|  |  |  |
| --- | --- | --- |
| МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ  РОССИЙСКОЙ ФЕДЕРАЦИИ | | |
| Федеральное государственное бюджетное образовательное  учреждение высшего образования  «Пермский государственный национальный исследовательский  университет» | | |
|  | Кафедра математического  обеспечения  вычислительных систем | |
| **ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №4** | | |
|  | | Работу выполнил студент  группы ПМИ 1,2 4 курса  механико-математического  факультета  Мячев Н.С.  «\_\_\_» \_\_\_\_\_\_\_\_2019 г. |
|  | |  |
| Пермь 2019 | | |

# Общее описание приложения

Цель: изучение методов коммуникации процессов в сети, а также средств динамического конфигурирования распределенных приложений.

Выбранная предметная область для разработки приложения – лингвистическая обработка документов. Данная предметная область была выбрана потому что она хорошо подходит для разработки распределенной системы, т.к. существуют ситуации, когда пользователям необходимо обрабатывать большое количество крупных документов, однако собственных мощностей у них может не хватать. Так же использование серверных технологий позволит писать различные клиенты, которые могут быть запущены на различных видах и типах устройств, но при этом выполнять одинаковые функции, это также позволить снизить нагрузку при портировании.

Приложение позвляет:

* //перечислить возможности

# Архитектура системы

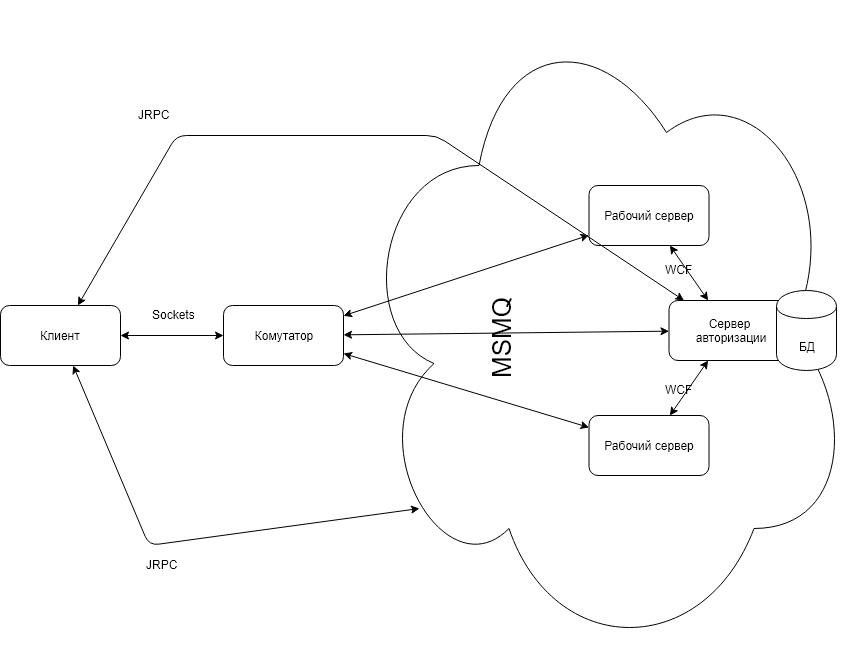


Рисунок 1 Архитектура приложения

Архитектура приложения представлена на рисунке 1.

Данная архитектура была разработана с учетом специфики предметной области, а также для возможности многопоточной работы. Узким местом данной архитектуры является коммутатор, однако он не выполняет сложных операций, поэтому при достаточно хорошем оборудовании, даже при большом количестве клиентов проблемы не возникнут.

## Алгоритм работы приложения

В первую очередь включается комутатор/диспетчер, он выполняет функции распределения нагрузки и организации общения между сервером и клиентом. Диспетчер должен находиться всегда по одному и тому же адресу, иначе клиент не сможет подключиться к нему. После включаются сервера. При подключении сервер сообщает при помощи систем очередей сообщений о своем местоположении в сети диспетчеру, для дальнейшей работы. Среди серверов важно подключить сервер авторизации, который занимается авторизацией пользователей, а также сбором статистики. После того как подключен сервера авторизации и рабочий сервер, клиент может начать работу.

