

Investigating the Cloud Computing Business Framework - Modelling and Benchmarking of Financial Assets and job submissions in Clouds

Accepted by “Research Clouds: Hype or Reality?” Workshop

Victor Chang¹, Gary Wills¹, David De Roure^{1,2}, Clinton Chee³

1. School of Electronics and Computer Science, University of Southampton, Southampton SO17 1BJ

2. Oxford e-Research Centre, University of Oxford

3. Commonwealth Bank, Sydney, Australia

{vic1e09, gbw}@ecs.soton.ac.uk; david.deroure@oerc.ox.ac.uk; cykchee@yahoo.com

Abstract

Literature identifies three business challenges in clouds: (i) little linkage between qualitative and quantitative cloud business frameworks in the same domain; (ii) few structured frameworks to measure cloud business performance and (iii) application portability from desktops to clouds, and later on between clouds offered by different vendors. To address these three problems, we propose the Cloud Computing Business Framework (CCBF), which contains Financial Cloud Framework (FCF), Middleware Framework (MF) and the other two frameworks. FCF and MF are to deal with portability issue. In FCF, we select Monte Carlo Methods (MCM) for pricing and Black Scholes Model (BSM) for risk analysis. In MF, we select OMII-UK's GridSAM 2.3 to demonstrate job submission in clouds, and compare benchmarking results with our MCM and BSM models. Our objective is to demonstrate portability, speed, accuracy and reliability of applications in the clouds, and present how modelling, simulation and benchmarking fit into FCF and MF. Experiments are performed in public and private clouds, where portability, speed, accuracy and reliability from desktop to clouds are successfully demonstrated.

1. Introduction

There are three technical and business challenges in cloud computing to be identified, and front of which include (i) vendors' lock-in; (ii) security and (iii) interoperability for technical challenges [5]. Three business challenges are our research focus that we work with, and are briefly described as follows. Firstly, there is a little linkage between qualitative and quantitative cloud business frameworks in the same domain [1]. Secondly, there are not many structured frameworks to measure cloud business performance [1, 4]. Thirdly, application portability from desktops to clouds, and later on between clouds offered by different vendors, is challenging [2, 5]. To address three issues, we propose the Cloud Computing Business Framework (CCBF), which contains Financial Cloud Framework (FCF), Middleware Framework (MF) and the other two frameworks, where FCF and MF are aimed to demonstrate portability, the third research issue. In FCF, Monte Carlo Methods (MCM) and Black Scholes Model (BSM) are selected as they are standard models for pricing and risk analysis. Our objective is to demonstrate portability, speed, accuracy and reliability of financial

models in the public and private clouds. Modelling, simulation and experiments are used for methodologies.

2. Monte Carlo Methods (MCM)

A number of methods for calculating prices include MCM, Capital Asset Models and Binomial Model. However, the most commonly used method is MCM. Hence, MCM is used for this portability demonstration. MCM is used in stochastic and probabilistic financial models, and provides data for investors' decision-making [3]. MATLAB is used due to its ease of use with relatively good speed. While the volatility is known and provided, prices for buy and sale can be calculated. The following code demonstrates calculation of prices. Call prices are for buy and put prices are for sale. The program calculates the lower limit, ideal value and the upper limit for each buy and sale category.

```
> fareastmc
```

```
[LowerLimit MCPrice UpperLimit]
```

```
Call Prices: [4.196694 4.248468 4.300242]
```

```
Put Prices: [7.610519 7.666090 7.721662]
```

2.1 The role of VBA in Finance

Visual Basic for Applications (VBA) is very commonly used in Finance applications, which include a wide range of software and tools. On contrast, HPC languages are less commonly used than VBA in Finance. In order to demonstrate portability, we write one MCM application in the form of a VBA Excel program to calculate the best call and put prices. Here is an example: Spot Price = 100; Strike Price = 105; Volatility = 0.1; Risk free rate = 0.05, Option Maturity = 1; Time steps = 10 and Number of simulations are 10000. The VBA Excel program will calculate the best call price as 4.009, and best put price as 3.903. This reduces complexity in using and analysing MCM, and this VBA Excel enables portability to Public (Dropbox) and Private clouds, which include Platform as a Service (PaaS) in particular.

2.2 Monte Carlo Methods in Banking

Mathematical models such as MCM are used in Operational Risk in Risk Management area, where models

are used to simulate the risk of exposures to various types of operational risks. MCM simulations are written in Fortran and C#. Such simulations may take several hours or over a day. The results may be needed by the bank for the quarterly reporting period.

