Financial Business Cloud for High-Frequency Trading

A Research on Financial Trading Operations with Cloud Computing

Arden Agopyan
Software Group
IBM Central & Eastern Europe
Istanbul, TURKEY
arden@tr.ibm.com

Emrah Şener
Center for Computational Finance
Özyeğin University
Istanbul, TURKEY
emrah.sener@ozyegin.edu.tr

Ali Beklen
Software Group
IBM Turkey
Istanbul, TURKEY
alibek@tr.ibm.com

Abstract — **This paper defines a new business cloud model to create an efficient high-frequency trading platform while validating the portability and also cost-efficiency of cloud execution environments for financial operations. High-frequency trading systems, built to analyze trends in tick-by-tick financial data and thus to inform buying and selling decisions, imply speed and computing power. They also require high availability and scalability of back-end systems which, require high cost investments. The defined model uses cloud computing architecture to fulfill these requirements, boosting availability and scalability while reducing costs and raising profitability. It incorporates data collection, analytics, trading, and risk management modules in the same cloud, all of which, are the main components of a high-frequency trading platform.**

Keywords — high-frequency trading, cloud computing, portability, cost-efficiency, financial business cloud.

I. INTRODUCTION

Financial markets are broad and complex systems in which, market players interact with each other to determine the prices of different assets.

Advances and innovations in computer technologies have changed the nature of trading in financial markets [1]. As a result of these innovations, transmission and execution of orders are now faster than ever, while the holding periods required for investments are compressed. For this reason a new investment discipline, high-frequency trading, was born [2].

In very broad terms, high-frequency trading refers to analyzing trends in tick-by-tick data and basing buying and selling decisions on it.

Exchanges supporting high-speed low-latency information exchange have facilitated the emergence of high-frequency trading in the markets. In 2009, in the United States, high-frequency equity trading was 61% of equity share volume and generated $8 billion per year (Figure 1) [3]. Again in the United States, high-frequency trading also accounted for up to 40% of trading volume in futures, up to 20% in options, and 10% in foreign exchange [5]. It has already become popular in Europe and is also manifesting itself in some emerging markets, like Latin America and Brazil [5]. It is estimated that about 30% of Japanese equity trading is high-frequency [5]. This compares with up to 10% in all of Asia, up to 10% in Brazil, about 20% in Canada, and up to 40% in Europe [5].

Hong Kong Stock Exchanges is building a data centre where traders can place their computers next to Hong Kong Exchanges' own systems [3]. The National Stock Exchange of India has rented out racks of computer space for traders, and the Australian Securities Exchange plans a centre offering co-location by August 2011 [3]. The speed with which, exchanges are building such facilities is a sign of the global spread of the High-Frequency Trading phenomenon [3].

High-frequency trading platforms incorporate trading, data collection, analytics, and run-time risk management modules to create systems which, search for signals in markets, such as price changes and movements in rates. This helps to spot trends before other investors can blink. Then finally orders and strategies are executed or changed within milliseconds on the exchanges. The trading module hosts trading algorithms built on top of the statistical models, and executes orders on electronic execution platforms like exchanges. The data collection module collects tick-by-tick data from data providers and feeds trading and analytics modules. This data can also be exported to external data analysis tools. The analytics module is used to analyze historical financial data, to generate automated reports and to help creating new trading algorithms.

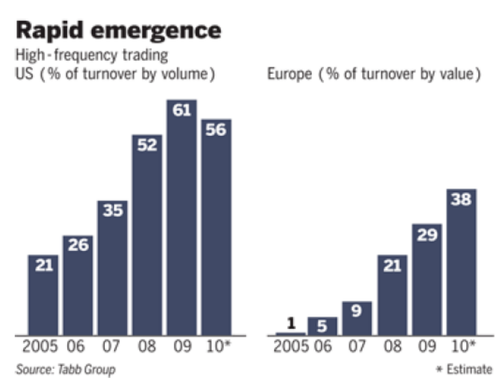

Figure 1. High-frequency trading in the United States and Europe [3].

Finally, the run-time risk management module is responsible for maintaining the whole system within pre-specified behavioral and profit and loss boundaries. These modules can be accessed via web and rich mobile applications which, enhance management capabilities and increase the speed of user interactivity and control.

High-frequency trading systems imply speed, as high-frequency trades are done in milliseconds, and also require high availability and readiness to trade at anytime. The speed of execution is secured by powerful hardware and co-location of the systems with the electronic execution platforms to minimize the network latency [2][6]. High availability is achieved by adding more resources to the system and by clustering the datacenters. All of these necessitate high cost investments.

Cloud computing refers to both the applications delivered as services with Software as a Service (SaaS) model over the Internet, and the hardware and systems software in the datacenters that provide those services [7]. A cloud is the ensemble of applications delivered as services and datacenter hardware, software and networking.

From the cloud user and consumer perspective, in the cloud, computing resources are available on demand from anywhere via the Internet and are capable of scaling up or down with near instant availability. This eliminates the need for forward planning forecasts for new resources [8]. Users can pay for use of computing resources as needed (e.g., processors by the hour and storage by the day) and release them as needed, thereby rewarding conservation by letting machines and storage go when they are no longer useful [7]. The cost impact of over-provisioning and under-provisioning is eliminated [8] and consumers no longer need to invest heavily or encounter difficulties in building and maintaining complex IT infrastructures [9]. Cost elements like power, cooling, and datacenter hardware and software are eliminated, as well as labor and operations costs associated with these. Using computing as a utility [8] with infinite and near instant availability and low entry costs gives enterprises the opportunity to concentrate on business rather than IT in order to enter and exploit new markets. There is also no cost for unexpectedly scaling down (disposing of temporarily underutilized equipment), for example due to a business slowdown [7]. In our world, where estimates of server utilization in datacenters range from 5% to 20% [7], elastic provisioning to scale up and down to actual demand creates a new way for enterprises to scale their IT to enable business to expand [8].

In cloud computing, business process as a service is a new model for sharing best practices and business processes among cloud clients and partners in the value chain [10]. A business cloud covers all scenarios of business process as a service in the cloud computing environment [10].

This paper presents a financial business cloud model for high-frequency trading to create an efficient trading platform and IT infrastructure using cloud computing architecture for financial institutions. In this model, trading, data collection, analytics, and run-time risk management modules are deployed to the cloud. An Enterprise Service Bus, a standard-based integration platform [11], integrates these modules and handles routing, data transformations, mediations and messaging between them. Cloud Manager is responsible for essential tasks like policy management, account management, authorization & access, security, application management, scheduling, routing, monitoring, auditing, billing and metering [10]. It exposes modules as high availability financial cloud services accessible from anywhere in the world via the Internet. The whole cloud is co-located in datacenters close to the electronic execution platforms to avoid data movement costs and network latency, and to assure the speed of execution [6][7].

Cloud computing is a unique opportunity for batch-processing and analytics jobs which, analyze terabytes of data and take hours to finish, as well as automated tasks responsible for responding as quickly as possible to real-time information [7]. As these are essential jobs in high-frequency trading operations, and require high computing power, high-frequency trading platforms are ideal candidates for cloud computing.

Total cost of ownership can be reduced by using high-frequency trading platforms as financial business clouds instead of deploying capital intensive on-premise infrastructure. Adopting this model reduces the IT dependence of high-frequency trading while increasing profitability. Existing systems can be designed to exist in a cloud, as portability can be achieved while moving to cloud environments [12].

