Computing Resources Sensitive Parallelization of Neural Networks for Large Scale Diabetes Data Modelling, Diagnosis and Prediction

A Thesis submitted for the Degree of

Doctor of Philosophy

By

Hao Qi

Department Electronic and Computer Engineering

School of Engineering and Design

Brunel University, UK

October 2011

Abstract

Diabetes has become one of the most severe deceases due to an increasing number of diabetes patients globally. A large amount of digital data on diabetes has been collected through various channels. How to utilize these data sets to help doctors to make a decision on diagnosis, treatment and prediction of diabetic patients poses many challenges to the research community. The thesis investigates mathematical models with a focus on neural networks for large scale diabetes data modelling and analysis by utilizing modern computing technologies such as grid computing and cloud computing. These computing technologies provide users with an inexpensive way to have access to extensive computing resources over the Internet for solving data and computationally intensive problems.

This thesis evaluates the performance of seven representative machine learning techniques in classification of diabetes data and the results show that neural network produces the best accuracy in classification but incurs high overhead in data training. As a result, the thesis develops MRNN, a parallel neural network model based on the MapReduce programming model which has become an enabling technology in support of data intensive applications in the clouds.

By partitioning the diabetic data set into a number of equally sized data blocks, the workload in training is distributed among a number of computing nodes for speedup in data training. MRNN is first evaluated in small scale experimental environments using 12 mappers and subsequently is evaluated in large scale simulated environments using up to 1000 mappers. Both the experimental and simulations results have shown the effectiveness of MRNN in classification, and its high scalability in data training.

MapReduce does not have a sophisticated job scheduling scheme for heterogonous computing environments in which the computing nodes may have varied computing capabilities. For this purpose, this thesis develops a load balancing scheme based on genetic algorithms with an aim to balance the training workload among heterogeneous computing nodes. The nodes with more computing capacities will receive more MapReduce jobs for execution. Divisible load theory is employed to guide the evolutionary process of the genetic algorithm with an aim to achieve fast convergence. The proposed load balancing scheme is evaluated in large scale simulated MapReduce environments with varied levels of heterogeneity using different sizes of data sets. All the results show that the genetic algorithm based load balancing scheme significantly reduce the makespan in job execution in comparison with the time consumed without load balancing.

Table of Contents

List of Figures

List of Tables

Acknowledgements

First and foremost, I would like to thank my supervisor Dr. Maozhen Li. With his inspiration and guidance, invaluable advice and support, I've built up my strong interest and capability in research. I am also grateful to the EPSRC and China Market Association for their sponsorship in this study. I also would like to thank Dr Maysam Abbod for his valuable advice.

I would like to thank my parents for their continuously unreserved support throughout these years.

I give my sincere gratitude to Yang Liu, Nasullah Khalid Alham, Shiva Prasad Gudimalla and Mohamed A Benshaban for their support in this research.

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

Chapter 1:
Introduction

This chapter briefly gives a background on diabetes, and introduces the challenges in tackling the decease, and presents the major contributions of the thesis.

1.1 Background

Diabetes is decease due to the fact that the body does not produce or properly use insulin. Diabetes is a long-term condition which begins with changes in blood glucose levels. This is known as diabetes mellitus which is also defined as a consequence of the body's failure to effectively control the usage and storage of glucose in the body. This situation does not always lead to serious health problems but in many cases it does lead to kidney failure, foot amputations, blindness and heart attacks. According to Diabetes UK there are 2.8 million people diagnosed with diabetes in the UK in 2011 and a predicted 850,000 people who have diabetes without knowing it[1]. In 1996 there was a global diabetes mellitus spread of 120 million and this was estimated to more than double to 250 million by 2025, due to the increasing population, obesity, lifestyles and changes in diet.

There are two types of diabetes: type 1 and type 2. Type 1 diabetes (insulin-dependent diabetes mellitus) usually starts suddenly at a young age, such as childhood or adolescence. Type 2 diabetes (non-insulin dependent diabetes mellitus) comes on gradually, generally in people aged 40 or over. In most cases, type 2 diabetes mellitus start as a result of resistance to insulin. At first, insulin secretion is increased to overcome this resistance but when islet cells fail type 2 diabetes mellitus may occur. Genetic factors and ethnicity are extremely important in the move to type 2 diabetes. Type 2 diabetes mellitus affects about 15% of the population

[1] http://www.diabetes.org.uk/Guide-to-diabetes/Introduction-to-diabetes/

by the age of 65, whereas type 1 diabetes mellitus only affects about 0.3% of adolescents. diabetes mellitus is linked with some macro vascular complications, such as coronary, cerebral and peripheral vascular diseases, as well as with micro vascular complications like retinopathy, nephropathy and neuropathy. These complications of diabetes mellitus also indicate a link with the quality of glycaemic control. diabetes mellitus is related to sorbitol accumulation and protein glycation, as well as peripheral neuropathy, which predisposes sufferers to foot ulcer development and a low ability to fight infection.

In the developed world, more people who have diabetic foot complications visit a hospital regularly than other diabetic patients and they stay for longer. A large number of diabetic foot problems consist of non-resolving ulcers on weight bearing areas. These ulcers can have very serious consequences. They lead to a high probability of acquiring infections in other parts of the body which will probably spread quickly and damage surrounding tissue.

The diagnostic definition of diabetes mellitus is a raised blood glucose level which occurs after a meal and causes the pancreas to secrete insulin through physico-chemical processes in order to decrease the elevated blood glucose. The pancreas acts as the manager of glucose levels in the blood. In type 1 diabetes, which is insulin-dependent diabetes (IDDM), the pancreas cannot produce the necessary amount of insulin to control the glucose and there is thus a failure of control. In non-insulin dependent, or type 2, diabetes (NIDDM) the pancreas is able to produce some insulin to deal with the level of glucose but not enough. Therefore drug therapy is needed to control the remaining functionality of the feedback system. In the case of type 1 patients control can only be achieved with insulin therapy due to pancreatic failure.

In practice, however, the condition is much more complex. Firstly, in general physiological conditions insulin does help to control the level of glucose but there are other factors as well,

depending on a range of other hormones such as glucagon and adrenalin. Whereas insulin decreases the level of glucose in the blood, other hormones increase the level of glucose in the blood when it is low. These physiological and path physiological state processes occur at different times, making them much more complex to manage. These physiological mechanisms can be analysed to diagnose the disease through assessment, planning, treating and monitoring, which are main responsibilities of decision makers. Moreover, the management of diabetic patients also depends on information about different dynamic characteristics such as diet, exercises, insulin injections, the effects of other drug therapies and alcohol [1].

Although this feedback model gives information for the clinical management it does not deliver the full story that is needed to help with understanding all the complex interactions. This means that in circumstances where clinical decision makers do not have all the patient information needed to make a decision they should take anticipatory action. One way of understanding all these complex interactions is to analyse the dynamics of this system through suitable implementation of dynamic modelling techniques. This technique has been used since the end of the 1970s when, with the advent of the microcomputer, the emphasis changed to the many possible applications of information technology in relation to diabetes. Many computer algorithms and mathematical models, such as causal probabilistic networks and time series methods, are used to give diabetic patients their insulin dosage

The use of computer methods and techniques for analyzing the blood glucose level and the control measures were started in 1960. In 1960 Goldman [2] proposed a cybernetic approach for analyzing the diabetes and control measure of it.

In the late 1970 and early 1980 dynamic modeling had achieved considerable maturity in terms of modeling methodology, incorporation of isomorphic physico-chemical effect and

attention to potential clinical applicability [3,4]. In 1980, microcomputers were developed and now the systems are getting more powerful.

One of the earliest hospital-based databases and information management systems in diabetic clinic was the diabetes system developed at St. Thomas Hospital, London [5]. Later, exert systems were available for advising insulin dosage. Diet maintenance is available for the patient and doctors. Currently these systems are still playing a vital role in human life.

The ability that an expert system can make a decision is based on machine learning techniques such as artificial neural network, decision tree, bagging, k-nearest neighbour (k-NN), support vector machine (SVM), boosting, Bayes net. These machine learning techniques can learn from historical data via a training process, and can help doctors to diagnose the effects of diabetes via a classification process.

1.2 Motivation of the Research

Mathematical models on diabetes study including linear/nonlinear, probabilistic, compartmental/non-compartmental, parametric models have provided a means of understanding diabetes dynamics. However, these models are only valid under certain conditions and assumptions, and their widespread applications have been limited due to uncertainties associated with the estimation of blood glucose profiles. It is desirable to have comprehensive mathematical models featuring hundreds of diabetes factors and variables for accurate diabetes analysis and prediction. For this purpose, extensive computing resources must be made available for such a model to be processed in real time.

The aim of the thesis is to investigate large scale mathematical models for real time diabetes data modeling and analysis by utilizing modern computing technologies such as high performance computing, grid computing and cloud computing. These computing

technologies provide an inexpensive way to utilize a huge amount of computing resources on the Internet for solving data and computationally intensive problems. It is expected that enabled by modern computing technologies an effective mathematical model would be able to make the following scenario happen which will benefit the large group of diabetes patients:

A 65-year-old woman with diabetes has developed high blood pressure but she tells her doctor she is reluctant to take an antihypertensive drug on top of all the medicines she is already taking for her diabetes. Concerned, her doctor logs on to his computer, where he enters her particulars, including her age, weight, race, blood pressure, current medications, family history, and medical history. Then he types in a few questions:

- *What will happen if the patient does not take blood pressure medication?*
- *What would the patient's chances be of having a myocardial infarction or stroke in the next 5 years, in the next 10 years of developing renal failure?*

Then, he asks the computer to calculate the likelihood of such complications if the patient does take the blood pressure lowering drug. The doctor may also ask the computer what lifestyle changes can lower the patient's risk over the next 5–10 years?

In a matter of minutes, the computer provides answers synthesised from data derived from clinical trials, epidemiological data, and based on sophisticated models that can take into account such things as the physiological effects of the different combinations of drugs that might be prescribed. The doctor prints out the answers. He can now give the woman evidence-based projections of what is likely to happen if she decides to take the new medicine or not.

Among machine learning techniques neural networks show an outstanding performance in classification because of their capability in imitating the behaviors of human beings and solving non-linear problems. However, neural networks incur high overhead in classification

when the size of the training data set is large. Extensive research has been carried out in parallelization of neural network to speed up the training process using the Message Passing Interface (MPI) [6, 7, 8, 9]. MPI is primarily targeted at homogeneous computing environments and has limited support for fault tolerance. Furthermore, inter-node communication in MPI environments can create large overheads when shipping data across nodes. Although some progress has been made by these approaches, however, exiting parallel algorithms usually partition large datasets into smaller parts with the same size which can be used efficiently only in homogeneous computing environments in which the computers have similar computing capabilities. Currently heterogeneous computing environments are increasingly being used as platforms for resource intensive distributed applications. One major challenge in using a heterogeneous environment is to balance the computation loads across a cluster of participating computer nodes.

1.3 Research Methodology

The thesis investigates the paradigms in parallelization of neural networks which include data parallelism, training session parallelism, neuron parallelism and weight parallelism. The data parallel approach is adopted in this work as it offers a lower communication overhead than other approaches. A number of machine learning techniques including neural networks are evaluated and their performance in data training and classification is compared. The evaluation results show neural networks performed best in terms of accuracy in classification which further confirms the right decision on the adoption of the model in this research. Eight attributes are identified for training purpose which are number of times of pregnant, plasma glucose concentration measured using a two-hour oral glucose tolerance test, diastolic blood pressure, triceps skin fold thickness, 2-hour serum insulin, body mass index, diabetes pedigree function, age of the patient. A number of parallel and distributed computing technologies are studied in parallelization of a back propagation neural network as it is one of

the most widely used learning algorithms to estimate the values of the weights in neural networks. As a result, a parallel neural network was developed which was initially evaluated in small scale experimental computing environments. Subsequently the parallel neural network is evaluated in large scale simulated computing environments with a focus on its scalability in classification. Both experimental and simulation results show that the parallelized neural network is accurate in classification and highly scalable in training a large amount of data set.

1.4 Major Contributions

The major contributions of the thesis are summarized in the following aspects:

- It evaluates eight representative machine learning techniques in diabetic data training and classification, i.e. artificial neural network, decision tree, bagging, k-nearest neighbour (k-NN), support vector machine (SVM), boosting, Bayes net. Among them neural networks performs best in terms of classification accuracy.
- It designs and implements MRNN, a parallel neural network building on the MapReduce distributed computing model [10, 11, 12]. MapReduce has become an enabling technology in support of data intensive applications in cloud computing. Compared with MPI in parallel computing, the MapReduce framework facilitates a number of important functions such as partitioning the input data, scheduling MapReduce jobs across a cluster of participating nodes, handling node failures, and managing the required network communications. Another feature of MapReduce is its support for heterogeneous environments in which computer nodes may have varied resources in computing.
- It develops a load balancing scheme and integrates the scheme with MRNN to achieve high scalability. The main reason is that MapReduce only provides first-in-

first-out (FIFO) and fair scheduling without load balancing taking into consideration the varied resources of computers [13, 14, 15, 16]. The load balancing scheme is implemented using a genetic algorithm which optimizes resource utilization in heterogeneous computing environments. To deal with the problem of slow convergence of classic genetic algorithms, the divisible load theory [17] is employed in the genetic algorithm which facilitates a quick convergence evaluation process.

- Extensive tests are carried out to evaluate the performance of MRNN and the load balancing scheme. MRNN is initially evaluated in small scale experimental MapReduce environments in terms of its accuracy and efficiency in classification. Subsequently MRNN is evaluated in large scale simulated MapReduce environments. Both experiment and simulation results show the high scalability and efficiency of MRNN in training and classification.

1.5 Organization of the Thesis

The thesis is structured with 6 chapters. The rest of the thesis is organized as follows.

Chapter 2 gives a literature review. It reviews the causes of diabetes, and its treatments from the medical point of view. It also reviews a number of machine learning techniques. This chapter critically assesses existing representative works in parallelization of neural networks in data training and classification, and discusses their limitations.

Chapter 3 reviews diabetes mellitus from a clinical point of view. It introduces the types of diabetes with a focus on type 2 diabetes, analyzes the diagnosis of type 2 diabetes mellitus, the treatments and management of type 2 diabetes. It also discusses the complications of diabetes.

Chapter 4 presents the design and implementation of MRNN, a parallel back propagation neural network building on the MapReduce framework. It gives a brief introduction to

MapReduce, the principles of the programming model, and the functions of mappers and reducers. Then it describes how the neural network is parallelized based on MapReduce.

Chapter 5 presents a genetic algorithm based load balancing scheme which optimizes the utilization of heterogeneous computing resources in MapReduce cluster environments. It introduces genetic algorithms, and describes in detail the implementation of the load balancing scheme. The scheme tries to optimize that at every MapReduce wave all the mappers would finish their jobs at almost the same time. In this way, computing nodes with low computing capacities will receive less jobs than the nodes with high computing capacities.

Chapter 6 evaluates the performance of the load balanced MRNN in diabetic data training and classification. It first introduces the diabetic data set used in the evaluation, and compares the performance of neural networks with that of a number of machine learning techniques. Then it evaluates the performance of MRNN from the aspects of accuracy and efficiency in small scale experimental MapReduce cluster environments. Subsequently it evaluates the performance of MRNN in large scale simulated MapReduce environments. The effectiveness of the load balancing scheme is also evaluated in this chapter.

Finally, Chapter 7 concludes the thesis and points out some future work.

Chapter 2:

Literature Review

This chapter briefly introduces a number of representative machine learning techniques. Finally it discusses some related work on parallelization of neural networks for high efficiency in classification.

2.1 Machine Learning Techniques

This section reviews a number of machine learning based classifiers - Artificial Neural Networks, Bayesian Networks, Decision Trees, Support Vector Machines, k-Nearest Neighbour, and combined classifiers such as Bagging and Boosting.

2.1.1 Artificial Neural Networks

Artificial neural networks (ANNs) [21] are a model used in statistical classifications. ANN consists of an interconnected group of artificial neurons and processes.

Neurons in an ANN are usually grouped into three classes:

- input neurons, which receive information to be processed;
- output neurons, which produce the results of the processing;
- and neurons in between known as hidden neurons.

The input to a neuron consists of a number of values x_1, x_2, ..., x_n, while the output is single value y. Both input and output have continuous values, usually in the range (0, 1). The neuron computes the weighted sum of its inputs, subtracts some threshold T, and passes the result to a non-linear function f. Each element in ANN computes the following:

where w_i are the weights. The outputs of some neurons are connected to the inputs of other neurons. Figure 2.1 shows a neuron model in which x_i represents the input set of the neuron j which will combine the weighted input set, f is a activation (scaling) function, y_j is the output of the neuron, and w_i is the weight set of neuron j.

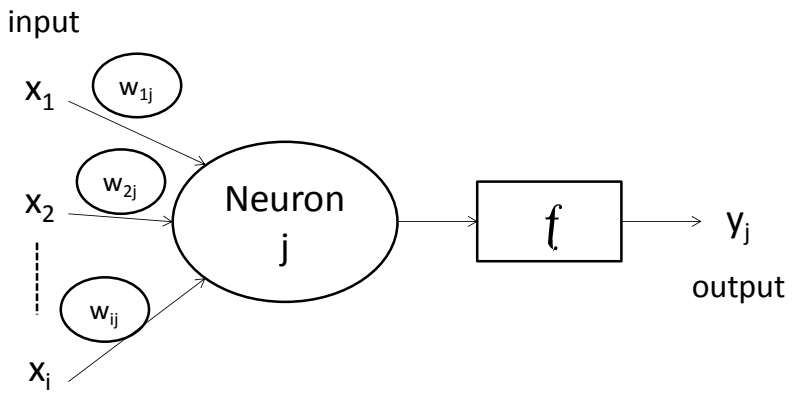

Figure 2.1: A neural model.

A multi-layer perceptron is especially useful for approximating a classification that maps input vector (x_1,x_2, ... x_n) to one or more classes C_1,C_2,...C_m. By optimizing weights and thresholds for all nodes, the network can represent a wide range of classification functions. Supervised learning can be used to optimise the weights so that the network learns from the large number of examples. Gandhi [18] claims ANNs are useful because they can handle non-convex decisions.

A feed-forward neural network is normally processed as a nonlinear mathematical function which transforms a set of input variables into a set of output variables. The process of the transformation is computed by a set of parameters called weights whose values can be determined on the basis of a set of examples of the required mapping. The process of determining these parameters values is often called learning or training, and may be a computationally intensive process. Kotsiantis et al. [19] mentioned that the most well-known and widely used learning algorithm to estimate the values of the weights is the Back

Propagation (BP) algorithm proposed by proposed by Rumelhart et al. [20]. BP networks learn by training - using a learning set that consists of some input examples and the known-correct output for each case. Figure 2.2 shows how weights are updated in the learning process of a neuron in which d_j represents the desired output, and E is the error which is propagated back to the network for updating the weights.

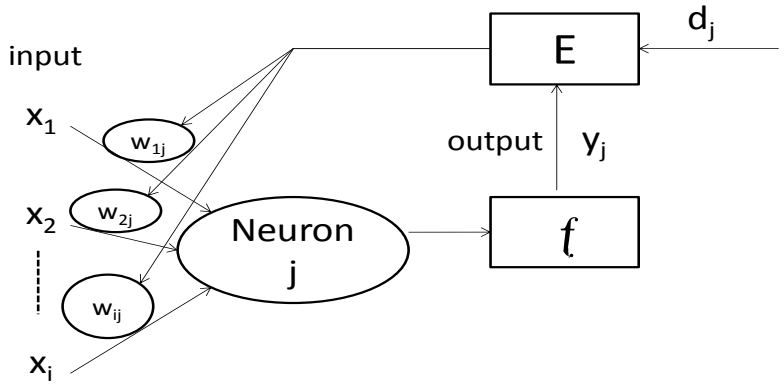

Figure 2.2: Learning process of a neuron.

The learning process of BP networks is done in a number of iterations. The training data is first fed into the network, then the network generates some output using the computed weights of the neurons. At the beginning the output of the network will be random. This output is compared with a given result and an error is calculated. This error is used by each hidden layer to compute the contribution of its neurons to this error which is propagated backwards through the network. As a result, each layer will make some changes to their weights. The whole learning process continues for each of data training cases until the overall error value satisfies a predefined threshold.

2.1.2 Support Vector Machines (SVMs)

The basic idea of SVM [22, 23, 24, 25, 26] is to create a hyperplane as the decision plane, which separates the positive (+1) and negative (-1) classes with the largest margin. An optimal hyperplane is the one with the maximum margin of separation between the two

classes, where the margin is the sum of the distances from the hyperplane to the closest data points of each of the two classes. These closest data points are called Support Vectors (SVs). Given a set of training data D, a set of points of the type

$$D = \{(x_i, c_i) \mid x_i \in \Re^p, c_i \in \{1,1\}\}_{i=1}^{n}$$

where c_i is either 1 or -1 indicative of the class to which the point x_i belongs, the aim is to give a maximum margin hyperplane which divide points having $c_i = 1$ from those having $c_i = -1$.

Any hyperplane can be constructed as a set of point x satisfying;

$$w.x - b = 0$$

The vector W is a normal vector. We want to choose W and b to maximize the margin. These hyperplanes can be described by the following equations:

$$w.x - b = 1$$

$$w.x - b = -1$$

The margin *m* is

$$m = 1/\|w\|_2$$

The dual of the SVM can be shown to be the following optimization problem:

Maximize (in α_i)

$$\sum_{i=1}^{n} \alpha_i - 1/2 \sum_{i,j} \alpha_i \alpha_j y_i y_j x_i x_j$$

Subject to $\alpha_i > 0$ and $\sum_{i=1}^{n} \alpha_i y_i = 0$

There is a one-to-one association between each Lagrange multiplier and each training example. Once the Lagrange multipliers are determined, the normal vector $\vec{w}$ and the threshold b can be derived from the Lagrange multipliers:

$$\vec{w} = \sum_{i=1}^{n} y_i a_i \vec{x}_i$$

$$b = \vec{w}.\vec{x}_k - y_k$$

for some $a_k > 0$. Not all data sets are linearly separable. There may be no hyperplane exist that separate the training examples with positive signs from the examples with negative signs. SVMs can be further generalized to non-linear classifiers. The output of a non-linear SVM is computed from the Lagrange multipliers:

$$u = \sum_{i=1}^{n} y_i a_i K(X_i, X) + b$$

where *K* is a kernel function that measures the similarity or distance between the input vector $\vec{x}$ and the stored training vector $\vec{x}_i$.

2.1.3 Bayesian Networks

Formally, a Bayesian network [27] is directed acyclic graphs in which the nodes represent variables and the edges encode conditional dependencies between the variables. Let $U = \{x_1 ... x_n\}, n \geq 1$ be a set of variables. A Bayesian network B over a set of variables U is a network structure B_S. The classification job is to classify a variable $y = x_0$ called the class variable given a set of variables $x = x_1 ... x_n$ called attribute variables. A classifier $h : x \rightarrow y$ is a function that maps an instance of x to a value of y. The classifier is learned

from a dataset *D* consisting of samples over (x, y). To use a Bayesian network as a classifier, one simply calculates $\arg\max_y P(y|x)$ using the distribution $P(U)$ represented by the Bayesian network. The advantage of using Bayesian Networks is that they can be used to reason in the two different directions. Another advantage of a Bayesian Network is the usefulness of the graph itself; the graph is a compact representation of the knowledge surrounding the system.

2.1.4 k-Nearest Neighbour

The k-Nearest Neighbour (k-NN) algorithm [28, 29, 30] is a non-parametric classifier. The training examples are vectors in a multi dimensional feature space. The space is partitioned into regions by locations and labels of the training samples. A point in the space is assigned to the class *c* if it is the most frequent class label among the *k* nearest training samples. The training stage of the algorithm only stores the feature vectors and class labels of the training samples. In the classification stage, a test sample is represented as a vector in the feature space. Distances from the new vector to all stored vectors are computed and *k* closest samples are selected. There are a number of ways to classify a new vector to a particular class. One of the widely used techniques is to predict the new vector to the most common class amongst the *k* nearest neighbours. Non-parametric classifiers can naturally handle a huge number of classes, and avoid over fitting of parameters which is a central issue in learning based approaches. In addition, non-parametric classifiers do not require learning/training phases.

2.1.5 Composite Classifiers

In machine learning, a number of classifiers can be used together for high accuracy in classifications. They are proposed to improve the classification performance of a single classifier. The combination makes it possible to complement the errors made by the individual classifiers on different parts of the input space.

2.1.5.1 Bagging

In the bagging technique, a number networks are trained separately by different training sets using the bootstrap method [31]. Bootstrapping builds *n* replicated training data sets to construct *n* independent networks by randomly re-sampling the original given training data sets, but with replacements. That is, each training example may appear repeated but not all in any particular replicated training data set of *n*. Then, the *n* networks are combined using an appropriate combination method, such as majority voting.

2.1.5.2 Boosting

The boosting algorithm consists of iteratively learning weak classifiers with respect to a distribution and adding them to a final strong classifier [32]. When they are added, they are typically weighted in a way that is usually related to the weak learner's accuracy. After a weak learner is added, the data is reweighed: examples that are misclassified gain weight and examples that are classified correctly lose weight. Thus future weak learners focus more on the examples that previous weak learners misclassified. One of the main drawbacks of boosting algorithm is in its initial assumptions; hence a large number of training examples are required [33].

2.2 Parallelization of Neural Networks

While the classification capabilities of neural networks are impressive, the training and learning process is computationally intensive especially when the size of a neural network is large. It is worth noting that the structure of feed-forward neural networks is inherently parallel. Extensive research has been carried out in parallelization of neural networks for high efficiency in training [6, 7, 8, 9]. This section reviews the representative works in this aspect.

2.2.1 Parallelization Strategies

There are a variety of parallelization strategies that have been proposed for neural networks [34]. Nordstrom and Svensson [35] identified four strategies that can be used to efficiently parallelize a neural network on a cluster of computers. Each of these strategies represent a different level of granularity.

1. **Data parallelism**. This approach partitions the training dataset into equally sized data segments and distributes these data segments to a number of processors. Each processor has an identical copy of the initial neural network. Once a processor completes its computation calculating the error gradients, it exchanges its gradients with every other processor. After a processor receives gradient information from every other processor, it updates the weights of its neural network. Thus once an iteration is finished, every processor will have an identical copy of the network with weights that have been trained on the entire dataset.
2. **Training session parallelism**. The neural network runs on a number of processors with varied initial training parameters with an aim that one instance of the parallel neural work might be able to find the best solution, and the best parameters can be determined. No communication is needed among the processors which leads to a good speedup. However, interdependence between training instances has to be considered to modify the parameters of the neural network between the training sessions. Seiffert et al. [36] mentioned that such interdependence limits the utility of this parallelism.
3. **Neuron parallelism**, which distributes the computations of individual neurons among the processors for computation. Each neuron operates independently - processing the input, updating the weights, and propagating its output. If the neural network deals with the problems of pattern-update, it is worth noting that only neurons in a single layer can be processed in parallel as the neurons in succeeding layers rely on these

activations and output for their input. If pattern-update is not required, an alternative pipeline approach with a subset of processes for each layer is possible.

If the neural network is not fully interconnected, the locality of connections between neurons can be exploited. A non-overlapping contiguous block of nodes to each processor can be assigned to reduce the communication between processes so that it occurs only across block boundaries.

In the neuron parallelism strategy, each processor must send the output of its neurons for a layer to all processes involved in the computation of the next layer. Consequently, a large number of relatively small messages will be generated. To reduce the overhead in communication, the neurons of a layer are normally allocated to only a subset of the processors instead of the whole set of processors.

4. **Weight parallelism**. Nordstrom and Svensson [35] stated that weight parallelism is the finest grained solution in parallelizing neural networks. In this strategy, the input from each synapse is calculated in parallel for each neuron, and the network input is summed via a suitable communications scheme. As weight parallelism provides no additional capabilities over the neuron parallelism strategy, and it also introduces significantly more short messages, it is not a suitable parallelization strategy for a designing a parallel neural network on a cluster of computers.

2.2.2 Parallelization Algorithms

Initially neural networks were parallelised using special hardware. Work has been done on parallel neural networks either on general purpose single instruction multiple data (SIMD) parallel computers like transputers, the MasPar MP-1, hypercube machines, connection machines (CM-1, CM-2, CM-5), or even on hardware specialized in neurocomputing like the

dynamically reconfigurable extended array multiprocessor (DREAM) or systolic array designs [37, 38, 39, 40, 41, 42, 43].

