Application of High Performance Computing Methods to the Large-Scale Portfolio Simulation for Asset Liability Management

Takuma Matsuo, Hideyuki Torii / Numerical Technologies Incorporated

The Basel II framework for regulatory capital sets the size limit of risk exposure.

The 2007-08 financial crisis spotted counterparty credit risk (CCR) as well as traditional market and credit risk, which persuaded the Basel Committee of Banking Supervision (BCBS) to issue new proposals concerning the regulatory framework.

Asset Liability Management (ALM) in the next generation should maximize profits within this limitation. To avoid potential failures of the current ALM process, the following must be achieved: 1) realistic accounting by each transaction, 2) market rates and corporate credit rating scenario paths proceed with time, 3) evaluation along each scenario for at least several years at daily resolution. Such calculations require large-scale simulations which have not been realistic so far, because they are computationally expensive.

This research contribution showed that the latest high performance computing (HPC) makes such simulations feasible, even for the largest banks in today's world.

Keywords

- ALM (Asset Liability Management)
- VaR (Value at Risk)
- EaR (Earnings at Risk)
- EDF (Expected Default Frequency)
- TP (Transfer Pricing)
- Basel II
- CCR (counterparty credit risk)
- CVA (Credit Valuation Adjustment)

Background and Purpose/Goal

Various risk management methods used nowadays in financial institutions are similar to the ancient Indian story called “the blind men and the elephant”.

The blind men had trouble guessing what the object was because each one focused on a different part of the elephant – so are we when it comes to our portfolios.

Looking at the elephant holistically neither costs too much nor is it impossible. This idea is almost too well-known so that regulatory authorities and even main-stream economists have never thought doubting it.
In reality, you can see several blind men (=risk management methods below) flourishing in different risk management areas.

Fig.1 [Blind monks examining an elephant [Source: Wikipedia]]

1. Maturity Ladder
   Adjust the hold positions into the balance along the maturity ladder with one dimension. Apply full term for maturity and notional amount for balance. This is enough for the risk management in funding, which is the most basic accounting operation for financial institutions. It basically keeps an eye on liquidity risk; however, it cannot handle any other risks.

2. Static ALM
   Classify the maturity ladder by its interest maturity and asset class and sort it into several charts. It clarifies some points such as interest rate sensitivity and safety.

   Above 2 methods are used in almost all commercial banks and are known among almost all accountants. They also accomplish the basics of BIS regulation. However, they are inadequate for investment banks and major risk management.

3. Dynamic ALM
   In addition to static ALM, dynamic ALM calculates the risk-return by dividing the return by the balance at certain time periods. It enables you to take control of EaR (earning at risk). Furthermore, dynamic ALM simulates several future market scenarios and builds charts for each scenario path. Looking at such parallel worlds with probability leads to expected returns and expected EaR. This method is very powerful and also favored by managers who graduated from business school. Furthermore, it is usually used for bonus calculations. However, different from banking accounts, it doesn’t help trading accounts a lot. Moreover, it is not acceptable for most accountants since mathematics including probability is a topic they have avoided since high-school.

4. Position sheet
   Traders use various versions of converted maturity ladders. For example, a trader dealing with fixed income bonds takes the interest maturity or the duration as maturity axis instead of the treasury maturity, and the present value or PV01 (value change per basis point) as balance.

   An expanded version of this method is widely used for simple bonds and stocks and for complicated derivatives. Furthermore, this method has the clear failure that different instruments cannot be managed together. It means that it satisfies one trader’s risk management but not the whole financial institution’s. Finally, this method concentrates too much on market value to have a time concept so that it even fails to achieve the simple function that method 1 above offers. Traders are aware of this failure, but consider it as a problem of repo desks.

5. VaR (Value at Risk)
   By modeling correlation matrices between each risk factor, unite several position sheets. Despite of exposure complexity, with this method any financial institution can quantify risks by one unitary numerical number, value at risk (VaR). This is Pillar I, a basis of the current Basel II framework. While former VaR was defined as market VaR targeting market risks, current VaR was extended to credit VaR targeting credit risks.

   On the other hand, mark to the model has to apply to commodities which don't include mark to the market, because this method needs market value to function. Historically, this complexity has repeatedly lead to huge mistakes such as LTCM in 1998, Bear Stearns in 2008 and AIG, entangling the finest brains in finance, including even Nobel Prize laureates.

