

FOFX Batch Performance Monitoring, Analytics, Ticket Management Metrics

A Major Qualifying Project Report:

Submitted to the Faculty

of the

WORCESTER POLYTECHNIC INSTITUTE

In partial fulfillment of the requirements for the

Degree of Bachelor of Science

By

Bogomil Tselkov

Alec Cunningham

Date: December 14, 2007

In cooperation with:

Greg Friel
Tom Mollica
Lehman Brothers

Approved:

Professor Arthur Gerstenfeld, Major Advisor

Professor Michael J. Ciaraldi, Co-Advisor

Table of Contents

Abstract	3
Acknowledgements	4
Executive Summary	5
Background	7
Methodology.....	10
Tools and Results.....	22
Future Recommendations.....	41
References	45

Abstract

The goal of the project is to produce metrics and analysis of the overnight batch processes at Lehman Brothers. Due to recent market forces and business initiatives there is a marked increase in trading volumes. Our task was to produce tools, which can help analyze, visualize and interpret the run time data, volume information and the delivery expectations, based on different statistic techniques.

Another target of the project is to provide metrics and monitoring capability for the ticket management system at Lehman Brothers and to produce analysis for simple presentation of the ticket data in a variety of forms.

Acknowledgements

We would like to extend thanks to our sponsors for allowing us into their business and teaching us what we needed in order to complete this project. Specifically at Lehman Brothers we would like to thank Greg Friel, Tom Mollica and Bhrugu Giri for their constant support in the completion of this project. We would also like to thank our advisors, Professors Gerstenfeld and Ciaraldi for their efforts to help us throughout.

1. Executive Summary

This MPQ focuses mainly on the Lehman Brothers' FOFX Batch performance monitoring, analytics and on the improvement and enhancement of the Ticket Management Dashboard System.

In order to achieve that, our team started with the FOFX Batch performance analysis, which included:

- Identifying the key milestones of the system
- Obtaining and storing data
- Representation of the data and creating metrics

Also, Lehman Brothers management wanted useful metrics of the ticket system. The information was already stored in a database, the only question was how to extract it and present it. The first step was to figure out what data each ticket contained, and to ask the people who were going to be using the ticket management system what parts of the data they wanted to see. The state of the system when we took over was an excel file which had the data imported into one of its sheets, and several other sheets with some tables on them. Our task was to correct problems with the current tables, add new tables, and create graphs. The tables needed appropriate formulas to retrieve the appropriate information from the data sheet.

After we improved the dashboard system, we continued our work on creating Excel based analytical tools. Both trading volume analysis and runtime analysis tools were created, being separated into:

- FOFX Key Jobs Runtime Performance graph
- FOFX Key Box Runtime Performance Graph
- Job Run Time with Respect to the Average Run Time Analysis
- Job Run Time with Respect to Floating Standard Deviation Time Analysis
- Job Run Time with Respect to Moving Average Time Analysis

For further recommendations and plans to evolve the project, we came up with the following data analysis ideas, well-known as trading analysis tools. However, they are still applicable in capturing trends for the batch runtimes or SAL proximity and that is why are definitely worth implementing:

- Volatility Channels
- Bollinger Breakouts Analysis
- Donchian Trends

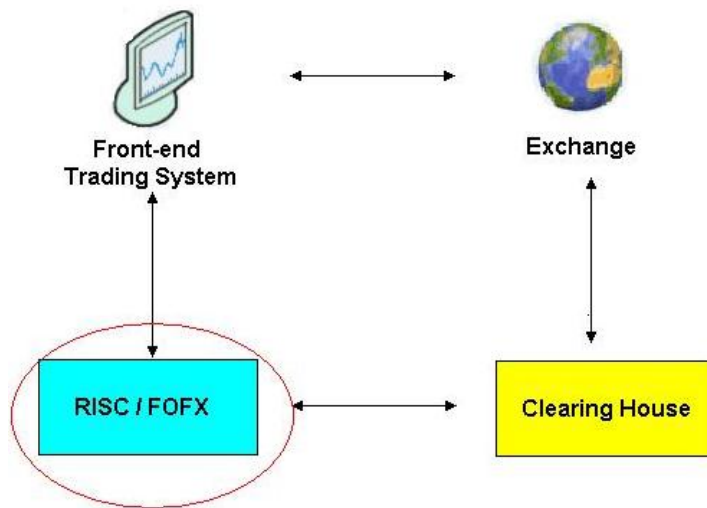
Other further recommendations were also provided.

2. Background

In order to understand our project better, a few key areas needed to be researched. Main part of this project is FOFX runtime analytics. That is why it was extremely important to understand where FOFX stands in the business cycle. This led us to the specific business area, that this system supports, and we had to get familiar with some financial and business concepts in order to create appropriate metrics. For the analysis part a specific research on statistical tools was required and some of our ideas were born thanks to this research. Understanding of the current software systems used at Lehman Brothers was also needed.

2.1 Where do we stand at business?

In order to understand and produce analytics for the system on which we were going to work, we started with understanding the place in the business work it occupies.



The FOFX System (Futures Options and Foreign Exchange) is a connection between the Front-end Trading system and the Clearing Houses that clear the traders ordered through the Exchange.(Friel 2007). Its main responsibility is to ‘Lehmanize’ (format, arrange and distribute) the data produced by another core system called RISC.

2.2 Key business concepts

As part of the Futures Options and Foreign Exchange Settlements team, that is an application development team responsible for the development, implementation and support of settlement and clearing functions for listed derivatives and FX products. (Lehman Brothers 2007), we had the responsibility to analyze the runtime of different

batch processes and compare it to their client service level agreements (SLAs). (Project Proposal 2007). That is why we had to get familiar with the concepts of the SLA: An SLA is a formally negotiated agreement between two parties. It is a contract that exists between customers and their service provider, or between service providers. It records the common understanding about services, priorities, responsibilities, guarantee, and such—collectively, the *level of service*. For example, it may specify the levels of availability, serviceability, performance, operation, or other attributes of the service like billing and even penalties in the case of violation of the SLA. (Lee 2002)

Also, since we were part of the derivatives support group, we had to understand these business concepts:

What is *option*?

Def: Options are financial instruments that convey the right, but not the obligation, to engage in a future transaction on some underlying security, or in a futures contract.

There are mainly two types of options: call option and put option.

A *call option* gives the owner, the right to buy the underlying asset by a certain date for a certain price.

A *put option* gives the owner the right to sell the underlying asset by a certain date for a certain price.

The price in the contract is known as the exercise price or strike price. The date in the contract is known as the expiration date or maturity. There are mainly two types of options – *American* and *European*. (There are also other types of options like *Bermudian* options and *Barrier* options, but they will not be used in the paper).

American options can be exercised at any time up to the expiration date. European options can be exercised only on the expiration date itself.

We also found worth looking at specific business days with unusual volume activity - just like the *Triple Witch Day* on the third Friday of every March, June, September, and December.

Just for information - *Triple witching hour* is the last hour of the stock market trading session (3:00-4:00 P.M., New York Time) on the third Friday of every March, June, September, and December. Those days are the expiration of three kinds of securities:

- *Stock index futures*
- *Stock index options*
- *Stock Options*

2.3 Database systems and Software

An Oracle database was our primary database at Lehman Brothers. That is why we got familiar with more advanced Oracle database manipulations like grouping, procedures and triggers. We also tried to get basic understanding of the DBArtesian Software, which is a database program used at Lehman Brothers for accessing Oracle databases, constructing and executing queries.

2.4 Statistic Analysis

Different books and sources on Statistics were used for our preparation. A complete list can be found in the Reference section of this paper.

Some of the concepts included in the paper are:

- **Standard Deviation:** Standard Deviation of values is a measure of the spread of its values. The standard deviation is usually denoted with the letter σ (lower case sigma). It is defined as the square root of the variance
- **Moving Average:** In statistics, a moving average or rolling average is one of a family of similar techniques used to analyze time series data. It is applied in finance and especially in technical analysis. It can also be used as a generic smoothing operation, in which case the raw data need not be a time series. A simple moving average (SMA) is the unweighted mean of the previous n data points. For example, a 10-day simple moving average of closing price is the mean of the previous 10 days' closing prices.
- **Linear Regression:** Linear regression is a form of regression analysis in which observational data are modeled by a function which is a linear combination of the model parameters and depends on one or more independent variables. In simple linear regression the model function represents a straight line. The results of data fitting are subject to statistical analysis.

3. Methodology

This year's project at Lehman Brothers focuses mainly on the FOFX Batch performance monitoring, analytics and Ticket Management System metrics and improvements.

3.1 FOFX Batch Performance and Analytics

Our plan for the work was mainly based on the scope of the project:

- Identifying the key milestones of the system
- Obtaining data
- Representation of the data and creating metrics

3.1.1 Identify the Key FOFX Jobs

In order to accomplish our goal of creating tools for analysis, we first had to get familiar with the FOFX processes, which are monitored by the Operation Technology group. For this task we mainly used the Lehman Brother's online database page, known as Lehman Live. Then based on key-words search, we were able to obtain information and relations between the different FOFX jobs. As mentioned earlier, the FOFX system includes a list of many batch processes. There is a quite complicated structure of connections among these batches, but to monitor the whole process it is enough to keep track of the key FOFX jobs, which show the progress of the whole system.

After comparing the relations within the Lehman Live system, we were able to identify the ten processes with the most dependencies. This made us think that those processes are quite important for the overall performance.

Later on we continued our research with a conference call with Gautam Mahatme, member of the FOFX support team, who helped us establish the following table with important processes within the FOFX. We were able to identify 60 such key processes.

3.1.2. Identify the Key Box Jobs and the Time stamps SLAs to them.

We also managed to create a group mapping of those processes, which shows their consequence and their primary task.

That is how we were able to identify 15 boxes of FOFX jobs, which are essential for runtime of the whole FOFX.

Here is the mapping table we created:

BOXES	JOB_NAMES	SEQUENCE
Cameo_Asia_Ex_Japan	FOFX_POS_LOAD_PASIA_Job	1
Cameo_Asia_Ex_Japan	FOFX_BAL_UPD_PNS_PASIA_Job	1
Cameo_Asia_Ex_Japan	FOFX_CAMEO_PME_PASIA_Job	2
Cameo_Asia_Ex_Japan	FOFX_POS_FILE_PASIA_Job	2
Cameo_Asia_Ex_Japan	FOFX_GSSR_EQUITY_R3_Job	2
Asia_Memo	FOFX_GQ_POSITION_ASIA_Job	2
Asia_Memo	FOFX_POS_LOAD_ASIA_Job	1
Asia_Memo	FOFX_GSSR_EQUITY_ASIA_Job	2
Cameo_Futures_Cust	FOFX_POS_LOAD_CUST_Job	1
Cameo_Futures_Cust	FOFX_ACV_LOAD_CUST_Job	1
Cameo_Futures_Cust	FOFX_PNS_LOAD_CUST_Job	1
Cameo_Futures_Cust	FOFX_BAL_UPD_PNS_CUST_Job	1
Lehman_Risc	FOFX_GEN_POSFUT_CUST_Job	2
Lehman_Risc	FOFX_GEN_POSOPT_CUST_Job	2
DMS_NewYork	FOFX_STM_FX_LOAD_DMS_Job	1
DMS_NewYork	FOFX_DMS_POS_Job	2
DMS_Tokyo	FOFX_DMS_TK_Job	2
DMS_London	FOFX_DMS_LON_Job	2
DMS_NewYork	FOFX_DMS_REST_Job	3
Cameo_Futures_Firm	FOFX_PNS_LOAD_Job	1
Cameo_Futures_Firm	FOFX_POS_LOAD_Job	1
Cameo_Futures_Firm	FOFX_ACV_LOAD_Job	1
Cameo_Futures_Firm	FOFX_BAL_UPD_FROM_PNS_Job	1
GQUEST_London_NY	FOFX_GQ_POSITION_Job	2
Cameo_Futures_Cust	FOFX_CAMEOPOS_CUST_Job	2
Cameo_Futures_Cust	FOFX_CAMEOACV_CUST_Job	2
Cameo_Futures_Cust	FOFX_CAMEOPNS_CUST_Job	2
Cameo_Futures_Cust	FOFX_CAMEO_PME_CUST_Job	2
Cameo_Futures_Firm	FOFX_CAMEOPOS_REST_Job	2
Cameo_Futures_Firm	FOFX_CAMEOACV_REST_Job	2
Cameo_Futures_Firm	FOFX_CAMEOPNS_REST_Job	2
Cameo_Futures_Firm	FOFX_CAMEO_PME_REST_Job	2
DOLFIN_London	FOFX_GEN_TRADES_Job	2
DOLFIN_London	FOFX_GEN_TRADES_SORT_Job	3
DOLFIN_London	FOFX_GSSR_EQUITY_R3_Job	2
MUREX	FOFX_MUREX_CASH_Job	1
PALS_London	FOFX_PALS_FX_CASH_Job	1
EPAS	FOFX_EPAS_POS_PNS_Files_Job	2
DOLFIN_Asia_Ex_Japan	FOFX_FTP_EDCOM_R3_TLM_Job	3
Cameo_Asia_Ex_Japan	FOFX_FTP_POS_READY_PASIA_Job	3
Cameo_Futures_Cust	FOFX_FTP_CAMEOPME_CUST_Job	3
Cameo_Futures_Firm	FOFX_FTP_CAMEOPME_REST_Job	3
Cameo_Futures_Cust	FOFX_FTP_CAMEOACV_CUST_Job	3
Cameo_Futures_Cust	FOFX_FTP_CAMEOPNS_CUST_Job	3
Cameo_Futures_Firm	FOFX_FTP_CAMEOACV_REST_Job	3
Cameo_Futures_Firm	FOFX_FTP_CAMEOPNS_REST_Job	3