Numerous algorithms have been designed in parallelizing neural networks using parallel software libraries such as the Message Passing Interface (MPI) and the Parallel Virtual Machine (PVM). Quoy et al. [44] employed PVM in their implementation of a parallel neural network in which the computational pathways (CP) is processed in parallel. Each CP corresponds to a functional part of the global neural network architecture and runs on a different workstation. The network is segmented into a number of independent subgroups and the parallelization is performance at a subgroup level. Mörchen [45] implemented an MPI based parallel neural network on a Beowulf cluster. The data parallel approach is adopted in this work as it offers a lower communication overhead than other approaches. Weight averaging is proposed to compromise between low communication overhead for fast convergence. The calculated weight updates are applied locally with the same block size as used in the sequential version, and the set of weights is averaged over all processors in larger intervals. The results show improvement of performance over pure training set parallelism without introducing convergence problems. Pethick et al [46] investigated a parallel neural network on a cluster of computers using MPI. From the tests they also confirm that generally the data parallel approach outperforms the node parallel strategy. Liu et al [47] presented a parallel neural network using the MapReduce programming model for classification of a large mobile data set. Compared with MPI, the MapReduce framework has a number of additional functions such as targeting at heterogeneous computing environments, handling node failures, and managing the required network communications.

2.3 Summary

Machine learning based classifications have been widely used in diagnosis, prognosis, or detection of certain consequences in diabetes research. Neural network was adopted in this thesis because of its capability in imitating the behaviors of human beings and solving non-linear problems. From the evaluation results presented in Section 6.2 it can be observed that neural network performs best in terms of accuracy in classification compared with other classifiers. However, these results also show that neural network incurs high overhead in the training process ranking second to the worst among the classifiers evaluated. As a result, the main focus of the thesis was to study the methodologies in parallelising neural networks to speed up the process of data training building on the MapReduce framework which has been widely employed in support of data intensive applications.

Chapter 3

A Clinical Review of Diabetes Mellitus

This chapter reviews diabetes mellitus from a clinical point of view including diabetes types, the epidemiology, diagnosis, treatment and complications of type 2 diabetes.

3.1 Types of Diabetes

Almost everyone in the world knows someone who has diabetes. It is a disorder of metabolism— the way the body uses digested food for growth and energy. Most of the food people eat is broken down into glucose - the form of sugar in the blood. Glucose is the main source of fuel for the body, especially for the brain. After digestion, glucose passes into the bloodstream, where it is used by cells for growth and energy. For glucose to get into cells, insulin must be present. Insulin is a hormone produced by the pancreas, a large gland behind the stomach. When people eat, the pancreas automatically produces the right amount of insulin to move glucose from blood into the cells. In people with diabetes, however, the pancreas either produces little or no insulin, or the cells do not respond appropriately to the insulin that is produced, so the level of blood glucose is higher than normal. Glucose builds up in the blood, overflows into the urine, and passes out of the body in the urine. Thus, in people with diabetes, their body loses its main source of fuel even though the blood contains large amounts of glucose.

It is estimated that about 23.6 million people in the United States—7.8 percent of the population—have diabetes, and everyone knows it is a serious, lifelong condition. Of those, 17.9 million have been diagnosed, and 5.7 million have not yet been diagnosed. In 2007, about 1.6 million people ages 20 or older were diagnosed with diabetes. While the number of people suffering from diabetes in People’s Republic of China is more than 90 million and that is amazing.

There are four main types of diabetes :

- type 1 diabetes
- type 2 diabetes
- gestational diabetes
- other types of diabetes

3.1.1 Type 1 Diabetes

Type 1 diabetes is a kind of autoimmune disease. When we talk about an autoimmune disease , it means that it is results when the body's system for fighting infection—the immune system—turns against one part of the body. In diabetes, the immune system attacks and destroys the insulin-producing beta cells in the pancreas. The pancreas then produces little or no insulin. Thus the body can't deal with the blood glucose properly. A person who has type 1 diabetes must take insulin daily to live. At the time about 90 years ago when insulin wasn't found and applied on the patients with type 1 diabetes mellitus, the only fate they would face was to die within a few month or years after being diagnosed.

However, at present, scientists do not know exactly what causes the body's immune system to attack the beta cells although tens of thousands of scientists and physicians devote their time and wisdom to explore the unknown field, but they believe that autoimmune, genetic, and environmental factors, possibly viruses, are involved. Type 1 diabetes accounts for about 5 to 10 percent of diagnosed diabetes in the patients. It develops most often in children , adolescents and young adults but can appear at any age.

In patients with type 1 diabetes, the symptoms such as polydipsia, diuresis, polyphagia and fatigue usually develop over a short period, although beta cell destruction can begin several years earlier. Other symptoms may include increased thirst, constant hunger, weight loss, blurred vision, and so on. If not diagnosed and treated with insulin correctly, a person with

type 1 diabetes can lapse into a life-threatening diabetic coma, also known as diabetic ketoacidosis.

3.1.2 Type 2 Diabetes

Type 2 diabetes is the most common form of diabetes. That means about 90 to 95 percent of people suffering from diabetes belong to type 2. Evidences show that this form of diabetes is most often associated with family history of diabetes, previous history of gestational diabetes, older age, obesity, physical inactivity, and certain ethnicities. About 80 percent of people with type 2 diabetes are overweight and this kind of people are often called "TV potatoes"- that means they are usually fond of sitting on the sofa, eating potato chips while watching TV programs. Type 2 diabetes is being diagnosed increasingly in children and adolescents, especially among African American, Mexican American, and Pacific Islander youth.

When type 2 diabetes is diagnosed, the pancreas is usually producing some insulin, although not enough for the body, and at the same times for unknown reasons the body cannot use the insulin effectively, a condition we called insulin resistance. After several years, insulin production decreases gradually. The result is the same as for type 1 diabetes—glucose builds up in the blood and the body cannot make efficient use of its main source of fuel.

Half of the patients have no symptoms of type 2 diabetes at the time diagnosed, and then they develop gradually, one by one. Their onset is not as sudden as in type 1 diabetes. Symptoms also include fatigue, frequent urination, increased thirst and hunger, weight loss, blurred vision, and slow healing of wounds or sores.

3.1.3 Gestational Diabetes

Some women who have no history of diabetes develop gestational diabetes late in pregnancy. Although this form of diabetes usually disappears after the birth of the baby, women who have had gestational diabetes have a great risk of developing type 2 diabetes within 5 to 10

years. Maintaining a reasonable body weight and being physically active may help prevent development of type 2 diabetes.

It is estimated that about 3 to 8 percent of pregnant women in the United States develop gestational diabetes. As with type 2 diabetes, gestational diabetes occurs more often in some ethnic groups and among women who have a family history of diabetes. Gestational diabetes is considered to be caused by the hormones of pregnancy or a shortage of insulin. Women with gestational diabetes may not experience any symptoms.

3.1.4 Other Types of Diabetes

A number of other types of diabetes exist. A person may exhibit characteristics of more than one type. For example, in latent autoimmune diabetes in adults (LADA), also called type 1.5 diabetes or double diabetes, people show signs of both type 1 and type 2 diabetes.

Other types of diabetes include those caused by

- genetic defects of the beta cell—the part of the pancreas that makes insulin—such as maturity-onset diabetes of the young (MODY) or neonatal diabetes mellitus (NDM)
- genetic defects in insulin action, resulting in the body's inability to control blood glucose levels, as seen in leprechaunism and the Rabson-Mendenhall syndrome
- diseases of the pancreas or conditions that damage the pancreas, such as pancreatitis and cystic fibrosis
- excess amounts of certain hormones resulting from some medical conditions—such as cortisol in Cushing's syndrome—that work against the action of insulin
- medications that reduce insulin action, such as glucocorticoids, or chemicals that destroy beta cells
- infections, such as congenital rubella and cytomegalovirus
- rare immune-mediated disorders, such as stiff-man syndrome, an autoimmune disease of the central nervous system

- genetic syndromes associated with diabetes, such as Down syndrome and Prader-Willi syndrome

3.2 The Epidemiology of Diabetes Type 2

In the four types of diabetes, type 2 diabetes is the predominant form all around the world, accounting for about 90% to 95% of all cases as we mentioned before [48]. An epidemic of type 2 diabetes is under way in both developed and developing countries, for example, there is a famous article published on The New England Journal of Medicine(NEJM) by Professor Yang Wenying which reported amazing number of people with diabetes in China [49]. Studies in Hispanic populations, Native American and Canadian communities, Pacific and Indian Ocean island populations, and in India and Australian Aboriginal communities show that the disorder is disproportionately in non-European populations [50]. The spread of the speed of this disease is increasing so fast because of the changing lifestyle that governments around the world have to deal with it seriously. A case is the Pacific island of Nauru, in which diabetes was virtually unknown 50 years ago and is now present in approximately 40% of adults. Previously, the number of people with diabetes is expected to rise from the current estimate of 150 million to 220 million in 2010 and 300 million in 2025, while we have to change it because alarming increases in the prevalence of diabetes have occurred in various countries especially in China and India. Type 2 diabetes has become one of the world's most important public health problems. It can totally destroy the efforts the developing such countries made to their economy.

There are many factors that are responsible for the development of type 2 diabetes, and some are summarized in Table 3-1.

Nowadays, type 2 diabetes is considered to occur in genetically predisposed individuals, and then they are exposed to a series of environmental influences that precipitate the onset of clinical disease. The genetic basis of type 2 diabetes is complicated, but the syndrome

consists of monogenic and polygenic forms that can be differentiated both on clinical grounds and in terms of the genes that are involved in the pathogenesis of these disorders.

Table 3-1: Epidemiologic determinants and risk factors of type 2 diabetes [53].

Genetic factors
Genetic markers, family history, "thrifty gene(s)"
Demographic characteristics
Sex, age, ethnicity
Behavioral and lifestyle-related risk factors
Obesity
Less of Physical Activity
Diet
Stress
Westernization
Metabolic determinants and intermediate risk categories of type 2 diabetes
Impaired glucose tolerance
Insulin resistance
Pregnancy-related determinants

Some factors such as sex, age, and ethnic background are important in determining risk for the development of type 2 diabetes [51]. As we know now that the disorder is more common in females, and in certain racial and ethnic minority groups the prevalence can increase rapidly. Age is another critical factor. When someone is getting older and older, many organs can degenerate including pancreas which can cause less insulin be produced and the blood glucose will increase as a result. That’s the reason why type 2 diabetes has been viewed in the

past as a disorder of aging with an increasing prevalence with age. This is true even today. However, another trend which bother parents heavily has become apparent in which the prevalence of obesity and type 2 diabetes in children and adolescent is rising dramatically. In the past, scientists believed that the overwhelming majority of children with diabetes had type 1 diabetes, while only 1% to 2% of children considered to have type 2 or other forms of diabetes [52]. But recent reports suggest that about 8% to 45% of children with newly diagnosed diabetes are not caused by immune factors. The surveys also shows that although the majority of these children have type 2 diabetes, other types are also being increasingly identified. In the black population, an idiopathic non–immune-mediated form of diabetes has been reported.

3.2.1 Diagnostic Criteria for Type 2 Diabetes Mellitus

The diagnosis of type 2 diabetes depends on the measurement of plasma glucose levels. The diagnostic criteria for diabetes has changed many times and the most recent one was in 1997 [54]. The most significant changes is the cut point of the level of fasting plasma glucose (FPG) which is recognized as diagnostic factor for diabetes and it was decreased from 140 to 126 mg/dL. Current criteria for the diagnosis of diabetes, impaired fasting glucose and impaired glucose tolerance (IGT) are shown in Table 3-2.

Because plasma glucose concentrations change continuously, the criteria are based on estimates of the threshold for the complications of diabetes which is called “cut point”. The primary end point used to evaluate the relationship between glucose levels and complications is retinopathy. The prevalence of retinopathy in comparison with FPG and 2-hour plasma glucose has been evaluated.

Table 3-2: Criteria for diagnosis of diabetes.

Normoglycemia	Impaired Fasting Glucose or Impaired Glucose Tolerance	Diabetes *
FPG <110 mg/dL	FPG ≥110 and <126 mg/dL (IFG)	FPG ≥126 mg/dL
2-hr PG <140 mg/dL	2-hr PG ≥140 and <200 mg/dL (IGT)	2-hr PG ≥200 mg/dL
		Symptoms of diabetes and casual plasma glucose concentration ≥200 mg/dL
FPG: fasting plasma glucose IFG: impaired fasting glucose IGT: impaired glucose tolerance PG : plasma glucose.		

* A diagnosis of diabetes must be confirmed on another day by measurement of FPG, 2-hour PG, or random plasma glucose if symptoms are not present. The FPG test is greatly preferred because of ease of administration, convenience, acceptability to patients, and lower cost while we can't ignore 2-hour PG which can lead to wrong evaluation. Fasting is defined as no caloric intake for at least 8 hours.

In two famous large studies, DCCT & UKPDS, there is an association between FPG and 2-hour plasma glucose and risk of macrovascular and cardiovascular disease. Another study which is called the Paris Prospective Study also showed that the incidence of fatal coronary heart disease was related to both FPG and 2-hour plasma glucose that were determined at a baseline examination [55].

There will be significant variation in the results if we repeat tests in adults after a 2- to 6-week interval. So reproducibility of the plasma glucose concentration is an important issue when we explain the results of diagnostic tests for diabetes. The intraindividual coefficient of variation in one study was 6.4% for the FPG and 16.7% for the 2-hour plasma glucose value. Thus, it is very important that we need to confirm abnormal results by a repeated test. Although the oral glucose tolerance test (OGTT) is an invaluable tool in research, some scientists believed it is NOT recommended for routine use in the diagnosis of diabetes. Some physicians in developed countries believe that in the vast majority of cases the diagnosis can be made on the basis of either an elevated fasting glucose concentration or an elevated random glucose determination in the presence of hyperglycemic symptoms. But in most Chinese big hospitals we usually operate OGTT and insulin release tests to find more and more patients with diabetes and pre-diabetes. Although it is a little bit expensive and inconvenient for patients, we think such tests are valuable compared to the cost of complications of diabetes.

Levels of hemoglobin A_{1c} (HbA_{1c}) - the most effective method for monitoring the effectiveness of diabetes treatment were not recommended for the diagnosis of diabetes. The major reasons maybe were the lack of standardization of the assays for HbA_{1c} and the imperfect correlation between HbA_{1c} and FPG and 2-hour plasma glucose. But recently, scientists changed their opinions. In the "Standands of Mdeical Care in Diabetes-2011"-brought out by ADA, $HbA_{1c} \geq 6.5\%$ is one of the current criteria of diagnosis of diabetes.

Current criteria for the diagnosis of diabetes [56]:

- A1C ≥6.5%. The test should be performed in a laboratory using a method that is National Glycohemoglobin Standardization Program (NGSP)-certified and standardized to the Diabetes Control and Complications Trial (DCCT) assay.

- fasting plasma glucose (FPG) ≥126 mg/dl (7.0 mmol/l). Fasting is defined as no caloric intake for at least 8 h, or
- 2-hour plasma glucose ≥ 200 mg/dl (11.1mmol/l) during an oral glucose tolerance test (OGTT). The test should be performed as described by the World Health Organization, using a glucose load containing the equivalent of 75 g anhydrous glucose dissolved in water.
- in a patient with classic symptoms of hyperglycemia or hyperglycemic crisis, a random plasma glucose ≥ 200 mg/dl(11.1 mmol/l).
- in the absence of unequivocal hyperglycemia, result should be confirmed by repeat testing. Testing for diabetes in asymptomatic patients.

Detection and diagnosis of gestational diabetes mellitus (GDM):

- Screen for undiagnosed type 2 diabetes at the first prenatal visit in those with risk factors, using standard diagnostic criteria.
- In pregnant women not known to have diabetes, screen for GDM at 24 –28 weeks of gestation, using a 75-g 2-h OGTT and the diagnostic cut points in the "Standards of Medical Care in Diabetes—2011".
- Screen women with GDM for persistent diabetes 6–12 weeks postpartum.
- Women with a history of GDM should have lifelong screening for the development of diabetes or prediabetes at least every 3 years.

3.2.2 Screening for Type 2 Diabetes

It is estimated that there are about 5 to 7 years between the onset of diabetes and diagnosis .The amount of people between normal and diabetes, which is called pre-diabetes including IGT,IFG etc., is as much as that with diabetes [57]. Up to 50% of affected people, diabetes and pre-diabetes are undiagnosed [58]. Although unknown, people with IGT and

undiagnosed type 2 diabetes are at significantly increased risk for coronary heart disease, stroke, and peripheral vascular disease. Thus, this delay in the diagnosis of type 2 diabetes causes an increase in complications such as microvascular and macrovascular disease. In addition, affected individuals have a greater likelihood of having dyslipidemia, hypertension, and obesity, all of which may caused by the same reason-insulin resistence. Therefore, it is very important for the physicians to screen for diabetes or pre-diabetes despite the cost in subjects who demonstrate some major risk factors for diabetes which are showed as summarized in Table 3-3. Recommendations for screening are summarized below by American Diabetes Association in 2011.

Table 3-3: Major risk factors for type 2 diabetes [59].

Family history of diabetes (including grandparents, parents ,siblings , children, grandchildren ,cousins etc. with diabetes)
Overweight (BMI ≥25 kg/m^2)
Physical inactivity
Ethnicity (e.g., Native Americans, Asian or Pacific Islanders)
IFG or IGT
Hypertension (≥140/90 mm Hg in adults)
HDL cholesterol ≤35 mg/dL (0.90 mmol/L) and/or a triglyceride level ≥250 mg/dL (2.82 mmol/L)
History of GDM or delivery of a baby weighing >9 lb
Polycystic ovary syndrome
BMI: body mass index GDM: gestational diabetes mellitus

HDL: high-density lipoprotein

IFG: impaired fasting glucose

IGT: impaired glucose tolerance.

Testing for diabetes in asymptomatic patients:

- Testing to detect type 2 diabetes and assess risk for future diabetes in asymptomatic people should be considered in adults of any age who are overweight or obese (BMI _25 kg/m2) and who have one or more additional risk factors for diabetes. In those without these risk factors, testing should begin at age 45 years.
- If tests are normal, repeat testing carried out at least at 3-year intervals is reasonable.
- To test for diabetes or to assess risk of future diabetes, A1C, FPG, or 2-h 75-g OGTT are appropriate.
- In those identified with increased risk for future diabetes, identify and, if appropriate, treat other cardiovascular disease (CVD) risk factors.

3.2.3 Pathogenesis of Type 2 Diabetes

The pathogenesis of type 2 diabetes is very complicated. Till today, the scientists find some monogenic factors while they still can't tell things like the interaction of genetic and environmental factors. As it is well known, some environmental factors have been demonstrated to play a critical role in the development of diabetes, especially excessive caloric intake –well known as western lifestyle which affect developing countries now leading to overweight and obesity. The clinical presentation is also quite different among the patients, for instance, severity of associated hyperglycemia, degree of obesity and kinds of complications. People with type 2 diabetes consistently demonstrate three cardinal abnormalities:

1. resistance to the action of insulin in peripheral tissues, particularly muscle , fat and liver.
2. defective insulin secretion, particularly in response to a glucose stimulus.
3. increased glucose production by the liver.

Although we have some understandings of the onset of type 2 diabetes , we still don't know the precise manner in which these genetic, environmental, and pathophysiologic factors interact to lead to the disease. It is gradually agreed among the scientists and physicians in this field that the common forms of type 2 diabetes are polygenic in nature and are due to a combination of abnormal insulin secretion and insulin resistance. From a pathophysiologic view, it is the inability of the pancreatic beta cell to adapt to the reductions in insulin sensitivity that occur over the lifetime of human beings in response to puberty or pregnancy, a sedentary lifestyle, or overeating leading to weight gain that precipitates the onset of type 2 diabetes.

3.2.4 Genetic Factors in the Development of Type 2 Diabetes

Genetically, type 2 diabetes consists of monogenic and polygenic forms. Although uncommon by most people, the monogenic forms are still important and scientists have found a number of the genes involved. But the genes involved in the common polygenic form or forms of the disorder are very difficult to be identified and characterized.

3.2.4.1 Monogenic Form of Diabetes

The patients with the monogenic forms, the gene involved is both necessary and sufficient to cause the disease. We can also that environmental factors play little or no role in determining whether or not a genetically predisposed individual develops clinical diabetes. Generally, the monogenic forms of diabetes occur in youth or adolescents, that is the first two to three decades of life.

Table 3-4: Monogenic forms of diabetes.

Associated with insulin resistance
Mutations in the insulin receptor gene
Type A insulin resistance
Leprechaunism
Rabson-Mendenhall syndrome
Lipoatrophic diabetes
Mutations in the PPAR-γ gene
Associated with defective insulin secretion
Mutations in the insulin or proinsulin genes
Mitochondrial gene mutations
Maturity-onset diabetes of the young (MODY)
Mutations in the genes for
HNF-4α (MODY 1)
Glucokinase (MODY 2)
HNF-1α (MODY 3)
IPF-1 (MODY 4)
HNF-1β (MODY 5)
NeuroD1/Beta2 (MODY 6)
HNF: hepatocyte nuclear factor IPF: insulin promoter factor MODY: maturity-onset diabetes of the young NeuroD1/Beta2: neurogenic differentiation 1/beta cell E-box trans-activator 2 PPAR: peroxisome proliferator-activated receptor.

The monogenic forms of diabetes are summarized in Table 3-4 and can be divided into two types:

- the mechanism is a defect in insulin secretion
- defective responses to insulin which is called insulin resistance.

3.2.4.2 Insulin resistance

Insulin resistance is well known among the physicians and patients since it plays a key role in the development of glucose intolerance and diabetes especially in those overweight or obese. Insulin resistance is a common finding in patients with type 2 diabetes, and it can exist many years before the onset of diabetes [60]. Many randomized, multi-center, prospective trials show that insulin resistance predicts the onset of diabetes [61].

Insulin resistance should be considered by decreased insulin-stimulated glucose transport and metabolism in adipocytes and skeletal muscle and by impaired suppression of hepatic glucose output. Many factors including age, weight, ethnicity, body fat (especially abdominal), physical activity, and medications can influence insulin sensitivity and cause insulin resistance.

Thus, the term insulin resistance can be defined as the presence of an impaired biologic response to insulin (either exogenously administered or endogenously secreted). Insulin resistance is associated with the progression to IGT and type 2 diabetes. We can't forget that diabetes is rarely seen in insulin-resistant persons without some degree of beta cell dysfunction [62].

For those who are first-degree relatives of type 2 diabetics ,they have insulin resistance even at a time when they are non-obese, which imply a strong genetic component in the development of insulin resistance. There is also a strong influence of environmental factors on the genetic predisposition to insulin resistance and to diabetes.

3.2.4.3 Hyperinsulinemia and Insulin Resistance

Hyperinsulinemia means elevated concentrations of insulin and it can cause insulin resistance by down-regulating insulin receptors and desensitizing postreceptor pathways. Some surveys showed that after 24 and 72 hours of sustained physiologic hyperinsulinemia in normal individuals , we can find the ability of insulin to increase nonoxidative glucose disposal specifically inhibited in association with an impaired ability of insulin to stimulate glycogen synthase activity.

3.3 Insulin Secretion and Type 2 Diabetes

It is common sense that normal insulin secretory function is essential for the maintenance of normal glucose tolerance, while abnormal insulin secretion is invariably present in patients with type 2 diabetes.

3.3.1 Quantitation of Beta Cell Function

It is widely used to measure peripheral insulin concentrations by radioimmunoassay and it can be a good method for quantifying beta cell functions in vivo. But there are many defects:

- Physicians can use the results to evaluate the function of beta cells of pancreas, but we still can’t count the precise percent of remained function.
- It is also limited by the fact that 50% to 60% of the insulin produced by the pancreas is extracted by the liver without ever reaching the systemic circulation [63].
- The standard radioimmunoassay for the measurement of insulin concentrations is also unable to distinguish between endogenous and exogenous insulin, making it ineffective as a measure of endogenous beta cell reserve in the insulin-treated diabetic patient. Under such conditions, we can use c-peptide as a substitute.
- Anti-insulin antibodies can make insulin measurements in insulin-treated patients inaccurate.

- Conventional insulin radio-immunoassays are also unable to distinguish between levels of circulating pro-insulin and true levels of circulating insulin. Fortunately, such technique can be achieved now.

Insulin is derived from a single-chain precursor, pro-insulin. Pro-insulin can be divided into three forms: insulin, C peptide, and two pairs of basic amino acids. Insulin is subsequently released into the circulation at concentrations equal to those of C peptide. In addition, small amounts of intact pro-insulin and pro-insulin conversion intermediates are released and both can be detected in the circulation, where they constitute 20% of the total circulating insulin-like immunoreactivity. In vivo, proinsulin has a biologic potency that is only about 10% of that of insulin and the potency of split proinsulin intermediates is between those of proinsulin and insulin. C peptide has no known conclusive effects on carbohydrate metabolism, although certain physiologic effects of C peptide have been proposed. Unlike insulin, C peptide is not extracted by the liver and is excreted almost exclusively by the kidneys. Its plasma half-life of approximately 30 minutes contrasts sharply with that of insulin, which is approximately 4 minutes.

Because C peptide is secreted in equimolar concentrations with insulin and is not extracted by the liver, many investigators have used levels of C peptide as a marker of beta cell function. The use of plasma C-peptide levels as an index of beta cell function is dependent on the critical assumption that the mean clearance rates of C peptide are constant over the range of C-peptide levels observed under normal physiologic conditions. This assumption has been shown to be valid for both dogs and humans, and this approach can be used to derive rates of insulin secretion from plasma concentrations of C peptide under steady-state conditions. However, because of the long plasma half-life of C peptide, under non–steady-state conditions (e.g., after a glucose infusion) peripheral plasma levels of C peptide do not change in proportion to the changing insulin secretory rate. Thus, under these conditions, insulin

secretion rates are best calculated with use of the two-compartment model initially proposed by Eaton and co-workers. Modifications to the C-peptide model of insulin secretion have been introduced. This approach combines the minimal model of insulin action with the two-compartment model of C-peptide kinetics and allows insulin secretion and insulin sensitivity to be derived after either intravenous or oral administration of glucose.

3.3.2 Insulin Secretion in Type 2 Diabetes Mellitus

Because of the presence of insulin resistance, the level of blood insulin in patients with type 2 diabetes are often high which is called hyperinsulinemia. People who has hyperinsulinemia while blood glucose level is normal means a condition of pancreas function between the real "normal" & abnormal. If such condition is ignored, the burden of beta cell will continue or even heavier and gradually the reservation of beta cell will exhaust, eventually the level of insulin decreased sharply and the blood glucose level increase from normality to a high level even cause emergent condition such as diabetic ketosis acidosis (DKA).If given an intravenous glucose load, the beta cell defect in patients with type 2 diabetes mellitus could be seen an absent first-phase insulin and C-peptide response and a reduced second-phase response.

In the definition of Diabetes Mellitus , insulin secretion defect and/or insulin response defect are the most important factors that cause the disease. In patients with overt insulin resistance we couldn't ignored the idea that they may have an intrinsic defect in the beta cell. In some studies, abnormalities in first-phase insulin secretion have also been observed in first-degree relatives of patients with type 2 diabetes who have only mild glucose intolerance. In other trials, the insulin response to oral glucose has been observed decreased in normoglycemic co-twins of patients with type 2 diabetes—a group at high risk for type 2 diabetes and who can legitimately be classified as pre-diabetic. Beta cell abnormalities may therefore happen before the development of overt type 2 diabetes for many years.

In patients with type 2 diabetes pro-insulin levels are usually high in serum as well as the pro-insulin/insulin molar ratio. It seems that the amount of pro-insulin produced in blood are related to the blood glucose level no matter how long the diabetic state continue.

In people without type 2 diabetes, the ratio of basally and postgrandially secreted insulin is equal while in patients with type 2 diabetes insulin it is much more under basal conditions. The scientists believe that the reduction in the proportion of insulin secreted postprandially is related in part to a reduction in the amplitude of the secretory pulses of insulin occurring after meals rather than to a reduction in the number of pulses. There are some similar findings in patients with IGT studied under the same experimental conditions and in a further group of type 2 diabetic patients studied under fasting conditions.

The most important question for patients with type 2 diabetes appears to be how to save the function of beta cell. An enhancement of beta cell secretory activity absolutely can improve the blood glucose level. This increased endogenous production of insulin seems to be independent of how you treat the disease and is in particular associated with increases in the amount of insulin secreted postprandially. Although improvements in glycemic control can be observed in some patients after appropriate treatment, beta cell function is not normalized to the totally normal level, suggesting that the intrinsic defect in the beta cell persists.

Sulfonylurea glyburide is one kind of medicine used by many patients since it can increase the amount of insulin secreted in response to meals but it cannot correct the underlying abnormalities in the pattern of insulin secretion.

Some trials have also investigated the effects on insulin secretion of improving insulin resistance in subjects with IGT by using the insulin-sensitizing agent troglitazone, a thiazolidinedione. Nowadays , this medicine has been forbidden by FDA for its side effect. The reason that Troglitazone therapy can improve insulin sensitivity is that it can enhance ability of the pancreatic beta cell to respond to a glucose stimulus.

