



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ «Информатика и системы управления»

КАФЕДРА ИУ-7 «Программное обеспечение ЭВМ и информационные технологии»

РЕФЕРАТ
по курсу «Распределенные системы обработки
информации»
НА ТЕМУ:

Паттерн Реплицированные сервисы распределением нагрузки

Студент ИУ7И-12М

_____ Динь Вьет Ань

Преподаватель

_____ Ступников А. А.

Содержание

Введение	3
1 Основные понятия и определения	4
1.1 Понятие реплицированных сервисов	4
1.2 Распределение нагрузки в информационных системах	4
1.3 Паттерн Реплицированные сервисы с распределением нагрузки	6
2 Принципы работы паттерна RLBS	7
3 Преимущества и проблемы паттерна RLBS	9
3.1 Преимущества использования паттерна RLBS	9
3.2 Проблемы при использовании RLBS	10
4 Реальные примеры использования RLBS	12
Заключение	14

Введение

В современном мире, где информационные технологии играют ключевую роль в обеспечении конкурентоспособности бизнеса, вопрос оптимизации работы распределённых систем становится особенно актуальным. С увеличением объемов данных и числа пользователей, с которыми необходимо взаимодействовать, повышается требование к надежности и доступности сервисов. Одним из методов, способствующих выдерживанию этих требований, является использование паттерна «Реплицированные сервисы с распределением нагрузки».

Данный паттерн позволяет создавать несколько экземпляров одного и того же сервиса, что способствует улучшению отзывчивости системы и её отказоустойчивости. Репликация сервисов обеспечивает равномерное распределение входящих запросов между различными экземплярами, что позволяет не только существенно уменьшить нагрузку на отдельные компоненты системы, но и повысить общую производительность. Особенное значение это имеет в условиях высоких нагрузок и в ситуациях, где критически важна скорость обработки данных и непрерывность работы.

Цель работы — проведение информации о реплицированных сервисах и паттерне «Реплицированные сервисы с распределением нагрузки».

В рамках выполнения работы необходимо решить следующие задачи:

- провести основные понятия и определения;
- провести принцип работы паттерна RLBS;
- рассмотреть преимущества и проблемы паттерна RLBS;
- привести несколько примеров использования паттерна RLBS.

1 Основные понятия и определения

1.1 Понятие реплицированных сервисов

Реплицированные сервисы — это сервисы, которые функционируют в нескольких экземплярах (репликах) одновременно для обеспечения отказоустойчивости, масштабируемости и повышения производительности системы. Когда сервис реплицируется, запросы могут быть распределены между репликами для обработки, что позволяет балансировать нагрузку и обеспечивать более быстрый отклик на запросы пользователей.

Реплицирование может быть реализовано на уровне аппаратного обеспечения, программного обеспечения или комбинации обоих. Это позволяет снизить риск отказа сервиса за счет наличия нескольких независимых экземпляров, готовых взять на себя работу в случае проблем с другими репликами.

Кроме того, реплицированные сервисы могут использоваться для обеспечения географической отказоустойчивости, когда реплики размещены в разных регионах или дата-центрах, чтобы обеспечить доступность сервиса в случае отказа в одном из регионов.

Таким образом, реплицированные сервисы играют важную роль в обеспечении надежности, масштабируемости и производительности распределенных систем.

1.2 Распределение нагрузки в информационных системах

Планирование нагрузки — задача, которую надо решать еще на этапе разработки интернет-проекта. Надо понимать, что пользователи должны получать ответы на их вопросы и реакцию на действия максимально быстро, а «падение» сервера может повлечь за собой не только моральные, но и материальные последствия. Первый этап решения проблемы с нехваткой производительности — наращивание мощности сервера и оптимизация рабочих кодов, алгоритмов. Но со временем даже эти меры могут оказаться недостаточными.

В этом случае прибегают к масштабированию: объединению нескольких серверов в один кластер. Может выполняться как вертикальное, так и горизонтальное наращивание структуры. Кластеризация обеспечивает стабильность

работы оборудования при повышенных нагрузках, позволяет распределять потоки запросов на соседние серверы в случае, если один из них выйдет из строя. Эффективность работы системы зависит от того, насколько качественно реализована балансировка нагрузки. Это комплекс методов, способствующих автоматическому распределению нагрузки между компонентами, входящими в один кластер. Она может выполняться с применением как программных, так и аппаратных инструментов.