Cameo_Futures_Cust	FOFX_FTP_CAMEOPOS_CUST_Job	3
Cameo_Futures_Firm	FOFX_FTP_CAMEOPOS_REST_Job	3
DMS_Tokyo	FOFX_FTP_DMS_TK_Job	3
DMS_London	FOFX_FTP_DMS_LON_Job	3
DMS_NewYork	FOFX_FTP_DMS_REST	4
Lehman_Risc	FOFX_FTP_LR_CMDY_OPT_Ready	3
Lehman_Risc	FOFX_FTP_LR_CMDY_FUT_Ready	3
GQUEST_London_NY	FOFX_LONEQ_FTP_POSITION_Job	3
DOLFIN	FOFX_FTP_GEN_TRADES_GEDS_Job	4
MUREX	FOFX_FTP_MUREX_CASH_Job	2
PALS_London	FOFX_FTP_PALS_FX_CASH_Job	2
GSSR_Asia_Memo	FOFX_FTP_GSSR_EQUITY_ASIA_Job	3
GQUEST_Asia	FOFX_FID_FTP_POSITION_ASIA_Job	3
EPAS	FOFX_FTP_EPASFile_Job	3

Table 1: Job Map

With the help of the FOFX Support Team, we were able to obtain the SLA times for the required boxes.

SLA shows the cut-off time, by which the run time of the box should be completed.

3.1.3. Storing and obtaining data

Lehman Brothers already had a database in which they store some information about all the processes of the FOFX system. In our case this is an Oracle Database. To build tables, execute queries, and view or change the content of a database, we used primary SQLPlus and DBArtesian Software, which is a database program capable of accessing Microsoft SQL, Oracle and Sybase platforms over a network.

The information for the daily FOFX processes is stored in the data table FOFX_DAILY_BATCH_METRICS and includes the fields: FOFX_JOB_NAME, FOFX_JOB_RUN_DATE, FOFX_START_TIME, FOFX_JOB_END_TIME, FOFX_JOB_STATUS, FOFX_JOB_REMARKS

In order to obtain information for our grouping, we created a new database mapping table called FOFX_NAME_MAPPING, which connected the key processes with the Box jobs to which they were assigned. It contains fields JOB_NAMES, BOXES NAME and SEQUENCE. Then by using a grouping select statement, we created a Data View – FOFX_BOX_AGG_VW, containing the distribution of the job processes in Boxes.

Later on we implemented our idea of creating views based on both tables for calculating the runtime of the key Boxes.

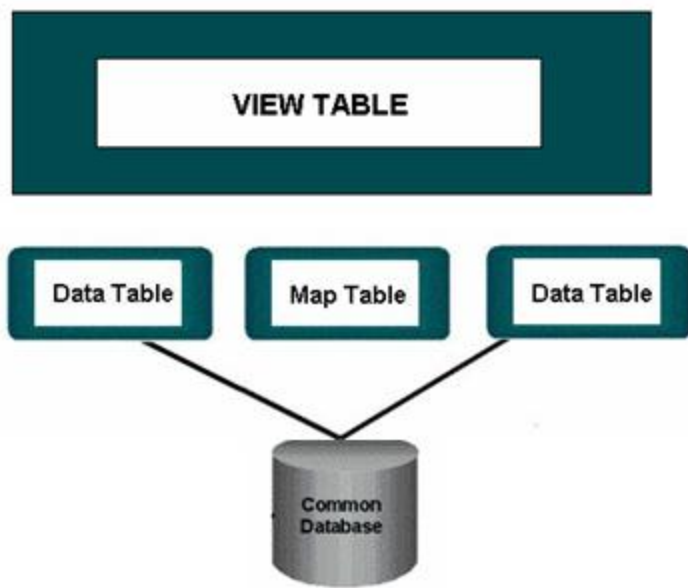


Figure 1: Data

This approach gives us great flexibility for changes, since we only have to modify the mapping table if a particular job needs to be added or removed. Also, a simple change in that table will be automatically updated in the view.

Later on, we used the same logic for creating different data tables and different views, which can be seen in the Appendix I.

Another piece of information that is extremely important for the run time of all batch processes is the current volume of trades and open positions that the bank has. As was mentioned earlier, there has been a steady increase in trading volume in the past couple of months, so it was obvious that capturing and measuring this data is valuable as well.

For this reason, we created an Oracle data table called FOFX_VOLUME_INFO. Its purpose is to capture volume information about the FOFX system, using automatic scripts provided by the FOFX Support Team.

We also created different data views based on the data tables, so calculations based on the instrument type, type of trade and locations were available on the database level.

3.1.4. Data presentation and manipulation

Once we had the data, we were ready to begin the creation of the analysis tools. The first question that we faced on this stage was – What type of software would be best for representing and analysis of the data? Initially, we had different ideas – Microsoft Excel, Dynamic Java Graphs or other third party data analysis tools.

However, taking into consideration, that on Wall Street and in Lehman Brothers as well, Microsoft Excel is one of the most well-known and widely used software applications, our team decided that it would be best to use Excel 2003.

Some of the great advantages that Excel gives us are:

- User friendly interface
- Familiar environment for Lehman Brothers
- Scalability
- Reliability
- Great build-in functionality

The connection to the Oracle Database is established by a Microsoft Excel's Oracle ODBC driver. This produced an easy way to import data to the spreadsheet from both Oracle data tables and data views.

Our team used two main ways to import the data into the Excel spreadsheets:

- 1) Importing regular data to columns

	A	B	C	D	E	F
1	Trading Volume					
2						
3			RUN_DATE	CATEGORY	INSTRUMENT_TYPE	VOLUME
4			11/20/2007	ACTIVITY	ALL	940608
5			11/20/2007	ACTIVITY	FUTURES	745097
6			11/20/2007	ACTIVITY	FX	240424
7			11/20/2007	ACTIVITY	OPTIONS	45991
8			11/20/2007	POSITIONS	ALL	3009116
9			11/20/2007	POSITIONS	FUTURES	2417645
10			11/20/2007	POSITIONS	FX	892921
11			11/20/2007	POSITIONS	OPTIONS	574083
12			11/21/2007	ACTIVITY	ALL	940608
13			11/21/2007	ACTIVITY	FUTURES	745097
14			11/21/2007	ACTIVITY	FX	240424
15			11/21/2007	ACTIVITY	OPTIONS	45991
16			11/21/2007	POSITIONS	ALL	3009116
17			11/21/2007	POSITIONS	FUTURES	2417645
18			11/21/2007	POSITIONS	FX	892921
19			11/21/2007	POSITIONS	OPTIONS	574083

Figure 2: Trading Volume Data

Since we are producing dynamic sheets that should be updated every day, we faced a problem of fixing regions with data, since every day there is new set of data populated and therefore sheet is growing in rows. That is why we combined this data extraction method with some Excel VBA programming. We created macros, producing dynamic ranges of data, which automatically shrink or expand if data has been deleted or added.

2) Importing Data directly as a Pivot Table Report

	A	B	C	D
1	FOFX Key Boxes Actual Runtime			
2				
3				
4		Runtime	BOXES	
5		Date	Asia Memo	Cameo Futures - Customer
6		2007/10/23	0.00755787	0.013553241
7		2007/10/24	0.00556713	0.011944444
8		2007/10/25	0.005902778	0.01837963
9		2007/10/30	0.005694444	0.015856481
10		2007/10/31	0.004131944	0.011655093
11		2007/11/01	0.005196759	0.014548611
12		2007/11/06	0.006631944	0.016087963
13		2007/11/07	0.00494213	0.017928241
14		2007/11/08	0.006006944	0.016006944
15		2007/11/12		0.003877315
16		2007/11/13	0.00755787	0.022766204
17		2007/11/14	0.007418981	0.035277778
18		2007/11/20	0.006898148	0.039444444
19		2007/11/22	0.007291667	0.028171296
20		2007/11/27	0.005208333	0.030925926
21		2007/11/28	0.005208333	0.030925926
22		2007/11/30	0.00693287	0.036608796

Figure 3: FOFX Runtime

As can be seen from Fig.3, using this method provides additional functionality of representation of the data.

We specified the layout of the pivot table in the excel spreadsheet, so we have a selection on the date on the y-axis and the job/box process on the x-axis. This format and model is consistent in all the tools build, which makes the user interface more friendly and easy to understand and work with. Overall, pivot table representation gives a superb way to summarize data and is powerful tool for data analysis.

3.2 Ticket Management System metrics and improvements.

Lehman Brothers has a significant amount of infrastructure and many different applications. Naturally there are problems and areas that need improvement. When the people who use these systems see something like that, they submit a ticket to the operations team. This ticket contains information on what application it applies to, what region it is from, what type of problem there is, and what the priority level is. These tickets go into a queue from which the members of the Operations Technology team can access and resolve them, as shown in this flow chart:

