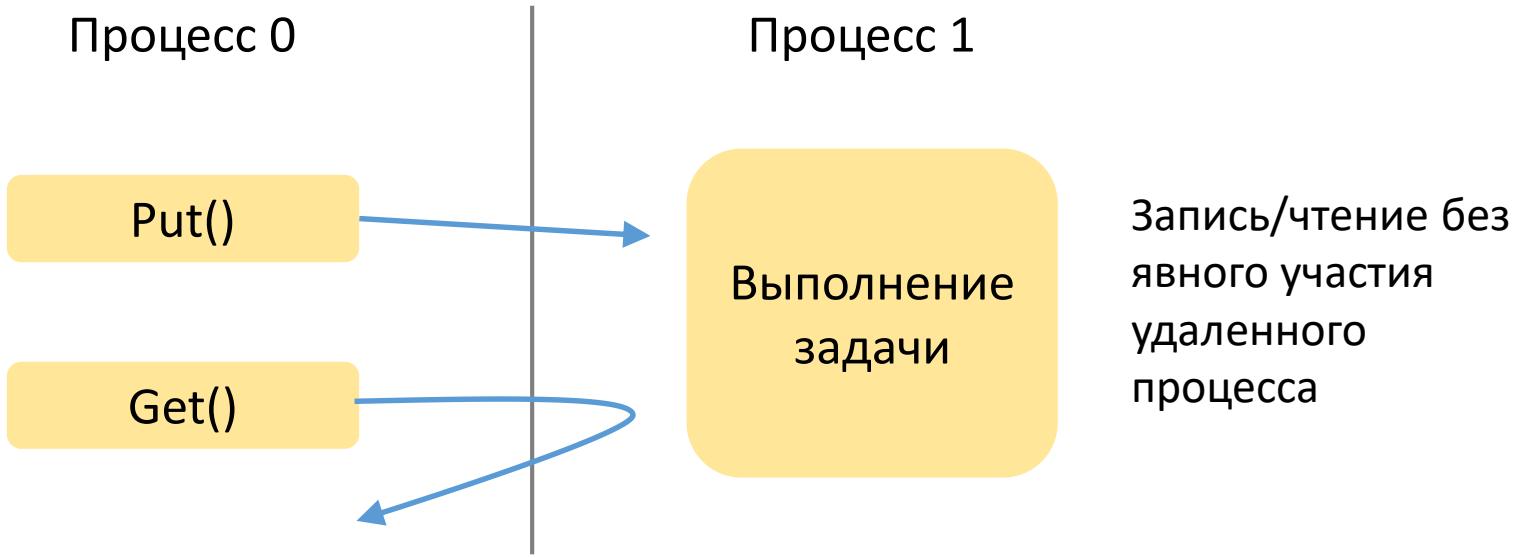
Основная идея модели односторонних обменов заключается в обмене данными между процессами без необходимости явного участия удаленного процесса в процедуре синхронизации.

Отличие от двусторонних обменов

Двусторонний обмен



Односторонний обмен



Основные преимущества RMA-операций

- Возможность выполнения множественных передач данных, используя синхронизацию лишь единожды, что повышает производительность и масштабируемость.
- Возможность выбора степени участия процессов в обмене данными: процесс может быть, как активным участником обмена, так и пассивным.
- Облегчают создание параллельных программ с динамически изменяющимися моделями доступа к данным, где распределение данных фиксировано или медленно изменяется.
- Наличие аппаратной поддержки прямого доступа к памяти (RDMA) в сетях InfiniBand, IBM PERCS, BlueGene/Q, «Ангара» и др., позволяющей получать непосредственный прямой доступ к памяти других удаленных процессов без вмешательства в ход их выполнения и без участия ОС.

Woodall T., Shipman G., Bosilca G., Maccabe A. **High performance RDMA protocols in HPC** // Proceedings of the 13th European PVM/MPI User's Group conference on Recent advances in parallel virtual machine and message passing interface (EuroPVM/MPI'06). Springer-Verlag, Berlin, Heidelberg. 2006. P. 76–85.

Порядок работы с MPI RMA

1. Определение процессами области памяти, которая будет доступна для других процессов
2. Обмен данными
3. Синхронизация
4. Освобождение общедоступной памяти

MPI_Win_create()

…

MPI_Win_lock()

…

MPI_Put()
MPI_Get()

…

MPI_Win_unlock()

…

MPI_Win_free()

