

Дополнительный материал



МЕДЛЕННЫЙ ВВОД/ВЫВОД

Операции ввода/вывода, несомненно, являются самыми медленными из всех основных операций в компьютере.

Доступ к ОЗУ занимает несколько наносекунд (10^{-9} секунды), а доступ к данным на диске или в сети занимает несколько милли секунд (10^{-3} секунды).

С пропускной способностью та же история: ОЗУ имеет скорость передачи порядка нескольких ГБ/с, в то время как для диска и сети эта скорость варьируется от нескольких МБ/с до гипотетического ГБ/с.

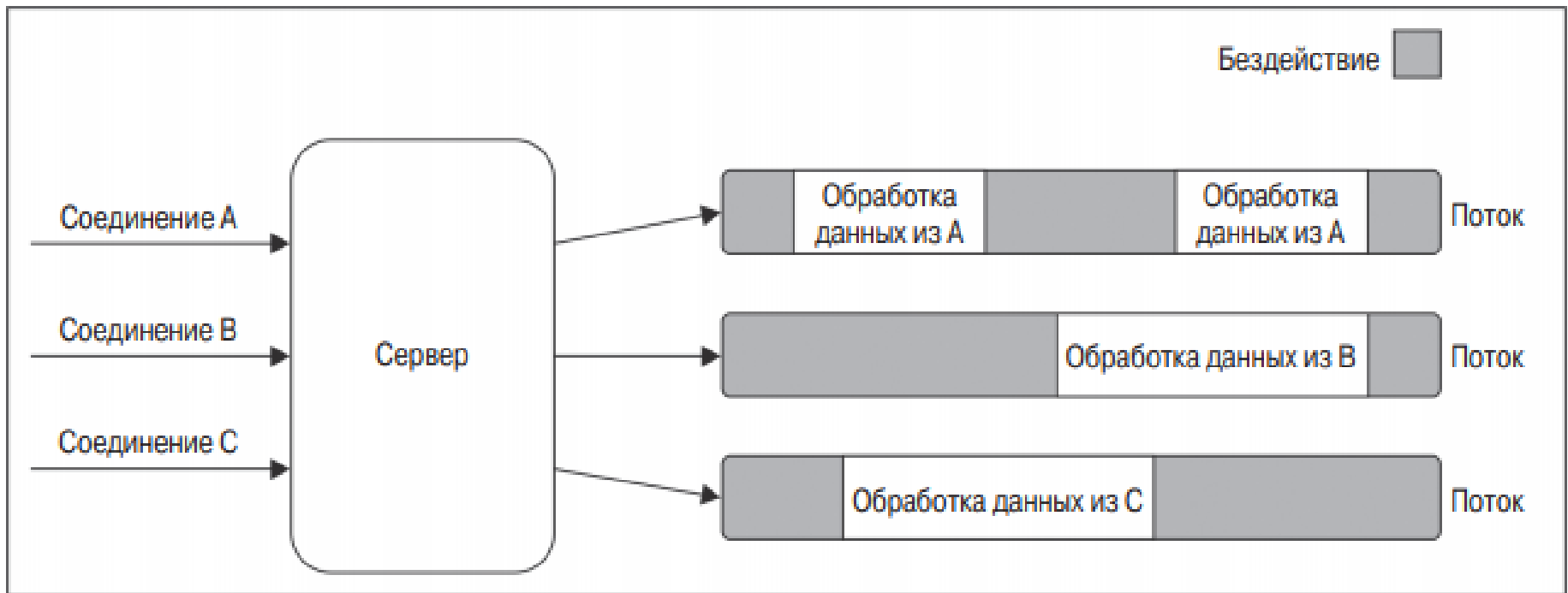
БЛОКИРУЮЩИЙ ВВОД/ВЫВОД

Веб-сервер, реализующий блокирующий ввод/вывод, не в состоянии обрабатывать несколько соединений в одном потоке. Любая операция ввода/вывода через сокет будет блокировать обработку других соединений.