   Moreover, it is quite inadequate for managing liquidity risks and its mathematics is far beyond what accountants normally understand. Consequently nowadays even CFOs tend to consider VaR as an allocated number for Basel Capital.

6. CVA (credit value adjustment)
   As the extended version of current position sheets derivatives traders use, there is a potential exposure. This is a method to simulate based on various future market scenarios and to calculate the present value along every scenario path. To evaluate the value the expected value of the latent profitable side and the confidence interval must be considered. Since a future latent gain is a credit towards counterparty, that counterparty would be asked to pay collateral or the transaction would be suspended if the counterparty did not have enough credit. CVA (credit value adjustment) introduces credit VaR here. Firstly, calculate your and your counterparty's future credit rating at every transaction using the same method as for Credit VaR. Secondly, calculate this up to transaction maturity with small enough intervals.

   The addition of regulatory capital, based on future credit VaR per counterparties resulting from those calculations, has currently been discussed at BCBS. This idea gains support from the derivative industry as a provision for the financial crisis in 2007-08. Consequently, it doesn't match with management methods at commercial banks. Moreover, the currently discussed CVA method is converted for traders' use such that it tends to neglect correlation architecture of credit risks. Though derivative traders and rating agencies are enthusiastic about advanced mathematics, their common knowledge about credit risks is as little as that of a loan officer. There also exist huge modeling risks. This is just like the CDO price modeling situation where the sub-prime crisis was born.

Calculation Models

Here is the model we constructed.

1. Each transaction is dealt with a real accounting method. Each transaction is evaluated one by one, especially for derivatives with nonlinear risk characteristics and corporative or large transactions, for which the law of great numbers applies less.

   Because it is based on accounting principles, each financial transaction is itemized into various accounting subjects, not into simple market value. Fig.2 shows how a simple transaction is categorized in future accounting

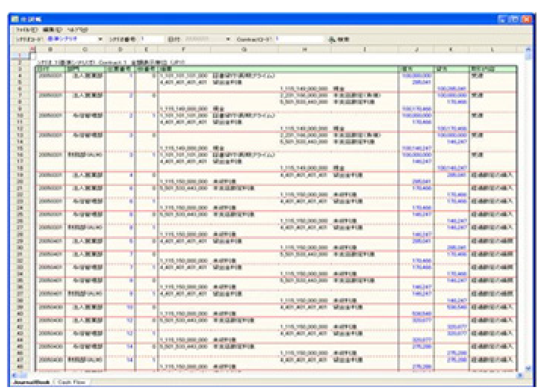

Fig.2 A simple transaction categorized in future accounting items.

items, where you can see 10 times more information compared to simple cash flow charts. While structured transactions naturally depend on cases, basically prices are divided into back assets and chased by nominal corpus. This method eliminates negative effects often produced by excess mark to model methods within investment banks.

2. Scenario paths of market rates and credit ratings of each company are generated with Monte Carlo methods.

   Correlation structures between market risks and credit risks have to be reflected to evaluate the whole portfolio. The biggest difference from VaR models here is that it has a consecutive time axis as time horizon instead of a single spot in the future. That is to say, it has one more dimension compared to VaR models as you can see in Fig.3. At generating market scenarios, we only put a mild restriction by analysis of only principal components, which is different from yield curves that are often used in pricing models for derivatives. This is because environmental factors where normal yield curve models don’t apply are needed to be concerned in scenarios in risk management.

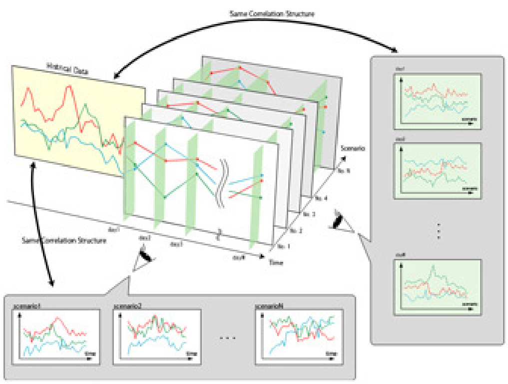

Fig.3 A 3 dimension structure of the future scenario.

