Proceedings of ICAPP '12
Chicago, USA, June 24-28, 2012
Paper 12312

A HIGH LEVEL ARCHITECTURE OF A SIMULATION FUTURES TRADING SERVER

Weihua Hao
School of Computing, DePaul University
243 South Wabash, Chicago, IL, 60604
Tel: (312)532-4772, Email:whao@cdm.depaul.com

Abstract –*Maybe people has well known the knowledge of stocks, but I think most of them are not familiar with futures. The main reason is because some futures concepts are not easy to understand if you do not have a lot of professional financial knowledge. But there is another reason that is so many futures trading software is too complicated for us. Fortunately I used to be working as a futures trading Software Company for almost two years. So I want to build a simple introductory futures trading platform to let people easily to learn futures. The goal of this paper is a part of that scheme. This paper is only concentrating on the back-end of the simulation trading platform.*

I. INTRODUCTION

The aim of this simulation futures trading platform is to let users learn everything about futures and experience the trading of futures. And we want user learn something from the simulation trading of futures. So the trading server is the core part of this system.

We hope out trading server can deal with all kind of request from clients correctly and meanwhile with a high performance. And as the growth of users, the server need to be easily scalable to keep the same performance. This server is really a very challengeable distributed system.

II. BASIC CONCEPTS OF FUTURES

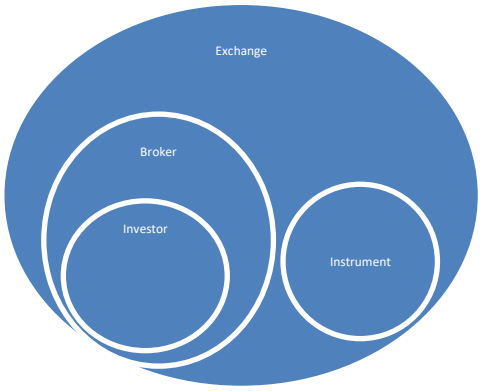
EXCHAGE: Exchange is the place we make a trade of futures. It will provide some standard instruments used by all the investors to trade. And the exchange will make this deal legal and done.

INSTRUMENT: A futures instrument is an agreement to buy or sell a specified amount of a product or financial instrument at an agreed upon price on or before a given date in the future.

BROKER: A broker is a firm or individual who executes orders to buy or sell contracts on behalf of clients and charges them a commission.

INVESTOR: Investors are a group people who want to get some benefits from trading of futures and based on their own prediction of price. We also can call them speculator.

We can see the basic relationship among this concepts from the figure1.



Figures1: The relationship among these concepts.

III. BRIFE BUSSINESS REQUIREMENT

The simulation trading platform aims to provide the same trading experience like the futures trading system of real world, it need provide two main functionalities which are trading and quotation. Basically, we can use some different user stories to depict our business requirements.

User story 1: as a user, I want to create an account for trading. After creating this account, I can transfer some virtual money to this account. If I got enough balance, then I can use this account to buy or sell the instruments which provided by the exchange.

User story 2: as a user, I want to get the real-time quotations of the instruments which I subscribed from the exchange.

User story 3: as a user, When I got some positions about some instruments, and I need get the real-time update of my account benefits if the price of these instruments changed, so I can make a quickly decision to sell or buy these instruments.

User story 4: as a user, I get a trade notification if the order I placed a while ago get traded. For simplicity, we assume the order cannot be split to get traded partially.

User story 5: as an administrator, I want know the running status of all the components of this system. And I can easily find which components are down, after getting some components down, I can restart these components. And meanwhile I hope the other components can be aware of the joining of the new instance of these components.

User story 6: as a settlement administrator, I want to make a settlement daily after the trading time is over.

IV. SYSTEM REQUIREMENT

This system is a distributed system with the demand of high performance and scalability and availability, and meanwhile all the data related with user and instruments must be persisted. We hope our system is developed based on the Linux Operating System and the communication between processes is based on the socket.

V. ASSUMPTIONS

Because this is a simulation system and it is difficult and expensive to use the real-world data, and we want our system is simple enough to accomplish, we introduce some assumptions into this system.