По этой причине для параллельной обработки в веб-серверах создается новый поток или процесс (или повторно используется один из имеющихся в пуле) для каждого из обрабатываемых подключений. Таким образом, блокировка потока операцией ввода / вывода не влияет на обработку других запросов, поскольку все они обрабатываются в отдельных потоках.

К сожалению, потоки потребляют значительные объемы системных ресурсов – расходуют память и вызывают переключение контекста, поэтому иметь достаточно долго выполняющийся поток для каждого подключения и не использовать его большую часть времени является далеко не лучшим решением с точки зрения эффективности.

ПАРАЛЛЕЛЬНАЯ ОБРАБОТКА НЕСКОЛЬКИХ ПОДКЛЮЧЕНИЙ



Для параллельной обработки в веб-серверах создается новый поток или процесс (или повторно используется один из имеющихся в пуле) для каждого из обрабатываемых подключений.

Недостаток: 1) потоки потребляют значительные объемы системных ресурсов – расходуют память и вызывают переключение контекста; 2) каждое подключение не использует свой поток большую часть времени.

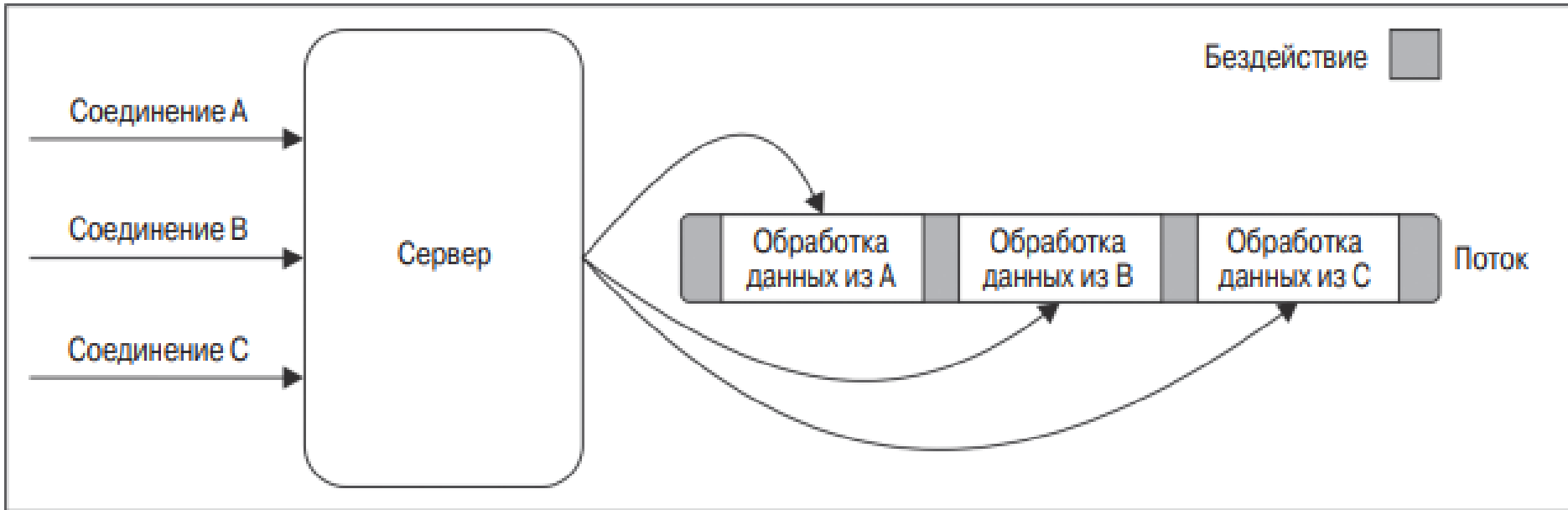
НЕБЛОКИРУЮЩИЙ ВВОД/ВЫВОД

При неблокирующем использовании системные вызовы немедленно возвращают управление, не ожидая выполнения чтения или записи данных. Если на момент вызова отсутствуют какие-либо результаты, функция возвращает предварительно определенную константу, указывающую на невозможность вернуть в этот момент данные.

Основным шаблоном реализации неблокирующего ввода/вывода является активный опрос ресурса в цикле, пока он не сможет вернуть реальные данные. Этот шаблон носит название цикл ожидания (busy-waiting).