Клиент первый раз подключается к диспетчеру при авторизации или регистрации. Диспетчер посылает запрос на авторизацию серверу авторизации и при получении положительного ответа, отправляет клиенту сообщение о успешной авторизации, а также адрес сервера авторизации для дальнейшего отслеживания статистики или переподключения.

После успешной авторизации клиент может начать работу. Чтобы выполнить некоторую функцию клиент обращается к диспетчеру, диспетчер передает клиенту адрес сервера, который не является нагруженным и может выполнить эту функцию. После того как клиент получил адрес сервера, он подключается к нему посредствами JRPC протокола и отправляет данные для обработки. После выполнения работы, сервер возвращает результат, а также через WCF отправляет запрос серверу авторизации для изменения статистики этого пользователя.

# Архитектура компонентов

## Клиент

Представляет из себя обычно клиентское оконное приложение. Организует взаимодействие пользователя с системой. Состоит из 2 основных окон: окна авторизации и рабочего окна. В архитектуре заложены средства взаимодействия с диспетчером (сокеты) и сервером (JRPC).

Алгоритм работы

При запуске приложения открывается окно авторизации, через него пользователь может зайти в систему или открыть окно регистрации. После успешной авторизации открывается рабочее окно, через которое пользователь может загружать свои документы, выбирать функции и методы обработки, а также получать результаты.

## Диспетчер

## Сервер

# Методы коммуникаций компонентов системы

## Клиент-диспетчер

Для связи между клиентом и диспетчером используются сокеты. Выбор обоснован тем, что у диспетчера фиксированный ip, что позволяет удобно подключаться к нему через TCP. Также подключение через сокетов поддерживается почти любым оборудованием, что позволит создавать клиенты для разных устройств и систем. Также сокет соединение является довольно быстрым.

## Диспетчер-сервер

Для связи диспетчера и сервера выбрана система очередей сообщений. Это обусловлено тем, что системы очередей сообщений поддерживают автоматическую коммуникацию, это означает, что если сервер на короткое время пропадает из сети, то сообщение будет доставлено спустя некоторое время, что положительно сказывается на работоспособности всей системы. Также использование очередей сообщений приемлемо, потому что сервера и диспетчер поддерживаются разработчиками, что позволит ставить все необходимое программное обеспечение.

## Сервер-сервер

Для связи сервера с сервером используется WCF. Это обусловлено тем, что сервера будут написаны на C#, что позволяет использовать удобные средства коммуникации, специально разработанные для .NET приложений.

## Клиент-сервер

Для связи клиента с сервером используется JRPC. Протокол удаленного вызова процедур буквально создан для этой задачи. JRPC позволит производить асинхронный вызов функций на сервере, а также поддерживается большинством устройств и систем, что позволит писать различные клиенты.

# Способ передачи данных

//тут надо подумать

# Структура передаваемых данных

//тут тоже надо подумать

# Отказоустойчивость системы

## Отключение одного из серверов

До вызова функции

Если отсутствуют сервера, способные выполнить заданную функцию, то диспетчер отправит сообщение о том, что нет доступных серверов.

Во время вызова функции

Если сервер отключится во время выполнения функции, то по истечению таймера клиенту выдастся сообщение о таймауте ожидания.

Восстановление

Если сервер будет подключен заново, то он свяжется с диспетчером и сообщит свое новое местоположение, после чего может продолжать выполнять свою работу.

## Отключение диспетчера

## До подключения клиента

Если диспетчер не был подключен или его адрес был изменен, то при попытке авторизации клиент сообщит о невозможности подключения к диспетчеру/серверу авторизации.

После подключения клиента

Если диспетчер был отключен после подключения клиента, то при попытке выполнить функцию будет выведено сообщение о невозможности подключиться к серверу.

Восстановление

Диспетчер не может быть восстановлен на другом узле сети, если только не поменять адрес этого узла. Однако при возвращении диспетчер продолжит работу.

# Работа с базой данных