Наибольшую сложность поиск оптимального решения представляет при работах с публичными информационными системами, социальными сетями, видеохостингом, интернет-магазинами, досками объявлений. Познакомимся с технологией балансировки более подробно, определим ее цели и методы, на основании которых будет распределяться информационная нагрузка

Если говорить о целях, то наиболее часто это:

- Удовлетворение запросов каждого пользователя. Системные ресурсы должны выделяться на обработку каждой заявки. Ситуации, когда обрабатывается один запрос, в то время как все остальные ожидают своей очереди должны быть исключены.
- Высокая эффективность работы каждого сервера. Нагрузка между всеми серверами, входящими в кластер, должна распределяться максимально равномерно. Нельзя допускать ситуацию, когда одни аппаратные устройства загружены на полную, а другие – простаивают.
- Минимизация времени обработки запроса и отклика. Время от момента получения заявки от пользователя и до выдачи ему результата должно быть оптимизировано.
- Предсказуемость. Изначально следует понимать, когда и при работе с какими нагрузками выбранный алгоритм будет максимально эффективным.
- Способность к масштабированию. При наращивании системы выбранные рабочие алгоритмы должны справляться с поставленными задачами.

При выполнении работ следует еще четко понимать, о чем идет речь: о распределении или балансировке нагрузки. Несмотря на свою схожесть, эти

понятия нельзя назвать взаимозаменяемыми. Так, распределение нагрузки предполагает ее равномерное разделение между серверами. А вот балансировка – это уже ее частный случай, учитывающий ряд факторов, подверженных изменению.

На практике широкое применение получили следующие методы распределения.

- Статическое.
- Псевродинамическое.
- Динамическое.

1.3 Паттерн Реплицированные сервисы с распределением нагрузки

Паттерн Реплицированные сервисы распределением нагрузки (англ. Replicated Load-Balanced Services, RLBS) в распределенных системах относятся к стратегии репликации служб на нескольких серверах или узлах и распределения входящих запросов между ними для достижения оптимальной производительности, отказоустойчивости и высокой доступности. В распределенной системе службы реплицируются, чтобы гарантировать, что несколько экземпляров одной и той же службы существуют в разных местах или узлах.

Эта репликация гарантирует, что даже если один экземпляр выйдет из строя, другие смогут продолжать обслуживать запросы, тем самым повышая надежность. Балансировка нагрузки используется для равномерного распределения входящих запросов между этими реплицированными экземплярами. Балансировщики нагрузки действуют как посредники, направляя клиентские запросы к разным экземплярам служб на основе различных алгоритмов, таких как циклический перебор, наименьшее количество подключений или использование ресурсов. Это предотвращает перегрузку любого отдельного сервера и обеспечивает более эффективное использование ресурсов современных сетевых средах.

2 Принципы работы паттерна RLBS

RLBS работают путем объединения репликации служб и балансировки нагрузки для повышения производительности, надежности и масштабируемости распределенных систем. Ниже приведено описание того, как работает RLBS.

1. Репликация сервиса: В RLBS создаются и распределяются по различным узлам сети несколько экземпляров (или реплик) сервиса. Эти реплики могут быть размещены в разных физических местах или на разных серверах для обеспечения избыточности. Репликация гарантирует, что даже если один экземпляр сервиса выйдет из строя или станет перегруженным, другие экземпляры останутся доступными для обработки запросов, что повышает надежность и доступность системы.
2. Балансировщик нагрузки: Балансировщик нагрузки действует как посредник между клиентом и экземпляром службы. Когда клиент отправляет запрос, балансировщик нагрузки решает, какой экземпляр службы должен обработать запрос, на основе предопределенных алгоритмов. Цель состоит в том, чтобы распределить входящие запросы по нескольким репликам, чтобы избежать перегрузки любого экземпляра.
3. Обработка ошибок: если экземпляр выходит из строя или становится недоступным, балансировщик нагрузки автоматически прекращает отправку запросов к этому экземпляру и перенаправляет трафик на другие активные экземпляры. Этот механизм отказоустойчивости гарантирует, что служба продолжит бесперебойную работу даже в случае сбоя.
4. Масштабирование: при увеличении трафика можно создавать и добавлять в пул дополнительные реплики, что позволяет RLBS обслуживать больше клиентов без ущерба для производительности. Балансировщик нагрузки автоматически начнет распределять трафик между этими новыми экземплярами, обеспечивая высокую масштабируемость системы.
5. Сохранение сеанса: в некоторых случаях балансировщику нагрузки может потребоваться обеспечить перенаправление запросов из одного сеанса пользователя в один и тот же экземпляр службы. Это часто

называют «прикрепленным сеансом» или «сохранением сеанса», и он помогает поддерживать взаимодействие со службой с сохранением состояния.

