

Функциональное описание Q.TraceR

Exported on Jul 08, 2019

Q.Tracer - это проект, обеспечивающий механизмы для отправки, получения, хранения и визуализации трассировки. Инструмент позволяет подробно исследовать активность между серверами, сервисами и получать наиболее полную картину о том, что действительно происходит внутри компонентов системы.

Q.Tracer предоставляет возможность исследовать последовательность вызовов в иерархическом виде. Наглядно становится понятно, на каких этапах обработки запроса или отправки ответа происходит задержка, т.к информация о времени отображается на всех шагах трассировки.

Внедрение механизма трассировки доступно для многих технологий и платформ: C, C++, C#, Go, Java, JavaScript, Ruby, Scala, PHP, Python.

Обзор архитектуры механизма трассировки

Элементы трассировки, собирающие всю информацию, внедряются в приложения и сохраняют данные о времени и промежуточных данных, необходимых для анализа. Например, веб сервер, в который внедрен трассировщик, записывает данные о том, когда он получает запрос и когда он посылает ответ. Собранная информация называется спаном (Span).

Внедрение трассировщика безопасно для боевой конфигурации программного обеспечения и оборудования. Трассировщик не потребляет ощутимых ресурсов и не создает дополнительную нагрузку на оборудование. Иерархия вызовов выстраивается путем передачи ID номера инициатора получателю, чтобы сообщить последнему, что трассировка находится сейчас в процессе. Готовые спаны (Spans) передаются в Q.TraceR асинхронно, без ощутимых затрат вычислительных ресурсов. Например, когда процессу, который обрабатывается трассировщиком, нужно сделать исходящий http запрос, несколько заголовков (Headers) добавляются к нему, чтобы предоставить ID номера. Заголовки не используются для отправки детальной информации, такой как имя процесса и прочего.

Компонент в приложении с внедренным трассировщиком, который передает данные в Q.TraceR, называется репортер (Reporter). Репортеры отправляют трассировочную информацию с помощью одного из нескольких каналов доставки Q. Tracer, которые, в итоге, передают данные в хранилище. Затем через программный интерфейс приложения происходит взаимодействие с хранилищем, чтобы предоставить визуальную информацию о трассировке в веб приложении, входящем в комплект Q.TraceR.

В качестве хранилища прекрасно подходит входящий в комплект поставки Q.KeepR.