3. Black Scholes Model (BSM)

Methods such as Fourier series, stochastic volatility and BSM are used for volatility. As a main stream option, BSM is selected for risk analysis in this paper, since BSM has finite difference equations to approximate derivatives. We write fdcall.m to calculate call price and also risk analysis based on BSM, and contain key values such as

- strike price: the price targeted for sale.
- upper boundary: the highest possible range a price or risk can reach.
- risk free rate: interest investors would expect from an absolutely risk-free investment over a period of time.
- maturity: the loan is due to be repaid on a fixed date.
- volatility: used to quantify the risk of assets.
- dividend yield: the return on investment for an asset.
- asset steps: a specific BSM method called explicit time steps. The more steps, the more accurate the analysis.

This allows us to calculate and track call prices if variations for maturity, risk free rate and volatility change. Similarly, we can modify our code to track volatility for risk analysis if other variables are changed.

4. Experiment and Benchmark in the Clouds

Code was written for Variance-Gamma (VG) Processes (a specific technique in MCM) to be used for experiments and benchmark in the clouds, since VG processes are suitable in reducing errors [6]. Methodologies include simulations, modelling and experiments. The hardware descriptions are summed up in Table 1.

Desktop	2.67 GHz Intel Xeon Quad Core and 4 GB of memory (800 MHz)	32-bit Windows XP
Public cloud	A large resource instance of dual core CPU, with 2.33 GHz speed and 7.5GB of memory.	Amazon EC2, 64-bit Ubuntu 8.04 (MCM & BSM); 32-bit CentOS 5.4 (JSDL, Section 5)
Private cloud	2 cores of 2.67 GHz and 4GB of memory at 800 MHz.	32-bit Windows XP virtual server
Private cloud	2.8 GHz Quad Core Xeon, 16 GB of memory	64-bit Windows server

Table 1: Hardware and operating systems comparisons

All these four settings have installed Octave 3.2.4, an open source compiler equivalent to MATLAB. 5000, 10,000 and 15,000 MCM simulations are performed three times, and the time taken at each of a desktop, private clouds and EC2 public clouds are recorded and averaged with three attempts. Private cloud (rack server) has the best hardware

configuration with the fastest download speed and unlimited bandwidth, thus it runs faster than the rest.

Table 2 summarises the timing benchmark result while running the modelling of assets (MoA) code.

Number of simulations and time taken (sec)	5,000	10,000	15,000
Desktop	11.08	11.92	12.71
Public cloud (large instance)	11.95	12.30	13.15
Private cloud (virtual server)	11.31	12.13	12.90
Private cloud (rack server)	9.63	10.51	11.48

Table 2: Timing benchmark to run MoA code on Octave 3.2.4

All hardware infrastructures would ideally have the same CPU speed and operating system (with variations in memory) but it was difficult to synchronise since those hardware were from different sources.

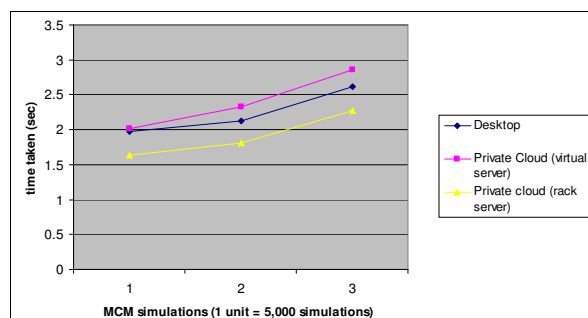


Figure 1: Timing benchmark comparison for MATLAB 2007

Figure 1 refers to benchmark results if using MATLAB 2007, which compile faster than Octave, are only available on desktop, private cloud (virtual server) and private cloud (rack server) hosted on Windows. The same code runs faster on MATLAB 2007, but it comes with higher prices.

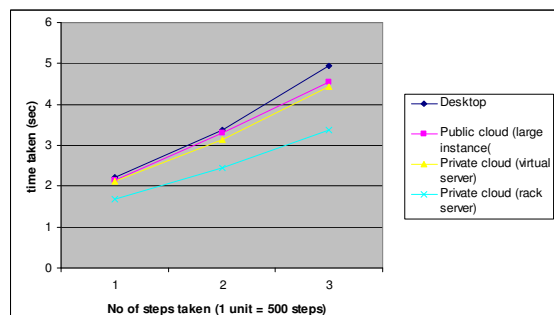


Figure 2: Timing benchmark comparison for Octave 3.2.4

Figure 2 shows benchmark results while running BSM. 500, 1,000 and 1,500 BSM simulations are performed three times, and the time taken at each of a desktop and two private clouds are recorded and averaged with three attempts. Time series used in BSM can take accommodate up to 1,500 simulations. Private cloud (rack server) has the best hardware configuration with the fastest network speed and unlimited bandwidth, thus it runs the fastest.