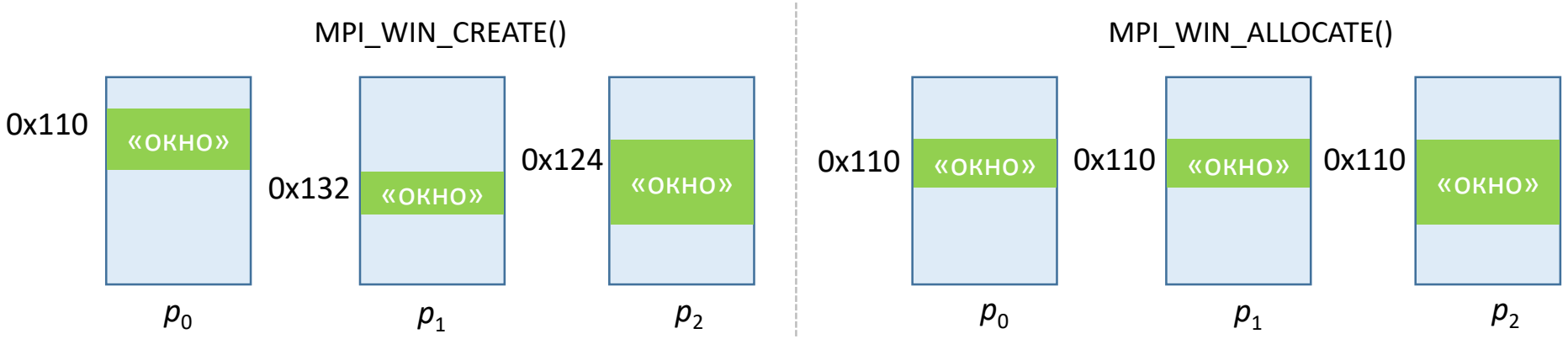
Hoefler T., Dinan J., Thakur R., Barret B., Balaji P., Gropp W., Underwood K. **Remote Memory Access Programming in MPI-3** //
ACM Transactions on Parallel Computing. Volume 2 Issue 2. July 2015. Article No. 9.

Память, доступная другим процессам

По умолчанию любая память в рамках процесса является локальной т.е. недоступной для других.

Для того чтобы другие процессы имели возможность осуществить удаленный доступ к данному процессу, необходимо выполнить MPI вызов, объявляющий некоторую область памяти процесса как общедоступную.

Смежная область памяти каждого процесса, которая может использоваться в RMA-операциях, называется **«окном»**.



Все процессы в рамках окна могут выполнять чтение и запись данных в память любого другого процесса явно и неявно в зависимости от модели синхронизации.

Создание «окна»

Стандарт MPI предлагает следующие основные коллективные вызовы для создания «окна»:

MPI_WIN_CREATE:

Рекомендуется к использованию, если существует уже ранее выделенный буфер.

Недостаток: Возможность указания произвольных областей памяти может привести к большим таблицам трансляции и падению производительности в системах с поддержкой RDMA.

MPI_WIN_ALLOCATE:

Аналогично MPI_WIN_CREATE. Позволяет выделить новую область памяти.

Преимущество: Память выделяется по идентичным адресам во всех процессах (т.е. симметрично), что позволяет избежать больших таблиц трансляций в системах с поддержкой RDMA.

Создание «окна»

Стандарт MPI предлагает следующие основные функции для создания «окна»:

MPI_WIN_ALLOCATE_SHARED:

Создание «окна» в разделенной области памяти. Рекомендуется использовать, если процессы расположены на одном узле.