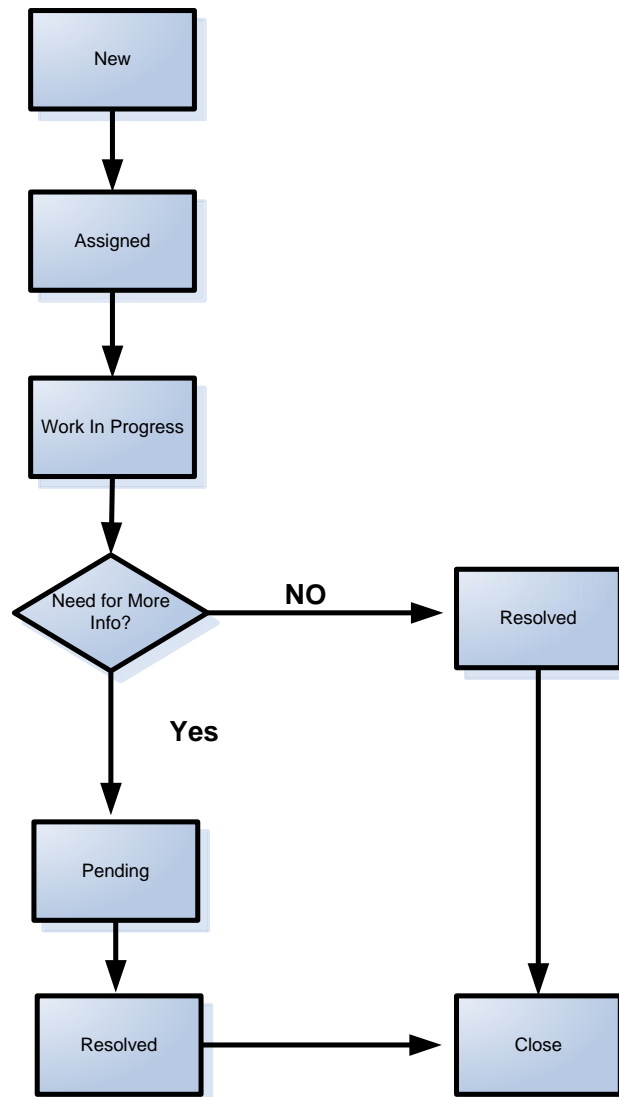


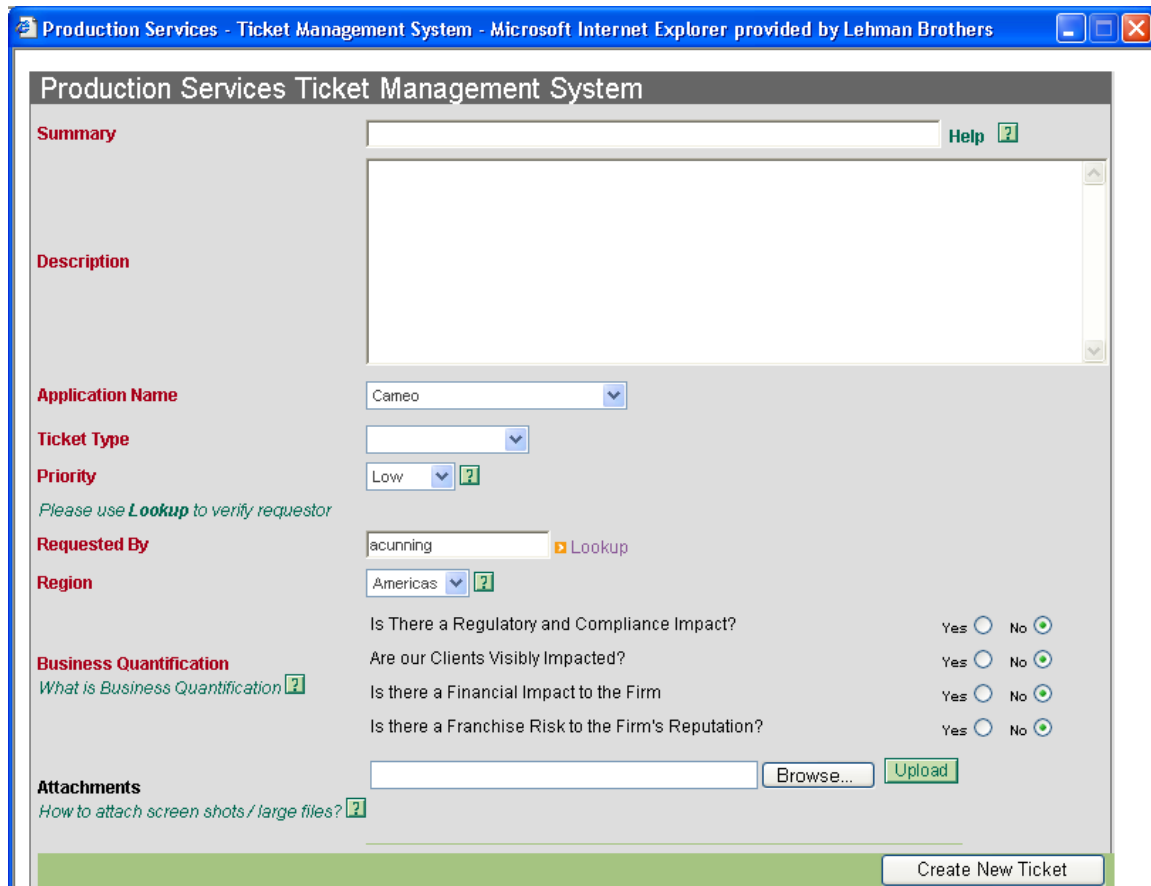
Figure 4: Ticket Flow Chart

The Operations Technology group would like to have statistical information about the tickets so that they can do their jobs better.

Lehman Brothers management wanted useful metrics of the ticket system. The information was already stored in a database, the only question was how to extract it and present it. The first step was to figure out what data each ticket contained, and to ask the people who were going to be using the ticket management system what parts of the data they wanted to see. The state of the system when we took over was an excel file which had the data imported into one of its sheets, and several other sheets with some tables on them. Our task was to correct problems with the current tables, add new tables, and create graphs. The tables needed formulas to retrieve the appropriate information from the data sheet. The graphs just took the information that they needed from the relevant table.

3.2.1 Parts of a Ticket

Our first task was to familiarize ourselves with the ticket system. We needed to know what information was contained in a ticket. We went to the website where you can submit tickets and looked at the form:



The screenshot shows a web browser window titled "Production Services - Ticket Management System - Microsoft Internet Explorer provided by Lehman Brothers". The main content area is titled "Production Services Ticket Management System" and contains a form with the following sections:

- Summary:** A text input field and a "Help" link.
- Description:** A large text area for entering the ticket details.
- Application Name:** A dropdown menu currently set to "Cameo".
- Ticket Type:** A dropdown menu.
- Priority:** A dropdown menu currently set to "Low".
- Requested By:** A text input field containing "acunning" and a "Lookup" button.
- Region:** A dropdown menu currently set to "Americas".
- Business Quantification:** A section with a link "What is Business Quantification?". It contains four questions, each with "Yes" and "No" radio buttons:
 - Is There a Regulatory and Compliance Impact? (Yes, No)
 - Are our Clients Visibly Impacted? (Yes, No)
 - Is there a Financial Impact to the Firm (Yes, No)
 - Is there a Franchise Risk to the Firm's Reputation? (Yes, No)
- Attachments:** A text input field, a "Browse..." button, and an "Upload" button. A link "How to attach screen shots / large files?" is also present.

At the bottom right of the form is a "Create New Ticket" button.

Figure 5: Ticket Submission Form

The summary and the description were only relevant to the people who were resolving the tickets; they were not useful for creating metrics. The application information was very important because if you know which application has the most problems then you know which application to focus on improving. The ticket types included things such as “bug”, “issue”, and “business request”. This information was useful because it is much more significant to get twenty bug reports than it is to get twenty enhancement requests. The ticket priority information was very important for our purposes. The “urgent” and “high” tickets were far more important than the “medium” and “low” tickets when considering metrics. Lehman Brothers has offices in New York, London, Tokyo, and India. Some offices are bigger than others and so naturally have more tickets assigned to them, but the smaller offices are expected to grow and it will be useful to see the growth in number of tickets assigned to these offices. Not seen in the screenshot but included in the ticket database is the date submitted. This is used to calculate how old an individual ticket is, which is useful to us because we can calculate the age of the tickets that haven’t yet been resolved.

3.2.2 The Excel File

When we started working on the excel file it already contained a data sheet titled ‘ALL QUEUES’ which was set to automatically retrieve the ticket information from the database whenever the excel file was opened. The information from each ticket was divided into nineteen columns. The file also included ‘Daily’, ‘Weekly’, and ‘All Time’ sheets, as well as a set of monthly sheets which at the moment only contains ‘October 2007’ and ‘November 2007’. The final sheet was called ‘Date Information’ and was used in calculations.

3.2.2.1 Adding a table

User Priority level	No. of Tickets
Urgent	4
High	32
Med	24
Low	82
Totals	142

Figure 6: User Priority Chart

The tables all used the same format to make the sheet easier to read. Each sheet used a different color in the header boxes to make it easy to see which sheet you were currently using. If the table had a ‘Totals’ box at the bottom then it was always the same color (as shown in the image), to make the tables easier to read.

There were three different ways of putting information in the cells in a table. The leftmost column in this table (Aging timeline) contains the scale; this information is simply entered and is static.

Aging Timeline (Open Tickets)	No. of Tickets	Urgent	High	Percentage
0-5	16	0	2	24%
6-10	28	0	9	41%
11-20	11	0	1	16%
21-30	13	1	2	19%
31-90				
91-180				

Figure 7: Aging Timeline Table

The middle three columns (No. of Tickets, Urgent, and High) involve somewhat complicated calculations.

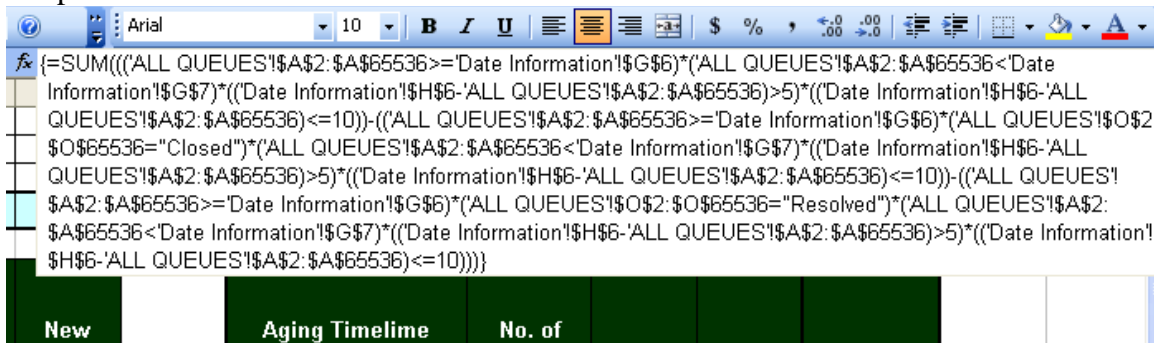


Figure 8: Aging Formula

This formula uses data from the 'ALL QUEUES' sheet and the 'Date Information' sheet. This part of the formula: 'ALL QUEUES'!\$A\$2:\$A\$65536 says to look in all of the cells in the 'A' column (the column that has the assign date information) of the 'ALL QUEUES' sheet. It is compared to the information in this table on the 'Date Information' sheet:

Month	Month Start Date	Month End Date
Oct-07	10/1/2007	10/31/2007
Nov-07	11/1/2007	11/30/2007
Dec-07	12/1/2007	12/31/2007
Jan-08	1/1/2008	1/31/2008

Figure 9: Date Table

This table contains the start (G column) and end (H column) dates for each month. So in the formula pictured above, 'Date Information'!\$G\$6 would return 11/1/2007.

Therefore, this section of formula

('ALL QUEUES'!\$A\$2:\$A\$65536>=Date Information!\$G\$6)*('ALL QUEUES'!\$A\$2:\$A\$65536<Date Information!\$G\$7)

refers to all of the tickets in the month of November. It can be read as (All tickets after 11/1/2007) and (Before 12/1/2007). The next section of formula

((('Date Information'!\$H\$6-'ALL QUEUES'!\$A\$2:\$A\$65536)>5)*(('Date Information'!\$H\$6-'ALL QUEUES'!\$A\$2:\$A\$65536)<=10))

subtracts the date of each ticket from the end date of the month and checks to see if it is between five and ten days old. Once the total number of tickets in the time frame that are in the range of days we want has been calculated, we subtract the tickets that are closed or resolved because we are only interested in open tickets. Adding a simple “*(‘ALL QUEUES’!\$O\$2:\$O\$65536=“Closed”)” to the previous statement and subtracting it from the original will subtract the tickets that have been closed. Similarly, adding “*(‘ALL QUEUES’!\$K\$2:\$K\$65536=“Urgent”)” calculates only those tickets that are urgent.

The final column of the aging chart is very simple. The formula “=IF(K19=0,"0%",K19/SUM(K19:K22))” calculates the percentage of open tickets between zero and five days (row K19) by doing some simple math on the values that are in the table, without needing to access the data sheet. If there are no tickets it simply displays “0%” without doing any calculation because otherwise Excel won’t calculate it properly.

3.2.2.2 Adding a Graph

Graphs such as this one are superior to the tables because you can grasp the necessary information in a single glance.

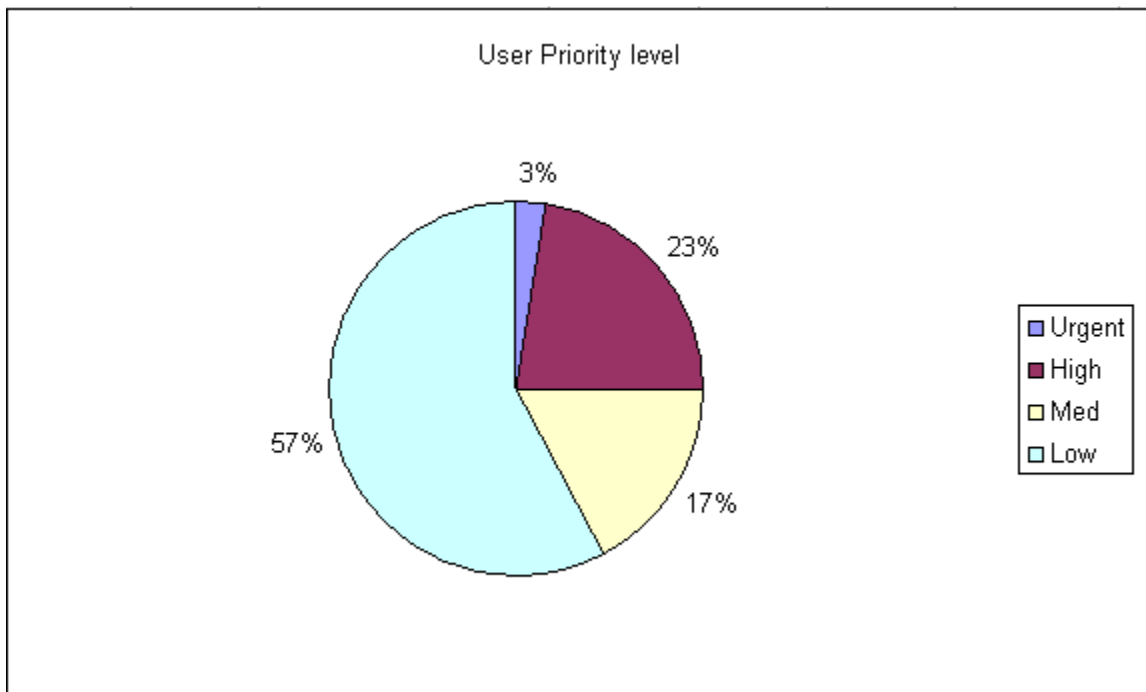


Figure 10: User Priority Graph

User Priority level	No. of Tickets
Urgent	4
High	32
Med	24
Low	82
Totals	142

Instead of:

Figure 11: User Priority Table

Creating the graph is quite simple. The top left cell of the table is used as the title of the graph, the column under the title is the legend, and the numbers in the other columns are the data. If there is more than one column then multiple series of data are used. When only a few variables are being tracked pie charts are used since they are the most readable. When lots of variables are being tracked however, pie charts become unreadable and bar graphs are superior.