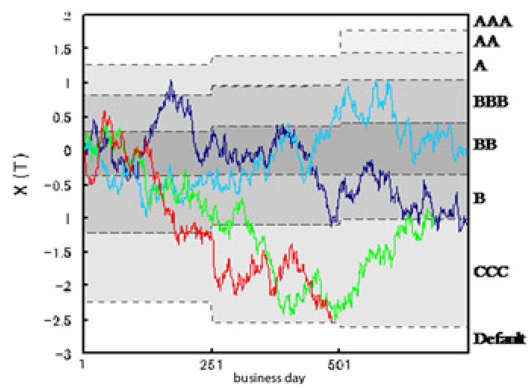

Fig.4 Daily changes of EDP and rating determinations.

   In credit risk management, not only default events but also daily EDP (expected default probability) on scenario paths is chased, besides ratings are chased by using the Merton [1974] model. This is shown in Fig.4. As a matter of fact, CVA at every combination of transaction is computed internally.

3. Daily evaluation for each scenario path for several years was done.

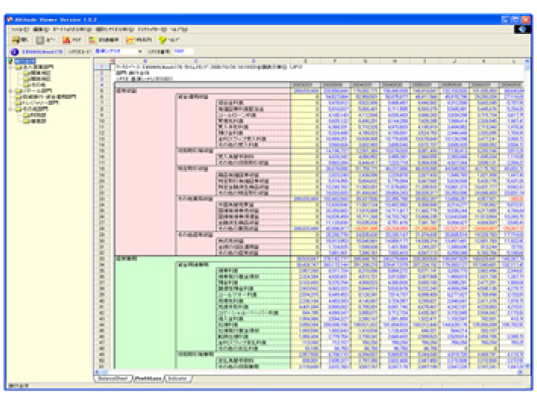

Fig.5 Future income statement output by this model.

If you only need market VaR and credit VaR to be calculated as an output of the model, calculation of market rate and credit ratings at time horizon spot is what you need. If you consider CVA, you need calculations until the maturity of transactions. Different from CVA, if you consider ALM, it only takes several years of data but outputs future financial statements as you can see in Fig.5, not only future present values.

Input Data

In case with the biggest bank in our country, the amount of required input data for simulations is the following: ca. 3 million transactions and ca. 1000 days for 3 years, besides Monte Carlo simulation for 0.1~1 million times should be done at actual financial institutions. The total amount of these calculations is equal to the product of above, which is massive. Considering the time for the board of meeting to give a management judgment, this calculation should finish within at least 24 hours. In this research, we consider those above information and set parameters as in Table 1.

Number of transactions	ca. 3.4 million
Number of total cash flow	ca. 0.5 billion
Number of accounting processing	ca. 4.4 billion transactions per scenario
Number of counterparty	ca. 0.2 million
Simulation period	3 years
Monte Carlo simulation	1～100 thousands simulations
Number of used nodes (Maximum)	135 nodes / 2160 cores
accounting processing rules	market value method + Amortization・Accumulation processing

Table.1 Basic input data

To help you to understand how many computations Table 1 requires, let’s compare to CVA, which is widely known for its expensive calculating costs. According to Morgan Stanley’s Canabarro [3], a typical CVA system for 2 to 10 million transactions it asks 1000-2000 market paths (1/10 of this research), 100 days for time horizon (1/10 of this research) [3].

(Numbers are 1/10 of this research, because it doesn’t consider itemized accounting.) All of these factors contribute by multiplication, which means the total amount of calculation at Morgan Stanley is only 1/1000 of this research. Just for the record, Morgan Stanley proudly announced that they had built a CVA system with large scaled grid computing exclusively for this calculation.

Research in Early Stage

This research has started with the assignment for enlarging new users of grid computing system called “Development of an ALM (Asset Liability Management) system for the banking and insurance industry” which took place 3 years ago. At this time, it was mainly to transplant one of our applications, Numerical Technologies Altitude®, into TSUBAME, the high performance computing environment at Tokyo Institute of Technology. The Altitude system contains the calculation model, explained in the last chapter, and the grid computing function supporting Windows and UNIX. Consequently the first task was to modify the new OS environment.

