Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 108C (2017) 626–634

Procedia Computer Science

International Conference on Computational Science, ICCS 2017, 12-14 June 2017, Zurich, Switzerland

Utilizing Intel Advanced Vector Extensions for Monte Carlo Simulation based Value at Risk Computation

D. N. S. S. Liyanage, G. V. M. P. A. Fernando, D. D. M. M. Arachchi, R. D. D. T. Karunathilaka, and A. S. Perera

Department of Computer Science and Engineering, University of Moratuwa, Moratuwa, Sri Lanka
nipunasankalpa.12@cse.mrt.ac.lk, pubudu.12@cse.mrt.ac.lk, dilinimma.12@cse.mrt.ac.lk, dilip.tharuka.12@cse.mrt.ac.lk, shehan@cse.mrt.ac.lk

Abstract

Value at Risk (VaR) is a statistical method of predicting market risk associated with financial portfolios. There are numerous statistical models which forecast VaR and out of those, Monte Carlo Simulation is a commonly used technique with a high accuracy though it is computationally intensive. Calculating VaR in real time is becoming a need of short term traders in current day markets and adapting Monte Carlo method of VaR computation for real time calculation poses a challenge due to the computational complexity involved with the simulation step of the Monte Carlo Simulation. The simulation process has an independent set of tasks. Hence a performance bottleneck occurs during the sequential execution of these independent tasks. By parallelizing these tasks, the time taken to calculate the VaR for a portfolio can be reduced significantly. In order to address this issue, we looked at utilizing the Advanced Vector Extensions (AVX) technology to parallelize the simulation process. We compared the performance of the AVX based solution against the sequential approach as well as against a multithreaded solution and a GPU based solution. The results showed that the AVX approach outperformed the GPU approach for up to an iteration count of 200000. Since such a number of iterations is generally not required to gain a sufficiently accurate VaR measure, it makes sense both computationally and economically to utilize AVX for Monte Carlo method of VaR computation.

Peer-review under responsibility of the scientific committee of the International Conference on Computational Science

Keywords: Monte Carlo Simulation, Value at Risk, AVX, GPU

1 Introduction

Value at Risk (VaR) is a commonly used method in the finance industry for assessing market risk and, Monte Carlo simulation is one of the most commonly used methods for calculating VaR in practice. In today's large markets it is common to see large volumes of transactions occurring at any given time. This leads to the possibility of having considerable intraday fluctuations in markets. The common practice of calculating the VaR at the end of the day on a daily basis [1] is not sufficient for capturing these intraday movements. A real-time solution for calculating

1877-0509 © 2017 The Authors. Published by Elsevier B.V.
Peer-review under responsibility of the scientific committee of the International Conference on Computational Science
10.1016/j.procs.2017.05.156

the VaR would be ideal since it would be able to capture the intraday fluctuations the daily VaR fails to capture. However, the compute-intensive nature of the Monte Carlo simulation poses a serious challenge in implementing an efficient enough real-time Monte Carlo Simulation for calculating the VaR [2].

In this study we identified the performance bottleneck of the Monte Carlo simulation method for VaR calculation and looked at utilizing Intel Advanced Vector Extensions (AVX) technology found in modern Intel CPUs for improving the performance to see whether it would result in a performance gain which makes it a viable option for computing the VaR in real-time. We compared the performance improvement gained by 3 methods (concurrent implementation, AVX + OpenMP and GPU utilization) against the general implementation to see how effective AVX can be for Monte Carlo simulation based VaR assessment. The financial portfolios considered for VaR calculation were restricted to stock portfolios since our real-time VaR solution focused only on stock portfolios for simplicity. But this should be extendable for other portfolios.

The rest of the paper is structured as follows. Section 2 describes the use of Monte Carlo Simulation in computing VaR. Section 3 gives a brief overview of AVX and its uses while section 4 delineates how GPUs can be used for Monte Carlo Simulation. Section 5 describes the methodology followed and section 6 illustrates the results of the research followed by the conclusion in section 7.

2 Monte Carlo Simulation based Value at Risk

Value at Risk was initially developed by Dickson H. Leavens and further developed later on by Markowitz and Roy, William Sharpe and many others [3]. At present, Value at Risk is a predominantly used method to measure the market risk [4] of financial portfolios comprising of various assets such as bills, stocks and bonds etc. VaR is usually defined using three properties: a base currency, a given time horizon and a given confidence level [5]. Given these, a person interested in calculating the VaR would collect the losses/profits gained during the time horizon, build a distribution from these data and take the relevant percentile for the given confidence interval to determine the VaR.

There are three predominantly used methodologies to calculate the Value at Risk: Historical Simulation, Parametric Method and Monte Carlo Simulation. Monte Carlo Simulation differs from the other 2 methods since instead of using actual losses/profits to build the distribution, it uses randomly generated prices based on an actual price to build the distribution, which then will be used to determine the VaR. The Geometric Brownian motion equation given by Eq. 1 can be used to calculate random stock values.

$$S_t = S_{t-1} \exp\left([\mu - \frac{\sigma^2}{2}]t \ + \ \sigma Z\sqrt{t}\right) \tag{1}$$

Where:

$\mu =$ mean of the distribution of the log returns

$\sigma =$ standard deviation of the distribution of the log returns

$Z =$ random value from the standard normal probability distribution

$S_t =$ current stock value

From this simulation process, a specified number of simulated stock prices to calculate market values and build a loss distribution out of those values which would then in turn be used to estimate the VaR metric.