Collaborative and Scalable Financial Analysis with Multi-agent Technology

Jerome Yen
Department of Systems Engineering and Engineering Management
The Chinese University of Hong Kong
chyen@se.cuhk.edu.hk

Alan Chung, Heron Ho, Birgitta Tam, Rocky Lau, Michael Chua, and Kai Hwang
Department of Computer Science and Information Systems
The University of Hong Kong, Hong Kong

Abstract

During the past two years, business world started to use clusters of PCs and workstations, multi-agent technology, Java programming, Intranet, and Internet to support financial computing, decision analysis, electronic business, etc. In this paper, we integrated these technologies and present a distributed multi-agent model for network-based financial computing or economic analysis. In particular, we focus on the potential to apply cluster computing to finance, business, investment, insurance, and socioeconomic application domains. These commercial domains demand special computing characteristics, which differ in many ways from the scientific computing requirements.

A prototype of the distributed financial computing has been implemented to illustrate the idea of multi-agent computing on workstation clusters. The cluster platform and the multi-agent model has been proved to be applicable in other domains such as electronic business, market analysis, financial decision making, and economic crisis management. A three-tier multi-agent architecture has been designed to capture the structure of Hong Kong economic activities. The system calculates risks and returns based on more than thirty financial indicators that cover major business activities in Hong Kong. One more application of using such multi-agent system is to generate a composite economic index to measurement the economic performance.

1. Introduction

Information and data overflow has become a very serious problem that challenged decision makers in financial market. For example, for the users of textual information, it is impossible to trace all the sources, such as, news articles, Internet, or financial information service provider, for all the information relate to a particular incidence or issue. For numeric data, it is difficult to monitor all the time series or statistics that that may affect the economy of a region, for example, Dow Jones Index and interest rates. It has been estimated that more than 200 time series may affect the economy of the US [3]. Super computers and parallel computing have been used extensively to help experts searching for patterns or identifying relationships among these time series. For example, Wall Street had kept eyes open on IBM Deep Blue machine. Which is able to simultaneously process huge number of time series or data sets to identify their underneath relationships and this was defined to be the deep computing.

However, one of the trends of computing platform is moving from heavy investment on super computers to cluster computers. Where a group of workstations or personal computers (PCs) will be linked together to provide the processing power that similar to the super computers. To solve the data overflow problem, we integrated the Internet, multi-computer clustering, and multi-agent technology to develop a solution. In order to provide such solution and support users with user friendly interface, several issues need to be solved.

Scalability is also an important issue. To add more memory or more hard disk space is not a true solution for solving the scalability issue. For example, it over 5000 time series may affect the economy of the world, how to identify their relationships. The major difficulty lies in the fact that such relationships are time-dependent and it is difficult to use a simple regression analysis to do the job. When the number of such time series increases, the complexity of relationships among them will also increase. Such speed of increasing could be greater than that of the size of memory or the processing speed. On the other hand, the performance of PCs has been increased very rapidly and it is easy to use and maintain. This leads us to use clusters of PCs or workstations to cope with the scalability issue.

The other issue in financial computing is tractability. For example, when a system that supports risk assessment sends a warning signal to indicate that certain markets may have trouble, users must be able to understand why and such system must help users to trace the sources. To solve such problem, we proposed to use multi-agent technology. Multi-agent systems is a special type of software agents in which a group of software agents sit on different PCs or workstations to jointly accomplish a set of tasks.

The integration of cluster computers and multi-agent technology provides a new approach to

0-7695-0001-3/99 $10.00 (c) 1999 IEEE

support collaborative computing. Collaborative computing is a term that describes a set of computers of information systems that reside at different places, for example, headquarters of firms, to work together to create the joint benefits or profits. For example, supports the management of supply chains or distributed design projects. Such computing environment can be considered as a society or community. There are different types of members, called agents, each member has its set of capabilities, such as, the ability to collect information or data, to reason, to communicate and interact, and to make decision, as well as a set of tasks to finish. In order to get the joint assignment done, all or most of the members inside such environment must work together.

We have implemented a multi-agent system that consists of 40 software agents to run on the cluster at the University of Hong Kong. This system is able to simultaneously monitor or process 30 economic indexes. It takes at least ten workstations in a Unix cluster to reach a satisfactory performance. We have tested it on a cluster of workstations and servers. This cluster is constructed from 30 plus PCs, workstations, and high-performance servers, such as, SGI Power Challenger and Sun Enterprise 6000, which are connected by both fast Ethernet and ATM switches with an aggregate bandwidth of 5Gbps.

We selected economic analysis to be the problem to test the system. Analysis or study a country's economy is a very complex and difficult task. It may involve several industries or socioeconomic sectors. For example, those 30 time series that we selected to monitor cover the major business activities of five industries or sectors in Hong Kong: manufacturing, retail, financial/banking, real estate, and tourism. Our design is a three-tier architecture of agents, which reflects the structure of Hong Kong's economy.