In the assignment in 2007, 1,000-10,000 times of Monte Carlo simulations were done without any major changes to the default code. With 1000 simulations, we changed the calculation nodes from 8 (128 core) to 96 (1536 core), adding one more node for the control panel, to see the performance of the cluster shown in Fig.6.

We didn’t dare to code specially for the TSUBAME environment. This is because even those clients of us, who have installed small scaled grid computing environments, are not equipped with such high performance networks as TSUBAME. As long as it is a commercial application for non-HPC users such as financial institutions, it should be designed to work with high performance and high reliability in an environment without an HPC system.

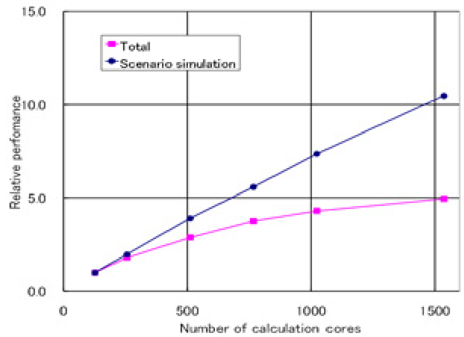

Fig.6 Cluster efficiency curve
(Before improvement of credit scenario generating stage)

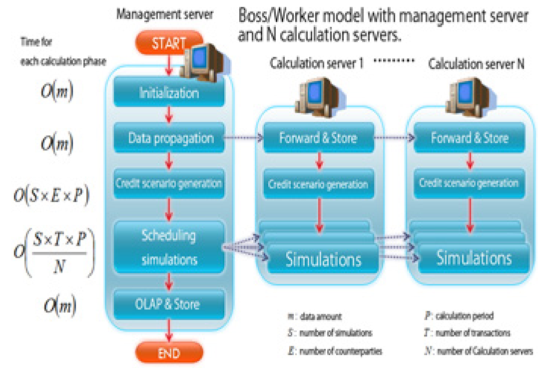

Fig.7 processing procedures at Altitude.

Discovery of Hot Spot

Altitude is composed with the proceeding stages shown in Fig.7. As for loading, data distribution, OLAP and storage, despite of the number of simulations there is no big difference in proceeding time, which makes the stage where credit scenarios are generated and the stage where scenarios candidates for parallel computing are simulated. It was clear before this research that the scenario simulating part would have a problem, so it had been already rewritten to support grid computing. As blue line in Fig.6 shows, it scales beautifully even with 1500 parallels.

At the early research stage, a problem occurred at the data distribution phase when data from Table 1 was inputted. It means that the time duration for data distribution of ca. 2 Gbyte, which normally finishes within a short time, became non-negligible because of node increases at the calculation server. This problem is solved by the pipeline method indicated in Fig.8. Fig.9. shows the improvement. Though we found that Atsushi Manabe at KEK was developing Dolly + application [4] with the same method afterwards, we decided not to claim since the performance of Altitude was higher than what Dolly + thesis proved.

The total performance by now is shown with red line in Fig.6. Although more parallels increase the quality of the performance, because of Amdahl's law the performance curve became gradually flat and parallelization efficiency became 50% at 1536 parallels. From this result, you can say that our Altitude performs excellent by comparison to famous commercial risk management products from NumeriX in US and from SAS, a famous general-purpose statistics software vender in the US, where their parallelization efficiency goes under 50% only at 100 parallels. However, this performance curve shows clearly that correspondence to a large scaled grid computing is necessary.

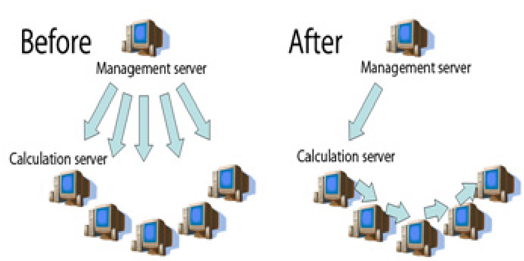

Fig.8 Improvement at data distribution stage.

