

Программное обеспечение "Программный модуль игровой среды взаимодействия с сервером"

Описание функциональных характеристик программного обеспечения

Оглавление

1. Назначение.....	5
2. Общая архитектура.....	5
3. Минимальная конфигурация	7
4. Отказоустойчивая конфигурация	7
5. Горизонтальная масштабируемость и балансирование нагрузки.....	8
6. Планирование выполнения задач по расписанию.....	8
7. Цепочки задач (роутинг)	8
8. Взаимодействие с БД	9
9. Безопасный мягкий запуск и остановка ноды шины данных.....	10
10. Гарантированное исполнение задач	10
11. Сервисы восстановления	11
12. Управление лимитами	11
13. Мониторинг работы	11
14. Адаптеры (хэндлеры).....	11
15. Сохранение истории выполнения задач	11
16. Работа с внешними системами	11
17. Синхронная и асинхронная доставка.....	12
18. Проксирование запросов.....	12
19. Режимы выполнения задач	12
20. Консоль.....	12

Введение

Документ содержит описание основных функциональных характеристик программного обеспечения (далее-ПО) "Программный модуль игровой среды взаимодействия с сервером"(шина),

Раздел «Назначение» содержит сведения о назначении ПО "Программный модуль игровой среды взаимодействия с сервером",

В раздел «Общая архитектура» содержится информация о шине данных, взаимодействии шины с задачами, экземплярах очередей, хранилища шины данных и описание синхронизации данных.

Раздел «Минимальная конфигурация» описывает работу шины данных с ActiveMq, Nginx и MariaDb.

В раздел «Отказоустойчивая конфигурация» предоставлено описание работы шины данных с различными серверами.

Раздел «Горизонтальная масштабируемость и балансирование нагрузки» содержит информацию о балансировке поступающих http-запросов на шину.

В разделе «Цепочка задач (роутинг)» предоставлен пример цепочки на рисунке 2 и описание самого механизма.

В разделе «Взаимодействие с БД» приведено описание создания новых задач, после получения запроса и работа стратегии шард.

Раздел «Консоль» содержит в себе перечень подразделов, таких как «Мониторинг и управление», «Управление нодами шины данных», «Управление типами задач», «Просмотр истории выполнения задач», «Добавление внешних систем» и «Управление конфигурацией внешних систем».

Термины, определения, сокращение

Шина	Standalone java приложение, реализующее конвейерную обработку задач. Может работать на нескольких узлах
Node (нода)	Отдельный компьютер, на котором запущен экземпляр шины. Как правило, шина работает на нескольких нодах одновременно
TaskType	Параметризованное описание задачи (шаблон), хранящееся в базе данных, на базе которого шина создаёт экземпляры задач
Task (таска, задача)	Экземпляр TaskType, созданный шиной для выполнения конкретного задания в определенный момент времени
Взаимодействие	Набор задач, либо одна задача, запускающихся в определенной последовательности для выполнения какой-либо цели. Например, скачивание файла с удаленного сервера, парсинг его и сохранение данных в БД
Main	Главная БД шины, хранящая системную информацию
Shard (шард)	Подчинённые БД шины, работающие совместно с main. Шардов может быть несколько
Репликация	Перенос шиной данных между main и shard базами данных
Консоль шины	Frontend шины. Выполняется в браузере, позволяет управлять работой шины, тасок и т.д
Мягкий старт/стоп шины	Запуск и остановка нод шины из консоли. При этом обеспечивается корректное завершение всех исполняемых задач
Адаптер	Компонент корпоративной шины (набор взаимодействий), обеспечивающий взаимодействие различных источников в требуемых форматах, например, 1С-Адаптер, X5-Адаптер и т.д
Handler (обработчик)	Java класс, реализующий какой-либо алгоритм. Хендлер привязывается к TaskType, таким образом TaskType реализует действия, делегируя их хендлеру
System (система, alias)	URL внешней системы, к которой может обращаться задача
Method (метод)	Идентификатор сущности внутри системы, например, таблица в БД, или маппинг ресурса в рамках веб-сервера
Scheduler (шедулер, планировщик задач)	Механизм, запускающий задачи по расписанию
Watchdog	Алгоритм внутри шины, отслеживающий и останавливающий задачи, время выполнения которых превысило определенный предел
Глобальная задача	Задача группирующая несколько дочерних задач выполняющихся в цепочке
Дочерняя задача	Это задача привязанная к какой-то глобальной задаче. Одна из задач цепочки
Цепочка задач	Последовательность из нескольких задач создаваемых и выполняемых как по порядку, так и параллельно, при этом они все привязаны к одной и тоже глобальной задаче
Routing (роутинг)	Хранящееся в базе данных описание порядка выполнения задач в цепочке
Endpoint	Относительный адрес на шине для приема входящих запросов

1. Назначение

Программный модуль игровой среды взаимодействия с сервером предназначен для сбора данных из различных источников и доставки этих данных во внешние системы. Шина данных позволяет внешним системам выполнять задачи синхронно и асинхронно, запрашивать статус выполнения задачи и результат выполнения задачи. Для синхронных задач устанавливать гарантированные временные интервалы выполнения. На шине данных присутствует модуль планирования задач, позволяющий задавать гибкое расписание.

