High Performance Monte Carlo Computation for Finance Risk Data Analysis

A thesis submitted for Degree of

Doctor of Philosophy

By

Yu Zhao

Electronic and Computer Engineering

School of Engineering and Design

Brunel University

November 2013

Abstract

Finance risk management has been playing an increasingly important role in the finance sector, to analyse finance data and to prevent any potential crisis. It has been widely recognised that Value at Risk (VaR) is an effective method for finance risk management and evaluation.

This thesis conducts a comprehensive review on a number of VaR methods and discusses in depth their strengths and limitations. Among these VaR methods, Monte Carlo simulation and analysis has proven to be the most accurate VaR method in finance risk evaluation due to its strong modelling capabilities. However, one major challenge in Monte Carlo analysis is its high computing complexity of $O(n^2)$.

To speed up the computation in Monte Carlo analysis, this thesis parallelises Monte Carlo using the MapReduce model, which has become a major software programming model in support of data intensive applications. MapReduce consists of two functions - Map and Reduce. The Map function segments a large data set into small data chunks and distribute these data chunks among a number of computers for processing in parallel with a Mapper processing a data chunk on a computing node. The Reduce function collects the results generated by these Map nodes (Mappers) and generates an output. The parallel Monte Carlo is evaluated initially in a small scale MapReduce experimental environment, and subsequently evaluated in a large scale simulation environment. Both experimental and simulation results show that the MapReduce based parallel Monte Carlo is greatly faster than the sequential Monte Carlo in computation, and the accuracy level is maintained as well.

In data intensive applications, moving huge volumes of data among the computing nodes could incur high overhead in communication. To address this issue, this thesis further considers data locality in the MapReduce based parallel Monte Carlo, and evaluates the impacts of data locality on the performance in computation.

Acknowledgements

I would like to give my biggest thanks to my supervisor Professor Maozhen Li. With his guidance, advice and support during my research, I overcame many difficulties and finally arrive at today’s achievement step by step.

I give my honest thanks to my parents, my younger brother, my wife and my daughter. They have given me the spiritual supports to study in a foreign land.

I would like to thank my friends: Yang Liu and Zelong Liu. They helped me so much during my study.

Author’s Declaration

The work described in this thesis has not been previously submitted for a degree in this or any other university and unless otherwise referenced it is the author’s own work.

Statement of Copyright

Table of Contents

List of Figures

List of Tables

Chapter 1: Introduction

1.1 Background

Financial risk means that there are possibilities that investors can lose money during their economic activities. The common financial risks include credit risk, liquidity risk, market risk and operational risk [1]. The credit risk means that the debtor cannot repay the debt principal and interest, so that creditors may be economic losses. For example, the financial crisis in 2008 was caused by credit risks. The liquidity risk refers to the economic entities such as firm loss due to the uncertain changes of the financial liquidity of the assets such as cash flow break. The market risk means the potential losses caused by changes of market factors such as stock market prices, interest rates, exchange rates and other changes in value [1]. The operational risk is the risk of loss which resulted by internal processes, people or systems' mistakes [2], or other external operations and relevant events.

Financial risks cannot be eliminated, but can be controlled. The control approach is the financial risk management, which means controlling the possible loss undertaken within the limits in the market economic activities. The famous financial risk management method is Value-at-Risk (VaR)[2],which focuses on the hidden risks and potential losses. Furthermore, the VaR includes three elements: a preset level of loss, a fixed period of time which risk is evaluated and a confidence interval [2].

There are three main approaches to measure the Value at Risk: the Variance-Covariance method, the Historical Simulation and the Monte Carlo simulation [3]. The Variance-Covariance method originates a probability distribution of the hidden risky values through relative simple computing. The advantage of this method is simpleness. There have four steps be concerned when using this method to map the risks. First, it requires users to take each individual asset in portfolio and translate the asset to standardized instrument. Second, each asset is explained as set of position in the standardized instrument which was indicated in the step one. The third step is the key of the whole process while once the asset in portfolio had been identified in the standardized instrument and then the user has to assess the variances and covariance by searching historical data. The final step is computing the VaR by using the weights from the step two and the variances and covariance from the step three. Even the Variance-Covariance method is very simple to calculate, it has three weaknesses. First is the wring distributional supposition. This means if the returns distribution is abnormal and then the computed VaR will be much lower than the true VaR. Second is the input error. Even the distributional supposition is correct the result still possible be wrong due to used incorrect variances and covariance. Third is non-stationary variable.

The historical simulation is a simple way to compute the VaR of numbers of portfolios by using the historical data of a specify asset in the portfolios. The wakness with the historical simulation is the past performance doesn’t equal to the performance in future. In those three VaR approach the historical simulation is the most depend on historical data method. The second weakness is the historical data has its trend. In the historical simulation all data in the equal period of time are weighted equally during the process of computing VaR. but in difference historical time the data are influenced by various factors such as the price changes [3]. The third weakness of the historical simulation is difficult to compute the VaR for a new asset or a new market risk.

The Monte Carlo method bases on the theory of probability and statistics. Monte Carlo method is used widespread because of it can realistically simulate the characteristics of things and the physical experiments to solve problems which difficult to be done by numerical methods. In other words, the Monte Carlo simulation can be considered as huge random experiments which are used to compute the specific unknown result.

1.2 Motivation

As described above, the VaR approach has three common methods: the Variance-Covariance method, the Historical simulation and the Monte Carlo simulation [2]. Although the VaR is a very famous tool in risk management, it has many limitations. Almost methods need using more or less historical data to compute the VaR. Comparing with other methods the Monte Carlo method is the only one that can compute VaR without historical data. It bases on the theory of probability and statistics. So the Monte Carlo simulation depends on repeating random sampling and statistical analysis [4][5]. Simulating more iterations mean that the calculated value is more close to the true value [6]. In other words, more computations would lead to that the calculated probability of risk occurred is closer to the truth. On the other hand, more computations mean more time cost. The financial market is constantly changing, so it cannot provide real help for the decision-making process in a timely manner. In other words, spend less time to calculate the result which is inaccurate, but accurate result takes long time.

The thesis focuses on speeding up the computation of Monte Carlo parallel by using MapReduce [78]. The computation of Monte Carlo is repeating the calculating process, so the simulation is very suitable for parallelization. The MapReduce is developed to compute large-scale data [84] in parallel environments. It uses Map and Reduce functions to process data. First, it divides large data files into file chunks (usually 16MB – 64MB) which are formatted in the <key, value> pairs [84]. Those chunks are used as input data and processed by the Map function which is built up by Mappers and running on many nodes. Comparing with large input file, the chunks will be processed very quickly because of those chunks are computed parallel by many nodes at the same time. Then those Mappers produce the intermediate data <key, value> pairs [84]. Those pairs will be identified and sorted in different groups by the same key and sent to the Reducers which are running on many nodes. Similar to the

Map function, those data also are parallel computed. At last, the result that is produced by each of Reducer will be mixed and added into the final output file. The time cost of using MapReduce to compute large-scale data is less than compute in normal way. The MapReduce model can work on a cluster or any other distributed systems which may contain a lot of various personal computers. Therefore, it is a good parallel computing platform.

1.3 Major Contributions

The major contributions of the thesis are summarized below.

- The thesis reviews VaR methods and assesses their limitations. Each method of VaR has its own limitations which may lead to incorrect results. In some particular situations the computing errors may be large enough to lead to mistakes in user's decision making process. Usually the VaR methods less concern the market risks during its computation process. That means the true values of risks are bigger than the computed VaR values.

- It presents a MapReduce based parallel Monte Carlo algorithm. For a Monte Carlo job, a number of Mappers are used to process data segments in parallel which speeds up the computation process. All the output data of Mappers also are processed by Reducers in parallel. The parallel Monte Carlo algorithm is evaluated in both a Hadoop cluster and a simulation environment, and the results show that effectives of the algorithm.

- To further improve the performance of the parallel Monte Carlo, data locality is considered in a Hadoop MapReduce cluster. Hadoop jobs are allocated to these nodes that are close to the data. This reduces the overhead in data transmission. The results show the performance improvement of the Hadoop cluster with data locality with different sizes of input data; different sizes of chunks; different number of nodes and different number of replicas. It also indicates that using data locality significantly improves the performance of the parallel Monte Carlo simulations.

1.4 Methodology

There are many methods can deal with the simulating of Monte Carlo method. In this thesis, Matlab and MapReduce were used to do tests. Matlab embed a lot of simulators in its program, Monte Carlo is one of them. Matlab is easy to use and it also can provide visualization of the simulated result. But its operation is opaque that means it is difficult to optimize the process or identify the problems of simulating process.

Compare with Matlab, MapReduce is an open-source [78] platform which can deal with large-scale data. Its operation is easy to parallelize the Monte Carlo algorithm. The important point is all the stages of MapReduce can be set or adjusted by users. This means that has great potential for optimizing. On the other hand, users can observe and quickly identify the problems. For example, the input data are divided into chunks, the sizes of chunks can be set by users. How many Mappers and Reducers are used and the sort factor also are set by users.

1.5 Thesis Structure

The rest of the thesis is structured in the following way.

Chapter 2 presents the literature review of all relevant topics of this thesis, such as different types of financial risks, advantages and disadvantages of Value-at-Risk methods, the Monte Carlo method and its limitations.

Chapter 3 presents the details of Monte Carlo simulation. It discusses development of Monte Carlo method, and discusses in-depth the characteristics of Monte Carlo simulation.

Chapter 4 presents the design and implementation of the parallel Mote Carlo building on the MapReduce programming model which has become the major model in support of data intensive applications in cloud computing systems [78]. It analyzes the implementation and performance in distributed system of MapReduce programming model. The performance of Monte Carlo simulation is also evaluated in both an experimental and simulated MapReduce cluster environments.

Chapter 5 evaluates data locality aware scheduling process in MapReduce model.

Data locality helps to improve the performance of MapReduce when processing a large-scale data and keeps computation close to data. The testing results show the effectiveness of data locality which reduces time in computation.

Chapter 6 concludes the thesis and points out some future works.

Chapter 2: Literature Review

2.1 Background of Value at Risk

Financial risks cannot be eliminated, but can be controlled. The control approach is the financial risk management, which means controlling the possible losses undertaken within the limits in the market economic activities. The Value at Risk (VaR) is a popular and widely used tool to measure financial risks by financial institutions in recent years [7], and it becomes a kind of technical standard [8]. The VaR approach is one of the most effective risk management techniques on market level currently. Computing VaR values become more possible and easier due to the increasing of computer simulating capabilities.

The competition between financial institutions has been increased and became fiercely due to the financial derivatives markets' rapid development in the world. The large international banks and securities [7][15] firms recognized and developed their own risk management systems in the first time when they were conscious of influences of effectivities of market risk management on value changes of their financial products such as portfolios [9]. Can be measured is one characteristic of financial risk [10], so a technical measure of risk can be carried out to analyze the risk degree of losses. Which means the basis of risk management is measuring risks in values. Financial institutions are more likely to determine and control risk if a more accurate measure method is adopted. Value at Risk as a famous financial tool to help managers during their decision making [7][9][11] process is widely known. The VaR is the most advanced and most widely applied method comparing with other various risk measure methods. It is the latest financial risk management tool used to measure mainstream market risks and is a famous approach to measure financial market risks by using of integrated modern mathematical techniques [12] and the increased of complex of financial risks since 1990s [16]. The VaR method focuses on the hidden risks and potential losses [13]. Every banks and firm fear of liquidity risk such as cash flow crisis which can cause the firms in dangerous even bankruptcy [10][14]. It is used by investment banks and financial institutes to calculate the potential losses of their financial products or trading portfolios in value over a specified period of time. It is important to those financial firms or banks who are try to make the investment decisions without the hidden risky influences on their cash flow [14].

The index of VaR was widely used due to the strongly recommended and encouraged by the Basel Committee on Banking Supervision (BCBS) [15]. Organizations, financial institutions and banks as the members had to publish their daily VaR reports [15] by response the requirements of Basel Committee on Banking Supervision. The reason of VaR approach so attractive is it can express the potential losses by using of currency measure units, which is the core of risk management. Currently, the VaR method is used in various types of financial risks measurement and management in the world. Usually to measure the VaR needs to preset a confidence interval, in other words, a confidence level [16]. For example, calculate a VaR value with set the 95% confidence level with a £100 million asset during a one-week time, which means over one week time there has only 5% chance that the losses will more than £100 million [9][10][17][19].

The definition of VaR means how the sizes are of the maximum potential values may suffer losses of an individual assets or a group of assets or portfolios under normal market condition with a given confidence level [17][18]. In other words, the financial instruments and portfolios are facing a potential maximum amount of losses or worst-case amount of losses in a certain holding period of time and a certain degree of confidence. An example is shown in Figure 2.1.

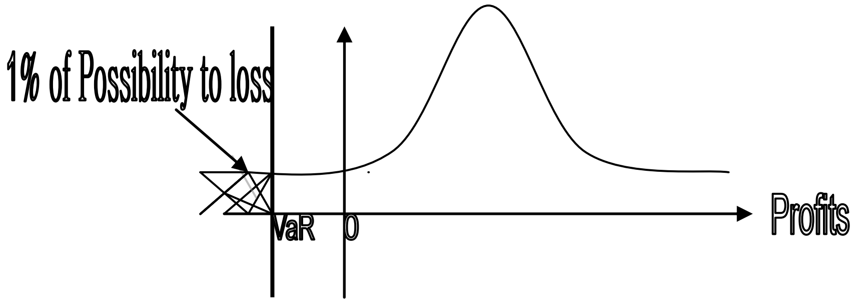


Figure 2.1: The maximum of possibility to loss in one day with 99% confidence.

Value at Risk is a tool to measure securities' financial risk by using of the statistical technique. Its mathematical definition is:

$$\mathrm{P}\{\, \Delta p \rangle VaR \,\}=1\text{-}\mathrm{c} \qquad (2\text{-}1)$$

Where Δp are the securities portfolio losses in a period of holding time (Δt). VaR is the value at risk with confidence level c [20].

In other words, for a particular portfolio, VaR gives the maximum possible expect losses of the portfolio under normal market condition [21] with a given confidence level in a period of time. This means the VaR has answered that the possibility of occurrence of loss is more than c, in other words, it ensures the probability of the loss will not exceed the VaR is 1-c. For example, the Bankers Trust Company in its 1994's annual report disclosed that its average daily VaR is $35 million with 99% confidence level in 1994 [15]. Which means the company promised that the average loss of its each product on a specific point of time will not be more than 35$ million in the next 24 hours. The company's capital in contrast was $4.7 billion and it achieved $615 million annual profits in 1994, the bank's risk profile was shown at a glance through a VaR report.

2.1.1 The General VaR Calculation Method

Actually the VaR is the area between the expectation value and the minimum value of the portfolio at a certain confidence level under the normal circumstances. According to Jorion's definition [18], VaR can be defined as:

$$\text{VaR} = E(\omega) - \omega^{*} \qquad (2\text{-}2)$$

Where E（ω） is the expectation value or expectation value of the asset or portfolio;

ω is the finally true value of the portfolio;

ω^* is the minimum value of asset or portfolio at the end with confidence level α.

And supposing that $\omega = \omega_0$（1+R） (2-3)

Where ω_0 is the value of the portfolio [18] when held at the beginning;

R is the rate of return of portfolio in a preset period of holding time (usually one year).

$$\omega^* = \omega_0 (1+R^*) \qquad (2\text{-}4)$$

R^* is the lowest rate of profit of the portfolio with the confidence level α.

According to the basic nature of mathematical expectation, put Equation (2-3) and (2-4) into Equation (2-2), and then got Equation (2-5) [18]:

$$\begin{aligned} \mathrm{VaR} &= \mathrm{E}[\omega_0 (1+\mathrm{R})] - (\omega_0 + \mathrm{R}^*) \\ &= \mathrm{E}\,\omega_0 + \mathrm{E}\,\omega_0 (\mathrm{R}) - \omega_0 - \omega_0 \mathrm{R}^* \\ &= \omega_0 + \omega_0 \mathrm{E}(\mathrm{R}) - \omega_0 - \omega_0 \mathrm{R}^* \\ &= \omega_0 \mathrm{E}(\mathrm{R}) - \omega_0 \mathrm{R}^* \\ &= \omega_0 [\mathrm{E}(\mathrm{R}) - \mathrm{R}^*] \qquad (2\text{-}5) \end{aligned}$$

Through series of mathematical transformations, then the Equation (2-5) is the VaR of the portfolio. According to Equation (2-5), if the value of R^* at confidence level can be identified, and then the value of VaR will be calculated.

2.1.2 The Special Case

If the future value of the portfolio can be in line with the normal distribution [7][8][9][17][19][22], then the above VaR Equations can be simplified to the process of finding the standard deviation of the portfolio [18].

Let the future value of the portfolio R obeys normal distribution with mean $\mu\Delta t$ and variance $\sigma^2\Delta t$ [18]. It is R~N（$\mu\Delta t$， $\sigma^2\Delta t$）. Then $\frac{R-\mu\Delta t}{\sigma\sqrt{\Delta t}}$ is a standard normal distribution with mean 0 and variance 1. That is $\frac{R-\mu\Delta t}{\sigma\sqrt{\Delta t}}$ ~N（0，1）， and the probability density function is shown in Equation (2-6):

$$\Phi(x)=\frac{1}{\sqrt{2\pi}}e^{-\frac{x^2}{2}} \qquad (2\text{-}6)$$

If R is in line with the normal distribution, to find the value of R* at a given confidence level c just site on the point z in the standard normal distribution table, such as 1−c=$\int_{-\infty}^{-z}\varphi(x)d\omega$, then the confidence level c can be calculated by Equation (2-7):

$$R^* = -z\sigma\sqrt{\Delta t} + \mu\Delta t \qquad (2\text{-}7)$$

Put the Equation (2-7) into Equation (2-5), then can got Equation (2-8) [18]:

$$\begin{aligned}\text{VaR}&=\omega_0 \text{【}E(R) - R^*\text{】}\\ &=\omega_0(\mu\Delta t - R^*)\\ &=\omega_0(\mu\Delta t + z\sigma\sqrt{\Delta t} - \mu\Delta t)\\ &=\omega_0\, z\sigma\sqrt{\Delta t} \qquad (2\text{-}8)\end{aligned}$$

To calculate the VaR by using Equation (2-8), the key point is the calculation of standard deviation in Equation (2-5).

2.2 The Current Methods used of VaR

2.2.1 The Known Conditions

Generally two known conditions are given when calculating the VaR. First is the confidence level. The definition of confidence level is the credibility of the results of the probability measurement. The confidence level selection may reflect with investor’s attitudes on risk. The higher degree of risk aversion means the more losses cost, so more capitals required for compensate to the losses, and then the confidence level is set higher. For example, the Bank Trust uses 99% confidence level; Chemical and Chase bank uses 97.5% confidence level; Citibank uses 95.4% confidence level; Bank of America and JP Morgan Bank use 95% confidence level [15][16]. Second is the period of holding time. The longer period of holding time means the higher volatility of the value of portfolio [13]. Generally, the period of holding time can use one day, or one week, or 10 days, or two weeks, and or one month and so on. According to the proposal proposed by Basel Committee in 1995, financial institutions can use the 99% confidence level and the time of analysis was limited to ten working days [15], in other words, two weeks.

2.2.2 Assumptions

Usually, the VaR method assumes two conditions. First is the market efficiency hypothesis. Second, assuming market volatility is random and there is no autocorrelation [13][14][19]. In general, use mathematical models to analyze economic phenomenon must follow those assumptions.

2.3 Calculation Methods

Currently, there are many methods can calculate VaR. From the point of view of reference set, those methods can be divided into three categories: parametric methods, semi-parametric methods and non-parametric methods [11][13][14][16][18][25]. A common feature of these methods is calculating quantiles by using distributions of profits or losses in future, and then obtaining the values of VaR indirectly [25]. Therefore, from this point of view, these methods also can be referred to indirect methods [34]. The parameter method mainly refers to the Analytic Variance-Covariance approach [16]. The core of approach based on estimating of variance-covariance matrix of returns or losses. One of the most popular and famous representative methods is the Risk Metrics method which developed and used by JP Morgan Bank [16][24][25]. The semi-parametric method considers on extreme case because of partial peak heavy tail is the feature of the loss distribution. So it is using the extreme value theory such as the Heavy Tail Model [26]. The non-parametric method does not make assumptions for distributions and mainly be divided into historical simulation method and Monte Carlo simulation method [27].

2.3.1 The Analytic Variance-Covariance Approach

The core concept of the method is based on the estimation of the Variance-Covariance matrix [28][43] of the assets. The Variance-Covariance method originates a probability distribution of the hidden risk values through relative simple computations. The advantage of this method is simpleness [29]. There is a simple example of the Variance-Covariance method, suggest that calculating the VaR of a single asset, where the hidden risk values are in line with normal distribution with a mean of £100 million and a monthly standard deviation of £10 million. Set the confidence level as 95% and evaluate the value of asset will not downside below £60 million or raise above £100 million in future [27][30].

There are four steps when using this method to address the risks. First, it requires the user to take each individual asset in portfolio and translate the asset on standardised instrument. Second, each asset is explained as a set of positions in the standardised instrument which was indicated in the step one. The third step is the key point of the whole process of the method because of once the asset in portfolio has been identified on the standardised instrument and then the user has to assess the variances and covariance by searching historical data. The final step is computing the VaR by using the weights from the second step and the variances and covariance from the third step [27][28][33][52][61].

The JP Morgan’s Risk Metrics method [16] is the most representative method of Variance-Covariance method. There are two important assumptions of this approach.

First this method is the linear assumption [58]. Which means the relationship between value changes of an asset in a given period of time and the returns of its risk factors is in line with a linear changing. That is:

$$\Delta\omega = \sum \delta_k \Delta S_k / S_k \qquad (2\text{-}9)$$

Second is the normal distribution assumption. Which means the returns of risk factors: R$_s$= $\Delta S_k / S_k$ are in line with the normal distributions. Which is denoted as R~(μ , $\sum$). Assume that the returns of a portfolio is in line with normal distribution in future, and Σ is the N*N covariance matrix [31][32].

2.3.1.1 The Method Introduction

The basic ideas of the Variance-Covariance approach are: First, use the historical data to obtain the variance, standard deviation and covariance [34] of the returns of a portfolio; second, assume that the returns of a portfolio is in line with a normal distribution in future, and then it can be obtained the threshold which reflects the degree of deviation from the mean to the returns of the distribution with a certain confidence level; finally, deduced the VaR values which associate with the assets at risk or potential losses [33][41][39].

2.3.1.2 Application Conditions

This method suitable used for large-scale assets [34] and portfolios which contain a few financial derivatives such as option. From this view point of time, it suitable for the short term VaR measurement. This method operates easily because of it only requires the current market price and risk data [34][35] which means volatility. It also can be used to measure market risks of all financial assets, such as bank's credit risks [15], operational risks and so on. It is good quantified based on the financial risks. So it can be used for optimize the allocation of financial assets, risk assets management, the analysis of bank's strategic business decision making processes, performance evaluations such as risk-adjusted rate of returns, etc.

2.3.1.3 Advantages and Disadvantages

The advantages and disadvantages of the Variance-Covariance approach are shown in Table 2.1.