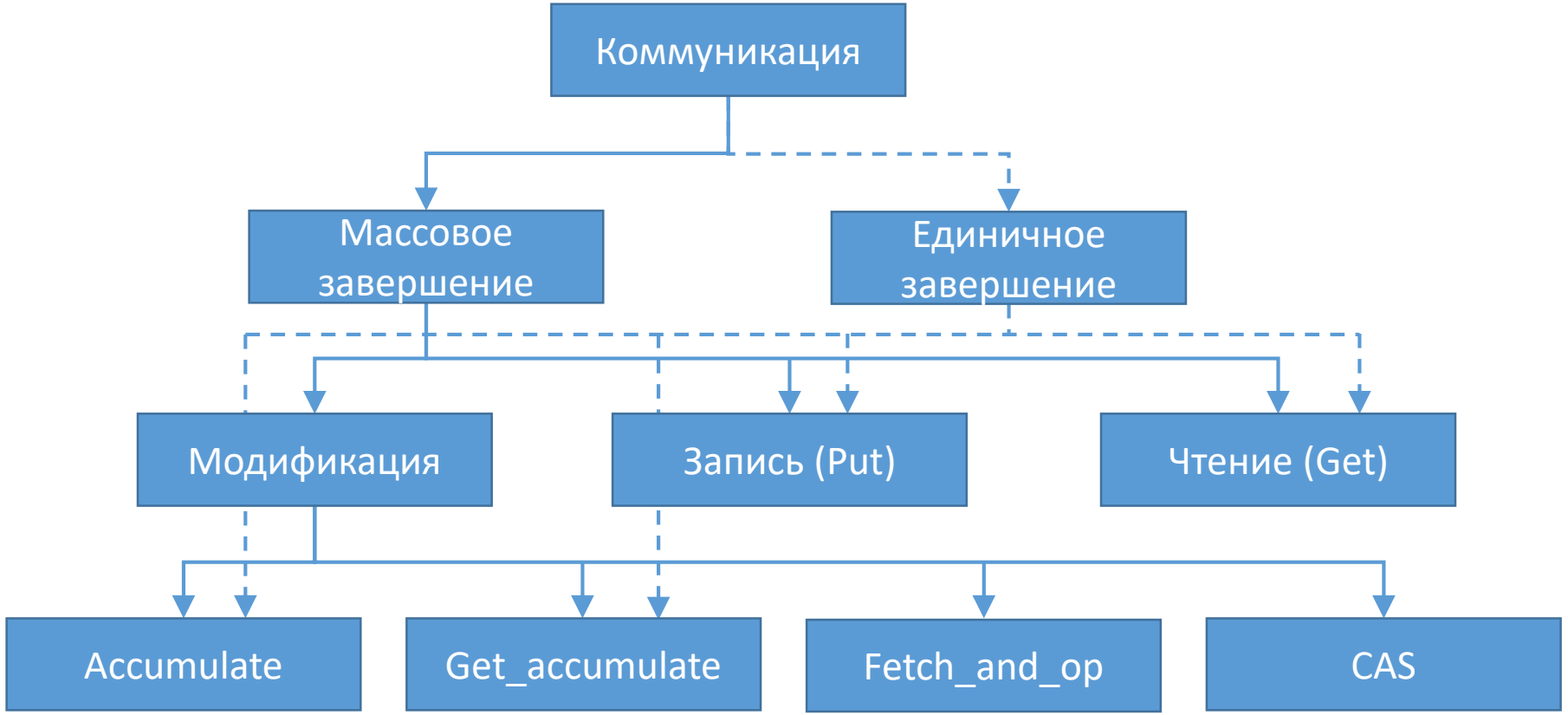
Преимущество: Позволяет снизить накладные расходы связанные с коммуникацией и организацией доступа к памяти через MPI-прослойку.

MPI_WIN_ALLOCATE_DYNAMIC:

За создаваемым «окном» изначально не закрепляется какая-либо область памяти. Закрепить её можно позже.

Обмен данными

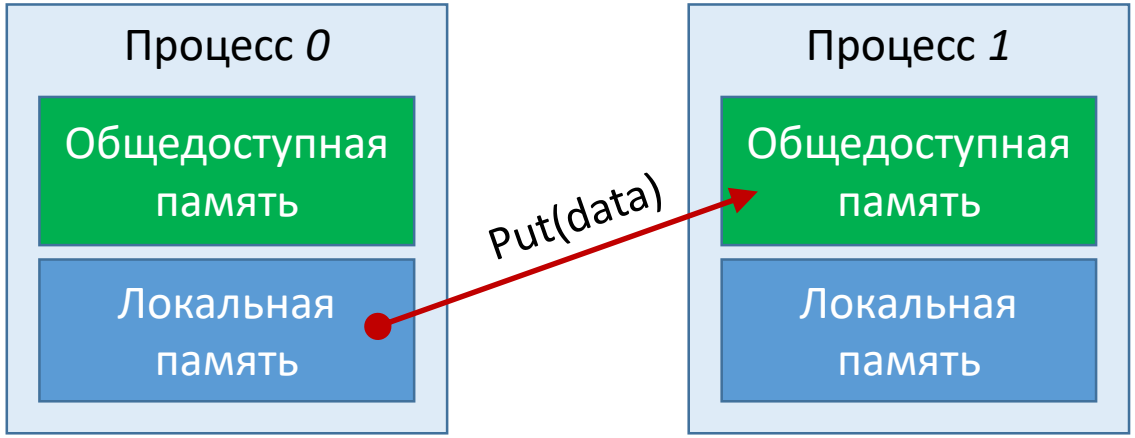
В рамках «окна» MPI предоставляет возможность чтения, записи и атомарной модификации данных при помощи следующих вызовов:



