

РАСПРЕДЕЛЕННАЯ ТРАССИРОВКА

Кива В. С., Брановицкий А. А.

Кафедра программного обеспечения информационных технологий, Белорусский государственный университет информатики и радиоэлектроники

Минск, Республика Беларусь

E-mail: knjiaz@gmail.com, art7yom@gmail.com

В статье описан один из способов решения проблемы диагностики производительности программного обеспечения в тестовой или рабочей среде, проблема локализации сбоев или необходимость построения графа зависимостей распределенной системы. И одним из способов решить эти проблемы – внедрение распределенной трассировки в существующую систему.

ВВЕДЕНИЕ

Распределенная трассировка, также называемая распределенной трассировкой запросов, представляет собой метод, используемый для профилирования и мониторинга приложений, особенно тех, которые построены с использованием архитектуры микросервисов. Распределенная трассировка помогает точно определить, где происходят сбои и что приводит к снижению производительности программного обеспечения.

I. СЛОЖНОСТИ ВНЕДРЕНИЯ ТРАССИРОВКИ

Внедрение распределенной трассировки является сложной задачей, потому что инструментарий должен распространять контекст трассировки как внутри, так и между процессами. Выполнение внедрения затрагивает почти каждую часть приложения. В частности, контекст трассировки должен быть передан через:

- Автономные сервисы программного обеспечения с открытым исходным кодом (NGINX, Cassandra, Redis и т. д.);
- Пакеты программного обеспечения с открытым кодом, связанные с пользовательскими службами (gRPC, ORM и т. д.).

Самое сложное в распределенной трассировке – это распределенная часть. Любая система трассировки нуждается в способе понимания причинно-следственной связи между действиями во многих отдельных процессах, независимо от того, связаны ли они через формальные структуры RPC, системы подписчик-издатель, общие очереди сообщений, прямые HTTP-вызовы, UDP-пакеты или иным способом.

Некоторые распределенные системы отслеживания (например, Project5 от 2003 г., WAP5 от 2006 г. или The Mystery Machine из 2014 г.) выводят трассировку за границы процесса (подход черного ящика). Что касается качества, то OpenTracing является явным стандартом инструментов распределенной трассировки, и он намного лучше соответствует подходам, таким как X-Trace от 2007 года, Dapper от 2010 года, или многочисленным системам трассировки с открытым исходным кодом, таким как Zipkin или Jaeger (среди прочих).[1]

Нецелесообразно просить все службы программного обеспечения с открытым исходным кодом, внедрить одного поставщика трассировки в их пакеты и в весь код приложения. Тем не менее, если у них нет общего механизма для описания и распространения спанов, трассировка нарушается, а спаны не являются полными. Поэтому нужен единый стандартный механизм для описания поведения систем.

II. ФОРМАТ ТРАССИРОВКИ

Для упрощенного интегрирования между библиотеками программного обеспечения по трассировке запросов введен один из общих форматов – OpenTracing. OpenTracing состоит из спецификации программных интерфейсов, интерфейсов и библиотек, которые реализовали спецификацию и документацию проекта. OpenTracing позволяет разработчикам добавлять трассировочный инструментарий в код своего приложения с помощью программных интерфейсов, которые не привязывают их к какому-либо конкретному продукту или поставщику.

OpenTracing стандартизирует:

- Управление «спанами»: программный интерфейс приложения для запуска, завершения и декорирования временных операций;
- Межпроцессное распространение: программный интерфейс приложения для передачи контекста трассировки за границы процесса;
- Управление активным спаном: программный интерфейс приложения для хранения и извлечения активного спана в одном процессе;
- Внутрислое контекстное кодирование: спецификация точного формата кодирования для отслеживания контекста, передаваемого вместе с данными приложения между процессами;
- Кодирование данных внеполосной трассировки: спецификация того, как кодируются декорированные данные трассировки и спана, когда они направляются к поставщику распределенной трассировки.