Шина данных позволяет балансировать нагрузку между поставщиками и получателями информации. На шине имеется функциональность для агрегации получаемых и отправляемых данных. Шина данных позволяет управлять количеством параллельных клиентов к внешней системе. Шина данных регистрирует отправляемые и получаемые данные внешних систем.

С помощью консоли шины данных администраторы получают возможность управлять жизненным циклом задачи, производить мониторинг состояния системы и анализировать различные метрики работы системы.

2. Общая архитектура

Общая схема работы Шины данных представлена на рисунке 1.

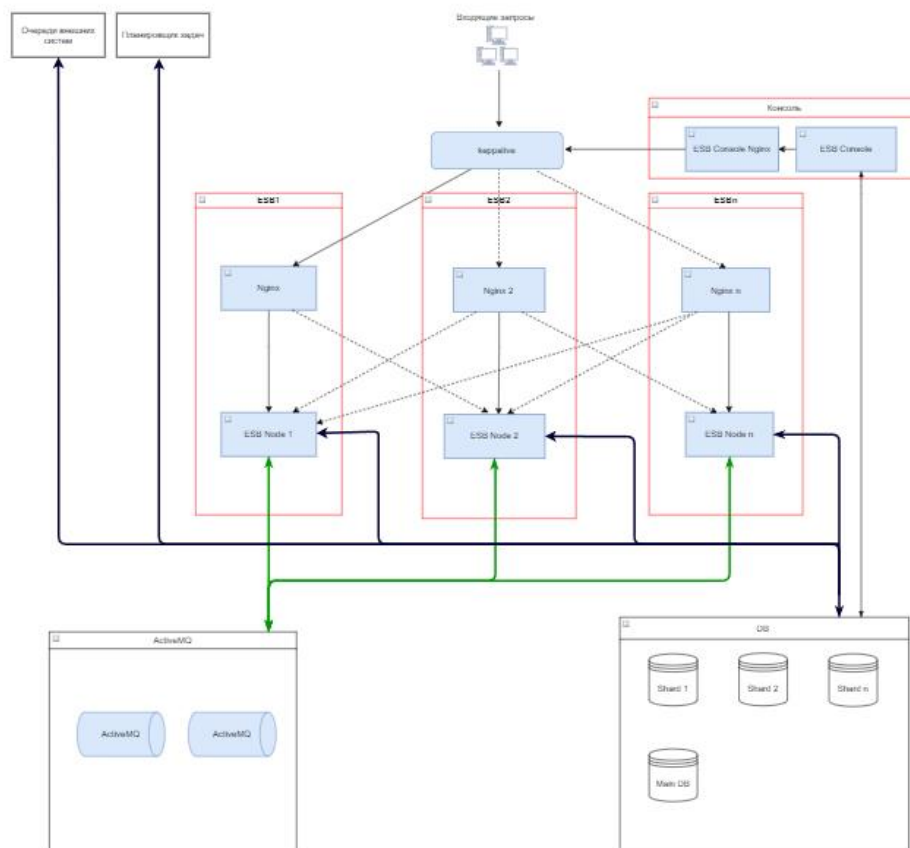


Рисунок 1

Шина данных представляет собой масштабируемую и отказоустойчивую систему. Данное свойство обусловлено независимыми нодами шины данных, которые можно добавлять и удалять в зависимости от нагрузки и степени отказоустойчивости.

Задачи могут поступать в шину от внешних систем через http(s) запросы, генерироваться на основании сообщений от внешних брокеров очередей, а также порождаться самой шиной, исходя из прописанной в ней логики, например, создание задач по планировщику. После создания, задача сохраняется в БД, что гарантирует то, что она не потеряется в дальнейшем. После этого задача поступает в виде сообщения с идентификатором задачи в очередь (используется ActiveMq). Из этой очереди сообщение забирается одной из работающих в кластере нод (так осуществляется балансирование нагрузки между нодами шины), и поступает в обработку.