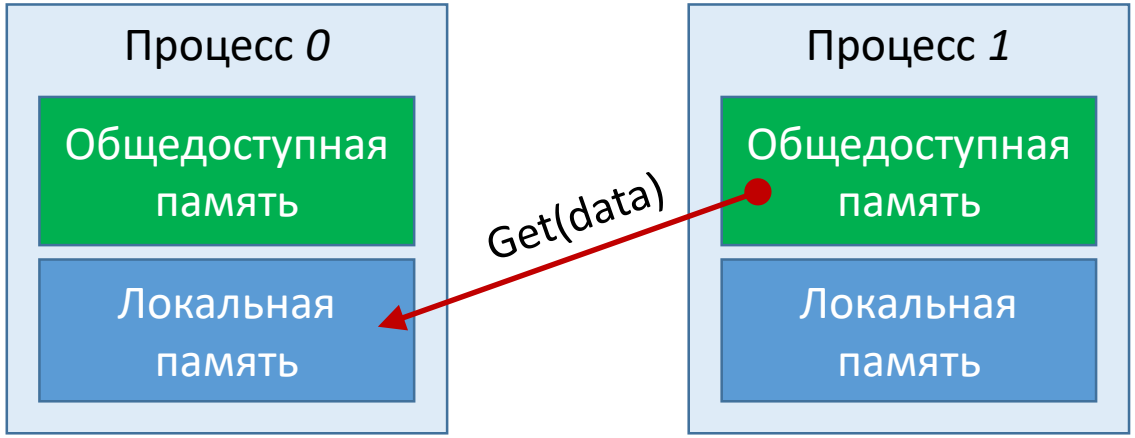
Все операции – **неблокирующие**.

Перемещение данных: Put / Get

MPI_PUT: Запись данных из памяти процесса-инициатора в «окно» процесса-адресата



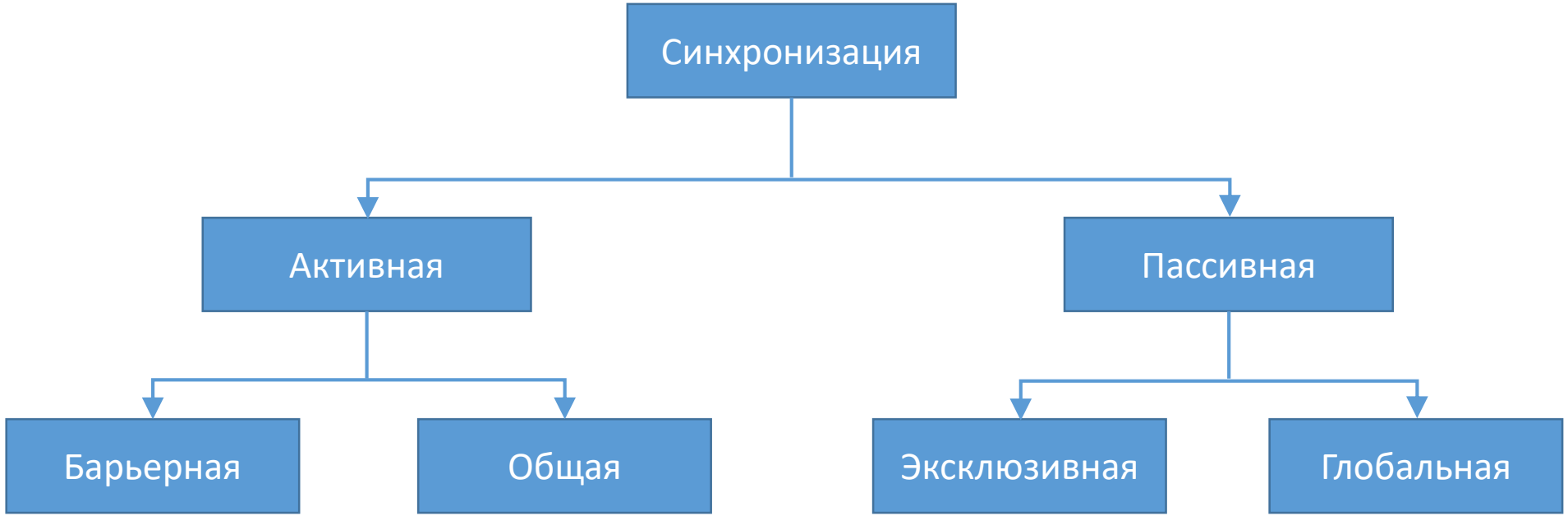
MPI_GET: Чтение данных из «окна» процесса-адресата в буфер процесса-инициатора



Упорядочивание RMA-операций

- Порядок выполнения операций чтения/записи не гарантирован.
- Возможна гонка данных при записи по одному и тому же адресу, а результат при чтении будет неопределенным.
- Результат конкуренции операций атомарных модификаций данных по одному и тому же адресу зависит от используемых параметров. В качестве оптимизации пользователь может отключить упорядочивание или указать любое другое (RAW – read-after-write, WAR, RAR, или WAW).