Cloud computing gives financial institutions the opportunity to outsource their IT infrastructure and operations, and to concentrate on business rather than IT. It also helps to reduce their operational risk and risk management costs because, availability and service delivery are assured by cloud providers via Service Level Agreements (SLAs) [9]. Cloud computing has a big future for high-frequency trading clients, and can be used increasingly to allow firms to implement strategies that previously might have been considered too short-term to justify implementation [13].

Section 2 of this paper, presents work related to this subject. Section 3 discusses why high-frequency trading requires the adoption of cloud computing as Information Technology (IT) infrastructure. This section also includes the reference component architecture of a contemporary on-premise high-frequency trading platform. Section 4 reveals the proposed model with a research which, helped to determine the requirements of the model and also its feasibility and portability. Section 5 presents the conclusion and future work.

II. RELATED WORK

There are many published studies to assist in understanding high-frequency trading and cloud computing individually. Irene Aldridge published a book exploring various aspects of high-frequency trading [2], and references [7] [8] [9] [10] are valuable studies on cloud computing. Regarding financial cloud applications, V. Chang, G. Wills and D. De Roure proposed the Financial Cloud Framework [12]. This study demonstrates how portability, speed, accuracy and reliability can be achieved while moving financial modeling from desktop to cloud environments.

This study proposes a financial business cloud model and addresses high-frequency trading. It proposes cloud reference architecture for efficient high-frequency operations.

III. High-Frequency Trading and Cloud Computing

This section examines why high-frequency trading requires the adoption of cloud computing as IT infrastructure. The reference component architecture of a contemporary on-premise high-frequency trading platform is also presented.

A. *High-Frequency Trading*

In time, masters of physics and statistics, quants, gave birth to quantitative trading. This is a new trading style using innovative and advanced mathematical trading models which, make portfolio allocation decisions based on scientific principles. The objective of high-frequency trading is to run the quant model (the model developed after quantitative analysis) faster, and to capture the gain from the market, as high-frequency generation of orders leaves very little time for traders to make subjective non-quantitative decisions and input them into the system.

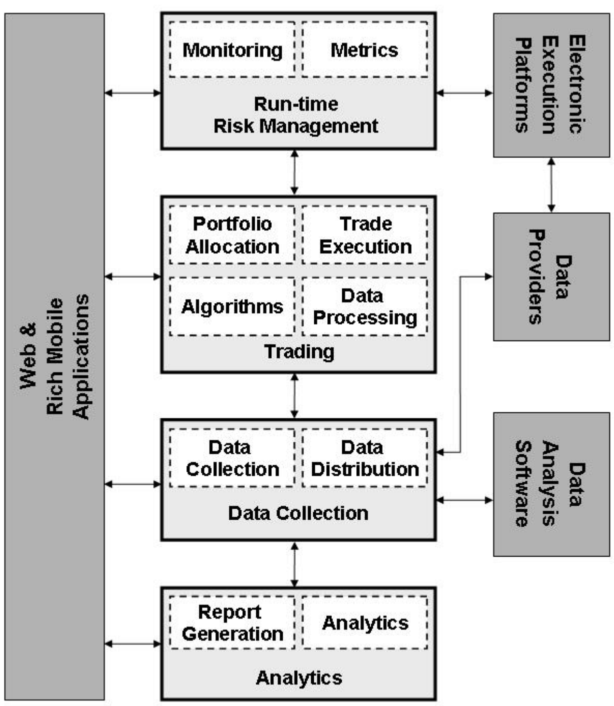

Figure 2. Reference component architecture of a contemporary on-premise high-frequency trading platform [1].

Many high-frequency traders collect tiny gains, often measured in pennies, on short-term market gyrations [14]. They look for temporary "inefficiencies" in the market and trade in ways that can make them money before the brief distortions go away [14].

The need for speed, to make and execute trading decisions and strategies, requires investment in fast computers. These strategies are established by designing algorithms including generation of high-frequency trading signals and optimization of trading execution decisions. The need to be ready to trade at anytime requires high availability of the trading and execution systems. This high availability is assured by adding more resources to the system and by clustering the datacenters. With all of these aspects, high-frequency trading operations are IT dependent.

This IT dependence of high-frequency trading generates two drawbacks from a cost perspective:

- Profitability: Trading itself already entails a transaction cost, and high-frequency trading generates a large number of transactions, leading to exorbitant trading costs. As high-frequency traders look for tiny gains, the combination of trading and IT infrastructure costs reduces profitability.
- Lead time to deploy trading algorithms and strategies: Implementing high-frequency trading platforms to deploy algorithms and strategies created by quants and traders requires experienced IT labor and this adds another layer to the operation, costing time and money.

B. *Contemporary High-Frequency Trading Platforms*

Contemporary high-frequency trading platforms incorporate trading, data collection, analytics, and run-time risk management modules. They may also be accessed via web and rich mobile applications to provide user control and enhanced management capabilities.

Figure 2 shows the reference component architecture of a contemporary on-premise high-frequency trading platform [1]. In this architecture:

- The trading module incorporates optimal execution algorithms to achieve the best execution within a given time interval, and the sizing of orders into optimal lots while scanning multiple public and private marketplaces simultaneously. These algorithms are generally academic researches and proprietary extensions which, are coded and embedded into the software. This module accepts and processes data from data providers via the data collection module and real-time data coming from exchanges. It generates portfolio allocation and trade signals, and records profit and loss while automating trading operations.
- The data collection module is responsible for collecting real-time and historical financial data coming from data providers. High-frequency financial data are observations on financial

variables taken daily, or on a finer time scale, and this time stamped transaction-by-transaction data is called tick-by-tick data [13]. Data providers (or aggregators) are companies who generally provide 24-hour financial news and information including this high-frequency real-time and historical price data, financial data, trading news and analyst coverage, as well as general news. Collected tick-by-tick data and financial news in machine readable format are distributed to trading and analytics modules to feed trading algorithms, to support decision making processes, and to generate reports. This data can be exported to external data analysis software to be used in algorithmic research.

- The analytics module is responsible for automated report generation from historical financial data as well as providing multi-dimensional analytics.
- The run-time risk management module ensures that the system stays within pre-specified behavioral and profit and loss bounds using pre-defined metrics. Such applications may also be known as system-monitoring and fault-tolerance software [2].
- The electronic execution platform is the exchange or market facilitating electronic trading (preferably in high-speed and low-latency) which, is a must for high-frequency trading operations. Platform independent high-frequency systems can connect to multiple electronic execution platforms. Intermediary languages like Financial Information eXchange (FIX), a special sequence of codes optimized for the exchange of financial trading data, helps organizations to change the trading routing from one executing platform to another, or to several platforms simultaneously [15].
- Web and rich mobile applications are channels developed to enhance management capabilities, and increase the speed of user interactivity and control. They may also incorporate modules under the same interface to create a single point of control.

Modules can be developed in-house, or alternatively proprietary software sold by major software vendors can be used. Modules are deployed on-premise following high investments in expensive datacenters including hardware, software and network connectivity [7]. Generally, each module is deployed on-premise to separate hardware with very low or no virtualization. They interact with each other independently with different communication protocols and data types. Development, deployment, operation and maintenance of these systems require experienced IT labor which, is expensive and drives costs upwards.