Очередь использует файловую БД KahaDB, и является отказоустойчивой. На двух независимых серверах работают отдельные экземпляры очереди. Одна из них работает в master-режиме, другая находится в slave-режиме.

Определение, какая из очередей является мастером, происходит за счёт блокировки файла, который создаётся на распределённой файловой системе (используется GlusterFS). Очередь, которая не смогла заблокировать файл

переходит в slave-режим и периодически пробует снова заблокировать файл. Если это удаётся, то она становится мастером, и загружает в себя все данные из БД KahaDb (эта база также располагается на распределённом файловом хранилище, доступном обеим экземплярам очереди).

В качестве хранилища данных шина использует реляционную СУБД. Каждая отдельная БД представляет собой шард, который хранит только часть всех данных. Шарды позволяют балансировать нагрузку на хранилище данных и обеспечивать постоянное среднее время выполнения задачи независимо от нагрузки. Каждый шард является сам по себе отказоустойчивым кластером (в последней версии используется MariaDb Galera Cluster). Кроме того, используется файловое хранилище, отказоустойчивость которого обеспечивается GlusterFS. Файловое хранилище используется для данных, сохранённых как файлы.

Для повышения эффективности и производительности шина данных использует кэши данных и запросы только для чтения в зависимости от логики запроса.

3. Минимальная конфигурация

В простейшем случае, в минимальной конфигурации, шина данных может работать в единственном экземпляре, с одним ActiveMq, Nginx и MariaDb, запущенными на одном и том же сервере. В этом варианте конфигурации в случае выхода из строя любого из этих компонентов, система становится неработоспособной. Данная конфигурация поддерживает гарантированное выполнение задач, не связанное с выходом из строя узла конфигурации.

4. Отказоустойчивая конфигурация

Для построения максимально отказоустойчивой конфигурации, необходим Keeralive сервис, два сервера для шины данных, Nginx, ActiveMq, кластер БД из трёх серверов (например, Galera Cluster) и файловая система GlusterFS.

Шина принимает внешние запросы через Keeralive. Если Nginx на первом сервере не функционирует, происходит переключение на второй сервер. На Nginx запрос перенаправляется на шину на том же сервере. Если шина не отвечает, Nginx перенаправляет запрос на шину на втором сервере. Также имеется возможность настройки балансировки запросов сразу на все сервера.

Два экземпляра ActiveMq работают в режим master-slave и используют для сообщений БД KahaDb, находящуюся на распределённой файловой системе

GlusterFS. Обе ноды шины взаимодействуют с экземпляром ActiveMq, находящимся в режиме master. В случае, если этот экземпляр по той или иной причине падает, второй экземпляр переходит в master-режим, подтягивает из KahaDb все сообщения, которые находились в первом экземпляре очереди, и шина данных автоматически переключается на работу с этим экземпляром ActiveMq.

5. Горизонтальная масштабируемость и балансирование нагрузки

Балансировка поступающих http-запросов на шину настраивается на сервер Nginx, равномерно распределяя нагрузку параллельно по всем серверам. При этом один из серверов шины остается резервным, и используется только в случае отказа остальных серверов.

Все активные экземпляры ESB участвуют в обработке поступающих задач. Распределение задач между нодами шины происходит через очередь ActiveMQ. Добавление нового экземпляра ESB осуществляется простым подключением этого экземпляра к БД и ActiveMq, с которыми работают остальные экземпляры. Также используется шардинг БД.

6. Планирование выполнения задач по расписанию

Планирование выполнения задач по расписанию происходит через добавление администратором шины данных через консоль шины данных нового планировщика. При создании нового планировщика администратор выбирает тип задач и указывает расписание выполнения задачи. Консоль шины данных предоставляет возможность запуска и остановки созданных планировщиков, изменения расписания выполнения задачи.

7. Цепочки задач (роутинг)

Пример цепочки задач (роута) представлен на рисунке 2.

