

• Getway

Принимает соединение по TCP
While(true)
socket.accept

new MyTask(socket)



















Перенаправляет клиентский запрос менее
загруженному серверу
И отправляет обработанный запрос клиенту

new MyTask(socket)

new MyTask(socket)

Принимает соединение по UDP
While(true)
dataGram.recieve(packetIn)

Присылает адрес и порт
менее загруженного сервера

	Client	GateWay	LoadBalncer	Server
Client				
GateWay				 
LoadBalncer				
Server				

GateWay

- Должен иметь публичный адрес и порт для внешних TCP соединений.
 - Должен иметь порт для входящих UDP пакетов
 - Хранить данные по менее загруженному серверу (адрес сервера и порт)
 - Эти данные приходят по UDP соединению от **LB** (каждые 100мс)
1. Принимать соединение от клиента по TCP
 2. Перенаправлять (работать как мост) соединение серверу.
 3. Принимать обработанные данные от сервера
 4. Отправлять обработанные данные клиенту

GateWay

Потоки

1. UDP – слушатель LB

- Данные от LB содержат адрес и порт менее загруженного сервера

2. TCP - клиентский слушатель

- Принимает новые соединения от клиента и создает отдельный поток для обработки каждого нового соединения.
 - Задачи этих потоков отправлять данные клиента серверу и отправлять обработанные данные обратно клиенту.

LoadBalancer

- Порт для приема данных от серверов
 - **Данные:** адрес, порт и нагрузка каждого отдельного сервера
- Хранит **данные** о серверах/сервере
 - Если о серверах, то должен уметь находить и возвращать данные о менее загруженном сервере
 - Если о серверах, необходимо очищать структуру данных от не активных серверов (те, которые какое то время не присылали о себе данных)
- Раз в определенное время (100 мс) посылает данные GateWay о менее загруженном сервере (адрес и порт сервера)

LoadBalancer

Потоки

- UDP – слушает сервера
 - принимает данные от серверов
- UDP – отправляет данные GW
 - Отправляет данные менее загруженного сервера GW
- **Если будем сохранять несколько серверов*
- *Раз в какое то время будет очищать нашу структуру данных от не активных серверов*

Address
Port
Load

Address
Port
Load

ServerData

Address, port, load, latUpdateTime.

Class Source

```
Map <int, ServerData>  
ArrayList<ServerData> = new ....  
    getBest()  
clear() –удаление не активных серверов
```

Server

- Сервер имеет свой порт на котором он отрабатывает TCP соединения от **GW**
- Сервер должен знать свою загруженность (количество соединений)
- Отправлять свои данные по UDP LoadBalancer'у каждые 1000мс
- Принимает новые соединения от **GW** и создает отдельный поток для обработки каждого нового соединения.
- Отправлять обработанные данные обратно **GW**