3.4 Complications of Diabetes Mellitus

Although diabetes can be considered a complicated syndrome, all forms of diabetes, both inherited and acquired, are characterized by hyperglycemia, a relative or absolute lack of insulin, and the development of diabetes-specific microvascular pathology in the retina, renal glomerulus, and peripheral nerve. Diabetes is also associated with accelerated macrovascular disease affecting arteries that supply the heart, brain, and lower extremities. Pathologically, this condition resembles macrovascular disease in nondiabetic patients but is more extensive and progresses more rapidly. As a consequence of its microvascular pathology, diabetes mellitus is now the leading cause of new blindness and end-stage renal disease (ESRD).

The patients with diabetes mellitus are the fastest growing group of renal dialysis and transplant recipients. The life expectancy of patients with diabetic end-stage renal failure is only 3 or 4 years. More than 60% of diabetic patients are affected by neuropathy, which includes distal symmetrical polyneuropathy(DSPN), mononeuropathies, and a variety of autonomic neuropathies causing erectile dysfunction, urinary incontinence, gastroparesis, and nocturnal diarrhea. Accelerated lower extremity arterial disease in conjunction with neuropathy makes diabetes mellitus account for 50% of all nontraumatic amputations in the United States. The risk of cardiovascular complications is increased by twofold to sixfold in subjects with diabetes. Overall, life expectancy is about 7 to 10 years shorter than for people without diabetes mellitus because of increased mortality from diabetic complications. That’s the reason why more and more government consider this disease a great enemy for their people [64].

3.4.1 Retinopathy, Macular Edema and Other Ocular Complications

Diabetic retinopathy is a well-characterized, sight-threatening, chronic microvascular complication that affect almost all patients with diabetes mellitus. It is characterized by

gradually progressive alterations in the retinal microvasculature, leading to areas of retinal nonperfusion, increased vasopermeability, and pathologic intraocular proliferation of retinal vessels. The complications associated with the increased vasopermeability, termed macular edema, and uncontrolled neovascularization, termed proliferative diabetic retinopathy (PDR), can result in severe and permanent visual loss. Despite decades of research, there is presently no known means of preventing diabetic retinopathy and, despite effective therapies, diabetic retinopathy remains the leading cause of new-onset blindness in diabetic patients.

During the whole process of diabetes it is closely associated with the onset and severity of diabetic retinopathy. In patients with type 2 diabetes, approximately 20% have retinopathy at the time of diabetes diagnosis and most have some degree of retinopathy later. It has been reported that nearly all patients with type 1 diabetes and more than 60% of patients with type 2 diabetes develop some degree of retinopathy after 20 years of diagnosis [65].

Among many kinds of diseases, diabetic retinopathy is the most frequent cause of new-onset blindness in patients with type 2 diabetes aged 20 to 74 years. In a study on diabetic retinopathy, approximately 4% of patients younger than 30 years of age at diagnosis and nearly 2% of patients older than 30 years of age at diagnosis were legally blind. In the younger-onset group, 86% of blindness was due to diabetic retinopathy. In the older-onset group, where other eye diseases were also common, 33% of the cases of legal blindness were due to diabetic retinopathy.

Usually diabetic retinopathy has two stages non-PDR (NPDR) and PDR categories as shown in Table 3-5. Macular edema may coexist with either group and is not used in the classification of level of retinopathy. The historical terms background retinopathy and preproliferative diabetic retinopathy have been abandoned.

An accurate ocular examination detailing the extent and location of retinopathy-associated findings is critical for making monitoring and treatment decisions in patients with diabetic retinopathy.

Table 3-5: Progression to PDR by NPDR level [66].

	Chance (%) of high-risk PDR	
Retinopathy Level	1 Year	5 Years
Mild NPDR	1	16
Moderate NPDR	3–8	27–39
Severe NPDR	15	56
Very severe NPDR	45	71
PDR with < high-risk characteristics	22–46	64–75
NPDR: nonproliferative diabetic retinopathy PDR: proliferative diabetic retinopathy		

Below is the recommendations for retinopathy screening by ADA. Retinopathy screening recommendations:

- To reduce the risk or slow the progression of retinopathy, optimize glycemic control.
- To reduce the risk or slow the progression of retinopathy, optimize blood pressure control.
- Adults and children aged 10 years or older with type 1 diabetes should have an initial dilated and comprehensive eye examination by an ophthalmologist or optometrist within 5 years after the onset of diabetes.

- Patients with type 2 diabetes should have an initial dilated and comprehensive eye examination by an ophthalmologist or optometrist shortly after the diagnosis of diabetes.
- Subsequent examinations for type 1 and type 2 diabetic patients should be repeated annually by an ophthalmologist or optometrist. Less-frequent exams (every 2–3 years) may be considered following one or more normal eye exams. Examinations will be required more frequently if retinopathy is progressing.
- High-quality fundus photographs can detect most clinically significant diabetic retinopathy. Interpretation of the images should be performed by a trained eye care provider. While retinal photography may serve as a screening tool for retinopathy, it is not a substitute for a comprehensive eye exam, which should be performed at least initially and at intervals thereafter as recommended by an eye care professional.
- Women with pre-existing diabetes who are planning a pregnancy or who have become pregnant should have a comprehensive eye examination and be counseled on the risk of development and/or progression of diabetic retinopathy. Eye examination should occur in the first trimester with close follow-up throughout pregnancy and for 1 year postpartum.

3.4.2 Diabetic Nephropathy

The criteria of diabetic nephropathy is persistent proteinuria greater than 500 mg/24 hours in a person with diabetic retinopathy and without other renal disease.[20] The diabetic nephropathy is the chief cause of end-stage renal disease (ESRD) in many countries, and it is accounted for 44.5% of ESRD patients . Typically, diabetic ESRD patients have serious co-morbid conditions, especially heart, eye, and peripheral vascular diseases. It is quite normal that caring for afflicted individuals imposes a major financial burden on family and governments.

Diabetic nephropathy can be seen in both type 1 and type 2 diabetes patients. The distribution of renal disease due to type 2 diabetes is uneven in different racial groups. In some studies, American Indians, African Americans, and Mexican Americans have a greater incidence than non-Hispanic whites. Genetic predisposition, environmental factors, delayed diagnosis of type 2 diabetes, and subadequate medical care in minority groups contribute in undefined amounts to such disparity.

Since type 1 diabetes is usually possible to specify the exact time of onset, the natural history of diabetic nephropathy has been extensively studied in type 1 diabetes. First described by Mogensen, now it is common sense that there are five distinct stages of diabetic nephropathy [67].

- Glomerular hyperfiltration and renal enlargement.
- Early glomerular lesions or silent stage with normal albumin excretion.
- Incipient diabetic nephropathy or microalbuminuric stage.
- Clinical or overt diabetic nephropathy, proteinuria and falling glomerular filtration rate.
- End of stage renal disease.

As discussed above, the pathogenesis of diabetic nephropathy is a multistage process starting with a genetic predisposition injured by the elevated glucose concentration. Since a significant number of type 1 patients do not develop diabetic nephropathy, it seems that hyperglycemia is necessary but not sufficient for the development of diabetic nephropathy. Gene determined the differences in the renal response to hyperglycemia. In my clinical experience, I did find some interesting phenomena, for example, one patient in her 80’s with hyperglycemia for more than 20 years found no evidence of diabetic nephropathy. She is lucky, but there are other people who were not lucky as her and was found this complication a few years after diagnosis.

3.4.3 Diabetic Neuropathy

Diabetic neuropathy (DN) is also a common and troublesome complication of diabetes mellitus, which can lead to great morbidity and mortality and resulting in a huge economic burden for the patient and the society. It is a common form of neuropathy in the developed countries of the world and is responsible for 50% to 75% of nontraumatic amputations. Diabetic neuropathy is not a single complication of diabetes but a set of clinical syndromes that can affect distinct regions of the nervous system, singly or combined.

Classifications of diabetic neuropathy range from subclinical to clinical manifestations depending on the classes of nerve fibers involved. According to the San Antonio Convention, the main groups of neurologic disturbance in diabetes mellitus include the following:

- Subclinical neuropathy, determined by abnormalities in electrodiagnostic and quantitative sensory testing without concomitant clinical sign and symptoms.
- Diffuse clinical neuropathy, which may be proximal or distal and have large symmetrical sensorimotor or small-fiber and autonomic dysfunction.
- Focal neuropathies, which include mononeuropathies and entrapment syndromes.

Mononeuropathies occur primarily in the older population, their onset is generally acute and associated with pain, in patients with such pain can find that their course is self-limiting, resolving within 6 to 8 weeks. The reason is due to vascular obstruction after which adjacent neuronal fascicles take over the function of those infarcted by the clot.

Entrapment syndromes start slowly, progress, and persist without intervention and it must be distinguished from mononeuropathies. Entrapment syndromes are found in one third of patients with diabetes. If recognized, the diagnosis can be confirmed by electrophysiologic studies.

3.4.4 Diabetic Heart Disease

Coronary vascular disease (CVD) is the leading cause of mortality in patients with diabetes mellitus and it is estimated that approximately 75% of the cardiovascular deaths attributed to diabetes are directly related to coronary artery disease. The economic burden of CVD in patients with diabetes far exceeds other complications even including ESRD. Despite the well-recognized benefits of intensive blood glucose control in reducing the risk of microvascular complications of diabetes, as evidenced from the results of large randomized clinical trials, a similar result for macrovascular complications has not been firmly established.

As we know, more than 90% of all patients with diabetes have type 2 diabetes, and most of them are middle-aged or elderly. In these patients, the excess morbidity and mortality associated with diabetes and elevated glucose remained even after adjustment for traditional CHD risk factors.

3.4.5 The Diabetic Foot

The diabetic foot is a common complication of diabetes of which the clinical manifestation differ from mild to most serious even lead to death. Fortunately, the foot problems in patients with diabetes can be prevented effectively. The founder of famous Joslin diabetes center located in Boston, USA, Joslin, made a conclusion in 1934 that "diabetic gangrene is not heaven-sent, but earthborn," and he was right: The development of foot ulceration or gangrene mostly results from the way in which patients care for themselves.

Foot is very important for our daily life while few of us care about our feet. I have experienced an old patient with diabetic foot with a nail in his right foot for nearly 4 days without any sensation. His whole family thought it would be very easy to solve the problem. Unfortunately, after careful examination we found it hard to deal with because the blood vessels were destroyed severely. The end of the patient was a tragedy. For lack of money, the family of the old patients gave up the treatment.

In recent years, more and more attention has been paid by scientists and physicians in China. Many forums are held every year to discuss this kind of complication of diabetes mellitus. Increasing interest in the diabetic foot has resulted in a better understanding of the factors that interact to cause ulceration and amputation and the increase in the knowledge of pathogenesis will cause the design of appropriate screening programs for risk and preventive education.

Table 3-6 shows the clinical stages of diabetic foot.

Table 3-6: Wagner diabetic foot ulcer classification system [68].

Grade	Description
0	No ulcer, but high-risk foot (e.g., deformities, callus, insensitivity)
1	Superficial, full-thickness ulcer
2	Deeper ulcer, penetrating tendons, without bone involvement
3	Deeper ulcer with bone involvement, osteitis
4	Partial gangrene (e.g., toes, forefoot)
5	Gangrene of whole foot

3.4.6 Hypoglycemia

Although hyperglycemia is the target of our treatment in patients with diabetes, we can't ignore the problem of hypoglycemia. The manifestations of hypoglycemia are nonspecific, vary among individuals, and may change from time to time in the same individual. They are also typically episodic. Thus, although the history is of fundamental importance in suggesting the possibility of hypoglycemia, the diagnosis cannot be made solely on the basis of symptoms and signs.

The diagnosis of hypoglycemia should is based on the level of plasma glucose. Since symptoms commonly occur with plasma glucose levels less than 3.0 mmol/L (54 mg/dL) [69], it has been accepted by most physician as the criteria for diagnosis.

Table 3-7 shows the reasons that cause hypoglycemia and we usually focus on the affect of drugs.

Table 3-7: Clinical classification of hypoglycemia.

Postabsorptive (Fasting) Hypoglycemia
Drugs
Especially insulin, sulfonylureas, alcohol
Also pentamidine, quinine
Rarely, salicylates, sulfonamides
Others
Critical illnesses
Hepatic failure
Cardiac failure
Renal failure
Sepsis
Inanition
Hormonal deficiencies
Cortisol or growth hormone, or both
Glucagon and epinephrine
Non-beta cell tumors
Endogenous hyperinsulinism
Pancreatic beta cell disorders
Tumor (insulinoma)
Nontumor
Beta cell secretagogue (e.g., sulfonylureas)

Autoimmune hypoglycemia
Insulin antibodies
Insulin receptor antibodies
Beta cell antibodies
Ectopic insulin secretion
Hypoglycemias of infancy and childhood
Postprandial (Reactive) Hypoglycemia
Congenital deficiencies of enzymes of carbohydrate metabolism
Hereditary fructose intolerance
Galactosemia
Alimentary hypoglycemia
Idiopathic (functional) postprandial hypoglycemia

3.5 Management of Type 2 Diabetes

For many years, the guideline for treatment of type 2 diabetes has changed a lot according to many large randomized, multi-center studies and physician's clinical experience. Since more and more people are found diabetic, it becomes a serious problem for both the patients and the whole society. More pharmacologic agents and monitoring technology are available now than that of before. The treatment of diabetes has made it possible to lower glucose safely to the near-normal range in the majority of patients and lower the risk of complications. Both corporate and government health insurance providers have greatly improved the extent to which diabetes equipment and supplies are covered.

An important and excellent source of my information to make right decisions on these issues that is updated annually is the American Diabetes Association's Clinical Practice

Recommendations. It is usually published as the first supplement to the journal Diabetes Care each January and is available on line.

Before we travel by bus, train or plane, the first thing we should make it clear is the destination. So the first thing for treatment of diabetes is to know the goals for blood glucose level, HbA1c, blood pressure, etc.

To find the goals of treatment of diabetes , many prospective randomized clinical trials have been done. UK Prospective Diabetes Study (UKPDS) is one of the most famous international study which indicate improved rates of microvascular complications in patients with type 2 diabetes treated to lower glycemic targets. In this study, patients with new onset diabetes were treated with diet and exercise for 3 months with an average reduction in HbA_{1c} from approximately 9% to 7% (upper limit of normal 6%). Those patients whose FPG are greater than 108 mg/dL (6 mmol/L) were randomly assigned to two treatment policies. In the standard intervention, patients began the lifestyle intervention at first. Only if the FPG reached 15 mmol/L (270 mg/dL) or the patient became seriously symptomatic, pharmacologic therapy started. In another group which is called intensive treatment program, all patients were randomly assigned and treated with either sulfonylurea, metformin, or insulin as initial therapy, with the dose increased to try to achieve an FPG less than 108 mg/dL. Combinations of medicines were used only if the patients became seriously symptomatic or FPG was greater than 270 mg/dL (15 mmol/L). Unfortunately, although the HbA_{1c} fell initially to about 6%, over the average 10 years of follow-up it rose to approximately 8%.

Compared with intensive treatment group, the average HbA_{1c} in the standard treatment group was approximately 1% higher. On the other hand, the risk of severe hypoglycemia was small—on the order of 1% to 5% per year in the insulin-treated group—and weight gain was modest. Although the level of HbA_{1c} is much lower in intensive treatment group , the risk of

severe hypoglycemia were higher in patients randomly assigned to insulin and and weight gain were lower in those receiving metformin. Associated with this improvement in glycemic control, there was a reduction in the risk of microvascular complications (retinopathy, nephropathy, and neuropathy) in the intensive group. Although there was a trend toward reduced rates of macrovascular events in the more intensively treated group, it did not reach statistical significance.

This study greatly influenced the view of physicians and the way to treat the disease around the world for many years and intensive became more and more popular. However, there are many other large prospective, randomized and multi-center study in recent years which against the opinion and strengthen the point of view that risk of severe hypoglycemia is more dangerous and it can greatly reduce the improvement of microvascular and macrovascular complications.

3.5.1 Glycemic Goals in Adults

Lowering A1C to below or around 7% has been shown to reduce microvascular and neuropathic complications of diabetes and, if implemented soon after the diagnosis of diabetes, is associated with long-term reduction in macrovascular disease. Therefore, a reasonable A1C goal for many nonpregnant adults is <7%.

Till today, how to control HbA_{1c} is still a big problem in both China and USA. The average HbA_{1c} in the United States is estimated to be in the 7.5% to 9.5% range, so the argument about whether the HbA_{1c} target should be 6.5% or 7% is of limited practical significance.

Because additional analyses from several randomized trials suggest a small but incremental benefit in microvascular outcomes with A1C values closer to normal, providers might reasonably suggest more stringent A1C goals for selected individual patients, if this can be achieved without significant hypoglycemia or other adverse effects of treatment. Such

patients might include those with short duration of diabetes, long life expectancy, and no significant cardiovascular disease.

Conversely, less stringent A1C goals may be appropriate for patients with a history of severe hypoglycemia, limited life expectancy, advanced microvascular or macrovascular complications, extensive comorbid conditions, and those with longstanding diabetes in whom the general goal is difficult to attain despite diabetes self-management education, appropriate glucose monitoring, and effective doses of multiple glucose lowering agents including insulin.

3.5.2 Lifestyle Intervention

Lifestyle intervention is critical for diseases like diabetes mellitus since they cannot be cured by the physicians and will last for a life time. The components of lifestyle intervention include medical nutrition counseling, exercise recommendations, and comprehensive diabetes education with the purpose of changing the paradigm of care in diabetes from provider-focused to patient-focused. In china, people used to medicines and are always bother by pay more attention by themselves. After more than 10 years of continuous education, this situation is gradually changed and the patients make it clear that western lifestyle do harm to their health and correct intervention is important. So many significant clinical trials convinced that each component of lifestyle intervention, when appropriately administered, can contribute to improved outcomes. Unfortunately, lifestyle intervention hasn’t been a covered benefit for most people in china. Although full implementation of these regulations is still in progress, we need a dramatically expansion on the proportion of the population with diabetes with insurance coverage for these essential services.

3.5.2.1 Diabetes Self-Management Education (DSME)

People with diabetes should receive DSME according to national standards when their diabetes is diagnosed and as needed thereafter. Effective self-management and quality of life

are the key outcomes of DSME and should be measured and monitored as part of care. DSME should address psychosocial issues, since emotional well-being is associated with positive diabetes outcomes. Because DSME can result in costsavings and improved outcomes , DSME should be adequately reimbursed by third-party payors. The Standards of Medical Care in Diabetes [56] have the following recommendations:

- Medical nutrition therapy (MNT) General recommendations:
  - Individuals who have pre-diabetes or diabetes should receive individualized MNT as needed to achieve treatment goals, preferably provided by a registered dietitian familiar with the components of diabetes MNT.
  - Because MNT can result in cost-savings and improved outcomes , MNT should be adequately covered by insurance and other payors.
- Energy balance, overweight, and obesity recommendations:
  - In overweight and obese insulinresistant individuals, modest weight loss has been shown to reduce insulin resistance. Thus, weight loss is recommended for all overweight or obese individuals who have or are at risk for diabetes.
  - For weight loss, either low-carbohydrate, low-fat calorie-restricted, or Mediterranean diets may be effective in the short term (up to 2 years).
  - For patients on low-carbohydrate diets, monitor lipid profiles, renal function, and protein intake (in those with nephropathy) and adjust hypoglycemic therapy as needed.
  - Physical activity and behavior modification are important components of weight loss programs and are most helpful in maintenance of weight loss.
- Recommendations for primary prevention of diabetes:
  - Among individuals at high risk for developing type 2 diabetes, structured programs that emphasize lifestyle changes that include moderate weight loss (7% of body weight) and regular physical activity (150 min/week), with dietary strategies

including reduced calories and reduced intake of dietary fat, can reduce the risk for developing diabetes and are therefore recommended.

- Individuals at high risk for type 2 diabetes should be encouraged to achieve the U.S. Department of Agriculture (USDA) recommendation for dietary fiber (14 g fiber/1,000 kcal) and foods containing whole grains (one-half of grain intake).

- Recommendations for management of diabetes: macronutrients in diabetes management:
  - The best mix of carbohydrate, protein, and fat may be adjusted to meet the metabolic goals and individual preferences of the person with diabetes.
  - Monitoring carbohydrate, whether by carbohydrate counting, choices, or experience-based estimation, remains a key strategy in achieving glycemic control.
  - For individuals with diabetes, the use of the glycemic index and glycemic load may provide a modest additional benefit for glycemic control over that observed when total carbohydrate is considered alone.
  - Saturated fat intake should be _7% of total calories.
  - Reducing intake of trans fat lowers LDL cholesterol and increases HDL cholesterol ; therefore, intake of trans fat should be minimized.
- Other nutrition recommendations:
  - If adults with diabetes choose to use alcohol, daily intake should be limited to a moderate amount (one drink perday or less for adult women and two drinks per day or less for adult men).
  - Routine supplementation with antioxidants, such as vitamins E and C and carotene, is not advised because of lack of evidence of efficacy and concern related to long-term safety.

- Individualized meal planning should include optimization of food choices to meet recommended daily allowance (RDA)/dietary reference intake (DRI) for all micronutrients.

- Physical activity:
  - People with diabetes should be advised to perform at least 150 min/week of moderate-intensity aerobic physical activity (50–70% of maximum heart rate).
  - In the absence of contraindications, people with type 2 diabetes should be encouraged to perform resistance training three times per week.
- Psychosocial assessment and care:
  - Assessment of psychological and social situation should be included as an ongoing part of the medical management of diabetes.
  - Psychosocial screening and follow-up should include, but is not limited to, attitudes about the illness, expectations for medical management and outcomes, affect/mood, general and diabetes-related quality of life, resources (financial, social, and emotional), and psychiatric history.
  - Screen for psychosocial problems such as depression and diabetes-related distress, anxiety, eating disorders, and cognitive impairment when self-management is poor.
- Retinopathy treatment General recommendations:
  - Promptly refer patients with any level of macular edema, severe nonproliferative diabetic retinopathy (NPDR), or any proliferative diabetic retinopathy (PDR) to an ophthalmologist who is knowledgeable and experienced in the management and treatment of diabetic retinopathy.
  - Laser photocoagulation therapy is indicated to reduce the risk of vision loss in patients with high-risk PDR, clinically significant macular edema, and some cases of severe NPDR.

- The presence of retinopathy is not a contraindication to aspirin therapy for cardioprotection, as this therapy does not increase the risk of retinal hemorrhage.

3.5.3 Pharmacotherapy of Type 2 Diabetes

Scientists are making endless effort to deal with the big problem of modern society – diabetes mellitus. The available oral anti-diabetic agents can be divided by mechanism of action into insulin sensitizers with primary action in the liver, insulin sensitizers with primary action in peripheral tissues, insulin secretagogues, and agents that slow the absorption of carbohydrates. Insulin therapy in patients with type 2 diabetes are uncommon for the past decades while it has been more popular than before.

Here are some drugs that are mostly used in many hospitals in China:

- Insulin Secretagogues ：Sulfonyluria & Repaglinide
- Insulin Sensitizers with Predominant Action in the Liver: Biguanides
- Insulin Sensitizers with Predominant Action in Peripheral Insulin-Sensitive Tissues: Thiazolidinediones
- Carbohydrate Absorption Inhibitors: α-Glucosidase Inhibitors

3.5.4 Treatment of Diabetic Nephropathy

Animal studies show that a low-protein diet can reduce glomerular hypertension and prevents glomerular injury and albuminuria. In type 1 diabetic patients with microalbuminuria and glomerular hyperfiltration, short-term dietary protein restriction (0.6 to 0.8 g/kg per day) decreases urinary AER and hyperfiltration.

Clinical experience of the past several decades convincingly demonstrates that intensive control of hyperglycemia and adequate lowering of hypertension are the key components of diabetic nephropathy management. Many perspective, multi-center, randomized clinical trials demonstrated that better control of both hyperglycemia and hypertension can modify the

natural course of diabetic nephropathy by reversing functional changes and by stabilizing progression of structural abnormalities.

As a worldwide well known trial, DCCT, which strived for near-normoglycemia in patients with type 1 diabetes, showed a 39% reduction in the risk of developing microalbuminuria and a 54% reduction in the occurrence of albuminuria. Another famous study, UKPDS, comparing intensive blood glucose with conventional therapy in type 2 diabetes, found a 25% risk reduction (7% to 40%; P = .0099) in microvascular complications, including progressive nephropathy. Such data was emphasized for too many times in all kinds of books, articles , conferences etc. and thus intensive control was the most popular treatment for the past decade.

With regard to neuropathy screening and treatment, the Standards of Medical Care in Diabetes [56] have the following recommendations:

- All patients should be screened for distal symmetric polyneuropathy (DPN) at diagnosis and at least annually thereafter, using simple clinical tests.
- Electrophysiological testing is rarely needed, except in situations where the clinical features are atypical.
- Screening for signs and symptoms of cardiovascular autonomic neuropathy should be instituted at diagnosis of type 2 diabetes and 5 years after the diagnosis of type 1 diabetes. Special testing is rarely needed and may not affect management or outcomes.
- Medications for the relief of specific symptoms related to DPN and autonomic neuropathy are recommended, as they improve the quality of life of the patient.

3.5.5 Management of Diabetic Foot Ulcers

We can apply basic principles of wound healing to diabetic foot ulcers and a diabetic foot ulcer will heal if the following three conditions are satisfied:

- Arterial inflow is adequate.

- Infection is treated appropriately.
- Pressure is removed from the wound and the immediate surrounding area.

Although this approach may seem simplistic, failure of healing of diabetic foot ulcers is normally due to neglect of one or more of the considerations above.

With regard to foot care, the Standards of Medical Care in Diabetes [56] have the following recommendations:

- For all patients with diabetes, perform an annual comprehensive foot examination to identify risk factors predictive of ulcers and amputations. The foot examination should include inspection, assessment of foot pulses, and testing for loss of protective sensation (10-g monofilament plus testing any one of: vibration using 128-Hz tuning fork, pinprick sensation, ankle reflexes, or vibration perception threshold).
- Provide general foot self-care education to all patients with diabetes.
- A multidisciplinary approach is recommended for individuals with foot ulcers and high-risk feet, especially those with a history of prior ulcer or amputation.
- Refer patients who smoke, have loss of protective sensation and structural abnormalities, or have history of prior lower-extremity complications to foot care specialists for ongoing preventive care and life-long surveillance.
- Initial screening for peripheral arterial disease (PAD) should include a history for claudication and an assessment of the pedal pulses. Consider obtaining an ankle-brachial index (ABI), as many patients with PAD are asymptomatic.
- Refer patients with significant claudication or a positive ABI for further vascular assessment and consider exercise, medications, and surgical options.

3.5.6 Treatment of Hypoglycemia in Diabetes

In patients with asymptomatic hypoglycemia (detected by SMBG) and mild to moderate symptomatic hypoglycemia , it can be effectively self-treated by ingestion of glucose tablets or carbohydrate such as juices, soft drinks, milk, candy, or a meal. It is recommended that the dose of glucose is about 20 g (0.3 g/kg in children). In patients with severe hypoglycemia who are unable to take carbohydrate orally(sometimes it can be very dangerous), glucose solution or glucagon is commonly injected subcutaneously or intramuscularly by a nurse or family member [70]. Because the glycemic response is transient, a subsequent glucose infusion is often needed and food should be provided orally as soon as the patient is able to take it safely. Then the reason that cause the state of hypoglycemia should be looked for and avoided in the future.

3.7 Summary

This chapter reviewed diabetes mellitus from a clinical point of view. It introduced the types of diabetes with a focus on type 2 diabetes. It analyzed the diagnosis of type 2 diabetes mellitus, the treatments and management of type 2 diabetes. It also discussed the complications of diabetes.

The present-day management of type 2 diabetes is significantly more effective and easier for patients than what it was decades ago. A better understanding of the barriers to effective diabetes management and how to overcome them would be of great benefit. Novel pharmaceutical agents including glucagon receptor antagonists, inhibitors of gluconeogenic and glycogenolytic pathways, activators of the insulin signaling pathways, modifiers of lipid metabolism, and antiobesity agents can light the way of both the patients and physicians.

Chapter 4:

MRNN - A MapReduce based Parallel Neural Network

This chapter presents the design and implementation of a parallel neural network called MRNN building on the MapReduce distributed computing paradigm. It starts this chapter with a brief introduction to the MapReduce framework.