Assumption 1: The exchange is a virtual exchange, it is come up with by us, and our system only support one exchange.

Assumption 2: The instruments is virtual as well, and all instruments is finical instruments, it can only can be settled with cash.

Assumption 3: This system does not support partially trading, that means one order cannot be split into some sub orders to get traded.

Assumption 4: The investors cannot create account from the exchange, they only can register from a broker. In our system, we assume there is only one broker existing, and this broker is the exchange itself.

Assumption 5: This system can only support two kind of order which are market order and limited order.

VI. DETAILED RULE ANALYSIS

- Exchange: Although this exchange is a virtual exchange, we can assign some property to it like a real one. we can see the main properties form the figure 2.

Exchange ID	Exchange Name	Weed day Pre-open	Weed day open	Week day Pause	Weed day open	Weed day Settlement	Weekend
SEF	SE Futures Exchange	8:00am-8:30am	8:30am – 11:30am	11:30am-12:30pm	12:30pm-3:00pm	3:00pm-4:00pm	closed

Figure 2: the detailed exchange information

- Instrument

Instrument is the most significant data of this system, Each instrument has some specific trading hours in a trading day. In this system, we suppose all the instruments has the same trading hours as the exchange. And the trading hours like the following image Figure 3.

SE - FINANCIALS (ELECTRONIC)							
Name	Symbol	Exchange	Sat/Sun Open	Weekday Pre-Open	Weed day open	Week day Pause	Weed day Settlement
10-Yr Interest Rate Swap	SR	SE	Closed	8:00 AM – 8:30 AM	8:30am – 11:30am 12:30pm-3:00pm	11:30am-12:30pm	3:00pm-4:00pm
10-Yr US T-Note	ZN	SE	Closed	8:00 AM – 8:30 AM	8:30am – 11:30am 12:30pm-3:00pm	11:30am-12:30pm	3:00pm-4:00pm

Figure 3: the detailed trading hour information

Each Instrument has a symbol, and a month associated this instrument. We assume the expiration date of this instrument is the third Friday of that month. And we use a list of monthly symbols like the table below.

MONTHLY SYMBOLS											
Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
F	G	H	J	K	M	N	Q	U	V	X	Z

Figure 4: the monthly symbols table

Since we focus on the financial instrument, we use the point-value to define a price of an instrument.

MARKETGROUP - FINANCIALS (ELECTRONIC)					
Name	Symbol	Exchange	Size	Months	Point-Value
10-Yr Interest Rate Swap	SR	SE	$20 x Index	H,M,U,Z	$20
10-Yr US T-Note	ZN	SE	$20 x Index	H,M,U,Z	$20

Figure 5: the price information of instruments

We allow user to use very high leverage relative to stock markets. Each user's rate of leverage might be different, depends on their credit. Then how does it work? The answer is margin.

- Initial Margin

When you open a futures contract, the futures exchange will state a minimum amount of money that you must deposit into your account. This original deposit of money is called the initial margin.

- Maintenance Margin

the maintenance margin is the lowest amount an account can reach before needing to be replenished. For example, if your margin account drops to a certain level because of a series of daily losses, brokers are required to make a margin call and request that you make an additional deposit into your account to bring the margin back up to the initial amount.

MARKETGROUP - FINANCIALS (ELECTRONIC)				
Name	Symbol	Exchange	Init. Margin	Maintenance Margin
10-Yr Interest Rate Swap	SR	SE	$ 2035	$ 1017.5
10-Yr US T-Note	ZN	SE	$ 1485	$ 742.5

Figure 6: the margin information of instruments

- Investor

Investors is the core of this system. A lot of data are associated with Investors. Like fund, trade, order, position, commission rate, margin rate, etc. Simply we

can use the figure 7 to illustrate these data relationship.