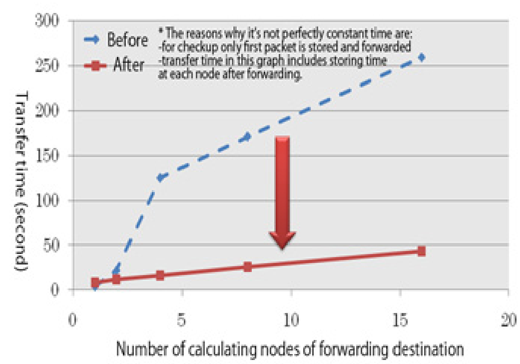

Fig.9 Effect of improvement at data distribution stage

Supporting Grid Computing at Generating Scenario Stage

We had thought that generating the credit scenario stage requires less calculations compared to the simulating scenario stage. However, TSUBAME raised the parallelization so that calculating the duration of the simulating scenario stage shortened. Besides, the data distribution stage was also speeded up thanks to the pipeline method. This is how generating the credit scenario stage becomes distinguished as a new hot spot as it is shown in Fig.10.

At generating the credit scenario stage, it deals with changes of ratings and defaults along the credit scenarios generated for each company at the same time. Generally there are relations between counterparties so it may affect each other, which is why transactions cannot simply be divided by nodes in parallelization computing.

For this reason, we first researched the relations among clients in advance and categorized the proceeding stages and then synchronized within nodes on each stage and distributed information for parallelization computing. Consequently, the performance has improved dramatically as shown in Fig.11.

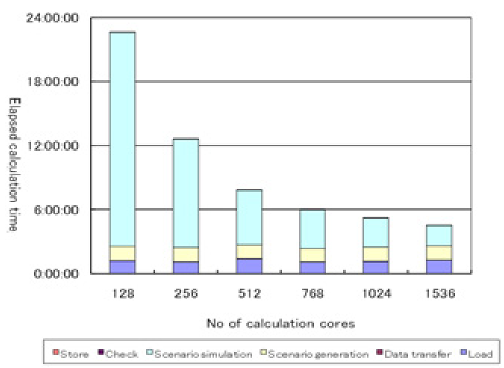

Fig.10 Calculation time for every stage (before improvements of the credit scenario generating stage).

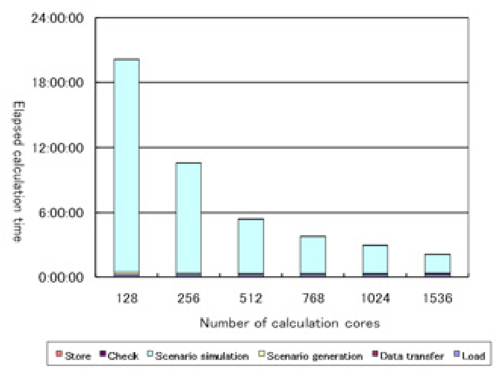

Fig.11 Calculation time for every stage (after improvements of the credit scenario generating stage).

In this improved method, it is necessary to distribute credit scenario information calculated in each node to every node, which needs to pass along bigger amounts of data. Though this would never be a problem in the luxury environment such as TSUBAME, at actual environments of clients discussions would be held whether they should install an Infiniband-class fast network in a view of cost-benefit performance.

Fig.12 shows cluster performances after the credit scenario generating stage has improved. You can see that it still scales beautifully even at 1500 parallelization.

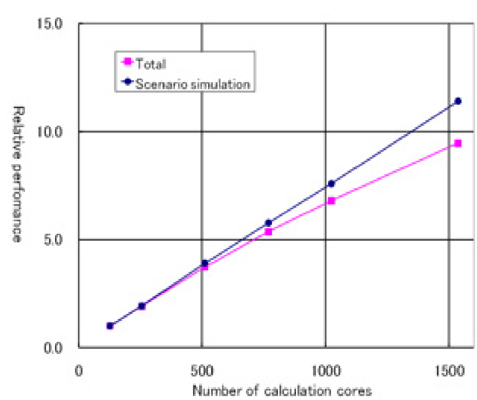

Fig.12 Cluster efficiency curve (after improvements of the credit scenario generating stage).

Flat-Out Test

Finally we had an entire/complete simulation as a flat-out test with the assumptions of Table 1, in case of the biggest bank in our country. As output, we gained TP allocated forecasted financial statements along every scenario path besides risk indexes such as VaR and EaR. It confirmed the operation of ALM simulations at 0.1 million class with 135 nodes (2160 cores) within 118.5 hours (ca.5 days or less), which means it is possible to simulate a mega-bank scaled ALM simulation within a practical time at an actual sized grid environment.