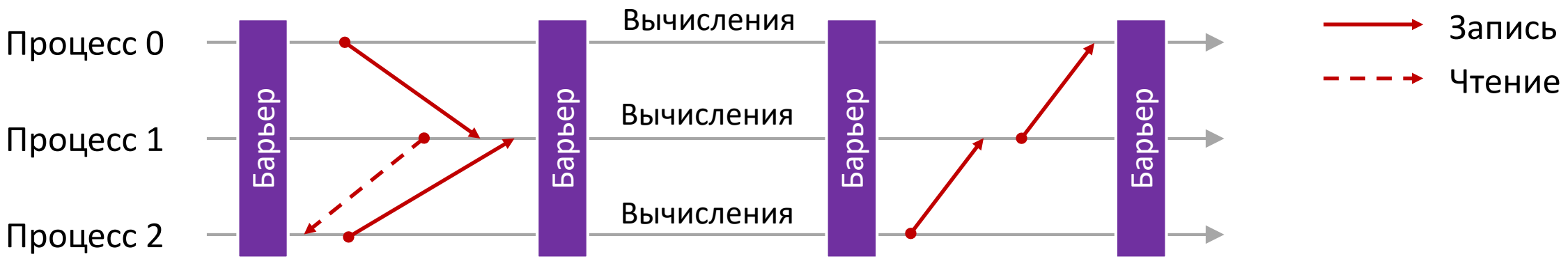
Механизмы синхронизации



При **активной** синхронизации данные перемещаются из памяти одного процесса в память другого, при этом *оба процесса* явно участвуют в процессе передачи.

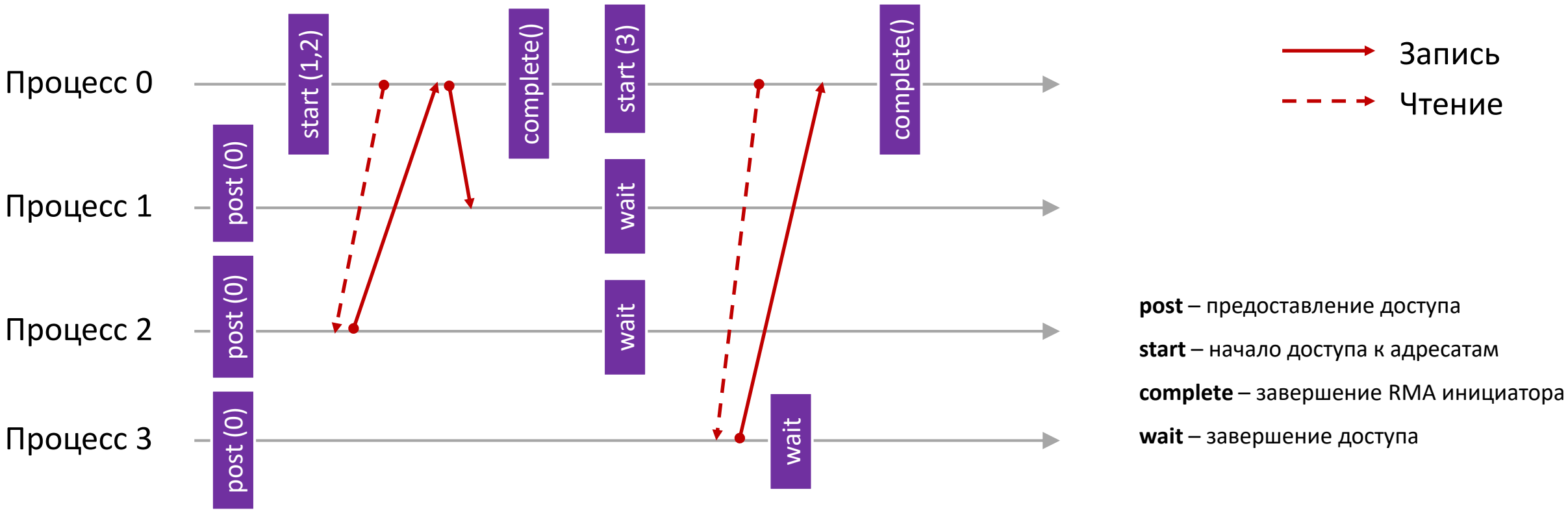
При **пассивной** синхронизации данные перемещаются из памяти одного процесса в память другого, но *только процесс-инициатор* обмена явно участвует в передаче. При этом несколько процессов могут взаимодействовать при доступе к одному удаленному «окну».