4.1 MapReduce

MapReduce is a programming model that helps with processing large data sets in a parallel fashion [12]. The MapReduce model is inspired by functional programming languages. The input and output data have a specific format of key/value pairs. The users express an algorithm using two functions: the Map functions and the Reduce function. The Map function is written by the application developer. It iterates over a set of the input key/value pairs, and generates intermediate output key/value pairs. The MapReduce library groups all intermediate values by key and introduces them to the reduce function. The Reduce function is also written by the application developer, it iterates over the intermediate values associated by one key. Then it generates zero or more output key/value pairs. The output pairs are sorted by their key value.

(input) <k1, v1> -> map -> <k2, v2> -> reduce -> <k3, v3> (output)

MapReduce allows application developers to use data-centric techniques and permits the implementation of the MapReduce structure for parallel processing networks. It is usually applied to the completion of major concerns involving enormous amounts of computing time. Both the Google MapReduce framework and the open source Hadoop framework have taken over implementation strategies to support batching by the model: every map and reduce output stage had to be materialized before there can be stable storage. This batch unit allows the checkpoint restart fault tolerance that is critical for large distributions.

When using MapReduce technology the application designer expresses their desired computation as a number of jobs. Each job consists of two steps: first, the map function produces a list of intermediate key-value pairs from each input record. The output of all of the maps will be separated and each separation will be sorted. There will be one map output for each reduce task. Every partition has keys and values, then a reduce method combines all the results to key/value pairs. Second, the reduce function passes on the list of intermediate values associated with each distinct key in the map output. The MapReduce program automatically distributes the implementation of these functions and ensures fault tolerance. Figure 4.1 shows the split of the input into logical chunks and each chunk is processed independently by a map task. The results of these processing chunks can be physically partitioned into separate sets, which are then sorted. Each sorted chunk is passed to a reduce task.

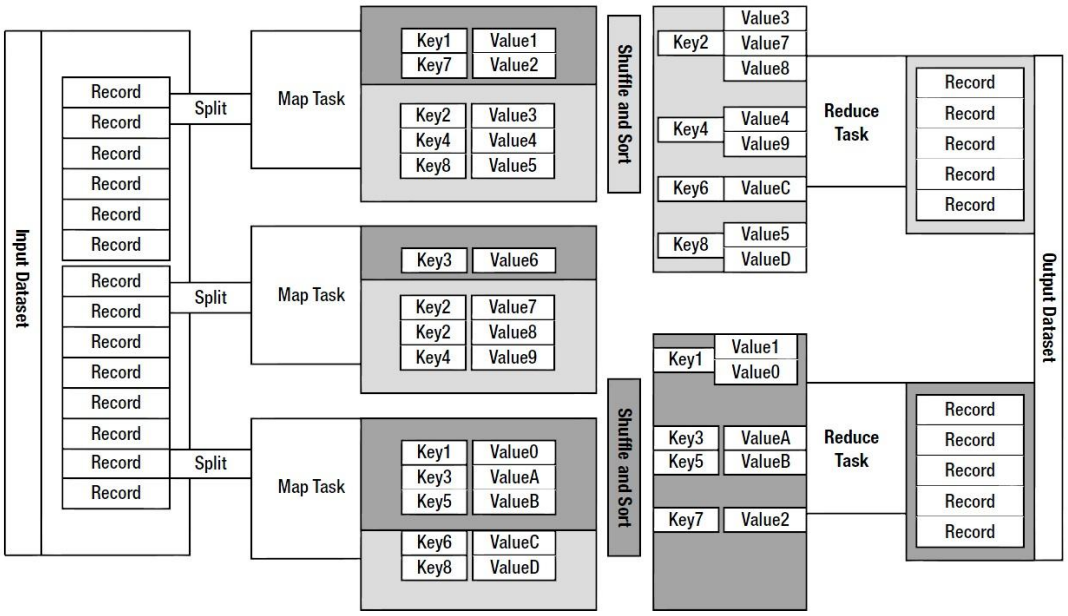

Figure 4.1: The MapReduce model.

While the programming model is abstracted, it is the job of the implementation to deal with the details of parallelization, fault tolerance, data distribution, load balancing. Several

implementations of MapReduce have been proposed some of them provided by academic research including Mars [71], Phoenix [72], and Google's implementation [73]. Among them, Hadoop [74] has become the most popular one due to its open source feature. The Apache Hadoop project [1] is the most popular and widely used open-source implementation of Google's MapReduce. It is written in Java for reliable, scalable, distributed computing.

4.2 MapReduce Hadoop

Hadoop is the most popular implementation of MapReduce due to its open source nature. This section briefly describes the major components of Hadoop.

4.2.1 JobTracker and TaskTracker

Hadoop consists of a single master JobTracker and one slave TaskTracker per cluster-node. JobTracker is responsible for scheduling tasks on the TaskTrackers, monitoring them and re-executing the failed tasks. The TaskTrackers execute the tasks as directed by the JobTracker. Files are shared on the system using Hadoop Distributed File System (HDFS). Figure 4.2 shows the job submission and execution process in Hadoop.

Job Submission: Client asks the JobTracker for a new job ID, checks the specifications of the job, calculates input splits for the job, and copies the resources needed to run the job including the job configuration file.

Job Initialization: JobTracker puts the submitted job into an internal waiting queue from where the job scheduler will pick it up and initialize it. Initialization involves creating an object to represent the job being run, retrieve the input splits computed by the Client. Then it creates one map task for each split. The number of reduce tasks to create is determined by job specifications. All map and reduce tasks are given IDs for tracking.

[1] http://hadoop.apache.org

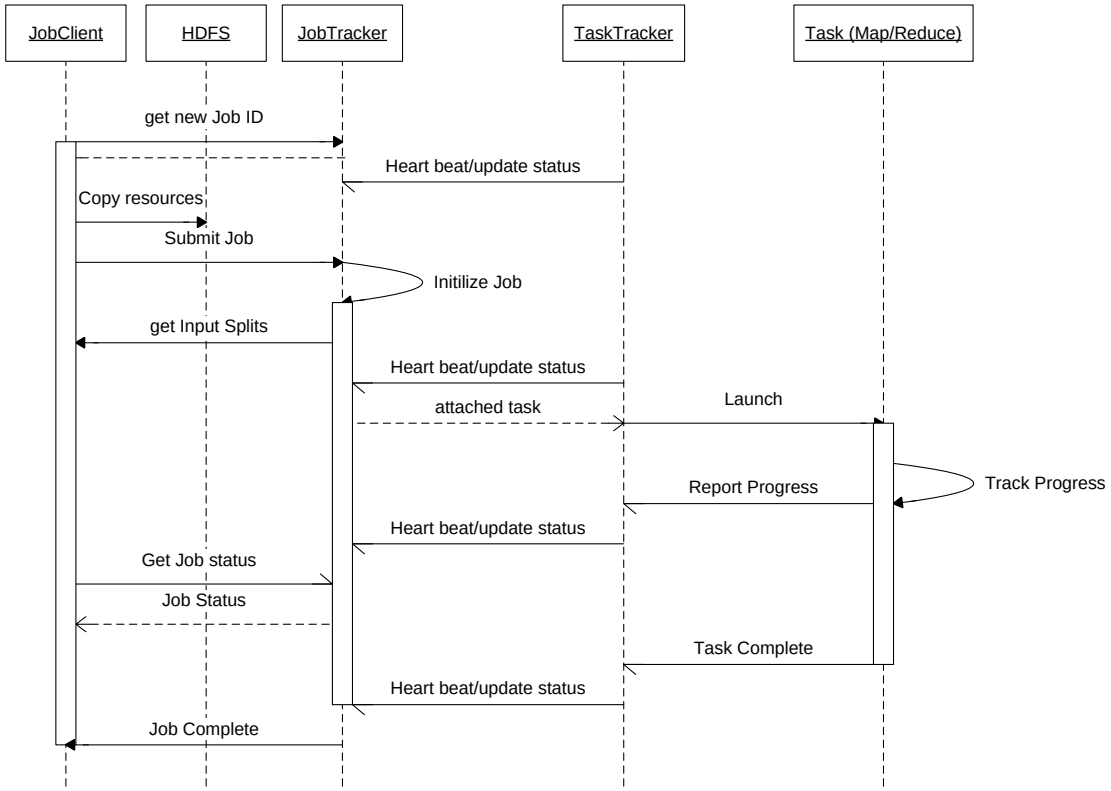

Figure 4.2: Job submission and execution in Hadoop.

Task Assignment: A simple loop running periodically every "heartbreat" updates the JobTracker with the TaskTrackers' status. Furthermore, in every loop run, each TaskTracker will check if it is ready to run new tasks. Then the JobTraker will allocate the new tasks by using its assigned scheduler. TaskTrackers have fixed number of task slots for map tasks and for reduce tasks. This is defined in cluster configuration file. To achieve better performance, assigning map task needs more scheduling work to ensure data locality so the TaskTracker will be as close as possible to map input split. Assigning reduce task is simpler. The JobTracker simply takes the next waiting task in the queue and run it on the current available TaskTracker slot.

Task Execution: Now the TaskTracker has been assigned a task, it creates a local working directory for the task. TaskTracker will launch a new Java Virtual Machine to run each task.

Progress and status update: A job and each of its tasks have a status, which includes state of the job or task, the progress of maps and reduces, and the values of the job's or task's counters. The progress of map tasks is the proportion of the input that has been processed. The progress of reduce tasks, is divided to three phases: shuffle, sort and reduce. Tasks also have a set of counters that count various events as the task runs. The JobTracker combines these updates to produce a global view of the status of all the jobs being run and their tasks. Finally, the JobClient receives the latest status by polling the JobTracker.

Job Completion: When the last task for a job is completed, the JobTracker will changes the status for the job to indicate that is successful. When the JobClient polls for status, it learns that the job has completed successfully. In Hadoop, clients can be configured to receive callbacks by providing URL of returned call at the "job.end.notification.url" property. In MRSim, callback is implemented by providing the ID of JobClient, which is of type SimJava entity id. Finally, In Hadoop and MRSim, the JobTracker cleans up its working state for the job, remove it from running queues, and keep log history of the job its tasks on the file system.

4.2.2 TaskTracker

Each machine at the cluster has at most one TaskTracker component. TaskTracker run tasks assigned by JobTracker master node and send progress reports back to it. In MRSim, TaskTracker has access to the machine resources such as CPU, hard disk and network. All map/reduce tasks running in certain machine will share machine resources through the TaskTracker interface running on that machine.

4.2.3 The Map Model

If data inputs are not divided by the user Hadoop divides them into fixed-size pieces called "splits". Hadoop creates a map task for each split. Data locality is optimization in Hadoop by

trying to run the map task on a node where the input split resides in HDFS. However, some splits would have to be transferred across the network to the node running the map task. Map tasks generate intermediate output and write it to local hard disk and not HDFS.

When the map function starts producing output, the output is not simply written to disk. First it is buffered in memory buffer. Each map task has a circular memory buffer that it writes the output to. The buffer size is defined by the "ioSortMb" parameter in the job description. When the content of the buffer reaches a certain threshold size (also defined by the job description), it spills the contents to disk. Before it writes to disk, the map task first divides the data spill into partitions equal to the number of reducers. Within each partition, in memory the sort operation is performed, and if there is a combiner function, it is applied to the output of the sort. After the map task has written its last output spill, there could be several spill files. The spill files are merged into a single partitioned and sorted output file. Combiner is applied again on the resulting file if it is defined in the job description. The configuration parameter "ioSortFactor" controls the maximum number of spills to be merged simultaneously. Compression of output data could be enabled by the job configuration. If enabled, the merged spill will be compressed before it is written to the hard disk. This usually increases the performance of map tasks and reduces the task by shrinking the data sizes to be written to hard disk and to be transferred over the network. The output data are made available to the reducers over the network. Figure 4.3 summarizes the flow control of the map task.

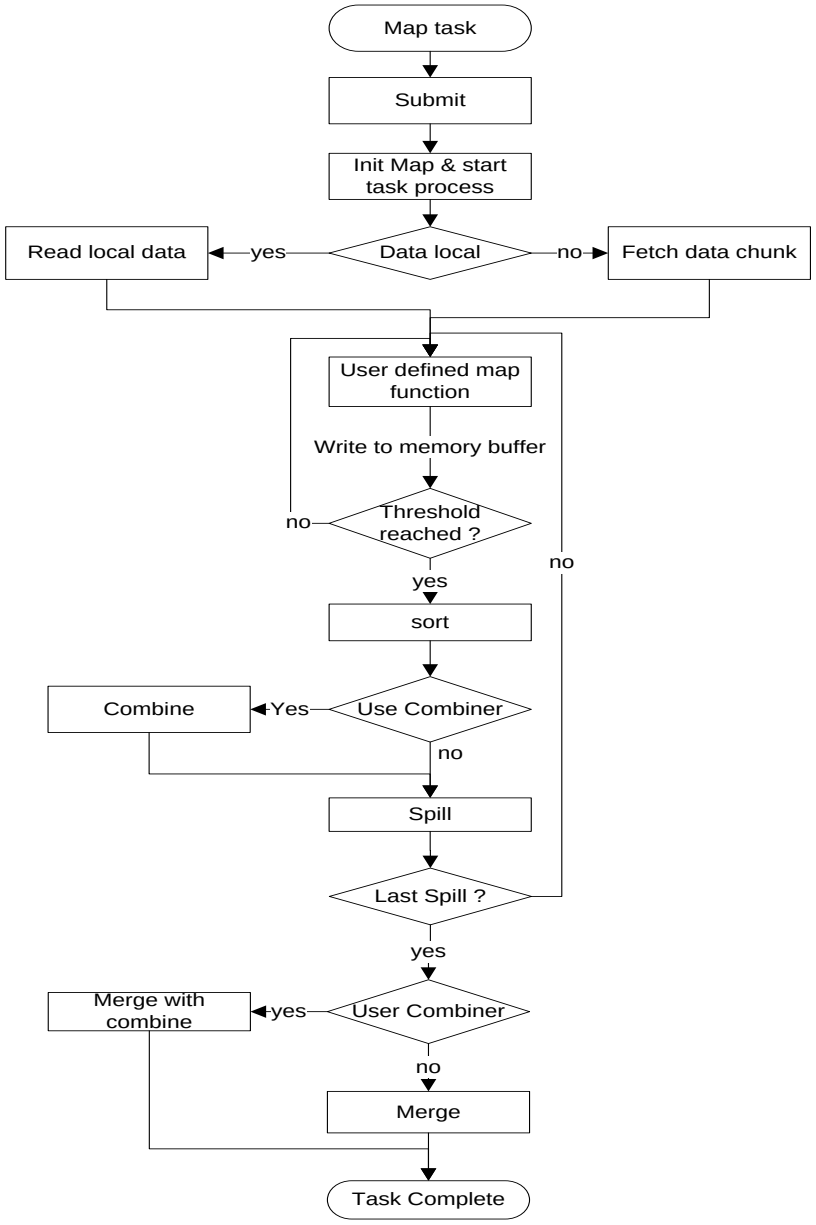

Figure 4.3: The map model.

4.2.4 The Reduce Model

Each reducer task normally gets its data input from the output of all mappers. Thus usually there is no data locality in reduce tasks. The sorted map outputs are transferred across the network to the node where the reduce task is running. Then, input partitions are merged and passed to the user-defined reduce function. The output of reduce tasks is normally stored in a distributed file system. If there are multiple reducers, the map tasks divide their output, each creating one partition for each reduce task. The data flow between map and reduce tasks is

known as "shuffle" as each reduce task is fed by many map tasks. The shuffle is an important phase where optimizing can have a large effect on job execution time. If there are zero reduce tasks, then map tasks write output data directly to HDFS.

The shuffle phase is more complicated than described above. And it is important to model it in more detail to get a more accurate prediction of a Hadoop MapReduce cluster. The map tasks may finish at different times, so the reduce task starts “shuffling” their output partitions as soon as each map completes. This is also known as the copy phase of the reduce task. The reduce task has a small number of copier processes that fetch map outputs in parallel. This number is defined by the "mapredReduceParallelCopies" job description property. The map outputs are copied to the reduce buffer memory if they are small enough. Also the buffer’s size is defined by the “mapredJobShuffleInputBufferPercent" property, which specifies the proportion of the memory heap to use for this purpose. If the map outputs are not very small, they are copied to disk. When the in-memory buffer reaches a threshold size (also defined in "mapredJobShuffleMergePercent"), or reaches a threshold number of map outputs (defined by "mapredInmemMergeThreshold"), it is merged and spilled to disk. There is also a background process that merges the spills into larger files.

When all the map outputs have been copied, the reduce task moves into the “sort phase”. In the sort phase, the merging process keeps merging maps’ output to larger ones and keeps the data sorted. The maximum files that can by merged at once are defined by the merge factor (ioSortFactor property).The merging process runs rounds of merges till it completes merging whole map outputs fetched to the reducer. The final merge can come from a mixture of data in-memory and data on-disk. In the last round that merges the resulting files, the merger directly feeds the reducer with the data. The reducer at this stage is in the "reduce phase”, where the user-defined reduce function is called for each key in the sorted output. The output

of this phase is written directly to the output HDFS. Figure 4.4 summarizes the flow control of the reduce model.

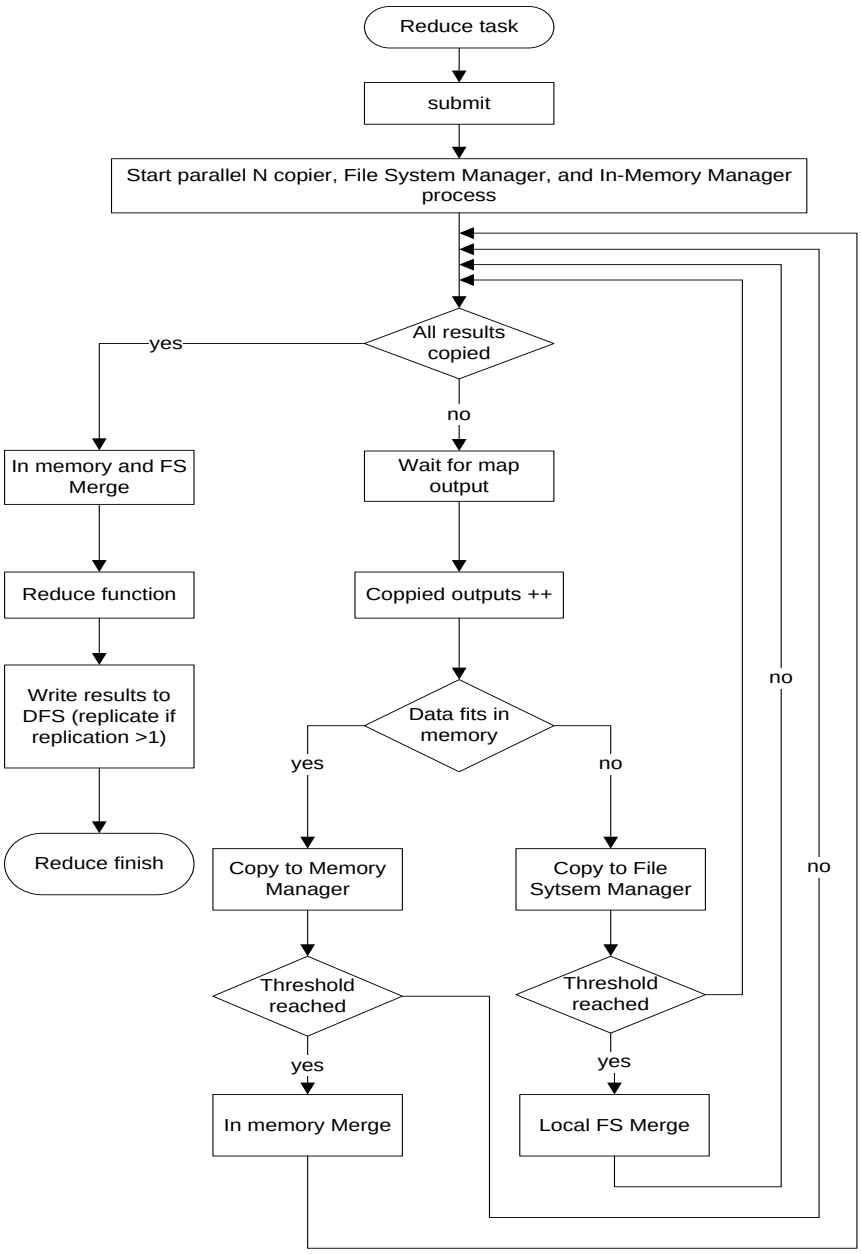

Figure 4.4: The reduce model.

4.2.5 The Combiner Model

The combiner function is run on the map output data buffered in memory, and sorted by the keys. Combiners may be run repeatedly over the input because there could be one or more data spills generated by the map task. Combiners do not affect the final result. Running

combiners makes for a more compact map output, so there is less data to write to local disk and to transfer to the Reducers. Usually the combiner uses the same or similar code to the Reducer code because combiners can be used when the reduce function is mathematically aggregated.

4.3 Design of MRNN

This section describes the design of MRNN, a parallel neural network based on the MapReduce programming model.

MRNN builds on a multi-layered back propagation neural network. The data parallelism strategy is adopted in MRNN as it offers a lower communication overhead than the approaches of session parallelism, neuron parallelism and weigh parallelism. partitions the entire training data set into equally sized smaller data chunks and assigns each data chunk to a single *map* task. The number of *map* tasks is equal to the number data chunks. Each *map* function optimizes a data chunk in parallel at each layer. All the mappers train the entire network using different data chunks. The outputs of the mappers are summered by a reducer to calculate the change on each weight. In this way, the weights are updated and used as a new set of input for training in the next iteration. Figure 4.5 shows the architecture of MRNN.

Let

- be the input to the neuron,
- be the output of the neuron,
- be the activity function of the neuron,
- be the error of the neural network over the training pattern ,
- be the error of the network over the entire training data set,
- be the number of neurons at the output layer for pattern ,

- be the desired output of the neuron,
- be the weight for the input of the neuron,
- be the previous weight for the input of the neuron,
- be the next to previous weight for the input of the neuron,
- be the learning rate,
- be the momentum factor,
- be the error term of the neuron.
- be the number of patterns.

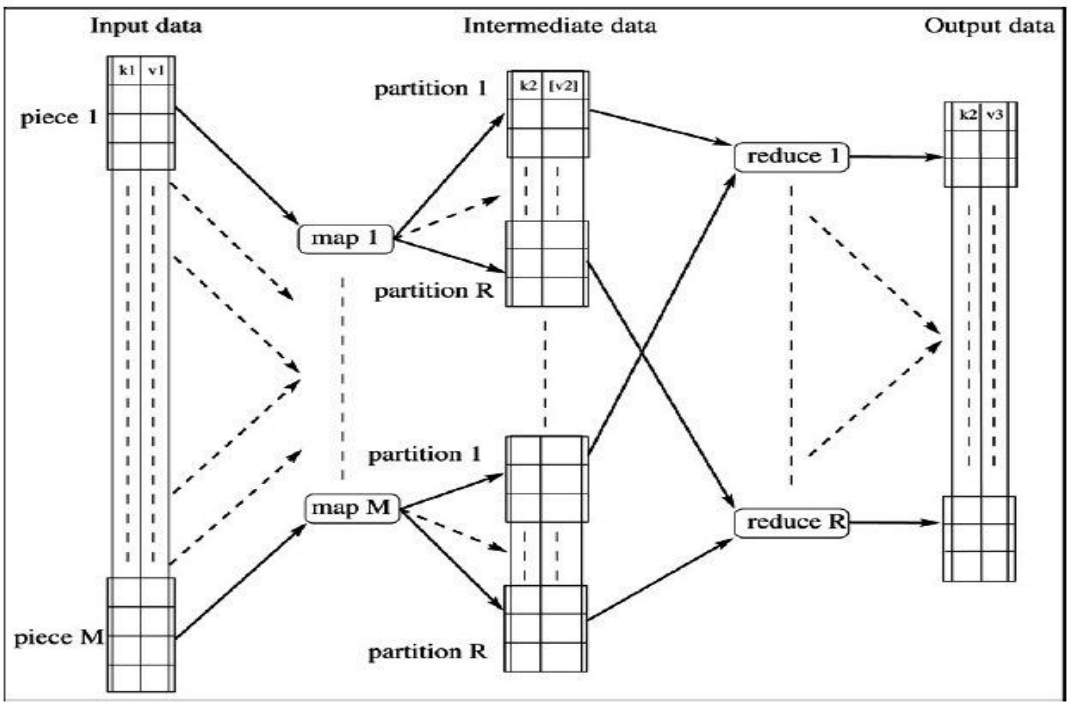

Figure 4.5: MRNN architecture.

Then the output of the neuron can be calculated with equation (4-1).

(4-1)

The activity function of a neuron is normally a sigmoid function varying from 0 to 1, as expressed in equation (4-2).

(4-2)

For the input of the neuron, the weight is adjusted by adding to the previous weight value together with a learning rate , an error term and the input of the neuron which is denoted in equation (4-3).

(4-3)

Here the error term for the neuron can be calculated by equation (4-4).

(4-4)

Once the error terms of the output layer are computed and the weights are adjusted, the network records the values and propagates these values to the back layer for updates. The same weight adjustment process which is determined by equation (4-3) is followed.

A revised weight adjust process can be computed using equation (4-5) which is a slightly modified version of the equation (4-3).

(4-5)

The momentum factor is used in equation (4-5) which allows a change to the weights to be useful in a number of adjustment cycles. The use of can improve the learning in some situations by helping to smooth out unusual conditions in the training set.

Then the error of the network over the pattern having neurons at the output layer can be calculated with equation (4-6).

(4-6)

Then the total error of the network over the entire training set can be computed using equation (4-7).

—— (4-7)

The mapper function and reducer function can be summarized as follows:

Procedure Mapper()

```
1    Input <key, value=weights>
2    For all the training instances
3        computes a local
4    End-For
4    Updates all the weights of the local neural network
5    Output <key, value=  > to Reducer
6    End-Procedure
```

Procedure Reducer()

```
1    Inputs <key, value=  >
2    Calculates  by summing up the values of the set of
3    Updates the weights of the network
4    If E satisfies expectation,
5        Stops the training process
6    End-If
7    Else
8        Outputs <key, value=updated weights> to  Mapper
9    End-Else
10   End-Procedure
```

4.4 Summary

This chapter introduced MRNN, a parallel neural network building on the MapReduce framework. MRNN partitions the training dataset into equally sized data blocks. MRNN performs well in homogeneous computing environments in which all the computing nodes have equal computing capabilities. The updates of the computed weights at each iteration can be done efficiently as all the computing nodes would complete the local training process at almost the same time. However, modern computing infrastructures are normally heterogeneous computing environments in which computing nodes have varied computing capabilities. When training a large data set using MRNN, the computing nodes with less resources will slow down the whole training process as fast computing nodes have to wait for exchange of computed weights of the neurons. Workload needs to be well balanced among the computing nodes so that at each iteration each local training can be finished at almost the same time.

Chapter 5:

Facilitating Resource Sensitivity of MRNN with Load Balancing

As mentioned in Chapter 4, MRNN only performs well in homogeneous computing environments in which all the computing nodes have equal computing capabilities. To enhance the performance of MRNN in heterogeneous computing environments, a genetic algorithm based load balancing scheme is designed to facilitate resource sensitivity of MRNN. This chapter presents the design of the load balancing scheme. It starts with a brief overview on genetic algorithms.

5.1 Genetic Algorithms

Genetic Algorithms (GAs) are a class of Evolutionary Algorithms (based on the Darwinian principles of natural selection and survival of the fittest) that originated from the work conducted by Holland [75]. GAs are the most popular nature's heuristic algorithm used for optimisation problems as they offer a robust search technique for large spaces in polynomial time using a structured and proven implementation. GAs are also widely employed for optimising schedules in grid computing environments for independent and dependent jobs. GA implementation is composed of:

- Problem representation (genetic representation and fitness function)
- GA Operators (selection, crossover and mutation)
- Termination function

Genetic representation translates a candidate solution into a fixed size array of bits. In biological terms, this is also known as a chromosome. Figure 5.1 shows two commonly used GA representations for scheduling optimization. Figure 5.1a refers to the genetic

representation proposed by Wang et al [76] and Figure 5.1b describes the two-dimensional coding scheme proposed by Hou, Ansari & Ren [77]. The separate matching and scheduling string representation (Figure 5.1a) is preferred as it simplifies coding. In this scheme, genetic representation is composed of:

- The scheduling part: defines the order of job execution. In DAG terms, this refers to a topologically sorted list.
- The matching part: defines the computing node on which each job will be executed.