Table 2.1: Advantages and disadvantages of Variance-Covariance approach.

Advantages	Disadvantages
✓ Easy Calculation: just a few minutes it can calculate the entire bank's risk exposure.	✧ Assume that the portfolio return is in line with the Normal Distribution. This means if the return distribution is abnormal and then the computed VaR will be much lower than the true

	Value at Risk [58].
✓ According to the Central Limit Theory, this method also can be used even if the distribution of risk factor is not in line with the normal distribution. As long as the risk factors are large enough [55] and independently of each other.	✧ Assume that the multivariate risk factors are in line with Lognormal Distribution. Therefore, it cannot handle ‘‘the heavy tail’’ distribution [58].
✓ No pricing model needed. It just needs the Greek alphabet system which can be got directly from bank’s current system.	✧ The estimation of correlation between the volatility and income of the risk factors is needed.
✓ Be easy to introduce incremental VaR methodologies.	✧ The Taylor theory can be used for approximately on behalf of the security income. However, the second-order expansion cannot fully reflect to the risk of option in some cases.
	✧ It cannot be used for sensitive analysis.[55]
	✧ It cannot be used for derive the confidence interval of VaR.[58]
	✧ The input error. Even the assumption of distribution is correct, the final result still possible be wrong because

	of if used the incorrect variance and covariance.

2.3.2 The Semi-Parametric Method

The value of VaR calculated by the Monte Carlo method often underestimated and with large errors if the probability distribution of portfolio is not in line with normal distribution [36][37]. The distribution of extreme return is particularly important because of the VaR analysis heavily relies on the rate of extreme return. The semi-parametric method is developed to solve the heavy tail problem [58] of the probability distribution, so it is also called the Heavy-tail method.

Assume that the function distribution of returns rate R is F(R), when $R \to \infty$ and under the moderate regular condition, then F(R) has Second-order Expansion shown in Equation (2-10):

$$F（R）=1-Br^{-a}L+Cr^{-b} \qquad (2\text{-}10)$$

Where B, C, a and b are parameters. The main parameter is a, in other words, a is the tail index which value is the size of the tail [38][58].

2.3.3 The Historical Simulation Approach

The historical simulation method is a simple way to compute the VaR of numbers of portfolios by using with the historical data of a specify asset in portfolios [33].

2.3.3.1 Method Introduction

The basic ideas of the historical simulation approach are: first, weight the asset at the current time to re-simulate the history of the portfolio [33] by using of the actual rates of return on assets [40] in the past period of time; second, arrange simulated portfolios from low value to high value in order to get the overall distribution of the virtual incomes; finally, the VaR at a given confidence level will be obtained from the distribution [33][40][42].

2.3.3.2 Application Conditions

The historical simulation method simulates the samples which picked up from historical data. Therefore, it is not need to assume any distributions of rates of return and value changes of portfolios [41].

2.3.3.3 Advantages and Disadvantages

The advantages and disadvantages of the historical simulation method are shown in Table 2.2.

Table 2.2: The Advantages and Disadvantages of Historical simulation method.

Advantages	Disadvantages
✓ No need to assume that the distribution of risk factors.[45]	✧ Entirely depend on the specific historical data [39]. This means that

	the extreme market condition [43] will be ignored because of it was not included in data, or distorted for certain purpose.
✓ No need to estimate volatility [45] and correlation. Which have already been hinted into the daily market factors data.	✧ It cannot consider the impacts of market structural changes, such as the birth of Euro in June 1999 [43].
✓ The method can handle heavy tail distribution and other extreme situations as long as there are sufficient data can use.	✧ The calculated VaR may bias and imprecise if the involved historical data not long time enough [33].
✓ It can directly summarize data in different markets.	✧ It cannot be used for sensitive analysis.
✓ It allows users to calculate the confidence interval of VaR method.	✧ The calculation is not effective if the portfolio contains complex securities [44]. It is an effective method to compute VaR for an individual asset. It can handle each asset in portfolio but cannot analyze them together.
✓	✧ The historical data has its own trends. In the historical simulation method all data in the equal period of time are weighted equally during the process of computing the VaR. But in

	different historical time the data are influenced by various factors such as the price changes, the waves of currency exchange rate and so on. Those factors lead to the computed VaR deviating the actual VaR. [33][43][58][67]
✓	✧ The historical simulation method is difficult to compute the VaR for a new asset or when it faces a new market risk because this method depends on the historical data of the specify asset. If there is no historical data that can be used, the method cannot handle the situation.

Boudoukh, Richardson and Whitelaw presented a variant to improve the historical simulation in their book in 1998, where they gave different weights to the data which were picked up from different periods of time [7]. In other words, they defined that the recent data weighted more than distant data by using a decay factor in their time weighting mechanism [7]. For example, set the decay factor as 1 to the recent data, and then the distant data will be set as 0.9 or less.

2.3.4 The Monte Carlo Simulation Approach

2.3.4.1 The Method Introduction

The basic idea of Monte Carlo simulation is repetition of financial variables and covers all possible random process situations [45]. If these variables are in line with the predetermined probability distribution, and then the process of Monte Carlo simulation is reproducing the value distributions of portfolio.

Use Monte Carlo simulation to calculate VaR has three basic steps. The first step is scenario generation. Select the stochastic processes and distributions of changes of market factors and estimate the corresponding parameters [68]. And then simulate the path of market factors changing and try to build up the scenarios of market factors changing in future [67]. The second step is valuing portfolio. Calculate portfolio's values and changes of the market factors in each scenario by using pricing formulas or other methods. The third step is evaluation the VaR. It relies on the simulation results of distribution of portfolio value changing to calculate the VaR at a given confidence level. [40][45][53][61][67][68][71][73].

2.3.4.2 Application Conditions

The Monte Carlo simulation can be considered as the best way to calculate VaR comparing with other methods. It can effectively deal with problems which other methods cannot handle. For example, the non-linear pricing risk, volatility risk, event risk, model risk, the variance changing over time, heavy-tail distribution, extreme scenarios and even the credit risk [58][67].

2.3.4.3 Advantages and Disadvantages of the Monte Carlo Simulation

The advantages and disadvantages of Monte Carlo Simulation are shown in Table 2.3.

Table 2.3: Advantages and Disadvantages of Monte Carlo Simulation.

Advantages	Disadvantages
✓ It can be applied to all kind of distributions.	✧ Some situations are not included into the distribution [45].
✓ The simulation model can contain any complex portfolios.	✧ This simulation is very complex and high depending on abilities of large amount of computations.
✓ It allows user to calculate confidence intervals of VaR.[45]	
✓ It allows user to perform sensitive analysis and stress tests if required [45][73].	

For the selection of those three methods above, users need to consider many factors, such as the ease of data collection, ease of method, the calculation speed, the market stability and the ability of assumption [46]. But both of those methods have a common disadvantage, that is they cannot reflect the degree of losses in case the market emergency situation occurred, such as the Asian financial crisis and the world financial crisis since 2008. From this point of view, a complementary approach is necessary and which is the stress testing approach [73].

2.3.5 The Stress Testing Approach

The Stress testing approach can be regarded as ‘disposable’ or ‘limited times’ Monte Carlo simulation [73], users can choose the path of return on asset without any historical data [47]. Therefore, it is an approach which can avoid depending on historical data in theory. The Stress testing approach assumes a value of extreme changing of assets, and then calculates the changing values of portfolio for those hypothetical extreme changes [73]. The selection of these extreme changes often based on happened crisis in history, but there is no clear selection standard rule. In other words, it is not ideal method to use in practice. That means the stress testing approach is usually used as a complement to the VaR approaches rather than used alone. So the completely risk management is constituted by the VaR approaches plus the Stress testing approach. Under normal market conditions, the VaR approaches capture revenue opportunities. Otherwise, the Stress testing approach reflects the extent of losses in case of unexpected events occurred or the market is in a period of confusion time.

2.4 The Applications of VaR Model in Financial Risk Management

The various calculation methods and basics of VaR were discussed, the application of VaR will be introduced in this part.

2.4.1 The Development of VaR Applications

The VaR becomes an active risk management tool in recent years. It helps institutions who mastered the VaR tool to balance the relationship between risk and return. The economic capital can be used as a function of business risk and adjusted performance according to trader's evaluation [48]. In the most advanced institutions, VaR approach is used to determine the scope of competitive advantages, or adjust the departments which risk may increasing internal the company.

The process of VaR development is shown in Figure 2.2 [49][55]. It can be seen that the companies' negative activities when they were facing risks in history and the changes of their activities, finally, VaR approach is used to control risks actively.

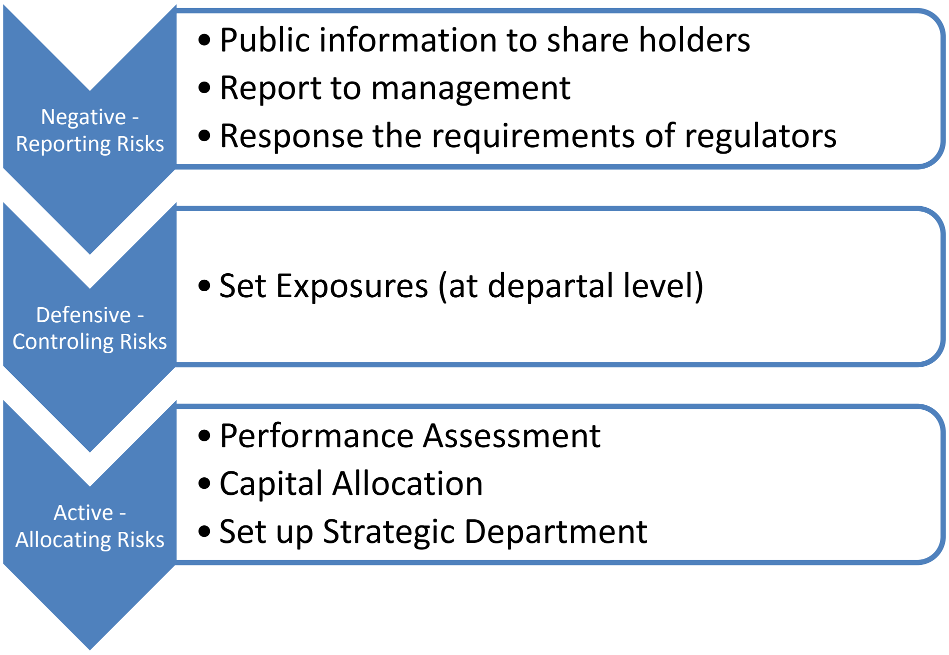

Negative - Reporting Risks
Public information to share holders
Report to management
Response the requirements of regulators
Defensive - Controling Risks
Set Exposures (at departal level)
Active - Allocating Risks
Performance Assessment
Capital Allocation
Set up Strategic Department

Figure 2.2: The development of VaR.

2.4.2 The Application of VaR

First, VaR is an information disclosure tool. Second, VaR is a risk management tool. Finally, VaR is an active management tool which used for self-management configuration. The active risk management contains strategic decision making, performance evaluations and capital allocation [50]. The capital allocation means the company allocates capital into products, business projects and a variety of transactions.

2.4.2.1 Risk Disclosure

Disclosure of the market risks is one of three pillars of guidelines of the credit risk which was argued by the Basel Committee [15]. The Basel Committee considered that public information about market risk is an effective way to achieve the market discipline. If the meaningful information are provided, and then investors, depositors, lenders and the both sides of traders can impose strong market discipline on financial institutions [15][33] together, which can make sure they manage their trading and derivatives along with established business objectives with the prudent manner. Risk Disclosure cannot only provides valuable reference information to investors, but also brings great benefits to the companies who had already disclosed their risk information. Because of comparing with other companies without risk disclosure, investors prefer to buy stocks from the companies with risk disclosure. [15][27][33][43][51].

2.4.2.2 Financial Regulation

According to the provisions which were formulated by the Basel Committee, the bank's capital is determined by the calculated VaR risk measurement models, while it gave suggestions and regulations on the using of calculation model. Based on the results of VaR, financial regulatory authorities can calculate the required minimum margin of financial institutions in order to avoid the market risk [15]. The VaR approaches also can be used by regulatory authorities to monitor the risks of banks and other financial institutions [33]. The value of VaR has become the uniform standard to measure the risks of financial intermediaries.

2.4.2.3 Risk Control

The VaR approaches are used for risk control, which are needed for risk management by banks and other financial institutions themselves, on the other hand which are also the responses for the financial regulatory requirements. In 1995, the Basel Committee had approved the banks use approved and accredited internal models to calculate VaR. On this basis, multiplied by 3 [15], and then the capital amounts meet the requirements of market risk. By 3 can provide necessary buffers because of the standard VaR is difficult to capture and loss probability will high in case in extreme risk market. Compare with using the Basel’s standardized approach for capital required, banks are using VaR methods can save up to 60% - 85% [15] for the capital required. This allows banks obtain advantages in compliance with regulatory capital requirements. Banks can raise quality and the operational efficiency of working capital through developing appropriate investment strategies, timely adjustment of the portfolio to diversify and avoid risks by using calculated values of VaR. The strict VaR management can significantly prevent losses in some financial transactions.

2.4.2.4 Performance Evaluation

The performance assessment of traditional traders and business departments depend on the returns on investment. Traders may disregard great risk and pursuit of high returns in the financial investment. So it may lead to a large number of risk occurrences if just simply use the returns on investment to carry out performance evaluation. Due to the necessary for the normal operation, companies have to limit trader’s possible excessive speculations [51]. Therefore, it is necessary to introduce the performance evaluation [52] of risk factors.

The VaR approaches ensure that management to adjust traders' level of income by according to different risks they are facing. Traders in different markets usually get different benefits due to different volatilities of markets where they are. This distinction does not come from levels of their operations. The values of VaR can be used as standard and reasonable evaluations of the investment performances. The Risk Adjusted Return on Capital (Raroc) [15][17] is a kind of more scientific design of measurement of performance evaluation. The formula is: Raroc = returns on investment / values of VaR. It can be seen from the formula, when traders engaged in high risk investments, even got high returns of investment, the results of the values of evaluation Raroc will not be high because the corresponding values of VaR are also high. Meanwhile, the VaR method is adjusted to help to reduce the existing moral hazard or adverse selection behaviors of traders by according to the different market risks.

2.5 Advantages and Disadvantages of VaR

Overall, VaR becomes the popular financial analysis tool used in a variety of companies or organizations. Therefore, to better understanding its advantages and disadvantages are important when use VaR.

2.5.1 The Advantages of VaR

VaR is built up by statistical approaches to measure the risk comprehensibly and integrality. Compare with other subjective and artistic traditional risk management methods, the VaR methods greatly increase the scientific of the risk management. Its advantages include:

- The VaR methods mixed the sizes and likelihoods of the expectation of future losses [52]. In other words, VaR lets the investors know the sizes and likelihoods of the losses when the losses occurred. Managers can more clearly understand their financial institutions through analyzing the results of VaR. It can show expect performances of the organization at different degree of risks just by adjusting the confidence levels. Therefore, the VaR methods facilitate the different management needs for different organizations.

- The VaR methods also can comprehensive measure various market risks such as interest rate risks, exchange rate risks, equity risks, commodity price risks and financial derivative instruments risks [53]. Therefore, VaR methods allow financial institutions to use values of VaR as the index which represent the risk profiles of their products and themselves. The VaR methods greatly facilitate to the exchanges of risk information [53] between departments and leaderships internal organizations. Meanwhile, the regulators also can propose uniform requirements conveniently to financial institutions through analyzing values of VaR.

- Unlike other risk management methods, the VaR methods pre-calculate risks before they are going to occur [54]. VaR can calculate the risks of individual asset and portfolios. Which the traditional financial risk management methods may cannot do.

2.5.2 The Disadvantages of VaR

On the other hand, VaR methods also have some limitations. In other words, their disadvantages are shown in below:

- There are some flaws of the principles and statistical estimations [53] of the VaR methods themselves. The VaR methods' calculations are based on the objective probabilities of risks [54]. This means the VaR in order to estimate the maximum possibility of losses occurrence by using statistical analysis of performance of the asset or portfolios in the past and then predict their prices volatility and correlation [33]. For example, the parameter methods, historical simulation method and stochastic simulation methods (Monte Carlo simulation method) [47][54]. The full financial risk management includes risks identification, measurement and control. To control risks is the ultimate goal in financial risk management [55] which involves the risk managers' risk appetites and risk price factors. Therefore, if a method only based its concepts on probability of risk may lead to the losses, or a method only concerned with the statistical characteristics of the risk, those are not the systematic risk management [33]. Because of itself of the probability of risk does not reflect to the wishes or attitudes of institutes to economic risks. So, the VaR methods are not comprehensive in financial risk management, managers cannot depend on those methods at all to balance all the risk factors before risk occurrence.

- The VaR methods are usually used to measure the market risks in normal market conditions which mean the VaR cannot handle situations in extreme market conditions [73]. In normal market conditions, the VaR methods are used much more effective due to there have a large amount of the historical data of transactions can be used for analysis. However, when the market is going away the normal conditions, the historical data of transactions become scarce, especially when the market crisis is occurred [56][57]. Therefore, those factors make users impossible to use the VaR methods effectively measure market risks at this time.

- According to the stringent requirement on the data of the VaR methods, the risk values are easily obtained when prices of frequent transactions and portfolios are easily got. But the measure risks ability is severely restricted when use VaR methods to analyze illiquid assets which with the lack of daily market trading prices data such as bank loans [58]. Therefore, users need to divide the illiquid financial products [59] into strong liquidity portfolios and then analyze the risks by using VaR methods.

- The main purpose of VaR methods is measuring the market risks, but it is difficult to reflect on liquidity risks, credit risks, operational risks and legal risks [59]. Therefore, VaR is a tool used to value the interest rates, the exchange rates and the other different types of market risks that may occur during the economic activities of financial institutions. But the valuation is narrowly focusing on market risks which cannot represent all risks avoided by the financial institutions and investors.

- Finally, from the view of technical perspective, the values of VaR indicate the maximum losses [60] at a certain degree of confidence level. The VaR cannot guarantee the values of true losses will exceed the values of VaR. For example, assume that one day VaR=$ 100 million at the 99% confidence level. Which means there still has 1% possibility that the losses will be more than $ 100 million [59] in a day. In case this situation happened, it will be catastrophic damage to the company.

Overall, the VaR methods are highly depending on historical data, so, the fundamental flaw of VaR is the history cannot always as a good guideline for future. The VaR approaches cannot cover every risks in the financial risk management, so users still need to integrated using of a variety of other qualitative and quantitative [61] methods.

2.6 Case Study of VaR Approaches

This case study uses Monte Carlo method to calculate the values of VaR. The case study starts with 200 days of the Shanghai Composite Index closing price data from June 2000 to November 2000. And then use the Monte Carlo simulation to calculate the next trading day's VaR of the Shanghai Composite Index [102]. Set the confidence level as 95% and select the day holding period. Use the geometric Brownian motion as the stochastic model to reflect of changes in the Shanghai Composite Index, and its discrete form can be expressed as follows [29]:

$$\Delta S_{t+1} = S_t(\mu\Delta t + \sigma\varepsilon\sqrt{\Delta t}) \qquad (2\text{-}11)$$

$$\Delta S_{t+1} = S_{t+1} - S_t \qquad (2\text{-}12)$$

Where: S_t is the Asset price at time *t*;

S_{t+1} is the Asset price at time *t+1*;

μ is the mean of return on assets;

σ is the volatility of return on assets;

ε is random variables.

Generally, Monte Carlo simulation method uses the standard deviation to measure the volatility of returns by assuming that is in line with normal distribution. σ is the standard deviation of the return of SCI, ε is random variable in normal distribution. Then we divide the one day holding period into 20 equally period of time. S_t is the SCI at initial period of time and S_{t+i} is the SCI at time *t+i*. ΔS_{t+i} is the variation of SCI at each period of time. The mean and the standard deviation [102] of SCI returns at each period of time are represented by $\frac{\mu}{20}$ and $\frac{\sigma}{\sqrt{20}}$. So the SCI at time t+i should be:

$$S_{t+i} = S_{t+i-1} + \Delta S_{t+i}$$

$$= S_{t+i-1} + S_{t+i-1}(\frac{\mu}{20}\Delta t + \frac{\sigma}{\sqrt{20}}\varepsilon_i\sqrt{\Delta t}) \quad \text{(2-13)}$$

Where $i = 1,2,3,\ldots,20$.

Here are the specific steps of calculating the VaR of SCI by using of Monte Carlo simulation method.

The first step is estimating mean and the standard deviation [102]. Use mean μ and the standard deviation σ of the 200 days SCI to calculate the mean $\frac{\mu}{20}$ and standard deviation $\frac{\sigma}{\sqrt{20}}$ [102] of each period of time.

The second step is generating the random numbers [55]. Produces 20 standard normal distributed random number such as $\varepsilon_1, \varepsilon_2, \varepsilon_3, \ldots \varepsilon_{20}$.

The third step is simulating a possible path of price changes of SCI. Substitute S_t, $\frac{\mu}{20}$, $\frac{\sigma}{\sqrt{20}}$ and ε_1 into Equation (2-13), then can get the SCI at time *t+1*:

$$S_{t+1} = S_t + S_t(\frac{\mu}{20}\Delta t + \frac{\sigma}{\sqrt{20}}\varepsilon_1\sqrt{\Delta t}) \quad \text{(2-14)}$$

And can be obtained as follows:

$$S_{t+2} = S_{t+1} + S_{t+1}(\frac{\mu}{20}\Delta t + \frac{\sigma}{\sqrt{20}}\varepsilon_2\sqrt{\Delta t}) \quad \text{(2-15)}$$

$$S_{t+3} = S_{t+2} + S_{t+2}(\frac{\mu}{20}\Delta t + \frac{\sigma}{\sqrt{20}}\varepsilon_3\sqrt{\Delta t}) \quad \text{(2-16)}$$

$$\vdots$$

$$S_{t+20} = S_{t+19} + S_{t+19}\left(\frac{\mu}{20}\Delta t + \frac{\sigma}{\sqrt{20}}\varepsilon_{20}\sqrt{\Delta t}\right) \quad \text{(2-17)}$$

$$=S_T$$

Where $S_{t+1}, S_{t+2}, S_{t+3}, \ldots, S_{t+20}$ is a possible path of price changes of SCI and S_T is the possible closing price of SCI.

The fourth step is simulating 10,000 possible closing prices. Just repeat the second step and the third step 10,000 times to obtain the 10,000 possible closing prices of SCI.

$$S_T^1, S_T^2, S_T^3, \dots, S_T^{10000}$$

The final step is calculating the VaR. Sort in order of $S_T^1, S_T^2, S_T^3, \dots, S_T^{10000}$ from small values to large values, and find out below the 5% quintiles $S_T^{min5\%}$, and then calculate the VaR at 95% confidence level. It is shown in Equation (2-18):

$$VaR = S_t - S_T^{min5\%} \qquad (2\text{-}18)$$

2.7 Summary

This Chapter reviewed the background and development of the Value-at-Risk methods such as the Variance-Covariance approach, the Historical simulation approach, the Monte Carlo simulation approach and the Stress Testing approach. It also presented the advantages and disadvantages of these methods and the applications of VaR methods in financial risk management. It pointed out limitations of these methods. Finally this Chapter analyzed a case study of the Monte Carlo approach to Shanghai Composite Index.