С помощью этой простой методики можно обрабатывать несколько ресурсов в одном потоке, но она не слишком эффективна. Так как цикл тратит драгоценное время центрального процессора на обход ресурсов, недоступных большую часть времени. Алгоритмы опроса обычно отличаются огромным количеством времени центрального процессора, потерянного зря.

ШАБЛОН ЦИКЛ ОЖИДАНИЯ



Особенность: Активный опрос ресурса в цикле, пока он не сможет вернуть реальные данные.

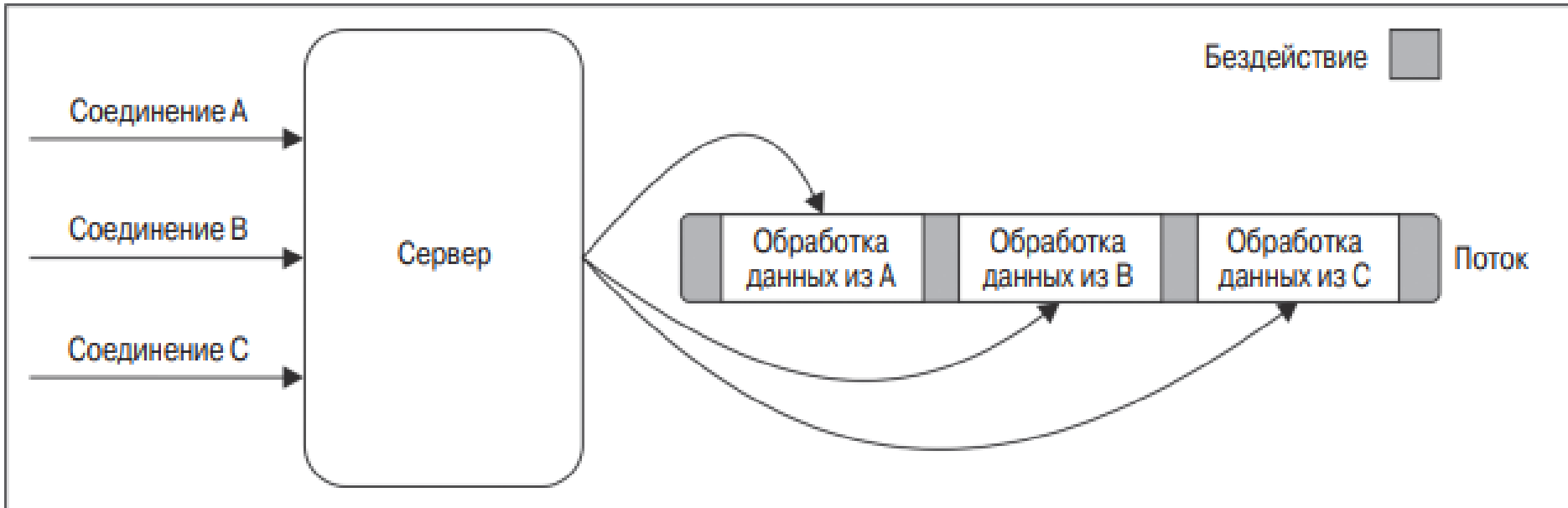
Недостаток: цикл тратит драгоценное время центрального процессора на обход ресурсов, недоступных большую часть времени

НЕБЛОКИРУЮЩИЙ ВВОД/ВЫВОД

Более эффективный механизм параллельной, неблокирующей работы с ресурсами называется синхронным демультимплексированием событий, или интерфейсом уведомления о событиях.