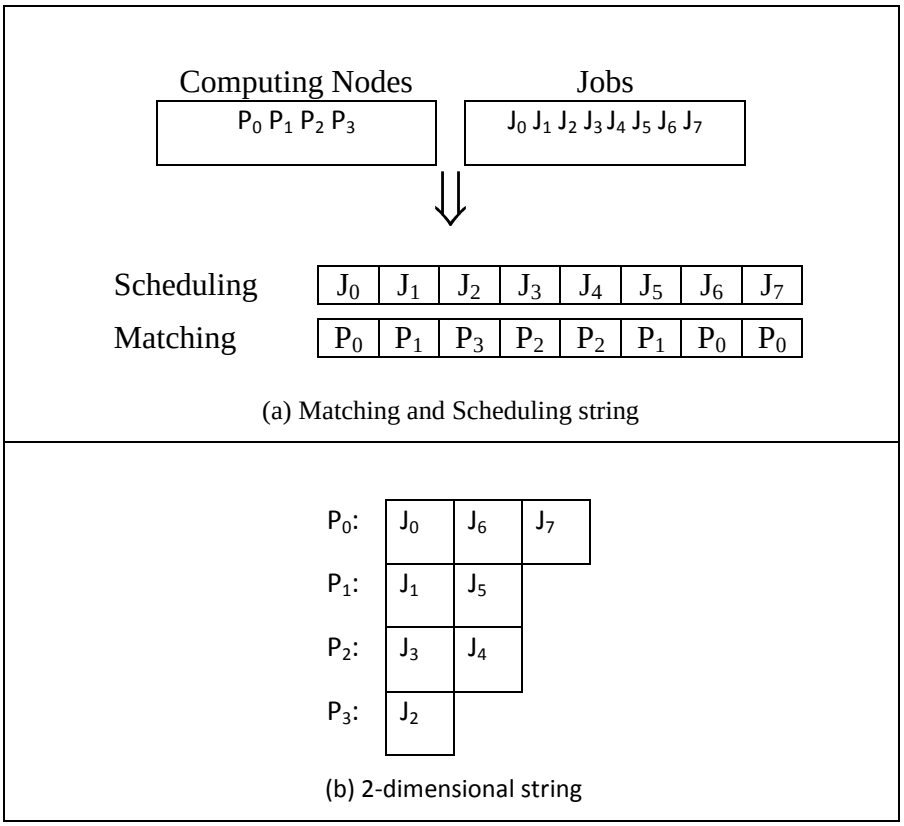

Figure 5.1: GA chromosome representation for scheduling.

The Fitness Function provides a value of goodness or fitness for a given chromosome. Selection refers to the technique used to select the chromosomes that will participate to the next generation. In most cases, the probability of a chromosome participating to the next generation is dependent on the fitness value. Roulette Wheel selection combined with Elitism

is commonly used. Selection criteria will be reviewed in Section 5.1.1.

Crossover consists of exchanging genetic material between two chromosomes to produce the next generation chromosome. The most common technique is the single-point crossover, where a crossover point is chosen and the data beyond the crossover point is exchanged between the two parents. Crossover is applied according to the crossover probability (Crossover Rate - P_c) and is usually between 0.6 and 1.0. If elitism is not used, a high value of crossover rate can result in loosing good candidate solutions. Figure 5.2 shows the crossover operation for the chromosome representation for dependent job scheduling. The crossover point (chosen at random) is also shown. The recombination process is performed as follows:

- The first part of the child chromosomes is identical to their parents (first 4 positions).
- The second part of chromosome $C_k^{'}$ is formed by reordering the second part of C_n according to the topological sort defined by C_m. The second part of the matching string of $C_k^{'}$ is also defined by C_m (i.e. the same job / processor pairs are maintained).

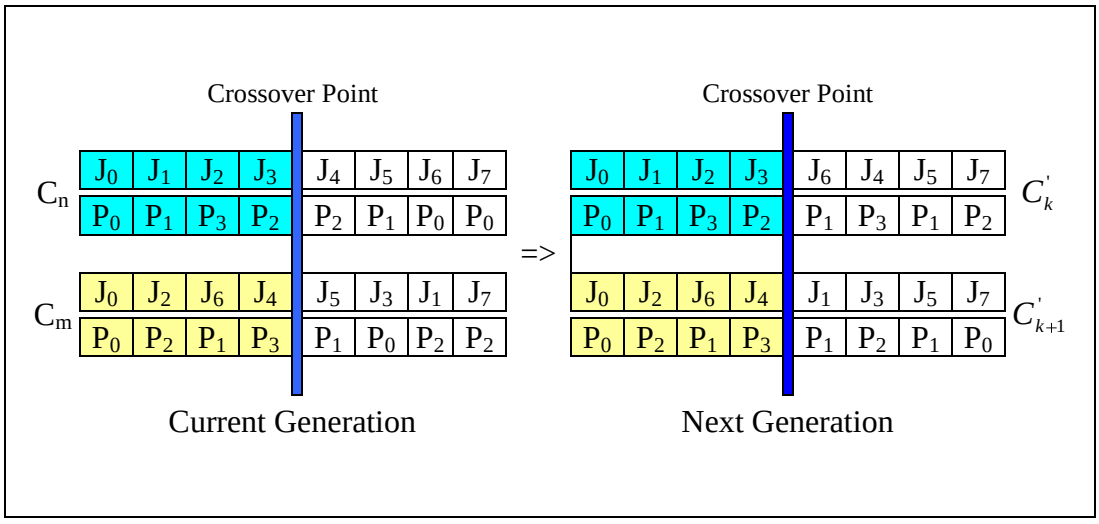

Figure 5.2: Crossover operation for matching and scheduling.

Mutation is analogous to biological mutation and is used to maintain genetic diversity from one generation of a population of chromosomes to the next. This is typically implemented by

randomly altering bits in the chromosome structure. Mutation rate (P_m) refers to the probability of a chromosome being mutated and is typically low. A high mutation rate is analogous to re-initialising the population. Mutation can be performed by either mutating the scheduling part or the matching part of the chromosome. Scheduling mutation (Figure 5.3) is performed as follows:

- Selecting at random the position to be mutated (in this case position 3 / job J_3)
- Locating the first parent to the left of the chosen job (in this case J_0) and the first child to the right (in this case J_5). This defines the range in which the chosen job can be shifted to maintain a topological sorted list.
- The new location is chosen at random by generating a number within the defined range. The random number generated in the example is 2.
- Job J_3 is relocated to position 2 (job / processor pairs are maintained).

Matching mutation consists of selecting a position at random and modifying the corresponding assigned processor with a randomly chosen processor. In Figure 5.4, position 3 (job J_3) is chosen and the assigned processor is modified from P_2 to P_0.

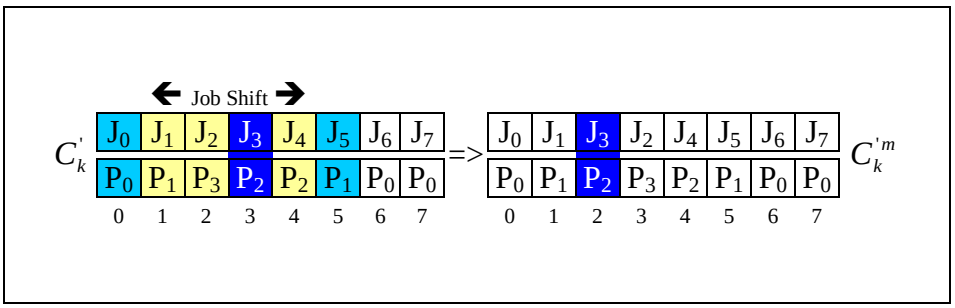

Figure 5.3: Scheduling mutation.

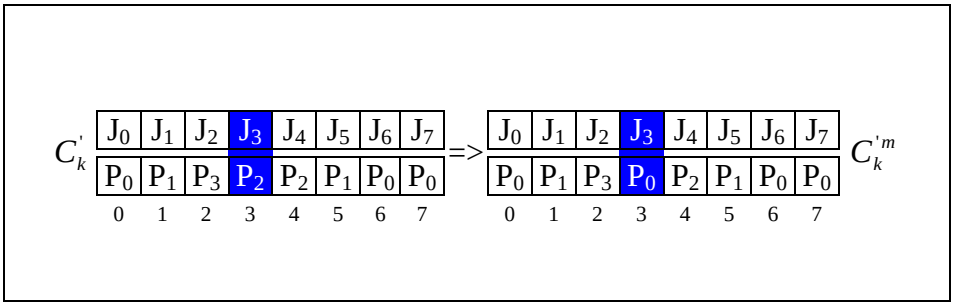

Figure 5.4: Matching mutation.

Termination is the criteria used to determine whether the solution evaluated is sufficient.

Typical criteria used are:

- A solution is found that satisfies the minimum criteria
- Fixed number of generations reached
- The GA search has converged to a solution.
- No improvement in best fitness over a specified number of generations.
- Combinations of the above.

Algorithm 5-1 provides the general structure for a GA implementation.

```
(1)  choose initial random population
(2)  Repeat
(3)      evaluate fitness for all chromosomes
(4)      randomly select pairs of best-ranking individuals to reproduce.
(5)      perform Crossover with crossover probability Pc
(6)      mutate the 2 offsprings with mutation probability Pm
(7)      place the resulting chromosomes in the new population
(8)
(9)  until convergence or generation limit
```

Algorithm 5-1: Classical GA implementation.

5.1.1 GA Selection

Various GA selection criteria exist [78, 79]. The most commonly used are:

a. Random selection is non fitness-proportionate and selection is based on a random choice that is not dependent on fitness.

b. Roulette-Wheel selection is a fitness-proportionate selection criterion in which each chromosome is allocated a slice of a circular roulette-wheel proportional to the fitness of the chromosome. Selection is then performed by randomly selecting a point on the roulette-wheel. The performance of Roulette-Wheel selection deteriorates when there are large differences between the fitness value of individuals in the population [80]. Alternative fitness evaluation methods used in literature are:

- $fitness = \frac{1}{makespan}$ is used by Li, Yu & Qi [81].
- $fitness = \frac{f_{ave}}{f_i}$ is used by Wang et al.[76], where f_{ave} is average makespan and f_i is the makespan of the individual chromosome.
- $fitness = C_{\max} - FT(I)$ is used by Hou, Ansari & Ren [77], where $C_{\max}$ is the maximum completion time observed so far and $FT(I)$ is the completion time.
- $fitness = \lambda \cdot makespan + (1-\lambda) \cdot mean(flowtime)$ is used by Carretero, Xhafa & Abraham [82], where $mean(flowtime)$ refers to the mean finalisation time of jobs. This is a form of multi-objective optimisation.

c. Elitism allows the GA to retain the *n* best chromosomes in each generation unchanged. This method overcomes one of the flaws of Roulette-Wheel selection, in which it is possible to lose some of the best chromosomes.

d. Rank selection (also known as Rank-Based Roulette-Wheel) allocates a slice on the Roulette-Wheel based on rank (not fitness) to *P* chromosomes. The 0^{th} ranked chromosome is assigned an angle A_0 and the ratio of angles allocated between the i^{th} chromosome and the $(i+1)^{th}$ is a constant, *R*:

$$R = \frac{A_i}{A_{i+1}} \quad where \;\; R > 1 \; and \;\; 0 \geq i > P - 1 \; ; P = Population \;\; Size$$

Typically: $R = 1 + \frac{1}{P}$

Rank selection avoids giving a large share of offspring to a small number of highly fit chromosomes, which may lead to early convergence. Wang et al. [76] demonstrate that rank based selection is less sensitive to crossover and mutation probability selections.

e. Baker selection, also known as Stochastic Universal Sampling, like Roulette-Wheel selection, allocates a slice of a circular roulette-wheel proportional to the fitness of the chromosome. Instead of performing random selection *P* times, *P* equally spaced locations on the Roulette-Wheel are selected.

f. In Tournament Selection, *n* members of the population are picked at random and each chromosome is allocated a Roulette-Wheel slice according to rank and another random number is generated to choose between the *n* chromosomes. Tournament Selection is computationally more efficient in that it does not require time consuming operations used in other fitness-proportionate methods (sorting chromosomes and computing sum of fitness).

5.1.2 Self Adaptation of GA Parameters

A key factor in GA implementation is choosing the correct values for the various parameters such as population size, crossover-rate and mutation-rate. Important factors for establishing GA parameters are:

a. GA parameters are inter-dependent and cannot be individually optimized [83, 78].

b. GA parameters are dependent on the type of problem being solved [84, 85, 86].

c. The optimal mutation rate is not only different for every problem encoding, but will also vary with evolution time [83].
d. Optimal values of GA parameters can be determined through experimentation [85, 86].

From a theoretical perspective, mutation is independent from fitness and is usually a small constant (typically in the range of 1 mutation every generation [87]). However, using a constant low mutation rate will result in an insufficient variation in the population to find each time dependent optimum, whereas a constant high mutation rate would be disruptive. The above considerations and the work done on GAs in Dynamic Environments have produced techniques for Self Adaptation of GA Parameters (guided GA Operators) and Triggered Hypermutation [88, 89]. These techniques are aimed at preventing local optimum, premature convergence and low convergence speed.

5.2 Load Balancing in MapReduce

MapReduce supports heterogeneous computing environments in which the computing nodes may vary in computing capacities. However, current implementation of the Hadoop implementation of MapReduce only supports first-in-first-out (FIFO) and fair scheduling without load balancing taking into consideration the varied resources of computers. A genetic algorithm based load balancing scheme is designed to speed up the performance of MRNN in training.

5.2.1 Genetic Algorithm Design

To solve an optimization problem, genetic algorithm solutions need to be represented as chromosomes encoded as a set of strings which are normally binary strings. However, a binary representation is not feasible as the number of *mappers* in a Hadoop cluster environment is normally large which will result in long binary strings. A decimal string to

represent a chromosome in which the data chunk assigned to a *mapper* is represented as a gene is employed.

In Hadoop, the total time ( ) of a *mapper* in processing a data chunk consists of the following four parts:

- Data copying time ( ) in copying a data chunk from Hadoop distributed file system to local hard disk. It depends on the available network bandwidth and the writing speed of hard disk.
- Processor running time ( ) in processing a data chunk.
- Intermediate data merging time ( ) in combining the output files of the *mapper* into one file for *reduce* operations.
- Buffer spilling time ( ) in emptying filled buffers.

(5-1)

Let

- be the size of the data chunk.
- be the writing speed of hard disk in MB/second.
- be the network bandwidth in MB/second.
- be the speed of the processor running the *mapper* process in MB/second.
- be the size of the buffer of the *mapper*.
- be the ratio of the size of the intermediate data to the size of the data chunk.
- be the number of frequencies in processing intermediate data.
- be the number of times that buffer is filled up.
- be the volume of data processed by the processor when the buffer is filled up.

- s be the sort factor of Hadoop.

Therefore

(5-2)

Here depends on the available resources of hard disk and network bandwidth. The slower one of the two factors will be the bottleneck in copying data chunks from Hadoop distributed file system to the local hard disk of the *mapper*.

(5-3)

When a buffer is filling, the processor keeps writing intermediate data into the buffer and in the mean time the spilling process keeps writing the sorted data from the buffer to hard disk. Therefore the filling speed of a buffer can be represented by . Thus the time to fill up a buffer can be computed by . As a result, for a buffer to be filled up, the processor will generate a volume of intermediate data with the size of which can be computed using equation (5-4)

(5-4)

The total amount of intermediate data generated from the original data chunk with a size of is . Therefore the number of times for a buffer to be filled up can be computed using equation (5-5).

(5-5)

The time for a buffer to be spilled once is , therefore the time for a buffer to be spilled for times is . Then we have

(5-6)

The frequencies in processing intermediate data can be computed using equation (5-7).

(5-7)

When the merging occurs once, the whole volume of intermediate data will be written into the hard disk causing an overhead of . Thus if the merging occurs times, the time consumed by hard disk IO operations can be represented by . We have

(5-8)

The total time to process data chunks in one processing wave in MapReduce Hadoop is the maximum time consumed by participating *mappers*, where

, (5-9)

According to divisible load theory [17], to achieve a minimum , it is expected that all the *mappers* to complete data processing at the same time:

(5-10)

Let

- be the processing time for the *mapper*.
- be the average time of the *mappers* in data processing,

Based on equations (5-9) and (5-10), the fitness function is to measure the distance between and . Therefore, the fitness function can be defined using equation (5-11) which is used

by the genetic algorithm in finding an optimal or a near optimal solution in determining the size for a data chunk.

(5-11)

5.2.2 Crossover

To maintain the diversity of the chromosomes, the algorithm needs functions of crossover. Crossover recomposes the homologous chromosomes via mating to generate new chromosomes or so called offspring. The generated offspring inherit the basic characteristics of their parents. Some of them may adapt to the fitness function better than their parents did, so they may be chosen as parents in next generation. Based on crossover, the algorithm can keep evolving until an optimal offspring has been found. In this algorithm, to gain the effective of design and operations, single-point crossover which refers to set only one crossover point randomly in the chromosome has been employed. The process of crossover is performed in the following steps:

- Randomly select pairs of the chromosomes (schedulers) as parents to mate.
- In each pair, randomly select a position as crossover point. If the length of the chromosome is k then there will be $k-1$ available points.
- In each pair, the chromosomes change their parts which are after the crossover point with each other according to crossover probability p.

However in the algorithm simply crossing the chromosome may cause one problem. As each gene is the value of the actual volume of data each Map instance takes, to change the members of genes may differentiate the original total volume of data $\sum_{i=1}^{k} D_i$. Assume the

original total volume of data is $\sum_{i=1}^{k} D_i$ and the volume of data after crossover is $\sum_{i=1}^{k} d_i$, then the difference $\Delta D = \left| \sum_{i=1}^{k} D_i - \sum_{i=1}^{k} d_i \right|$ should be considered and processed. In the algorithm ΔD is divided into k parts. The size of each part is randomly assigned. And then these k parts will be randomly added to or removed from k genes in the chromosome. Thus the total size of processed data in one wave could be guaranteed.

5.2.3 Mutation

To avoid the local optima of the algorithm, mutation has been introduced into our algorithm. Mutation could mutate genes in a chromosome based on smaller probabilities. Moreover new individuals could be generated. So that combined with crossover the information loss due to the selection could be avoided. Thus the validity of the algorithm could be guaranteed. Mutation contributes in two main aspects in the genetic algorithm.

- Improving the local search ability of the algorithm. The crossover operation could find a number of chromosomes with better adaptability from a global angle. These chromosomes are close to or helpful to gain the optimal solution. However crossover cannot execute local search in details. So using mutation to tune the values of certain genes from local detailed phase could make the chromosome much closer to the optimal solution. So the search ability is enhanced compare to that of only crossover involved.
- Maintaining the diversity of the colony moreover preventing the premature convergence of the algorithm. Mutation replaces the original genes with newly mutated genes so that the structure of a chromosome could be significantly affected.

The diversity of the colony could be maintained. And also the premature phenomenon could be prevented.

The algorithm mutates genes mainly based on simple mutation which refers that to mutate one or several genes in the chromosome based on mutation probability p . There are two steps in the mutation process:

- Randomly select a gene to be the mutation point. Base on mutation probability p to decide if the chromosome mutates.
- If the probability decides the gene should mutate, then the value of the gene will be mutated which means a new value replaces the original value. As a result a new individual is generated.

However, it is quite similar to crossover processes that when the value of one gene mutates, the original total volume of data $\sum_{i=1}^{k} D_i$ has been changed. Assume the original volume of the gene is D_i and the volume after mutation is d_i, then the difference $\Delta D = |D_i - d_i|$. To solve ΔD issue, ΔD is divided into k parts. The size of each part is randomly assigned. And then these k parts will be randomly added to or removed from k genes in the chromosome. Thus the total size of processed data in one wave could be guaranteed. Based on this design, the algorithm has a strong ability to change its searching direction to gain the optimal solution in a large search space.

5.3 Summary

This chapter briefly introduced genetic algorithms for job scheduling in computing environments. MapReduce does not have a sophisticated scheme in support of load balancing

in heterogeneous computing environments. For this purpose, a load balancing scheme based on a genetic algorithm was developed. The divisible load theory was employed in the implementation of the genetic algorithm with an aim to guide the evolution process for fast convergence.

Chapter 6:

Performance Evaluation of MRNN

This chapter evaluates the performance of MRNN in comparison with a sequential implementation of neural network. MRNN was first evaluated in a small scale real Hadoop cluster environment. It was then evaluated in large scale simulated Hadoop environment which involved up to 1000 mappers. The load balancing scheme presented in Chapter 4 was also evaluated and the results are presented in this chapter.

6.1 The Diabetic Data Set

The diabetic data set used in the evaluation is excerpted from the UCI Machine Learning Repository[1]. The data is organised in the format of ARFF (Attribute Relation File Format). An ARFF file describes a list of instances and relations with respective to their attributes. The extension format of these files is .arff. ARFF files are divided into two different sections - header and data.

The Pima Indians Diabetes Data Set contains 8 categories and 768 instances gathered from a larger databases belonging to the National Institute of Diabetes and Digestive and Kidney Diseases. The data set has the following 8 attributes:

- Pregnant: Number of times of pregnant,
- Plasma-Glucose: Plasma glucose concentration measured using a two-hour oral glucose tolerance test,
- DiastolicBP: Diastolic blood pressure (mmHg),
- TricepsSFT: Triceps skin fold thickness (mm),
- Serum-Insulin: 2-hour serum insulin (mu U/mt)

[1] http://repository.seasr.org/Datasets/UCI/arff/diabetes.arff

- BMI: Body mass index (w in kg/h in m)
- DPF: Diabetes pedigree function
- Age: Age of the patient (years)
- Class: Diabetes onset within five years (0 or 1)

Figure 6.1 shows the header of the data set, and Figure 6.2 shows the data part of the data set.

```
% 1. Title: Pima Indians Diabetes Database
%
% 2. Sources:
%    (a) Original owners: National Institute of Diabetes and
Digestive and
%                         Kidney Diseases
%    (b) Donor of database: Vincent Sigillito
(vgs@aplcen.apl.jhu.edu)
%                           Research Center, RMI Group Leader
%                           Applied Physics Laboratory
%                           The Johns Hopkins University
%                           Johns Hopkins Road
%                           Laurel, MD 20707
%                           (301) 953-6231
@relation pima_diabetes
@attribute 'preg' real
@attribute 'plas' real
@attribute 'pres' real
@attribute 'skin' real
@attribute 'insu' real
@attribute 'mass' real
@attribute 'pedi' real
@attribute 'age' real
@attribute 'class' { tested_negative, tested_positive}
```

Figure 6.1: The header of the data set.

6.2 WEKA Package[1]

The Waikato Environment for Knowledge Analysis (WEKA) has been widely used by the research community as a landmark system for data mining and machine learning. It provides a number of machine learning algorithms and data mining tools for researchers to quickly to evaluate on different data sets. It is worth noting that the algorithms provided in WEKA are sequential ones. WEKA workbench includes algorithms for regression, classification,

[1] http://www.cs.waikato.ac.nz/ml/weka/

clustering, association rule mining and attribute selection. WEKA was used in evaluating the performance of MRNN.

```
@data
6,148,72,35,0,33.6,0.627,50,tested_positive
1,85,66,29,0,26.6,0.351,31,tested_negative
8,183,64,0,0,23.3,0.672,32,tested_positive
1,89,66,23,94,28.1,0.167,21,tested_negative
0,137,40,35,168,43.1,2.288,33,tested_positive
5,116,74,0,0,25.6,0.201,30,tested_negative
3,78,50,32,88,31,0.248,26,tested_positive
10,115,0,0,0,35.3,0.134,29,tested_negative
2,197,70,45,543,30.5,0.158,53,tested_positive
8,125,96,0,0,0,0.232,54,tested_positive
4,110,92,0,0,37.6,0.191,30,tested_negative
10,168,74,0,0,38,0.537,34,tested_positive
10,139,80,0,0,27.1,1.441,57,tested_negative
1,189,60,23,846,30.1,0.398,59,tested_positive
5,166,72,19,175,25.8,0.587,51,tested_positive
7,100,0,0,0,30,0.484,32,tested_positive
0,118,84,47,230,45.8,0.551,31,tested_positive
7,107,74,0,0,29.6,0.254,31,tested_positive
1,103,30,38,83,43.3,0.183,33,tested_negative
1,115,70,30,96,34.6,0.529,32,tested_positive
3,126,88,41,235,39.3,0.704,27,tested_negative
8,99,84,0,0,35.4,0.388,50,tested_negative
7,196,90,0,0,39.8,0.451,41,tested_positive
```

Figure 6.2: The data part of the data set.

6.3 Performance Evaluation of Machine Learning Algorithms

Seven machine learning techniques were evaluated using WEKA and the diabetic data set. The purpose of the evaluation was to compare the performance of these algorithms in training and diabetic data set from the aspects of both accuracy and efficiency. The seven machine learning techniques are neural network, decision tree, bagging, k-nearest neighbour (k-NN), support vector machine (SVM), boosting, Bayes net. A number of tests were carried out on a Dell computer, Microsoft Vista, RAM- 1.00 GB, Processor-520 @1.60Ghz. Figure 6.3 shows the accuracy of the 7 classifiers increases when the numbers of instances are increased in the training process.

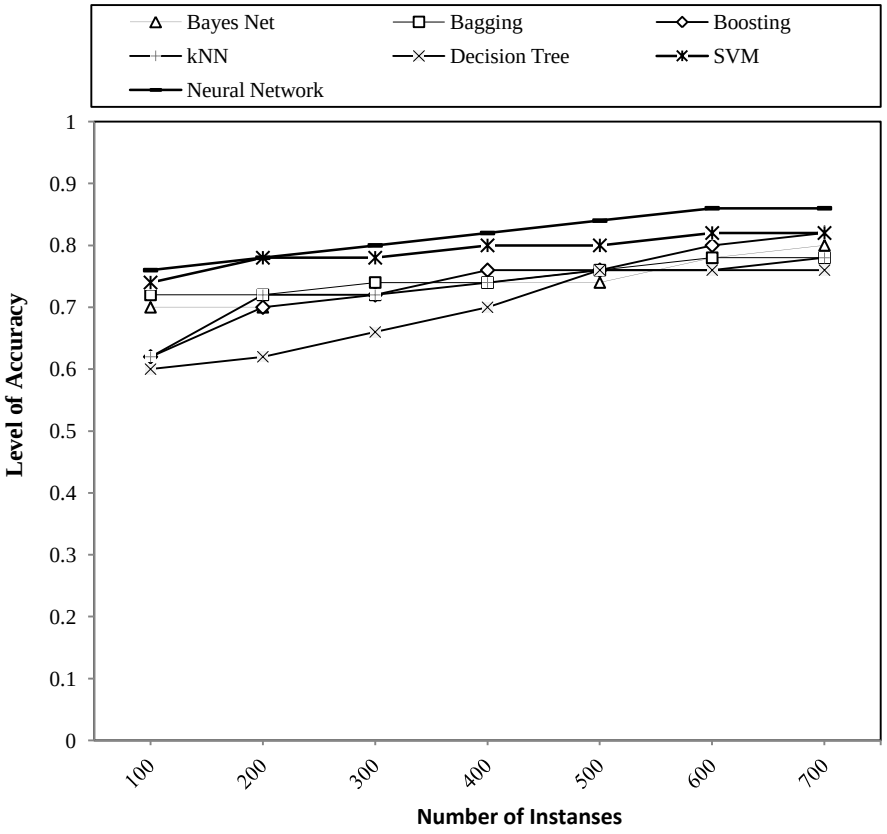

Figure 6.3: Accuracy evaluation.

Among the 7 classifiers, neural network performs the best producing most accurate results in annotating images mainly due to its non-linear learning feature. Neural network achieves a level of accuracy over 86% when 700 instances are used in the training. The decision tree C4.5 algorithm performs the worst with a level of accuracy of just 70%. The low level of accuracy is possibly due to the instability of the decision tree algorithm. Slight variations in the training data can result it different attribute selections at each choice point within the tree. The effect can be significant since attribute choices affect all descendent sub trees. However, from the results presented in Figure 6.4 we observe that neural network performs second the worst in efficiency experiencing high overhead in training the model. Based on these evaluation results neural network was selected in this research for parallelization.