C. Cloud Computing as Infrastructure for High-Frequency Trading

The adoption of cloud computing as infrastructure for high-frequency trading addresses the IT dependency of high-frequency trading platforms as follows:

- Investing in building and maintaining complex IT infrastructure is no longer necessary. Computing resources are billed on a usage basis.
- Computing resources are infinitely available on demand from anywhere via the Internet.
- The cloud provider is responsible for maintaining and operating the IT infrastructure.

Most of the tasks in high-frequency trading operations are automated based on algorithms. The whole system is responsible for responding as quickly as possible to real-time information coming from markets. Cloud computing provides the availability, speed and computing power required for these automated operations.

High-frequency trading operations include batch-processing and analytics jobs requiring high computing power. Cloud computing provides a unique opportunity in this regard [7].

Total cost of ownership can be reduced by adopting cloud computing as a high-frequency trading infrastructure instead of deploying capital-intensive on-premise infrastructures. Buyers can move from a capital expenditure (CAPEX) model to an operational expenditure (OPEX) one by purchasing the use of the service, rather than having to own and manage the assets of that service [6]. Adopting this model reduces the IT dependency of high-frequency trading while increasing profitability.

Nowadays, trading firms and hedge funds are already outsourcing their accounting and back-office operations. Cloud computing gives financial institutions the opportunity to outsource their IT infrastructure and operations, and concentrate on business rather than IT. It also helps to reduce their operational risk and risk management costs because availability and service delivery are assured by cloud providers via SLAs [9]. As high-frequency trading operations are already running in many countries, this model will facilitate the entry of other participants to the market at a low entry cost. Cloud computing has a big future for high-frequency trading clients and can be used increasingly to allow firms to implement strategies that previously might have been considered too short-term to justify implementation [13].

IV. Financial Business Cloud for High-Frequency Trading (FBC-HFT)

This section presents the Financial Business Cloud for High-Frequency Trading (FBC-HFT) to create an efficient trading platform and IT infrastructure for financial institutions using cloud computing architecture. A research which, helped to determine the requirements of FBC-HFT and to validate its feasibility is also exposed in this section.

A. A Research to Determine Requirements and to Validate the Feasibility of FBC-HFT

Implementation of a high-frequency trading platform consists of many components such as identified statistical models, coded algorithms using these models to analyze and clean the tick data, installations of hardware and software, and connections with exchanges and data providers as well as whole risk management structure of the platform. The objective of this research is to analyze historical high-frequency data using statistical models and algorithms to determine the main requirements of FBC-HFT for the data analysis phase, the most critical part of the operation affecting the autonomous decision making process. Our aim is also to execute these operations in a cloud environment as well as in on-premise hardware to confirm the feasibility of the model while simulating the autonomous decision making process. Implementation of other components required to build a complete high-frequency trading platform is subject to future work.

1) Data

High-frequency tick data is different from low-frequency data with its own properties. Utilization of tick data creates opportunities which, are not available at lower frequencies.

High-frequency Istanbul Stock Exchange 30 Index (XU030) tick data for 10-minute intervals between April 1st 2007 and June 30th 2010 are used for this application.

The Istanbul Stock Exchange (ISE) was established for the purpose of ensuring that securities are traded in a secure and stable environment, and commenced operating in January 1986 [16]. The ISE has contributed greatly to the development of Turkish capital markets and the Turkish economy since the date of its establishment [16]. The ISE 30 Index consists of 30 stocks which, are selected among the stocks of companies listed on the National Market, and the stocks of real estate investment trusts and venture capital investment trusts listed on the Corporate Products Market [16].

Data was obtained directly from the Turkish Derivatives Exchange (TURKDEX) as part of an exclusive research agreement between TURKDEX and Özyeğin University – Center for Computational Finance (CCF). TURKDEX, the first private exchange in Turkey, designs and develops markets where derivative contracts of assets, liabilities and indicators are traded in a competitive and secure environment [17].

High-frequency data may contain erroneous observations, data gaps and even disordered sequences [18]. These may result from human input errors, such as typing errors leading to data outliers; computer system errors, such as transmission failures leading to data gaps, and database bugs leading to mis-ordered time series observations [18]. Data problems may bring about misleading results from the analysis. To obtain a clean data set, we identified and discarded the records which, are not of interest using available information. The raw data is re-ordered, filtered and cleaned for mis-ordered ticks, repeated ticks and erroneous test ticks using Microsoft Excel functions. Business week and exchange operating hour restrictions are also applied to the data as the analysis is region specific.

The time series created from these data sets have following fields:

- A timestamp (ex: 01.04.2007 09:00:09)
- Last index value (ex: 48546.63)
- A financial identification code (ex: XU030)

2) Statistical Models and Analytical Methods

Experiments include the following analyses and calculations for ten-minute tick data:

a) Basic descriptive statistics:

- Mean: The weighted average of all possible values that a random variable can take on, or simply the expected value ($E(X)$). The larger the sample size, the more reliable is the mean [19]. For a data set, the mean ($\overline{X}$) is the sum of the values divided by the number of values:

$$\overline{X} = \frac{x_1 + x_2 + x_3 + \ldots + x_n}{n} \tag{1}$$

- Variance: A measure of how far numbers of a set are spread out from each other. It also describes how far the numbers lie from the mean (expected value). The variance is the expected value of the squared difference between the variable's value and the variable's mean:

$$Var(X) = E\left[(X - \overline{X})^2\right] \tag{2}$$

- Standard deviation: Shows how much variation there is from the mean. A low standard deviation indicates that the data points tend to be very close to the mean. A high standard deviation indicates that the data are spread out over a large range of values. Standard deviation is the square-root of the variance:

$$\sigma = \sqrt{E\left[(X - \overline{X})^2\right]} \tag{3}$$

- Skewness and kurtosis: Practitioners use skewness and kurtosis of returns when describing the shape of the distributions. Skewness measures the deviation of the distribution from symmetry. If the skewness is clearly different from 0, then that distribution is asymmetrical, while normal distributions are perfectly symmetrical [19]. Skewness illustrates the position of the distribution relative to the return average; positive skewness indicates prevalence of positive returns, while negative skewness indicates that a large proportion of returns are negative [2]. Skewness is calculated as follows [20]:

$$Skew = \frac{E(X - \overline{X})^3}{\sigma^3} \tag{4}$$

Kurtosis measures the "peakedness" of a distribution [18]. Distributions with values of less than 3 are called platykurtic, and those with values greater than 3 are called leptokurtic [20]. A distribution with a kurtosis value of 3 is known as mesokurtic, of which, the normal distribution is the prime example [20]. Kurtosis indicates whether the tails of the distribution are normal; high kurtosis signifies "fat tails," a higher than normal probability of extreme positive or negative events [2]. Kurtosis is calculated as follows [20]:

$$Kurt = \frac{E(X-\overline{X})^4}{\left[E(X-\overline{X})^2\right]^2} \tag{5}$$

Extreme negative returns can be particularly damaging to a trading strategy, potentially wiping out all previous profits and even equity capital [2].

b) Technical analysis:

- Z-Score: A statistical measure that quantifies the distance a data point is from the mean of a data set. This distance is measured in standard deviations. Z-Score is also called z-value, normal score, standard score and standardized variable. Z-Score is calculated as follows:

$$z = \frac{X-\overline{X}}{\sigma} \tag{6}$$

- Moving Average Convergence/Divergence (MACD): MACD is a technical momentum indicator that belongs to a family of indicators called oscillators. An oscillator gets its name from the fact that it moves or oscillates between two fixed values based on the price movement of a security or index. Here, taking the difference between two exponential moving averages (EMAs) with different periods, MACD produces an oscillator because the resulting curve swings back and forth across a zero line [21]. The MACD is calculated by subtracting the 26-day EMA from the 12-day EMA. A 9-day EMA of the MACD, called the "signal line" or "trigger line", is then plotted on top of the MACD, functioning as a trigger for buy and sell signals [22].