Барьерная синхронизация



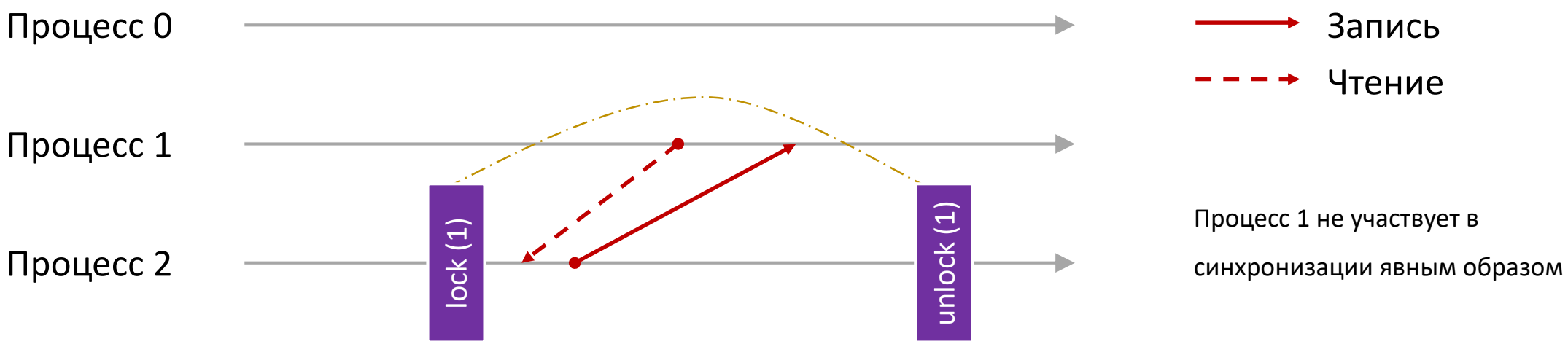
- Вызов функции MPI_WIN_FENCE – коллективная синхронизация.
- Рекомендуется использовать в параллельных вычислениях, в которых крупные блоки вычислений чередуются с крупными блоками межпроцессных обменов.
- Механизм полезен для слабосинхронных алгоритмов, в которых граф межпроцессных связей меняется очень часто, или в которых каждый процесс взаимодействует со множеством других.
- Пример – задачи поиска в графе.

Общая синхронизация



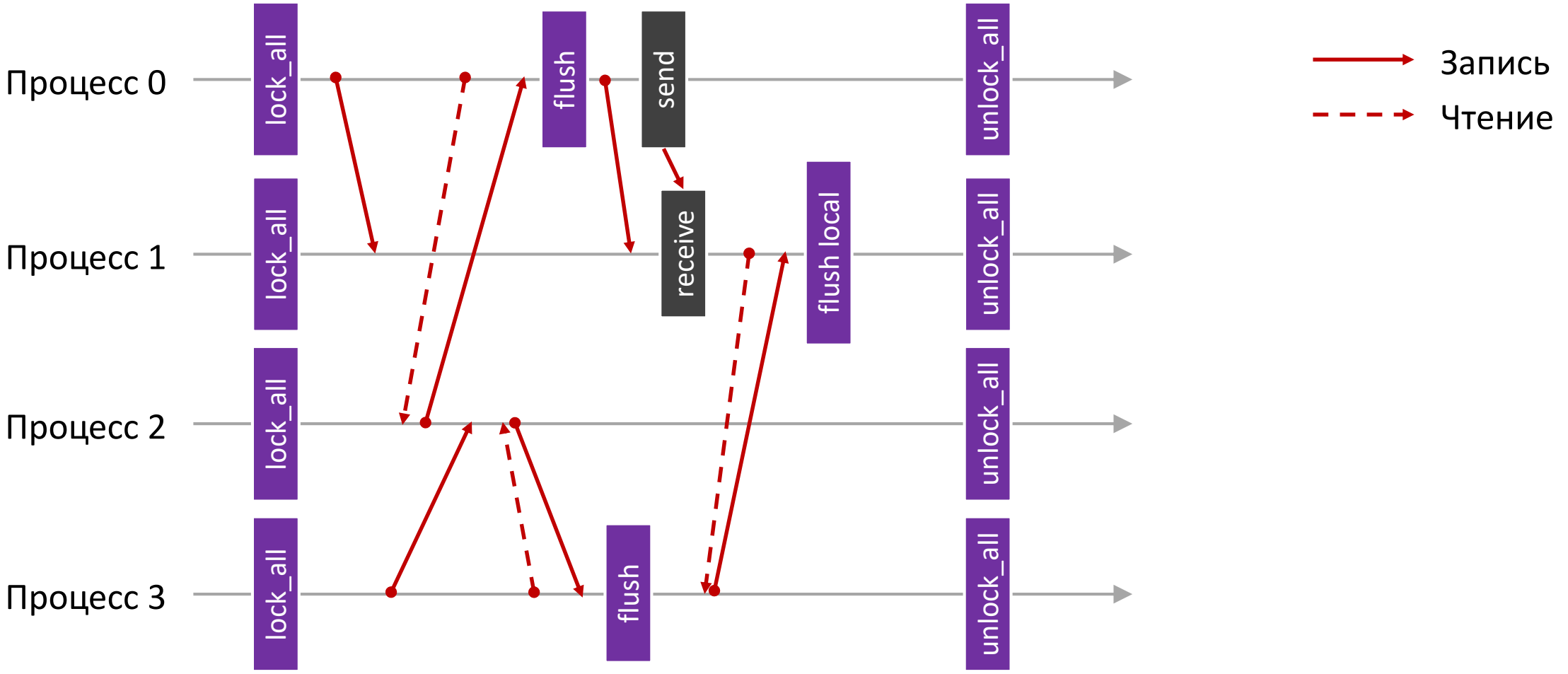
- Функции MPI_WIN_START, MPI_WIN_COMPLETE, MPI_WIN_POST, MPI_WIN_WAIT могут быть использованы для сведения синхронизации к минимуму: синхронизируются только пары взаимодействующих процессов.
- Механизм наиболее эффективен, когда каждый процесс обменивается данными с несколькими соседями, а граф межпроцессных связей фиксирован или изменяется нечасто.

