УДК 004.75 DOI: 10.31799/2077-5687-2022-1-38-43

МЕТОДЫ СОГЛАСОВАНИЯ ИЗМЕНЕНИЙ В РАСПРЕДЕЛЁННЫХ СИСТЕМАХ

А. А. Чернова, К. А. Курицын
Санкт-Петербургский государственный университет аэрокосмического приборостроения

Рассмотрены способы управления распределенных транзакциями, охватывающими множество подсистем и, в частности, проблема долгоживущих транзакций, которые удерживают ресурсы, откладывая завершение транзакций в локальных базах данных. В качестве альтернативы распределенным долгоживущим транзакциям исследован архитектурный паттерн SAGA, который представляет распределенную транзакцию последовательностью отдельных шагов, состоящих из локальных транзакций, а также набора компенсирующих действий. Проанализированы способы реализации SAGA через оркестрацию и хореографию и произведено сравнение с долгоживущими транзакциями.

Ключевые слова: распределённые системы, двухфазные транзакции, SAGA, долгоживущие транзакции, хореография, оркестрация.

***Для цитирования**:*
Чернова А. А., Курицын К. А. Методы согласования изменений в распределённых системах // Системный анализ и логистика: журнал.: выпуск №1(31), ISSN 2007-5687. – СПб.: ГУАП., 2022 – с.38-43. РИНЦ. DOI: DOI: 10.31799/2077-5687-2022-1-38-43.

RECONCILIATION METHODS OF CHANGES IN DISTRIBUTED SYSTEMS

A. A. Chernova, A. K. Kuritsyn
St. Petersburg State University of Aerospace Instrumentation

This article considers methods of distributed transaction management. The problem of long-lived distributed transactions locking resources of local databases has been marked. The SAGA approach has been suggested as an alternative way to manage changes in distributed systems. According to the design pattern one distributed transaction can be considered as a sequence of logical local steps to perform a global desired action and a sequence of local steps to compensate for the changes made by a failed step of a direct flow. We investigate implementation schemes of SAGA with choreography and orchestration and compare the methods.

Keywords: distributed systems, two-phase commit, 2PC, SAGA, long-lived transactions, choreography, orchestration.

For citation:
Chernova A. A., Kuritsyn A. K. Reconciliation methods of changes in distributed systems // System analysis and logistics.: №1(31), ISSN 2007–5687. – Russia, Saint-Petersburg.: SUAI., 2022 –p 38-43. DOI: 10.31799/2077-5687-2022-1-38-43.

Введение

Для обеспечения согласованности данных в транзакционных системах применяется так называемый концепт Atomicity Consistency Isolation Durability (ACID) - стандарт того, какие гарантии должна поддерживать база данных [1]. В системах с распределенной архитектурой транзакция может затрагивать несколько баз данных из разных подсистем - микросервисов. Если операция успешно выполнилась на каждом из микросервисов и изменения были сохранены в его базе данных, то транзакция должна быть зафиксирована (commit). Если выполнение операции оказалось неуспешным на любом из микросервисов, то транзакция и, следовательно, изменения в базах данных всех участвующих в операции микросервисов должны быть отменены (rollback). Фиксация изменений в базе данных отдельного микросервиса должна выполняться только после успешного выполнения всей операции, поскольку иначе нарушается парадигма ACID. Для реализации данного принципа в распределённых системах широко применяется алгоритм двухфазного коммита.

1. Распределенные транзакции и двухфазный коммит

Исходя из названия, алгоритм работает в режиме двух фаз: первая фаза, также

называемая подготовительной фазой, выполняет первичные действия в транзакции и проверяет возможность коммита; вторая фаза, также называемая завершающей фазой, отвечает за применение изменений первого шага и обеспечение видимости внешним потребителем. В такой системе отмена транзакций обеспечивается самой базой данных: в случае невозможности выполнить подготовительный этап в какой-либо базе данных распределённой системы, уже подготовленные транзакции отменяются.