4. Tools and Results

4.1 Trading Volume Analysis

As the business of the bank is growing, so does the trading volume. That leads to greater number of orders and traders, which on the other hand leads to more time consuming calculations of position, exposure and market conditions.

That is why it was extremely important for our analysis to produce a way of tracking this volume.

As can be seen on the figures below, we measure the volume data from different views:

- Total number of instruments, sorted by date

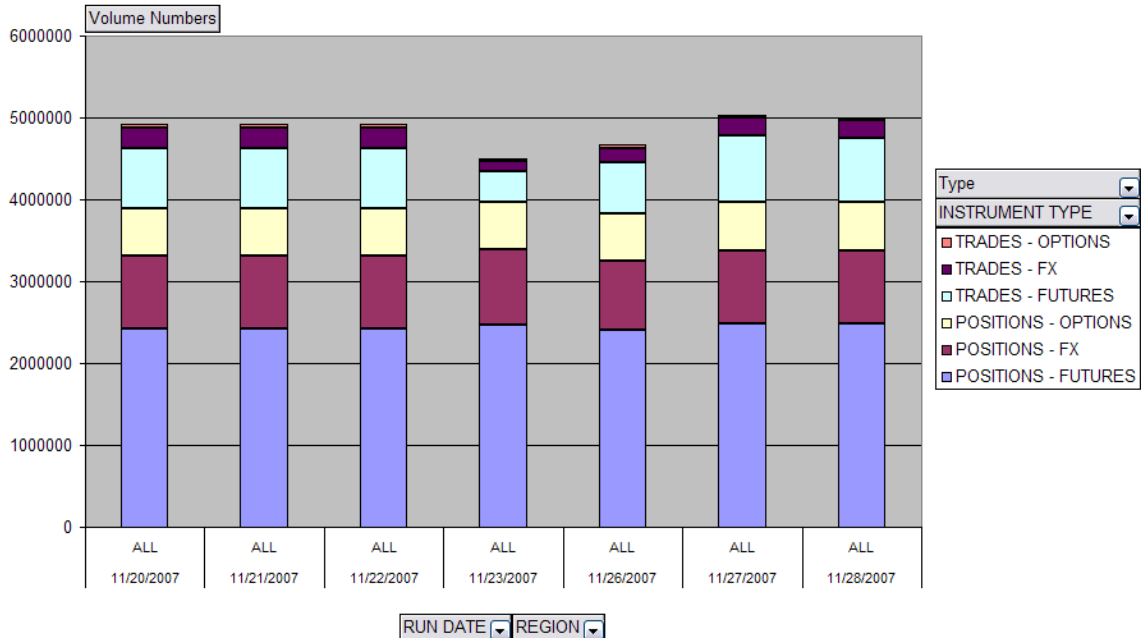


Figure 12: Instruments by Date

This gives the perspective how the volume grows as the time goes on. Different types of trend lines can be added to the graph, so a prediction for the future volume growth can be done.

This type of chart also provides us with knowledge which type of instrument or position has the biggest volume, so the senior management can easily track a day, which is out of ordinary – for example a day with more Future Trader than usual – just like the Triple Witch Day on the third Friday of every March, June, September, and December.

Just for information – **Triple witching hour** is the last hour of the stock market trading session (3:00-4:00 P.M., New York Time) on the third Friday of every March, June, September, and December. Those days are the expiration of three kinds of securities:

- Stock index futures index futures.
- Stock index options
- Stock Options

The simultaneous expirations generally increases the trading volume of options, futures and the underlying stocks, and occasionally increases volatility of prices of related securities.

- Another perspective of the volume metrics is the separation by region analysis:

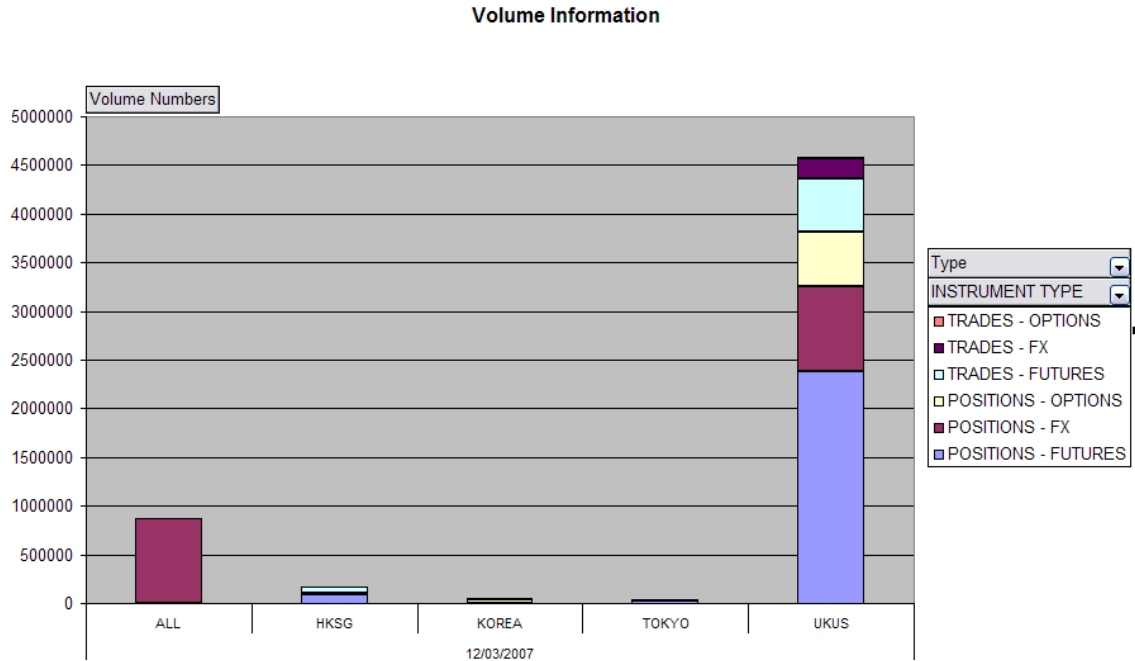


Figure 13: Volume Information by Day

Separated by date, a dynamic selection of trading instruments can be chosen, so the difference in the portion of the volume can be seen.

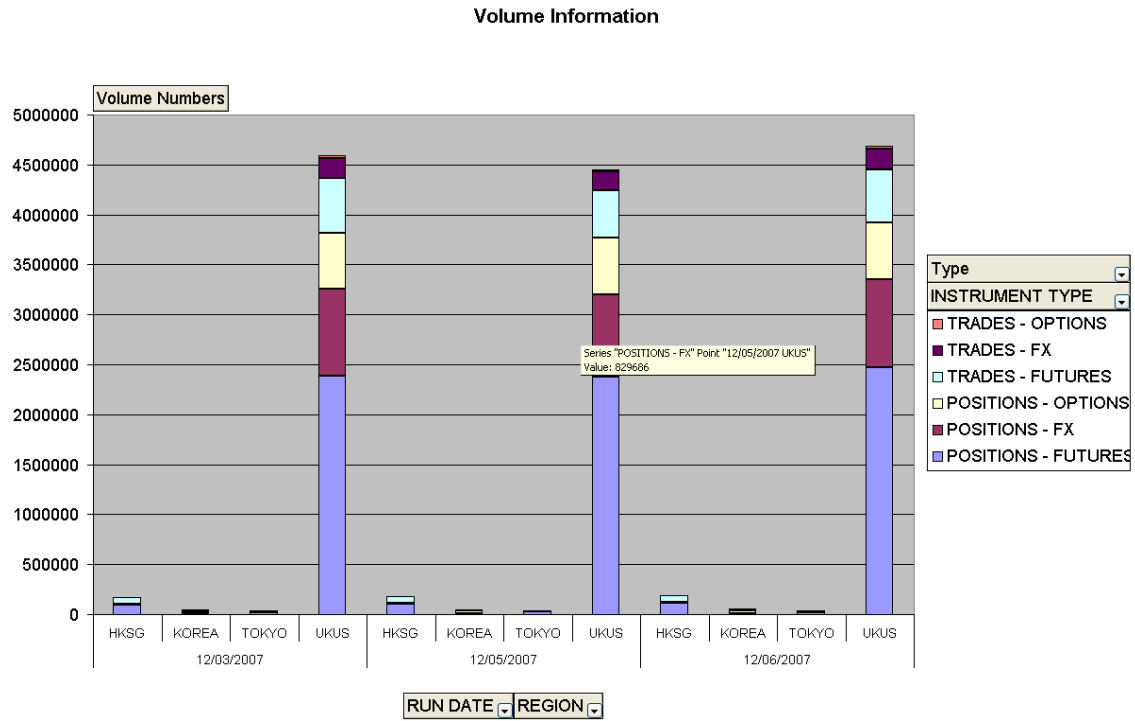


Figure 14: Volume Information by Region over Time

It is visible from Fig 13 and Fig 14 that the US/UK market has the biggest part of Options and Futures, compared to the Tokyo, Korea and Hon Kong's markets.

Another interesting part is the separation of the volume by instruments type within a specific region, as can be seen of the above figure.

4.2 FOFX Runtime Analysis

Moving on to our primary goal - namely analyzing the performance of the FOFX jobs, we created several tools, which are useful for that task:

1) *FOFX Key Jobs Runtime Performance graph:*

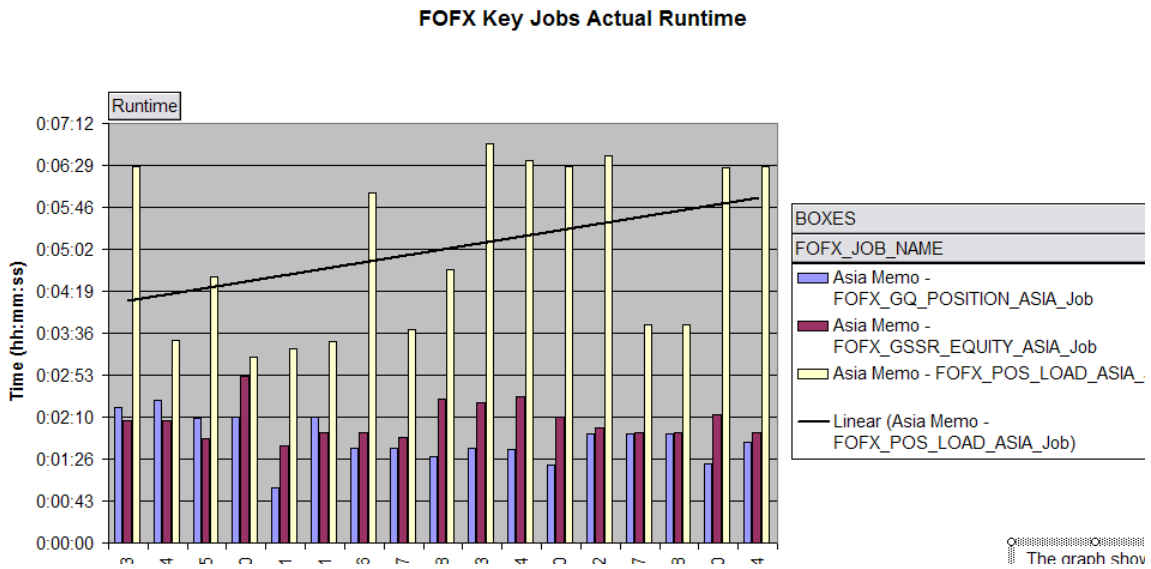


Figure 15: Runtime Performance

This is a two dimensional dynamic graph, that shows the runtime of the selected FOFX processes. The user should select a FOFX Box he wants to monitor and then the specific

Jobs within the Box and the dynamic graph will show the runtime of the selected items, sorted by date. Then a trend line can be specified for the particular job to identify the trend of the runtime.