Эксклюзивная синхронизация



- Один процесс блокирует «окно» удаленного процесса перед тем как осуществить доступ.
- Эксклюзивная синхронизация полезна для MPI приложений, которые эмулируют модель с общей памятью через MPI вызовы.
- Пример – модель «доска объявлений» (billboard), в которой процессы в случайные промежутки времени могут обращаться или обновлять различные части «доски объявлений».

Глобальная синхронизация



MPI_WIN_LOCK_ALL – блокировка «окон» всех процессов
MPI_WIN_FLUSH – завершение RMA операций
MPI_WIN_UNLOCK_ALL – разблокировка «окон» всех процессов

Глобальная синхронизация

- Каждый процесс блокирует все (lock all) периоды доступа ко всем другим процессам.
- В отличие от барьерной синхронизации, глобальная синхронизация не является коллективной.
- Процессы могут взаимодействовать друг с другом при помощи RMA операций для обновления данных и использовать операции синхронизации или двусторонние обмены для уведомлений.
- Данный механизм может быть использован для реализации мелкозернистых (fine-grained) блокировок, таких как MCS (Mellor-Crummey and Scott lock).

Mellor-Crummey J., Scott M.. **Algorithms for Scalable Synchronization on Shared-Memory Multiprocessors** // ACM Transactions on Computer Systems, 9(1):21-65, Feb. 1991.

Области применения односторонних обменов

Примеры:

- Быстрое преобразование Фурье
- Решение дифференциальных уравнений методом конечных разностей
- Другие трафаретные вычисления
- Использование атомарных операций для реализации распределенных, асинхронных, неблокируемых структур данных (например, неблокируемая очередь используемая в алгоритме MSC-блокировки)

Климов Ю. А., Орлов А. Ю., Шворин А. Б. **Программный инструментарий для трафаретных вычислений на гибридных суперкомпьютерах** // Программные системы: теория и приложения : электрон. научн. журн. 2012. Т. 3, № 2(11), с. 23–49.

Song S., Hollingsworth J. K. **Designing and auto-tuning parallel 3-D FFT for computation-communication overlap** // Proceedings of the 19th ACM SIGPLAN symposium on Principles and practice of parallel programming. 2014. P. 181-192.

Mellor-Crummey J., Scott M.. **Algorithms for Scalable Synchronization on Shared-Memory Multiprocessors** // ACM Transactions on Computer Systems, 9(1):21-65, Feb. 1991.

Спасибо за внимание!