Особенностью данного подхода является то, что длительное выполнение подготовительной фазы на одном из узлов распределённой системы приводит к появлению долгоживущих транзакций на остальных узлах такой системы, схематическое обозначение такого подхода см. на рисунке 1. Значительные задержки при выполнении первого шага могут быть вызваны множеством причин. Например, ожидание операций ввода/вывода, сетевые задержки, длительный доступ к другим объектам базы данных, выполнение сложных операций над данными или комбинация этих факторов. В качестве реального примера можно привести сбор статистики в банковской системе за один год.

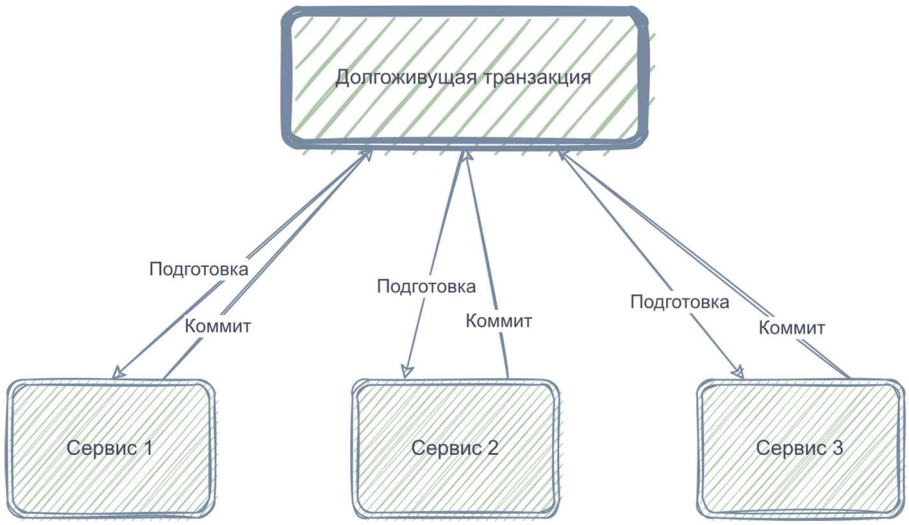

Рис.1. Долгоживущая транзакция при использовании двухфазного коммита

Двухфазный коммит гарантирует, что транзакция либо фиксируется на всех узлах распределенной системы, к которым он обращался, либо прерывается на всех них. Это позволяет избежать нежелательного результата, когда транзакция фиксируется в одном узле системы, но прерывается на другом. Данный подход обычно управляется координатором, который общается с узлами-участниками, которые вместе включают в себя все базы данных, к которым обращается распределенная транзакция. Сложность двухэтапной фиксации транзакции связана со всеми возможными сценариями отказа, которые необходимо программно обработать внутри координатора [2]. Обычно также выделяют специальный объект-менеджер распределенных транзакций. Данным менеджером выполняется действие по работе с сущностью распределенной транзакции. Так, например, транзакционный менеджер должен уметь создавать уникальный идентификатор, называемый идентификатором распределенной транзакции. Клиент также отслеживает другие детали, такие как время начала транзакции, длительность отдельных её частей, время окончания транзакции. Также на него ложится логика работы с механизмами блокировок баз данных для предотвращения взаимоблокировок нескольких распределенных транзакций, логика обработки сценариев ошибок, управление ресурсами баз данных, в том числе соединениями. Самый неприятный сбой происходит после того, как участник подтвердил готовность и до того, как он получил решение о коммите, например, происходит сбой координатора или связи участник-координатор. Обработка таких ситуаций и способов решения проблем, возникших в процессе сбоя, ложится на разработчиков координатора, приводя к высокой связности компонентов, а

также раскрытию некоторой внутренней информации о базах данных узлов-участников [3].

В большинстве случаев долгоживущие транзакции вызывают значительные проблемы с производительностью баз данных. Так, например, для обеспечения атомарности в базах данных применяются блокировки, связанные с транзакциями. И в случае, если другая транзакция желает получить доступ к объекту, захваченному долгоживущей транзакцией, такой транзакции придётся дождаться освобождения блокировки. В общем случае в рамках долгоживущих транзакций не существует способа решения указанных выше проблем.