The HKU cluster architecture and the concept of *single system image* (SSI) services will be briefly discussed in Section 2. In Section 3, we will discuss the structure of the economic activities in Hong Kong and the architecture of the multi-agent system. In Section 4, we discuss multi-agent technology and the architecture of the multi-agent system. We explain how such system uses financial indicators support to calculate risks and return to support decision making in financial investment. The software technology that used to develop the system will be discussed in Section 5.

2. Distributed Cluster Computing

A multicomputer cluster is a collection of complete computers (called nodes or hosts), which are physically connected by local area networks or high-bandwidth switches as illustrated in Fig.1. Collectively, the cluster nodes can work as an integrated computing resource, beside interactive use as individual computers [9]. Since each node runs with its own *operating system* (OS), a traditional network of computers have multiple system images on different nodes. On the other hand, a shared-memory multiprocessor server must have a *single system image* (SSI).

The computing trend is moving from clustering of high-end mainframes to clustering of desktop computers. This trend is triggered by the widespread use of PCs, workstations, and Gigabit networks. The SSI cluster gives the illusion of a single computing system. A cluster user sitting in front a workstation can have his job executed by any subset of nodes without worry about conflicts with other users. The clusters offer SSI at a wide range of abstraction levels. At one extreme, a cluster can function as a tightly-coupled multiprocessor system. At the other extreme, a cluster behaves like a distributed network system with multiple system images. In practice, single-system-image clusters are more desirable in both scientific and commercial computing applications.

To realize an SSI cluster, the following set of goals have been set by many designers [6, 7, 9]. We classify these goals in three categories: *complete transparency* in resources management, *scalable performance* by adding more cluster nodes, and *enhanced availability* in case of failure. In a cluster, it is desired to have an SSI supported by middleware between the node OS and user application environment.

The middleware consists of essentially two layers of software which glues all node OS together with a variable degree of SSI and enhanced availability. The availability infrastructure enables the cluster services of checkpointing, automatic failover, recovery from failure, and fault-tolerant operation among all cluster nodes. The SSI layer is needed to support collective cluster applications which demand operational transparency and scalable performance.

Major cluster design issues include size scalability, enhanced availability, system manageability, fast message passing, security and encryption, and distributed environments. Interested readers may check with refernces [1, 2, 6, 7, 9] for some coverage of these topics. The SSI cluster can be supported by hardare, software, or middleware. Some middleware packages for supporting cluster systems are available from public domains and some as commercial products.

The Testbed - HKU Workstation Cluster

The HKU cluster is built around four interconnected ATM switches with an aggregate bandwidth of 5 Gbps. Optical cables using 155 Mbps multimode fibers are used to interconnect 2 SMP servers and over 30 Sun and SGI workstations and PCs, all of which are located in the same building. Figure 2 shows a typical workstation cluster architecture that can be configured from the HKU cluster.

Through the Internet, an user accesses the cluster with a Web Browser. The *agent name server* (ANS) allocates and deallocate agents for a specific request. An agent is essentially a software process running on a workstation. All drafted agents must interact with the database server, which updates the Oracle database of the addressed application. The final decision result is returned to the user by the ANS through the Internet.

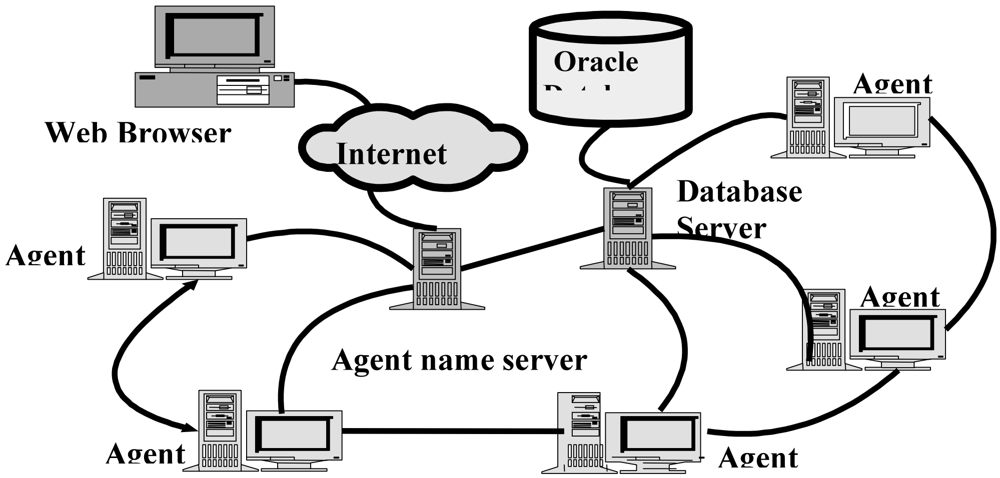

Figure 1. A Typical cluster of computers for distributed retrieval and processing of financial information from the Internet