The MACD Indicator mathematical formulae are as follows [22]:

$$MACD_{[0]} = (C_{[0]}.\%_{MACD}) + (MACD_{[1]}.(1-\%_{MACD})) \tag{7}$$

$$\%_{MACD} = \left(\frac{2}{Interval+1}\right) \tag{8}$$

where:

$C_{[0]}$= Closing price of the most recent period.

$\%_{MACD}$ = Percentage used to determine the exponential moving average length.

$MACD_{[1]}$= The MACD value from one period previous.

$Interval$ = Exponential moving average length in periods.

There are three popular ways to interpret the MACD indicator:

- Crossovers: The basic trading rule is to sell when the indicator falls below the trigger line [22]. Similarly, a buy signal occurs when it rises above the trigger line [22].
- Overbought / Oversold: When the shorter moving average pulls away dramatically from the longer moving average (i.e., it rises), it is likely that the price is overextending and will soon return to more realistic levels [22].
- Divergence: A bearish divergence occurs when it is making new lows while prices fail to reach new lows [22]. A bullish divergence occurs when it is making new highs while prices fail to reach new highs [22].

Relative Strength Index (RSI): Another technical momentum indicator that belongs to the oscillators' family. An RSI ranges between 0 and 100 and compares the magnitude of recent gains to recent losses in an attempt to determine overbought and oversold conditions of an asset [23]. RSI is calculated using the following formula:

$$RSI = 100 - \frac{100}{1+RS} \tag{9}$$

where:

$$RS = \frac{\text{Average of days' up closes}}{\text{Average of days' down closes}} \tag{10}$$

When the RSI turns up, developing a trough below 30, it suggests the price is oversold and likely to rally [23]. Conversely, when the RSI turns down, reaching a peak above 70, it suggests that the price is overbought and likely to drop [23].

3) Development and Execution Environments

Comparative analysis methodologies are used for this research. Basic descriptive statistics and technical analysis methods are developed as executable algorithms. These algorithms are deployed and executed on an on-premise hardware system and on a cloud system to determine the feasibility and effectiveness of cloud versus on-premise deployments.

a) Development:

Basic descriptive statistics and technical analysis methods are developed as executable algorithms in The R Project for Statistical Computing. R is a language and environment for statistical computing and graphics [24]. Similar to the S language and environment which, was developed at Bell Laboratories (formerly AT&T, now Lucent Technologies) by John Chambers and colleagues, R provides a wide variety of statistical (linear and nonlinear modeling, classical statistical tests, time-series analysis, classification, clustering etc.) and graphical techniques, and is highly extensible [24]. R is an integrated suite of software facilities for data manipulation, calculation and graphical display, available as Free Software under the terms of the Free Software Foundation's GNU General Public License in source code form [24]. It includes:

- An effective data handling and storage facility.
- A suite of operators for calculations on arrays, in particular matrices.
- A large, coherent, integrated collection of intermediate tools for data analysis.
- Graphical facilities for data analysis and display, either on-screen or on hardcopy.
- A well-developed, simple and effective programming language which, includes conditional, loops, user-defined recursive functions and input and output facilities.

In order to implement the desired high frequency trading algorithms, the Trade Analytics project under the R-Forge and Moments package is leveraged.

The Trade Analytics project is a transaction-oriented infrastructure for defining instruments, transactions, portfolios and accounts for trading systems and simulation. It intends to provide portfolio support for multi-asset class and multi-currency portfolios [25]. The Trade Analytics project consists of four contributory packages [26]:

- Blotter: Tools for transaction-oriented trading systems development.
- Financial Instrument: Infrastructure for defining instruments` meta-data and relationships.
- Quantstrat: Specifies build, and back-test quantitative financial trading and portfolio strategies.
- RTAQ: Contains a collection of R functions to carefully clean and match the trades and quotes data, calculate ex post liquidity and volatility measures and detect price jumps in the data.

Last index values are not meaningful in isolation while calculating the Basic Descriptive Statistics, so calculation of delta (Δ) value change series from the data sets is required. Changes are expressed as percentages and are calculated using the following formula:

$$\Delta = \ln\left(\frac{P_t}{P_{t-1}}\right) \tag{11}$$

Tables 3, 4, 5 and 6 show R execution codes for Basic Descriptive Statistics, Z-Score, MACD and RSI respectively. Figures 4 and 5 show MACD and RSI graphs of XU030 for two months period respectively.

b) Execution Environments

Developed algorithms are deployed on an on-premise hardware system and on a cloud system.

The on-premise hardware is provided by IBM Istanbul Innovation Center (IIC). IBM IIC offers ISVs, business partners and customers, trainings on cloud technologies, expertise of local subject matter experts, and leverages IIC capabilities for local business [27]. IBM IIC aims to help IBM customers & partners provision their VMs in a cloud environment or port their applications (SaaS) and databases as a service (DBaaS) in a cloud environment [27].

Amazon Elastic Compute Cloud (EC2) and Biocep-R Project are used as the cloud system. Amazon EC2 is a web service that provides resizable computing capacity in the cloud [28]. Amazon EC2 presents a true virtual computing environment, allowing you to use web service interfaces to launch instances with a variety of operating systems, load them with your custom application environment, manage your network's access permissions, and run your image using as many or as few systems as you desire [28]. In Amazon EC2, customers pay only for the resources that they actually consume, like instance-hours or data transfer. Biocep, a universal open-source computing platform that enhances the accessibility of mathematical and statistical computing, creates an open environment for the production, sharing and reuse of all the artifacts of computing [29]. With Biocep, R/Scilab computational engines are abstracted with URLs and can run at any location [29]. They can be interactively controlled from the user's laptop either programmatically, or via an extensible, highly productive data analysis workbench, or from highly programmable spreadsheets [29]. The Biocep-R software platform makes it possible to use mainstream statistical/scientific computing environments such as R, Scilab, SciPy, Sage and Root as a service in the cloud [29]. The full capabilities of the environments are exposed to the end user from within a simple browser. Users can issue commands, install and use new packages, generate and interact with graphics, upload and process files, download results, etc. using high-capacity virtual machines that can be started and stopped on-demand. The computational engines can be used as clusters on Grids and Clouds to solve computationally intensive problems, to build scalable analytical web applications, or to expose functions as web services or nodes for workflow workbenches [29]. Biocep-R virtual machine is available as Amazon Machine Image (AMI). An AMI is a special type of pre-configured operating system and virtual application software which, is used to create a virtual machine within the Amazon Elastic Compute Cloud (EC2) [30]. It serves as the basic unit of deployment for services delivered using EC2 [30]. Biocep-R AMI can be controlled via Elasticfox, Mozilla Firefox

TABLE I. TEST ENVIRONMENTS AND CONFIGURATIONS

	IBM IIC [31]	Amazon EC2 [28]
Type	On-premise	Virtual / Cloud
Model	IBM BladeCenter LS21	Standard Instances – Large
Memory	7.5 GB	7.5 GB
Computing Units	AMD Opteron 2.6 GHz - 2 cores	4 EC2 Compute Units (2 virtual cores- 2 EC2 Compute Units each)
Architecture	64-bit	64-bit
Operating System	Ubuntu Linux – Server	Ubuntu Linux – Server

TABLE II. COST ANALYSIS

	IBM IIC	Amazon EC2
Initial Price	$11400[a]	$0
Hourly Operating Cost	Unestimated	$0.34
Execution Time (hours)	2400	2400
Total Cost of Ownership (TCO)	**> $12400**	**$816**

a. Price quoted by IBM Turkey.
b. 2 years equal 2400 execution hours based on exchange operating hours.

