

Министерство образования Республики Беларусь
Учреждение образования
Белорусский государственный университет
информатики и радиоэлектроники

УДК 004.42:004.72

Кива
Владислав Сергеевич

Сквозная распределенная трассировка программного обеспечения с
микросервисной архитектурой

АВТОРЕФЕРАТ

на соискание академической степени
магистра технических наук

по специальности 1-40 80 05 – Математическое и программное обеспечение
вычислительных машин, комплексов и компьютерных сетей

Научный руководитель
Скобцов Вадим Юрьевич
к. т. н., доцент

Минск 2020

ВВЕДЕНИЕ

До 2005 года, когда Доктор Питер Роджерс ввел термин «Микро-Веб-Сервис», на рынке информационных услуг преобладали приложения с монолитной архитектурой и было гораздо проще справляться с такими проблемами как отладка, логирование и отслеживание взаимодействия программных модулей. В настоящее время всё больше компаний переходят на микросервисную архитектуру, в которой остро стоят все вышеперечисленные проблемы. Исследование Dimensional Research® «Глобальные тенденции в сфере микросервисов», проведенное в 2018 году, показало, что более 91% опрошенных специалистов используют или планируют использовать микросервисы в своих системах. В то же время 56% говорят, что каждый дополнительный микросервис «увеличивает эксплуатационные проблемы», а 73% считают, что «устранение неполадок труднее» в среде микросервисов.

Монолитное программное обеспечение построено на большой и обширной унаследованной кодовой базе, которая часто является настолько сильно связанной, что любые изменения в одном небольшом разделе часто приводят к нарушению одной или нескольких функций, которые зависят от него. В результате приложения, которые используют это обеспечение, являются хрупкими и подвержены взлому. Если это происходит, то у разработчиков есть несколько проверенных и отработанных методов для устранения неполадок.

Одним из ключевых методов поиска проблемы является отслеживание или трассировка. Трассировка следует за ходом запроса или системного события от его источника до конечного пункта назначения. Каждый след(трейс) — это информация, в которой находится история запроса, когда он проходит через систему. Например, многие приложения генерируют трассировку стека после ошибки времени выполнения. Трассировка сохраняется в журнале приложений, отображается в окне консоли / терминала, а затем анализируется и проверяется с помощью инструментов разработки, таких как Microsoft Visual Studio, Apple Xcode и т.д.

Несмотря на многочисленные недостатки, монолитные приложения имеют ряд преимуществ перед микросервисами и другими типами распределенных приложений. Первые обычно находятся в одном месте, работают на определенных устройствах и имеют определенные отношения с внешними системами. В результате, когда что-то идет не так, можно быстро найти источник запроса и относительно легко отследить его путь по системе. Однако в мире облачных микросервисов все становится на порядок сложнее из-за того, что каждый новый микросервис добавляет сложности к уже существующей системе.

ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

Цель и задачи исследования

Целью диссертационной работы является исследование моделей и методов мониторинга программного обеспечения и проектирования библиотеки объединяющую инструменты мониторинга.

Для достижения поставленной цели необходимо решить следующие задачи:

- изучить научно-методическую и справочную литературу по вопросу мониторинга программного обеспечения;
- проанализировать существующие инструменты мониторинга;
- разработать асинхронные и синхронные инструменты для сбора метрик;
- решить задачу сбора и передачи необработанных измерений или метрик с предопределенной агрегацией и набором меток;
- решить задачу построения графа зависимостей из данных по трассировкам;
- разработать библиотеку, объединяющую возможности захвата трассировки и сбора телеметрии.

Объектом исследования является программное обеспечение, в частности, с микросервисной архитектурой.

Предметами исследования являются методы и модели телеметрии и трассировки программного обеспечения.

Связь работы с приоритетными направлениями научных исследований и запросами реального сектора экономики

Работа выполнялась в соответствии с научно-техническим заданием и планом работ кафедры «Программное обеспечение информационных технологий» по теме «Разработка моделей, методов, алгоритмов, повышающих показатели проектирования, внедрения и эксплуатации программных средств для перспективных платформ обработки информации, решения интеллектуальных задач, работы с большими массивами данных и внедрение в современные обучающие комплексы» (ГБ № 16-2004, № ГР 20163588, научный руководитель НИР – Н. В. Лапицкая).

Личный вклад соискателя

Результаты, приведенные в диссертации, получены соискателем лично. Вклад научного руководителя В.Ю. Скобцова, заключается в формулировке целей и задач исследования.

Апробация результатов диссертации

Основные положения диссертационной работы докладывались и обсуждались на международной научной конференции «Информационные технологии и системы» (Минск, Беларусь, 30 октября 2019).

Опубликованность результатов диссертации

По теме диссертации опубликовано 2 печатных работы, из них 2 работы в сборниках трудов и материалов международной конференции.

Структура и объем диссертации

Диссертация состоит из общей характеристики работы, введения, четырёх глав, заключения, списка использованных источников, списка публикаций автора и приложения. В первой главе представлено обоснование выбора темы магистерской диссертации. Во второй главе произведён анализ предметной области, рассматриваются традиционные инструменты мониторинга. Третья глава рассказывает о построении модели данных метрик и основных инструментах для их подсчёта. В четвёртой главе производится описание программной реализации системы, описание архитектуры разработанной библиотеки, приводятся результаты.

КРАТКОЕ СОДЕРЖАНИЕ РАБОТЫ

Данная диссертационная работа посвящена трассировке и метрикам.

