Analysis of Multi Leg Stock Trading System

TAUSIF ANWAR

M-Tech, Computer Science and Engineering,
Dr.MGR Educational and Research Institute University.
Chennai-600095, India.

***Abstract*— In this paper it performs the share exchange operation. It is used to viewing the all current share information, company information, and share holder information, Moreover, the share holders and company can interact with the task for market share exchange. Moreover, the Share holder's and the company can interact with the task for selling share and buying shares. The main aim of this paper is to maintain the Share information with the particular share information and the all share details. This task is used to reduce the customer transaction-time as well as user can able to access easily. The Online Share trading is used for displaying all share information including the share amount. In this page, if the user select the particular share codes, then it will shows all information regarding that particular share, as well as user can register their account information. The company can launch the new shares and they can set the initial share amount of the particular share. Company and Admin can view all information of members**

***Keywords*— SHARE TRADING, MULTI- LEG SYSTEM.**

I. INTRODUCTION

We present the design, optimization and analysis of a highly flexible and efficient multi-leg share trading system. While the expressive power of multi-leg trading brings significant value to investors, it also poses major challenges to stock exchange architecture design, due to additional complexities introduced in performance, tradability, and fairness. Performance can be significantly worse due to the need to coordinate transactions among multiple stocks at once.

This paper studies the performance of multi-leg trading under different fairness constraints and variability in order price and order quantity. We identify the major performance bottlenecks when using traditional atomic commitment protocols such as 2- Phase Commit (2PC), and propose a new look-ahead algorithm to maximize transaction concurrency and minimize performance degradation. Currently, stock exchanges do not support automated multi leg trading of this type due to the overhead and complexity. The primary aim of the new system is to speedup transactions. Look-ahead approach is not based on speculation and need not roll back or keep track of the data dependencies.

II. PROBLEM ISSUES

This system involves a lot of manual entries with the applications to perform the desired task. It is beneficial to understand general stock exchange architecture and how orders are handled for single-leg trading. The architecture is typically multi-tiered and consists of multiple gateway nodes (GW) running order dispatchers, multiple execution venue nodes (EV) matching orders for one or more specified stock symbols, and multiple history recorder (HR) nodes writing trading results into persistent storage. EVs are the most critical components of the stock exchange where all trades are executed.

- Usage of papers in the payment process leads to less efficiency, less accuracy and less productivity.
- Less control of Amounts
- Time delay between the payment and its receipt.
- Persons who are present in different part of the world cannot transact efficiently.

III. RELATED WORK

A. ALGORITHM DESCRIPTION.

We identified that a major performance overhead of the base-line prototype is in coordinating a consensus: trade or not trade, for a Multi vs. Single or Single vs. Multi trade. A solution is to allow incoming orders to continue being processed before the coordinator sends final replies, which we called look-ahead. However it is not easy to determine which orders should be allowed to trade or not in the blocking duration. That's because the final tradability of the multi-leg order is unknown until the end of the multi-leg session and the priority ordering constraint must be enforced. To preserve consensus atomicity, the commitment that "the blocked multi-leg order can trade in EVHPQ" should not be compromised in the blocking duration before it receives its final reply (for non-coordinators) or before it sends out the final reply (for the coordinator). We find that as long as the above commitment could be kept (which means at least one single-leg order

remaining in the order book could be traded with the blocked order), allowing more orders to be traded in the blocking duration will neither affect the tradability of the multi-leg order nor violate its priority ordering Based on this finding, a look-ahead algorithm is designed, which allows orders in the order book to trade freely in blocking duration as if the blocked multi-leg order doesn't exist, if no conflicting condition is detected as violation of the commitment.

To implement the solution, we classify incoming orders into different types based on whether allowing their trade will violate the tradability of the blocked multi-leg order. We also designed one state machine to keep track of the order's transition from one type to another.

IV. MODULES DESCRIPTION

1.Admin Login:

This form is used for maintain the information of shareholder and all company information by using the admin authentication. This form is contain two modules

1: View Share Holder Information

2: View Company information

Admin have a full provision for accept or reject the information from shareholder as well as the company.

