УДК 004.432.2

Средства повышения производительности при разработке TCP сервера

Белова С.В., Корогода А.И.
Белорусский национальный технический университет

Транспортный протокол TCP ориентирован на логические соединения и предоставляет надежный коммуникационный путь между двумя конечными точками. Важное преимущество TCP в том, что он гарантирует доставку сообщений и правильный порядок пакетов.

Один из возможных подходов при разработке TCP сервера – реализация многопоточного сервера путем создания отдельных потоков для приема клиентских соединений и управления ими. Многопоточный сервер является довольно быстрым, но такая модель плохо масштабируется. Главный недостаток – большое число создаваемых и уничтожаемых потоков, которые требуют большого объема ресурсов.

Избежать возникающих проблем можно, используя ввод-вывод в одном потоке путем применения метода Select. Этот подход позволяет подключить одновременно гораздо больше клиентов. Однако сервер значительно медленнее при этом реагирует на запросы, чем предыдущий вариант. Сделать сервер более масштабируемым можно, используя асинхронную модель ввода-вывода. Асинхронная модель устраняет необходимость в создании потоков и управлении ими, что позволяет значительно упростить код и повышает эффективность ввода-вывода.

При асинхронном программировании для обработки входящих данных используются методы обратного вызова. С каждым .NET приложением сопоставлен пул потоков. Когда у функции асинхронного ввода-вывода имеются готовые к обработке данные, поток из пула потоков .NET выполняет функцию обратного вызова. После завершения обратного вызова поток возвращается в пул. Это отличается от подхода, при котором поток из пула использовался для обработки конкретного запроса; сейчас речь идет о том, что поток из пула служит для выполнения только одной операции ввода-вывода.

Таким образом, при использовании асинхронной модели для разработки TCP сервера, с одной стороны, повышается его производительность, с другой стороны, он может быть более масштабируемым.

Чтобы уменьшить нагрузку на сеть и увеличить производительность приложения необходимо перенести по возможности обработку данных на сервер, минимизировать количество обращений к серверу для доступа к данным и ограничить объем данных, загружаемых с сервера.