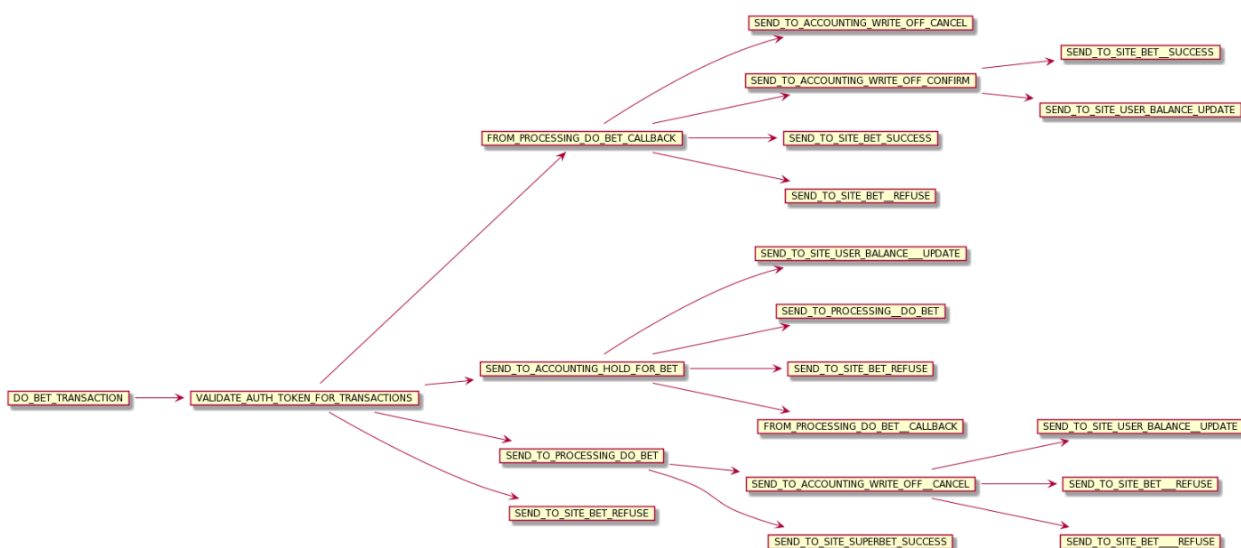


Рисунок 2

Механизм роутинга позволяет выстраивать из отдельных типов задач цепочки (маршруты), где каждая следующая задача цепочки порождается по определённому правилу на основании результата выполнения предыдущей задачи.

В каждом роуте есть глобальная задача, которая считается выполненной, только после того как завершились все её дочерние задачи в рамках цепочки.

8. Взаимодействие с БД

После получения запроса на Шине создается задача и сохраняется в Базе данных. Создание новых задач происходит последовательно на разных шардах - так реализуется механизм шардинга. Для распределения нагрузки Шина данных имеет возможность динамически переключаться между разными БД (шардами). Это также обеспечивает постоянное среднее время выполнения задачи независимо от нагрузки.

В архитектуре Базы данных есть центральная БД (Main DB). Предусмотрен только один экземпляр Main DB. Шардов может быть любое количество, но минимально один шард. Механизм шардинга позволяет балансировать нагрузку, добавляя при необходимости новые шарды.

На Main DB поступают данные со всех шардов. Она предназначена для отображения данных о задачах на консоли Шины данных. Main DB хранит упрощённую (неполную) информацию о задачах. На Main DB сохраняется конфигурация "типов задач" (TaskType), которые порождают конкретные экземпляры задач (Tasks). Создание задачи происходит на основе TaskType,

когда поступает запрос от внешней системы, по планировщику или как результат выполнения предыдущей задачи в цепочке.

На один и тот же шард записываются задачи, например, если ключ определенного шарда содержится в инфо к задаче. Последовательно выполняемые задачи также сохраняются на один и тот же шард, сериальность будет обеспечиваться выполнением одной задачи данного типа. Есть еще ряд задач, для которых предусмотрен механизм выбора конкретного шарда. В остальных случаях шард выбирается рандомно из перечня доступных.

Задачи не создаются на Main DB, данные о задачах попадают на нее в результате работы отдельной задачи по репликации задач. Репликация происходит по расписанию. Предусмотрен сервис, который производит перенос. Есть таблица, в которой указывается время последнего переноса. Задачи, у которых время переноса больше, чем указанное в таблице, переносятся с шардов на Main DB, и в таблицу записывается время последнего переноса.

Синхронизация данных между шардом и Main DB происходит в двух направлениях. Первое направление связано с миграцией изменений по задачам с шарда в центральную БД, второе - связано с миграцией справочных данных из Main DB на шард. Оба потока миграции идут параллельно и выполняются с помощью встроенного механизма планирования выполнения задач.

9. Безопасный мягкий запуск и остановка ноды шины данных

Шина данных предоставляет механизм мягкой остановки и запуска ноды шины данных. Данный механизм позволяет перед полной остановкой ноды шины данных завершить или дождаться завершения выполнения активных задач, прекратить принимать новые задачи и запросы от внешних систем. Таким образом, отключение ноды шины данных не влияет на выполнение задач, выполняющихся как на данной ноде, так и на других. Данный функционал позволяет безопасно отключать и включать ноды шины данных, не влияя на работу задач.