Балансировщики нагрузки могут использовать несколько алгоритмов для определения способа распределения трафика.

- Циклический алгоритм: запросы отправляются каждому экземпляру службы по очереди.
- Наименьшее количество подключений. Запросы отправляются на экземпляр с наименьшим количеством активных подключений.
- На основе ресурсов: балансировщик нагрузки отправляет запросы на основе доступных ресурсов (ЦП, память) каждого экземпляра.
- Геолокация: запросы направляются к ближайшей реплике, чтобы уменьшить задержку.

3 Преимущества и проблемы паттерна RLBS

3.1 Преимущества использования паттерна RLBS

Реплицированные службы с балансировкой нагрузки (RLBS) предлагают многочисленные преимущества в распределенных системах, повышая производительность, надежность и масштабируемость. Основные преимущества включают:

1. **Высокая доступность и отказоустойчивость:** репликация служб гарантирует, что несколько экземпляров службы доступны на разных узлах. Если один экземпляр выходит из строя, балансировщик нагрузки перенаправляет трафик на другие реплики, обеспечивая постоянную доступность и минимальное время простоя.
2. **Масштабируемость:** RLBS может легко обрабатывать растущий трафик, добавляя больше экземпляров службы. По мере развертывания новых реплик балансировщик нагрузки эффективно распределяет входящие запросы, позволяя системе масштабироваться горизонтально без снижения производительности.
3. **Повышенная производительность:** балансировка нагрузки распределяет запросы между несколькими экземплярами службы, предотвращая перегрузку любого отдельного экземпляра. Это приводит к лучшему использованию ресурсов, сокращению задержек и сокращению времени отклика для пользователей.
4. **Географическое распределение нагрузки:** реплицируя службы в разных регионах, RLBS может направлять запросы пользователей на ближайшую реплику, сокращая задержку и улучшая пользовательский опыт. Это особенно полезно в глобальных системах, где пользователи распределены по разным местам.
5. **Лучшее использование ресурсов:** Балансировка нагрузки обеспечивает равномерное распределение трафика по всем доступным экземплярам, предотвращая чрезмерное использование определенных ресурсов при недостаточном использовании других. Это помогает оптимизировать

использование ресурсов, сократить узкие места и избежать проблем с производительностью.

6. Автоматическое переключение при отказе: В случае отказа узла или службы балансировщик нагрузки автоматически перенаправляет трафик на рабочие реплики. Этот бесшовный механизм переключения при отказе гарантирует, что пользовательский опыт останется непрерывным, несмотря на сбой.

3.2 Проблемы при использовании RLBS

Хотя паттерн RLBS предлагает множество преимуществ в распределенных системах, они также представляют несколько проблем, которые необходимо решить для поддержания эффективной производительности и надежности. К этим проблемам относятся:

1. Управление согласованностью: репликация служб и данных на нескольких узлах поднимает проблему поддержания согласованности. Если один экземпляр службы обновляет данные, обеспечение того, чтобы все остальные реплики также обновлялись, может быть сложным. Это особенно проблематично в распределенных системах, где задержка сети или разделы могут задерживать обновления, что приводит к несогласованности данных или устареванию данных.
2. Задержка сети и разделение: в географически распределенных системах задержка сети может вызывать задержки в коммуникации между репликами и балансировщиками нагрузки. В случае сетевых разделов (когда части сети становятся изолированными) поддержание синхронизации между всеми репликами становится затруднительным, что может привести к ухудшению обслуживания или просто для определенных пользователей.
3. Узкое место балансировщика нагрузки: сам балансировщик нагрузки может стать единой точкой отказа или узким местом, если им не управлять должным образом. Если балансировщик нагрузки перегружен или выходит из строя, вся настройка RLBS может выйти из строя. Обеспечение масштабируемости и отказоустойчивости балансировщика нагрузки

(например, за счет избыточности или распределенных балансировщиков нагрузки) имеет решающее значение.

4. Сложность масштабирования: хотя RLBS разработан для поддержки масштабирования, динамическое добавление или удаление реплик может создавать проблемы. Система должна точно отслеживать нагрузку трафика и использование ресурсов, чтобы решить, когда масштабировать, а неправильно управляемое масштабирование может привести к избыточному или недостаточному предоставлению ресурсов.
5. Службы с сохранением состояния и службы без сохранения состояния: RLBS лучше всего работает с службами без сохранения состояния, где каждый запрос независим и не требует отслеживания данных сеанса. Для служб с сохранением состояния поддержание данных сеанса (например, закрепленных сеансов) может усложнить процесс балансировки нагрузки, поскольку система должна гарантировать, что запросы от одного и того же пользователя поступают в один и тот же экземпляр для поддержания непрерывности сеанса.