As you can see from the picture, the linear regression line shows that the FOFX Asia Memo is growing in runtime as the time goes on.

This trend line is confirmed also by our next measurement tool:

2) *FOFX Key Box Runtime Performance Graph:*

FOFX Key Boxes Actual Runtime

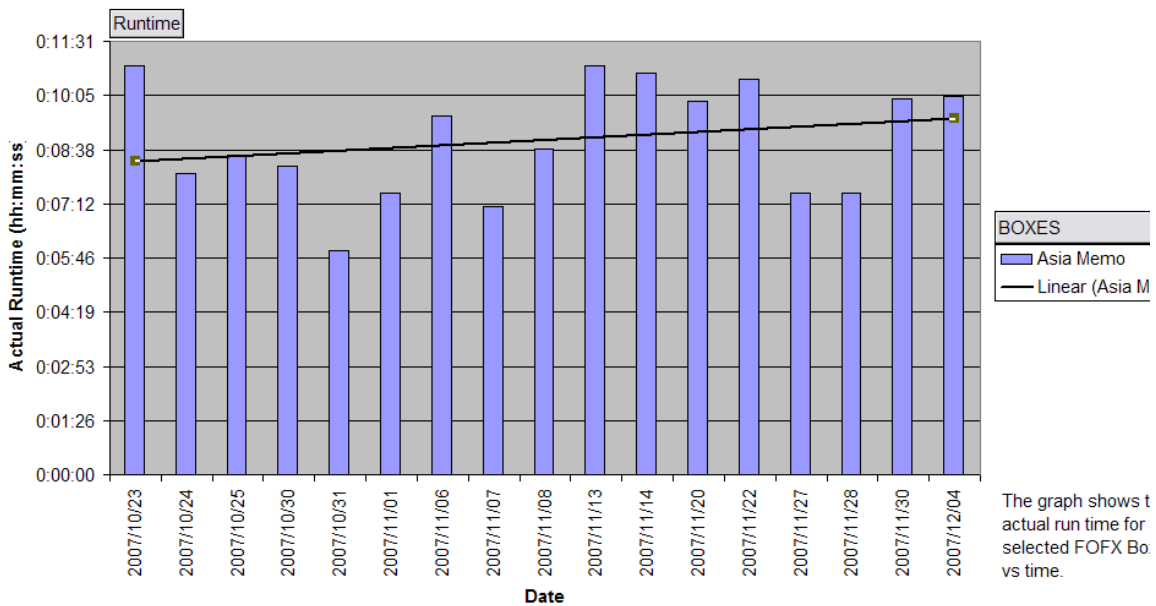


Figure 16: Box Runtime

This tool provides monitoring of the FOFX key Box processes. From the drop-down menu on the right side user can specify which Box process should be monitored and the graph will show the corresponding runtime separated by date.

Again we can see the same trend for the Asia Memo Box. On the other hand, multiple Boxes can be specified, so the runtime of sum of the boxes can be evaluated.

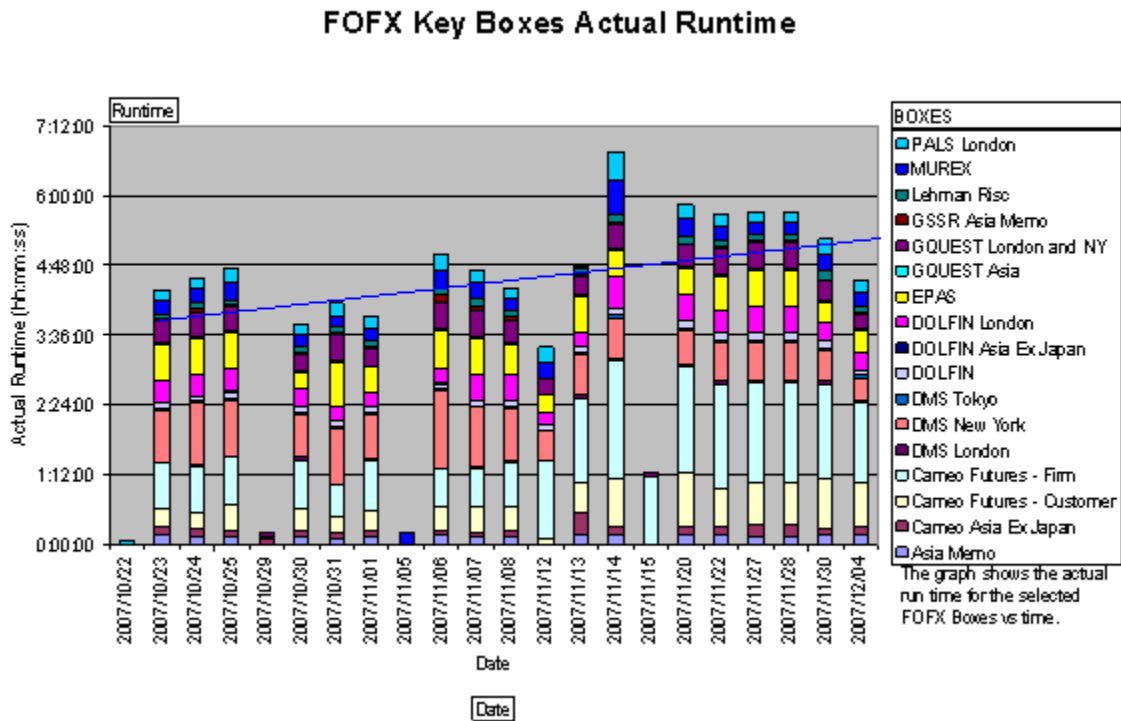


Figure 17: Box Runtime

The runtime information for boxes is not stored in the database. That is why we follow a three step process to obtain that information on database level, using combination of data tables, mapping table and a view:



Figure 18: Three Step Process

3) Job Run Time with Respect to the Average Run Time Analysis

Having built several graphic tools, we were looking for different type of representation of the data, which can give a different perspective of the information.

We decided to use Table view of the data with conditional formatting. Our idea is to compare the runtime of user specified processes with respect to the average runtime.

Job Run Time with Respect to the Average Run Time Analysis			
	Average Runtime	Average Runtime	Average Runtime
	0:08:24	0:01:42	0:02:00
JOB RUN TIME Table	FOFX_JOB_NAME		
DATE	FOFX_ACV_LOAD_CUST_Job	FOFX_FTP_CAMEOACV_CUST_Job	FOFX_FTP_CAMEOACV_REST_Job
2007/12/06	0:08:03	0:01:37	0:01:30
2007/12/05	0:08:58	0:01:21	0:01:31
2007/12/04	0:08:58	0:01:21	0:01:31
2007/11/30	0:09:26	0:01:49	0:01:34
2007/11/28	0:10:15	0:01:47	0:02:47
2007/11/27	0:10:15	0:01:47	0:02:47
2007/11/22	0:09:06	0:01:37	0:01:36
2007/11/20	0:13:45	0:02:00	0:02:05
2007/11/14	0:09:39	0:01:58	0:01:46
2007/11/13	0:06:53	0:01:34	0:01:38
2007/11/12		0:01:52	0:03:20
2007/11/08	0:09:23		
2007/11/07	0:06:23		
2007/11/06	0:05:08		
2007/11/01	0:08:21		
2007/10/31	0:05:40		
2007/10/30	0:08:26		
2007/10/25	0:06:34		
2007/10/24	0:05:58		
2007/10/23	0:08:20		
Average Run Time	0:08:24	0:01:42	0:02:00

Figure 19: Run Time Analysis using Average Run Time

If the cell turns out to be red – it means that the run time on this date was more than the average run time. Otherwise, the color is green.

Also, the selection of the date and Job processes are user defined by a drop down menu.

This is a powerful tool for identifying periods of time, in which the FOFX processes take more time. On the other hand, since this is a comparison with a static average as expected, almost half of the values are colored in red, which shows the ranges where our processes were more costly in time.

However, decisions based on the average run time are not always correct. (Especially when we have many data points)

We found the following different solutions to avoiding that:

- Usage of *Standard Deviation*

- Usage of *Moving Average*

4) *Job Run Time with Respect to Floating Standard Deviation Time Analysis*

Later, based on the previous idea, a more sophisticated version was born. This time our team decided to measure the average time and in addition to calculate the standard deviation of the selected processes.

Job Run Time with Respect to Floating Standard Deviation Time Analysis				
	Average Runtime	Average Runtime	Average Runtime	Average Runtime
	0:08:24	0:01:42	0:02:00	0:00:03
Number of Deviations	Standard Deviation	Standard Deviation	Standard Deviation	Standard Deviation
1	0.0014091767	0.0001541964	0.0004535313	0.0000081094
	Cut-off Line	Cut-off Line	Cut-off Line	Cut-off Line
	0:10:25	0:01:55	0:02:40	0:00:04
JOB RUN TIME Table	FOFX_JOB_NAME			
DATE	FOFX_ACV_LOAD_CUST Job	FOFX_FTP_CAMEOACV_CUST Job	FOFX_FTP_CAMEOACV_REST Job	FOFX_FTP_CAMEOPME_CUST Job
2007/12/06	0:08:03	0:01:37	0:01:30	0:00:03
2007/12/05	0:08:58	0:01:21	0:01:31	0:00:03
2007/12/04	0:08:58	0:01:21	0:01:31	0:00:03
2007/11/30	0:09:26	0:01:49	0:01:34	0:00:03
2007/11/28	0:10:15	0:01:47	0:02:47	0:00:04
2007/11/27	0:10:15	0:01:47	0:02:47	0:00:04
2007/11/22	0:09:06	0:01:37	0:01:35	0:00:02
2007/11/20	0:13:45	0:02:00	0:02:05	0:00:02
2007/11/14	0:09:39	0:01:58	0:01:46	0:00:03
2007/11/13	0:06:53	0:01:34	0:01:38	0:00:03
2007/11/12		0:01:52	0:03:20	0:00:02
2007/11/08	0:09:23			
2007/11/07	0:08:23			
2007/11/06	0:05:08			
2007/11/01	0:08:21			
2007/10/31	0:05:40			
2007/10/30	0:08:26			
2007/10/25	0:08:34			
2007/10/24	0:05:58			
2007/10/23	0:08:20			

Figure 20: Run Time Analysis using Standard Deviation

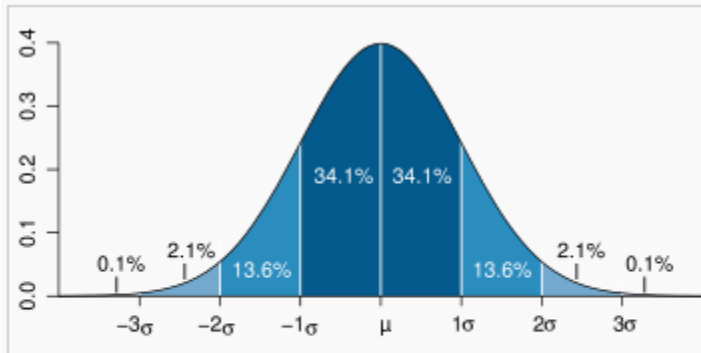
That is how we produced the

$$\text{Cut-off line} = \text{Average Runtime} + (\text{number of Std Deviations}) * (\text{Standard Deviation})$$

Again we used conditional formatting, based on the same principle – red if the value is above the cut-off value and green if it is below.

The whole table is dynamic – different dates can be specified from the user from a drop down menu and different processes can be selected with the same technique.

The number of standard deviations used in the cut-off calculation is also specified by the user in the “Number of Deviations” field.



Dark blue is less than one standard deviation from the mean. For the normal distribution, this accounts for 68.27 % of the set; while two standard deviations from the mean (medium and dark blue) account for 95.45 %; and three standard deviations (light, medium, and dark blue) account for 99.73 %.

and three standard deviations show about 99.73% in a normal distribution.

Figure 21: Standard Deviation Graph

This tool is partially useful to find abnormal processes, which required much more time than usual. This information is very easy captured, thanks to the conditional formatting – everything that is red shows high run time.