1: View Share Holder Information:

This module is used to view information regarding the all share holders. If admin can select particular share holder code, it will give some more information of the particular share holder's.

2: View Company information:

This module is used to view information regarding the all Company. If admin can select particular company code, it will give some more information of the particular company.

2. Share Holder Registration:

In this module a new share holder going to register gives their information like, Name, password, Acno, bank, working details, address, e-mail id etc. when a new shareholder registered, then the user details are updated in the admin page. The admin wants to see shareholder details click the name, and then show the entire shareholder details. But the admin can't change the shareholders details it can see only details.

Company Registration

New Company gives their information like, Name, password, Company details, address, and e-mail id, share Details, etc.

Buyer form

The member gives the inputs for their login into the system. Inputs are shareholder id and password. If they are valid, he enters into the system for selecting and buys the particular share from the list.

Seller form

The Company gives the inputs for their login into the system. Inputs are Company id and password. If they are valid, he enters into the system for launching the new shares and set the initial amount of ht shares.

View Shares

This module is used for View all share information. All share details were viewed form the data base. Share holder and the company can have a provision for view all share information through with this modules.

V. ARCHITECTURE DESIGN AND IMPLEMENTATION

Once the system has been designed, the next step is to convert the designed one in to actual code, so as to satisfy the user requirements as excepted. If the system is approved to be error free it can be implemented. When the initial design is done for the system, the department is consult for acceptance of the design so that further proceedings of the system development can be carried on. After the development of the system a demonstration was given to them about working of the system. The aim of the system illustration is to identify any malfunctioning of the system.

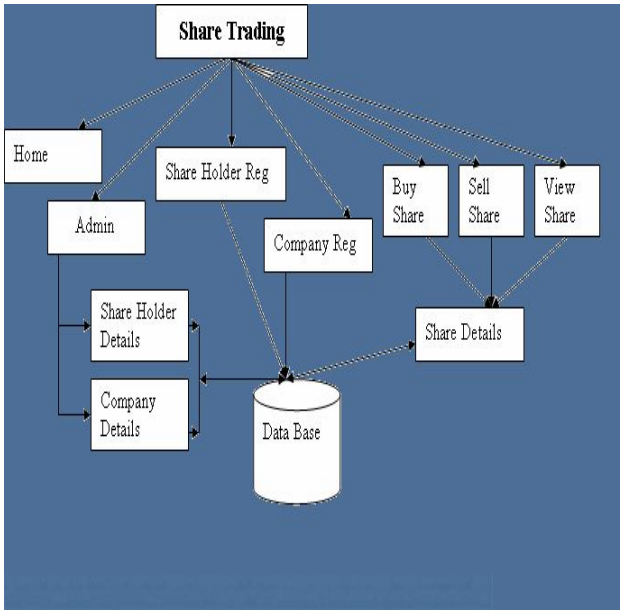

Fig. 1 Overview of Share Trading.

Implementation includes proper training to end-users. The implemented software should be maintained for prolonged running of the software.