4 Реальные примеры использования RLBS

RLBS широко используются в различных отраслях промышленности для обеспечения высокой доступности, масштабируемости и отказоустойчивости распределенных систем.

Несколько реальных случаев использования RLBS:

1. Глобальные сети доставки контента

Глобальные сети доставки контента, такие как Akamai, Cloudflare и Amazon CloudFront, распространяют статический и динамический контент (например, изображения, видео, веб-сайты) пользователям по всему миру. Поставщики CDN реплицируют контент в нескольких центрах обработки данных по всему миру. Балансировщики нагрузки направляют запросы пользователей в ближайшую или наименее загруженную реплику на основе географического положения и сетевых условий. Это снижает задержку, повышает производительность и гарантирует постоянную доступность контента, даже если региональный сервер выходит из строя.

2. Платформы электронной коммерции

Крупные платформы электронной коммерции, такие как Amazon, eBay и Alibaba, обрабатывают огромные объемы трафика, особенно в пиковые периоды, такие как распродажи. Такие сервисы, как аутентификация пользователей, поиск товаров, оформление заказов и обработка платежей, реплицируются в разных регионах. Балансировщики нагрузки распределяют входящие запросы по этим реплицированным сервисам, чтобы избежать перегрузки любого отдельного экземпляра. RLBS гарантирует, что даже при большом трафике клиенты будут получать быстрые и надежные транзакции, и ни один отдельный экземпляр сервиса не станет точкой отказа.

3. Услуги облачных вычислений

Поставщики облачных вычислений, такие как AWS, Microsoft Azure и Google Cloud, предлагают инфраструктуру как услугу (IaaS), платформу как услугу (PaaS) и программное обеспечение как услугу (SaaS)

клиентам по всему миру. Эти поставщики облачных вычислений реплицируют критически важные услуги (например, базы данных, виртуальные машины, оркестровку контейнеров) в нескольких центрах обработки данных. Балансировщики нагрузки автоматически распределяют запросы клиентов на ближайший или наиболее доступный экземпляр службы, обеспечивая высокую доступность и масштабируемость для приложений, размещенных на их платформах.

4. Платформы социальных сетей

Гигантам социальных сетей, таким как Facebook, Twitter и Instagram, необходимо обрабатывать миллиарды ежедневных взаимодействий пользователей, включая публикации, лайки, комментарии и загрузки мультимедиа. Эти платформы реплицируют пользовательские сервисы, такие как новостные ленты, обмен сообщениями и уведомления, в разных регионах. Балансировщики нагрузки направляют запросы пользователей в ближайшую или наименее загруженную реплику, обеспечивая низкую задержку для пользователей по всему миру. RLBS также имеет решающее значение для управления огромными всплесками трафика во время таких событий, как прямые трансляции или вирусные публикации.

Заключение

Паттерн Реплицированные службы балансировки нагрузки (RLBS) необходимы для обеспечения высокой доступности, масштабируемости и надежности в распределенных системах. Реплицируя службы на нескольких узлах и используя балансировщики нагрузки для распределения трафика, RLBS повышает производительность и минимизирует время простоя даже во время сбоев. Он широко используется в таких отраслях, как электронная коммерция, облачные вычисления, социальные сети и онлайн-игры. Хотя RLBS предлагает значительные преимущества, он также сопряжен с такими проблемами, как поддержание согласованности и управление сетевыми проблемами. Внедрение RLBS с передовыми методами гарантирует, что распределенные системы смогут справиться с растущим спросом, обеспечивая при этом бесперебойную работу пользователей.

Цель работа была достигнута. Все задачи были решены.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Replicated Load-Balanced Services [Электронный ресурс]. Режим доступа: <https://www.oreilly.com/library/view/designing-distributed-systems/9781491983638/ch05.html>.
2. Distributed System: Replicated Load-Balanced Services [Электронный ресурс]. Режим доступа: DistributedSystem: ReplicatedLoad-BalancedServices.
3. Replicated Load-Balanced Services (RLBS) in Distributed Systems [Электронный ресурс]. Режим доступа: https://www.geeksforgeeks.org/replicated-load-balanced-services-rlbs-in-distributed-systems/?ref=header_outind.