Методом решения проблемы распределённых транзакций является разбивка распределённой транзакции на последовательность отдельных транзакций. В таком случае каждый узел распределённой системы выполняет свою транзакцию, не дожидаясь исполнения транзакций на других узлах. Для обеспечения атомарности действия в распределённой системе отдельно выполняемые транзакции связываются между собой логически. Логическая связь транзакций обеспечивает компенсацию уже выполненных транзакций в случае неуспешного выполнения одной из них. Под компенсацией понимается отдельная транзакция на узле распределённой системы, позволяющая выполнить логическую отмену уже выполненной на предыдущем шаге транзакции. Таким образом, данный путь позволяет при выполнении транзакции на одном из узлов распределённой системы освободить ресурсы, занятые этой транзакцией.

Из-за особенностей микросервисной архитектуры, которая в данный момент широко применяется в промышленном программировании, использование двухфазного коммита сопряжено с трудностями. Архитектура микросервисов включает в себя множество приложений, реализованных с использованием различных технологий, языков и даже парадигм программирования, в которых часто применяются и поддерживаются разные спецификации, в свою очередь, двухфазный коммит предполагает высокую технологическую связность компонентов, что нивелирует преимущества микросервисной архитектуры, упомянутые выше. Из-за технологических различий в микросервисах и их бизнесовых задачах в них могут применяться разнообразные базы данных, согласовывать транзакции в которых с помощью двухфазного коммита будет достаточно сложно.