Chapter 3: Monte Carlo Simulation

The Monte Carlo method is also known as a random sampling or a statistical test method [45] [56] [59]. Since half a century ago, Monte Carlo method was proposed as an independent method due to the development of Science and Technology and the invention of computers. The first application of Monte Carlo method was during in testing and developing of nuclear weapons [57]. Compare with other calculation methods, Monte Carlo method is very different. Monte Carlo method based its ideas on the theory of probability and statistic. The Monte Carlo method is used widespread now because of it can realistic simulates the characteristics of things and simulates physical experiments to solve problems which are difficult to be done by other numerical methods.

3.1 Basic Concept

Although Monte Carlo method was proposed in the past of decades, its basic ideas are much older. It can be said that the ideas are based on two typical probabilities examples.

First example is Comte de Buffon's experiment which in order to evaluate the value of π by using of needles in 1768 [60]. In the experiment, people arbitrarily pelt a needle which the length was $2l$ on the ground, and then used the frequency of intersecting between a needle and a group of parallel lines and their distances were $2a(l < a)$ to instead of the probability$P$ [60][61]. It is shown in Figure 3.1.

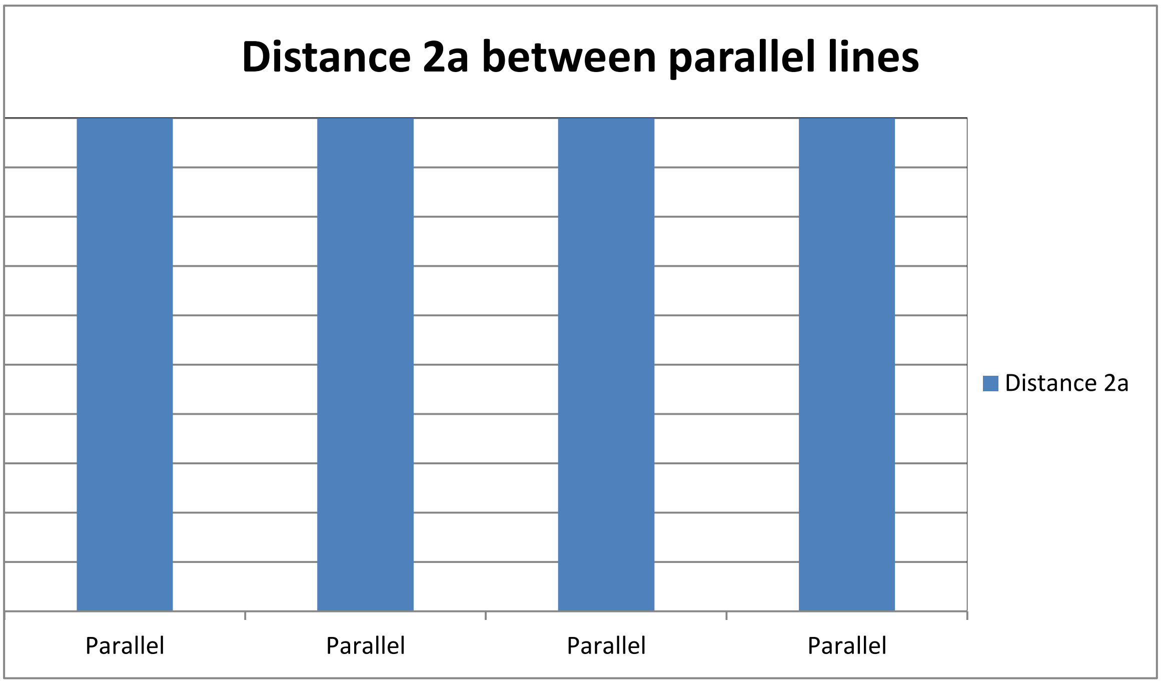


Figure 3.1: Buffon’s Experiment.

Then used the exact relationships:

$$P = \frac{2l}{\pi a} \qquad (3\text{-}1)$$

To calculate the values of π:

$$\pi = \frac{2\mathrm{l}}{\mathrm{aP}} \approx \frac{2\mathrm{l}}{\mathrm{a}}(\frac{\mathrm{N}}{\mathrm{n}}) \qquad (3\text{-}2)$$

Where N is the total numbers of needle has pelted. n is the number of times that the needle and parallel lines intersect. This is the Buffon’s experiment in the classical probability theory.

Some researchers conducted same experiments and the results are shown in Table 3.1.

Table 3.1: Results of some Buffon's experiments [61].

Resear-chers	Year	Total Numbers of Pelting	Values of π in experiments
Wolf	1850	5000	3.1596
Smith	1855	3204	3.1553
Fox	1894	1120	3.1419
Lazzarini	1901	3408	3.1415929

Another typical example is the shooting game. Let r represents the distance between shooters and targets, $g(r)$ represents the corresponding hit scored and $f(r)$ represents the distribution density function of points of bullet contacted. The shooting performance should be [64]:

$$g = \int_0^\infty g(r)f(r)dr \qquad (3\text{-}3)$$

In probability language, g is the mathematical expectation of a random variable$g(r)$ [64].

That is

$$g = E[g(r)]. \qquad (3\text{-}4)$$

Assume that athlete shoots N times, the points of bullet contacted each time are $r1, r2, r3, \dots, rN$ and the hit scores each time are $g(r1), g(r2), g(r3), \dots, g(rN)$.

Therefore, the arithmetic mean $\bar{g}_N$ represents the performance of the athlete.

That is:

$$\bar{g}_N = \frac{1}{N}\sum_{i=1}^{N} g(r_i) \qquad (3\text{-}5)$$

It can be seen from the above two examples, Monte Carlo method's basic idea is when the solution to the problem is the probability of an event, or a mathematical expectation of a random variable [64], or the value relevant to probability or mathematical expectation [65], through finding out the frequencies of occurrence of the event, or the arithmetic mean of the random variables to obtain the solution to the problem. In other words, the mathematical expectation [65] is the probability of an event [65] when the value of the random variable is only 1 or 0, or the probability of the event is the mathematical expectation of a random variable which value is only 1 or 0.

So Monte Carlo method is using of randomized trials to calculate integrals [66][68]. That means the integrals is the mathematical expectation of the random variable$g(r)$ and it is in line with density distribution of certain function $f(r)$.

That is:

$$g = \int_0^{\infty} g(r)f(r)dr \qquad (3\text{-}6)$$

In probabilistic language, sample numbers of samples from the distribution density function $f(r)$ are $r_1, r_2, r_3, \ldots, r_N$. The integral's approximates is the arithmetic mean of the corresponding random variables $g(r_1), g(r_2), g(r_3), \ldots, g(r_N)$.

That is:

$$\bar{g}_N = \frac{1}{N}\sum_{i=1}^{N} g(r_i) \qquad (3\text{-}7)$$

In order to obtain high degree accuracy approximation will require doing a lot of time experiments. It is difficult or even impossible to make large number of tests manually. Therefore, Monte Carlo method had been rarely used until the rapidly development of computers [67].

3.2 Monte Carlo Simulation

The computational simulation is a process of transfer the experiments to mathematical programs. We still use the above two cases as examples to explain this process.

3.2.1 Simulation of Buffon's Experiment

Assume that the positions on the ground of pelted needles can be described by a set of parameters [67] (x, θ), where x is the coordinates of the needle midpoint and θ is the angle between the needle and the parallel lines. Those are shown in Figure 3.2.

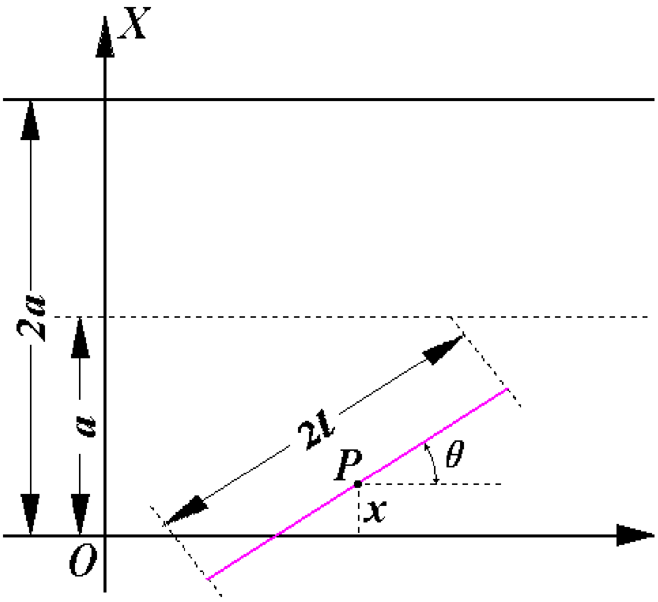


Figure 3.2: The position between the needle and the parallel lines [69].

Each time of needle is casted randomly, that means x and θ are taken arbitrarily. But the range of x is limited to $[0, a]$ and angle θ is limited to $[0, \pi]$. In this case, the needle and parallel lines intersected is in line with the mathematical conditions:

$$x \leq l \cdot sin\theta \qquad (3\text{-}8)$$

X is an arbitrary value in $[0, a]$, which means x is in line with the uniform distribution in $[0, a]$ and the distribution density function is:

$$f_1\,(x) = \begin{cases} 1/a, & 0 \leq x \leq a \\ \quad 0, & others \end{cases} \qquad (3\text{-}9)$$

Similarly, the distribution densities function of θ is:

$$f_2\,(\theta) = \begin{cases} 1/\pi, & 0 \leq \theta \leq \pi \\ \quad 0, & others \end{cases} \qquad (3\text{-}10)$$

So the process of generating an arbitrary (x, θ) includes sampling x from $f_1(x)$ and sampling θ from$f_2(\theta)$. And then can get:

$$x = a\xi_1 \tag{3-11}$$

$$\theta = \pi\xi_2 \tag{3-12}$$

Where $\xi_1$and ξ_2 are random variables distributed in (0,1).

Each experiment of needle cast is actually transferred to process of sampling from two uniform distribution random variables [70] to obtain (χ, θ) by using computers. And then the definition of the random variables of needle and parallel lines intersected can be described as $\mathrm{s}(\chi, \theta)$, then we got:

$$s(\chi, \theta) = \begin{cases} 1, & x \leq l \cdot sin\theta \\ 0, & others \end{cases} \tag{3-13}$$

If pelt the needle N times, then we got:

$$\bar{S}_N = \frac{1}{N}\sum_{i=1}^{N} s(x_i, \theta_i) \tag{3-14}$$

And the Calculated value is the estimation of the probability P of the needle and parallel lines intersected. Then can get:

$$\begin{aligned} \mathrm{P} &= \iint s(x, \theta) f_1(x) f_2(\theta) dx d\theta \\ &= \int_0^{\pi} \frac{d\theta}{\pi} \int_0^{lsin\theta} \frac{dx}{a} \\ &= \frac{2l}{\pi a} \end{aligned} \tag{3-15}$$

So [71]:

$$\begin{aligned} \pi &= \frac{2l}{aP} \\ &\approx \frac{2l}{a\bar{S}_N} \end{aligned} \tag{3-16}$$

3.2.2 Simulation of Shooting Game

Set the points of impact are distributed shown in Table 3.2.

Table 3.2: Hit marks and Probabilities [72].

Target Marks	7	8	9	10
Probabilities	0.1	0.1	0.3	0.5

To start a simulation of randomized trial needs to select a random number ξ and then follow the steps as follows [72]:

$$\begin{tikzcd}
\xi \le 0.1 \arrow[r, "\le"] \arrow[d, ">"] & t\arg et7 \\
\le 0.2 \arrow[r, "\le"] \arrow[d, ">"] & t\arg et8 \\
\le 0.5 \arrow[r, "\le"] \arrow[d, ">"] & t\arg et9 \\
t\arg et10 &
\end{tikzcd} \qquad (3\text{-}17)$$

Thus, it has done a randomized trial and has got a score$g(r)$. And then can get an approximation of the athlete shooting performance after N trials. The process is shown in Equation (3-18):

$$\bar{g}_N = \frac{1}{N}\sum_{i=1}^{N} g(r_i) \qquad (3\text{-}18)$$

3.3 The Convergence and Error of Monte Carlo Method

The Monte Carlo simulation as a calculating method its general concerns are convergence and error [73].

As described above, Monte Carlo method uses the approximation as the solution of the arithmetic average of sub-simples such as $X_1, X_2, X_3, \ldots, X_N$ which are from random variables X. The Equation is:

$$\bar{X}_N = \frac{1}{N} \sum_{i=1}^{N} X_i \qquad (3\text{-}19)$$

From the perspective of the law of large numbers [74], if $X_1, X_2, X_3, \ldots, X_N$ are independent and in line with identical distribution [74] and have limited expectations $(\mathrm{E(X)} < \infty)$, then we can get:

$$P(\lim_{N \to \infty} \bar{X}_N = E(X)) = 1 \qquad (3\text{-}20)$$

This means the arithmetic mean $\bar{X}$ of the random variable X, when the sub-sample number N is large sufficiently, the $\bar{X}$ converges to its expectation value E(X) with probability 1 [74].

The central limit theorem which is well known in probability theory, gives the solution to solve error problem between the true value and approximation calculated by Monte Carlo method [74]. It argues that if $X_1, X_2, X_3, \ldots, X_N$ are independent and in line with identical distribution [74] and have a limited non-zero variance σ^2, then we can get:

$$0 \neq \sigma^2 = \int \left(x - E(X)\right)^2 f(x)dx < \infty \qquad (3\text{-}21)$$

f(X) is a distribution density function [74] of X, then we can get:

$$\lim_{N\to\infty} P\left(\frac{\sqrt{N}}{\sigma}\left|\overline{X}_N - E(X)\right| < x\right) = \frac{1}{\sqrt{2\pi}}\int_{-x}^{x} e^{-t^2/2}dt \qquad (3\text{-}22)$$

When and where N is large sufficiently, there has an approximate formula shown in Equation (3-23):

$$P\left(\left|\overline{X}_N - E(X)\right| < \frac{\lambda_\alpha \sigma}{\sqrt{N}}\right) \approx \frac{2}{\sqrt{2\pi}}\int_{0}^{\lambda_\alpha} e^{-t^2/2}dt = 1 - \alpha \qquad (3\text{-}23)$$

Where α is confidence, and $1 - \alpha$ is the confidence level.

The inequality is shown in Equation (3-24):

$$\left|\overline{X}_N - E(X)\right| < \frac{\lambda_\alpha \sigma}{\sqrt{N}} \qquad (3\text{-}24)$$

The inequality is based on the probability $(1 - \alpha)$ And the error convergence rate [74] in line with $O(N^{-1/2})$.

Typically, the Monte Carlo method error ε can be defined as:

$$\varepsilon = \frac{\lambda_\alpha \sigma}{\sqrt{N}} \qquad (3\text{-}25)$$

Where λ_α and confidence α are correspondence each other. When the confidence level has determined, the λ_α also can be determined by checking the standard normal distribution table.

There are a few common values of α and λ_α shown in Table 3.3.

Table 3.3: Some common values [75].

α	0.5	0.05	0.003
λ_α	0.6745	1.96	3

The error of the Monte Carlo method indicates two features. First, the fundamental difference between the Monte Carlo method and other numerical methods is its error. It is only probability error, which determined by the basic ideas of Monte Carlo method. Second, the mean square σ is unknown, so must use its estimate to instead of the mean square during simulation. That is where error from.

$$\hat{\sigma} = \sqrt{\frac{1}{N}\sum_{i=1}^{N} X_i^2 - (\frac{1}{N}\sum_{i=1}^{N} X_i)^2} \qquad (3\text{-}26)$$

It can be seen from the Equation (3-26). Obviously, when a given confidence level α has determined, the degree of error depends on ε, σ and N. To reduce ε, can try to through increasing N, or reducing the variance σ^2. In the case of the value of σ is fixed, the improvement of the accuracy of an order of magnitude increasing will need the number of tests N in two orders of magnitude increasing. Therefore, simply increasing N is not an effective way.

On the other hand, if we reduce the estimated mean square error σ and then the final result error will reduce. For example, if the mean square error has reduced half, and then the final result error is reducing half. The performance is equals to increase N four times.

Generally, the variance reduction techniques lead to the trend of time increasing of observation on a sub-sample. In other words, the variance reduction techniques reduce the total number of observed samples in fixed time. Therefore, users can evaluate their simulation by using time cost of observations on a sub-sample and variance. This is also the concept of efficiency in Monte Carlo method. Which is defined as $\sigma^2 \cdot c$. Where c is the time cost of observation on a sub-sample. Obviously, smaller $\sigma^2 \cdot c$ means more efficient of the Monte Carlo simulation.

3.4 The Characteristics of Monte Carlo Method

The Monte Carlo method has many characteristics as shown in below:

- The Monte Carlo method can realistically describe the characteristics of random nature things and physical experiments.

  In this sense, the Monte Carlo method can partly replace several physical experiments. It also can get results even though the physical experiments cannot handle. Use Monte Carlo methods to solve practical problems [44] can directly start with simulations of actual problems rather than start with the mathematical expressions or equations [75]. From this point of view, it can show the image the problem intuitively.

- The Monte Carlo method is only a little bit influenced by geometric constraints [44][74].

For example, calculate the function in area D_S where in s-dimensional space. The function is shown in Equation (3-27):

$$g = \int \underset{D_s}{\cdots} \int g(x_1, x_2, \cdots, x_s) dx_1 dx_2 \cdots dx_s \qquad (3\text{-}27)$$

If the described conditions of the geometric characteristics of D_S are given, N points such as $(x_1^{(i)}, x_2^{(i)}, \cdots, x_s^{(i)})$ can be uniformly generated from the area regardless how the shape of the particular region D_S is. Then we can get the Equation (3-28):

$$\bar{g}_N = \frac{D_s}{N} \sum_{i=1}^{N} g(x_1^{(i)}, x_2^{(i)}, \cdots, x_s^{(i)}) \qquad (3\text{-}28)$$

Where D_S is the volume of region. The numerical methods are difficult to do like this. In addition, there is no difficulty in principle when using the Monte Carlo method even the shape of image is very complicated.

- Problem's convergence rate has nothing to do with the dimension [75].

According to the definition of error, the convergence rate of Monte Carlo method is $O(N^{-1/2})$ when a confidence level given, and has nothing to do with the dimension. Any changes of dimension only influence on the estimated calculation and sampling time. That means it does not affect the error. In other words, in the Monte Carlo simulation, the total number of sub-samples N is irrelevant to the dimension *S*. The increasing of dimension leads to the increasing of the corresponding calculations, nothing effect on errors. This feature determines the Monte Carlo method suitable for multidimensional problems. When use general numerical methods to calculate the definite integral problems, the computation time increases with the dimension power. If run those computations on computer, it will take up a large amount of computer memory. For general numerical method, these problems are difficult overcame.

- It has capacity with simultaneously computing multiple programs and multiple unknowns [76].

For the problem which is needed to calculate the number of options, the Monte Carlo method does not need to calculate the options one by one like other conventional methods. It can calculate all options at the same time and the time cost almost same to calculate one option. For example, to calculate the plate geometry with shield of the uniform medium, it only needs to calculate the probability of certain kinds of thickness of penetration. The probabilities of other thickness can be obtained simultaneously just deal with a little more calculations of the results.

- It can easily determine errors.

For general calculation methods, they are difficult to give the errors between the calculated results and the true values. According to the error formula of the Monte Carlo method, errors can be calculated at the same time while the results calculated even in a very complicated Monte Carlo simulation model.

A problem often exists in general methods, which is the loss of significant digits. For those methods are difficult to solve this problem. But Monte Carlo method eliminates this problem.

- The implementation of Monte Carlo simulation is simply and easily [76].

When simulate Monte Carlo method on computer, the simulation can be divide into many chunks [76] and implemented parallel.

- Slow convergence [76].

The convergence rate of the Monte Carlo method is $O(N^{-1/2})$ and it is not easy to get a high accuracy approximation of the convergence rate. Compare with other methods, the performance of Monte Carlo method is not effective when calculating small dimension problems.

- Probability errors [71].

Monte Carlo method has probability errors due to the result is estimated at a certain confidence level.

3.5 Simulation of Specific Case

In this case, we use the Monte Carlo simulation method to simulate the Brownian motion [35] due to its characteristics. A visual simulator will be established in Matlab environment and will intuitively demonstrate the advantages of Monte Carlo simulation method.

In 1827, the Scottish botanist Brown observed irregular movement of pollens in the water through using a microscope [58]. Therefore, people argued that the random continuous motion of particles in a fluid medium as the Brownian motion. In 1908 the Langevin theory [58] had published to prove that the mean of displacement square of a Brownian particle is proportional to time. From the perspectives above, random walk is the essential of Brownian motion. This characteristic is suitable to use Monte Carlo simulation to show.

The function of Langevin theory is defined in Equation (3-29):

$$m\frac{d^2x}{dt^2} = -\alpha\frac{dx}{dt} + F(t) \qquad (3\text{-}29)$$

Where $F(t)$ is the power of fluctuation, $\alpha = 6\pi a\eta$ is the Stokes formula, a is the particle radius, η is viscosity. Multiplied by x of both sides of Equation (3-29) and then we can get Equation (3-30):

$$\frac{1}{2}\frac{d^2}{dt^2}(mx^2) - m\dot{x}^2 = -\frac{\alpha}{2}\frac{d}{dt}x^2 + xF(t) \qquad (3\text{-}30)$$

Obtain the mean of all particles in Equation (3-30). Because of the mean value is 0 of $F(t)$. Then we can get Equation (3-31):

$$\frac{d^2}{dt^2}\overline{x^2}+\frac{\alpha}{m}\frac{d}{dt}\overline{x^2}-\frac{2kT}{m}=0 \tag{3-31}$$

So the Equation (3-31) is the second order common coefficient non - homogeneous linear differential equation of $\overline{x^2}$. Then we can get Equation (3-32):

$$\overline{x^2}=\frac{2kT}{\alpha}t+C_1e^{-\frac{\alpha}{m}t}+C_2 \tag{3-32}$$

After approximation, and assume that all particles are in area of $x = 0$ when the time $t = 0$, and then we got $c_2 = 0$. Therefore we got Equation (3-33):

$$\overline{x^2}=\frac{2kT}{\alpha}t \tag{3-33}$$

The Equation (3-33) shows that the Lagevin theory argued the mean of displacement square of a Brownian particle is proportional to time.

As described before, the Monte Carlo method is a calculation solution by using of continuous generation of random number. The Monte Carlo method is very appropriate to the Brownian motion due to both of them have regularity of certain randomness and statistical.