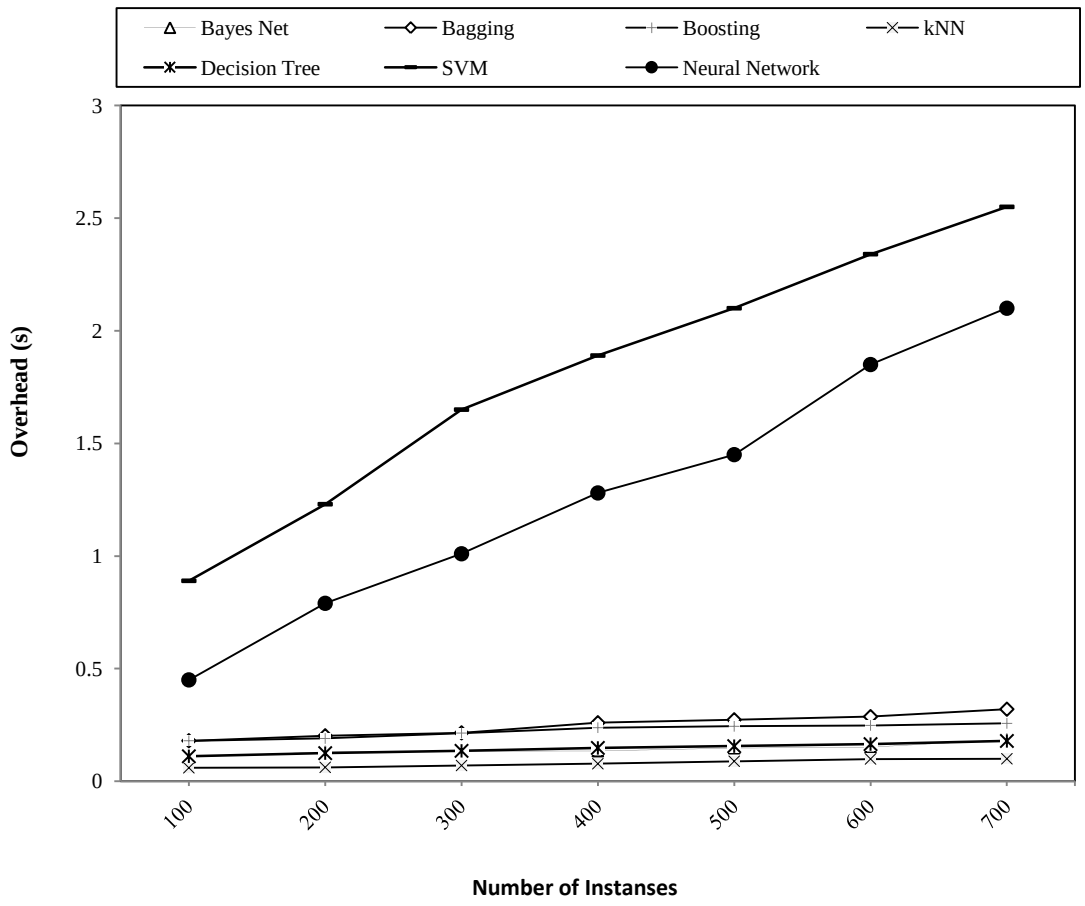

Figure 6.4: Overhead evaluation.

6.4 Evaluating MRNN in Experimental Environments

This section presents the evaluation results of MRNN in comparison with a sequential neural network. A MapReduce Hadoop cluster was set up and configured as shown in Table 6.1. We evaluated the performance of MRNN from the aspects of efficiency and accuracy. To evaluate the efficiency of MRNN, the size of the data set was increased with up to 60000 instances. Figure 5.5 shows the efficiency results of MRNN using 12 mappers.

Table 6.1: MapReduce Hadoop cluster configuration.

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
Software environment			
SVM	WEKA 3.6.0 (SMO)		
OS	Fedora10		
Hadoop	Hadoop 0.20		
Java	JDK 1.6		

From Figure 6.5 it can be observed that MRNN outperforms the sequential neural network with an increasing number of instances. The overhead of the sequential neural network increases sharply with an increasing number of instances. For example, the sequential neural networks consumes 1890 seconds in training the data set with 60000 instances.

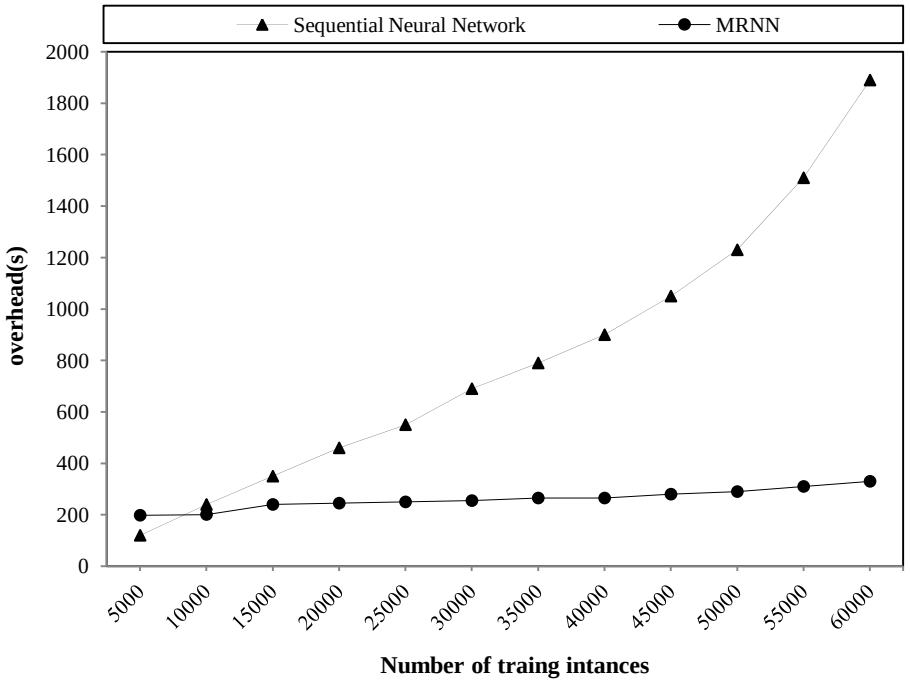

Figure 6.5: Efficiency of MRNN.

Compared with the sequential neural network, MRNN shows a stable overhead with small variations in training the data set with varied sizes. It is worth noting that the sequential neural network performs better than MRNN when the number of instances is 5000. The main reason is that when the number of instances is low like 5000, the overhead of MRNN in training is mainly caused by MapReduce Hadoop in management of job running.

The impact of the number of mappers on the efficiency of MRNN was also evaluated. Figure 6.6 shows the overhead of MRNN in training decreases with an increasing number of mappers which varies from 4 to 12. Figure 6.7 shows another view of the results presented in Figure 5.6.

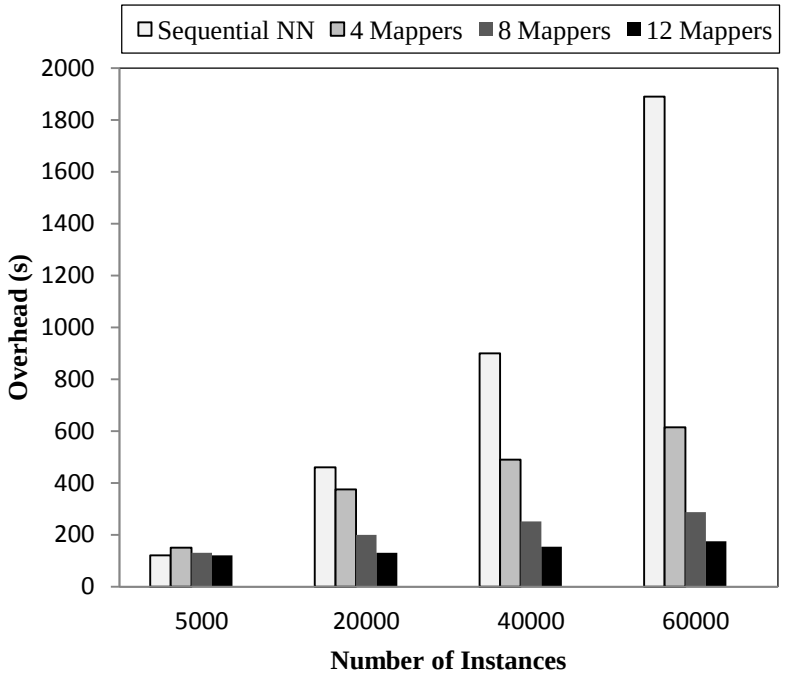

Figure 6.6: Reduced overhead of MRNN with increased number of mappers.

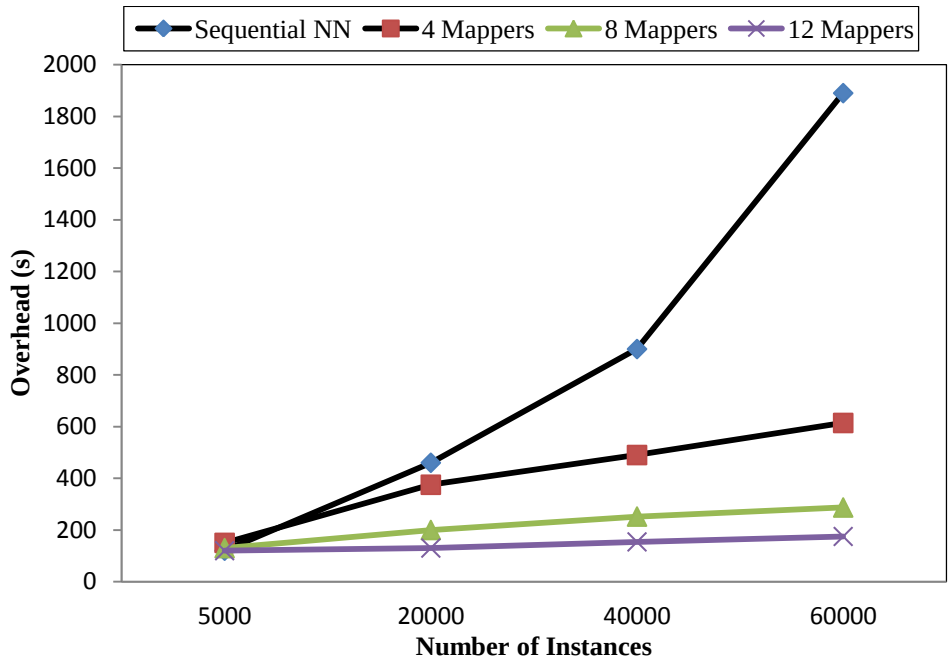

Figure 6.7: The efficiency of MRNN with varied numbers of mappers.

Furthermore we evaluated the accuracy of the sequential neural network and MRNN in classification and presented the results in Table 6.2 using 6000 instances. It is clear that the accuracy of MRNN is slightly lower than that of the sequential one due to the splitting of the data set with an accuracy level of 82%.

Table 6.2: Accuracy evaluation results.

	Sequential SMO	**MRNN 12 *Mappers***
Correctly Classified	≈ 86 %	≈ 82 %
Incorrectly Classified	≈ 14%	≈ 18%

6.5 Evaluation MRNN in Simulation Environments

MRNN was also evaluated in large scale simulated MapReduce Hadoop cluster environments. This section presents the simulation results. First, it briefly introduces the HSim [90], a Hadoop simulator which has been developed by the research group at Brunel University.

6.5.1 HSim - Hadoop Simulator

HSim follows a master-slave mode in its design in which there is one master (server) node and a number of slave (working) nodes. Figure 6.8 shows the architecture of HSim.

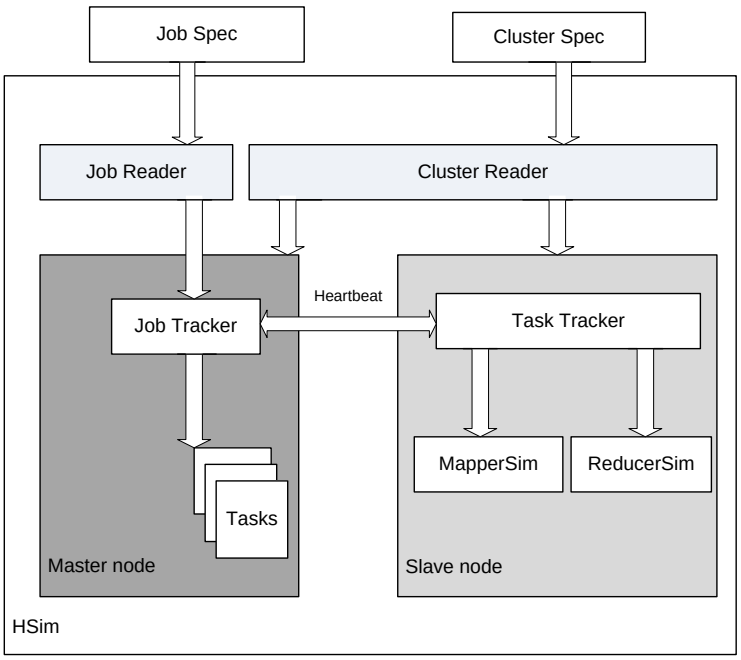

Figure 6.8: HSim components.

To perform a simulation, the Cluster Reader component reads the cluster parameters from the Cluster Spec to create a simulated Hadoop cluster environment. A specified number of nodes are initialized and arranged using a certain type of topology. After the cluster is configured, the node parameters will be processed by the Cluster Reader as well to specify the types of nodes including processors, hard disk, memory, Master node, Slave nodes, Map instances and Reduce instances. This initialization process can create both homogeneous nodes and heterogeneous nodes. Then the simulated cluster is ready for incoming jobs retrieved from the job queue using different job schedulers. The Job Spec will be processed by the Job Reader component and jobs will be submitted to HSim for simulation.

The simulated Map instances (MapperSim), Reduce instances (ReducerSim), JobTracker and TaskTrackers are located on these nodes. The Master node is the Namenode of Hadoop framework which contains JobTracker to correspond and schedule the tasks. The Slave nodes

are the Datanodes of Hadoop framework which contains TaskTrackers. On Slave nodes Map instances and Reduce instances perform data processing tasks. From Figure 3.1 it can be observed that when a job is submitted to a simulated Hadoop cluster, the JobTracker splits the job into several tasks. Then TaskTracker and JobTracker will communicate with each other via messaging based on heartbeats. If the JobTracker finds that all the Map tasks have been finished, then the Reduce instances will be notified to be ready for merging phase. Moreover if the JobTracker finds all Reduce tasks have been finished, then the job will be considered as finished. If the Map tasks have not been finished yet, the TaskTrackers will be notified to choose a Map task or a Reduce Task based on their availabilities.

HSim was validated firstly with 3 benchmark results presented by Pavlo et al. [91] - Grep Task, Selection Task and UDF Aggregation Task.

6.5.1.1 Grep Task

This task simulated exactly what Pavlo et al. did in their benchmarking work. HSim simulated the cluster using 1 node, 10 nodes, 25 nodes, 50 nodes and 100 nodes respectively. Two different scenarios have been tested, one is that each node is assigned 535MB data to process, and the other is that 1TB data is submitted to the cluster. Each scenario was evaluated 5 times. The simulation results are plotted in Figure 6.9 and Figure 6.10 respectively which are close to the benchmark results. The confidence intervals of the results are small in both scenarios (in the range of 0 and 2.6 seconds in the first scenario and in the range of 4.1 and 7.6seconds in the second scenario) showing a stable performance of HSim.

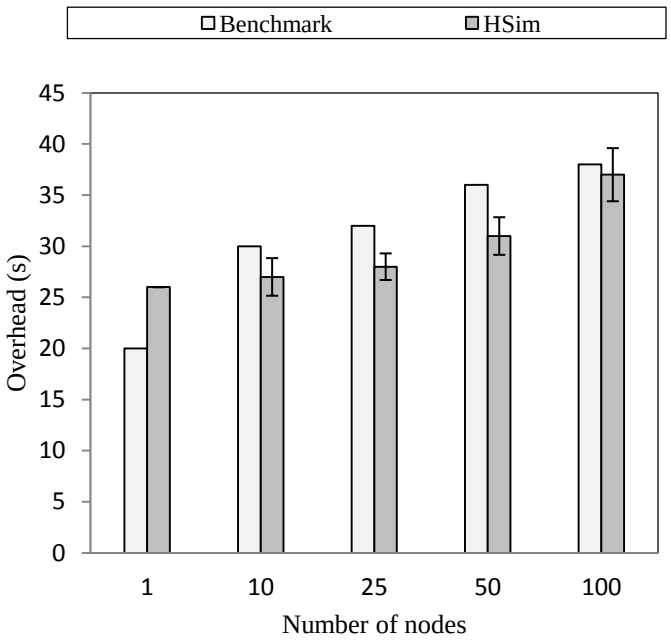

Figure 6.9: Grep Task evaluation (535MB/node).

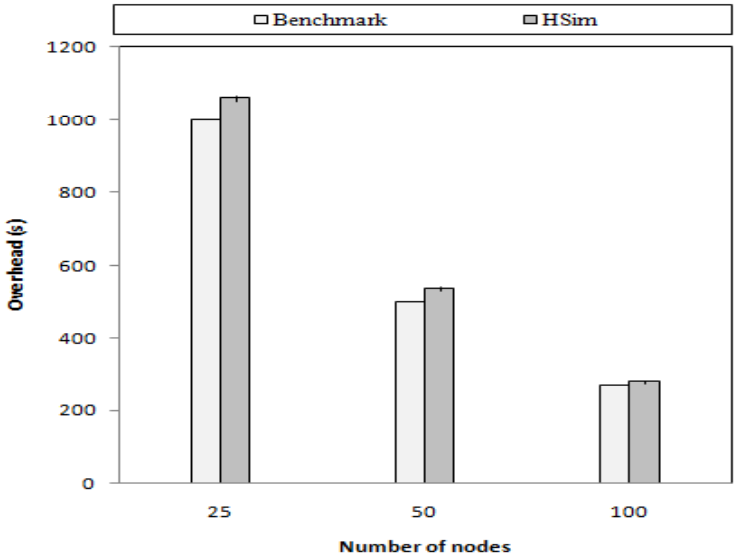

Figure 6.10: Grep Task evaluation (1TB/cluster).

6.5.1.2 Selection Task

The Selection Task was designed to observe the performances of Hadoop framework dealing with complex tasks. Each node processes 1GB ranking table to retrieve the target pageURLs with a user defined threshold. We simulated this task and the results are shown in Figure 6.11.

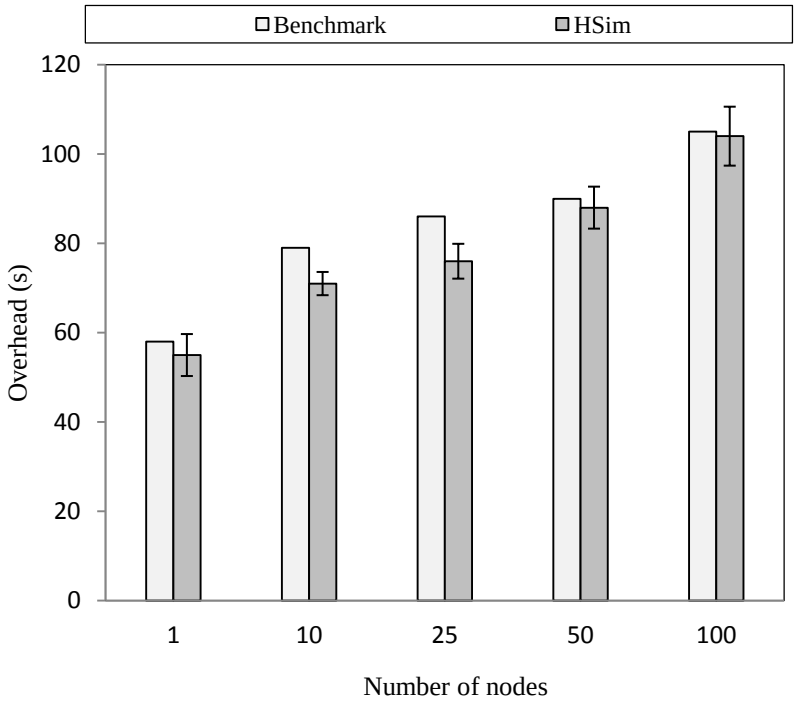

Figure 6.11: Selection task evaluation.

From Figure 6.11 it can be clearly observed that the simulated results are close to the benchmark results, and the confidence intervals are small, in the range of 2.6 and 6.6 seconds.

6.5.1.3 UDF Aggregation Task

The UDF Aggregation Task reads the generated document files and searches for all the URLs appeared in the contents. And then for each unique URL, HSim counts the number of unique pages that refers to that particular URL across the entire set of files. The simulation results are shown in Figure 6.12 which again are close to the benchmark results with small confidence intervals, showing a high stability of HSim.

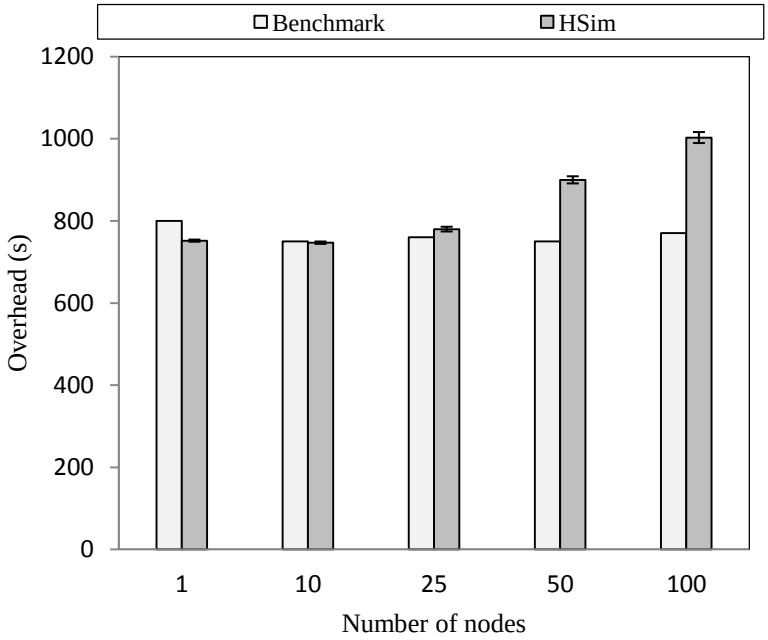

Figure 6.12: Aggregation task evaluation.

6.5.2 Evaluating the Impacts of Hadoop Parameters on the Performance of MRNN

To study the impacts of Hadoop parameters on performance of MRNN, a cluster has been simulated with the configurations as shown in Table 6.3. Each node has a processor with 4 cores. The number of *mappers* is equal to the number of processor cores. There are two *mappers* running on a single processor with two cores. The speeds of the processors were simulated in terms of the volume of data in MB processed per second. The following sections show the impacts of Hadoop parameters on the performance of MRNN.

Table 6.3: The simulated Hadoop environment.

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

6.5.2.1 Number of Reducers

From Figure 6.13 it shows that the number of *reducers* does not affect the performance of *mappers* greatly. This is because mappers and reducers work almost independently in Hadoop environments.

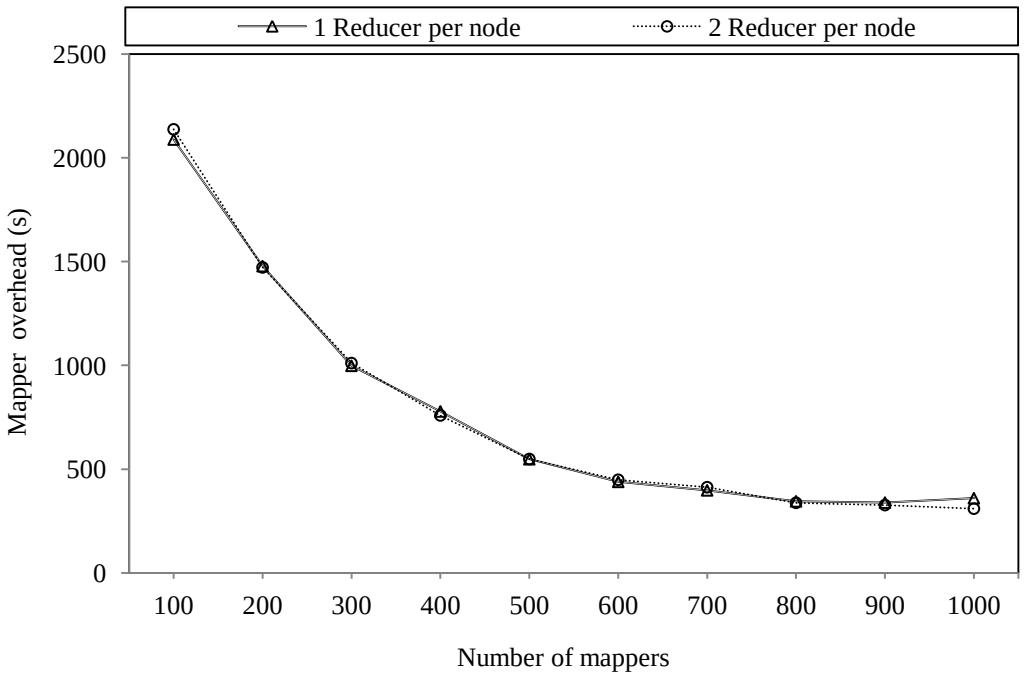

Figure 6.13: The impact of the number of reducers on *mapper* performance.

Figure 6.14 shows the impact of the number of *reducers* on the overall overhead when processing a job. Allocating multiple *reducers* on one node increases results in the shared resources issue. Especially for MRNN a number of hard disk operations involved, the shared hard disk gives worse performance in reducing phase of the *reducers* than that of unshared hard disk.

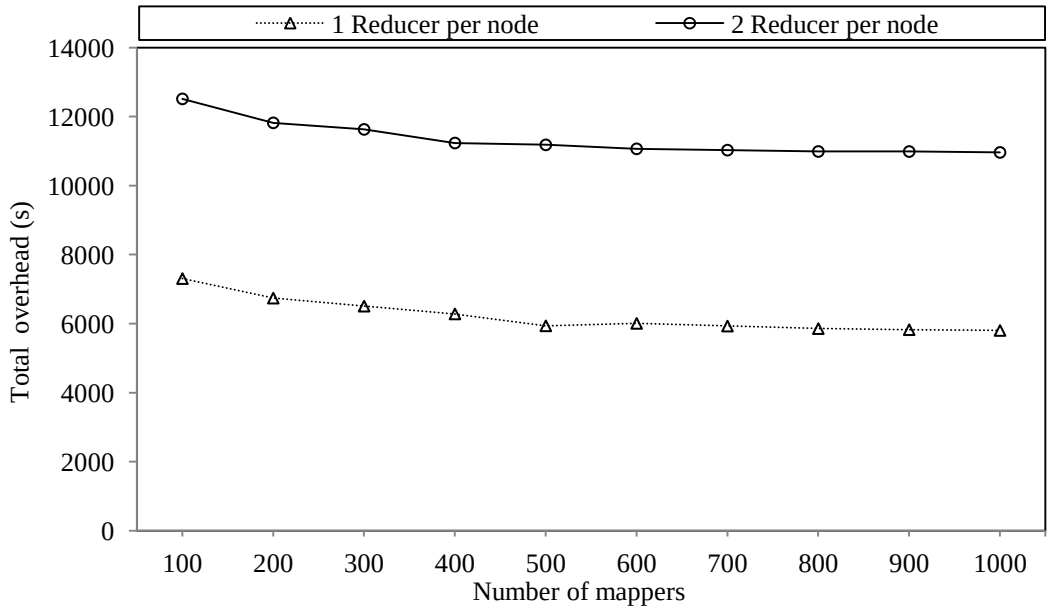

Figure 6.14: The impact of the number of reducers on the total process.

6.5.2.2 Sort Factor

In Hadoop, The parameter of sort factor controls the maximum number of data streams to be merged in one wave when sorting files. Therefore, the value of sort factor affects the IO performance of MR-LSI. From Figure 6.15 it can be observed that the case of using sort factor 100 gives a better performance than sort factor 10. When the value of sort factor is changed from 10 to 100, the number of spilled files will be increased which reduces the overhead in merging.

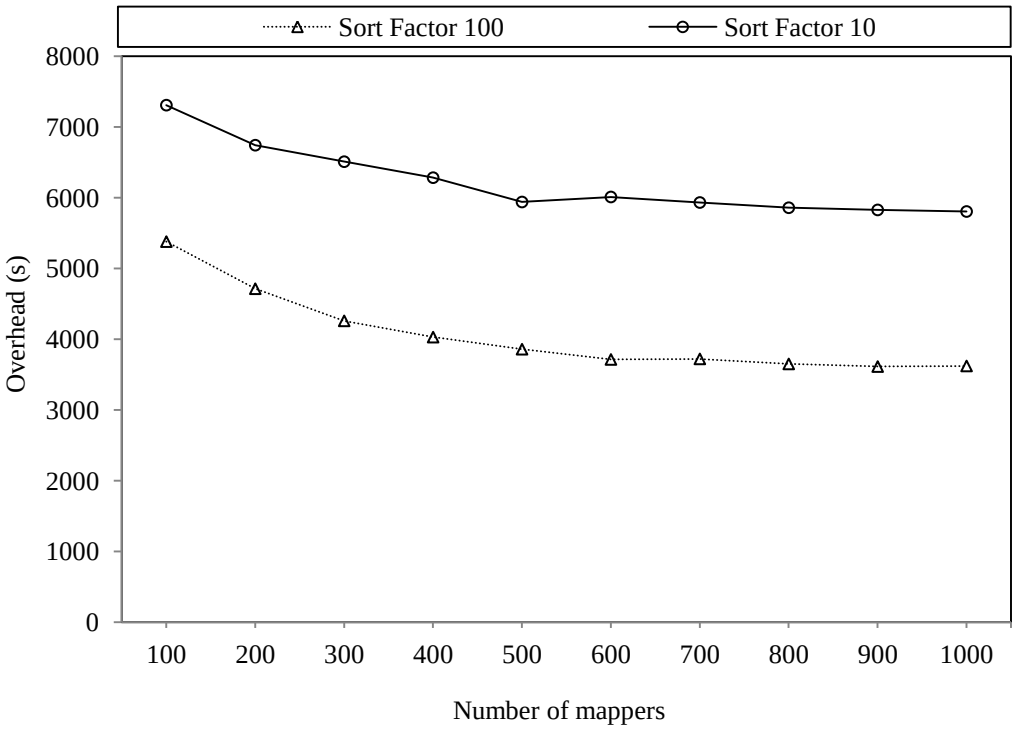

Figure 6.15: The impact of sort factor.