Все специфичные для языков программирования программные интерфейсы OpenTracing

имеют некоторые основные понятия и терминологию. Эти концепции настолько важны, что у них есть свой собственный репозиторий на github.[1]

III. СПАН

Одним из фундаментальных понятий OpenTracing является «Спан». «Спан» – это основной строительный блок распределенной трассировки, представляющий отдельную единицу работы, выполненную в распределенной системе. Каждый компонент распределенной системы вносит свой спан – именованную временную операцию, представляющую часть рабочего процесса.

Спаны могут (и обычно содержат) ссылки на другие спаны, что позволяет объединять несколько спанов в одну полную трассировку и визуализировать сроки действия запроса при его перемещении по распределенной системе.

В настоящее время OpenTracing определяет два типа ссылок спанов: ChildOf и FollowsFrom. Оба ссылочных типа специально моделируют прямые причинно-следственные связи между дочерним спаном и родительским спаном. В будущем OpenTracing может также поддерживать ссылочные типы для спанов с не причинно-следственными связями (например, спаны, которые объединены вместе, спаны, которые ждут в одной очереди и т. д.).[2]

Ссылки ChildOf: спан может быть ребёнком родительского спана. В ссылке ChildOf родительский спан в некоторой степени зависит от дочернего спана. Примеры ChildOf:

- Спан, представляющий серверную часть RPC, может быть ChildOf спана, представляющим клиентскую сторону этого RPC;
- Спан, представляющий вставку SQL, может быть ChildOf спана, представляющим метод сохранения ORM;
- Множество спанов, выполняющие параллельную (возможно, распределенную) работу, могут все по отдельности быть ChildOf одного родительского спана, который объединяет результаты для всех дочерних элементов, которые возвращаются в установленные сроки.

Каждый спан инкапсулирует следующее состояние в соответствии со спецификацией OpenTracing:

- Название операции;
- Метка времени начала и окончания;
- Наборы ключ-значение, содержащие теги;
- Наборы ключ-значение, содержащие логи.

IV. НАБОРЫ КЛЮЧ-ЗНАЧЕНИЕ

Теги – пары ключ-значение, которые позволяют задавать пользователем аннотации диапазонов для запроса, фильтрации и уточнение данных трассировки.

Логи являются парами ключ-значение, которые полезны для сохранения сообщений журнала, относящихся к спан, и других отладочных или информационных выходных данных самого приложения. Логи могут быть полезны для документирования определенного момента или события в промежутке (в отличие от тегов, которые должны применяться к промежутку в целом).

V. ПОТОКИ

В любом конкретном потоке есть «активный» спан, в основном отвечающий за работу, выполняемую окружающим кодом приложения, который называется активным спаном. Программный интерфейс приложения OpenTracing позволяет активировать только один спан в потоке в любой момент времени. Это управляется с помощью области видимости, которая управляет описывает активацию и деактивацию спана.[3]

Другие спаны, которые связаны с тем же потоком, будут удовлетворять одному из следующих условий:

- Начался;
- Не закончен;
- Не активный.

Например, в одном потоке может быть несколько спанов, если они:

- В ожидании ввода / вывода;
- Заблокированы на дочернем спане;
- Вне критического пути.

Если область видимости существует, когда разработчик создает новый спан, он будет выступать в качестве его родительского, если программист не вызовет специальный метод по игнорированию активного спана во время его создания или не укажет родительский контекст явно.

VI. ПОТРЕБИТЕЛИ

Команды информационно-коммуникационных технологий и администраторов используют OpenTracing для мониторинга приложений. Распределенная трассировка особенно хорошо подходит для отладки и мониторинга современных архитектур распределенного программного обеспечения, таких как микросервисы. Разработчики могут использовать распределенную трассировку для отладки и оптимизации своего кода.

1. Vendor-neutral APIs and instrumentation for distributed tracing [Электронный ресурс] / Documentation Home – Режим доступа: <https://opentracing.io/docs>. – Дата доступа: 29.09.2019.
2. GitHub is where people build software [Электронный ресурс] / Documentation Home – Режим доступа: <https://github.com/opentracing/specification>. – Дата доступа: 28.09.2019.
3. Medium, a place where words matter [Электронный ресурс] / Distributed Tracing in 10 Minutes – Режим доступа: <https://medium.com/opentracing/distributed-tracing-in-10-minutes-51b378ee40f1>. – Дата доступа: 28.09.2019.