10. Гарантированное исполнение задач

В шине данных реализован механизм гарантированного исполнения задач. Это означает, что в случае, если задача не смогла выполниться по любой причине (недоступность внешнего или внутреннего ресурса, отключение

питания и т.п.), она снова поступит в обработку после заданного в настройках конкретного типа задачи временного интервала. Количество попыток выполнения задачи по умолчанию не ограничено, но может быть также установлено в настройках конкретного типа задачи.

11. Сервисы восстановления

На шине данных предусмотрены ряд сервисов восстановления для защиты от случаев зависания задач, восстановления задач при отказе одной из нод и других сбоев.

12. Управление лимитами

Для каждого типа задачи могут задаваться лимиты на максимальное время выполнения задач и максимальное количество одновременно выполняющихся задач, связанных с одной внешней системой.

13. Мониторинг работы

На шине собирается множество различных метрик, позволяющих оценить ее текущую загрузку, состояние, проблемы. Каждая нода шины данных предоставляет возможность экспортирования этих данных в различные системы мониторинга.

14. Адаптеры (хэндлеры)

Каждый тип задачи имеет свой адаптер (хэндлер), определяющий, каким образом будут обрабатываться данные этого типа задач.

15. Сохранение истории выполнения задач

Все переходы между статусами задач (Ожидает обработки, Выполняется, Выполнена, Ошибка, Отменена, Заморожена) логируются в базе данных (идушие подряд одинаковые переходы между статусами агрегируются для удобства просмотра).

16. Работа с внешними системами

Добавление внешних систем осуществляется без модификации исходного кода. Шина данных позволяет ставить внешнюю систему на паузу или отключать как в рамках типа задачи, к которому она привязана, так и глобально сразу для всех типов задач. Если внешняя система поставлена на паузу, то задача, связанная с этой внешней системой, при попытке выполнения опять отправляется в очередь. Попытка ее выполнить будет происходить через настраиваемый интервал для статуса «в ожидании». Если

система отключена, то задача отменяется, и пишется соответствующее сообщение.

17. Синхронная и асинхронная доставка

Задачи могут быть синхронными и асинхронными. При выполнении синхронных задач при запросе к Шине происходит создание и обработка задачи, и только после этого внешней системе возвращается статус по задаче.

По асинхронным задачам Шина сразу отвечает внешней системе статусом «ОК», записывает задачу в Базу данных на шард. После этого сразу отправляет id задачи в очередь ActiveMQ на выполнение. Фактически в очередь отправляется сообщение, что задачу с определенным номером нужно обработать. Это сделано для максимального быстродействия. Та нода Шины, которая в данный момент свободна, забирает задачу в обработку.

18. Проксирование запросов

Шина данных поддерживает возможность проксирования HTTP(S) запросов от одной системы к другой, с возможностью их логирования и настройкой максимальной длительности запроса. Сама пересылка запроса может быть выполнена в режиме синхронного или асинхронного выполнения.

19. Режимы выполнения задач

Шина данных поддерживает возможность выполнения нескольких задач одного типа одновременно, либо одну за другой с контролем очередности (по времени создания задачи) и без контроля очередности.

20. Консоль

20.1 Мониторинг и управление

Шина данных предоставляет удобный сервис мониторинга работоспособности и управления в виде специальной консоли управления (отдельное приложение), права доступа в которую настраиваются через Active Directory. В консоли можно осуществлять мониторинг выполнения задач, в том числе, на графике в режиме реального времени, отменять задачи, производить настройку параметров типов задач, управлять доставкой данных во внешние системы.

20.2 Управление нодами шины данных

Через консоль управления имеется возможность включать/отключать отдельные ноды шины данных. В случае отключения, шина данных перестаёт принимать новые задачи, а задачи запланированные для обработки на этой ноде будут выполнены.

20.3 Управление типами задач

В любой момент времени, через консоль управления, на уровне типов задач может быть произведено включение/отключение создания новых задач этого типа и автоотмена задач этого типа после создания.

20.4 Просмотр истории выполнения задач

Консоль предоставляет функционал для просмотра выполнения задач.

20.5 Добавление внешних систем

Добавление внешних систем, в которые должны быть отправлены определённые данные, сбор и отправка которых уже реализованы на шине, осуществляется без модификации исходного кода через консоль управления в несколько действий (добавить новую систему, добавить метод системы, привязать метод к нужному типу задачи).

20.6 Управление конфигурацией внешних систем

Через консоль можно добавлять, изменять внешние системы, ставить их на паузу и отключать.