For example, assume that a drunken man will walk N steps from the origin. Each step has randomness in the direction of the line and has nothing to do with the previous step and each step size is in the specified range. Then each step in the *x* and *y* directions are:

$$(\Delta x_1, \Delta y_1),\quad (\Delta x_2, \Delta y_2),\quad \ldots\ldots,\quad (\Delta x_n, \Delta y_n)$$

After walked N steps, the distance R between the location of the origin and drunken man is in line with the following relationship:

$$\begin{aligned} R^2 &= (\Delta x_1 + \Delta x_2 + \ldots\ldots + \Delta x_n)^2 + (\Delta y_1 + \Delta y_2 + \ldots\ldots + \Delta y_n)^2 \\ &= \Delta x_1{}^2 + \Delta x_2{}^2 + \ldots\ldots \Delta x_n{}^2 + 2\Delta x_1 \Delta x_2 + \ldots\ldots + \Delta y_1{}^2 + \Delta y_2{}^2 \\ &+ \ldots\ldots + \Delta y_n{}^2 + 2\Delta y_1 \Delta y_2 + \ldots\ldots \end{aligned} \qquad (3\text{-}34)$$

If walk is in line with true random, and then the probabilities of each step walk directions are same. After take large enough number of steps, the cross-terms in Equation (3-34) will be cancelled, then we can get:

$$R^2 \approx \Delta x_1{}^2 + \Delta x_2{}^2 + \ldots\ldots \Delta x_n{}^2 + \ldots\ldots + \Delta y_1{}^2 + \Delta y_2{}^2 + \ldots\ldots + \Delta y_n{}^2 \quad (3\text{-}35)$$

Then we get:

$$R^2 \approx nr^2 \qquad (3\text{-}36)$$

Where *r* is the step size. Where *n* large enough is the condition of Equation (3-36) established.

To visualize the Brownian motion we need to create a three-dimensional spatial coordinate. It is shown in Figure 3.3.

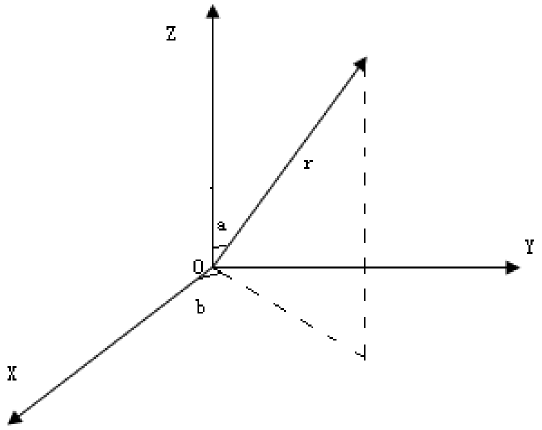


Figure 3.3: The three-dimensional spatial coordinates [68].

Then we use polar coordinates, they are shown in Equation (3-37):

$$X = r\sin a\cos b\,,\quad Y = r\sin a\sin b\,,\quad Z = r\cos a\,,$$

$$0 \le r \le +\infty\,,\quad 0 \le b \le 2\pi\,,\quad 0 \le a \le \pi\,, \qquad (3\text{-}37)$$

To allow the movement of these Brownian particles we need to give the moving directions and steps length. We choose step length as *ss=2* (*r=2*) and Define two angles *b=theta 1*, *a= theta 2* and then we give the motion directions in polar coordinates by using the random number generation function. The Monte Carlo simulation will run in the Matlab environment. First we simulate 200 Brown particles motions and then we track the moving of a single particle. Finally we record and draw the motion path.

The corresponding Matlab program codes are listed in below:

```
rand('state',sum(100*clock));     % Generating Random
ns=200;                           % Set Times
np=200;                           % Total number of
```

```
ss=2;                       % Set the step length
sl=zeros(np,3);
for a=1:ns
for c=1:np
theta1=rand*2*pi;        % Define the scope of theta 1
theta2=rand*pi;
sl(c,1)=sl(c,1)+(ss*cos(theta1))*sin(theta2); % Find the x component
if sl(c,1)>20
sl(c,1)=sl(c,1)-2*(ss*cos(theta1))*sin(theta2);
end
if sl(c,1)<-20
sl(c,1)=sl(c,1)-2*(ss*cos(theta1))*sin(theta2);
end
sl(c,2)=sl(c,2)+(ss*sin(theta1))*sin(theta2);
if sl(c,2)>20
sl(c,2)=sl(c,2)-2*(ss*sin(theta1))*sin(theta2);
end
if sl(c,2)<-20
sl(c,2)=sl(c,2)-2*(ss*sin(theta1))*sin(theta2);
end
sl(c,3)=sl(c,3)+ss*cos(theta2);
if sl(c,3)>20
```

```
sl(c,3)=sl(c,3)-2*ss*cos(theta2);
end
if sl(c,3)<-20
sl(c,3)=sl(c,3)-2*ss*cos(theta2);
end
end
plot3(sl(:,1),sl(:,2),sl(:,3),'.r')
axis([-20 20 -20 20 -20 20])
title('Brownian motion dynamic presentation','fontsize',10,'color','k')
xlabel('x'); ylabel('y'); zlabel('z');
grid on
pause(0.001)
rt(a,:)=([sl(1,1),sl(1,2),sl(1,3)])
end
%  Tracking the motion of a single Brownian particle
m=1;
figure                              % Open a new window
while m<=ns
axis([-20 20 -20 20 -20 20])
plot3(rt(m,1),rt(m,2),rt(m,3)'.')
title('A single Brownian motion particle dynamics presentation','fontsize',10,'color','k')
```

```
xlabel('x');ylabel('y');zlabel('z')
grid on
hold on
pause(0.05)
m=m+1
end
figure
axis([-20 20 -20 20 -20 20])
plot3(rt(:,1),rt(:,2),rt(:,3), rt(:,1),rt(:,2),rt(:,3),’.r’)
title('A single Brownian particle motion path','fontsize',10,'color','k')
xlabel('x');ylabel('y');zlabel('z');
grid on
```

And then all simulated results are shown in below.

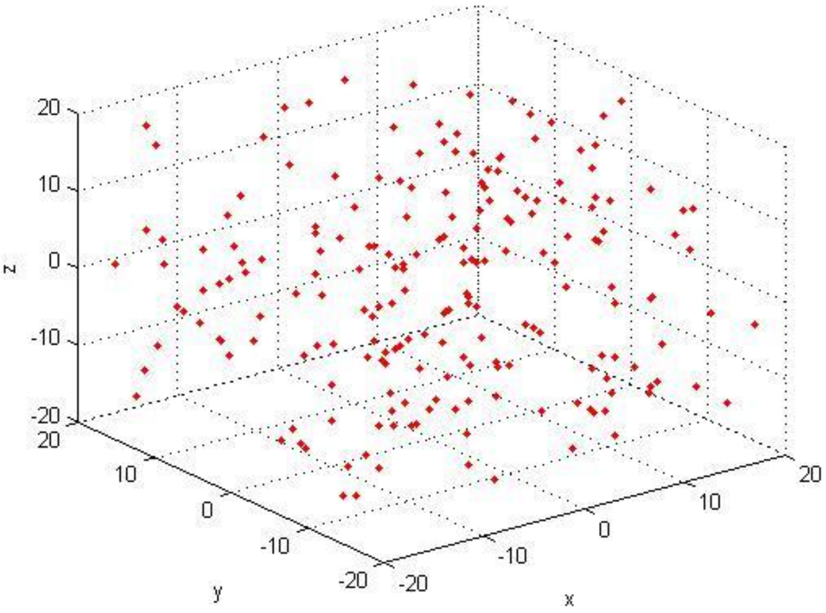


Figure 3.4: It shows the final positions of 200 Brownian particles that are evenly

distributed in the space.

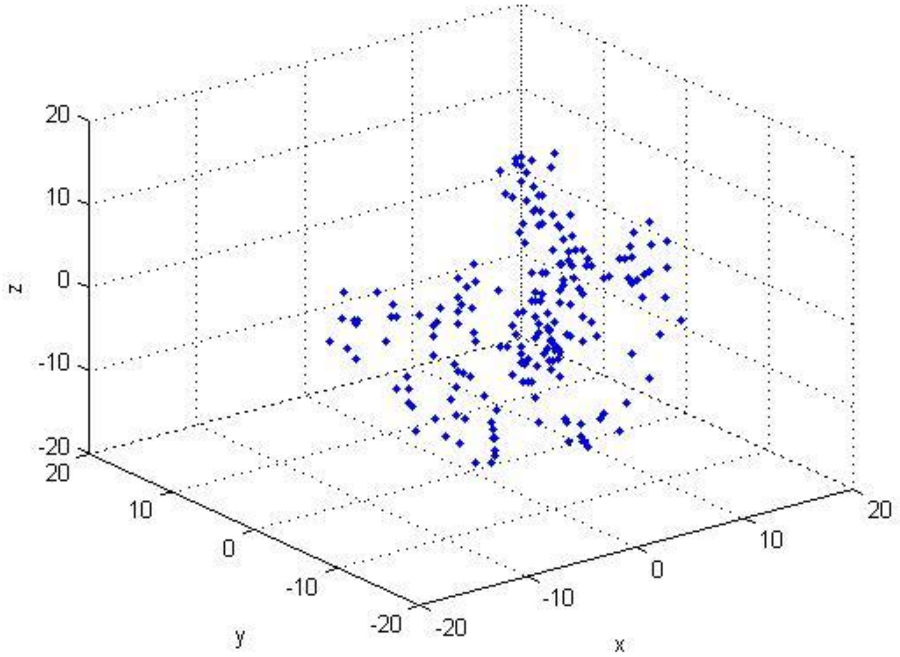


Figure 3.5: Those are tracking positions of a single Brownian particle after diffusion in 200 seconds.

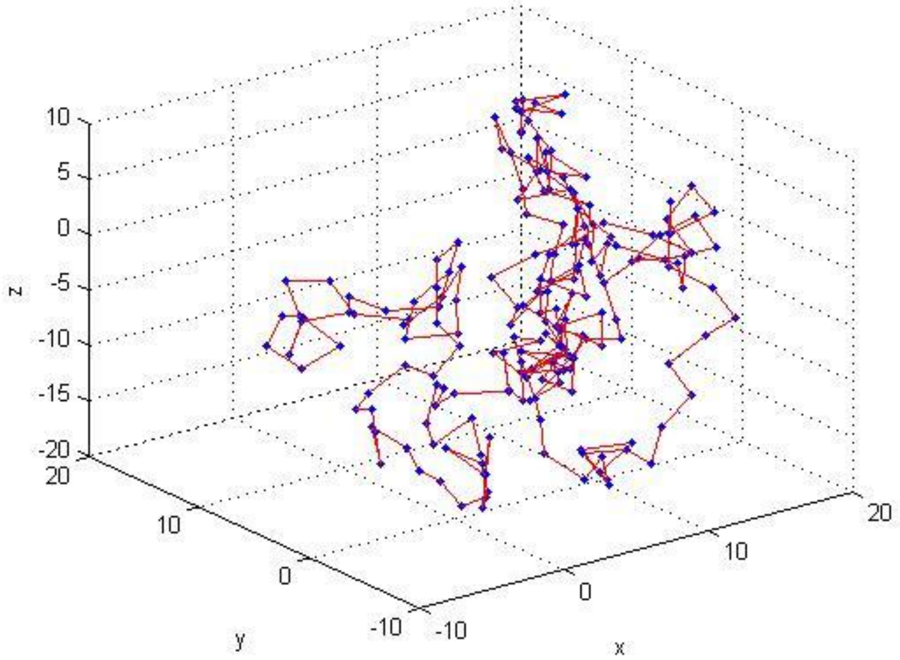


Figure 3.6: It shows a trace of a single Brownian particle motion path.

The Figure 3.6 result shows that the Brownian motion is in line with the randomly and trend distribution.

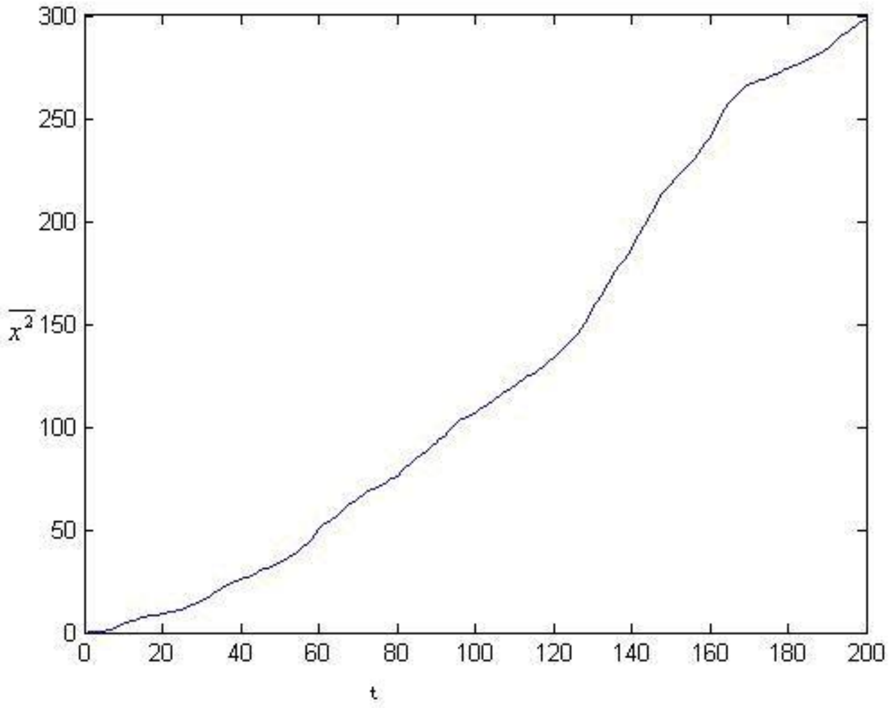


Figure 3.7: The mean square displacement of a Brownian particle is proportional to the time *t*.

The Monte Carlo method simulated the Brownian motion well in Matlab environment. It solves problems that the classical Brownian motion experiments hard to observe and record. Furthermore, compare with manually experiments, the Monte Carlo simulation method saves a lot of time and other costs.

3.6 Summary

This Chapter presented the history and development of the Monte Carlo method. It also discussed the characteristics of the Monte Carlo simulation. Monte Carlo simulation was conducted to simulate the Brownian motion in Matlab environment.

Chapter 4: Parallelizing Monte Carlo with MapReduce Model

In recent years, Google invented a new Distributed Computing Model which is named 'MapReduce' [78]. As Google's core technology, it has obvious advantages in process over TeraByte (TB) level of huge data area.

MapReduce is Google company's secret, but the Hadoop project is a Java's clone version which was created by the open source community [84]. In recent years, Stanford University's Phoenix [78] project has putted MapReduce into shared-memory architecture of hardware platform and has already made some achievements.

4.1 Distributed Computing

4.1.1 Moore's Law and Free Lunch

The Moore's law was defined as the number of circuits on integrated circuit chip which doubles in every 18 to 24 months [77]. From 1965 to the present, this law has been recognized by people. The various properties of CPU have improved exponentially in recent decades. Especially from 2000 to 2003, the frequency of CPU had been developed rapidly and successfully entered into the 3GHz era. However, the development trend has stopped since 2004. The main PC chip manufacturers such like AMD, Intel, Sparc, PowerPc and other companies have stopped their high frequency research projects, meanwhile they developed hyper-threading and multi-core technologies. It indicates that although Moore's Law predicted the pace of development in the past few decades, we can see that this trend would stop in the near future because of the limitation of hardware technology. In 2005, Herb Sutter, who is the chairman of C++ Standard Committee, issued a famous article "*The Free Lunch Is Over----A Fundamental Turn Toward Concurrency in Software*" [78][93]. He argued that if software wants to survive in multi-core era it must consider the concurrency problems. Then, he has made a fearless and amazing prediction: there is no silver bullet any more to solve the multi-core programming problems and no tools can solve the concurrent programming problems in an easy way. To develop high-performance parallel programs, software developers need fundaments to change their minds of programming method. In other words, it is not only changing in the traditional sequential programming method, but also changing sequential thinking habit of people over millions years [79][80].

Now, Herb Sutter's prediction has been confirmed. Single-core computers are out of dated and the prices of multi-core computer have dropped rapidly. People still look for an efficient way to solve multi-core programming problems. The ultimate goal is to improve computer. Google's MapReduce model makes use of a cluster of computers and utilizes functional ideas to improve performance of program implementations [78].

4.1.2 Sequential and Parallel Programming

In the early times, programs were often executed sequentially. Programs are sequences of instructions in the single-core computer era because of these instructions are sequentially executed one by one from the beginning to the end in a program [81].

Research for parallel programming is a way in order to improve the performance and efficiency of program executions [81]. In a parallel program, a process can be divided into several threads and each thread is executed concurrently. Each part of a program can run on the different cores simultaneously. These cores can be in one CPU or many CPUs and they work together.

The Parallel programs can run faster, also they are used to solve a large data set. If there are groups of already networked computers, and then we have groups of computable CPUs. Meanwhile we have strong ability to process a large-scale data set if the file systems have already been distributed.

4.1.3 Parallel Computing Concepts

The first step to establish a parallel program [83] is to distinguish tasks which can be done simultaneously, or parts of the data can be processed simultaneously. However, some programs are impossible to process parallel such as the Fibonacci function [82].

It is shown in Equation (4-1):

$$F_k \begin{cases} 1, & k = 0 \\ 1, & k = 1 \\ F_{k-1} + F_{k-2}, & k \geq 2 \end{cases} \qquad (4\text{-}1)$$

The value of this function depends on the value of the previous function, which means to calculate each step’s result needs to use the previous step’s result as the input data. Therefore, programs like this function are unable to process parallel.

There is a widespread phenomenon of parallel programs, in which the input data has the same structure. If the input data can be divided into the same size data chunks, then we can process the problem in parallel.

Fox example, we can break a large array down into several same size sub-arrays. Assume that each sub-array is executed in the same way without depending calculation on other sub-arrays and there is no communication of each process. Then the computation can be parallelized. It is shown in Figure 4.1.

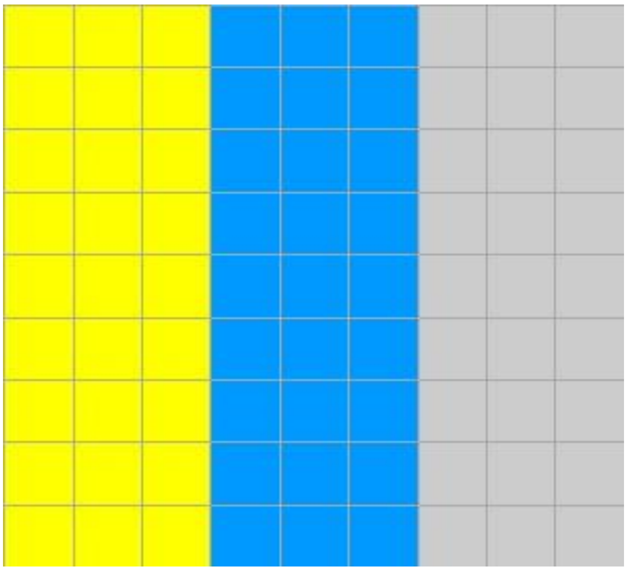


Figure 4.1: The whole array, each sub-array is identified in different color.

To solve a problem like this, usually people use a common Master/Worker model [81][83][92]. The structure of Master/Worker model is shown in below:

Master:

- Initialize the arrays, and then according to the number of workers [83] to disintegrate the arrays.
- Send sub-arrays to workers after disintegration.
- Receive results from each worker.

Worker:

- Receive sub-arrays from Master.
- Process sub-arrays.
- Send results to Master.

All steps above are the basic ideas of the Master/Worker model.

Another example of Master/Worker model is calculating the value of π. First of all, draw a square and a build-in circle as shown in Figure 4.2.

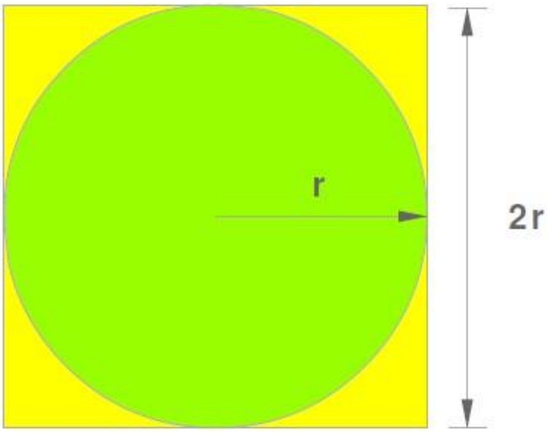


Figure 4.2: Calculating the Value of π.

The area of square is $A_s = (2r)^2 = 4r^2$; (4-2)

The area of circle is $A = \pi r^2$. (4-3)

Then we can get:

$$\pi = \frac{A_c}{r^2}$$

$$A_s = 4r^2$$

$$\Rightarrow r^2 = \frac{A_s}{4},$$

$$\Rightarrow \pi = \frac{A_c}{\frac{A_s}{4}}$$

$$= \frac{4A_c}{A_s} \quad (4\text{-}4)$$

The reason to do this algebraic derivation is that the calculation can be parallelized by using the following steps.

- We generate a number of points in the square in a random way.
- The we count the number of the points which are located in the circle.
- $P=\frac{A}{B}$, A represents the number of the points in the circle , and B represents the number of the points in the square.
- π=4p.

Consequently, we can use the following program to achieve the parallel calculation.

```
NUMPOINTS = 100000; // The bigger the number, the results more approximate
n = the number of worker
numPerWorker = NUMPOINTS / n;
worker:
countCircle = 0; // Each worker has one counter
// every worker does following program
for (i = 0; i < numPerWorker; i++) {
    Generates two random numbers within the square；
    xcoord = The first random number；
    ycoord = The second random number；
    If (xcoord, ycoord) in the circle
        countCircle++;
}
master:
```

15) Receive the values of countCircle from each worker;

16) Calculate π： pi = 4.0 * countCircle / NUMPOINTS;

This is a typical Monte Carlo method to parallel the calculation of the value of π.

4.1.4 MapReduce Programming Model

In the past few years, many Google's employees have done hundreds of calculations with a purpose to process a lot of original data, such as reptile documents, the request logs of Web pages [78][84]and so on. Calculations of all kinds of derive data include inverted index, various representations of the graph structures of web documents, the summary of number of creeping pages on each host, the largest number of daily requires and so on [78]. From the view of concepts, many of these calculations are easily understood. However, for large-scale data, only the calculations are distributed into hundreds of machines and then it will be able to complete in acceptable time. The parallel computation, data distribution and error identification, all those problems together lead an easy calculation process to become very hard to handle. Therefore, users need to write a lot of codes for the parallel programming. In order to respond this complexity, Google's engineer Jeffery Dean designed a new abstract model which allowed users to do easy calculations [78]. Meanwhile, users don't need to care the problems like hiding parallelization, fault-tolerant, data distribution and load balancing by putting them into a library. The inspiration of this abstract model is from

Lisp and original representations of Map and Reduce of other function languages. Actually, many calculations have the same operation: use Map functions in logical record of input data to calculate middle key/value set; and then use Reduce functions calculate those key/value sets in groups and combine all results to a final output file. The abstract model and users appointed Map and Reduce functions together in order to achieve massively parallel computing in an easy way. The abstract model uses the restart function as a primary mechanism can easily to achieve fault-tolerant. The main contribution of this work is via a simple and powerful interface to achieve automatic parallelization and large-scale [83] distributed computing. That means a large number of common PCs also can achieve high-performance computing just like a cluster.