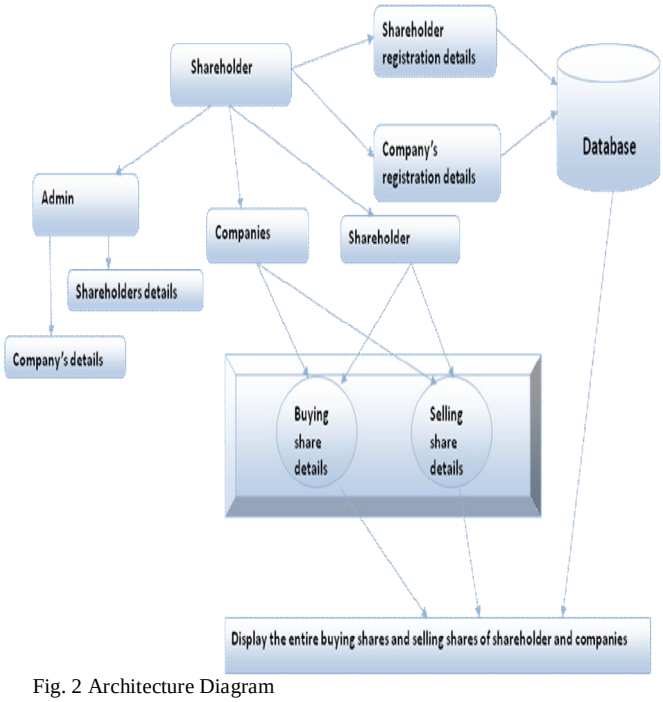

Fig. 2 Architecture Diagram

The systems architect establishes the basic structure of the system, defining the essential core design features and elements that provide the framework. The systems architect provides the architects view of the users' vision.

In above diagram contains admin login, shareholder registration page and company registration. In admin page has to shown entire details about shareholders and companies details. A new user wants to register open shareholder page as way a new company wants to register opened company registration page. Buying share page has shown details about selling shares of companies. Share trading maintained view of shares.

VI. Conclusions

The Stock Trading supports and improves many of the core functionality of the organization i.e. bidding also helps in quick easy monitoring of the reports that have been automatically generated as and when the user performs some transactions in the system. Using such a system helps the organization in minimizing the time consumed in fulfilling the day-to-day functionality's and cutting down the expenses incurred on the same. We have implemented two stock trading systems. The first is a base-line prototype based on 2PC and efficient session management. The second uses our look-ahead algorithm to reduce the overhead of coordinating multi-leg transactions on multiple nodes.

References

[1] M. Tamer Ozsu. 2007. Principles of Distributed Database Systems (3rd ed.). Prentice Hall Press, Upper Saddle River, NJ, USA.

[2] Michael J. Fischer. 1983. The Consensus Problem in Unreliable Distributed Systems (A Brief Survey). In Proceedings of the 1983 International FCT-Conference on Fundamentals of Computation Theory, Marek Karpinski (Ed.). Springer-Verlag, London, UK, 127-140.

[3] Tushar D. Chandra, Robert Griesemer, and Joshua Redstone. 2007. Paxos made live: an engineering perspective. In Proceedings of the twenty-sixth annual ACM symposium on Principles of distributed computing (PODC '07). ACM, New York, NY, USA, 398-407.

[4] WebSphere MQ Low Latency Messaging, http://www.ibm.com/software/integration/wmq/llm

[5] Hypervisor, http://en.wikipedia.org/wiki/Hypervisor

[6] Gong. Su, and Arun. Iyengar, A Highly Available Transaction Processing System with Non-Disruptive Failure Handling, IBM Research Report. December 2009.

[7] IBM Redbook, IBM System z10 Enterprise Class Technical Guide, http://www.redbooks.ibm.com/abstracts/sg247516.html

[8] IBM Redbook, HiperSockets Implementation Guide, March 2007.

[9] Torsten. Layda, An Order Matcher for an Electronic Stock Exchange, OOPSLA DesignFest'97.

[10] Fidelity, Trading Multi-leg Options https://scs.fidelity.com/webxpress/help/topics/learn_trading_multileg_options.shtml

[11] Ming Fan, Jan Stallaert, and Andrew B. Whinston. 1999. The design and development of a financial cybermarket with a bundle trading mechanism. *Int. J. Electron. Commerce* 4, 1 (September 1999), 5-22.

[12] Ming Fan, Jan Stallaert, and Andrew B. Whinston. 1999. A Web-Based Financial Trading System. *Computer* 32, 4 (April 1999), 64-70.

[13] J Jawad Abrache, Teodor Gabriel Crainic, Michel Gendreau, Models for bundle trading in financial markets, European Journal of Operational Research, Volume 160, Issue 1, Applications of Mathematical Programming Models, 1 January 2005, Pages 88-105

[14] Sven de Vries and Rakesh V. Vohra. 2003. Combinatorial Auctions: A Survey. *INFORMS J. on Computing* 15, 3 (July 2003), 284-309..

[15] Aleksandar Pekec and Michael H. Rothkopf. 2003. Combinatorial Auction Design. *Manage. Sci.* 49, 11 (November 2003), 1485-1503.