

УДК 004.75

## РЕАЛИЗАЦИЯ МЕТОДОВ ЭФФЕКТИВНОГО ВЗАИМОДЕЙСТВИЯ ПРОЦЕССОВ В РАСПРЕДЕЛЕННЫХ СИСТЕМАХ

Губарев В.Ю. (Университет ИТМО)

Научный руководитель – к.т.н., Косяков М.С.

(Университет ИТМО)

Размещение группы интенсивно взаимодействующих процессов распределенной системы на одном многопроцессорном узле может существенно ускорить выполнение некоторых задач этой системой. Для достижения минимальной временной задержки на передачу данных в пределах одного физического узла необходимо реализовать эффективные методы межпроцессного взаимодействия. В работе разработаны и реализованы методы межпроцессного взаимодействия на основе разделяемой памяти, примитива синхронизации `futex` и TCP-соединения.

**Введение.** Для некоторых систем эффективное межпроцессное взаимодействие является критически важной частью их работы. Требование к минимизации времени обработки запросов может напрямую следовать из области применения системы, как в случае с финансовыми системами для алгоритмической торговли на финансовых рынках. Обработка запросов в высоконагруженной системе множеством логически связанных процессов может быть существенно ускорена путем динамического размещения группы таких процессов на одном физическом узле и при применении наиболее эффективных методов межпроцессного взаимодействия.

**Основная часть.** В рамках настоящей работы реализованы методы межпроцессного взаимодействия через разделяемую память с оповещением о наличии данных через мультиплексор событий на основе примитива синхронизации `futex` и через разделяемую память с оповещением о наличии данных посредством TCP-соединения и системного мультиплексора событий `select`.

Экземпляр мультиплексора событий на основе примитива синхронизации `futex` для каждого процесса располагается в файле, отображаемом в памяти взаимодействующих процессов. Во время установления соединения процессы сообщают противоположной стороне о расположении мультиплексоров событий для отправки сигналов, номера сигналов и предоставляют файлы для передачи самих данных для отображения в памяти обоими процессами.

Обмен данными по каналу связи, использующему мультиплексор событий на основе примитива синхронизации `futex`, состоит из следующих шагов:

1. Процесс-получатель ожидает получения сигнала на `futex` (поток-получатель находится в состоянии сна).
2. Процесс-отправитель записывает данные в сегмент разделяемой памяти принимающей стороны.
3. Процесс-отправитель атомарными операциями “ИЛИ” выставляет необходимый сигнальный бит и соответствующий ему бит `futex` в мультиплексоре событий принимающей стороны и совершает системный вызов для пробуждения процесса-получателя.
4. Процесс-получатель пробуждается, получает значения `futex` и сигнальных битов операциями атомарного обмена и диспетчеризует обработку входящих соединений.

В случае интенсивного взаимодействия, например, при работе с несколькими соединениями, число системных вызовов, требуемых на передачу единицы данных, уменьшается из-за отсутствия необходимости пробуждать процесс-получатель (шаг один и

шаг три) и вся передача данных происходит исключительно через пользовательскую память, что приводит к уменьшению временной задержки на передачу данных.

**Выводы.** Реализованные в настоящей работе методы межпроцессного взаимодействия были испытаны в продукте Tbricks от компании Itiviti. Метод, основанный на разделяемой памяти и примитиве синхронизации `futex`, на практически значимой нагрузке показал как минимум вдвое меньшую временную задержку на передачу данных в 95% случаев, в сравнении с методом, использующим TCP-соединение для оповещения о наличии данных в разделяемой памяти. Основными факторами более эффективной передачи данных является использование памяти в пространстве пользователя и минимально необходимое использование ядра операционной системы. Однако оба реализованных метода не являются универсальными и требуют расширенных знаний об их устройстве, что может составить трудности при интеграции со сторонними системами.

В дальнейших исследованиях для совершенствования полученного метода межпроцессного взаимодействия, основанного на разделяемой памяти и примитиве синхронизации `futex`, планируется применение шаблона проектирования сетевых приложений `Leader/Followers` для уменьшения временной задержки на передачу данных за счет уменьшения временной задержки на диспетчеризацию обработки входящих соединений.

Губарев В.Ю. (автор)

Подпись

Косяков М.С. (научный руководитель)

Подпись