3. Economic Analysis: The Hong Kong's Case

Fundamental analysis and technical analysis are two major approaches to analyzing the performance or future movement of a country's economy or of a particular industry [4]. Fundamental analysis is based on the assumption that market value of a company, performance of an industry, or economy of a country is strongly related to its past performance, internal and external environments, threats and opportunities, as well as uncertainties. News articles, financial statements, government reports, financial indicators, statistics, etc., are the major sources of information to support such an analysis. In addition, certain financial indicators, such as, unemployment rate, volumes of export and import, and interest rates, can be more important to help understanding the economy.

For technical analysis, charts, statistical techniques, mathematical models, and computer programs are often used to detect patterns and identify trends from indicators or indices, such as, stock prices, indexes, interest rates, foreign currency exchange rates, etc. Many believe that identification of such patterns or trends is critical to their prediction or forecasting. In most cases, interpretation or analysis with such data requires special training or knowledge.

Both leading indices and time series are useful in

determining the general direction of the economy and in forecasting the pace of economic activities. As a key element to the allocation of a country's resources and respond to the fast changing economy, most countries have set up custodians and a set of economic indicators to monitor economic performance. For example, the Bureau of Economic Analysis of the Department of Commerce has been selected by the US Government to look after the official composite leading, coincident, and lagging indexes, which are the most important barometers to measure the economic condition of the US.

Also, the Business Cycle Indicators database, which include the composite indexes, a set of "leading indicators", and a set of more than 200 economic series are also provide information to monitor the economic condition [3]. No single time series can provide a complete picture about the economic condition. In order to cancel out the volatility of individual indicators, it is necessary to analyze a group of indicators or to use composite indexes. Leading, coincident, and lagging composite indices are calculated based on the weighted sums of sets of indicators.

Since different indicators have their own volatility, more indicators can smooth out such fluctuation and provide more useful and reliable information. For example, a three-month decline in the leading index indicates a recession in the US. In Hong Kong, it needs two consecutive three-month declines in GDP to signal a recession. The leading index of the US is a very important barometer to measure its economic condition and it is determined by the normalized weighted sum of ten (10) leading

0-7695-0001-3/99 $10.00 (c) 1999 IEEE

indicators.

In the case of Hong Kong, GDP, Hang Seng Index, and interest rate are the major economic indicators. However, the GDP is only announced four times a year and it is difficult to capture the short-term fluctuation. In order to solve such problem, we had designed a composite index to show how the system works. The model that we have selected was different from the leading index in the US. It is a three-tier model to capture the relationships between economic indicators and performance factors, which are specially designed to reflect the condition of Hong Kong's economy.

To analyze Hong Kong economy, we consider 10 industrial sectors, such as finance, banking, public utilities, etc. Companies in each industry or sector are affected by one or more factors. For example, tourism industry is affected mainly by factor – Tourism, which determined by four indicators: visitor arrivals, hotel occupancy rate, resident departure, average tourist expenditure and currency exchange rates. In Table 1, we have identified ten factors that have demonstrated the most significant impacts to Hong Kong economy in recent years. The mathematical model that we have developed to build the system was a simple *Multi-factor Model*. A factor can be determined by any algebraic combination of a set of indicators. We did not choose a complicated mathematical model, because our main goal is to demonstrate how to use the cluster computers and multi-agent technology, not indulging on the delicacy of modeling. Also, economy and performance of major industries are changing rapidly, which would significant affect the composite of the models. Therefore, it is important that these models are simple and easy to adjust in different time frames.

Table 1. Economic Analysis Factors and Indicators used in Hong Kong

Analysis Factor	Social and Economic Indicators			
1. Local Economy	Domestic GDP	Unemployment Rate		
2. Trade	Import Amount	Domestic Export	Re-export Amount	
3. Interest Rates	US Prime Rate	Mortgage Rate	Monthly Deposit Rate	
4. Exchange Rates	Deutsche Mark	British Pound	Japanese Yen	
5. Foreign Stock Markets	Nikkei Index	Dow Jones Index	London Reference Index	
6. Chinese Economy	GDP of China	Unemployment Rate	National Survey	
7. Real Estate Property	Average Property Transaction Price	Business Receipts for Real Estate Industry	Property Increase or Decrease Rate	
8. Transportation	Land Transport	Maritime Transport	Air Transport	
9. Tourism	Visitor Arrivals	Hotel Occupancy Rate	Tourist Expenditures	
10. Retail and Manufacturing	Total Production	Restaurant Receipts	Wholesale	Retail

Figure 2 shows how to add or modify the contents of an economic factor On the top of the screen a little box that indicates the factor to be modified is the Economic Factor. In the middle, it shows that Factor - Local Economic is determined by three indicators: domestic GDP, unemployment rate, and regional survey. At the bottom, it provides a short explanation about the background of the factor and what indicators it was based upon. On the right-hand side of the screen is the list of all the indicators.