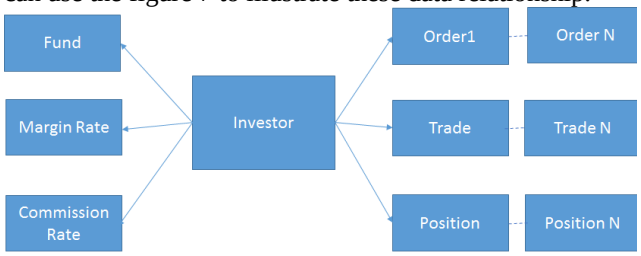


Figure 7: the data associated with investor

VII. USER CASE DIAGRAM

Basically, we can abstract the user case from the business requirements.

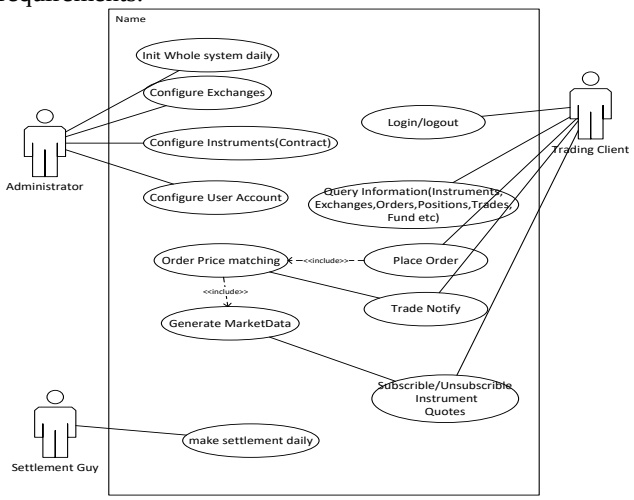


Figure 8: User case diagram

VIII. HIGH LEVEL ARCHITECTURE

According to the characteristic of this system and applying some design pattern. We come up with the high-level architecture like the image below.

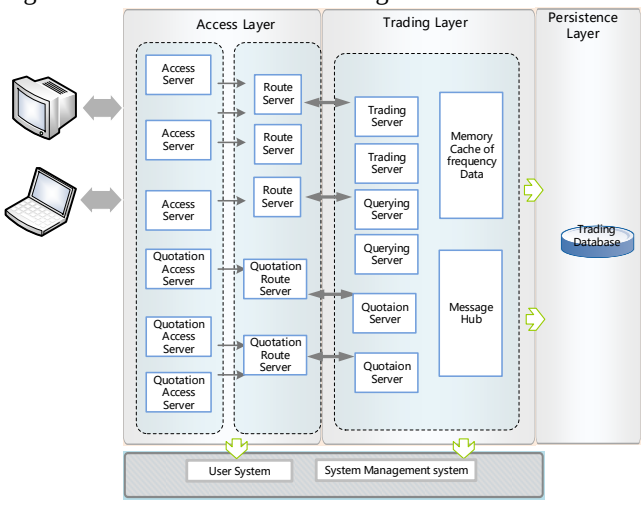


Figure 9: The high-level architecture

Access Server: This is the interface server with the clients. Clients connect to Access server with TCP/IP protocol.

Route Server: After the interface server received the request from the client. It would transfer this request to a specific route server, and the route server would send this request to a trading server according a special routing rule.

Trading Server: This is the core server for trading, it will perform the price matching and make a trade. After making a trade, it will push data back to the client and push some type of data to the Message Hub Server.

Querying Serer: This server only concentrating on dealing with the querying request.

Memory Cache Server: This server provides a high-performance technology to retrieve data and a concise way to share data within different computers. Some commonly used data like instruments information will be stored in the memory cache. Possibly we can use some sort of third-part memory cache product like Redis.

Message Hub: This is a very isolated server just providing the function of transferring data. This type of transferring data is on basis of subscribing. A process send a piece of data to the Message Hub, and Message Hub will send this data to all the subscribers who are interesting in this data. Nowadays there are a lot of standard third-part products like Rabbit MQ and IBM MQ providing these functions.

Quotation Access Server: This is the interface server providing receiving the query and pushing the real-time quotation data to the client.

Quotation Route Server: After the Quotation Access Server received the request from the client. It would transfer this request to a specific route Quotation Route Server, and the Quotation Route Server would send this request to a Quotation server according a special routing rule.