2. Способы программной реализации SAGA и их анализ

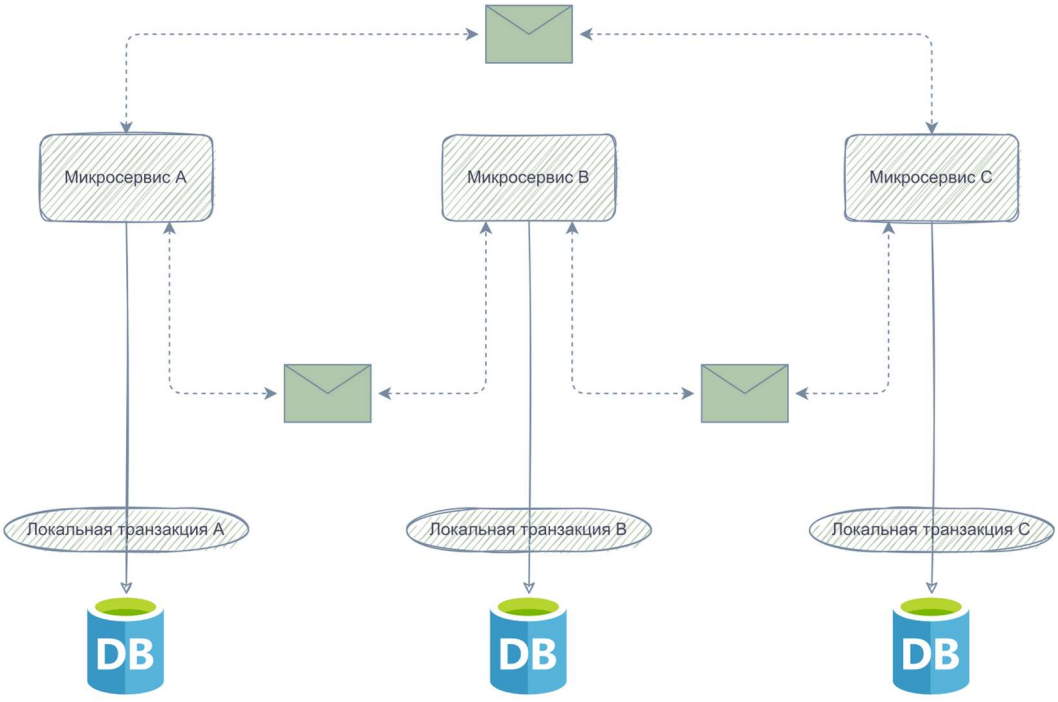

Рис.2. Координация распределённых транзакций способом хореографии

Концепция SAGA паттерна предложена в 1987 году [4]. Существует два способа координации обработки транзакций, применяемых в паттерне SAGA: хореография и оркестровка. Хореография представляет собой набор правил, которые при успешном завершении транзакции на одном из узлов распределённой системы публикуют событие, которое приводит к запуску транзакций в других узлах системы, схематическое обозначение данного метода представлено на рисунке 2. В случае ошибки при выполнении транзакции происходит генерация и публикация события о необходимости компенсации уже выполненных транзакций. В случае оркестровки выделяется отдельный объект, обладающий знаниями о том, какие транзакции должны быть запущены на соответствующих узлах сети. Этот выделенный микросервис отслеживает корректность выполнения транзакций и обеспечивает компенсацию в случае ошибок, схематическое обозначение данного метода приведено на рисунке 3.

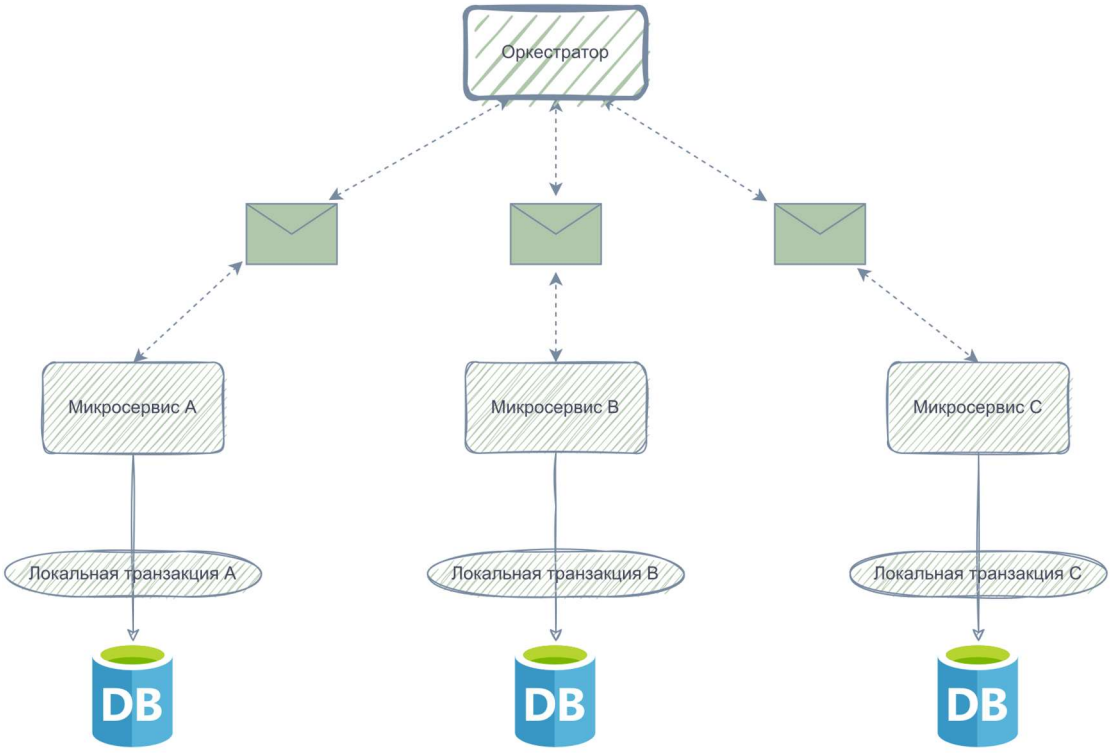

Рис.3. Координация распределённых транзакций способом оркестрации

Существенным недостатком программной реализации оркестровки является необходимость оркестратора напрямую взаимодействовать с каждым узлом распределённой системы, что в том числе означает необходимость дожидаться ответа от каждого из них. Такой подход показывает хорошую работоспособность в относительно маленьких распределённых системах, но в случае сотен и тысяч узлов распределённой системы подход приводит к деградации производительности всей системы и к сложности поддержки кода.

Из недостатков предыдущего подхода вытекают достоинства второго способа координации. Хореография обеспечивает низкую связность узлов распределённой системы между собой, повышает толерантность системы к временной недоступности одного из её узлов, а также позволяет быстрее и эффективнее разрабатывать программные реализации самих узлов.