Use the input key/value sets [78][84][92][95] to generate the output key/value [78][84][92][95] sets via computations. Users of MapReduce library need to use two functions express this calculation: the Map function and the Reduce function. Users can self-define the Map function [92] to accept input pairs [95], and then to generate the middle key/value pairs. Then the MapReduce library will integrate all middle values together by the same middle key [78], and then send them to the Reduce functions [84]. Same to the Map functions, users also can self-define the Reduce [92] functions. When the Reduce functions receive a middle key and its relevant value sets, it will combine these values together and build a small value set. Usually, each time of the Reduce functions only generates zero or one output value.

4.1.5 Word Count Case Study

Consider this problem: Count the number of occurrences of each word [84] in a huge document. Users will write similar pseudo codes shown in below:

```
map(String key，String value)：
//key：name of file
//value：content of file
for each word w in value：
EmitIntermediate(w，"1")；
reduce(String key，Iterator values)：
//key：one word
//values：one count list
int result=0；
for each v in values：
result+=ParseInt(v)；
Emit(AsString(resut))；
```

The Map functions can generate any words and the times of the specified word (In this simple example is 1). The Reduce functions count each word specified by the Map function. Furthermore, users use the names of input/output [84] files and optional regulation parameters to fill in a MapReduce specification object [92]. And then users call and send the specification objects to MapReduce functions. And then the Users' codes and the MapReduce library have been linked together [97].

4.1.6 Type Format

Even if the previous pseudo-codes are written in term format of character string input and output, but conceptive users have relevant types when they are writing Map and Reduce functions. Those formats are shown in below [96]:

map(k1，v1) −>list(k2，v2);

reduce(k2，list(v2)) −>list(v2)

4.1.7 Use Cases of MapReduce

Here are some simple programs shown in below which can easily executed by MapReduce.

Distributed Grep [78]. If the input data matched the given format, the Map function will transfer input data to intermediate data, and then the Reduce function only copy the intermediate data to produce output files [95].

Calculate frequencies of URL access [78]: The Map function processes the record of web page requests [92] and produces output data in a format of (URL, 1). Then the Reduce function sums up all values with the same URL and generates a output pair in the format of (URL, total numbers).

Reverse web-link graph [78]. The Map function produces outputs data of each link in the format of (Target, Source) pair [92]. Where each URL is called Target, the Source includes this URL page. And then the Reduce function will produce a list of those URLs and generate (Target, List (Source)) pairs as the output file.

4.2 MapReduce Implementation

4.2.1 Execution Overview

The MapReduce automatically partitions the input data into chunks and distributes those chunks to Mappers who are located at multiple computers. Those chunks can be parallel processed by each Mapper. Then Mappers produce intermediate data for the next step. And then the sort factor put those intermediate keys into groups. After this step, each Reducer will be allocated a group of data to process. At last, all result of each Reducer will be combined to the final output file. The implementation of MapReduce is shown in Figure 4.3.

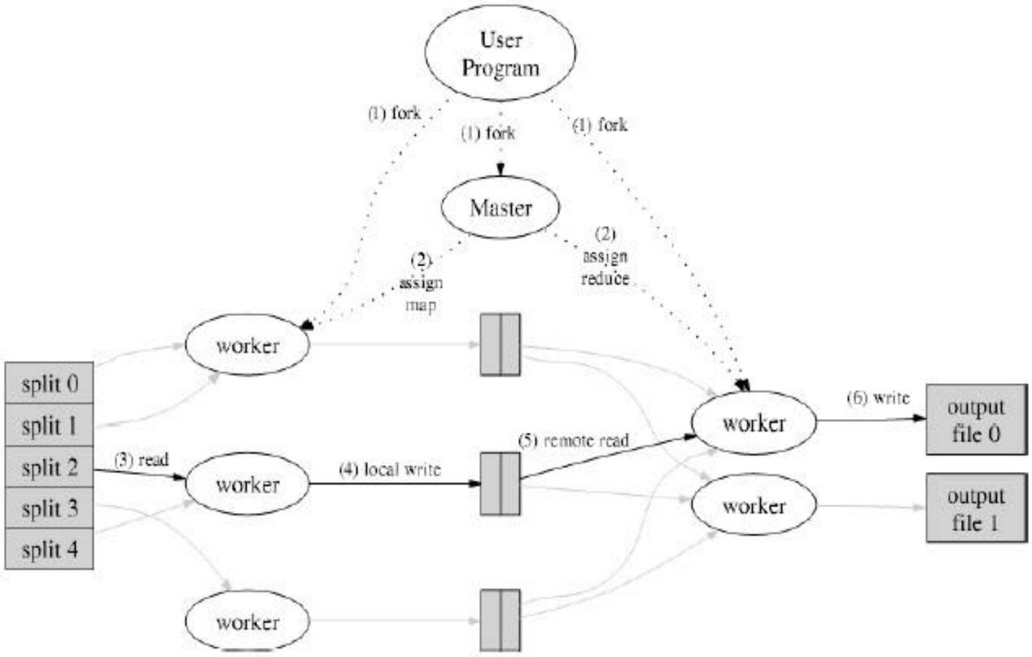


Figure 4.3: The implementation of MapReduce [78].

Figure 4.3 shows the MapReduce operation. When user's program calls [95] the MapReduce function, and then the following actions will occur (the numeral labels in Figure 4.3 are corresponding to the following list).

- First, the MapReduce library [78] divides the input files into M [83] chunks. Each size of chunk typically is 16MB-64MB [78] which can be controlled by using of optional parameter. And then those chunks will be amount copied in the cluster.
- One of the copies is set as a master [78]. The rest of copies are workers which receive and execute tasks from the master. Totally there are M Map function tasks and R Reduce function tasks [83] assigned. The master will drop a Map task or a Reduce task to an idle worker.
- When a worker is assigned task by master. That means the Map tasks and the

Reduce tasks may be computed by same worker. The intermediate data pairs produced by Mappers will be buffered into memory [95] when a Map task is running on the worker.

- Periodically, those intermediate data pairs are written into local hard disks. The partitioning function [78] is used to divide those data into R regions. And then the master receives the region location information of each data pair. The master will forward the information to all workers once it got them.
- When a Reducer worker receives the information from the master it will call intermediate data from the Map worker local hard disks. Before the Reduce workers get all the data. The sort factor grouping all values with the same key together. In this step, sorting is necessary because of there are many values with different keys [95] in the intermediate data.
- Each unique key and relevant pairs in the intermediate data will be sent to the user's self-defined Reduce function. All the output of each Reducer will be combined in a final output file [95].
- After the successful completion of all tasks, then the MapReduce program will return to the user interface.

After successful completion, the output data of the MapReduce are stored in R files (each Reduce task has a user-defined file) [78][84]. Usually, these files will be used as input data for another MapReduce program or another distribution which can handle those partitioned files.

4.2.2 Master Data Structures

The master keeps several data structures [95]. It stores the status of each Map task and Reduce task such as idle, working, or completion. So, it also avoids that the workers are in the idle status.

The master just like a tube where is a way of inputs data pairs from the Map function to the Reduce function. Therefore, the master stores all the locations information and sizes information of the R intermediate files which were produced by Map workers. All information will be sent to the running Reduce workers.

4.2.3Fault Tolerance

The MapReduce is designed to process the large-scale data [78] through using large number of computers or processors, so it must tolerate computers failures very well.

4.2.3.1 Worker Fault

The master pings each worker periodically [95]. If master dose not receives response from a worker in a certain period of time, and then the worker will be identified as a failed worker [84][94]. All map tasks were completed by a failed worker need to reset to their initial idle status, meanwhile, those tasks will be sent to other working workers. The completed Map tasks on a failure worker will be executed again because the output data are stored on the worker’s local hard disks which have had unable to visit. The completed Reduce tasks by the failure worker do not need to re-executed because Reducer's output data are stored in the global file system [96]. For example, another worker will execute a map task if the previous worker failed, this situation will be informed to all workers who are executing Reduce tasks. The Reduce worker who has not received data from the previous worker will get data from the new worker [95]. From this point of view, MapReduce can easily handle worker failures because it just simply re-executes tasks until the operation completed.

4.2.3.2 Master Fault

It is easily to let master cyclically to write checkpoints of the data structures described above. This means if the master process failed, another new master process will start since the latest checkpoint. However, due to there is only one master in the system, its failure is very troublesome. Therefore, we can achieve right now that if the master failed, the MapReduce implementation will be terminated [92][100] by the system. And then users will be noticed and can check this status and restart the operation or sort out problems.

4.2.3.3 The Error Handing Mechanism

A distribution system will be established during the MapReduce operating [78][84][96][97][99]. Each task has its own private temporary file in MapReduce. A

Reducer only has one such file, and a Mapper has R such files, each file is corresponding to one of Reducers. The information of the temporary R files will be sent to master after the completion of a Map task. The master will ignore the information message if it has already got one. Otherwise, master will record the information message into its data structures. After a Reduce task completed, each Reducer renames its own output file and combine all Reducers' files into the final output file. The underlying file system [95] is making sure that each Reduce task has only one file will be combined into the final output file. It is necessary if there are many Reducers assigned the same Reduce task by master.

4.2.4 The Storage Location

Network bandwidth is a limited resource in distribution system environment [95][98]. We save network bandwidth via store the input data controlled by Google File System (GFS)[84] into workers local hard disks. The GFS divides input data into 64MB chunks and stores several replicas of each chunk [95][96] in different workers. The master considers the location information of the input files [95] and tries to create a Map task on the worker which has relevant input data in its hard disks. If master's efforts failed, then it tries to create a Map task on a near worker. For example, the new selected worker is in one switch framework with the worker which contains input data. To save the network bandwidth and improve the performance of large-scale computations when use MapReduce in a cluster environment, almost input data are read from local hard disks.

4.2.5 Task Granularity

As described above, the Map tasks are divided into M numbers and the Reduce tasks are divided into R numbers [99]. The total number of M and R are much more than the total number of workers. Each worker performs several different tasks to improve the dynamic load balancing [93] and allows MapReduce to do fast recovery from worker failure. For example, all completed Map tasks by a failed worker will be assigned by master to other working workers. There are limitations of sizes of the scope of M and R because the master must dispatch O(M+R) times and save O(M*R) statuses into memory [98][101]. The statuses only use small amount of memories, each Map/Reduce pair approximately only uses one byte. In addition, usually R is limited by users [95], that is because of each Reduce task creates its own output file, more Reduce task means more memory will be used. Generally, R is set as the multiple of workers the users expected to use. For example, set the MapReduce implemented with M=80,000, R=2,000 in 800 workers.

4.2.6 Alternate Tasks

A laggard is one of the reasons to extend the operating time of MapReduce. In other words, a worker successfully completes one of Map or Reduce tasks in an unusual long time [85][91]. There are many possible reasons for producing a laggard. For example, a worker with hard disk errors occurrence frequently but those faults can be corrected may lead to the read performance from 80MB/s down to 10MB/s. Meanwhile, master has already distributed other tasks to this worker. There are many reasons can cause the slowly implementation of MapReduce such as the performance limitation of CPU, memory, hard disk, and network bandwidth [83][86].

There is a mechanism used for assuaging the laggard problems. When implementation of MapReduce almost finished, the master calls alternates workers to execute the remaining tasks which still in progress. Whether the tasks completed by original worker or alternates workers, the tasks will also is marked as completion. This mechanism will normally take more than a few percentages of the workers resources, but it significantly helps to save time cost of the MapReduce implementations.

4.3 MapReduce for Multi-core Systems with Phoenix

Phoenix is a kind of implementations of MapReduce based on the shared-memory systems [78]. Its goal is making the execution more efficiently and users do not care the multiple core platform. In fact, in the concurrency management even an experienced programmer is also may make mistakes. Phoenix is consisted by an efficiently runtime component and a few simple APIs which are open to application developers. In Phoenix implementation, the runtime component is used to handle parallelization, resource management, and bug fix.

4.3.1 Basic Operation and Control Flow

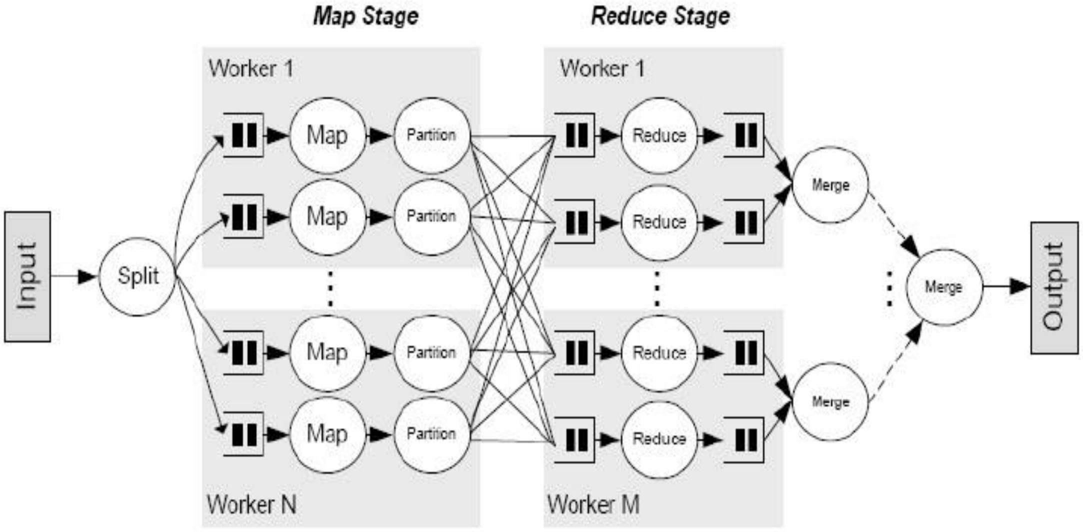


Figure 4.4: The basic flow of Phnoenix runtime [89].

As shown in Figure 4.4, the scheduler controls the runtime. The responsibilities of scheduler are creating and managing all threads of in progress Map and Reduce tasks. It is also managing the buffers which are used for communications between tasks. Programmers through using scheduler_args structure to initialize all the scheduler needing such as data and functional pointers. And then the scheduler detects the total number of processor cores can be used for the computation. The scheduler will establish a worker thread on each processor core and the Map and Reduce tasks are dynamically allocated to the thread for executions.

In the start step of Map phase, the Splitter function divides input data into number of equal-sized chunks [89][92]. The Splitter is called once [97] by each Map task and then returns data to Map tasks to process. Map tasks are dynamically distributed to workers [97] and each task produces the intermediate <key, value> pairs which will be used as input data of the Reduce tasks. In each buffer, values are sorted by the assistance of key. At this time, the Map phase is end. After all Map tasks completed, the scheduler starts the Reduce phase.

Reduce tasks are similar to Map tasks, also distributed dynamically to workers. The only difference is that each Map task is completely independent when the Map tasks are processing pairs. The Reduce tasks are quite differently, they must process all values with the same key [89] in one Reducer. Therefore, in Reduce phase, workers may appear unbalanced loading, so the dynamic allocation is crucial. Before the Reducer starts to work, all final output file produced by each Mapper will be merged together into a single buffer [95] and sorted by keys. This step is processed by the sort factor. The merging process has $\log_2 \frac{p}{2}$ steps, where P is the total number of using workers.

4.3.2 The Buffer Management

There are two different type of buffers between the Map phase and the Reduce phase

[94]. All buffers are assigned in shared memory and interviewed by specified function. When users are have to re-arrange buffers [95] in some conditions. For example, split across tasks. In this condition, we control pointers rather than the actual pairs because the data of pairs are very large. Furthermore, the intermediate buffer is not visible to the application program.

All the intermediate data pairs are stored in buffers. Each worker has its own private buffer. This buffer is defaulted the size during the system initialization and then the size will change dynamically based on demands in process. In sorting step, a key may corresponds to multiple key/value pairs. In order to accelerate the speed of Partition function, the Emit_intermediate function [95][96] is used to store all the same key pairs in the same buffer. At the time of the Map tasks completed, all the buffers are sorted in order by key. In this phase, each key is associated with only one value. After this step, the final output file is produced and it can be accessed by user assigned output data buffer.

4.3.3 Fault Tolerance

The runtime provides a limit repair service for solving the accidents or the persistent errors during the Map phase and the Reduce phase.

Phoenix uses timeout policy method to detect the system failures [96]. If a worker completes a task over the standard time cost which is made by considered cost time of other workers have done similar tasks. And then master can suspect that there are errors occurred within this worker. Of course, errors often lead to incorrect or incomplete data rather than totally system failed. The Phoenix system itself is unable to detect this situation and even cannot prevent shared memory polluted by erroneous tasks [99].

The Phoenix runtime and existing error detect technologies are used together to complement this defect. Phoenix provides relevant information to simplify the process of error detection. For example, Phoenix can notify the status of load/store address such as safe or unsafe to the hardware when it has known the address ranges of input data and output data. If the address may be unsafe, then Phoenix will send a unsafe signal to hardware.

When an error is occurred or pended, the runtime system tries to execute the failed task again [87][91]. Since the failed task may be still in progress, the new task will be assigned a new output data buffer. Otherwise, the new task address space is in conflict with the original task and this situation will cause the pollution of data.

When one of those two tasks finish, the runtime will adopt the result calculated by the completed task and put the result into the part of the output data. Then the scheduler assumes that the error occurrence just is an accident and continues to assign new tasks to the worker had failed before. If errors occur again on that worker in a short time, the scheduler determines that is a persistent error condition rather than an accident. Meanwhile, the scheduler will be no longer to assign any task to this worker.

Furthermore, the Phoenix scheduler itself does not provide bug fix services.

4.3.4 Concurrency and Locality Management

The runtime scheduling policy directly affects the efficiency of performance of all parallel tasks. Phoenix uses three methods during its decision making process:

- Use the default policy if a system which had been considered in its development;
- Dynamically determine the optimal scheduling policies via monitoring available resources and the runtime behaviours;
- Allow the programmer to custom policy [93][95][96].

4.4 Performance Evaluation

This section presents the performance evaluation results of the parallel Monte Carlo in an experimental environment.

4.4.1 Experimental Results

A small MapReduce Hadoop cluster was set up using 4 computers with the following configurations as shown in Table 4.1.

Table 4.1: Hadoop configurations.

Hardware environment			
	CPU	Number of Cores	RAM
Node 1	Intel Quad Core	4	4GB
Node 2	Intel Quad Core	4	4GB
Node 3	Intel Quad Core	4	4GB
Node 4	Intel Quad Core	4	4GB
Software environment			
OS	Fedora10		
Hadoop	Hadoop 0.20		
Java	JDK 1.6		

Figure 4.5 shows the speedup of the MapReduce enabled parallel Monte Carlo using a varied number of Mappers in comparison with the sequential one on different sizes of data. As can be observed from this figure, more Mappers are used, better speedup is achieved especially on a larger data set.

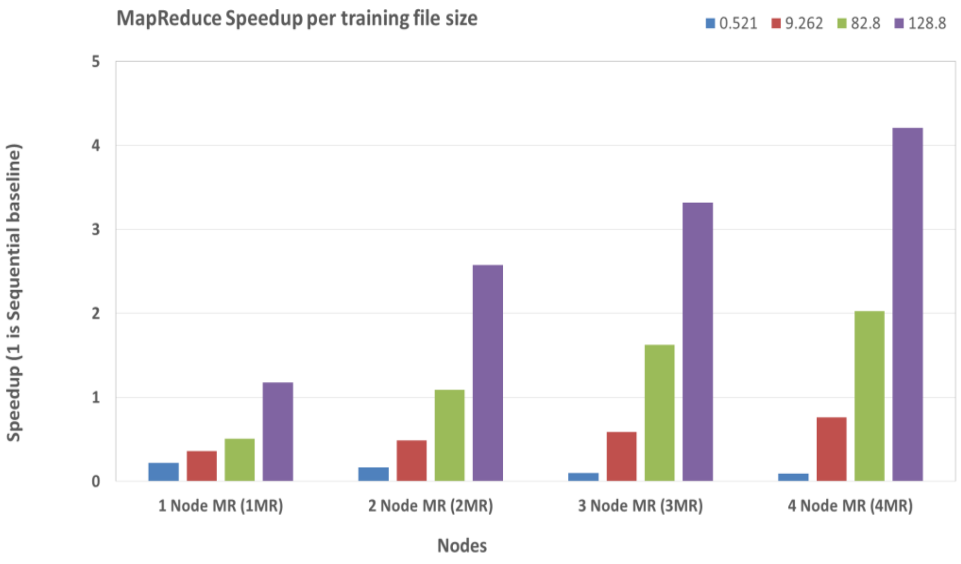


Figure 4.5: The speedup of different number of Mappers.

Figure 4.6 shows the accuracy of the parallel MapReduce Monte Carlo which shows the accuracy of the parallel Monte Carlo is highly close to that of the sequential one.

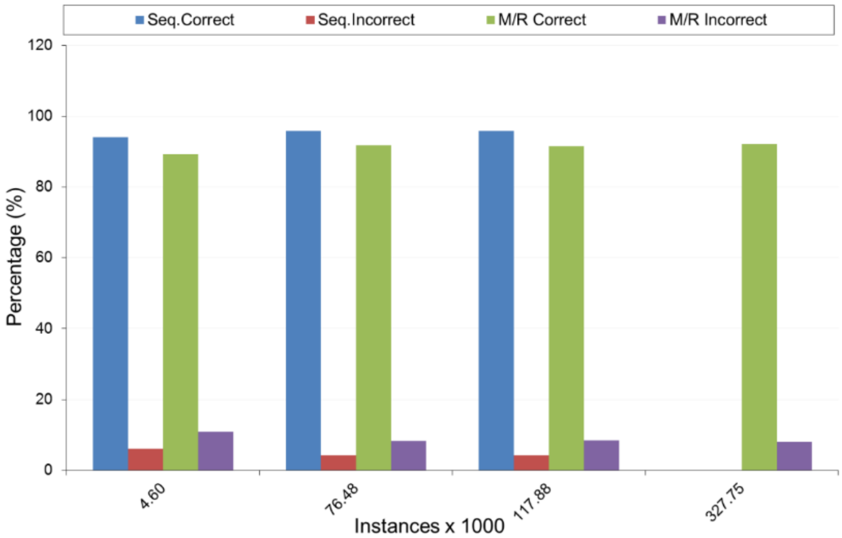


Figure 4.6: The accuracy of MapReduce Monte Carlo.