TABLE III. EXECUTION RESULTS

	IBM IIC	Amazon EC2
	Execution Time (s)	**Execution Time (s)**
Basic Descriptive Statistics	1.351	1.267
Z-Score	0.040	0.036
MACD	7.323	6.953
RSI	837.365	807.678

extension for interacting with Amazon EC2. [29] Figure 6 shows the technology environment of Biocep-R.

Near-identical computer configurations are used for tests. Used IBM IIC hardware and Amazon EC2 virtual environment configurations are shown in Table 1.

1) *Cost Analysis*

Table 2 shows the Total Cost of Ownership (TCO) of on-premise and cloud systems for 2 years.

Considering the execution hours based on exchange operating hours, and leveraging cloud's pay-as-you-go model, we concluded that cloud usage reduces the Total Cost of Ownership (TCO) compared to the usage of on-premise hardware systems, while not sacrificing execution time and performance.

2) *Execution Results*

Table 3 shows the execution times of the algorithms.

This research presents an essential part of high-frequency trading operations which, includes:

- Cleansing of the data coming from data providers.
- Conversion and manipulation of the data for different analysis software and tools.
- Routing the data to the tools.
- Analyzing the data for high-frequency characteristics
- Execution of financial algorithms and calculations.
- Decision-making based on the analysis.

Regarding this research, the following outputs are observed:

- Portability can be achieved while moving financial calculation environments from on-premise to cloud environments.
- Development of data conversions and transformations is time consuming and hampers the implementation.
- There is a need for integration between different tools and systems.
- Analyzing high-frequency data is computing power intensive.
- Usage of cloud systems reduces the TCO.

These outputs show that the adoption of cloud computing can address the computing power need. An Enterprise Service Bus (ESB), a standards-based integration platform combining messaging, web services, data transformation and intelligent routing in a highly distributed, event driven Service Oriented Architecture [11] can facilitate the development of data transformation and the integration of different systems.

B. *The Model*

The proposed reference model in this research incorporates high-frequency trading modules in short running; routing, data and protocol conversion based processes and reveals them as a business cloud.

Figure 3 shows the reference component architecture of the proposed Financial Business Cloud for High-Frequency Trading [1].

In this architecture, trading, data collection, analytics, and run-time risk management modules are deployed to the cloud. Existing systems can be designed to exist in a cloud as portability can be secured while moving to cloud environments [12]. Their functionalities and roles in the operation are the same as in contemporary high-frequency trading platforms. However, the integration of these modules, routing and data, and protocol conversions between them, are now handled with an ESB. Modules provide standardized interfaces to be accessed and managed in the cloud.

Cloud Manager (CM) is the common management system which, also manages request and response flows in the cloud. CM is directly connected to electronic execution platforms and data providers. Modules which, need interaction with electronic execution platforms and data providers use CM to access outside the cloud. All routing, data and protocol transformations, mediations and messaging between modules and CM are done via ESB. This provides flexibility and standardized integration of the system components.

CM provides web and rich mobile application channels as single points of control for the cloud, boosting the speed of user interactivity and control. Data for external data analysis software can be exported via Cloud Manager.

CM is also responsible for cloud specific management tasks:

- Account management for cloud users and consumers.

- Authorization and access control of users for modules and resources.
- Scheduling of jobs and tasks as well as selecting and provisioning suitable resources in the cloud.
- Routing of incoming requests from outside the cloud to the ESB to run associated processes, and vice versa.
- Application management for deployed applications (modules) including application specific configurations.
- Policy management for cloud resources and configuration of SLAs guaranteeing service availability and delivery.
- Monitoring of the entire cloud including users, tasks, processes, modules and resources.
- Security of the cloud.
- Providing audit records of the cloud.
- Metering, usage-based billing and billing management.

The whole cloud is co-located in datacenters close to the electronic execution platforms to avoid data movement costs and network latency, and to assure speed of execution [4][6][7].

The Financial Business Cloud for High-Frequency Trading is a model to adopt cloud computing as an IT for infrastructure financial institutions running high-frequency operations. It brings the benefits of cloud computing to high-frequency trading and addresses business specific issues explained in the previous sections.

I. CONCLUSION AND FUTURE WORK

This research presents a new business cloud model to create an efficient high-frequency trading platform. Portability can be achieved while moving financial calculation environments from on-premise to cloud environments. The adoption of cloud computing can address the computing power need while reducing the TCO.

These outputs show that current drawbacks and needs of high-frequency trading are addressed by the proposed reference model.

High-frequency trading has had three key effects on markets. First, it has meant ever-larger volumes of trading have been compressed into ever-smaller chunks of time. Second, it has meant strategic behavior among traders is occurring at ever-higher frequencies. Third, it is not just that the speed of strategic interaction has changed but also its nature. Yesterday, interaction was human-to-human. Today, it is machine-to-machine, algorithm-to-algorithm. For algorithms with the lifespan of a ladybird, this makes for rapid evolutionary adaptation.

Bid-ask spreads have fallen by an order of magnitude since 2004, from around 0.023 to 0.002 percentage points. On this metric, market liquidity and efficiency appear to have improved. High-frequency trading has greased the wheels of modern finance [32].

As it continues increasing importance of high-frequency in financial markets, it is obvious that cloud based models would be evolving accordingly.

Future research and work on this study will implement a complete real life prototype of this reference model to test performance benefits and the efficiency of the system, from both cloud consumer and cloud provider perspectives. The implementation and development of each component of the proposed model, and the development of security and management approaches are also subject to future work.

ACKNOWLEDGMENT

The authors would like to sincerely thank Prof. Dr. Vedat Akgiray (Capital Board of Turkey), Dr. Çetin Ali Dönmez (TURKDEX), Sayad R. Baronyan (Center for Computational Finance, Özyeğin University) and İbrahim Demir (Istanbul Innovation Center, IBM Turkey) for their cooperation with and contributions to this research. Also, special thanks go to IBM Turkey and the Turkish Derivatives Exchange (TURKDEX) for their material and data support for this research.