В первой главе обосновывается выбор темы. Суть которой заключается в том, что программное обеспечение может работать на любом реальном устройстве, включая носимые мобильные устройства, настольные компьютеры и серверы. Благодаря технологиям виртуализации, таким как гипервизоры и контейнеры, возможно размещение нескольких виртуальных машин и программного обеспечения на одном хосте. Между тем, безсерверные вычисления создали новый уровень абстракции. Современные приложения используют несколько веб-сервисов и API-интерфейсов, некоторые из которых могут быть продуктами организации, а многие могут быть приобретены у сторонних поставщиков. Поскольку цена облачных вычислений продолжает падать и качество обслуживания улучшается, многие компании переносят свое существующее локальное программное обеспечение в облака. Для упрощения поддержки и работы микросервисной архитектурой был разработан метод распределённой трассировки.

Во второй главе приводятся анализ предметной области. В ней рассказывается о том, что традиционные инструменты мониторинга были разработаны для монолитных систем, для отслеживания состояния и поведения одного экземпляра приложения. Они могут показать нам хронометрию событий этого единственного экземпляра, но почти ничего не знают о распределенной транзакции, которая прошла через него. Этим инструментам не хватает контекста запроса. Метрики или статистика — это числовые показатели, записанные приложением, такие как счетчики, датчики или таймеры. Собирать метрики очень дешево, поскольку числовые значения могут быть легко агрегированы, чтобы уменьшить накладные расходы на передачу этих данных в систему мониторинга. Они также достаточно точны и поэтому очень полезны для реального мониторинга и оповещения. Тем не менее, та же способность к агрегации делает метрики неподходящими для объяснения патологического поведения приложения. Объединяя данные, мы отбрасываем весь контекст, который был у нас в отдельных транзакциях.

В третьей главе рассматривается построение модели данных и определение инструментов для сбора метрик. Основной упор сделан на инструменты сбора метрик. Аддитивные инструменты используются для сбора информации о сумме, где, по определению, только сумма представляет интерес. Отдельные события считаются не значимыми для этих инструментов, количество событий не вычисляется. Это означает, например, что два события счетчика Add (N) и Add (M) эквивалентны одному событию счетчика Add (N + M). Это так, потому что Счетчик является синхронным, а синхронные аддитивные инструменты используются для регистрации изменений в сумме. В неаддитивных инструментах используется относительно недорогой метод агрегирования по умолчанию (MinMaxSumCount), но все же более дорогой, чем по умолчанию для аддитивных

инструментов (Sum). В отличие от аддитивных инструментов, где по определению представляет интерес только сумма, неаддитивные инструменты могут быть сконфигурированы с еще более дорогими агрегаторами.

В четвёртой главе рассказывается о программной реализации системы, первые главы рассматривают архитектуру библиотеки. Библиотека состоит из 2 пакетов: пакета API и пакета SDK. Пакет API является самодостаточной зависимостью в том смысле, что если приложение конечного пользователя или сторонняя библиотека зависят только от него и не подключают полную реализацию SDK, то приложение все равно будет собираться и запускаться без сбоев, хотя никакие данные телеметрии фактически не будут доставлены в бэкэнд телеметрии. Эта самодостаточность достигается следующим образом. Зависимость API содержит минимальную реализацию API. SDK реализует основные функции, необходимые для преобразования вызовов API в данные телеметрии, которые готовы к экспорту.

ЗАКЛЮЧЕНИЕ

Основные научные результаты диссертации

1. Произведён анализ инструментов мониторинга за программным обеспечением. Изучена научно-методическую и справочную литературу по вопросу мониторинга программного обеспечения;
2. Проанализированы существующие инструменты мониторинга.
3. Разработаны асинхронные и синхронные инструменты для сбора метрик.
4. Решена задача сбора и передачи необработанных измерений или метрик с предопределенной агрегацией и набором меток.
5. Решена задачу построения графа зависимостей из данных по трассировкам.
6. Разработана библиотека, объединяющую возможности захвата трассировки и сбора телеметрии.

Рекомендации по практическому использованию результатов

1. Результаты в виде разработанного ПС могут быть использованы для решения задач мониторинга.
2. Результаты в виде библиотеки могут быть использованы приложениями, использующие сторонние библиотеки, оснащенные API-интерфейсом библиотеки, будут иметь возможность включить или отключить фактическую доставку данных телеметрии. Приложение также может напрямую вызывать API телеметрии для получения дополнительных данных телеметрии. Чтобы включить телеметрию, приложение должно зависеть от библиотеки SDK, который реализует доставку телеметрии. Приложение также должно настроить экспортеров, чтобы SDK знал, куда и как доставить телеметрию. Сведения о том, как экспортеры включены и настроены, зависят от языка реализации.

СПИСОК ОПУБЛИКОВАННЫХ РАБОТ

1-А. Кива, В.С. Распределенная трассировка / Кива В. С., Брановицкий А. А. // Системы обработки информации: материалы международной научной конференции «Информационные технологии и системы» БГУИР, Минск, 30 окт. 2019 – Минск, 2019. – с. 48–49.

2-А. Кива, В.С. Шаблоны распределения ответственности по запросам команд и источник событий / Кива В. С., Борисов Д. В. // Системы обработки информации: материалы международной научной конференции «Информационные технологии и системы» БГУИР, Минск, 30 окт. 2019 – Минск, 2019. – с. 224–225.

Библиотека БГУИР