Users can modify the model by changing the weights on those existing indicators. Checking up the list on the right, one can modify the model simply by changing "ind_1" to "ind_13". Clicking the "Update" will change the model immediately or clicking the "Advance" will edit the configuration setting of the corresponding factor analysis agent. 4. Multi-Agent Software Technology

The most general way in which the term agent is used is to denote a hardware or (more usually) software-based system that has the following properties [8, 10, 11]:

- ◆ Autonomy: agents operate without the direct intervention of humans or others, and have some kind of control over their actions and internal state.

0-7695-0001-3/99 $10.00 (c) 1999 IEEE

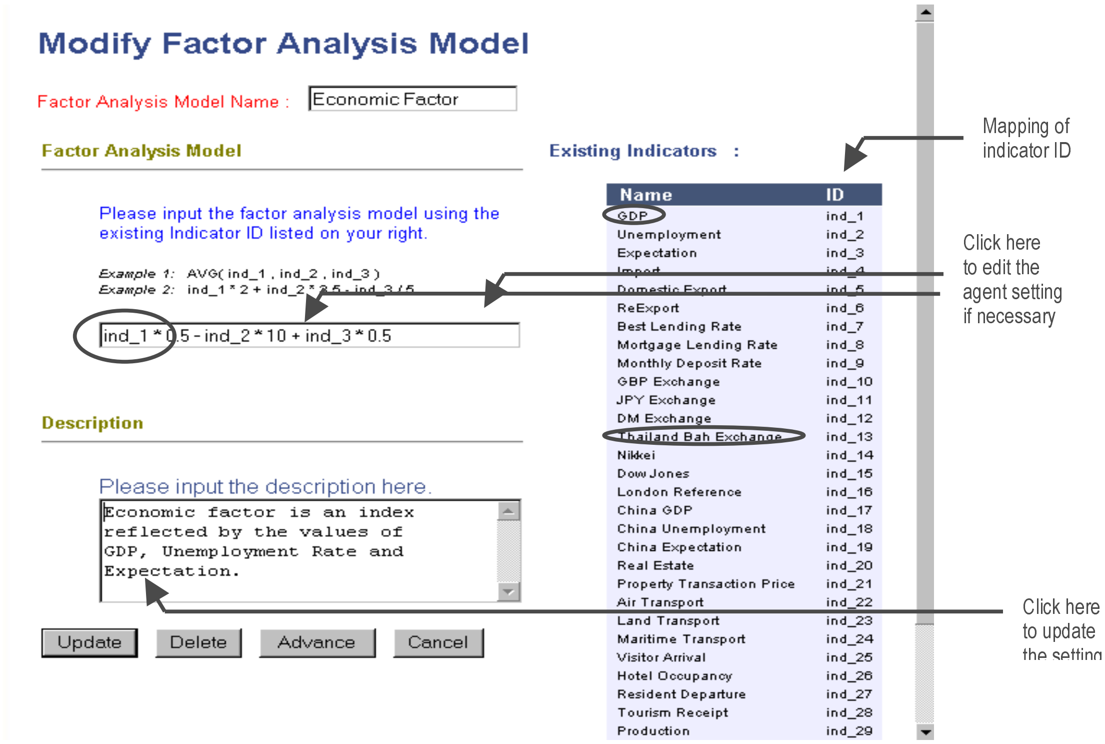

Figure 2. The model for modification of a factor analysis process

- Social ability: agents interact with other agents (and possibly humans) via some kind of agent communication language.
- Reactivity: agents perceive their environment, (which may be physical world, a user via a graphical user interface, a collection of other agents, the Internet, or perhaps all of these combined), and respond in a timely fashion to changes that occur in it.
- Pro-activeness: agents do not simply act in response to their environment, they are able to exhibit goal-directed behavior by taking the initiative.

A stronger notion of agency: For some researchers, particularly those working in artificial intelligence, the term "agent" has a stronger and more specific meaning than that sketched out above. These researchers generally mean an agent to be a computer system that, in addition to having the properties listed above, it either conceptualized or implemented using concepts that are more usually applied to human. For example, it is quite common in artificial intelligence to characterize an agent using mentalistic notions, such as knowledge, belief, intention and obligation, and such framework was the foundation to the development of several agent programming languages.