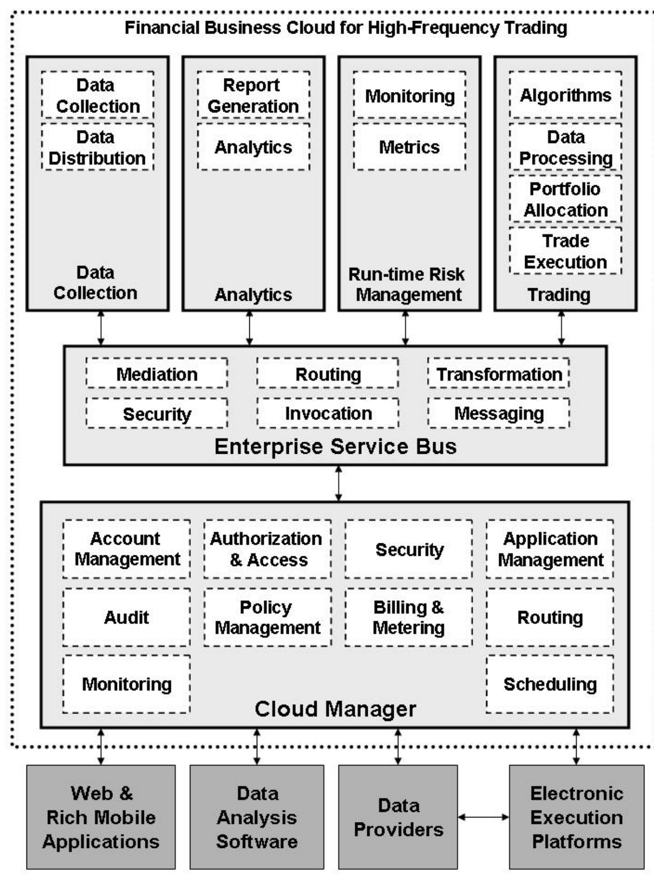

Figure 3. Reference component architecture of Financial Business Cloud for High-Frequency Trading [1].

REFERENCES

[1] A. Agopyan, E. Şener, and A.Beklen, "Financial Business Cloud for High-Frequency trading", CLOUD COMPUTING 2010 (CC2010): The First International Conference on Cloud Computing, GRIDs, and Virtualization, November 2010.

[2] I. Aldridge, High-Frequency Trading. New Jersey: Wiley, 2010.

[3] L. Tabb, R. Iati, and A. Sussman, "US Equity High Frequency Trading: Strategies, Sizing and Market Structure", 2009, Report by TABB Group.

[4] Internet: High-frequency trading: Up against a bandsaw. Available on WWW at URL: http://www.ft.com/cms/s/0/b2373a36-b6c2-11df-b3dd-00144feabdc0.html (Last access date: June 2011).

[5] Internet: High-frequency trading surges across the globe. Available on WWW at URL: http://www.dnaindia.com/money/report_high-frequency-trading-surges-across-the-globe_1319167 (Last access date: June 2010).

[6] Internet: High-frequency trading. Available on WWW at URL:http://www.vimeo.com/6056298 (Last access date: June 2010).

[7] M. Armbrust et al., "Above the Clouds: A Berkeley View of Cloud Computing", Technical Report, 2009. Electrical Engineering and Computer Sciences, University of California at Berkeley, February 2009.

[8] M. Skilton et al., "Building Return on Investment from Cloud Computing", White Paper, The Open Group, April 2010.

[9] R. Buyyaa, C. S. Yeoa, S. Venugopala, J. Broberg, and I. Brandic, "Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility", Future Generation Computer Systems, December 2008.

[10] L. Zhang and Q. Zhou, "CCOA: Cloud Computing Open Architecture", IEEE International Conference on Web Services, 2009

[11] D. A. Chappell, Enterprise Service Bus, O'Reilly Media, June 2004, p.1.

[12] V. Chang, G. Wills, and D. De Roure, "Towards Financial Cloud Framework - Modelling and Benchmarking of Financial Assets in Public and Private Clouds", University of Southampton.

[13] Internet: New wave of high-frequency traders to target European markets. Available on WWW at URL:http://www.thetradenews.com/node/4395 (Last access date: June 2010).

[14] Internet: What's Behind High-Frequency Trading. Available on WWW at URL: http://online.wsj.com/article/SB124908601669298293.html (Last access date: June 2010).

[15] Internet: What is FIX? Available on WWW at URL: http://www.fixprotocol.org/what-is-fix.shtml (Last access date: June 2010).

[16] Internet: Istanbul Stock Exchange. Available on WWW at URL: http://www.ise.org (Last access date: June 2011).

[17] Internet: Turkish Derivatives Exchange. Available on WWW at URL: http://www.turkdex.org.tr (Last access date: June 2011).

[18] B. Yan, and E. Zivot, "Analysis of High-Frequency Financial Data with S-Plus", 2003, White Paper

[19] T. Hill, and P. Lewicki, STATISTICS Methods and Applications. Tulsa:StatSoft, 2007.

[20] D. Gujarati, and D. Porter, Basis Econometrics (4^{th} Ed.). New York: McGraw-Hill/Irwin, 2004.

[21] Hills Capital Management, "An Analysis of the Moving Average Convergence / Divergence Indicator", Darren Brothers - Hills Capital Management.

[22] Internet: MACD (Moving Average Convergence/Divergence). Available on WWW at URL: http://www.investopedia.com/terms/m/macd.asp (Last access date: June 2011).

[23] J. W. Wilder, New Concepts in Trading Systems. Trend Research, 1978.

[24] Internet: The R Project for Statistical Computing. Available on WWW at URL: http://www.r-project.org (Last access date: June 2011).

[25] Internet: R-Forge: TradeAnalytics. Available on WWW at URL: https://r-forge.r-project.org/projects/blotter (Last access date: June 2011).

[26] Internet: R-Forge: TradeAnalytics – Contributed Packages. Available on WWW at URL: https://r-forge.r-project.org/R/?group_id=316 (Last access date: June 2011).

[27] Internet: IBM Istanbul Innovation Center. Available on WWW at URL: http://www.ibm.com/isv/spc/istanbul.html (Last access date: June 2011).

[28] Internet: Amazon Elastic Compute Cloud. Available on WWW at URL: http://aws.amazon.com/ec2 (Last access date: June 2011).

[29] Internet: Biocep-R, Statistical Analysis Tools for the Cloud Computing Age. Available on WWW at URL: http://biocep-distrib.r-forge.r-project.org/(Last access date: June 2011).

[30] Internet: Amazon Machine Images (AMIs).Available on WWW at URL: http://aws.amazon.com/amis (Last access date: June 2011).

[31] Internet: IBM BladeCenter LS21.Available on WWW at URL: http://publib.boulder.ibm.com/infocenter/bladectr/documentation/topic/com.ibm.bladecenter.ls21.doc/bls_ls21_product_page.html (Last access date: June 2011).

[32] Internet: More on High Frequency Trading and Liquidity. Available on WWW at URL: http://marginalrevolution.com/marginalrevolution/2011/10/more-on-high-frequency-trading-and-liquidity.html (Last access date: January 2012).