Таким образом, хореография обеспечивает децентрализованную координацию, при которой каждый микросервис получает сообщения от других и самостоятельно принимает решение о последующем действии, а оркестровка, наоборот, гарантирует централизованную координацию, при которой отдельный компонент (оркестратор) сообщает микросервисам,

какое действие необходимо выполнить далее.

Хореография также сохраняет за каждым компонентом микросервисной архитектуры право самостоятельно определять необходимые правила работы с транзакцией, внутренними ресурсами и хранилищами данных, что снижает связность компонентов и позволяет получить большую свободу в выборе технологий и подходов. Отсутствие единого оркестратора событий также благоприятно сказывается на надёжности системы, не создавая узкого места в виде единой точки входа. С другой стороны, повышается и способность распределенной системы самостоятельно восстанавливаться после падения отдельных компонентов распределённой системы. Например, применение брокеров сообщений может обеспечить доставку событий о необходимости компенсации определённого шага даже после значительного простоя всей системы.

Заключение

В рамках статьи рассмотрены основные особенности долгоживущих транзакций, описаны механизмы, приводящие к деградации производительности распределённой системы. Рассмотрен архитектурный подход SAGA, позволяющий избежать основных минусов распределённых транзакций при построении распределённой системы, а также два основных способа его реализации, приведён анализ каждого из них с точки зрения программной реализации.

Паттерн SAGA решает проблему долгоживущих транзакций, гарантируя сервисам согласованность данных без использования распределённых транзакций, однако модель программирования становится более сложной для реализации и отладки, поскольку возникает необходимость проектировать компенсирующие транзакции, которые смогут отменить изменения, сделанные ранее. Таким образом, паттерн SAGA отлично подходит для распределённых систем без тесной связи транзакций для отмены или компенсации при сбое одной из операций в последовательности. Однако его не следует применять в системах с тесно связанными транзакциями и циклическими зависимостями между узлами сети.

СПИСОК ЛИТЕРАТУРЫ

1. ГОСТ ИСО/МЭК ТО 100-32 - 2007. Эталонная модель управления данными. - М.: Стандартинформ, 2009. - 45 с.
2. Philip A.Bernstein, Eric Newcomer/ Principles of Transaction Processing (Second Edition). - Elsevier, 2009. - 223-244 p.
3. Двухфазный коммит [Электронный ресурс] // Martinfowler - URL: https://martinfowler.com/articles/patterns-of-distributed-systems/two-phase-commit.html (дата обращения: 05.03.2022).
4. Hector Garcaa-Molrna Kenneth Salem/ SAGAS. - N. J.: Department of Computer Science Princeton University Princeton, 08544, 1987. - 20 p.

ИНФОРМАЦИЯ ОБ АВТОРАХ

Чернова Анна Антоновна –
студент кафедры аэрокосмических компьютерных и программных систем
Санкт-Петербургский государственный университет аэрокосмического приборостроения
190000, Санкт-Петербург, ул. Большая Морская, д. 67, лит. А
E-mail: iostream.lib@gmail.com

Курицын Константин Александрович –
доцент кафедры аэрокосмических компьютерных и программных систем, к.т.н.
Санкт-Петербургский государственный университет аэрокосмического приборостроения
190000, Санкт-Петербург, ул. Большая Морская, д. 67, лит. А
E-mail: kuritsynk@mail.ru

INFORMATION ABOUT THE AUTHORS

Chernova Anna Antonovna –
associate professor of Aerospace Computer and Software Systems Department, Ph.D. tech. science
Saint-Peterdburg State University of Aerospace Instrumentation
67, BolshayaMorskaia str.Saint-Petersburg,190000, Russia
E-mail: iostream.lib@gmail.com

Kuritsyn Konstantin Alexandrovich –
Candidate of Technical Sciences, Associate Professor of the Department of Aerospace Computer and Software Systems
Saint-Peterdburg State University of Aerospace Instrumentation
67, BolshayaMorskaia str.Saint-Petersburg,190000, Russia
E-mail: kuritsynk@mail.ru