Multi-agent systems are special type of agents, which focus on the coordination and the communication among agents to collaboratively accomplish tasks. Communication and cooperation are the two most important capabilities of multi-agent systems. Multi-agent systems are designed to have the capability to either, collaborate, for example, to decompose and solve the problem, or compete, such as, search for the best deal for the users. The multi-agent system architecture is described in Fig. 4.

Communication is vitally important by which relevant information to support cooperation is exchanged. The KQML (Knowledge Query and Manipulation Language) is a language that supports the communication among agents [5]. However, agents must do more than just communication, they must have analysis and decision making capability that made possible by their knowledge bases. Design of the communication and cooperation protocols is important. However, so far there is no programs that runs at the back-end. The interaction between the multi-agent system and users is done by the front-end applications a software Agent to report what had happened to the users.

A. Mutli-Agent Processing Architecture

As discussed above, agents monitor the changes on indicators and alarm the users when there is a particular condition, such as, interest rate went up. This is achieved by the three-tier multi-agent architecture in Fig. 3. Specific agent functions are defined below:

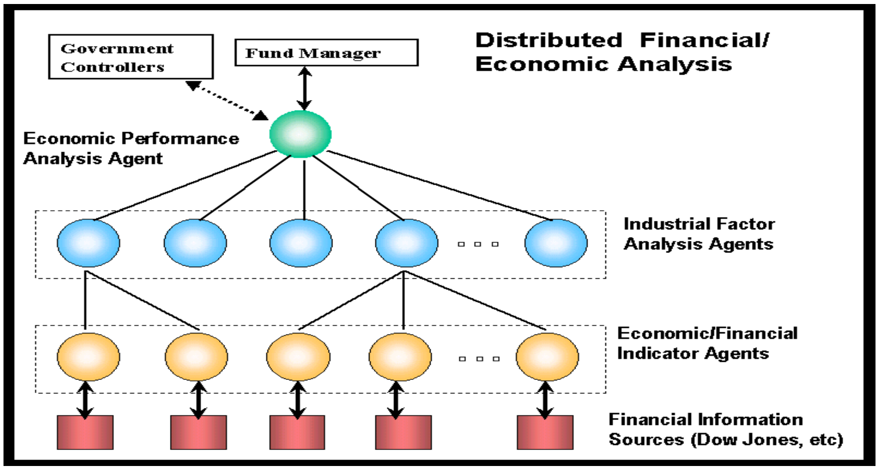

Figure 3. A multi-agent financial analysis system architecture

Economic Performance Agent - Each agent calculates and evaluates the performance of each economic index every 30 seconds. Alarm and give advice to the user, such as, a government officer, when particular situation happened.. Do daily backup on database to provide historical records for the charting function.

Factor Analysis Agent - Each factor analysis agent is responsible for a predefined factor analysis model, which can be adjusted easily by user, and monitors the corresponding set of indicators. If there is any change of those indicator values (i.e. new signal received from an Indicator Agent), then the factor analysis agent will re-do the factor analysis according to its defined factor analysis model. The Economic Performance Agent will then be informed of the new analysis result.

Indicator Agent – Each indicator agent is responsible for one predefined information source and such source can be easily modified. Once a change occurs on that indicator value, the Indicator Agent will inform its corresponding factor analysis agent of the latest value. If the latest value higher than a threshold or outside its predefined range, a warning signal will be generated also. The indicator values can be obtained from a pool of distributed information sources such as Hong Kong Government Statistics Department, News Agency, Reuters, and so on.

B. Multi-Agent Communications

The language used to support the communication among agents is Knowledge Query and Manipulation Language (KQML) [5]. KQML includes a language and protocol for exchanging information and knowledge among agents. KQML is both message format and message-handling protocol to support run-time knowledge sharing among agents. It provides the agents a common language and a set of standard message types. In the following, we will briefly discuss the process of running the system.

During the start-up of the system, all agents are connected to an Agent Name Server and undergo the registration process. For example, Indicator Agent monitors the unemployment rate and its initial value was 3.49 on May 1st, 1998. Information about host, port, and sender, which are required by KQML language to support communication among agents are also need to be initialized. After initialization of all the indicator agents, system initializes all the factor agents.

For example, Factor Analysis Agent of Local Economy initialized itself and start scanning inputs from all the 31 indicator agents. It then set up connections with three agents that provide the data to determine the Local Economy: GDP, Unemployment Rate, and Expectation (the Regional Survey). Finally, the Portfolio Management Agent sets up connection with server and initializes values of the following: portfolio information, factor information, beta values, and the expected returns.

