**Изящное Завершение работы**

Объясняет, как корректно завершить работу сервера gRPC, чтобы не вызывать сбои RPC у подключённых клиентов.

Обзор

Серверам gRPC часто требуется корректно завершать работу, чтобы выполняемые RPC-запросы завершались в разумные сроки, а новые RPC-запросы больше не принимались. Функция «корректного завершения работы» упрощает этот процесс, позволяя серверу плавно переходить в другой режим без резкого разрыва активных соединений.

При вызове функции «Изящного завершения работы» сервер немедленно уведомляет всех клиентов о необходимости прекратить отправку новых RPC. Затем, после того как клиенты получат это уведомление, сервер прекращает приём новых RPC. Текущие RPC могут выполняться до их завершения или до истечения указанного срока. Как только все активные RPC завершатся или истечёт срок, сервер полностью выключается.

Поскольку корректное завершение работы помогает предотвратить сбои RPC-запросов у клиентов, его следует использовать по возможности. Однако gRPC также предоставляет механизм принудительного завершения работы, который немедленно приводит к остановке сервера и закрытию всех подключений, что приводит к сбою всех выполняемых RPC-запросов.

Как выполнить Изящное Завершение работы сервера

Точная реализация «функции корректного завершения работы» зависит от используемого вами языка программирования. Однако общий принцип таков:

* Чтобы инициировать процесс корректного завершения работы, вызовите функцию «Корректное завершение работы» на объекте сервера gRPC. Эта функция блокирует работу до завершения всех запущенных RPC-запросов. Это гарантирует, что выполняемые запросы будут обработаны.
* Укажите период ожидания, чтобы ограничить время, необходимое для завершения выполняемых RPC-запросов. Очень важно отдельно вызвать функцию «Принудительное завершение работы» на объекте сервера с помощью механизма таймера (в зависимости от вашего языка), чтобы инициировать принудительное завершение работы по истечении заданного времени. Это служит дополнительной защитой и гарантирует, что сервер в конечном итоге завершит работу, даже если некоторые выполняемые RPC-запросы не завершатся в разумные сроки. Это предотвращает бесконечную блокировку.

Ниже показана последовательность событий, которые происходят во время корректного завершения работы. При корректном завершении работы сервера текущие RPC-запросы продолжают обрабатываться, но новые RPC-запросы отклоняются. Если некоторые текущие RPC-запросы не завершаются вовремя, сервер принудительно завершает работу.

ClientServerNew RPC Request 1New RPC Request 2Graceful Shutdown InvokedContinues Processing In-Flight RPCsDetects server shutdown and finds other servers if availableCompletes RPC 1Completes RPC 2Graceful Shutdown CompleteForceful Shutdown Invoked, terminating pending RPCsForceful Shutdown Completealt[RPCs complete within timeout][Timeout reached]ClientServer

Ниже приведено представление, основанное на состоянии

Server StartedGraceful Shutdown Called (with Timeout)In-Flight RPCs Completed (Before Timeout)Timeout ReachedForceful Shutdown CalledSERVINGGRACEFUL\_SHUTDOWNTERMINATEDTIMER\_EXPIRED

Языковая Поддержка

ЯзыкПримерC++ВпередПерейти к ПримеруJavaПример JavaПитон

Последнее изменение 15 января 2025 г.: руководства/функции: добавить документ о том, как корректно завершить работу сервера (#1388) (cbe3d9a)