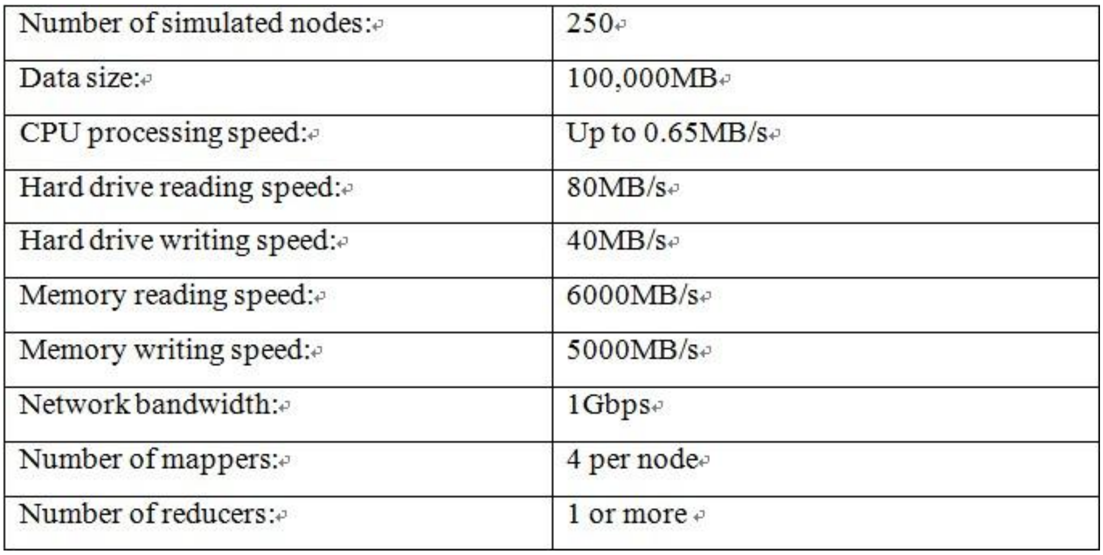
Table 4.2: summarises the accuracy the parallel Monte Carlo.

	Sequential Monte Carlo	MapReduce Monte Carlo
Correctly computed	≈ 94.04 %	≈ 96.23 %
Incorrectly computed	≈ 6.96 %	≈ 4.77 %
128,000 data samples (processing time in seconds)	≈ 532 s	≈ 115 s

4.4.2 Simulation Results

To further evaluate the performance of the parallel Monte Carlo in a large scale Hadoop cluster environment, we employed HSim [103], a Hadoop simulator developed by the research group at Brunel. Table 4.3 shows the configurations of the simulated Hadoop cluster.

Table 4.3: Hadoop simulation configurations.



Number of simulated nodes:	250
Data size:	100,000MB
CPU processing speed:	Up to 0.65MB/s
Hard drive reading speed:	80MB/s
Hard drive writing speed:	40MB/s
Memory reading speed:	6000MB/s
Memory writing speed:	5000MB/s
Network bandwidth:	1Gbps
Number of mappers:	4 per node
Number of reducers:	1 or more

Figure 4.7 shows the impact of the number of Reducers on the efficiency of the MapReduce Monte Carlo. It is observed that the number of Reducers does not make much difference on the efficiency of the parallel Monte Carlo. However, the processing time decreases with an increasing number of Mappers, as the computation of Monte Carlo is mainly carried by the Mapper nodes.

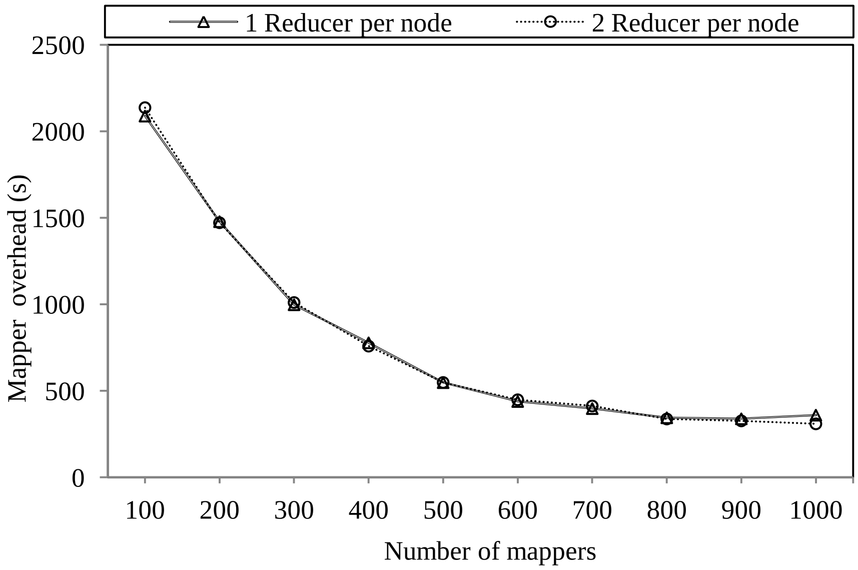


Figure 4.7: The impact of Mappers.

Figure 4.8 shows the effect of sort factor on the performance of the parallel Monte Carlo. It can be observed that the higher the sort factor is, the better performance is achieved which means that a lower overhead in computation is incurred. This is because the sort factor affects the performance of Hadoop in IO operations.

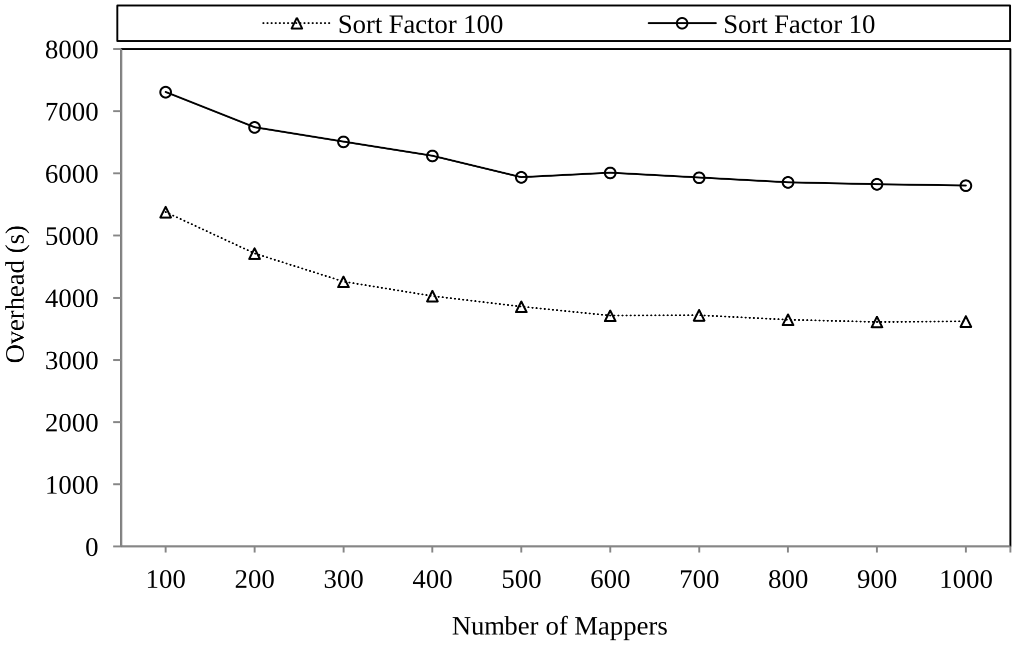


Figure 4.8: The impact of sort factor.

4.5 Summary

This Chapter presented the implemented parallel Monte Carlo algorithm with MapReduce model. It discussed distributed computing and programming and implementation of the MapReduce model. It also presented the multi-core systems with Phoenix MapReduce model and it also evaluated the performance of the parallel Monte Carlo in both an experimental environment and in a large scale Hadoop cluster environment. Both the experimental results and the simulation results showed that parallel Monte Carlo is faster than the sequential Monte Carlo. For large-scale input data, more Mappers used, a better performance can be achieved.

Chapter 5: Data Locality in MapReduce Model

5.1 Introduction

MapReduce is a parallel programming model developed by Google for processing large amounts of data in distribution system environment. Due to its features such as fault-tolerance, simplicity and scalability, MapReduce can be used as a framework to process large scale data [86] applications such as scientific simulation and web data mining. In addition, the MapReduce model has a very simple architecture because it processes parallelization details by only using the runtime, Map and Reduce functions. There are some implemented systems of the MapReduce such as Hadoop, Dryad, Phoenix, Mars and Sphere. Yahoo [84] has developed its own open source MapRedcue project which is called Hadoop. It runs jobs that produce massive data (over 300 TB compressed) on at least ten of thousand cores.

The Hadoop's basic principle is moving computation to the data or as close as possible to the data which are needed to compute [84]. From this point of view, usually there are large amount of nodes in a framework. Those nodes can be personal computers or clusters. A lot of data are needed to compute in local disks. Move those data to application in progress worker costing bandwidths and time. On the other hand, just move the application to or near the disks where the data located will improve the overall performance and save time cost. In the other words, the computation tasks are closer to data called data locality. Today’s cluster has thousands of shared nodes which transmit massive data that impose network load and create congestion. So, an efficiently scheduler can avoid unnecessarily data transmissions. The Scheduler considers data locality because it determines the performance of MapRedue due to the limitation of network bandwidth in distribution systems [77].

According to the distance between workers and input data [84], there are three levels of Map tasks in each worker. The most efficient locality is the first level locality where the in progress Map task is launched on the worker which holds the task input data. It is also called the node level locality. When a task cannot achieve the first level locality, then the scheduler will execute task on a node in the same rack where the computation node and data node are located together. So it is called rack level locality, also known as the second level locality. If the task still fails to achieve the second level locality and then the scheduler will launch the task on a node located at a different rack. This is called rackoff level locality, also known as the third level locality.

Otherwise, if locality is not achieved, the data transferring IO cost will seriously affect the performance of the whole system due to the limitation of shared network bandwidth. The default Hadoop scheduler provides some mechanisms to improve the data locality but they have some inherent deficiencies. To avoid unnecessary data transformations in Hadoop environment, researchers developed data-locality aware schedulers. In this part, we study these schedulers and describe how they improve the data locality in Hadoop environment. Furthermore, we analyze and evaluate these schedulers and present their strengths and weaknesses.

5.2. Hadoop MapReduce Overview

As described above, Hadoop is an open-source MapReduce model [84]. The Hadoop environment consists of two main components: the MapReduce framework and HDFS (Hadoop Distributed File System) [84].

The MapReduce framework is shown in Figure 5.1, it divided computations into Map and Reduce functions. In Map function, the job is divided into small tasks and assigned to specific slave nodes. In this stage, each Map task processes one chunk of input data in the form of <key, value> pairs [77] and produces the intermediate result data < key, value> pairs. The intermediate data < key, value> pairs are stored in worker's local disk. In Reduce function, the intermediate data are read in the form of <key, value> pairs and combined all values corresponding to same key to produce the final output. Both the Map and Reduce functions can independently execute each <key, value> pair, exposing significant parallelism [78][84].

Two software services, JobTracker and TaskTracker are implemented in Hadoop MapReduce framework. The JobTracker is run on master and TaskTracker is run on each worker. The JobTracker is responsible for splitting the input data into small chunks and scheduling each task on workers. It is monitoring the working conditions by receiving the signals from workers and recovering from failures by re-assigning tasks. The TaskTracker is running on each worker and accept to the Map and Reduce tasks from the JobTracker. Each TaskTracker contacts with JobTracker periodically via the heartbeat messages to report the task completion progress and request for new task when the worker is idle. When master receives signal from the worker, the master calls the scheduler (JobTracker) to assign a new task to the worker. The rest of steps have no difference to other MapReduce models such as Phoenix.

In the cluster environment, the Hadoop Distributed File System (HDFS) is designed to provide the global access for files [83]. HDFS is closely resembled to Google File System [81] and highly reliable because each input data chunk has multiple replicas (3 by default) are stored in multiple data nodes for the purpose of fault-tolerance [82]. HDFS has master/slave structure which includes a NameNode and many DataNodes. In Hadoop environment the input file is stored in HDFS and passed to Map function. HDFS splits the input file into even-size fragments (64 MB by default) which are distributed across to group of DataNodes for further processing. The NameNode manages file system’s namespace and standardize the client to access the files. The DataNode(s) manages the storage devices which are directly attached to each data node.

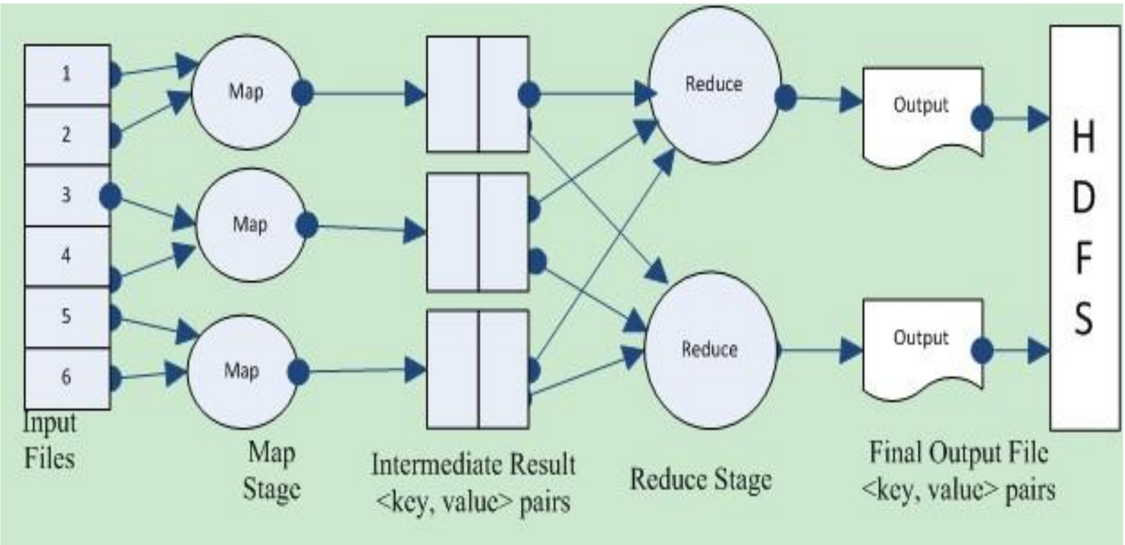


Figure 5.1: Hadoop MapReduce Framework.

5.3. Data Locality in Hadoop

As data locality is a determining factor in Hadoop environment, it significantly affects the performance of the system. There are several factors that affect the date locality such as the size of the cluster, the number of data replica and job execution stage. In a large size cluster and small amounts of jobs condition, the probability of the data locality is low. For example, if a job has 10 Map tasks which are submitted to a cluster which has 150 nodes. It is impossible to obtain a good locality rate. Since each task has 3 replicas of input data which are distributed on 3 different nodes, there are only one fifth workers that can get the input data. This means the probability of the data locality for the job is only 20%. If we decrease the number of nodes to 50, then the data locality of the job will increase to 60%.

Similarly, the number of replicas and job execution stage also influences on the data locality. Increasing the number of replicas of input data improves the data locality, but it consumes a lot of storages. At the job initialization stage, there are large number of unmapped tasks and input data stored on nodes, so the probability of the data locality is high. While at the end stage, the job has small number of unmapped tasks and input data are available on less number of nodes, so the data locality becomes much lower.

The Hadoop default scheduler schedules jobs by FIFO and has already considered on the data locality [85]. When a master receives a signal from workers which indicate that a free Map slot is available, then the JobTracker on master node will try to find the Map task workers which have the job input data. If the worker is found then the node level locality is achieved and task will be launched on that worker. When the node level locality is impossible to achieve, then JobTracker will try to seek the rack level locality. If the rack level locality still fails then task is arbitrary picked up and launched on an off-the-rack node. This simple scheduling algorithm is favoring the data locality but with deficiencies. For example, this algorithm strictly follows the FIFO policy where tasks are scheduled one by one and each task is scheduled without considering its impact on other tasks.

Let us consider a Hadoop cluster of three nodes (N1, N2 and N3) shown in Figure 5.2.

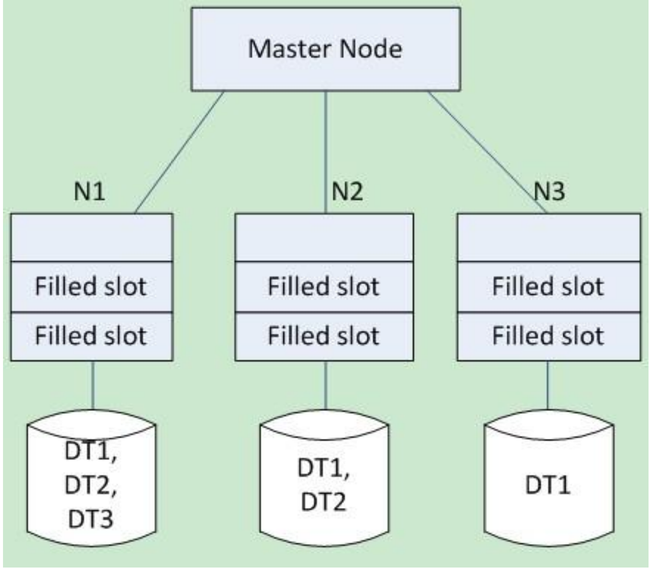


Figure 5.2: Hadoop cluster.

Each node has at least one free Map slot. Let us assume that there are three tasks (t1, t2 and t3). Each task input data has multiple replicas which are stored over multiple nodes for reliability purpose. Task t1 input data are stored on nodes N1, N2 and N3, task t2 input data are stored on nodes N1 and N2 and task t3 input data are stored on node N1shown in the Figure 5.3.

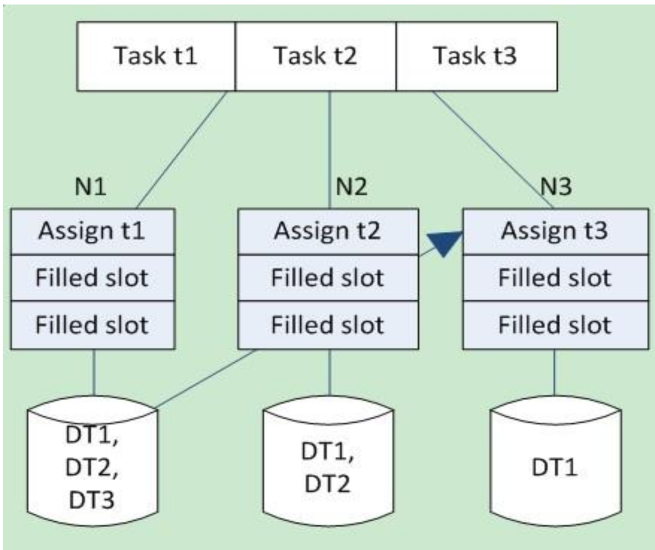


Figure 5.3: Hadoop cluster task t3 loses Locality.

Now how the default scheduler assign these tasks to nodes. The scheduler assigns task t1 to node N1 and achieves the node level locality, and then task t2 is assigned to node N2 and it also achieves the node level locality. Now, there is only one node C has the idle slot and only one unscheduled task t3. So the task t3 must be assigned to node C as shown in Figure 5.3. To summarized, task t1 and task t2 have gained the locality while task t3 has lost the locality. That is because of the scheduler processes the tasks one by one rather than considers all tasks on all available idle slots. All tasks will achieve the locality if scheduler processes all the tasks on all available idle slots at once as shown in the Figure 5.4.

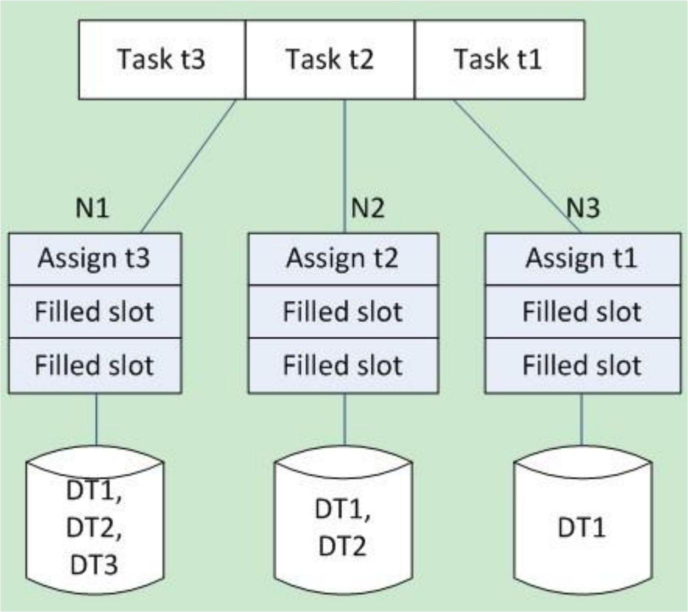


Figure 5.4: Task t3 gain locality.

5.4. Data Locality Aware Scheduling

In this section we study the state-of-art data locality aware scheduling algorithms in

Hadoop MapReduce environment.

5.4.1. DARE Scheduling

Abad et al. [79], designed DARE (Adaptive Data Replication) algorithm to improve data locality by dynamically replicating the popular data and reducing network congestion through redistributing the correlated access chunks on different nodes. The implementation of this algorithm is relying on two approaches: the greedy approach and the probabilistic approach.

In the current implementation, when the Map task is impossible to gain the local data, MapReduce framework will fetch data from remote node where the data located. Then those data are processed and will be discarded after the Map task completed. The greedy approach takes the advantages of remotely fetched data, it makes and inserts the subset of into HDFS at the node [89]. By doing this, DARE dynamically creates the replicas of data without any extra network cost. Replicate remotely fetched data chunks immediately leading to huge disk storage cost. To address this issue, DARE uses the eviction mechanism which is controlled by LRU (Least Recently Used) policy to delete recently least used data chunks to make sure that there are spaces for new replicas.

The probabilistic approach is unlike the greedy approach. It does not replicate remotely fetched data immediately but only replicate popular data. In this approach, each individual node runs the algorithm independently to generate replicas of the most popular data. This approach is also applied the aging eviction mechanism to rapidly expel the files with decreasing of popularity. When the storage budget is reached, the algorithm iterates dynamically replicated data chunks and evicts the files with decreasing of popularity to avoid files are being expelled.

5.4.2. Delay Scheduling

Zaharia et al [91] developed an algorithm which is called delay scheduling to enhance the data locality rate in Hadoop environment. The delay scheduler is applied into FairScheduler in Hadoop. FairScheduler has changed allocating equal share (time slot) of each job to allocating equal share of each user. Each user is assigned to its own spaces in a shared cluster. If a user cannot use its time slots, instead, other users can use them. If a user cannot get the minimum share meanwhile the preemption is being occurred. This means the scheduler reallocates the resources among the users. There are two approaches of preemption: first, killing the running jobs; second, waiting for running jobs to complete [100]. Killing the running jobs immediately scarifies the time had been run while waiting for running jobs to complete does not have such issue. But it scarifies the fairness.

The delay scheduling algorithm uses the waiting approach to achieve the data locality.

It is defined as if a node cannot process a local Map task then it will be waiting for a short time, and then it processes other tasks instead [92]. The delay scheduling relaxes the strict order of task assignment and delays job execution if there is no local Map task on a worker. Therefore, users specify the maximum delay time D is necessary. For example, if a worker has been skipped for longer than D time then it will be allowed to launch rack-level tasks. Thus, if it has been skipped for further longer time then it will be allowed to launch off-rack level tasks. These skip times are called delay time which is a very important factor in this algorithm. How to specify the values of delay time to achieve the target level locality is important. Usually, the values of the delay time are specified by using of either the 1.5 times to slave node heartbeat or the rate of the slots free up which is less than the average task length. The delay scheduling also resolves other locality problems such as Head-of-Line and Sticky-Slots [91].