Benchmark results show pricing and risk analysis can be calculated rapidly with accurate outcomes. Portability is achieved with a good reliable performance in clouds. These experiments demonstrate portability, speed, accuracy and reliability from desktop to clouds.

5. Job Submission by GridSAM 2.3

This relates to Middleware Framework (MF) which contains core CCBF components such as GridSAM for portability. GridSAM 2.3 is chosen as it is widely used in the UK community and is very easy to modify the Job Submission Description Language (JSDL) for job submission. GridSAM 2.3 allows multiple submissions for up to 20,000 jobs submitted at each instance. A Java program, Primes, is written to list prime numbers, and is able to submit jobs to Private and Public Clouds. Here is an example: Two JSDL files, primes-0to10000.jsdl and prime-10001to20000.jsdl are written to send 20,000 jobs, and a primes_file.pl is modified to lists prime numbers between 0 and 20,000. This includes two stages. Firstly, it involves with job submission and its status check. Secondly, the result is computed and stored in the GridSAM client. For Cloud demonstrations, the first stage is used to measure the time taken of the job submission, and compare required time with the same set-ups and experiments described in Section 4. The GridSAM 2.3 server does not support desktop. Hence, Public and Private Clouds are used for experiments, with their outcomes summed up in Figure 3.



Figure 3: Timing benchmark comparison for GridSAM 2.3

Time taken at Private clouds does not have a distinct advantage comparing to experiments in Section 4. This may relate to the way that GridSAM works. We may plan using an alternative tool for comparison.

6. The implications for Banking

There are implications for banking. Firstly, security is a main concern where some security issues still experience evolving challenges. This is in particular when Cloud vendors tend to mitigate this risk technically by segregating different parts of the Clouds but still need to convince clients about the locality of their data and data privacy.

X.509 are used in finance clouds, and in our experience, single sign-on could be more suitable. Secondly, financial regulators are imposing tighter risk management controls. Thus, financial institutions are involved in running more analytical simulations to calculate risks to the client organisations. This may present a greater need for the use of the Cloud computation and resources. Thirdly, portability of the Cloud can imply letting clients to install their own libraries. Users who run MATLAB on the Cloud may only need the MATLAB application script or executable and to install the MATLAB Runtime once on the Cloud. For financial simulations written in Fortran or C++, users may also need Mathematical libraries to be installed in the Clouds. Clouds must facilitate an easy way to install and configure user required libraries, without the need to write additional APIs like several practices do.

7. Conclusion and Future Work

Literature identifies three business challenges for clouds. This paper is to focus on the third issue, portability. MCM, BSM and GridSAM 2.3 are used to demonstrate how portability, speed, accuracy and reliability can be achieved while moving financial applications from desktops to clouds. This well fits-in an objective in the CCBF to allow portability on top of, secure, fast, accurate and reliable clouds for IaaS, PaaS and SaaS. Our research purpose is not intended to test pure computing performance of Cloud Computing; instead the intention is to port and test financial applications to run on the Clouds. It is not yet critical to use Dongarra's netlib functions for benchmark, but there are plans to use HPC languages such as C++ for next stage. We have Health Cloud Storage Framework (HCSF) that can demonstrate portability in details. We hope to strengthen our results and extents of collaboration.

8. References

- [1] Chang V, "A proposed Cloud Computing Business Framework", nine-month thesis technical report, School of Electronics and Computer Science, University of Southampton, September 2010.
- [2] Chang V, Wills G and De Roure D, "Towards Financial Cloud Framework: Modelling and Benchmarking of Financial Assets in Public and Private Clouds", submitted to IEEE Cloud 2010, the 3rd International Conference on Cloud Computing, Miami, Florida, July 5-10 2010.
- [3] Hull J C, "Options, Futures, and Other Derivatives", Seventh Edition, Pearson, Prentice Hall, 2009.
- [4] Klems M, Nimis J and Tai S, "Do Cloud Compute? A Framework for Estimating the Value of Cloud Computing", Cloud Computing and its Application CCA 08, Chicago, October, 2008.
- [5] Patterson D, Armbrust M et al, "Above the Clouds: A Berkeley View of Cloud computing". Technical Report No UCB/EECS-2009-28, UC Berkeley, February 2009.
- [6] Rebeiro C, Webber N (2002), "Valuing Path Dependent Options in the Variance-Gamma Model by Monte Carlo with a Gamma Bridge", Working Paper, No.02-04, Warwick Business School, Sep 2002.