To show a part of the HPC performance, there are PDF (probability density function) of recurring revenues for 3 years from the first day of simulation; Fig.13 (1,000 simulations), Fig.14 (10,000 simulations) and Fig.15 (100, 000 simulations). As the number of simulations increase, the frequency distribution curve becomes gentle. Percentile numbers as risk index in practice cannot be calculated without a certain number of simulations because they are marginal probabilities such as 99%, 99.9% and 99.97%. In these days Monte Carlo simulation running over 100,000 times is practiced normally and even higher number of simulations for more accurate results are gaining attentions.

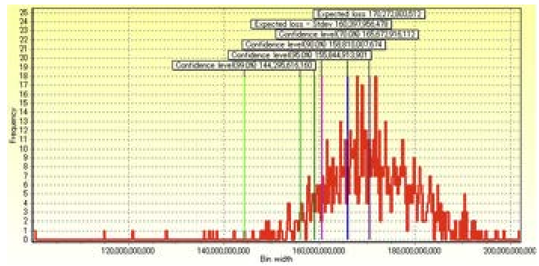

Fig.13 PDF (Number of simulations: 1000).

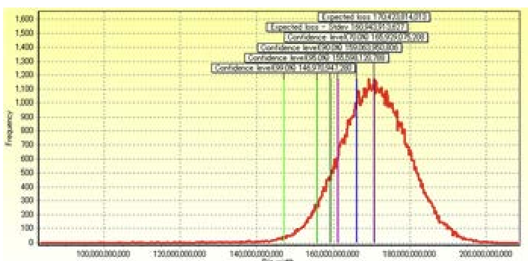

Fig.14 PDF (Number of simulations: 10000).

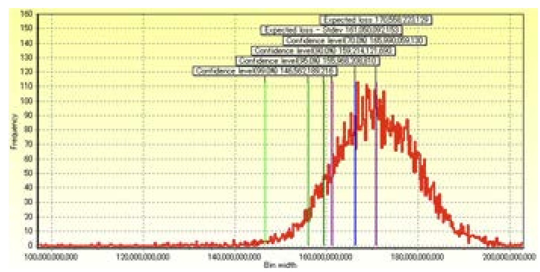

Fig.15 PDF (Number of simulations: 100000).

Conclusion

In this research, increasing performance characteristics along the increase of nodes and the calculation time for 100,000 simulations showed the simply estimated number of necessary cores to finish the calculation within a day. And that is 11200 cores. Although for commercial use a grid cluster with 10 thousand cores is quite big, for research an HPC environment these days is drawing closer to a commodity.

Besides, thanks to the great improvement of CPU performance itself (the CPU used for node in TSUBAME in this research is more than 3 years old.), we assume by now that it requires less than half the number of cores of what we described to get the same performance. That is to say, no matter what the regulatory agency and financial institutions insist, the amount of calculations can’t be the excuse of not being able to build a system to get a grip on the whole image of financial risks.

If the latest HPC can make the size of a system smaller, building costs can be reduced.

It results in more private enterprises to introduce the system. Consequently leading-edge and one-of-a-kind GPUGPU cluster TSUBAME 2.0, which is scheduled to be activated within year 2010, is starting to gain a lot of attention.

Acknowledgement

This research had been supported by Open Advanced Facilities Initiative for Innovation (Strategic Use by Industry) and MEXT Open Advanced Research Facilities Initiative. We'd like to thank here for the great support through this research.

Reference work

1. Philippe Jorion (1996), "Value at Risk: The New Benchmark for Controlling Derivative Risk," Irwin Professional Pub
2. J.P. Morgan & Co. Incorporated (1997), "CreditMetricsTM – Technical Document."
3. Eduardo Canabarro (2010), "Counterparty Credit Risk," Risk Books
4. Atsushi Manabe (2001), "Disk cloning program 'Dolly+' for system management of PC Linux cluster," KEK, Tsukuba, Japan
5. 塩田雅之 (2010)、"ポートフォリオ管理パッケージの並列化性能検証," Market Solutions Review Vol.3 No.2