TABLE IV. R EXECUTION CODE FOR BASIC DESCRIPTIVE STATISTICS

```
## HFTBasicStatistics calculates the mean, median, variance, standart deviation, kurtosis and skewness of the xu30 high-frequency
## data.
HFTBasicStatistics <- function() {
  x = data.matrix(data.frame((my.csv.data)))
  for (i in 1:length(x)){
   if (i>1){
      a[i-1] = x[i]/x[i-1]
      a[i-1]= log(a[i-1])
   }
  }
  print(mean(a))
  print(median(a))
  print(var(a))
  print(sd(a))
  print(kurtosis(a))
  print(skewness(a))
}

## "System.time" provides execution time of HFTBasicStatistics function.
system.time(HFTBasicStatistics())
```

TABLE V. R EXECUTION CODE FOR Z-SCORE

```
## CalcZscore function calculates zscore of new fictional record by using real XU30 data in the recordset.
CalcZscore <- function() {
  x = data.matrix(data.frame((my.csv.data)))
  m <- mean(x)
  print(m)
  s <- sd(x,na.rm=TRUE)
  z = (72000-m)/s
  print(z)
}

## "System.time" provides execution time of CalcZscore function.
system.time(CalcZscore())
```

TABLE VI. R EXECUTION CODE FOR MACD

```
## XU30macd function provides to implement Exit, Entry and Risk strategy based on Moving Average Convergence-Divergence
##(MACD) of XU30 high frequency data. It also draws final MACD graphic.

XU30macd = function(){

stock.str='XU30_matrix'
data(XU30_matrix)
XU30_matrix<-as.xts(XU30_matrix)

initDate='2007-04-01'
initEq=60000
portfolio.st='macd'
account.st='macd'

initPortf(portfolio.st,symbols=stock.str, initDate=initDate)
initAcct(account.st,portfolios=portfolio.st, initDate=initDate)
initOrders(portfolio=portfolio.st,initDate=initDate)

maType="EMA"
signalMA = 9
fastMA = 12
slowMA = 26

currency('TL')
stock(stock.str,currency='TL',multiplier=1)

stratMACD <- strategy(portfolio.st)
stratMACD <- add.indicator(strategy = stratMACD, name = "MACD", arguments = list(x=quote(Cl(XU30_matrix))) )

## Enter to market strategy
stratMACD <- add.rule(strategy = stratMACD,name='ruleSignal', arguments = list(sigcol="signal.gt.zero",sigval=TRUE, orderqty=70,
ordertype='market', orderside='long', threshold=NULL),type='enter')

## Stop to buy strategy
stratMACD <- add.rule(strategy = stratMACD,name='ruleSignal', arguments = list(sigcol="signal.gt.zero",sigval=TRUE, orderqty=-70,
ordertype='stoplimit', orderside='long', threshold=.60,tmult=TRUE),type='risk')

## Exit strategy
stratMACD <- add.rule(strategy = stratMACD,name='ruleSignal', arguments = list(sigcol="signal.lt.zero",sigval=TRUE, orderqty='all',
ordertype='market', orderside='long', threshold=NULL),type='exit')

stratMACD <- add.signal(strategy = stratMACD,name="sigThreshold",arguments =
list(column="signal",relationship="gt",threshold=0,cross=TRUE),label="signal.gt.zero")
stratMACD <- add.signal(strategy = stratMACD,name="sigThreshold",arguments =
list(column="signal",relationship="lt",threshold=0,cross=TRUE),label="signal.lt.zero")

getSymbols(stock.str,from=initDate)
start_t<-Sys.time()
out<-try(applyStrategy(strategy=stratMACD , portfolios=portfolio.st,parameters=list(nFast=fastMA, nSlow=slowMA,
nSig=signalMA,maType=maType)))
end_t<-Sys.time()
print(end_t-start_t)

start_t<-Sys.time()
updatePortf(Portfolio=portfolio.st,Dates=paste('::',as.Date(Sys.time()),sep=''))
end_t<-Sys.time()

print(end_t-start_t)

chart.Posn(Portfolio=portfolio.st,Symbol=stock.str)
plot(add_MACD(fast=fastMA, slow=slowMA, signal=signalMA,maType="EMA"))
}

## "System.time" provides execution time of XU30macd function.
system.time(XU30macd())
```

TABLE VII. R EXECUTION CODE FOR RSI

```
## XU30rsi provides to implement Relative Strength Index (RSI) based entry / exit trading strategy.
##It uses xu30 high frequency data. It also helps to add thresholds, rules and signals.

XU30rsi = function(){

# Strategy object
stratRSI <- strategy("RSI")

# Indicator
stratRSI <- add.indicator(strategy = stratRSI, name = "RSI", arguments = list(price = quote(getPrice(XU30_matrix))), label="RSI")

# RSI is greater than 70
stratRSI <- add.signal(strategy = stratRSI, name="sigThreshold",arguments = list(threshold=70, column="RSI",relationship="gt", cross=TRUE),label="RSI.gt.70")

# RSI is less than 30
stratRSI <- add.signal(strategy = stratRSI, name="sigThreshold",arguments = list(threshold=30, column="RSI",relationship="lt",cross=TRUE),label="RSI.lt.30")

# Buy when the RSI crosses below the threshold 30
stratRSI <- add.rule(strategy = stratRSI, name='ruleSignal', arguments = list(sigcol="RSI.lt.30", sigval=TRUE, orderqty= 500, ordertype='market', orderside='long',
pricemethod='market', replace=FALSE), type='enter', path.dep=TRUE)


stratRSI <- add.rule(strategy = stratRSI, name='ruleSignal', arguments = list(sigcol="RSI.gt.70", sigval=TRUE, orderqty='all', ordertype='market', orderside='long', pricemethod='market', replace=FALSE), type='exit', path.dep=TRUE)


# Sell when the RSI crosses above the threshold 70
stratRSI <- add.rule(strategy = stratRSI, name='ruleSignal', arguments = list(sigcol="RSI.gt.70", sigval=TRUE, orderqty=-500, ordertype='market', orderside='short', pricemethod='market', replace=FALSE), type='enter', path.dep=TRUE)

stratRSI <- add.rule(strategy = stratRSI, name='ruleSignal', arguments = list(sigcol="RSI.lt.30", sigval=TRUE, orderqty='all', ordertype='market', orderside='short', pricemethod='market', replace=FALSE), type='exit', path.dep=TRUE)

currency("TL")

symbols = c("XU30")
for(symbol in symbols){
   stock(symbol, currency="TL",multiplier=1)
           getSymbols(symbol)
}

applySignals(strategy=stratRSI, xu30_matrix=applyIndicators(strategy=stratRSI, xu30_matrix=symbols[1]))

initEq=60000
port.st<-'RSI'
initDate='2007-04-01'

initOrders(portfolio=port.st, initDate=initDate)
initAcct(port.st, portfolios=port.st, initDate=initDate)
initPortf(port.st, symbols=symbols, initDate=initDate)


start_t<-Sys.time()
out<-try(applyStrategy(strategy=stratRSI , portfolios=port.st, parameters=list(n=2) ) )
end_t<-Sys.time()

start_t<-Sys.time()
updatePortf(Portfolio=port.st,Dates=paste('::',as.Date(Sys.time()),sep=''))
end_t<-Sys.time()
}

## "System.time" provides execution time of XU30rsi function.
system.time(xu30rsi())
```