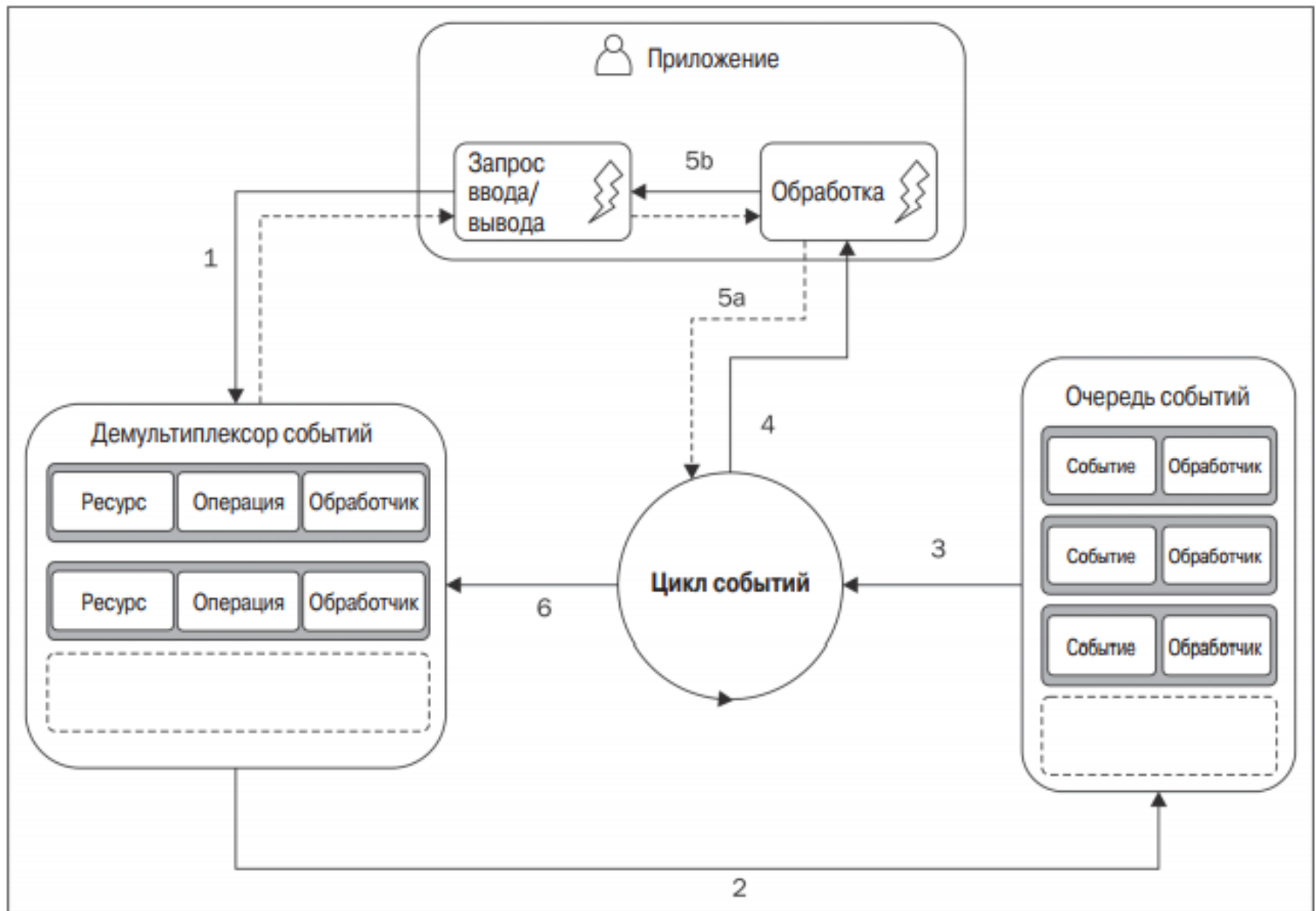
Использование одного-единственного потока не исключает возможности одновременного выполнения нескольких задач, связанных с вводом/выводом. Выполнение задач распределено по времени, а не разделено на несколько потоков. Явное преимущество заключается в сведении к минимуму общего времени простоя потока.

ДЕМУЛЬТИПЛЕКСИРОВАНИЕ СОБЫТИЙ



Особенность: осуществляются сборка и постановка в очередь событий ввода/вывода, поступающих из набора наблюдаемых ресурсов и последующая обработка новых событий в блокирующем режиме.

ШАБЛОН REACTOR



ЛИТЕРАТУРА

Материал взят из книги:

Шаблоны проектирования Node.js / пер. с англ. А.
Н. Киселева. – М.: ДМК Пресс, 2017. – 396 с.