After initialization, each Indicator Agent (IA) starts monitoring its data source. As discussed above, whenever there is a change, the Indicator Agents first analyzes how significant this change is. If such indicator exceeds the limit or threshold, for example, 4.0. (4.0) percent for unemployment rate and eleven (11) percent of interest rate, then indicator agent will send a message together with a warning message through TCP/IP by KQML language to the Factor Analysis Agent (FAA). Figure 4 shows the tasks to be accomplished by a FAA.

0-7695-0001-3/99 $10.00 (c) 1999 IEEE

```
main()
       START-UP PROCESSES
              * read the address file
              * instantiate the Factor Analysis Agent
              * create server thread
              * register the FA agent
              * connect to the Agent Name Server (ANS/router)
              * check up the list of corresponding indicators
              * start to receive messages
                     -- start()
                     -- wait for Act()
       CONTINUOUS PROCESSES
              * monthly update the corresponding factor value in database
Act(Object o)
       THIS METHOD WILL BE TRIGGERED AUTOMATICALLY WHENEVER A NEW INDICATOR VALUE IS
       FETCHED
       * do message parsing (e.g. to get the indicator value)
       * ensure that FA has acquired (i.e. newly received or already
         on memory) the indicator values from all the subordinate indicator agents
       * determine the factor value
       * read the configuration file to determine its variance
       * do factor analysis according to the model
       * update database with the latest factor value
     * inform the PM agent by KQML
         -- sendResultMessage(String receiver, String result)
         or sendErrorMessage(KQMLmessage kqml)
```

Figure 4. Pseudo Code and Tasks to be accomplished by a Factor Analysis Agent (FAA)

If the indicator does not exceed the limit, then such message only contains the value of the unemployment. Portfolio Management Agent (PMA). Figure 5 shows the pseudo codes of the tasks to be accomplished by a PMA.

```
main()
       START-UP PROCESSES
              * read the configuration and address file
              * instantiate the PM agent
              * create server thread
              * register the PM agent
              * connect to the Agent Name Server (ANS/router)
              * start initialization
                   -- instantiate objects
                   -- retrieve the portfolio ID for all existing
                        portfolios in database
                     -- initialize information of factors
                     -- initialize beta values and expected returns
              * start to receive messages
                     -- start()
                     -- wait for Act()
       CONTINUOUS PROCESSES
              * refresh the memory for every 10 min
              * monitor the risk & return for every 1/2 min
              * daily process to update risk & return in database
Act(Object o)
       THIS METHOD WILL BE TRIGGERED AUTOMATICALLY WHENEVER
       A NEW FACTOR VALUE IS FETCHED
       * do message parsing
       * determine the factor info -- value, var, mean, num
       * update DB for the new factor info -- var, mean, num
       * for each portfolio
              -- refresh the beta values on memory
              -- refresh the portfolio information on memory
              -- re-calculate the risk and return
              -- if risk or return is out of the user's limit
                     sendAlarmMessage(String receiver,String mesg)
```

Figure 5. Pseudo Code for a Portfolio Management Agent (PMA)

We use Figure 6 to illustrate how three different types of agents in such multi-agent system work together to determine the risks of economy. However it does not exceed the limit of 4.0 percent. Therefore only the new value will be forwarded to the Factor Analysis Agent of Local Economy.

After Factor Analysis Agent of Local Economy received the message from the Indicator Agent, it first checks if a warning message attached. If warning message is there, it will prepare another warning message to the Portfolio Management Agent. If not, then it performs analysis based on the model that entered by the user and the data from Indicator Agents: GDP and unemployment rate. After analysis, the Factor Analysis Agent calculates a new value of the Local Economy Factor - 100.21, and send it to the Portfolio Management Agent.

.