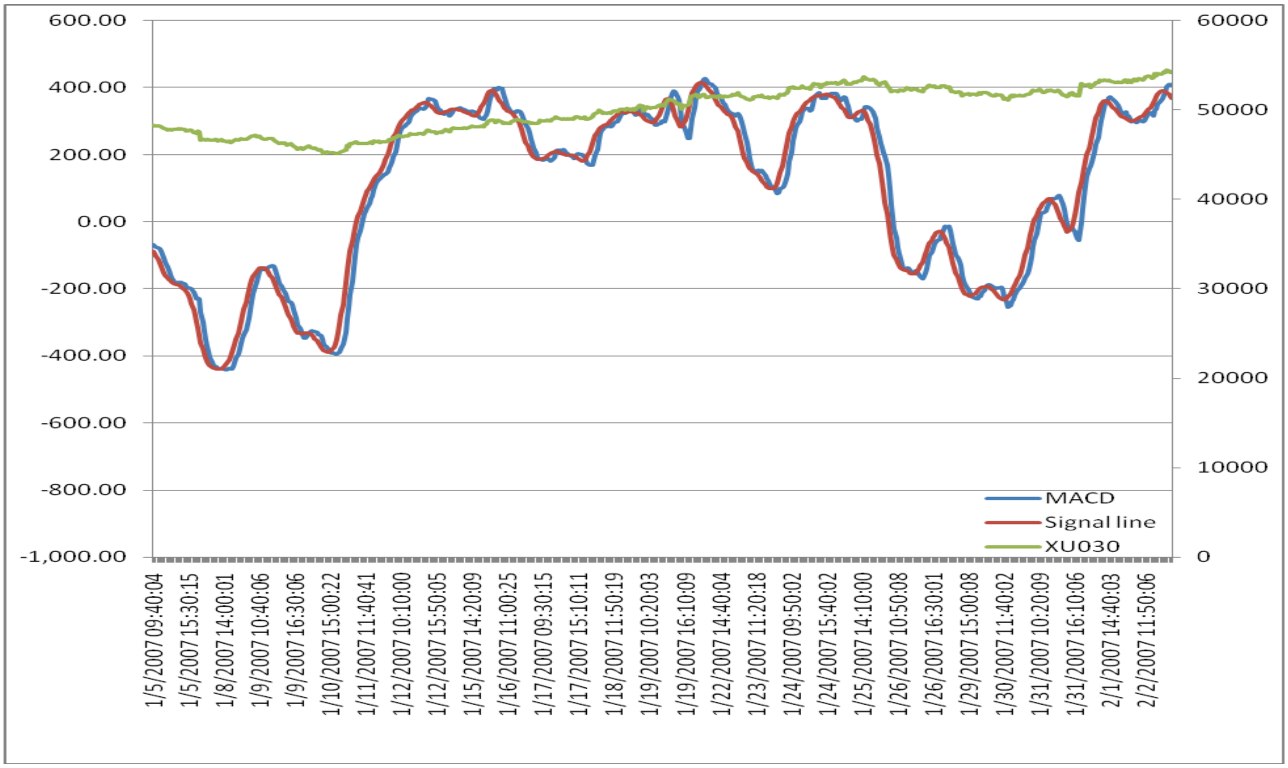

Figure 4. MACD graph for 2 months data.

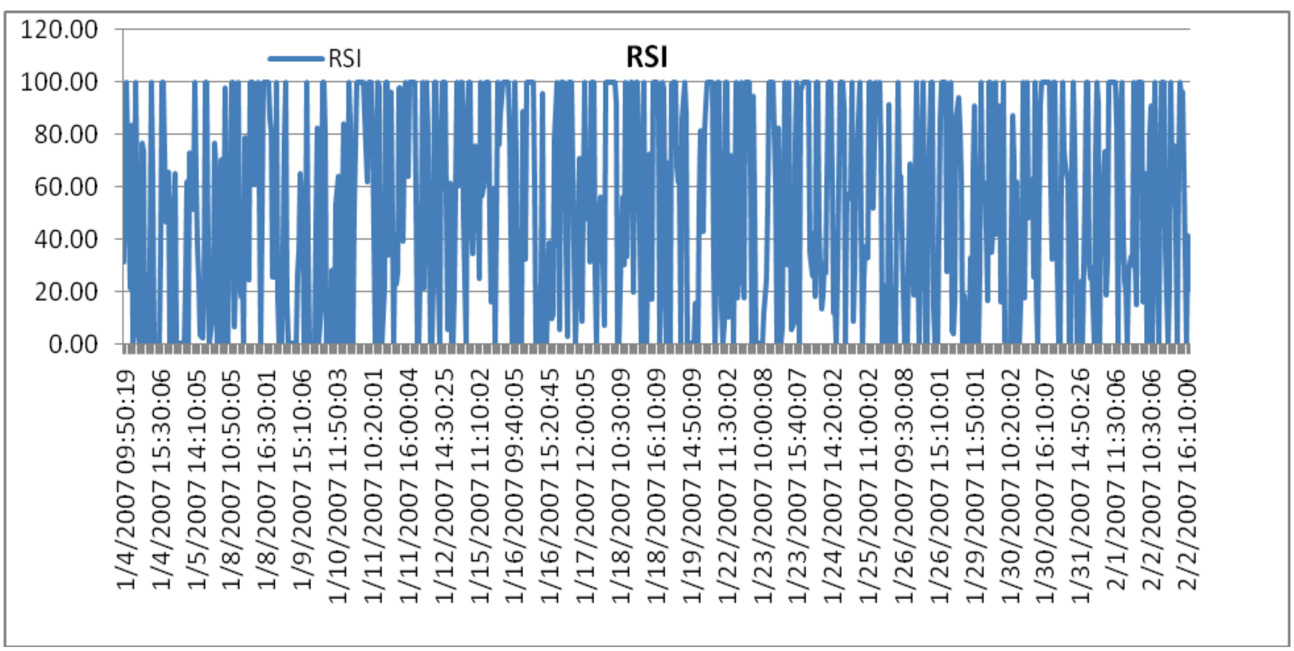

Figure 5. RSI graph for 2 months data.

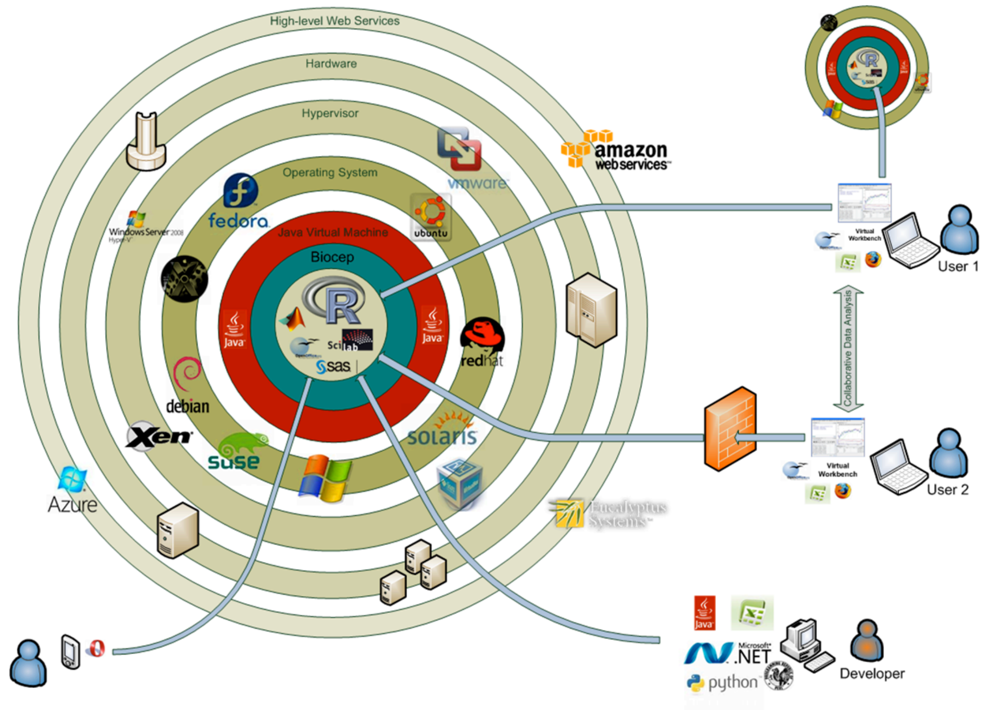

Figure 6. Biocep-R within the Technology Environment [29].