6.5.2.3 Buffer Size

The buffer size in Hadoop contributes to IO performance, and it affects the performance of a processor. The default value of a buffer size is 100MB. The performance of MRNN with a data size of 1000MB is tested. As shown in Figure 6.16, the *mappers* generate a small number of spilled files when using a large size buffer which reduces the overhead in merging. Furthermore, a large buffer size can keep the processor working without any blocking for a long period of time.

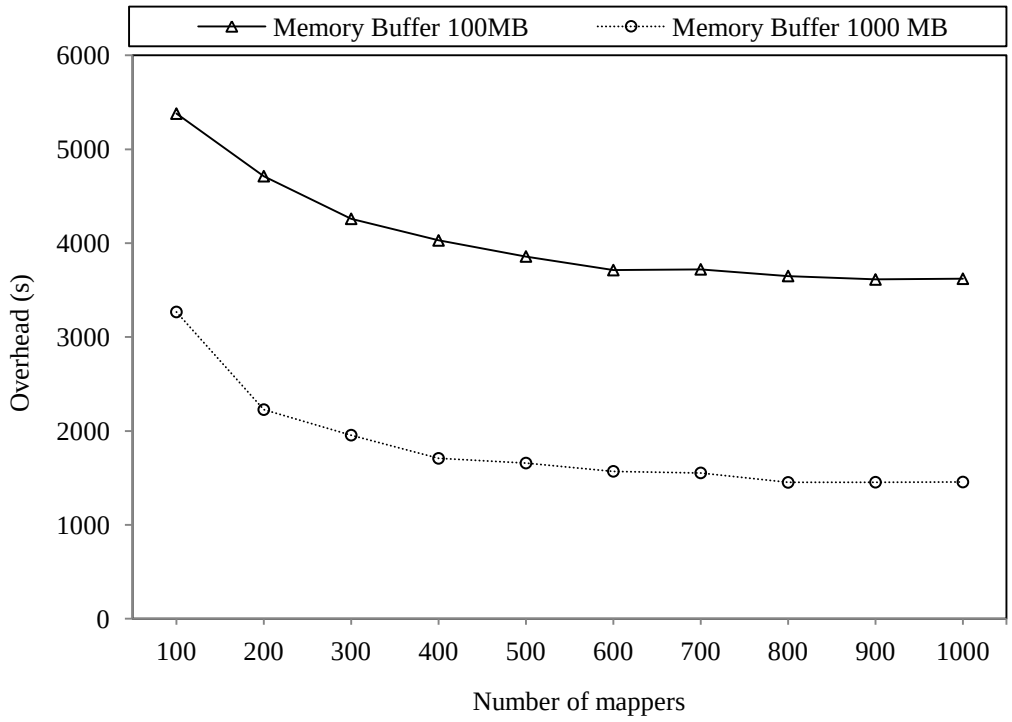

Figure 6.16: The impact of buffer size.

6.5.2.4 Chunk Size

Each *mapper* processes a data chunk at a time. Thus the size of data chunks highly affects the number of processing waves of *mappers*. From Figure 5.17 it can be observed that using a large size for data chunks reduces the overhead of mappers in processing, and also reduces the total overhead of the process as shown in Figure 6.18. However, both of the two chunk sizes produce the same performance when the number of *mappers* increases to 800 and 900 respectively. In the case of chunk size 64MB, to process 100,000MB data, using 800 *mappers* needs ——— waves to finish the job. In the case of chunk size 100MB, using 800 *mappers* needs ——— waves to finish the job. Similarly, using 900 *mappers* needs 2 waves to process the 100,000MB data in both cases. When the number of *mappers* reaches 1000, the performance of the two cases with different data sizes varies.

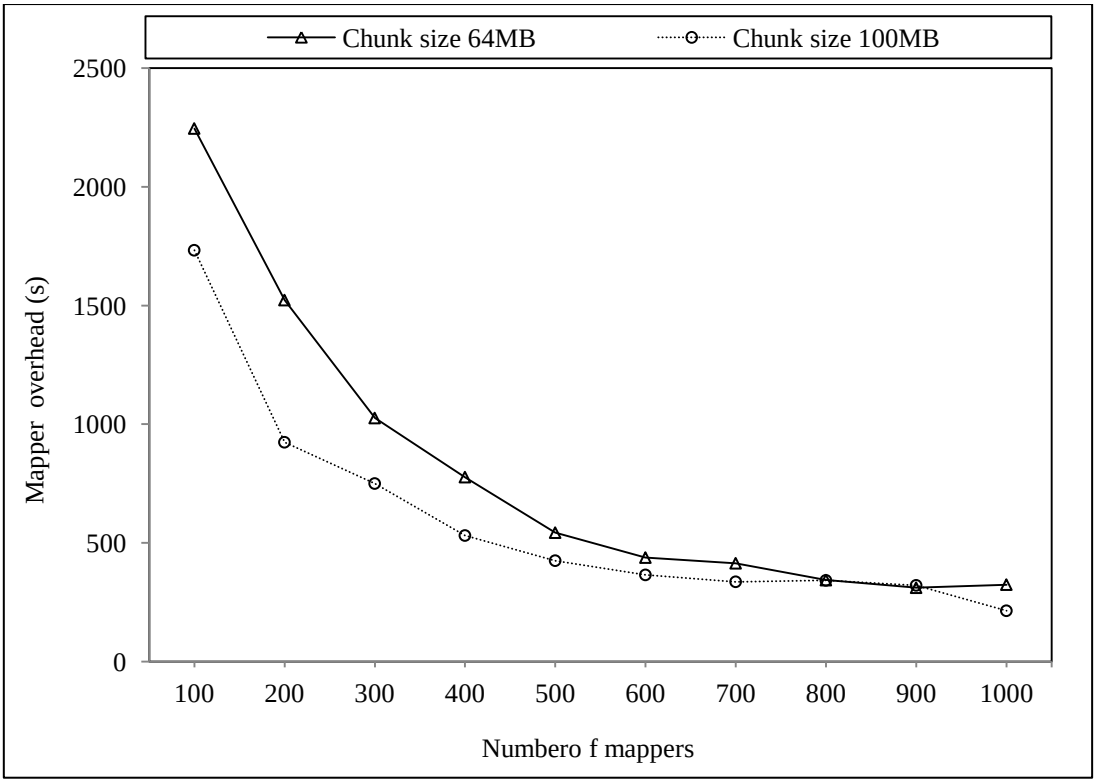

Figure 6.17: The impact of data chunk size on the *mappers* in MRNN.

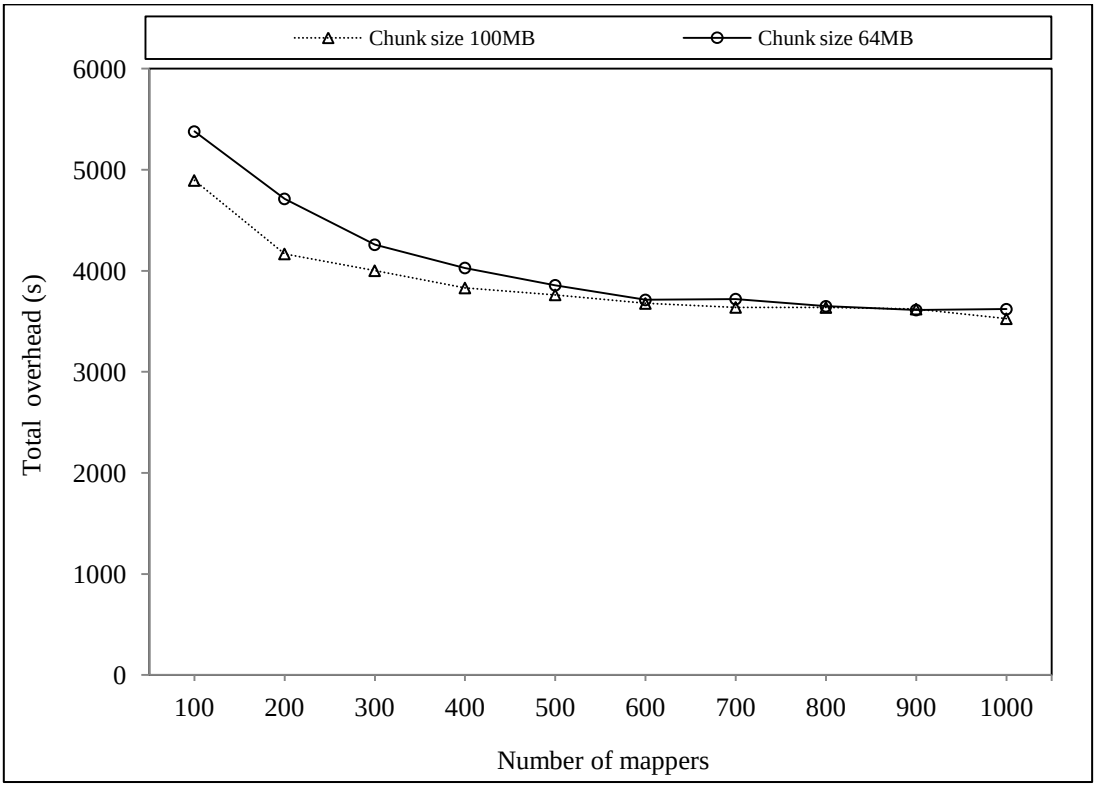

Figure 6.18: The impact of data chunk size on MRNN.

6.5.2.5 CPU Processing Speed

Figure 6.19 shows the impacts caused by different processing speed of processors. From the figure we can observe clearly that a faster processor can gain better performance compared to that of a slower processor.

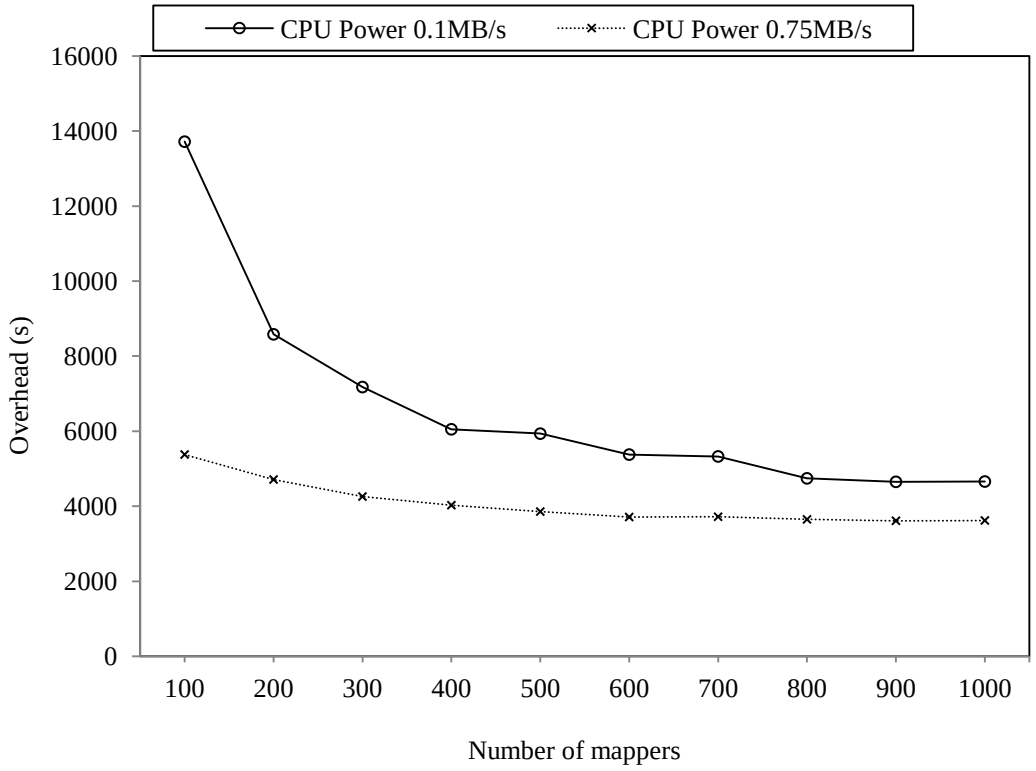

Figure 6.19: The impact of different CPU processing speeds.

6.5.2.6 Number of Reducers

Figure 6.20 shows that increasing the number of *reducers* enhances the performance of MRNN when the number of *reducers* small. More reducers are used more resources will need to be consumed due to Hadoop's management work on the *reducers*. In some cases multiple *reducers* need an additional job to collect and merge the results of each *reducer* to form a final result. This can also cause larger overhead.

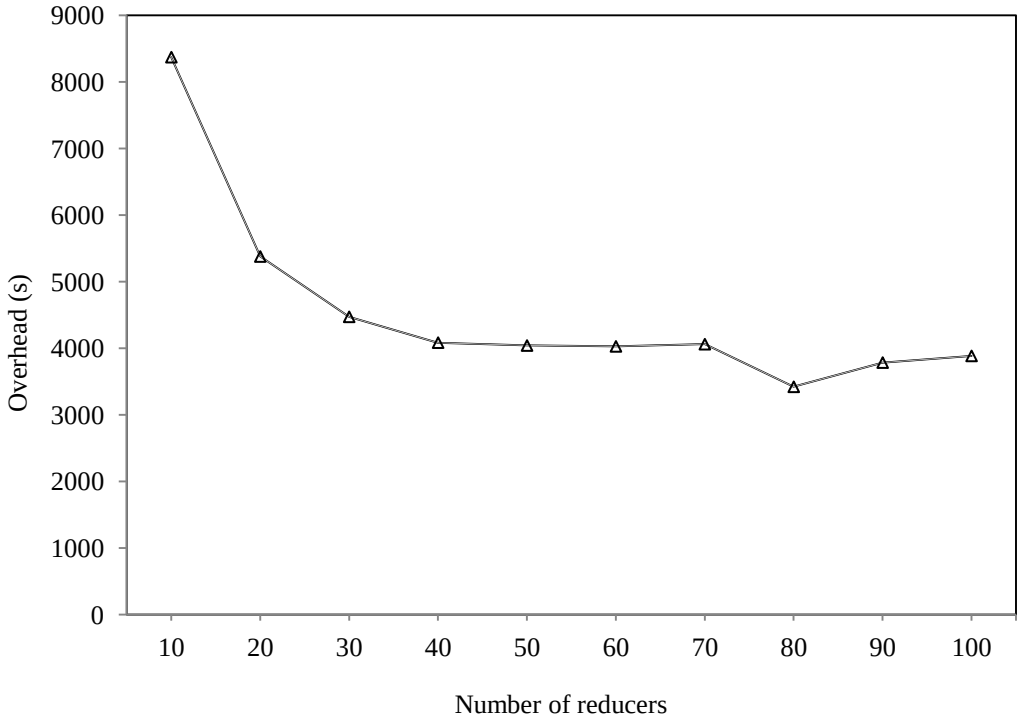

Figure 6.20: The impact of reducers.

6.5.3 MRNN Performance Evaluation in Simulation Environments

Using HSim, we simulated a number of Hadoop environments and evaluated the performance of MRNN from the aspects of scalability, the effectiveness in load balancing and the overhead of the load balancing scheme.

6.5.3.1 Scalability

To further evaluate the scalability of the MRNN algorithm, we employed HSim and simulated a number of Hadoop environments using a varying number of nodes up to 250. Each Hadoop node was simulated with 4 *mappers*, and 4 input data sets were used in the simulation tests. Table 6-4 shows the configurations of the simulated Hadoop environments.

Table 6-4: Simulated Hadoop cluster for scalability evaluation.

Simulation environment	
Number of simulated nodes:	250
Data size:	100,000MB
CPU processing speed:	0.75MB/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
Total number of Map instances:	4 *mappers* per node

From Figure 6.21 it can be observed that the processing time of MRNN decreases as the number of nodes increases. It is also worth noting that there is no significant reduction in processing time of MRNN beyond certain number of nodes. This is primarily due to the fact that Hadoop incurs a higher communication overhead when dealing with a larger number of computing nodes.

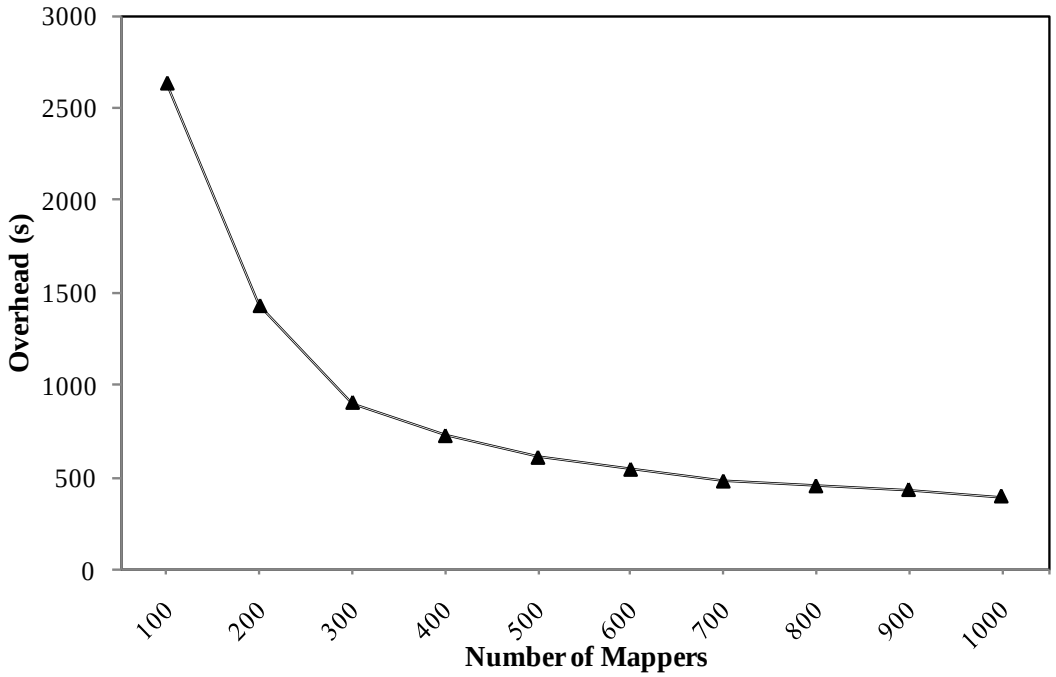

Figure 6.21: The scalability of MRNN in simulation environments.

6.5.3.2 Load Balancing

Table 6-5 shows the configurations of the simulated Hadoop environments in evaluating the effectiveness of the load balancing scheme of MRNN.

Table 6-5: Hadoop configurations for load balance evaluation.

Simulation environment	
Number of simulated nodes	20
Number of processors in each node	1
Number of cores in each processor	2
The processing speeds of processors	depending on heterogeneities
Heterogeneities	from 0 to 2.28
Number of hard disk in each node	1
Reading speed of Hard disk	80MB/s
Writing speed of Hard disk	40MB/s
Number of Mapper	each node employs 2 *map* instances
Sort factor:	100

To evaluate the load balancing algorithm we simulated a cluster with 20 computing nodes. Each node has a processor with two cores. The number of *mappers* is equals to the number of cores. Therefore we run two *mappers* on a single processor with two cores.

The speeds of the processors are generated based on the heterogeneities of the Hadoop cluster. In the simulation environments the total processing power of the cluster was

where *n* represents the number of the processors employed in the cluster and represents the processing speed of processor. For a Hadoop cluster with a total computing capacity denoted with , the levels of heterogeneity of the Hadoop cluster can be defined using equation (6-1).

(6-1)

In the simulation, the value of heterogeneity varied from 0 to 2.28. The reading and writing speeds of hard disk were measured from the experimental results. Figure 6.22 shows the performance of MRNN with load balancing and Figure 6.23 shows another view in this apsect.

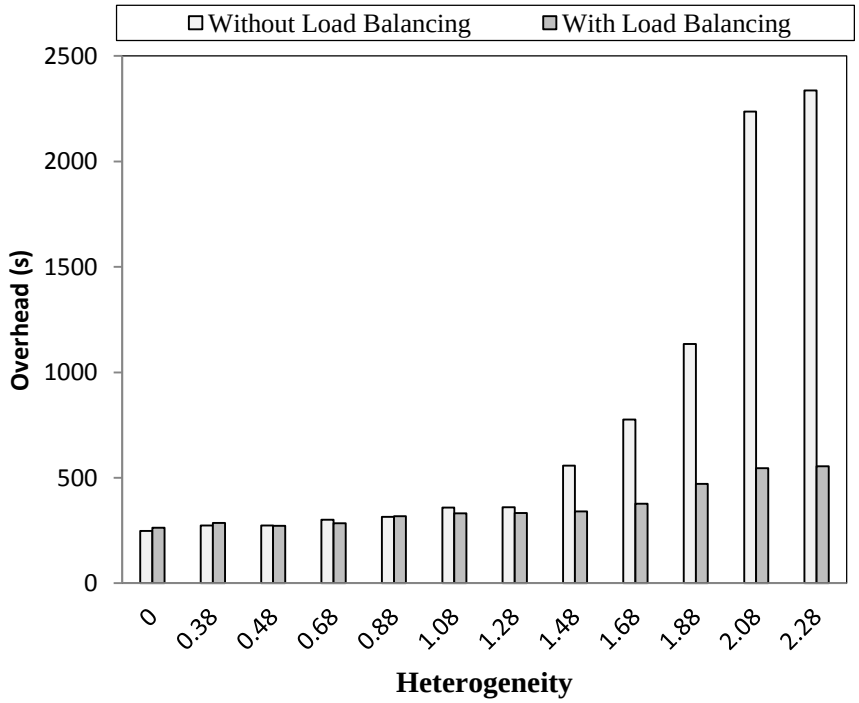

Figure 6.22: The performance of MRNN with load balancing.

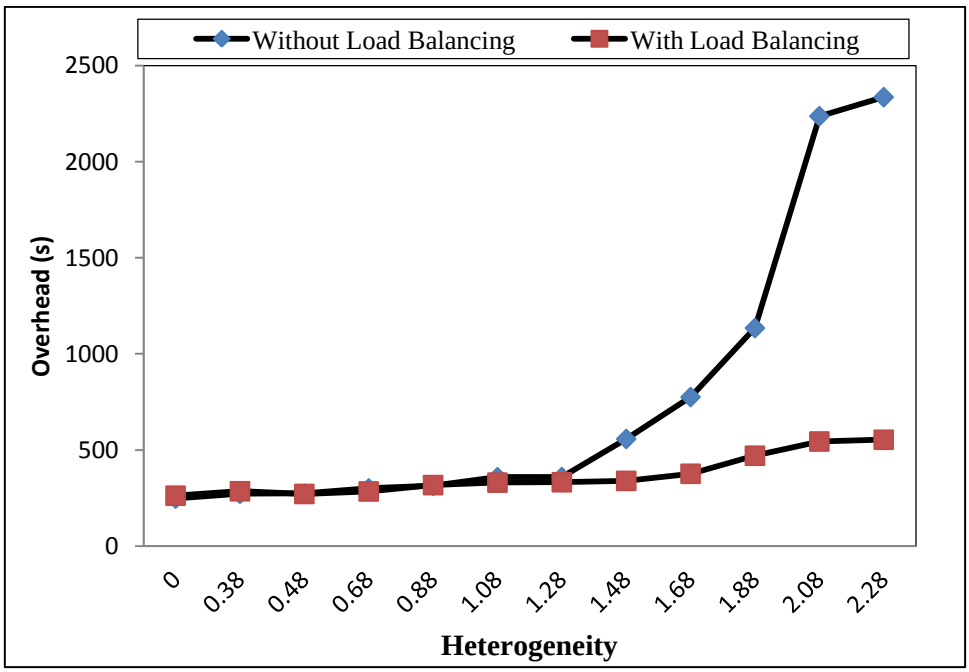

Figure 6.23: Another view on the performance of MRNN with load balancing.

From Figure 6.22 and Figure 6.23 it can be observed that when the level of heterogeneity is less than 1.08 indicating homogeneous environments, the load balancing scheme does not make any difference to the MRNN algorithm in performance. However the load balancing scheme reduces the overhead of MRNN significantly with an increasing levels of heterogeneity showing that the resource aware MRNN can optimize resource utilization in highly heterogeneous computing environments.

We kept the degree of heterogeneity the same in the simulated cluster but varied the size of data from 1GB to 10GB. This set of tests was used to evaluate how the load balancing scheme performs with different sizes of data sets. Figure 6.24 shows that the load balancing scheme always reduces the overhead of MRNN in data training using varied volumes of data, and Figure 6.25 shows another view in this aspect.

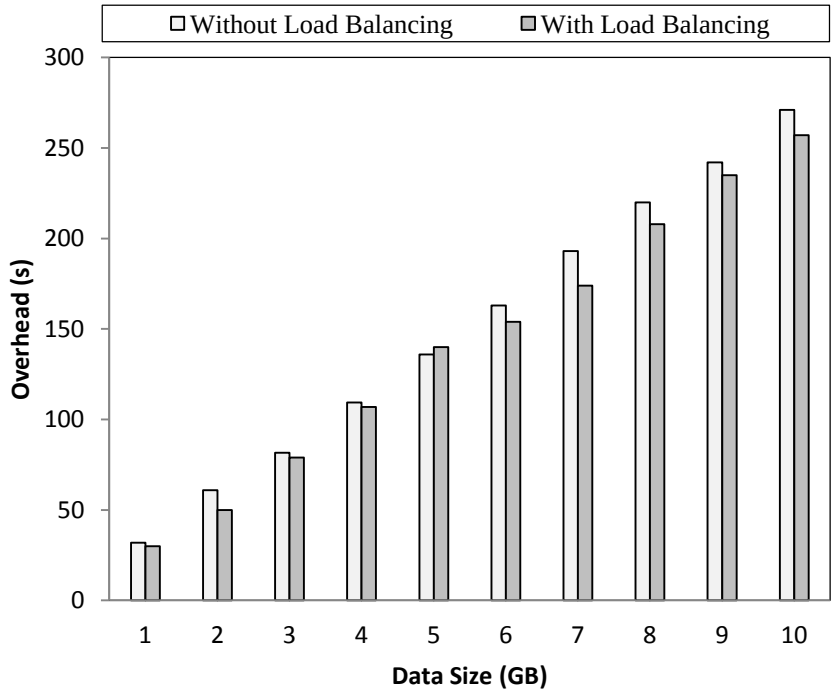

Figure 6.24: The performance of MRNN with varied sizes of data.

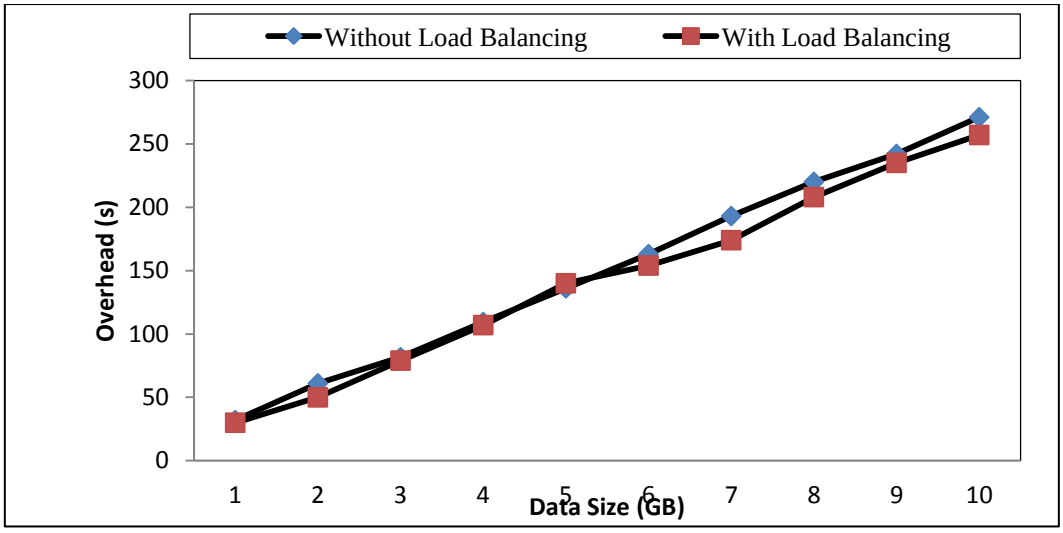

Figure 6.25: Another view on the performance of MRNN with varied sizes of data.