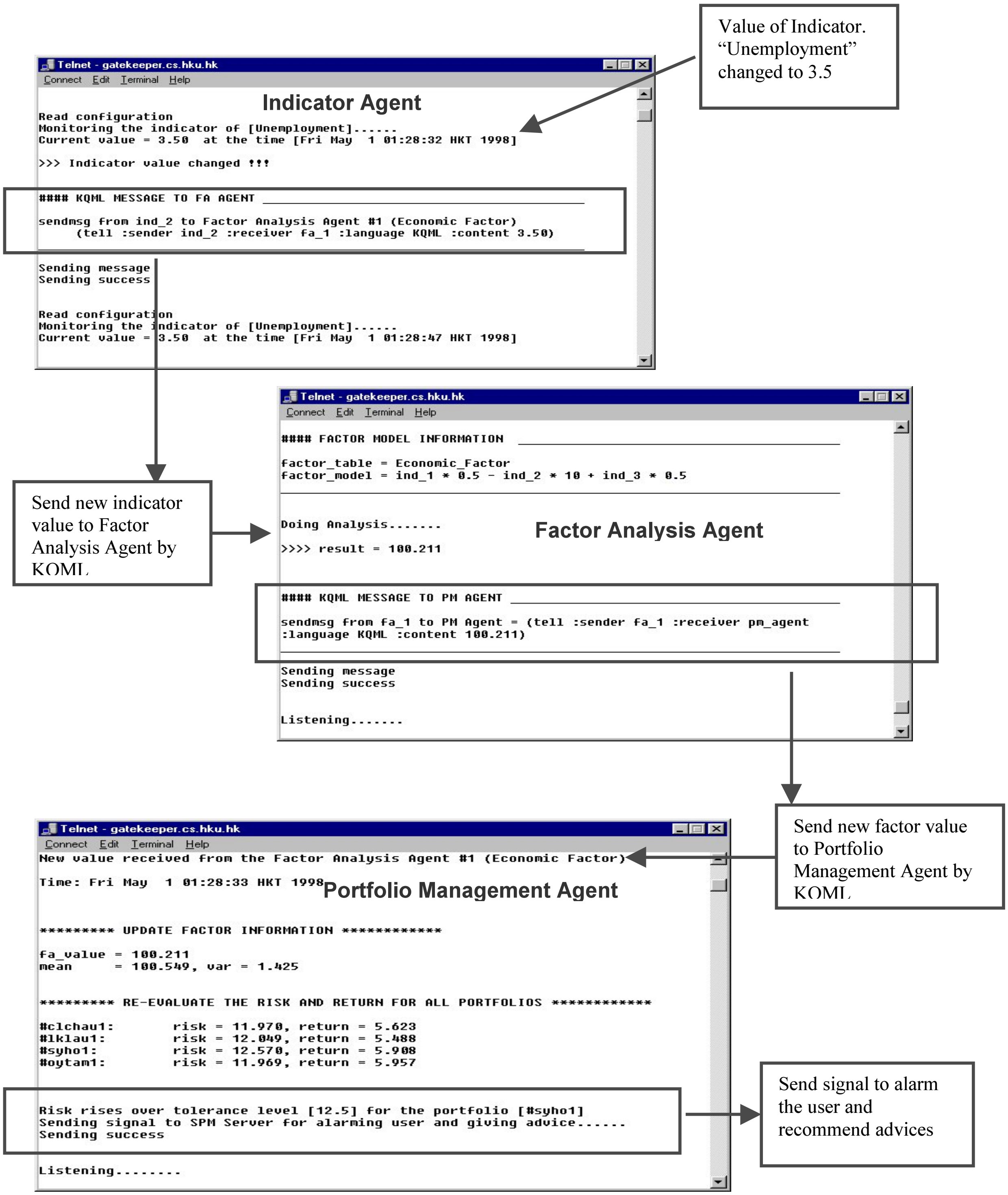

Figure 6. Interaction among different agents in distributed portfolio management

A portfolio is a basket of stocks to diversify the risks, which is similar to a leading index that consider influence of several factors. Each portfolio has its own risk and expected return and they are affected by one or more factors. As mentioned earlier, how sensitive and in what direction such influence is, are determined by the sensitivity – beta. After Portfolio Management Agent receives a message from the Factor Analysis Agent, it first checks if a warning message attached.

If not, it then calculates the new average of the Local Economy Factor. Based on the betas and the new average of Local Economy Factor (100.549), Portfolio Management Agent calculates the new risks and expected returns. Then Portfolio Management Agent has to check if any risks or returns exceed tolerances. For example, the new risk of Portfolio syho1 exceeds 12.5, which was set by the user earlier, then the agent will send a warning message to remind that an adjustment or further study on the portfolio is needed.This system has been designed to manage investors' portfolio and provide customized alarming services. It monitors the risk level and expected return of the users' portfolio continuously and gives advice to the investors on their portfolio selection. Alarming signals can be set according to the users' criteria on the risk level of portfolio or other stock data, such as, stock prices and Hang Seng Index. Besides, other stock information such as online stock quote, historical charting and stock news are also available with this system.

5. Software Tools for Cluster Processing

Table 2 summarizes the software tools been used in the development the system. For the front-end, Microsoft FrontPage has been used to develop the homepages. Visual Café was selected to build the interface. In contrast to the traditional text-editor-style programming, Visual Café enhances Java applet programming by dragging ready-made components from a library into a window GUI.

Microsoft Agent has been used to implement a client application - the Alarming Agent, which send alarm signals or recommendations to the users for solving particular condition. Microsoft Agent provides a property, that even when execution of the system has been moved to the backend or user left other tasks, it still can pop up to the front-end to alarm the user or send en e-mail to the list of people as designated by the user.