5) Job Run Time with Respect to Moving Average Time Analysis

In our previous heat map tables, we used the concept of total average value. This approach is good for finding abnormal values and in particular late run processes. However, due to the fact that all values are used, it is not very powerful in showing trends.

That is why we created different type of heat maps that are based on *moving average*.

In statistics, a *moving average* or *rolling average* is one of a family of similar techniques used to analyze time series data. It is applied in finance and especially in technical analysis. It can also be used as a generic smoothing operation, in which case the raw data need not be a time series.

A moving average series can be calculated for any time series. In finance it is most often applied to stock prices, returns or trading volumes. Moving averages are used to smooth out short-term fluctuations, thus highlighting longer-term trends or cycles. The threshold between short-term and long-term depends on the application, and the parameters of the moving average will be set accordingly.

We used similar conditional formatting and user interface as for the other heat map tools:

Job Run Time with Respect to Moving Average Time Analysis			
	Average Runtime	Average Runtime	Average Runtime
	0:19:04	0:00:25	
Moving Average Days			
5			
	Cut-off Line	Cut-off Line	Cut-off Line
	0:19:04	0:00:25	
JOB RUN TIME Table	FOFX_JOB_NAME		
DATE	FOFX_ACV_LOAD Job	FOFX_BAL_UPD FROM PNS Job	
2007/12/06	0:24:08	0:00:23	
2007/12/05	0:15:18	0:00:19	
2007/12/04	0:15:18	0:00:19	
2007/11/30	0:19:15	0:00:28	
2007/11/28	0:21:20	0:00:38	
2007/11/27	0:21:20	0:00:38	
2007/11/22	0:16:52	0:00:22	
2007/11/20	0:21:49	0:00:22	
2007/11/15	0:19:19	0:00:17	
2007/11/14	0:22:11	0:00:25	
2007/11/13	0:13:29	0:00:18	
2007/11/12	0:13:46	0:00:28	
2007/11/08	0:09:35	0:00:14	
2007/11/07	0:13:14	0:00:13	
2007/11/06	0:11:19	0:00:17	
2007/11/01	0:11:47	0:00:18	
2007/10/31	0:09:49	0:00:17	
2007/10/30	0:08:26	0:00:26	
2007/10/25	0:16:51	0:00:37	
2007/10/24	0:16:39	0:00:21	
2007/10/23	0:10:02	0:00:17	

Figure 22: Run Time Analysis using Moving Average

This tool also has functionality of selecting different processes (form a drop-down menu by the user), date and most important – the size of the moving average.

Based on the size of the moving average, the cut-off line measurement is recalculated, and then the page formatting updates the coloring of the sheet.

The tool is used for finding trends. For example if a 20 days moving average is selected , we can see that the last 10 days are red, which indicates that runtime has increased over the last week, so the trend is easily visible.

Another application of that sheet is tracking the efficiency of the implemented changes on a particular job/box. For example: If an update has been made to produce more efficient runtime, then the color code will specify if the update has successfully produced lower run time or not.

Also, a comparison can be made my selecting different sizes for the moving average, which can show how fast is the growth in the last 5 days compared to the growth of let say 30 days.

Since the job processes are part of Boxes, we implemented the same idea, but with the Box runtime, to monitor the key FOFX 17 Box processes:

Box Run Time with Respect to Moving Average Time Analysis						
	Average Runtime	Average Runtime	Average Runtime	Average Runtime	Average Runtime	Average Runtime
	0:04:06	0:15:26	0:13:13			
Moving Average						
20						
	Cut-off Line	Cut-off Line	Cut-off Line	Cut-off Line	Cut-off Line	Cut-off Line
	0:04:06	0:15:26	0:13:13			
Runtime Table	BOXES					
DATE	Lehman Risc	MUREX	PALS	London		
2007/12/06	0.002916667	0.009027778	0.007766204			
2007/12/05	0.00287037	0.010474537	0.009259259			
2007/12/04	0.00287037	0.010474537	0.009259259			
2007/11/30	0.004108796	0.01224537	0.010578704			
2007/11/28	0.002847222	0.008506944	0.007361111			
2007/11/27	0.002847222	0.008506944	0.007361111			
2007/11/22	0.002847222	0.010185185	0.008796296			
2007/11/20	0.002719907	0.013321759	0.010543981			
2007/11/14	0.004583333	0.025173611	0.019467593			
2007/11/13	0.002835648	0.001435185	0.001006944			
2007/11/12	3.47222E-05	0.012662037	0.010763889			
2007/11/08	0.002708333	0.008506944	0.008078704			
2007/11/07	0.00306713	0.012210648	0.009768519			
2007/11/06	0.002847222	0.013356481	0.011099537			
2007/11/05		0.008935185				
2007/11/01	0.003703704	0.009178241	0.008090278			
2007/10/31	0.002453704	0.008680556	0.00912037			
2007/10/30	0.002847222	0.00869213	0.007696759			
2007/10/25	0.002650463	0.012916667	0.010277778			
2007/10/24	0.002361111	0.009791667	0.008136574			
2007/10/23	0.002546296	0.009016204	0.008263889			
2007/10/22			0.001990741			

Figure 23: Box Run Time Analysis using Moving Average

This is the most powerful FOFX tracking system, since it monitors the essential processes of the FOFX. By easily observing the trends and the breakouts of the key processes, predictions and adequate decisions can be made for the future.

4.3 RISC Customer Batch Runtime Analysis

Another project component was optimizing and automating the RISC Customer Batch Runtime Analysis. It is used by the Lehman Brothers' Operations as a metric that indicates our daily SLA toward statement delivery.

It is a graph, showing the runtime of the RISC Customer Stream, on each day, with the batch's start and end time, compared to the SLA.

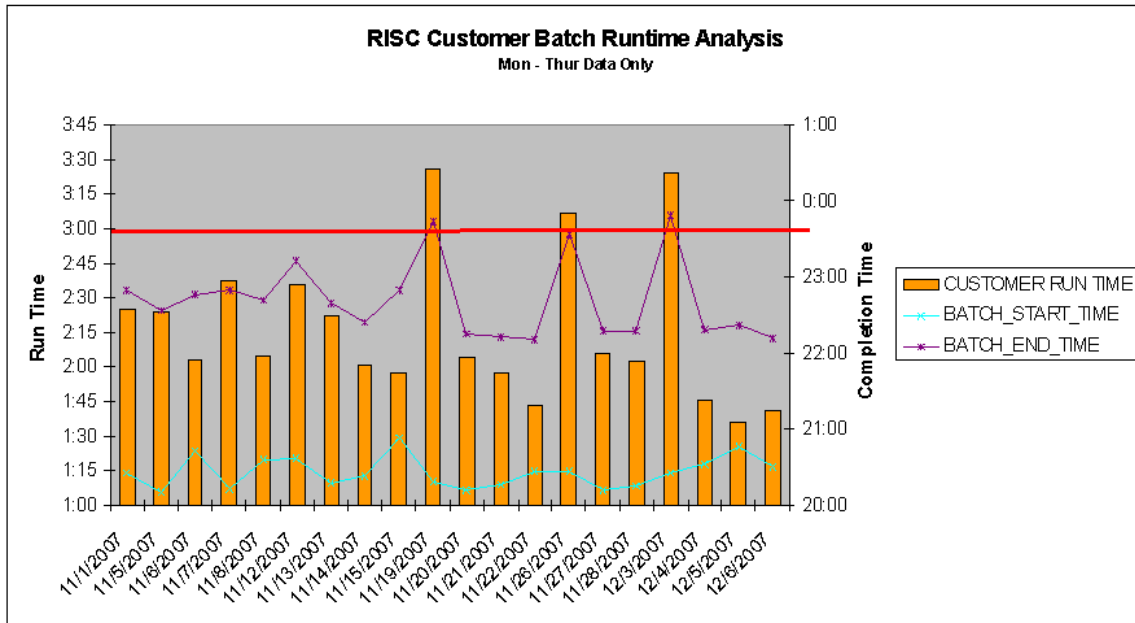


Figure 24: RISC Customer Batch Runtime Analysis

Up until now this analysis was done by hand, without dynamic and automated data extracting.

Now, our RISC Customer Batch Runtime Analysis tool uses dynamic direct connection to an Oracle Database, so each time the sheet is open, the data is updated. This is done with the help of a select statement from the RISC_DAILY_BATCH_METRICS_VW view table, which we created.

Statement Delivery					
RUN DATE	BATCH NAME	BATCH START TIME	BATCH END TIME	CUSTOMER RUN TIME	
11/1/2007	CUSTOMER	20:25	22:50		2:25
11/5/2007	CUSTOMER	20:10	22:34		2:24
11/6/2007	CUSTOMER	20:43	22:46		2:03
11/7/2007	CUSTOMER	20:13	22:50		2:37
11/8/2007	CUSTOMER	20:36	22:41		2:05
11/12/2007	CUSTOMER	20:37	23:13		2:36
11/13/2007	CUSTOMER	20:17	22:39		2:22
11/14/2007	CUSTOMER	20:23	22:24		2:01
11/15/2007	CUSTOMER	20:53	22:50		1:57
11/19/2007	CUSTOMER	20:18	23:44		3:26
11/20/2007	CUSTOMER	20:11	22:15		2:04
11/21/2007	CUSTOMER	20:16	22:13		1:57
11/22/2007	CUSTOMER	20:27	22:10		1:43
11/26/2007	CUSTOMER	20:27	23:34		2:04
11/27/2007	CUSTOMER	20:11	22:17		3:07
11/28/2007	CUSTOMER	20:15	22:17		2:06
12/3/2007	CUSTOMER	20:25	23:49		2:02
12/4/2007	CUSTOMER	20:32	22:18		3:24
12/5/2007	CUSTOMER	20:46	22:22		
12/6/2007	CUSTOMER	20:30	22:11		

However, due to the limitation of the Oracle data table and Excel capabilities, the produced data is not in the desired format and graphs cannot be updated.

Here we used macro VBA programming to

Figure 25: Statement Delivery

produce the button “Update Graphs” on the sheet, which does the formatting, calculation and charts the graph:

Statement Delivery					
RUN DATE	BATCH NAME	BATCH START TIME	BATCH END TIME	CUSTOMER RUN TIME	
11/1/2007	CUSTOMER	8:25:00 PM	10:50 PM		2:25
11/5/2007	CUSTOMER	8:10:00 PM	10:34 PM		2:24
11/6/2007	CUSTOMER	8:43:00 PM	10:46 PM		2:03
11/7/2007	CUSTOMER	8:13:00 PM	10:50 PM		2:37
11/8/2007	CUSTOMER	8:36:00 PM	10:41 PM		2:05
11/12/2007	CUSTOMER	8:37:00 PM	11:13 PM		2:36
11/13/2007	CUSTOMER	8:17:00 PM	10:39 PM		2:22
11/14/2007	CUSTOMER	8:23:00 PM	10:24 PM		2:01
11/15/2007	CUSTOMER	8:53:00 PM	10:50 PM		1:57
11/19/2007	CUSTOMER	8:18:00 PM	11:44 PM		3:26
11/20/2007	CUSTOMER	8:11:00 PM	10:15 PM		2:04
11/21/2007	CUSTOMER	8:16:00 PM	10:13 PM		1:57
11/22/2007	CUSTOMER	8:27:00 PM	10:10 PM		1:43
11/26/2007	CUSTOMER	8:27:00 PM	11:34 PM		3:07
11/27/2007	CUSTOMER	8:11:00 PM	10:17 PM		2:06
11/28/2007	CUSTOMER	8:15:00 PM	10:17 PM		2:02
12/3/2007	CUSTOMER	8:25:00 PM	11:49 PM		3:24
12/4/2007	CUSTOMER	8:32:00 PM	10:18 PM		1:46
12/5/2007	CUSTOMER	8:46:00 PM	10:22 PM		1:36
12/6/2007	CUSTOMER	8:30:00 PM	10:11 PM		1:41

Figure 26: Statement Delivery

And then automatically produce the needed graph analysis:

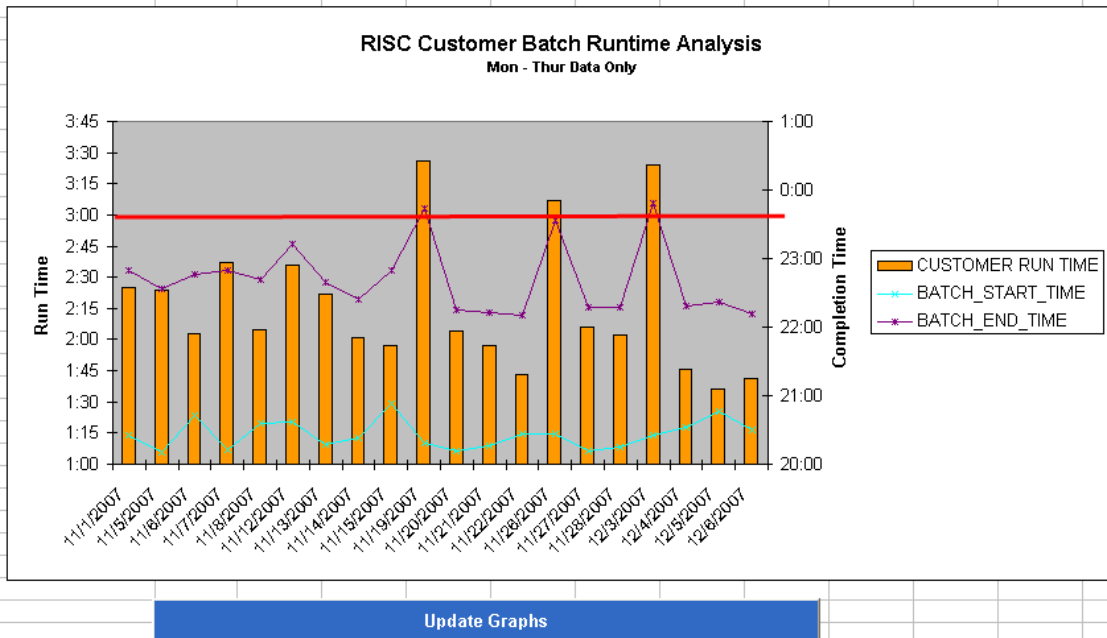


Figure 27: RISC Customer Batch Runtime Analysis

4.4 Ticket Management System

The result of this part of the project was an Excel file with five sheets containing graphs and tables. These sheets were: Daily, Weekly, October 2007, November 2007, and All Tickets. There were five basic charts and four basic graphs that the sheets had, although the Daily sheet had one less chart and one less graph while the All Tickets sheet had one more chart and one more graph.

4.4.1 Tickets by Application/Infrastructure

Tickets By Application/Infrastructure	No. of Tickets	Increase (Decrease) in Tickets	% Increase (Decrease)	Asia	India	Ldn	New York
RISC/FOFX	173			2	45	56	70
Cameo	131			4	3	27	96
CitiNetworks (TRAM)	10			0	4	4	2
Autosys	5			0	0	3	2
Lehman Live	0			0	0	0	0
RISC FX Cash Journal System	2			0	0	2	0
RISC FX Ledger Balance System	1			0	0	1	0
Risc FX Trade Browse System	3			0	0	1	2
CLS	2			0	1	0	1
SMTP	0			0	0	0	0
UDB	0			0	0	0	0
Oracle	0			0	0	0	0
Unix Server	0			0	0	0	0
VVA Systems	2			0	0	0	2
Totals	329	0	0	6	53	94	175

Figure 28: Tickets by Application/Infrastructure Table

This is the Tickets by Application/Infrastructure table. A quick examination shows that the vast majority of tickets come for the RISC/FOFX and Cameo applications. You can also see that New York has many more tickets than the others. This chart allows the user to notice things like the proportion of Indian tickets which is 45 to 3 for RISC/FOFX and Cameo, much different than the proportion for London or New York.

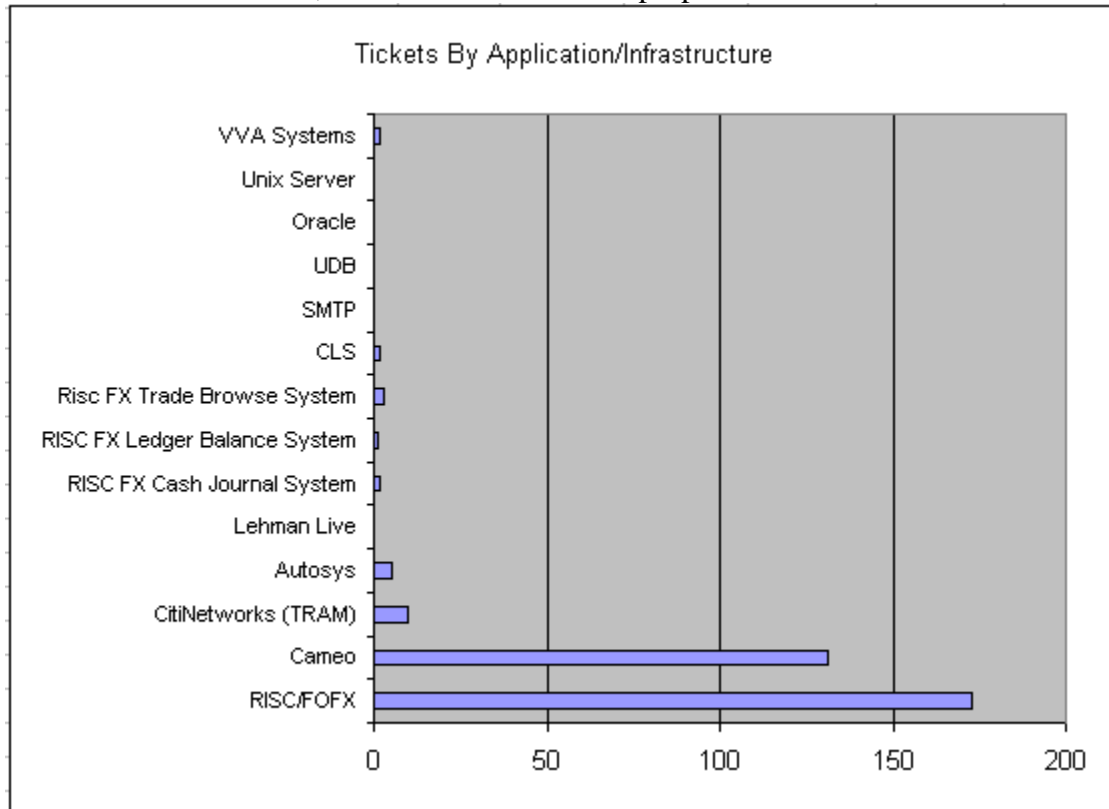


Figure 29: Tickets by Application/Infrastructure Graph

The graph for the Tickets by Application/Infrastructure table makes it easier to see the top two applications but doesn't provide the detailed breakdown by region.

4.4.2 User Priority Level

User Priority level	No. of Tickets
Urgent	4
High	32
Med	24
Low	82
Totals	142

Figure 30: User Priority Table

This is a simple table showing the breakdown of tickets by priority level. Users would be watching for unusually large numbers of urgent tickets.

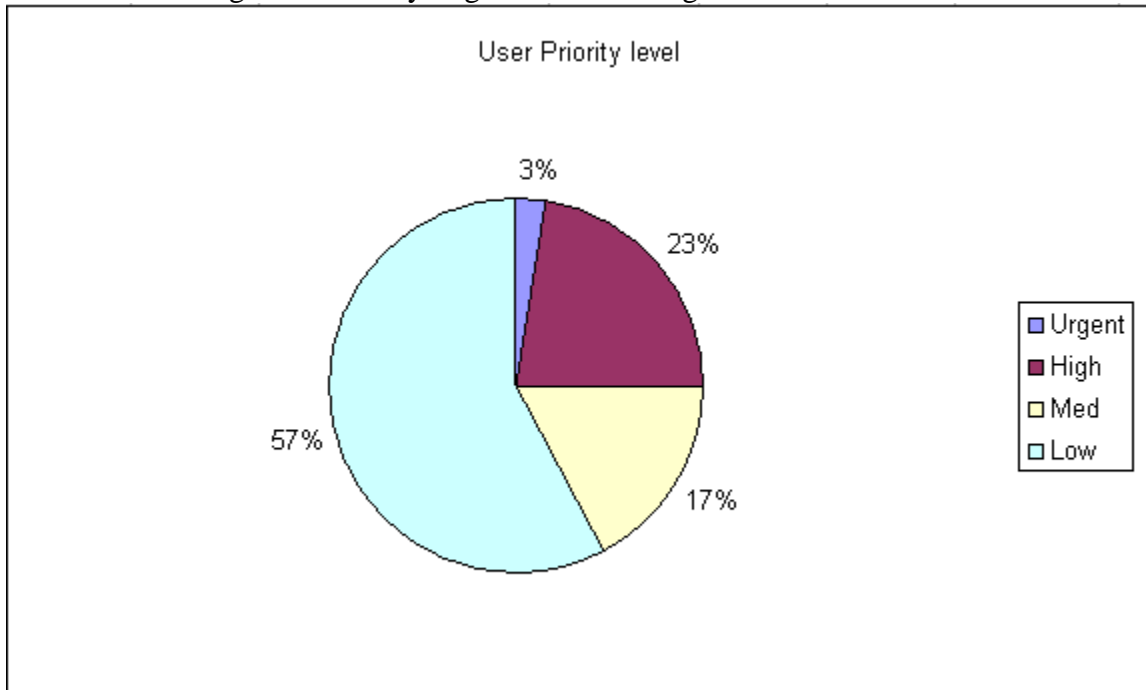


Figure 31: User Priority Graph

The graph from the user priority table allows the user to quickly see the percentage of tickets that are urgent.

4.4.3 Aging Timeline

Aging Timeline (Open Tickets)	No. of Tickets	Urgent	High	Percentage
0-5	16	0	2	24%
6-10	28	0	9	41%
11-20	11	0	1	16%
21-30	13	1	2	19%
31-90				
91-180				

Figure 32: Aging Timeline Table

The idea here is that tickets should be resolved as quickly as possible. A user looking at this chart would see that there is an urgent ticket between 21 and 30 days old and want to know why that was.

4.4.4 Tickets by Type

Tickets By Type	No. of Tickets	Increase (Decrease) in Tickets	% Increase (Decrease)	Asia	India	Ldn	New York
Account Management	0			0	0	0	0
Business Request	187			5	8	65	109
Content Provider Issue	0			0	0	0	0
Enhancement	6			0	0	1	5
How-To	0			0	0	0	0
Missing	1			0	0	0	1
Outage	16			0	5	7	4
Change Job Status	5			0	0	3	2
Error	0			0	0	0	0
Information	0			0	0	0	0
Prod Support	0			0	0	0	0
Replication Problem	0			0	0	0	0
Issue	100			1	33	16	50
Bug	1			0	0	0	1
Question	13			0	7	2	4
Totals	329			6	53	94	176

Figure 33: Tickets by Type Table

This table shows the breakdown of tickets by type and region. Issues and Business requests stand out as the biggest numbers. You can see that although India has much fewer tickets than either London or New York, they have more questions than both combined. India also has far fewer business requests proportionally and more issues proportionally. These are the kind of things that users can get from this table.

4.4.5 Tickets by Status

Tickets by Status	No. of Tickets	Increase (Decrease) in Tickets	% Increase (Decrease)	Asia	India	Ldn	New York
Unassigned	0			0	0	0	0
Assigned	39			1	3	11	24
Work In Progress	20			0	0	7	13
Pending	9			0	0	1	8
Closed	261			5	50	75	131
Totals	329			6	53	94	176

Figure 34: Tickets by Status Table

This table shows tickets by status with relation to region. It makes it easy to see if some regions are having a hard time or a harder time than the others in closing its tickets. In this table the “closed” and “resolved” rows have been merged.