6.5.3.3 Overhead of the Load Balancing Scheme

The load balancing scheme builds on a genetic algorithm whose convergence speed affects the efficiency of MRNN in training. To analyze the convergence speed of the genetic algorithm, we varied the number of generations and measured the overhead of MRNN in processing a 10GB dataset in a simulated Hadoop environment. Figure 6.26 shows that MRNN has a quick convergence process in reaching a stable performance.

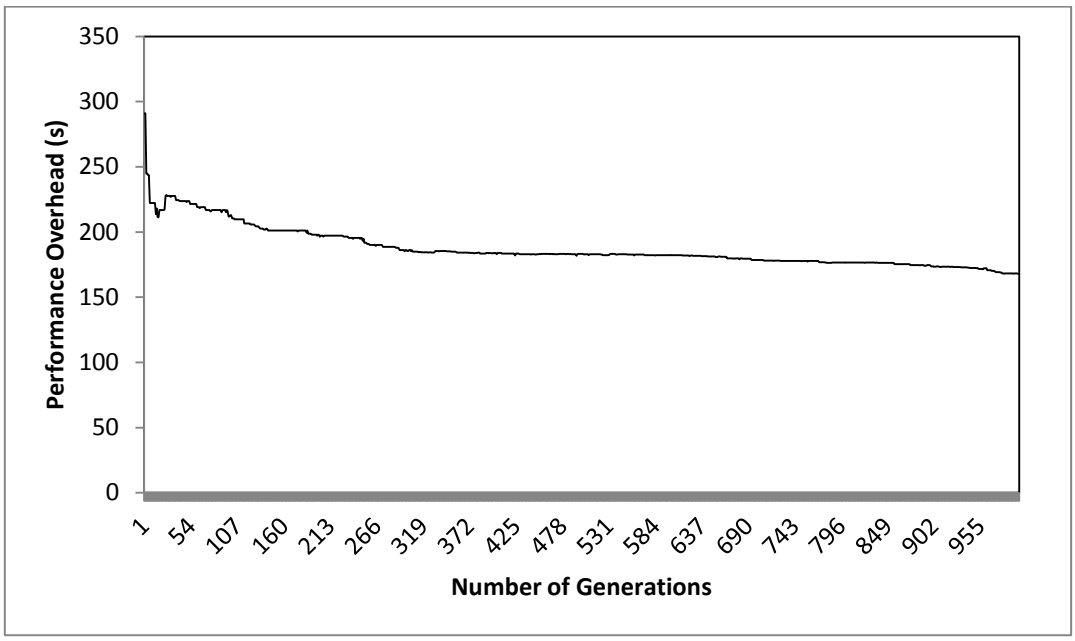

Figure 6.26: The convergence of the MRNN.

6.6 Summary

Seven sequential machine algorithms were evaluated from the aspects of accuracy an efficiency. From the evaluation results it was observed that neural network performed best in terms of accuracy but incurred high overhead in data training. This motivated the research on parallelization of neural network to speed up the training process.

MRNN was evaluated extensively from the aspects of scalability, load balancing, convergence of the genetic algorithm. Simulation results have shown its effectiveness in speeding up the neural network training process. The impacts of Hadoop parameters on the performance of MRNN were also evaluated.

Chapter 7:

Conclusion and Future Work

This chapter concludes the thesis and points out some future work.

7.1 Conclusion

Diabetes has become one of the most severe decease due to an increasing number of diabetes patients globally. A large amount of digital data on diabetes is collected through various channels. How to utilize these data sets to help doctors make a decision on diagnosis, treatment and prediction of diabetic patients poses many challenges to the research community.

This thesis investigated mathematical models with a focus on neural networks for large scale diabetes data modelling and analysis by utilizing modern computing technologies such as grid computing and cloud computing. These computing technologies provide users an inexpensive way to have access to unlimited computing resources over the Internet for solving data and computationally intensive problems.

The thesis evaluated the performance of seven representative machine learning techniques in classification of diabetes data in experimental environments. Among these techniques, neural network performed best in terms of accuracy in classification, but incurred high overhead in data training due to its non-linear learning process. Based on the evaluation results, neural network was selected in the research as a mathematical model for diabetic data modelling and analysis with a focus on neural network parallelization.

Various parallelization schemes were studied and data partition parallelism was adopted in the implementation. A parallel neural network called MRNN was developed. MRNN builds on the MapReduce programming model which has become an enabling parallel and

distributed computing technology in support of data intensive applications in the cloud. The MapReduce framework facilitates a number of important functions such as partitioning the input data, scheduling MapReduce jobs across a cluster of participating nodes, handling node failures, and managing the required network communications. Another feature of MapReduce is its support for heterogeneous environments in which computer nodes may have varied resources in computing. By partitioning the diabetic data set into a number of equally sized data blocks, the workload in training is split into among a number of computing nodes for speedup in data training. MRNN was evaluated from the aspects of accuracy in classification and efficiency in data training. It was first evaluated in small scale experimental environments which had 12 mapper and subsequently was evaluated in large scale simulated environments using up to 1000 mappers. Both experimental and simulations results have shown the effectiveness of MRNN in classification, and its high scalability in data training.

With regard to the fact that MapReduce does not have a sophisticated job scheduling scheme for heterogonous computing environments in which the computing nodes may have varied computing capabilities, the thesis developed a load balancing scheme based on genetic algorithms with an aim to balance the training workload among heterogeneous computing nodes. The nodes with more computing capacities will receive more MapReduce jobs for execution. One challenge of using genetic algorithms in solving optimization problems is that it is hard to determine the number of evolutionary generations to achieve an optimal or a near optimal value. It is obvious that more generations are used better results would be achieved, but with a slower convergence. As a result, classical genetic algorithms normally suffer from the problem of slow convergence. A genetic algorithm based load balancing scheme was developed in the thesis for scheduling MapReduce jobs in heterogeneous computing environments with an aim to achieve minimum makespan in job execution. The divisible load theory was employed to guide the genetic algorithm the direction for evolution which has

shown a fast convergence. The load balancing scheme was evaluated in large scale simulated MapReduce environments with varied levels of heterogeneity using different sizes of data sets. All the results have shown that the genetic algorithm based load balancing scheme significantly reduce makespan in job execution in comparison with the time consumed without load balancing.

7.2 Future Work

Although the results achieved in the thesis are encouraging, the thesis can be further improved in the following two aspects.

7.2.1 Ontology Enhancement

The splitting a data set into a number of small data blocks can potentially degrade the accuracy of MRNN in classification. The modeling, specification and representation of real-world elements as a set of inter-linked concepts within a domain describe basic ontology. Ontologies [92] enable the re-use of domain knowledge by ensuring formal and unambiguous concept representation. They provide a sound basis for information exchange and automated processing, including varying degrees of reasoning. In the context of machine learning, the majority of ontology based work tends to be biased towards the application of such techniques for the description and representation of user preferences. This is believed to be primarily due to the simplicity of the approach as well as the potential for contributed effectiveness towards improving machine learning from an end user perspective.

The application of ontology and semantics in the context of machine learning facilitates the definition and understanding of the diabetic data set in a better and more formal way [93, 94, 95]. The ability to exchange intelligence and subsequently the potential for machines to process it in a formal and interoperable fashion provides numerous opportunities. Annotating

diabetic data sets with metadata brings numerous benefits including augmented intelligence, context richness and formalization. The incorporation of domain knowledge can facilitate, improve automated filtering processes as well as increase the scope for classification accuracy from a machine learning perspective [96, 97]. The accuracy of MRNN can be enhanced with the ontology augmented intelligence to the original training data sets and re-compute the neural network model with the user augmented intelligence.

7.2.2 Dynamic Load Balancing

The developed genetic algorithm based load balancing scheme only considers static MapReduce cluster environments in which the computing capacities of the computers do not dynamic change from time to time. However, many MapReduce enabled environments such as a public cloud have varied computing capacities due to the following two reasons:

- The workload of computers may dynamically change with varied user requests.
- Computers may join or leave a public cloud frequently.

Further work is needed to put MRNN to work in a dynamic MapReduce environment. One notable way to design a dynamic load balancing algorithm is to implement a static load balancing algorithm repeatedly in a number of time intervals for a dynamically changed environment [98]. Maeng et al. [99] proposed an algorithm following the same way. The experimental results show that it is a proper way of implementing static load balancing algorithm in a time interval to adapt to a dynamic environment. The approach can enhance the performance of the cluster. Zomaya et al. [100] followed the same way to design their load balancing algorithm. They employed genetic algorithm in each time interval to achieve the optimized job scheduler according to the speeds of processor employed in the cluster. To facilitate the design, they firstly use a fixed ‘window size’ [101, 99] representing the time interval. Secondly they restrict the iterations of the genetic algorithm to be 10 times. From

their experimental results, the performance of cluster is enhanced greatly using their scenario. All these research efforts will need to be investigated for their effectiveness on enhancement of the performance of MRNN.

7.2.3 Evaluating MRNN in Real Cloud Environments

Nowadays a number of real cloud environments are available including Amazon EC2 Cloud[1]. A further work will need to be considered in evaluating the performance of MRNN in such a large scale real cloud system to further evaluate its scalability in data training.

[1] http://aws.amazon.com/ec2

References

[1] E. R. Carson, Decision support systems in diabetes: a systems perspective, Comput. Meth. Prog. Biomed 56 (1998) 77–91.

[2] S. Goldman, Cybernetic aspects of homeostasis, Miner. Metab vol. 1a, pp. 61–100, 1960.

[3] C. C. E.R. Carson, L. Finkelstein, Mathematical Modelling of Metabolic and Endocrine Systems: Model Formulation, Identification and Validation, Wiley, 1983.

[4] C. Cobelli and R. N. Bergman, Carbohydrate metabolism: quantitative physiology and mathematical modelling. Chichester West Sussex ; New York: Wiley, 1981.

[5] E. R. Carson, Decision support systems in diabetes: a systems perspective, Computer Methods Programs Biomed, vol. 56, pp. 77-91, May 1998.

[6] J. Ghosh and K. Hwang, Mapping Neural Networks onto Message Passing Multi-computers, J. Parallel and Distributed Computing, Apr. 1989.

[7] Y. Fujimoto, N. Fukuda, and T. Akabane, Massively Parallel Architecture for Large Scale Neural Network Simulation, IEEE Trans. Neural Networks, vol. 3, no. 6, pp. 876-887, 1992.

[8] V. Kumar, S. Shekhar, and M.B. Amin, A Scalable Parallel Formulation of the Back-Propagation Algorithm for Hypercubes and Related Architectures, IEEE Trans. Parallel and Distributed Systems, vol. 5, no. 10, pp. 1073-1090, Oct. 1994.

[9] S. Suresh, S. N. Omkar, and V. Mani, Parallel implementation of backpropagation algorithm in networks of workstations. IEEE Trans. Parallel Distrib. Syst., 16(1):24–34, 2005.

[10] Dean, J., & Ghemawat, S. (2004). MapReduce: Simplified Data Processing on Large Clusters. In Proc. of OSDI'04: Sixth Symposium on Operating System Design and Implementation, San Francisco, CA.

[11] Lämmel, R. (2007). Google's MapReduce programming model — Revisited. Sci. Comput. Program. 68, 208-237.

[12] J. Dean, S. Ghemawat, MapReduce: simplified data processing on large clusters, Communications of the ACM, 51 (2008) 107-113.

[13] Chao Tian, Haojie Zhou, Yongqiang He, Li Zha: A Dynamic MapReduce Scheduler for Heterogeneous Workloads. GCC 2009: 218-224.

[14] Quan Chen, Daqiang Zhang, Minyi Guo, Qianni Deng, Song Guo: SAMR: A Self-adaptive MapReduce Scheduling Algorithm in Heterogeneous Environment. CIT 2010: 2736-2743.

[15] Matei Zaharia, Andy Konwinski, Anthony D. Joseph, Randy H. Katz, Ion Stoica: Improving MapReduce Performance in Heterogeneous Environments. OSDI 2008: 29-42.

[16] Thomas Sandholm, Kevin Lai: MapReduce optimization using regulated dynamic prioritization. SIGMETRICS/Performance 2009: 299-310.

[17] T.G. Robertazzi, Processor equivalence for daisy chain load sharing processors, IEEE Trans. Aerospace Elect. Syst. 29 (4) (1993) 1216–1221.

[18] S. Shah, V. Gandhi (2004), Image Classification Based on Textural Features using Artificial Neural Network (ANN), JOURNAL-INSTITUTION OF ENGINEERS INDIA PART ET ELECTRONICS, INSTITUTE OF ENGINEERS INDIA.

[19] Sotiris B. Kotsiantis, Ioannis D. Zaharakis, Panayiotis E. Pintelas: Machine learning: a review of classification and combining techniques. Artif. Intell. Rev. 26(3): 159-190 (2006).

[20] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representation by error propagation," in Parallel Distributed Processing: Exploration in the Microstructure of Cognition, D. E. Rumelhart and J. L. McClelland, Eds: MIT Press, 1986.

[21] S. Haykin: Neural networks: a comprehensive foundation, 2nd Edition. Prentice Hall, New Jersey. 1999.

[22] W. Wong, S. Hsu, Application of SVM and ANN for image retrieval, European Journal of Operational Research, 173 (3) (2006) 938-950.

[23] Y. Gao, J. Fan, Semantic image classification with hierarchical feature subset selection, in: Proceedings of the ACM Multimedia Workshop on Multimedia Information Retrieval, 2005, pp. 135-142.

[24] M, Boutell, J. Luo, X. Shen, CM, Brown learning multi-label scene classification, Pattern Recognition, 37(9) (2004) 1757-1771.

[25] Y. Chen, J.Z Wang, Image categorization by learning and reasoning with regions, Journal of Machine Learning Research, 5 (2004) 913-939.

[26] C. Cusano, G. Ciocca , R. Schettini, Image annotation using SVM, in: Proceedings of SPIE Conference on Internet Imaging, 2004, pp. 330-338.

[27] Friedman N, Linial M, Nachman I, Pe'er D (2000) Using Bayesian networks to analyze expression data. Journal of computational biology : a journal of computational molecular cell biology 7: 601-620.

[28] Bremner D, Demaine E, Erickson J, Iacono J, Langerman S, Morin P, Toussaint G (2005). "Output-sensitive algorithms for computing nearest-neighbor decision boundaries". Discrete and Computational Geometry 33 (4): 593–604,

[29] Nigsch F, Bender A, van Buuren B, Tissen J, Nigsch E, Mitchell JB (2006). "Melting point prediction employing k-nearest neighbor algorithms and genetic parameter optimization". Journal of Chemical Information and Modeling 46 (6): 2412–2422.

[30] Hall P, Park BU, Samworth RJ (2008). "Choice of neighbor order in nearest-neighbor classification". Annals of Statistics 36 (5): 2135–2152.

[31] L. Breiman: Bagging predictors. Machine Learning 1996; 24(2): 123-140.

[32] Y. Freund and R. Schapire, ―Experiments with a new boosting algorithm‖ , In Machine Learning: Proceedings of the Thirteenth International Conference, pages 148–156, (1996).

[33] Y. Freund and R. Schapire, A short introduction to boosting, Journal of Japanese Society for Artificial Intelligence, 14(5):771—780, (1999).

[34] R. Owen Rigers, A framework for parallel data mining using neural networks. Technical report, Queen's University, Canada, 1997.

[35] T. Nordstrom and B. Svensson, Using and designing massively parallel computers for artificial neural networks, Journal of Parallel and Distributed Computing, 14(3), 1992, 260–285.

[36] U. Seiffert, Artificial neural networks on massively parallel computer hardware, Proceedings of the European Symposium on Artificial Neural Networks (ESANN'2002), Belgium, 2002, 319–330.

[37] M. Witbrock and M. Zagha. An implementation of backpropagation learning on GF11, a large SIMD parallel computer. Parallel Computing, 14(3):329–346, 1990

[38] E. Deprit, "Implementing Recurrent Back-Propagation on the Connection Machines," Neural Network, vol. 2, pp. 295-314, 1989.

[39] S. Mahapatra, "Mapping of Neural Network Models onto Systolic Arrays," J. Parallel and Distributed Computing, vol. 60, no. 6, pp. 667-689, 2000.

[40] S.Y. Kung and J.N. Hwang, "A Unified Systolic Architecture for Artificial Neural Networks," J. Parallel and Distributed Computing, vol. 6, pp. 357-387, 1989.

[41] B.K. Mak and U. Egecloglu, "Communication Parameter Test and Parallel Backpropagation on iPSC/2 Hypercube Multiprocessor," IEEE Frontier, pp. 1353-1364, 1990.

[42] D.S. Newhall and J.C. Horvath, "Analysis of Text Using a Neural Network: A Hypercube Implementation," Proc. Conf. Hypercubes, Concurrent Computers, Applications, pp. 1119-1122, 1989.

[43] W.M. Lin, V.K. Prasanna, and K.W. Przytula, "Algorithmic Mapping of Neural Network Models onto Parallel SIMD Machines," IEEE Trans. Computers, vol. 40, no. 12, pp. 1390-1401, Dec. 1991.

[44] Mathias Quoy, Sorin Moga, Philippe Gaussier, Arnaud Revel: Parallelization of Neural Networks Using PVM. PVM/MPI 2000: 289-296.

[45] Fabian Mörchen: Analysis of speedup as function of block size and cluster size for parallel feed-forward neural networks on a Beowulf cluster. IEEE Transactions on Neural Networks 15(2): 515-527 (2004).

[46] M. Pethick, M. Liddle, P. Werstein, and Z. Huang. Parallelization of a backpropagation neural network on a cluster computer. In: Proceedings of the 15th IASTED international conference on parallel and distributed computing and systems (PDCS 2003),2003,pp. 574–82.

[47] Zhiqiang Liu, Hongyan Li, Gaoshan Miao: MapReduce-based Backpropagation Neural Network over large scale mobile data. ICNC 2010: 1726-1730.

[48] Zimmet P, Alberti KG, Shaw J. Global and societal implications of the diabetes epidemic. Nature 2001; 414:782–787.

[49] Prevalence of Diabetes among Men and Women in China, March 25, 2010 Yang W., Lu J., Weng J., et al. N Engl J Med 2010; 362:1090 – 1101.

[50] Boyko EJ, de Cowten M, Zimmer PZ, et al. Features of the metabolic syndrome predict higher risk of diabetes and impaired glucose tolerance: A prospective study in Mauritius. Diabetes Care 2000; 23:1242–1248.

[51] Ramachandran A, Snehalatha C, Latha E, et al. Rising prevalence of NIDDM in an urban population in India. Diabetologia 1997; 40:232–237.

[52] Type 2 diabetes in children and adolescents. American Diabetes Association. Diabetes Care 2000; 23:381–389.

[53] From Zimmet P, Alberti KG, Shaw J. Global and societal implications of the diabetes epidemic. Nature 2001; 414:782–787.

[54] Report of the Expert Committee on the Diagnosis and Classification of Diabetes Mellitus. Diabetes Care 1997; 20:1183–1197.

[55] Charles MA, Shipley MJ, Rose G, et al. Risk factors for NIDDM in white population: Paris prospective study. Diabetes 1991; 40:796–799.

[56] Executive Summary: Standards of Medical Care in Diabetes—2011.

[57] Harris MI, Klein R, Welbom JA, Knuiman MW. Onset of NIDDM occurs at least 4–7 yr before clinical diagnosis. Diabetes Care 1992; 15:815–819.

[58] Harris MI, Hadden WC, Knowler WC, Bennett PH. Prevalence of diabetes and impaired glucose tolerance and plasma glucose levels in U.S. population aged 20–74 yr. Diabetes 1987; 36:523–534.

[59] From Screening for diabetes. Diabetes Care 2002; 25:21–24.

[60] Himsworth H, Kerr RB. Insulin-sensitive and insulin-insensitive types of diabetes mellitus. Clin Sci 1939; 4:119–152.

[61] Warram JH, Martin BC, Krowelski AS, et al. Slow glucose removal rate and hyperinsulinemia precede the development of type II diabetes in the offspring of diabetic parents. Ann Intern Med 1990; 113:909–915.

[62] DeFronzo RA. Lilly lecture 1987. The triumvirate: beta-cell, muscle, liver. A collusion responsible for NIDDM. Diabetes 1988; 37:667–687.

[63] Polonsky K, Jaspan J, Emmanouel D, et al. Differences in the hepatic and renal extraction of insulin and glucagon in the dog: evidence for saturability of insulin metabolism. Acta Endocrinol (Copenh) 1983; 102:420–427.

[64] Skyler J: Diabetic complications: the importance of glucose control. Endocrinol Metab Clin North Am 1996; 25:243–254.

[65] National Diabetes Data Group. Diabetes in America, 2nd ed. Washington, DC, US Government Printing Office, 1995.

[66] Aiello LP, Gardner TW, King GL, et al. Diabetic retinopathy: technical review. Diabetes Care 1998; 21: 143–156.

[67] Mogensen CE: Definition of diabetic renal disease in insulin-dependent diabetes mellitus based on renal function tests. In Mogensen CE (ed). The Kidney and Hypertension in Diabetes Mellitus, 5th ed. Boston, Kluwer, 2000, pp 13–28.

[68] Oyibo S, Jude EB, Tarawneh I, et al: A comparison of two diabetic foot ulcer classification systems: the Wagner and the University of Texas wound classification systems. Diabetes Care 2001; 24: 84–88.

[69] Schwartz NS, Shah SD, Clutter WE, et al. Glycemic thresholds for activation of glucose counterregulatory systems are higher than the threshold for symptoms. J Clin Invest 1987; 79:777–781.

[70] Wiethop BV, Cryer PE. Alanine and terbutaline in the treatment of hypoglycemia in IDDM. Diabetes Care 1993; 16:1131–1136.

[71] He, B., Fang, W., Luo, Q., Govindaraju, N. K. and Wang, T. (2008). Mars: a MapReduce framework on graphics processors. In: PACT '08: Proceedings of the

17th International conference on Parallel architectures and compilation techniques, 260–269.

[72] Taura, K., Kaneda, K., Endo, T., and Yonezawa, A. (2003). Phoenix: a parallel programming model for accommodating dynamically joining/leaving resources, SIGPLAN Not., 38, 216–229.

[73] Aarnio, T. Parallel Data Processing with Mapreduce, Available at: http://www.cse.tkk.fi/en/publications/B/5/papers/Aarnio_final.pdf.

[74] White, T. (2009). Hadoop: The Definitive Guide (2nd Ed.). CA : O'Reilly Media.

[75] Holland, J. H. (1975). *Adaptation in Natural and Artificial Systems*. Ann Arbor: University of Michigan Press.

[76] Wang, L., et al. (1997), Task Matching and Scheduling in Heterogeneous Computing Environments using a Genetic-Algorithm-Based Approach, Journal of Parallel and Distributed Computing, vol. 47, no. 1, pp. 8-22.

[77] Hou, E. S. H., et al. (1994), A Genetic Algorithm for Multiprocessor Scheduling, IEEE Transactions on Parallel and Distributed Systems, vol. 5, no. 2, pp. 113-120.

[78] Mitchell, M. (1998). An Introduction to Genetic Algorithms. London: MIT Press.

[79] Prügel-Bennett, A. (1998), A Course on Micro-Evolution and Genetic Algorithms.

[80] Yu, J. and Buyya, R. (2005), A Taxonomy of Workflow Management Systems for Grid Computing, Journal of Grid Computing, vol. 3, no. 3-4, pp. 171-200.

[81] Li, M., et al. (2006), PGGA: A Predictable and Grouped Genetic Algorithm for Job Scheduling, Future Generation Computer Systems, vol. 22, no. 5, pp. 588 - 599.

[82] Carretero, J., et al. (2005), Genetic Algorithm Based Schedulers for Grid Computing Systems, International Journal of Innovative Computing, Information and Control, vol. 3, no. 5, pp. 1-19.

[83] Bäck, T. (1992), The Interaction of Mutation Rate, Selection and Self-Adaptation within a Genetic Algorithm, In: Parallel Problem Solving from Nature, Brussels, 28-30 September 1992. pp. 87-96. Elsevier.

[84] Fogarty, T. C. (1989), *Varying the Probability of Mutation in the Genetic Algorithm*, In: Proceedings of the third International Conference on Genetic Algorithms, George Mason University, Fairfax, Virginia, June 89. pp. 104-109. Morgan Kaufmann Publishers Inc.

[85] Smith, J. and Fogarty, T. C. (1996), Self Adaptation of Mutation Rates in a Steady State Genetic Algorithm, In: Proceedings of IEEE International Conference on Evolutionary Computation, Nagoya, 20-22 May 1996. pp. 318-323.

[86] Tuson, A. and Ross, P. (1998), Adapting Operator Settings in Genetic Algorithms, Evolutionary Computation, vol. 6, no. 2, pp. 161-184.

[87] Beaty, S. J., et al. (1996), Using Genetic Algorithms to Fine-Tune Instruction-Scheduling Heuristics, In: Proceedings MPCS-96 - IEEE International Conference on Massively Parallel

Computing Systems, Ischia, 6-9 May 1996.

[88] Han, L. and Kendall, G. (2003), Guided Operators for a Hyper-Heuristic Genetic Algorithm, In: AI 2003: Advances in Artificial Intelligence, Perth, 3-5 December 2003. pp. 807-820. Springer.

[89] Klinkmeijer, L. Z., et al. (2006), A Serial Population Genetic Algorithm for Dynamic Optimization Problems, In: Benelearn'06: Proceedings of the 15th Belgian-Dutch Conference on Machine Learning, Ghent. pp. 41-48.

[90] Y. Liu, M. Li, N. K. Alham, S. Hammoud, HSim: A MapReduce Simulator in Enabling Cloud Computing, Future Generation Computer Systems (FGCS), Elsevier Science (published online 26 May 2011, http://www.sciencedirect.com/science/article/pii/S0167739X11000884).

[91] Pavlo, E. Paulson, A. Rasin, D. J. Abadi, D. J. DeWitt, S. Madden, M. Stonebraker, A comparison of approaches to large-scale data analysis, in: Proceedings of the 35th ACM International Conference on Management of Data (SIGMOD), pp. 165-178, 2009.

[92] Gruber, T. R. (1993). A translation approach to portable ontology specifications. Knowledge Acquisition, 5(2): 199–220. Academic Press.

[93] Adam Funk, Kalina Bontcheva: Ontology-Based Categorization of Web Services with Machine Learning. LREC 2010.

[94] Chang-Shing Lee, Mei-Hui Wang, Hani Hagras: A Type-2 Fuzzy Ontology and Its Application to Personal Diabetic-Diet Recommendation. IEEE T. Fuzzy Systems 18(2): 374-395 (2010).

[95] Mei-Hui Wang, Chang-Shing Lee, Kuang-Liang Hsieh, Chin-Yuan Hsu, Giovanni Acampora, Chong-Ching Chang: Ontology-based multi-agents for intelligent healthcare applications. J. Ambient Intelligence and Humanized Computing 1(2): 111-131 (2010).

[96] Chang-Shing Lee, Mei-Hui Wang, Giovanni Acampora, Vincenzo Loia, Chin-Yuan Hsu: Ontology-based Intelligent Fuzzy Agent for Diabetes Application. IEEE IA 2009: 16-22.

[97] Giovanni Acampora, Chang-Shing Lee, Mei-Hui Wang: FML-Based Ontological Agent for Healthcare Application with Diabetes. Web Intelligence/IAT Workshops 2009: 413-416.

[98] Zhang, Y., Kameda, H., and Hung, S. L. (1997). Comparison of dynamic and static load-balancing strategies in heterogeneous distributed systems. IEEE Proc., Comput. Digit. Tech. 144, 2, 100–106.

[99] Maeng, H. S., Lee, H. S., Han, T. D., Yang, S. B., and Kim, S. D. (2002). Dynamic Load Balancing of Iterative Data Parallel Problems on a Workstation Clustering. High Performance Computing and Grid in Asia Pacific Region, International Conference on High-Performance Computing on the Information Superhighway, HPC-Asia '97. Seoul, Korea.

[100] Zomaya, A. Y., and Teh, Y. H. (2001). Observations on Using Genetic Algorithms for

Dynamic Load-Balancing. IEEE transactions on parallel and distributed systems, 12, 899-911.

[101] Hwa-Chun Lin, C. S. Raghavendra: A Dynamic Load-Balancing Policy With a Central Job Dispatcher (LBC). IEEE Trans. Software Eng. 18(2): 148-158 (1992).