

РАЗРАБОТКА СИСТЕМЫ ОБНАРУЖЕНИЯ СЕРВИСОВ С ИСПОЛЬЗОВАНИЕМ ТЕХНОЛОГИИ ВЫЗОВА УДАЛЕННЫХ ПРОЦЕДУР И И СЕРВЕРА УПРАВЛЕНИЯ КОНФИГУРАЦИЕЙ

Поляков И.Р.

Университет ИТМО, Санкт-Петербург

Научный руководитель – Хвастунов А.П.

Университет ИТМО, Санкт-Петербург

Введение. Современные информационные системы состоят из множества компонентов, взаимодействующих в компьютерных сетях, а это подразумевает высокую вероятность влияния человеческого фактора на ошибки при конфигурировании. В результате работы разработано программное обеспечение, реализующее механизм обнаружение сервисов в компьютерной сети с помощью RPC и сервиса управления конфигурацией Apache Zookeeper.

Основная часть. Задача обнаружения сервисов затрагивает любое программное обеспечение, предназначенное для обращения по сети, особенно стремящегося к высокой доступности. Например, при работе на нескольких физических серверах в избыточной конфигурации. Многие технологические компании создали собственные, предназначенные для внутреннего использования системы, многие из которых были выпущены в качестве программного обеспечения с открытым исходным кодом. На высоком уровне существует несколько различных способов решения этой задачи.

1. Разрешить клиентам обращаться к любому узлу (например, с помощью циклического балансировщика нагрузки). Если в этом узле случайно окажется секция, которая нужна для ответа на запрос, то он обработает запрос непосредственно, в противном случае перенаправит его соответствующему узлу, получит от него ответ и переправит его клиенту.
2. Отправлять все поступающие от клиентов запросы сначала маршрутизирующему звену, которое определяет, какой узел должен обрабатывать данный запрос, и перенаправляет его соответствующим образом. Это маршрутизирующее звено само не обрабатывает никаких запросов, а служит только учитывающим секции балансировщиком нагрузки.
3. Потребовать, чтобы клиенты учитывали секционирование и распределение секций по узлам. В таком случае клиент может подключаться непосредственно к соответствующему узлу, без всяких посредников.

Во всех случаях основная проблема формулируется следующим образом: откуда отвечающий за маршрутизацию компонент (это может быть один из узлов, маршрутизирующее звено или клиент) знает об изменениях в распределении секций по узлам?

Это непростая задача, ведь нужен консенсус всех участвующих сторон — в противном случае запросы будут отправляться только не тем узлам и обрабатываться неправильно. Существуют протоколы достижения консенсуса в распределенной системе, но реализовать их правильно непросто, эта проблема является предметом отдельного исследования.

Примером существующих реализаций может служить база данных Espresso социальной сети LinkedIn, использующая для управления кластером Helix. Базы данных HBase, SolrCloud и Kafka также применяют Zookeeper для отслеживания распределения секций. Архитектура MongoDB аналогична, но основана на собственной реализации сервера конфигураций и демонов mongos в качестве маршрутизирующего звена.

В разработке, являющейся результатом данной работы, для отслеживания метаданных секций кластера используется отдельный сервис координации — Zookeeper, так как большинство собранной статистики говорило в его пользу. Все узлы регистрируются в сервисе, который поддерживает актуальную карту соответствий секций узлам. Другие компоненты, например, маршрутизирующее звено или учитывающий секции клиент, могут

подписываться на данную информацию в Zookeeper. При каждой смене узла-владельца секции сервис оповещает маршрутизирующее звено, так что маршрутизация остается актуальной.

Сообщение между компонентами осуществляется с помощью удаленного вызова процедур с использованием ProtocolBuffers от Google. Основное преимущество такого подхода — возможность адаптировать любой бинарный протокол для общения между сервисами. После старта главный сервис публикует логическую схему RPC-серверов в Zookeeper, что позволяет однозначно идентифицировать адрес для запроса необходимой клиенту секции.

Выводы. В результате работы были рассмотрены обобщенные подходы к проектированию механизмов обнаружения сервисов, рассмотрены существующие аналоги, а также разработано программное обеспечение, реализующее рассматриваемый механизм с использованием сервера координации Zookeeper и RPC.

Поляков И.Р. (автор)

Подпись

Хвастунов А.П. (научный руководитель)

Подпись

Email: ipolyakov@gmail.com

Телефон: +79214316283