Quotation Server: The main responsibility is to get the newest quotation data from the Message Hub and then push back to the Quotation Route Server.

User System: This server mainly provide authentication of users. Each other server can identify the user through the User System.

System Management Subsystem: This is a sub-system for the administrator user to monitor and manage the entire system. And this system also expose some interfaces to other server to get the information they need. In this subsystem, we can use some third-part management cluster product like Zookeeper.

Trading Database: This is located in the persistence layer, it is kind of relationship database produce such as Oracle and SQLServer.

IX. STATE MACHINE OF WHOLE SERVER

Since we know the open hour, pre-open hour and settlement hour each day of the exchange, we easily can get a state machine about the entire system. Each process in this system need to be aware of this state machine and co-work in term of this state machine. Please check out the detailed state machine from the image below.

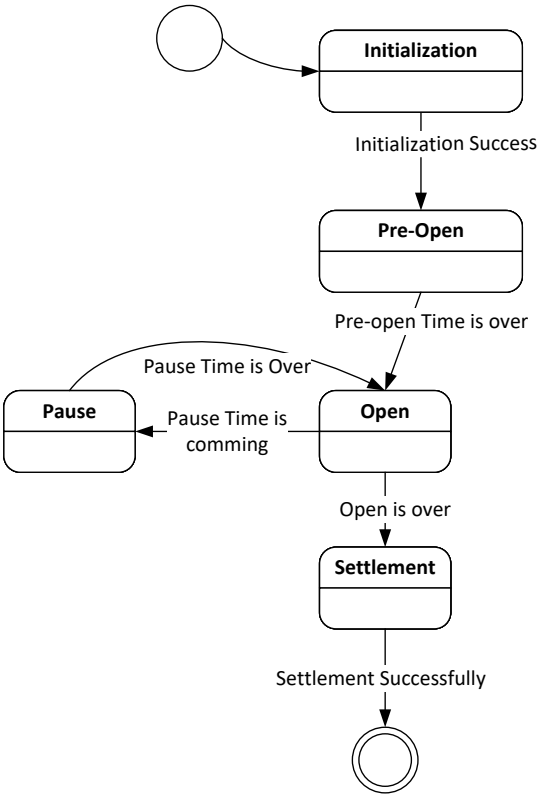


Figure 10: The state machine of entire server

X. COMPONENT SEQUENCE DIAGRAM

- Sequence Diagram of placing order

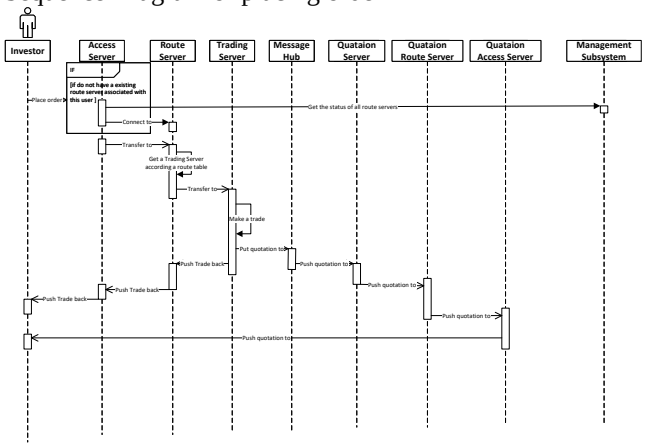


Figure 11: The sequence diagram of placing order

- Sequence Diagram for login

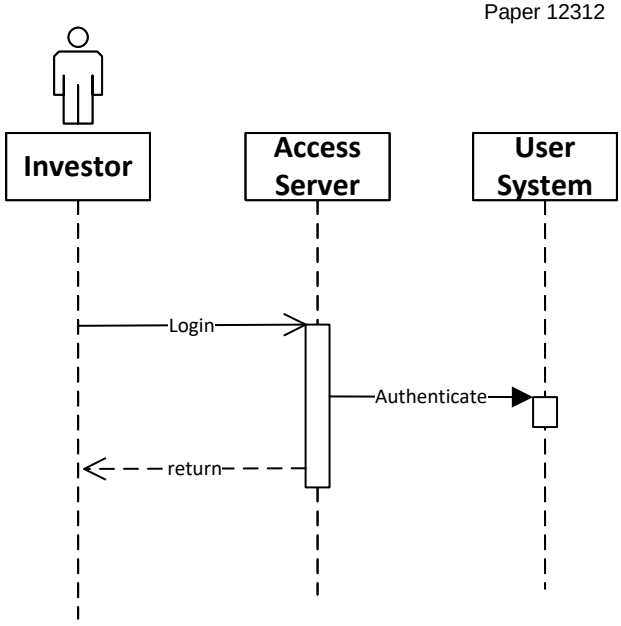


Figure 12: The sequence diagram of login

- Sequence Diagram for querying

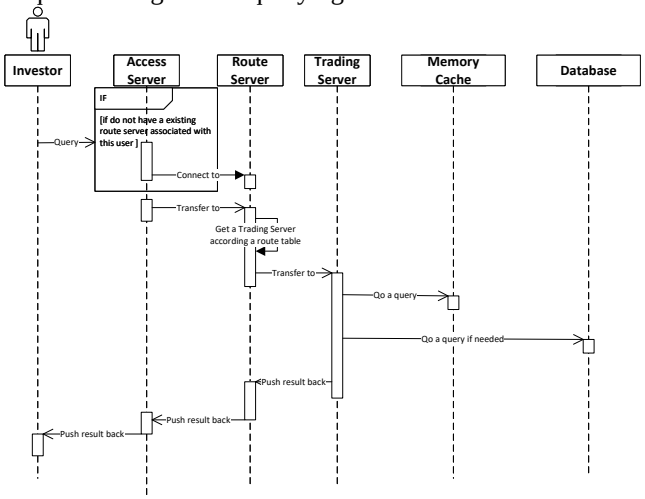


Figure 13: The sequence diagram of querying

XI. THE PRICIPLE OF ROUTING

How can we make sure the data will return to the correct client? And besides returning correctly we should make our system scalable and no single failure problem. To achieve this challengeable accomplishment. I prefer to use a fully connected graph to guarantee no single failure point and introduce a management subsystem to let each server know the other servers which it concerns very easily. We can see the connection channel between each servers from the image below.