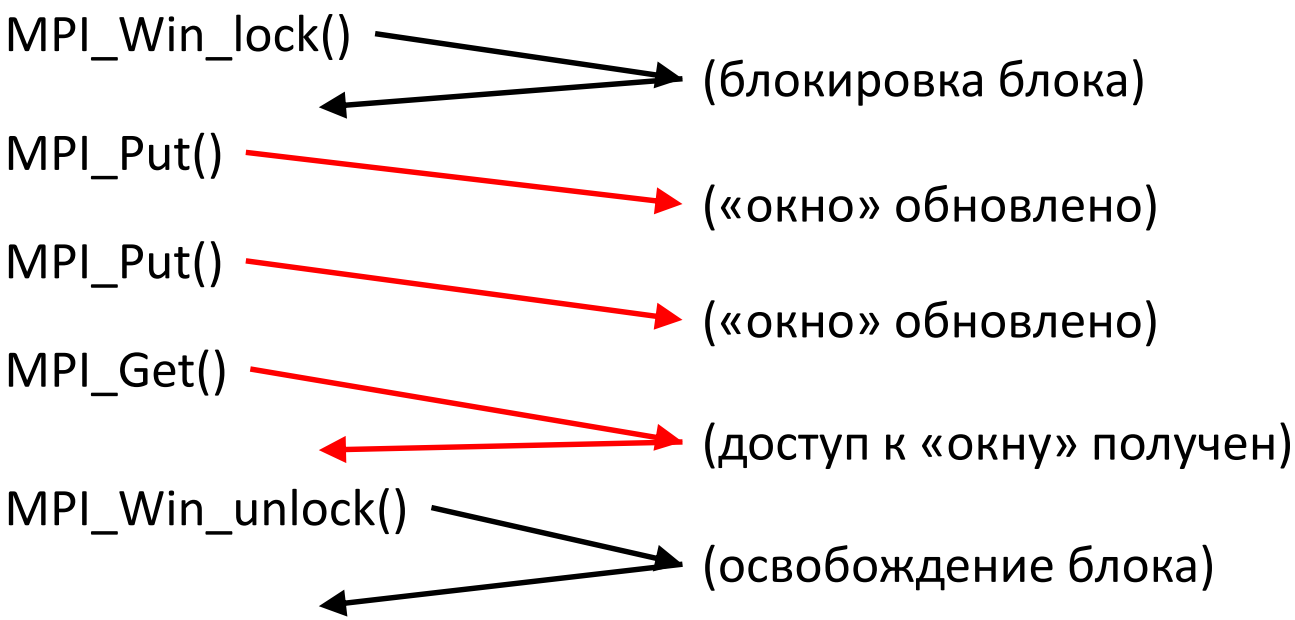
Другие модели односторонних обменов

- **UPC (Unified Parallel C)**

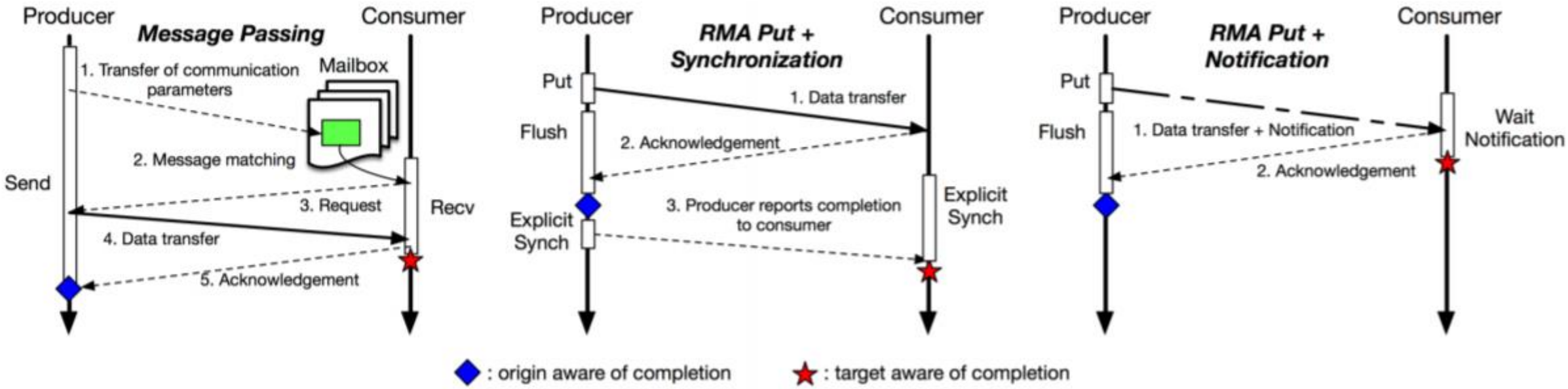
  Использование модели распределенной общей памяти (DSM – Distributed Shared Memory) или виртуальной общей памяти (VSM – Virtual Shared Memory)

- **ARMCI (Aggregate Remote Memory Copy Interface)**
- **Другие**

Перемещение данных



Идея дальнейшего развития RMA



Hoefler T. **Active RDMA - new tricks for an old dog** // (Presentation) presented in Gleneden Beach, OR, USA, Apr. 2016, Invited talk at Salishan Meeting.

Литература по теме

- MPI: **A Message-Passing Interface Standard. Version 3.1**. URL: http://mpi-forum.org/docs/mpi-3.1/mpi31-report.pdf
- Hoefler T., Dinan J., Thakur R., Barret B., Balaji P., Gropp W., Underwood K. **Remote Memory Access Programming in MPI-3** // ACM Transactions on Parallel Computing. Volume 2 Issue 2. July 2015. Article No. 9.
- Gerstenberger R., Besta M., Hoefler T. **Enabling Highly-scalable Remote Memory Access Programming with MPI-3 One Sided** // Proceedings of SC13: International Conference for High Performance Computing, Networking, Storage and Analysis (SC '13). ACM. New York, USA. Article 53. 2013.
- *Squyres J.* **The New MPI-3 Remote Memory Access (One Sided) Interface** // CISCO blogs. URL: http://blogs.cisco.com/performance/the-new-mpi-3-remote-memory-access-one-sided-interface
- *Hoefler T.* **Active RDMA - new tricks for an old dog** // (Presentation) presented in Gleneden Beach, OR, USA, Apr. 2016, Invited talk at Salishan Meeting.
- Jeff R. Hammond, Sayan Ghosh, Barbara M. Chapman. **Implementing OpenSHMEM using MPI-3 one-sided communication**