5.4.3. Matchmaking Scheduling Algorithm

He et al [92] developed Matchmaking scheduling algorithm to enhance the data locality in MapReduce cluster. The main concept of this algorithm is making sure that each node has a fair chance to be assigned a local task before a non-local task. This process is similar to the Delay Scheduling. It can be defined as if a node has failed to be assigned a local task in the queue, and then the algorithm will continue to search the succeeding tasks. For the purpose of giving a fair chance to each node to get a local Map task, a no non-local task will be assigned as the second choice when a node cannot find a local Map task at the first time. In other words, if a node still failed to get a local map task twice time, matchmaking scheduler will assign a non-local task to the node to avoid wasting of computation resources. This algorithm assigns a locality marker value to each node to represent the status of node. When a new job is added, all previously locality marker values will be cleared because of the new job may include local Map tasks for many slave nodes.

5.4.4. Pre-fetching and Pre-shuffling Algorithm

Sangwon et al. [89] developed two innovative techniques: Prefetching and Pre-shuffling. Those can improve the overall performance of the MapReduce cluster. The prefetching technique enhances the data locality while the pre-shuffling reduces the time cost of shuffling of intermediate data produced by Map functions. In order to reduce the network overhead, the pre-shuffling technique is trying to predict the target Reducer where the intermediate result data are partitioned before the Mapper implementation. As we focus on data locality aware scheduling, so we only discuss the prefetching technique.

The prefetching is a bi-directional technique which performs the complex computation in one side and in the other side pre-fetches and assigns data of the corresponding tasks. This technique pre-fetches the required data chunks of Map tasks which are near to the computation node or in local rack. This technique includes two modules: the predictor and the pre-fetching. The predictor generates list of data chunks C which is required for the target task. First, the pre-fetcher identifies the location of replica c in the list of data chunks C. If the replica c does not exist in the local rack, but the access frequency of c is larger than the threshold, then the pre-fetcher will try to replicate c in local node. If there is no enough space in the local node, then a replica of c will be created in the local rack. The pre-fetcher module is also monitoring the synchronization status between the computation and pre-fetching, and both activities are performing simultaneously.

5.4.5. NKS Algorithm

Zhang at el. [94] designed next-k-node scheduling (NKS) algorithm to improve the data locality of Map task in homogeneous environment. The algorithm preferentially schedules the node level locality tasks. If there are no such Map tasks available, then NKS algorithm will calculate the probability of each task and schedule the highest probability task.

In this algorithm, the main factor is predicting the next node to issue request for task, this node is called k node. The method to determine the next k node is based on the progress reports of the running tasks. In Hadoop environment, all TaskTrackers on nodes are periodically reporting the progress statuses to the JobTracker. The size of the whole input data and the size of processed data can be used to calculate the progress of the running tasks. In homogeneous environment, all the nodes are identical in term of processing and disk capacities, so the process speeds of tasks are the same. Therefore, the task with the highest progress will be completed first and then the worker will request for next task. This means the request of this node is earlier than other workers. Then we know that the NKS algorithm is based on the progress of the running task to predict the next k node. However, in the conditions of different sizes of input data of the Map tasks, the NKS algorithm cannot predict the next k node correctly by using task progress because of those tasks will be completed at different time. In this case the NKS algorithm takes the input data sizes to equals to the Map tasks as a solution. To correct the prediction of next k node, the progress of the imaginary task is map with original task progress.

To sum up, the advantages and limitations of all algorithms described above are shown in Table 5.1.

Table 5.1: Summarized data locality aware scheduling.

Scheduler	Strengths	Weaknesses
DARE Scheduler[79]	• Creating replica without any extra network cost • Simple technique	• Multiple replicas consumes unnecessary huge amount of storage • Distribution of multiple replicas difficult in large-scale environment
DELY Scheduling	• Best perform in environment where	• Perform worse in environment where jobs are longer and

[91]	jobs are shorter and nodes are free up quickly. • Simplicity of scheduling algorithm	nodes are free up slowly • Need careful turning.
NKS [94]	• Applied in Homogeneous Environment	• Not applied in heterogeneous environment. • Need careful tuning
Pre-fetching and pre-shuffling [89]	• No particular	• Occupy much network bandwidth • Complex algorithm
Matchmaking [92]	• High cluster utilization • Every node has fair chance to grab a local task • Performed better than delay scheduling • No tuning required	• No particular

5.5. Discussion

There are several points to discuss regarding the data locality scheduling aware algorithms. Both of these algorithms are trying to resolve the problems of data locality, but cause some other serious issues. For example, the DARE scheduler is a simple technique to improve the data locality by dynamically replicating the remote or popular data, but this technique also has several limitations.

- In a large-scale cluster, create replicas is very costly.
- It increases the cost because storing large numbers of replicas require large-scale disk capacity.
- Distributing large numbers of replicas are creating network overhead in Hadoop cluster.

To address these limitations, a scheme was developed to implement a low-overhead file replication [99]. Another remedy was proposed to implement the data placement strategy [82]. Similarly, the Delay scheduling algorithm is working well if the most jobs are short and nodes are freeing up quickly. This algorithm does not suitable for the condition that the most jobs are long and nodes are freeing up slowly. Another issue of the Delay scheduling algorithm is the configuration of the values of delay time D. If the value of D is configured too small then it will cause assigning non-local tasks too fast. On contrary, if the value of D is configured too large then the job starvation may occur and it will affect the performance of the system. For the purpose of get the best data locality, this algorithm always need users to careful tuning by configuring the value of D.

NKS and pre-fetching methods are developed for the purpose of the improvement of data locality. However, these techniques have several deficiencies. For Example, the pre-fetching algorithm is very complex and utilizing huge network bandwidth. The NKS method is only applied in homogeneous environment and it cannot work in heterogeneous environment. In heterogeneous environment, nodes have different levels of computation power in term of processing and storage capacities. Therefore, it is very difficult to correctly according to running task progress to predict the next k node. Another reason is the task speculation issue making it difficult to predict next k node. For the purpose of overcome the task speculation issue, researchers developed LATE (Longest Approximate Time to End) scheduler [100] and SAMR (Self Adaptive MapReduce Scheduling) algorithm [101]. In NKS algorithm, k is very critical factor which represents the difference between actual order and predicted order of the task requests. So users must configure the value of k very carefully. If the value of k is set too small then the next node request of task will cannot be captured. If the value of k is set too high then there are large numbers of nodes will be captured and this is unnecessary to improve data locality. Therefore, the carefully tuning is required for the purpose of better performance.

5.6 Testing Results on Data Locality

The default test platform configuration is shown as follows:

- Totally 10 computers are divided into 2 frameworks, each framework has 5 computers;
- The inter-framework router bandwidth is 1000Mbit/s;
- The export bandwidth of switches between frameworks is 1000Mbit/s;
- Each node has four Mappers, the entire cluster has 40 Mappers and 1 Reducer.

- The sizes of file chunks are set to 64MB;
- Each Mapper’s private memory is set to 100MB and the spill threshold is set to 0.8;
- Each Reducer has 5 replicas threads;
- the sort factor = 10;
- The CPUs processing speeds are 7.5MB/s;
- The sustained read/write speeds of hard disks are 80MB/s and 60MB/s;
- The read/write speeds of memory are 6000MB/s and 5000MB/s.
- Unless otherwise specified, the test results are computed by using this default configuration described above.

Table 5.2: 10 Nodes in 2 frameworks everything is default.

Data Size	With DL (s)	Without DL (s)
10,000 MB	86	101
20,000 MB	165	180
30,000 MB	232	264
40,000 MB	308	335
50,000 MB	382	410
60,000 MB	454	490
70,000 MB	527	563
80,000 MB	602	655
90,000 MB	677	727
100,000 MB	754	800

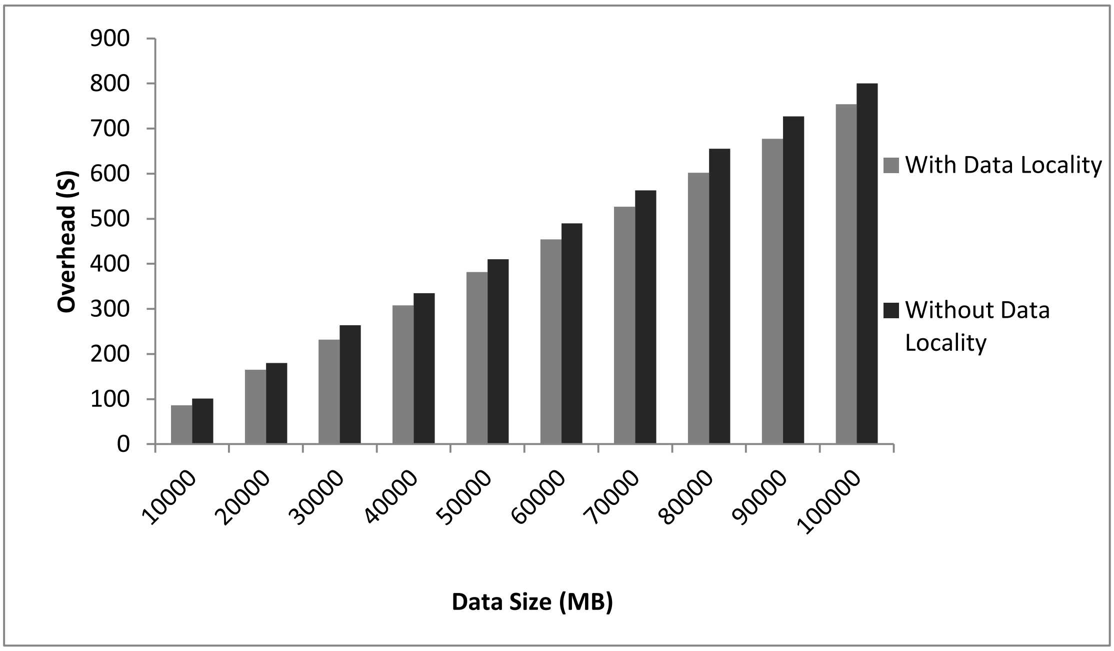


Figure 5.5: 10 Nodes in 2 frameworks everything is default.

The Table 5.2 and Figure 5.5 show that the performance comparison of algorithm in the default test environment. We used same algorithm, one test had used the data locality strategy (the default policy) and the other one had not used the data locality strategy (single replica of random access memory). It can be seen from the reults, when the sizes of input data had increased from 10,000 MB to 100,000MB, the algorithm with data locality (default system with three storage replicas) always used less time comparing with the algorithm without data locality (default system with single replica of the random access memory). In other words, data locality can improve the performance of Monte Carlo algorithm in MapReduce environment whatever the sizes of input data are. That means apply a certain number of replicas of input data chunks in the storage system can bring benefits of the performance improvement.

Table 5.3: 10 Nodes in 2 frameworks, 40 mappers and 1 reducer in total, data size = 51200MB, rep=3, rep=6.

Chunk size(MB)	Rep=3 (s)	Rep=6 (s)
64	390	373
100	396	373
150	414	382
200	417	385
250	422	418
300	430	423
350	438	414
400	466	439
450	461	422
500	473	440
40	384	369
30	388	370
20	374	367
10	618	1190

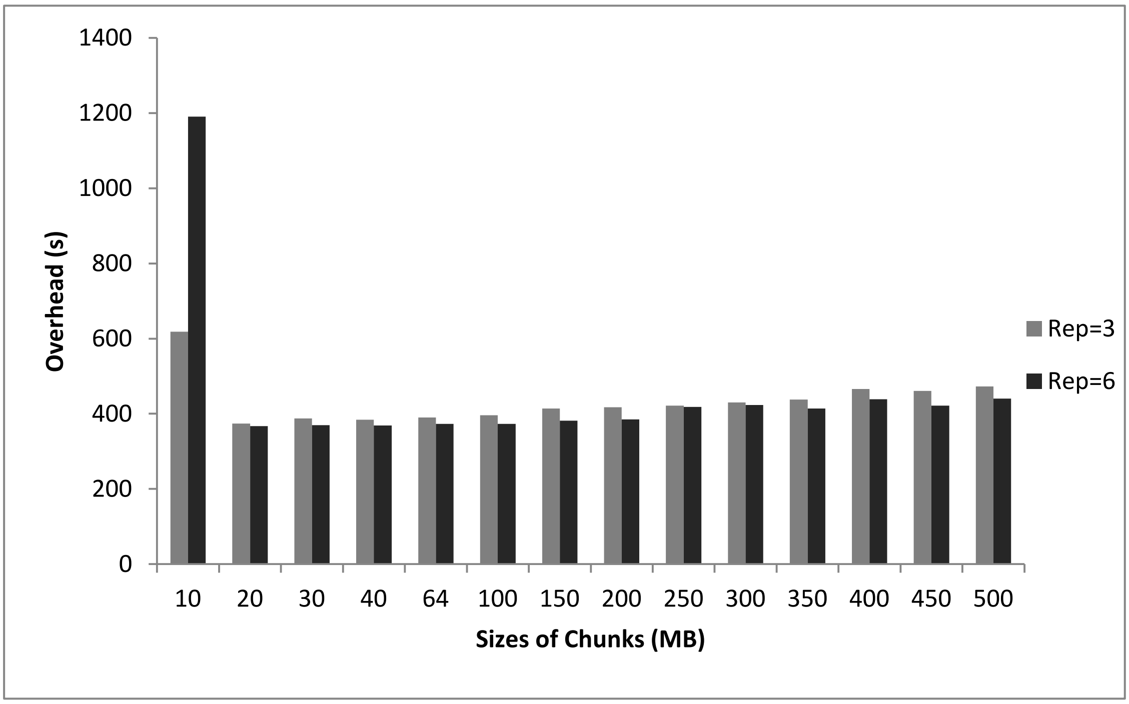


Figure 5.6: 10 Nodes in 2 frameworks, 40 mappers and 1 reducer in total,

data size = 51200MB, rep=3, rep=6.

Figure 5.6 shows that how the performance relationship is between the number of replicas (three replicas or six replicas) and the sizes of chunks. We can see from Figure 5.6, when the size of chunk was very small (such as 10MB), the performance of multiple-replicas was extremely poor. Because of the system cost a lot of time to process the IO. With the increased of size of chunk, the performance of 6 replicas was better than the performance of 3 replicas. The reason is the system IO distribution policy of local multi-replicas instead of the networks streaming IO policy to reduce the network transmission delays. Also can be seen from the Figure 5.6, with the increased of sizes of chunks caused to the gradually increased of time cost of the algorithm implementation. Because of the large size chunks are hard to be replicated. This means there was no local replicas of data. And the transmission efficiency of large size replicas between frameworks is relatively low due to the switches' external interface bandwidth is smaller than the internal interface bandwidth. From the Figure 5.6 we can see that the sizes of chunks from 10MB to 500MB, there is an optimal value between the number of replicas and the size of chunks. To determine this optimal value, we should consider the size of chunks, the network conditions, the number of replicas and the hardware IO capabilities.

Table 5.4: 10 - 100 nodes in 2 frameworks, 4mappers in each node, 1 reducer in total, data size from 2560 - 25600MB, rep=3 and 12, mapper for one wave, chunk size = 64MB.

Node no & Data size MB	Rep=3 (s)	Rep=12(s)
10. 2560	27	23
20. 5120	37	29
30. 7680	49	41
40. 10240	61	55
50. 12800	73	69
60. 15360	84	80
70. 17920	95	95
80. 20480	107	104
90. 23040	122	120
100. 25600	135	128

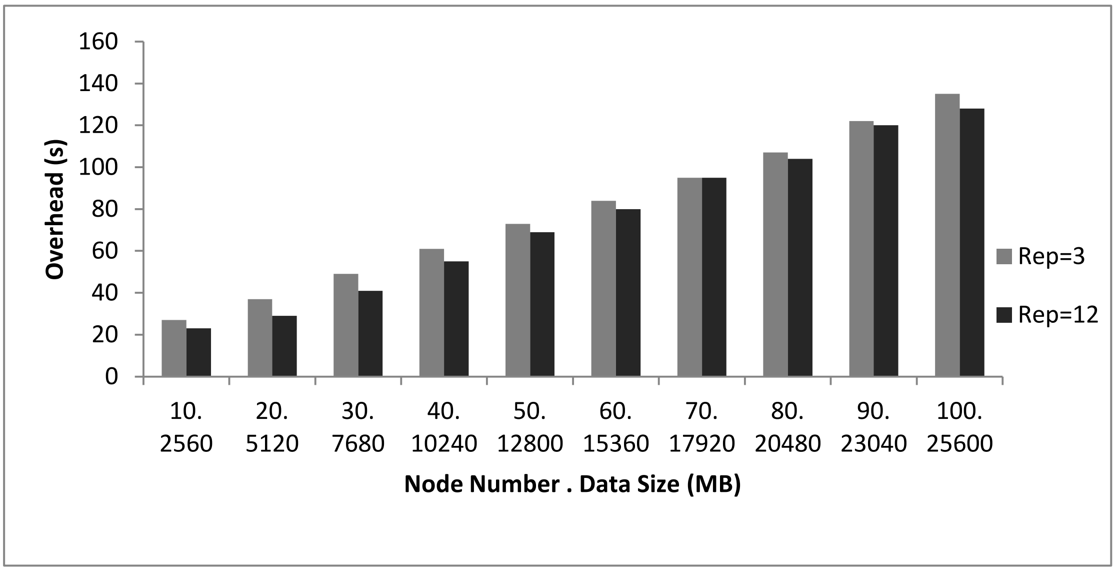


Figure 5.7: 10 - 100 nodes in 2 racks, 4mappers in each node, 1 reducer in total, data size from 2,560 – 25,600MB, rep=3 and 12, mapper for one wave, chunk size 64MB.

The test as shown in Figure 5.7 expressed in a Mapper wave the impaction on the system performance of different number of replicas. It is showed only one Mapper wave because of performance of each wave is same. In order to ensure the test can be completed successfully in the environment where the number of nodes from 10 to 100, so the data is also simultaneously from 2,560 MB to 25,600 MB. Because there are 10 nodes, each node has 4 Mapper and size of chunk is 64 MB, then the data should be 10*4*64=2,560 MB. Empathy, 20 * 4 * 64, 30 * 4 * 64, …, 100 * 4 * 64 = 25,600 MB. From the Figure 5.7 we can see that the performance of more number of replicas was better than the performance of less number of replicas. But more number of replicas likely lead to the overhead of system IO. So the system performance is difficult to linear growth with the increasing of number of replicas, nodes and Mappers. Some Hadoop relevant documents show that with a default replica policy configuration, the performance of certain algorithms will probably be around 10%. The Figure 5.7 also shows that when the numbers of nodes are 70 and 90, the performances are substantially the same. Because of the replicas storage locations are certain randomness. That means replicas storage location (no local replica storage) is difficult to optimize the performance of system IO. So the optimization the replica storage policy can help to improve the performance of Hadoop cluster.

5.7 Summary

This Chapter discussed data locality in Hadoop MapReduce model. It also presented data locality aware schedulings in Hadoop cluster such as DARE scheduling, Delay scheduling, Match making scheduling algorithm, Pre-fetching and Pre-shuffling algorithm and NKS algorithm. It analysed the limitations of each scheduling. The testing results showed the improvement of performance of data locality in Hadoop cluster environment. This chapter showed that for large-scale data computation, the cluster with data locality is much faster than the cluster without data locality strategy. It also evaluated the impact of different sizes of chunks and different numbers of replicas on the performance of a Hadoop cluster.

Chapter 6: Conclusion and Future Works

6.1 Conclusion

This thesis reviewed possible risks within the economic activities and presented the Value-at-Risk tool which can measure those risks. Any risks can lead to loss during in the trading process, so the VaR has become the famous financial tool in the world. During the development of VaR approaches that there are many methods developed to improve the efficiency and practicality of VaR tool such as the Variance-Covariance method, the Historical simulation method and the most famous Monte Carlo simulation method.

In the early years, the Variance-Covariance approach has been widely used by the entire banks to calculate their risk exposure due to the easy calculation of this approach. But the Variance-Covariance method cannot be used for sensitive analysis and its calculation is based on the normal distributed assumption which causes to the calculated value of VaR is lower than the true value of risk. It also misleads the VaR in case by using of wrong input data. For example, incorrect variances or covariance are used in the calculation.

For some financial institutes or firms, the historical simulation method was used to calculate the VaR for their products such as securities or portfolios. This method concerns the products' historical performance when expects the product development trend in future. Which means this method is highly depending on the historical data of a specified asset or product. The limitation of the historical simulation method is if a new asset which doesn't have any historical data at all then the method basically is not working. So all VaR tool users are looking for a comprehensive approach which closer to reality.

The Monte Carlo method is a comprehensive approach developed in the middle of last century, but it was widely used since 1990s due to the rapidly development of computers. The Monte Carlo simulation method uses the random variables as input data, so it does not need to consider the historical data or the performance of the specified asset. The calculated result of the Monte Carlo simulation method is a value of the probability of the event occurred. In other words, it is the approximation value of the event occurred in reality. Compare with other two methods, the Monte Carlo method is the most independent method to calculate the VaR without relying on relevant data. It repeats random samplings and statistical tests to calculate the probability, and then the value of probability is used as the result of VaR. From this point of view, more iterations can get a more accurate probability, this means the calculated value will closer to the reality. In other words, during the calculation process of VaR, the Monte Carlo method relies on large amount of computations. It means more time cost and more works to do. This is the reason of why it had not been widely used in decade's years since the Monte Carlo method was developed. The rapid development of computational ability in last twenty years lets Monte Carlo method ushered in the spring. Now, do more iterations in a possible shortest period of time to complete is a issue was overcame by users. To find a suitable way and platform for computing of the Monte Carlo algorithm was evaluated in this thesis.

The Monte Carlo method completes a round of random sampling and statistic analysis can be concerned as once experiment. The whole process of Monte Carlo method repeats the experiments again and again. At last, all results will be analyzed together and provides an approximation which can be used as the final result of whole experiments. According to the characteristics of Monte Carlo method, parallelization is an efficient way to accelerate the process speed of the simulation. Each thread completes one or more experiments, more threads means more experiments can be completed at the same time. In Monte Carlo simulation, in a fixed period of time, more iterations means more reliable results, so there is a large amount of data to be processed at the same time. From this point of view, in this thesis the MapReduce model was used to handle the parallel Monte Carlo simulation algorithm. The MapReduce was developed by Google for the purpose of processing their large-scale data sets such as web index in its business. In a MapReduce model, all data are divided into many small file chunks which are identified in the type of <key, value> pairs. Those chunks will be passed to the Map function where each Mapper will receive one chunk pair to process. After this step, the output of Mappers are called the intermediate data, which also in the format of <key, value> pairs. Then those pairs are sorted in groups by key and sent to the Reduce function, where the Reducers will process those data and produce the output files. At last all output files will be mixed together into the final result file. For example, in the Monte Carlo algorithm, the process of repetitions of random samplings can be paralleled simulated by Mappers, then the output data will be processed and analyzed by Reducers.