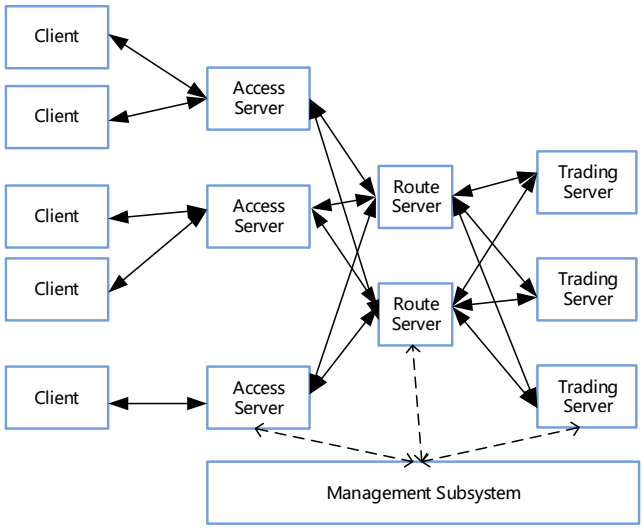


Figure 14: The illustration of connection of servers

And I will illustrate the route principle of placing order with the image below

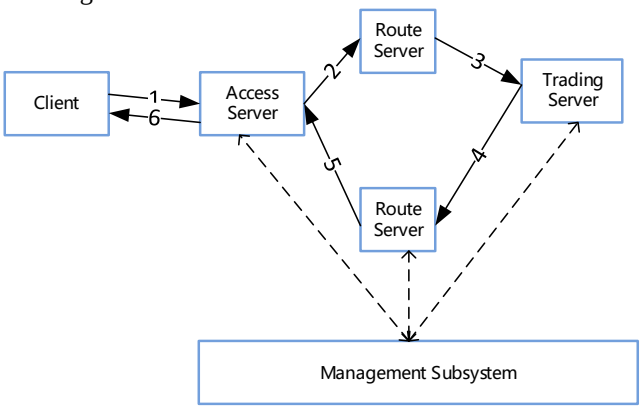


Figure 15: The data transferring route of placing order

Edge 1: When a client connected an instance of Access server, this instance of access server will keep a map of map<User ID, Socket Handle> , and at the same time it will report to the Management Subsystem this user ID have logged in this instance of Access Server. So inside the Management Subsystem we need create a map to store the relationship between user ID and the instance of Access Server. After the Management Subsystem updating the map, it will notify all the Route Server to sync with the new version of this map.

Edge 2: Actually according to our fully connected graph theory, this instance of Access Server have a TCP channel with each Route Server. So it will choose a Route Server to transfer a request smartly, our first aim is to achieve a simple Round Robin load balance algorithm to transfer request.

Edge 3: Like the transferring in Access Server the Route Server also will use special rule algorithm to transfer the request to a Trading Server. Acutely we need to use another map table to finish his job. When each server trading started, it will registry a list of Instruments which it can handle to the Management subsystem, meanwhile all the Route Server will sync this list and make sure it has the newest version. So Route Server will transfer request with the same instrument information to an identical Trading Server,

Edge 4: When the Trading Server finished this request, it will use Round Robin algorithm to return a package to a Route Server.

Edge5: When a Route Server received a package from Trading Server, it will unpack this package to check which user ID is engaged with this package, so it will check the inside route map table to send this package to a specific Access Server which this user ID is logging in it.

Edge 6: After an Access Server receiving a package from Route Server, it will return this package to a Client which it concerns about this package according to the inside map table which kept on the Edge 1 step.

If this request is a kind of querying request, the route will be a little bit different with the placing order request. We can see the different edge 3 in red color.

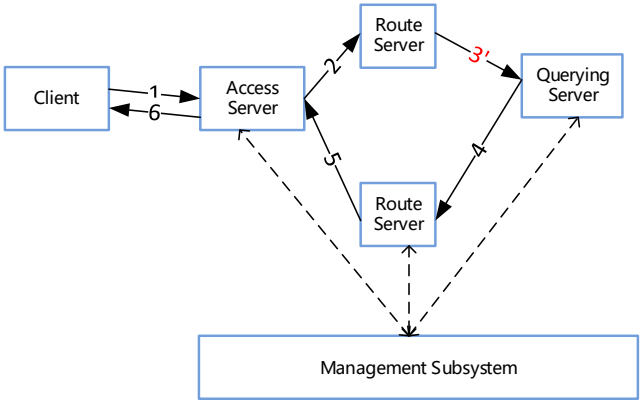


Figure 16: The data transferring route of querying

Edge 3': Not like the transferring data to Trading Server, Route Server will use Round-Robin load balance to decide which Querying Server will be transferred.

In terms of the characteristic of quotation, the server will push the quotation data to client actively.so the stream of quotation data is more different from the trading data.

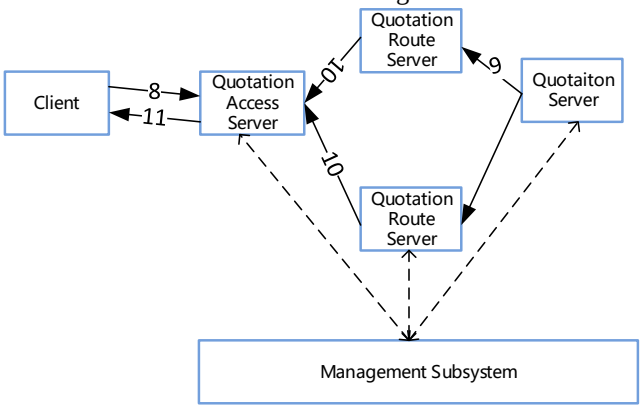


Figure 17: The data stream of quotation

Edge 8: A client connects to an instance of Quotation Access Server. And it will subscribe a list of instruments to notify the Quotation Access Server to push these quotations of this list of instruments back.