4.4.6 Tickets by User

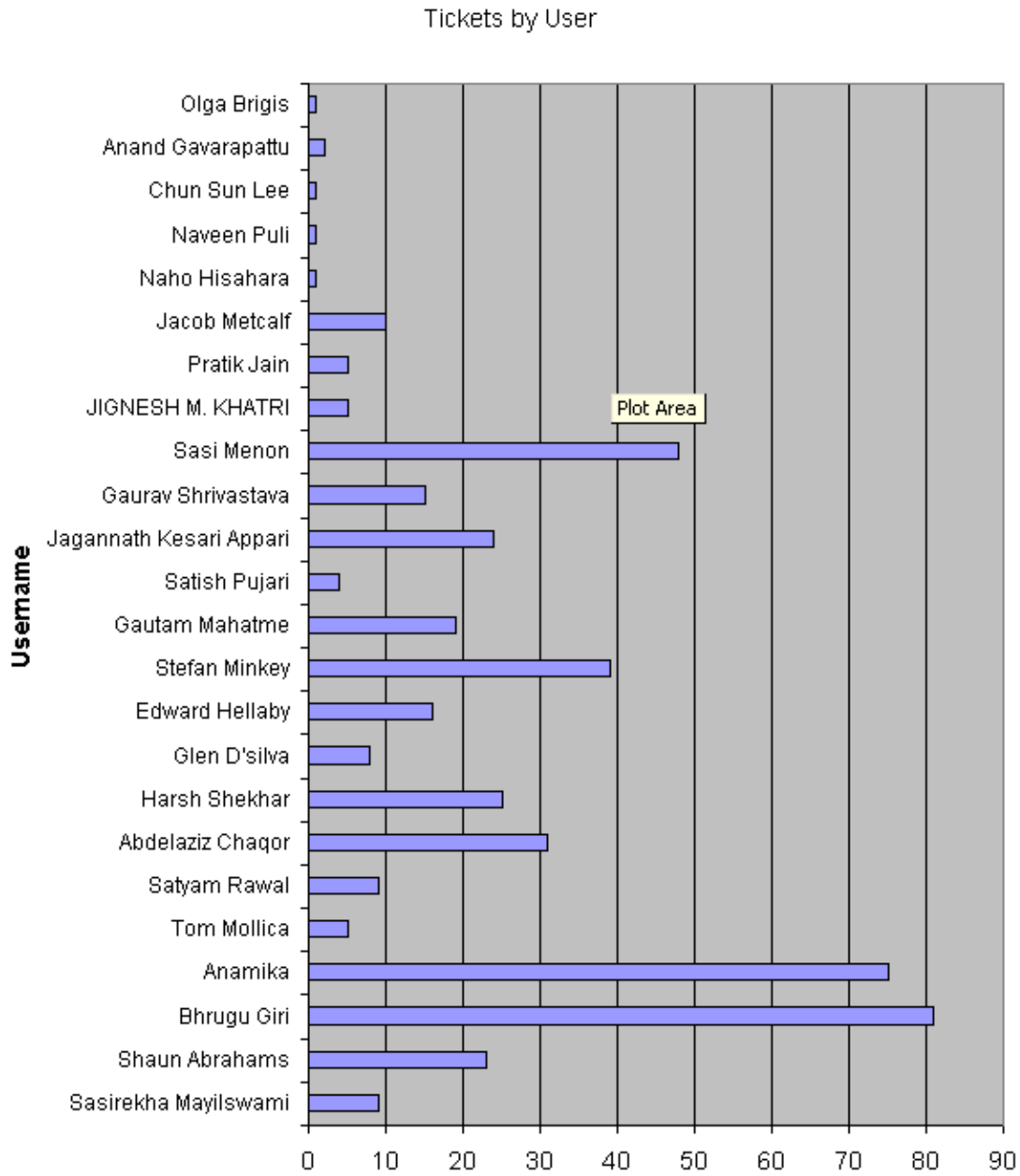


Figure 35: Tickets by User Graph

Simple graph, showing how many tickets each user has resolved.

5. Future Recommendations

During the period of our work at Lehman Brothers we came across some ideas that we couldn't implement, but which we believe are interesting and worth mentioning. They may be developed or implemented as future projects, or as a continuation of the WPI projects at Lehman Brothers.

Generally, our recommendations can be separated into these categories:

- Further Analytics
- Database access/connection
- Data representation

5.1 Further Analysis

The following data analysis idea came to mind from well-known trading analysis tools. However, they are still applicable in capturing trends for the batch runtimes or SAL proximity and that is why are definitely worth implementing:

- Volatility Channels
- Bollinger Breakouts Analysis
- Donchian Trends
- Comparison between Moving Averages

5.1.1 Volatility Channels:

In trading, these are measurements, built by adding a specific amount of price to a moving average that is based on a measure of market volatility - typically a fixed amount or standard deviations.

For example:

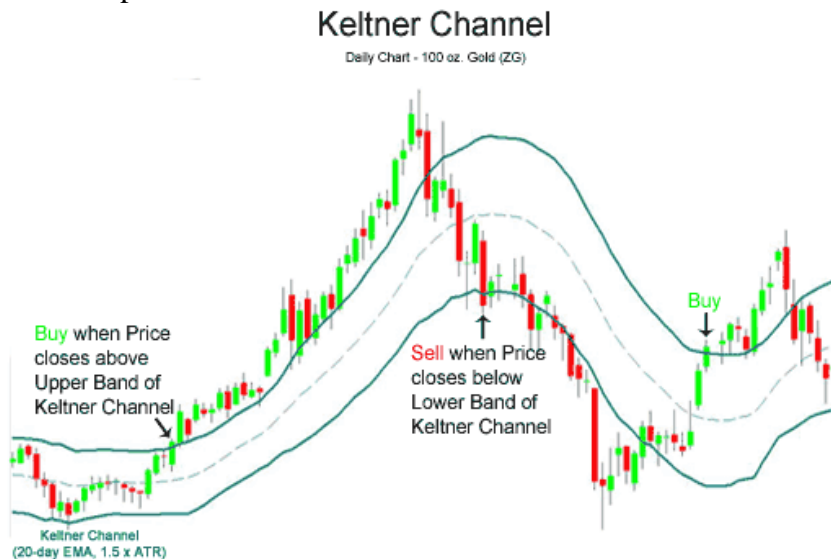


Figure 36: Keltner Channel

The idea is to plot a dynamic moving average for a particular period of time (Dynamic, since we have different value of the moving average for each day). Then by using a fixed amount for window size, both the upper bound and the lower bound lines are created. This produces a volatility channel. Once the plotting is done, the idea is to plot on the same graph the data on which the moving average is calculated.

In the example of Figure 36, a volatility channel based on the price of gold has been created, and then the price of gold has been plotted as a color line.

In general volatility channels are used for finding trends. Although they are not extremely powerful tool in trading for known reasons, they can be very useful for analyzing trends for the batch runtimes or SAL proximity

In our case, it seems reasonable since we already measure the moving average of the key FOFX job and boxes, to capture that information and produce volatility channels of this type, based on run time data:

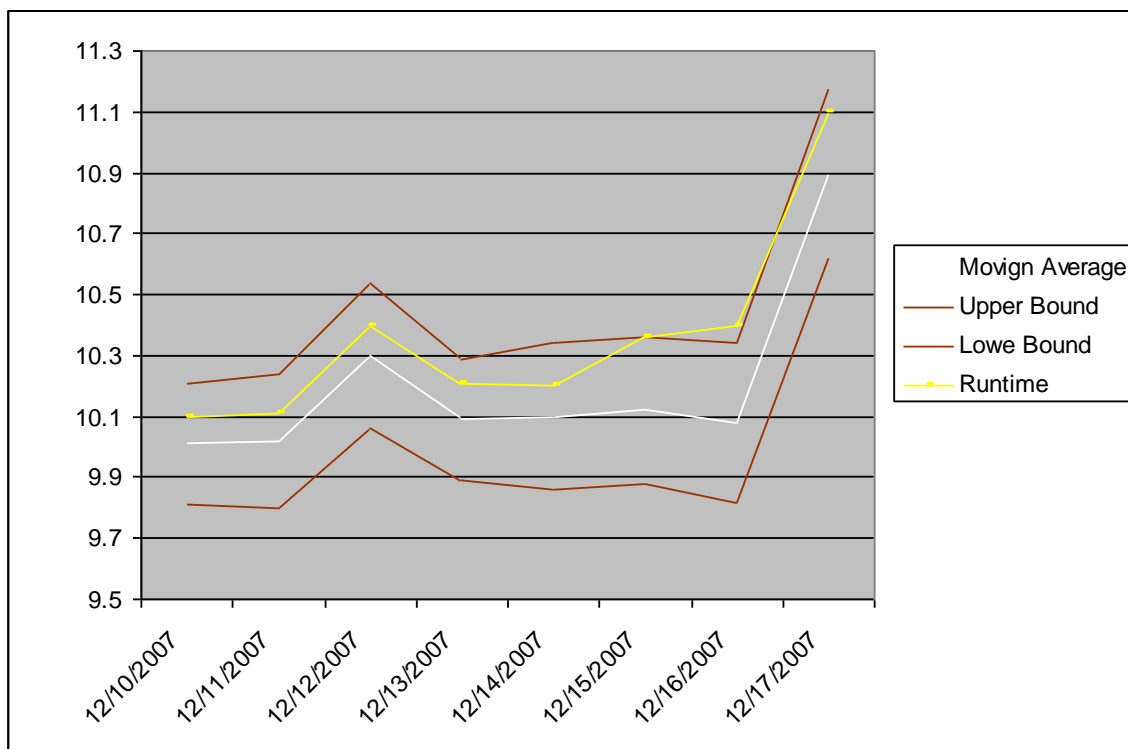


Figure 37: Volatility Channel Graph

As you can see from the picture the red lines identify the volatility channel.

By using this type of analysis we can identify mainly two things:

- If there is a trend (if we are getting closer to the boundary line)
- If there was a breakout – if we incidentally passed the breakout line – this can indicate that on this particular day there was a single event like a server was down, which was the reason why our runtime was longer.

We can also try to improve this volatility channel, by using a specific one:

5.1.2 Bollinger breakout Analysis

Bollinger Analysis is a volatility channel that uses standard deviations for window size of the channel.

Usually they use two standard deviations and we highly recommended the usage of this type of volatility channel, since the volume and the runtime of the FOFX processes have increasing trend. (Therefore fixed number for windows size will not produce desired results)

Here is an example of a Bollinger Band for the S&P 500 Index:



Figure 38: Bollinger Bands

Using standard deviations for the channel will make it more flexible and more dynamic. Then specific breakouts can be captured much easier than with the normal one.

5.1.3 Donchian Trends

The **Donchian channel** is an indicator used in market trading. It is formed by taking the highest high of the daily maxima and the lowest low of the daily minima of the last n days, then marking the area between those values on a chart. The Donchian channel is a useful indicator for seeing the volatility of a market price. If a price is stable the Donchian channel will be relatively narrow. If the price fluctuates a lot the Donchian channel will be wider. Its primary use, however, is for providing signals for long and short positions. If a security trades above its highest n day high, then a long is established. If it trades below its lowest n day low, then a short is established.

In our case, we can use the Donchian channel to easily capture the trends in the system. If the channel is in steady state (in particular the upper bound does not move) then there is no increasing trend.

We experienced some inefficiency with the database connection:

5.1.4 Comparison Between Moving Averages

The idea of graphical representation of the moving averages with respect to time can also be applied in the runtime processes analysis. It requires plotting different averages to a

time line graph, and comparing in to the graph of the run time, for the same period of time. This will produce a graph with comparison between the run time and different moving averages graph, which can be used to identify the power of already observed trend.

5.2 Database connection via Microsoft Excel:

Currently, the access to the Lehman Oracle server is done through Excel's ODBC for Oracle Data source and the importing of data is done through Excel's build-in functions for importing new Database queries.

However, this method lacks functionality and flexibility in formatting and capturing data. The imported data is not always in the format which is required, although a strict formatting has been specified for the cell. Also, using direct importing requires using additional dynamic ranges in order of additional data manipulation.

We recommend the usage of Database connection via Excel VBA. One particular example is using DAO object to connect to the Oracle database and execution of a query within a macro code. This can give more flexibility in obtaining the data in a dictionary or an array or list, so further modification or calculation with it can be done easier.

5.3 Data Representation

Currently the implemented tools are represented as links on the Lehman Brothers' Operation Technology Group page. We believe, we could achieve a better accessibility if the tools are directly implemented within a web page. Some solutions for that might be:

- Usage of imbedded objects for inserting Excel files within a Web Page
- Using third party software products to represent data directly on a web page

6. References

John C. Hull, *Options, Futures and Other Derivatives*, Prentice Hall; 6 edition (June 20, 2005)

Steve E. Shreve, *Stochastic Calculus for Finance I*, Springer

<http://wikipedia.org>

Martin Baxter, *Financial Calculus*, Cambridge University Press

Little, Jeffrey & Rhodes, Lucian. (2004), *Understanding Wall Street*, 4th edition, McGraw-Hill, USA

Malkiel, Burton. (2005), *A Random Walk Down Wall Street. The Time-Tested Strategy for Successful Investing*, 8th edition, W. W. Norton, USA

Marshall, John & Ellis, M. () *Investment Banking and Brokerage*, McGraw-Hill, USA.

Clifford J. Sherry and Jason W. Sherry, *The Mathematics of Technical Analysis: Applying Statistics to Trading Stocks, Options and Futures*

Curtis Faith, *Way of the Turtle*, McGraw-Hill (2007)

www.lehman.com

Solomon Kullback, *Information Theory and Statistics*, McGraw-Hill

Weiss, David. (1993), *After the Trade is Made*, New York Institute of Finance, USA.

Peter Dalgaard, *Introduction to Statistics with R*, Springer (2006)