Table 2. Development Tools and Application Software

User Interface	Cluster System Software
· Microsoft FrontPage	· JDK 1.1 (Java)
· JDK 1.1 (Java)	· CGI (Oraperl, Perl)
· Visual Café	· JATLite
· Microsoft Internet Explorer	· KQML
· Microsoft Agent	· Apache HTTP server
· Adobe PhotoShop	· Oracle SQL
· VB Script	· C++
· Java Script	· UNIX Shell Scripts

All agents in the EACIS are connected to a special program called *Agent Name Server* (ANS) which provides a bulletin board and message routing services. ANS receives messages from the registered agents and routes the messages to the correct target agents. Consequently, each agent does not need to remember and maintain all the possible receivers' addresses, which may be changed frequently. The sender can ask about the receiver address to the ANS if it does not know the receiver's current address. This also allows the agent to change their addresses frequently without affecting the functioning of the system.

For the back-end, JATLite (Java Agent Template, Lite) was used as the back-end control. It is a package of Java programs that provides the basic infrastructure for developing the multi-agent systems and agent communication. It especially facilitates the construction of agents that exchange KQML messages based on open Internet standards, TCP/IP, SMTP, and FTP. It also provides greater flexibility in the back-end implementation as it supports both stand-alone agents and applet agents on various platforms such as Windows 95, Windows NT, Solaris, etc.

Knowledge Query Manipulation Language (KQML) was used as the communication language in the back-end [5]. It is a language and protocol for exchanging information and knowledge among agents, it is highly suitable for building multi-agent systems. Java Development Kit (JDK) 1.1.5 was used in Java programming. It helps to create applets that run in Java-enabled browsers and develop Java applications. It was used to program the agents and applets such as interactive charting

9

and login system of this system.

6. Conclusions

In this paper, we presented the prototype of the distributed financial computing that has been implemented to illustrate the idea of multi-agent computing on workstation clusters. The combination of the cluster platform and the multi-agent model has been proved to be useful to solve the issues of scalability and tractability, which have become very important to the financial computing. Other potential application domains of such technologies are electronic business, market analysis, financial decision making, and economic crisis management.

A three-tier multi-agent architecture has been designed to capture the structure of Hong Kong economic activities. We have conducted a series of experiments to test the usefulness and user acceptance with other systems that support economic analysis or investment analysis. One of these experiments was to let the system a generates a composite economic index, instead of returns and risks, for the decision makers to measurement the economic performance. The results indicated that the system was able to provide more information than a simple GDP (Gross Domestic Product) index and which was closer to what the residents of Hong Kong experienced.

More research in this area is needed. Computer hardware, software systems to support Single System Image, protocols and programming languages to support communication and coordination among agents, etc., are some areas that scientists can make significant contribution.

Refererences

[1] T.E. Anderson, D.E. Culler, D.A. Patterson, et al, "A Case for NOW (Networks of Workstations)", *IEEE Micro*, Feb. 1995, pp. 54-64.

[2] M.A. Baker, G.C. Fox and H.W. Yau, *"Cluster Computing Review*", Northeast Parallel Architectures Center, Syracuse University, Nov. 1995.

[3] *Business Cycle Indicators*, http://www.tcb-indicators.org/, published by The Conference Board, Inc., New York, 1997. [9] Essinger, E. The Investment Manager's Handbook. Chapman and Hall, London, 1993. .

[4] E. Essinger, *The Investment Manager's Handbook*. Chapman and Hall, London, 1993.

[5] T. Finin, R. Fritzson, D. McKay, and R. McEntire, "KQML as an Agent Communication Language", *in Proceedings of the Third International Conference on Information and Knowledge Management* CIKM'94, ACM Press, November 1994.

[6]. K. Hwang and Z. Xu, *Scalable Parallel Computing: Technology, Architectuure, Programming,* McGraw-Hill Book Company, San Francisco, 1998

[7]. K. Hwang, Z. Xu, and C. L. Wang, "Viable Approaches to Realizing Single System Image in Mlticompuuter Clusters", *IEEE Concurrency*, submitted Dec. 10, 1997.

[8] P. Maes, "Agents that Reduce Work and Information Overload", *Communications of ACM*, Vol. 37, No. 7, pp. 30-40, July 1994.

[9]. G.F. Pfister, *In Search of Clusters*, (second Edition), Prentice-Hall PTR, New Jersey, 1998

[10] M. M. Waldrop, "Software Agents Prepare to Sift the Riches of Cyberspace", *Science*, Vol. 265, pp. 882-883, August 1994.

[11] M. J. Woodridge and N. R. Jennings, "Agent Theories, Architectures, and Languages: A Survey". In M. J. Woodridge and N. R. Jennings, editors, *Intelligent Agents.* Springer-Verlag, Berlin, 1995.

[12] S. Zhou, "LSF: Load Sharing and Batch Queueing Software", Platform Computing Corporation, North York, Canada, 1996.

0-7695-0001-3/99 $10.00 (c) 1999 IEEE

10