In the experimental environment, a small MapReduce Hadoop cluster was set up by using of four computers, and each computer can be seen as a node. The Monte Carlo algorithm was simulated in one node and four nodes, and then we compared the results differences in the sequential Monte Carlo environment and in the MapReduce parallel Monte Carlo environment. On each platform, there were 128,000 data samples simulated. The correctly computed calculation in sequential Monte Carlo environment was approximately to 96.23% and in MapReduce Monte Carlo environment the correctly computed calculation was approximately to 94.04%. The time cost of the sequential Monte Carlo was approximately to 532 seconds and the time cost of MapReduce Monte Carlo was approximately to 115 seconds. In the sequential Monte Carlo environment there was only one node in use, so it can be seen as only one Mapper was used. The experimental results showed that for the purpose of promise the same level of accuracy, more Mappers used, the better speedup achieved especially on a larger data set. In other words, more Mappers are used means the parallel Monte Carlo algorithm is more faster than sequential Monte Carlo algorithm.

In a large-scale data Hadoop cluster environment, we employed HSim to do further performance evaluation of the parallel Monte Carlo algorithm. The HSim is a Hadoop simulator developed by the research group at Brunel. The cluster contained 250 simulation nodes, there were 4 Mappers and 1 or more Reducer in each node and the size of input data is 100,000 MB. We set different numbers of Reducers to evaluate their impactions on the performance of the parallel Monte Carlo algorithm. The evaluation results showed that the numbers of Reducers did not make more differences on the efficiency of the parallel Monte Carlo algorithm. However, the time cost of calculation had decreased with the increased of numbers of Mappers. Which means the speedup of computation of Monte Carlo is mainly carried by the Mappers. Because of more Mappers means more threads to repeat random samplings of Monte Carlo algorithm. Another evaluation focused on the sort factor in MapReduce. The sort factor is used to sort the output pairs of Mappers in groups before they sent them to Reducers. The test results showed that the higher the sort factor is, the better performance is achieved which means that a lower overhead in computation is incurred.

To further improve the performance of parallel Monte Carlo algorithm in Hadoop MapReduce environment, we focused on the process of scheduling of Hadoop. A Hadoop cluster includes a few or a lot of nodes, each node contain Mappers and Reducers and the input data are stored in the local hard disks. So the performance of the whole cluster is limited by location of data and network bandwidth. In other words, moving data to computation is more difficult than moving computation to the data. To solve this issue, we used the data locality to improve performance of data transmission. We evaluated a parallel Monte Carlo simulation in Hadoop cluster with and without the data locality strategy and set the sizes of input data from 10,000 MB to 100,000 MB. The results showed that the time cost of simulation with data locality strategy is average 25 seconds less than the time cost of simulation without data locality strategy. That means the data locality can efficiently improve the performance of the simulation. Within the data locality strategy, it also has some issues such as the sizes of chunk selection and numbers of replicas selection. In the simulation, we set the sizes of chunk from 10 MB to 500 MB and the numbers of replicas are three and six. The results showed that with large-scale input data when the size of chunk was very small such as 10 MB, the performance of six replicas was extremely poor due to the IO processing cost a lot of time. The performance of six replicas was better than the performance of three replicas when we increased the sizes of chunk, because of the data transmission delays had decreased. Then we set the size of chunk is fixed with different numbers of nodes, different sizes of data and different numbers of replicas such as three and twelve. The results showed that the performance of more replicas was better than the performance of fewer replicas. But the replicas cannot be used too more, otherwise, the performance of Hadoop cluster will decrease.

All results of experimental test and simulation test showed that the parallel Monte

Carlo algorithm can visibly speed up the computation, their performances were much better than the performance of sequential Monte Carlo algorithm. And in a Hadoop MapReduce cluster, the data locality strategy can effectively further improve the performance of parallel Monte Carlo algorithm.

6.2 Future Works

Our future research will focus on the parallel Monte Carlo algorithm in variety distribution systems. In practice, the parallel Monte Carlo simulation can be used by various firms or financial institutes. They may only have a few personal computers which can be used to build up a small distribution system or have dedicated processing cluster or have a lot of different configurations of computers. How to adapt to different hardware environments is a future research direction.

In our experimental or simulation environments, the size of input data is much smaller than that of being used in a large multi-national bank. This algorithm needs to be further improved to deal with huge amounts of data. On the other hand, those banks or financial institutes use Monte Carlo method to identify the VaR of their products or portfolios, usually those products or portfolios are mixed by a lot of different assets and securities. How to identify the relationship between different input data sets for MapReduce parallelization is another research direction.

References

[1]. Beder, T.S. (1995) ‘VAR: Seductive but Dangerous’, *Financial Analysts Journal*, September/October, pp.12-24.

[2]. Bagg, J. (1996) ‘Risk Management – Taking the Wider View’, *International Derivative Review*, June, pp.12-14.

[3]. Garman, M. (1996) ‘Improving on VAR’, *Risk*, 9, 5, pp.61-63.

[4]. Longin, F. (1997) ‘From Value at Risk to Stress Testing: The Extreme Value Approach’, *CERESSEC Working Paper pp.97-004*.

[5]. Kao, D.L. and Kallberg, J.G. (1994) ‘Strategies for measuring and managing risk concentrations in loan portfolios’, Journal of Commercial Lending, January 1994, pp.18-27.

[6]. Fallon, W. (1996) *Calculating Value-at-Risk*, Mimeo, Columbia University.

[7]. Boudoukh,J., Richardson, M., and Whitelaw, M. (1998) ‘The Best of Both Worlds’, *Risk*, v11, pp.64-67.

[8]. Basak, S. and Shapiro, A. (2001) ‘Value-at-Risk Based Management: Optimal Policies and Asset Prices’, *Review of Financial Studies*, v14, pp.371-405.

[9]. Ju, X. and Pearson, N.D. (1998) ‘Using Value-at-Risk to Control Risk Taking: How wrong can you be?’, *Working Paper*, University of Illinois at Urbana-Champaign.

[10]. Embrechts, P. (2000) ‘Extreme Value Theory: Potential and Limitations as an

Integrated Risk Management Tool', *Working Paper*. Available at: http://gloria-mundi.com/Library_Journal_View.asp?Journal_id=5229 [Accessed 11 September 2013].

[11]. Raychaudhuri, S. (2008) 'Introduction to Monte Carlo Simulation', *Proceedings of the 2008 Winter Simulation Conference*: held at Hotel Intercontinental Miami, Miami, FL, USA, 7-10 December 2008. Available at: http://informs-sim.org/wsc08papers/prog08soc.html [Accessed 16th September 2013].

[12]. Larsen, N., Mausser, H. and Ursyasev, S. (2001) '*Algorithms for Optimization of Value-at-Risk'*, Research Report, University of Florida.

[13]. Cormac, B. (1999) *Mastering Value at Risk: a step-by-step guide to understanding and applying VaR*. London: FT Prentice Hall.

[14]. Allen, R. et al. (1997) '*VAR: Understanding and Applying Value-at-Risk'*. London: KPMG Risk Publications.

[15]. Basle Committee on Banking Supervision, (1996) '*Amendment to the Capital Accord to Incorporate Market Risks'*.

[16]. Morgan, J.P. (JPM) (1994-5) *RiskMetrics. Technical Documentation Releases 1-3*. New York: JP Morgan.

[17]. Sharpe, W.F. (1995) *Investments*. Englewood Cliffs: Prentice Hall.

[18]. Jorion, P. (1997) *Value at Risk*. Chicago: Irwin.

[19]. Alexander, C. (1996) *Risk Management and Analysis*. London: John Wiley & Sons Ltd.

[20]. Fabozzi, F. and Modigliani, F. (1996) *Capital Markets*. Upper Saddle River, New Jersey: Prentice Hall.

[21]. Fabozzi, F. and Konishi, A. (1997) *The Handbook of Asset Liability Management*. Chicago: Irwin.

[22]. Galitz, L. (1996) *Financial Engineering*. London: Pitman.

[23]. Ritter, L. and Silber, W. (1993) *Principles of Banking and Financial Markets*. New York: Basic Books.

[24]. Hull, J. (1995) *Introduction of Futures and Options Markets*. New York: Prentice Hall.

[25]. Dennis, J.E. and Schnabel, R.B. (1996) *Numerical Methods for Unconstrained Optimisation and Non-linear Equations*, Englewood Cliffs: Prentice Hall.

[26]. O’Brien, B. (1995) *FX Value-at-Risk Incorporating FX Options*, The Bank of New York.

[27]. Froot, K.A., Scharfstein, D.S. and Stein, J.C. (1993) ‘Risk Management: Coordinating Corporate Investment and Financing Policies’, *Journal of Finance,* 48, pp. 1,629-58.

[28]. Shimko, D. (1995) ‘What is VAR?’, *Risk* 8, 12, pp.27.

[29]. Ogden, J. (1996) ‘Putting all your Risks in One Basket’, *Global Finance*, March, pp.38-40.

[30]. Jordan, J.V. and Mackay, R.J. (1996) ‘Assessing Value-at-Risk for Equity Portfolios: Implementing Alternative Techniques’. *Working Paper, Center for*

study of Futures and Options Markets, Virginia Polytechnic Institute.

[31]. Kupiec, P. (1995) ‘Techniques for Verifying the Accuracy of Risk Measurement Models’, *Journal of Derivatives,* Winter, pp. 73-84.

[32]. Smithson, C. and Minton, L. (1996) ‘Value-at-Risk’, *Risk*, 9,1, pp. 25-29.

[33]. Hendricks, D. (1996) ‘Evaluation of Value-at-Risk Models Using Historical Data’, *Federal Reserve Bank of New York Economic Policy Review*, April, pp.39-70.

[34]. Hsieh, D.A. (1993) ‘Implications of Nonlinear Dynamics for Financial Risk Management’, *Journal of Financial and Quantitative Analysis*, 28, pp.41-64.

[35]. Jackson, P. (1995) ‘Risk Measurement and Capital Requirements for Banks’, *Bank of England Quarterly Bulletin,* 35, no 2, pp.177-184.

[36]. Laycock, M.S. and Paxson, D.A. (1995) ‘Capital Adequacy Risks: Return Normality and Confidence Intervals’, *Bank of England mimeo, Presented at the Annual Meeting of the European Financial Management Association*.

[37]. Marshall, C. and Siegel, M. (1997) ‘Value at Risk: Implementing a Risk Measurement Standard’, *Journal of Derivatives* Spring, pp, 91-111.

[38]. Leong, K. (1996) ‘The Right Approach’, *Risk,* Special Supplement on Value-at-Risk, June, pp. 9-14.

[39]. Little, J.D.C. (1970) ‘Models and Managers: The Concept of a Decision Calculus’, *Management Science,* 16, 8, pp.466-485.

[40]. Merton, R.C. (1994) ‘Influence of Mathematical Models in Finance on Practice: Past, Present and Future’, *Phil, Trans. R. Soc. Lond*., 347, pp.451-463.

[41]. Boudoukh, J., Richardson, M. and Whitelaw, R. (1995) ‘Expect the Worst’, *Risk* 8, 9, pp.100-101.

[42]. Abken, P.A. (2000) ‘An empirical evaluation of value at risk by scenario simulation’, *Journal of Derivatives* 7, pp. 12-30.

[43]. Bremaud, P. (1999) *Markov Chains: Gibbs Fields, Monte Carlo Simulation and Queues*, New York: Springer-Verlag.

[44]. Hammersley, J.M. and Handscomb, D.C. (1964) *Monte Carlo Methods*, New York: Wiley.

[45]. Liu, J. (2001) *Monte Carlo Strategies in Scientific Computing*, New York: Springer-Verlag.

[46]. Propp, J. and Wilson, D. (1996) ‘Exact sampling with coupled Markov chains and applications to statistical mechanics’, *Random Structures and Algorithms*, 9, 1, 2, pp.223-252.

[47]. Rubenstein, R. Y. (1981) *Simulation and the Monte Carlo Method*, New York: John Wiley & Sons.

[48]. Bauer, K.W., Venkatraman, .S. and Wilson, J.R. (1987) ‘Estimation procedures based on control variates with known covariance matrix’, pp.334-341 in *Proceedings of the Winter Simulation Conference*, IEEE Press, New York.

[49]. Beasley, J.D. and Springer, S.G. (1977) ‘The percentage points of the normal distribution’, *Applied Statistics* 26, pp.118-121.

[50]. Boyle, P. P. (1977) ‘Options: a Monte Carlo approach’, *Journal of Financial Economics* 4, pp. 323-338.

[51]. Boyle, P., Broadie, M. and Glasserman, P. (1997) ‘Monte Carlo methods for security pricing’, *Journal of Economic Dynamics and Control,* 21, pp.1267-1321.

[52]. Brace, A., Gatarek, D. and Musiela, M. (1997) ‘The market model of interest rate dynamics’, *Mathematical Finance* 7, pp.127-155.

[53]. Bratley, P., Fox, B. L. and Schrage, L. (1987) ‘*A Guide to Simulation*’, Second Edition, New York: Springer-Verlag.

[54]. Britten, J. M. and Schaefer, S. M. (1999) ‘Non-linear value-at-risk’, *European Finance Review* 2, pp.161-187.

[55]. Bucklew, J. A., Ney, P. and Sadowsky, J. S. (1990) ‘Monte Carlo simulation and large deviations theory for uniformly recurrent Markov Chains’, *Journal of Applied Probability* 27, pp.44-59.

[56]. Cardenas, J., Fruchard, E., Picron, J. F., Reyes, C., Walters, K. and Yang, W. (1999) ‘Monte Carlo within a day’, *Risk* 12, February, pp.55-59.

[57]. Crouhy, M., Galai, D. and Mark, R. (2001) ‘*Risk Management*’, New York: McGraw-Hill.

[58]. Devroye, L. (1986) ‘*Non-Uniform Random Variate Generation*’, New York: Springer-Verlag.

[59]. Duffie, D. and Garleanu, N. (2001) ‘Risk and valuation of collateralized debt obligations’, *Financial Analysts Journal* 57, January and February, pp.41-59.

[60]. Duffie, D. and Glynn, P. (1995) ‘Efficient Monte Carlo simulation of security prices’, *Annals of Applied Probability* 5, pp.897-905.

[61]. Duffie, D. and Pan, J. (1997) ‘An overview of value at risk’, *Journal of*

Derivatives 4, Spring, pp.7-49.

[62]. Duffie, D. and Pan, J. (2001) ‘Analytical value-at-risk with jumps and credit risk’, *Finance and Stochastics* 2, pp.155-180.

[63]. Fishman, G. S. (1996) *‘Monte Carlo: Concepts, Algorithms, and Applications’*, New York: Springer-Verlag.

[64]. Fu, M. C. and Hu, J. Q. (1995) ‘Sensitivity analysis for Monte Carlo simulation of option pricing’, *Probability in the Engineering and Information Sciences* 9, pp.417-446.

[65]. Garcia, D. (2003) ‘Convergence and biases of Monte Carlo estimates of American option prices using a parametric exercise rule’, *Journal of Economic Dynamics and Control* 27, pp.1855-1879.

[66]. Gentle, J. E. (1998) *‘Random Number Generation and Monte Carlo Methods’,* New York: Springer-Verlag.

[67]. Hammersley, J. M. (1960) ‘Monte Carlo methods for solving multivariable problems’, *Annals of the New York Academy of Sciences* 86, pp.844-874.

[68]. Morokoff, W. J. and Caflisch, R. E. (1995) ‘Quasi-Monte Carlo integration’, *Journal of Computational Physics* 122, pp.218-230.

[69]. Moskowitz, B. and Caflisch, R. E. (1996) ‘Smoothness and dimension reduction in quasi-Monte Carlo methods’, *Mathematical and Computer Modelling* 23, pp.37-54.

[70]. Owen, A. B. (1994) ‘Lattice sampling revisted: Monte Carlo variance of means over randomized orthogonal arrays’, *Annals of Statistics* 22, pp.930-945.

[71]. Papageorgiou, A. and Traub, J. (1996) ‘Beating Monte Carlo’, *Risk* 9, June, pp.63-65.

[72]. Rogers, L. C. G. (2002) ‘Monte Carlo valuation of American options’, *Mathematical Finance* 12, pp.271-286.

[73]. Shaw, J. (1999) ‘Beyond VAR and stress testing’, pp.231-244 in *Monte Carlo: Methodologies and Applications for Pricing and Risk Management*, Dupire, B., Risk Publications, London.

[74]. Siegmund, D. (1976) ‘Importance sampling in the Monte Carlo study of sequential tests’, *Annals of Statistics* 4, pp.673-684.

[75]. Sloan, I. H. and Wozniakowski, H. (1998) ‘When are quasi-Monte Carlo algorithms efficient for high dimensional integrals?’, *Journal of Complexity* 14, pp.1-33.

[76]. Tausworthe, R. C. (1965) ‘Random numbers generated by linear recurrence modulo two’, *Mathematics of Computation* 19, pp.201-209.

[77]. Dean, J. and Ghemawat, S. (2008) ‘MapReduce: Simplified Data Processing on Large Clusters’, *Communications of the ACM*, Vol. 51, No. 1, pp.107-113.

[78]. Google MapReduce. Available at: https://developers.google.com/appengine/docs/python/dataprocessing/overview [Last accessed: 09 October, 2013].

[79]. Abad, C. L., Lu, Y. and Campbell, R. H. (2011) “DARE: Adaptive Data Replication for Efficient Cluster Scheduling”, *IEEE Int’l Conf. Cluster Computing,* pp. 159-168.

[80]. Guo, Z., Fox, G. and Zhou, M. "*Investigation of Data Locality in MapReduce*", Published by School of Informatics and Computing Indiana University Bloomington.

[81]. He, B., Fang, W. and Luo, Q. et al. (2008) "Mars: MapReduce Framework on Graphics Processors", *In Proceeding of 17th Int'l Conf. Parallel Architectures and Compilation Techniques, ACM, Toronto, 2008,* pp.260-269.

[82]. Xie, J., Yin, S. and Ruan, X. et al. (2010) "Improving MapReduce Performance through Data Placement in Heterogeneous Hadoop Cluster", *IEEE Int'l Symposium, Parallel and Distributed Processing, Workshop and PhD (IPDPSW),* pp. 1-9.

[83]. Tantisiriroj, W., Patil, S. and Gibson, G. (2008) "*Data-intensive File Systems for Internet Services: A rose by any other name. Technical report*", Pittsburgh: Carnegie Mellon University.

[84]. Yahoo! Launches World's Largest Hadoop Production Application. Available at: http://developer.yahoo.com/blogs/hadoop/posts/2008/02/yahoo-worlds-largest-production-hadoop/ [Last Accessed: 09 October, 2013].

[85]. White, T. (2009) "*Hadoop: The Definitive Guide*", Farnham: O'Reilly Media, Inc.

[86]. Isard, M., Prabhakaran, V. and Currey, J. et al. (2009) "Quincy: Fair Scheduling for Distributed Computing Clusters", *Proceeding in the 22nd Symposium on Operating Systems Principles (ACM SIGOPS).*

[87]. Ranger, C., Raghuraman, R. and Penmetsa, A. et al. (2007) "Evaluating

MapReduce for Multi-core and Multiprocessor Systems", *IEEE Int'l Symposium, High Performance Computer Architecture*, Vol 13, pp. 13–24.

[88]. Chao, T., Zhou, H. and He, Y. et al. (2009) "A Dynamic MapReduce Scheduler for Heterogeneous Workloads", *In Proceeding of the 8th Int'l Conf. Grid and Cooperative Computing*, China, 2009, pp. 218-224.

[89]. Sangwon, S., Ingook, J. and Woo, K. et al. (2009) "HPMR: Prefetching and Pre-shuffling in Shared MapReduce Computation Environment", *in Proceeding of IEEE Int'l Conf. Cluster Computing and Workshop*, 2009, New Orleans, LA, pp. 1-8.

[90]. Jiahui, J., Junzhou, L and Aibo, S. et al. (2011) "BAR: An Efficient Data Locality Driven Task Scheduling Algorithm for Cloud Computing", *in Proceeding of the 11th IEEE/ACM Int'l Symposium, Cluster, Cloud and Grid Computing (CCGrid)*, USA, 2011, pp. 295-304.

[91]. Zaharia, M., Borthakur, D. and Sarma, J. et al. (2010) "Delay Scheduling: Simple Techniques for Achieving Locality and Fairness in Cluster Scheduling", *in Proceeding of the EuroSys 2010, ACM*, New York, pp. 265-278.

[92]. He, C., Lu, Y. and Swanson, D. (2011) "Matchmaking: A New MapReduce Scheduling Technique*", in Proceeding of the 3rd IEEE Int'l Conf. Cloud Computing Technology and Science (CloudCom)*, 2011, pp. 40-47.

[93]. Chen, P. C., Su, Y. L. and Chang, J. B. et al. (2011) "Variable-Sized Map and Locality-Aware Reduce on Public Resource Grid", *ACM Journal, Future Generation Computer Systems*, vol 27, pp.6.

[94]. Zhang, X., Zhong, Z. and Feng, S. et al. (2011) “Improving Data Locality of MapReduce by Scheduling in Homogeneous Computing Environment”, *in Proceeding of the 9th IEEE Int’l Symposium, Parallel and Distributed Processing with Application (ISPA),* 2011, pp. 120-126.

[95]. Isard, M., Prabhakaram, V. and Currey, J. et al. (2007) “Dryad: Distributed Data-Parallel Programs from Sequential Building Blocks”. *In Proceeding of the EuroSys 07, New York, NY, USA, ACM,* 2007, pp. 59-72.

[96]. Gu, Y. and Grossman, R. L. (2009) “Sector and Sphere: The Design and of High- Performance Data Cloud”, *Philosophical Transaction of the Royal Society A: Mathematical, Physical and Engineering Science*, Vol. 376 (1897), pp. 2429-2445.

[97]. Wegener, D., Mock, M., Adranale, D. and Wrobel, S. (2009) “Toolkitbased High-Performance Data Mining of Large Data on MapReduce Clusters”, *in Proceeding of the ICDM Workshops*, 2009, pp. 296–301.

[98]. Zhang, S. Han, J., Liu, Z., Wang, K. and Feng, S. (2009) “Spatial Queries Evaluation with MapReduce”, *in Proceeding of the GCC’09, Gansu, China*, 2009, pp. 287–292.

[99]. Shen, H. and Zhu, Y. (2008) “Plover: A Proactive Low-Overhead File Replication Scheme for Structured P2P Systems”, *in Proceeding of the IEEE Int’l Conf. Communication, ICC 08*, 2008, pp. 5619-5623.

[100]. Zaharia, M., Konwinski, A. and Josepg, A. D. et al. (2008) “Improving MapReduce Performance in Heterogeneous Environment”, *in Proceeding of the*

8th USENIX Symposium on Operating System Design and Implementation, OSDI 2008, San Diego, USA, 2008.

[101]. Chen, Q., Zhang, D. and M. Guo, et al. (2010) “SAMR: A Self Adaptive MapReduce Scheduling algorithm in Heterogeneous Environment”, *in Proceeding of the 10th IEEE Int’l Conf. Computer and Information Technology (CIT’ 10),* 2010, pp. 2736 – 2743.

[102]. Du, H. T. (2000) ‘Application of VaR model in the Security market’ *China Security Market*, Vol 8, pp. 57-61.

[103]. Liu, Y., Li, M. Z., Alham, N. K. and Hammoud, S. (2013) ‘Hsim: A MapReduce Simulator in Enabling Cloud Computing’, *Future Generation Computer Systems*, Vol 29, pp. 300-308.