Edge 9: We suppose one instance of Quotation Server only

concerns the quotations of some of the instruments. When Quotation Server get a new updated Quotation data of specific instrument, it will choose one of the Quotation Route Servers according to the Round-Robin load balance Algorithm.

Edge 10: Like the Quotation Server, the Quotation Route Server will broadcast the data to all the connected Quotation Access Server.

Edge 11: The Quotation Access Server will push this instrument quotation data to a certain clients according to the subscribing list.

XII. AVAILABLITY

- ACCESS SERVER
  As the number of client increases, we can add some Access Server to our system very easily, and we can limit the number of client for each Access Server. Possibly one instance of Access Server could support thousands of TCP long connections. Maybe in front of Access Server we can put a TCP load-balance server to let client connect the Access Server more transparently.

- ROUTE SERVER
  Route Server is a server with a very simple logic. And it also can be scaled horizontally like Access Server.

- TRADING SERVER
  Trading Server is the most complicated server in this system. From logical level we are going to start two instances of Trading Server for each instrument, one is a master server, and the other one is a slaver. So the following image can illustrate the physical deployment of all the Trading Servers.

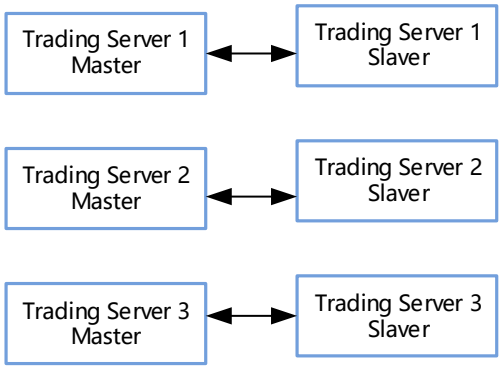
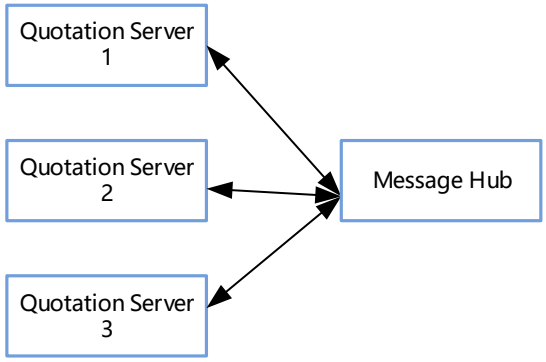


Figure 18: The deployment of Trading Servers

The master server is in charge of all the business logic, but another job is to sync everything changed on order queue to the slaver server. And the main responsibility is to upgrade when master is down, and do the business logical.

- QUERYING SERVER
  Not like Trading Server, The business logic of Query Server is very independent. So in our design the Querying Server is of the high scalability and availability. It can be scaled on demand.

- QUOTATION SERVER
  For the high availability of the Quotation Server we need to introduce a Message Hub Product with a public package selectively. So that means we can start a cluster of Quotation Server and make sure each package only can be handled by one instance of Quotation Server.



- QUOTATION ROUTE SERVER
  Like the Route Server, the Quotation Route Server can be deployed as a cluster. It can be scaled horizontally to fulfil the increase of data transferring.

- QUOTATION ACCESS SERVER
  This server also can be scaled in terms of the number of the clients.

- MESSAGE BUS
  We are going to use the Rabbit MQ as the Message Bus Server. As claimed by the Rabbit MQ, Several Rabbit MQ servers on a local network can be clustered together, forming a single logical broker. And Queues can be mirrored across several machines in a cluster, ensuring that even in the event of hardware failure your messages are safe.

- MEMORY CACHE
  Because of the high performance and the new cluster functionality of Redis, it is one of the best choices for us to apply it to as a memory cache server.

- Management Subsystem
  Zookeeper is a perfect open-source product to share the significant server information. And it provide the high consistence and availability.

REFERENCES

1. https://www.quandl.com/collections/futures
2. https://dzone.com/articles/better-explaining-cap-theorem
3. https://www.rabbitmq.com/
4. http://redis.io
5. https://zookeeper.apache.org/