

**Modeling and Analyzing the Power Consumption in Query Processing**  
**For Distributed Database**

By  
Teh Pey Si

Dissertation submitted in partial fulfillment of  
the requirements for the  
Bachelor of Technology (Hons)  
(Information and Communication Technology)

September 2012

Universiti Teknologi PETRONAS  
Bandar Seri Iskandar  
31750 Tronoh  
Perak Darul Ridzuan

CERTIFICATION OF APPROVAL

**Modeling and Analyzing the Power Consumption in Query Processing  
For Distributed Database**

By

Teh Pey Si

A dissertation submitted to the  
Information Technology Programme  
Universiti Teknologi PETRONAS  
in partial fulfillment of the requirement for the  
BACHELOR OF TECHNOLOGY (Hons)  
(INFORMATION & COMMUNICATION TECHNOLOGY)

Approved by,

---

(Ms. Rozana bt Kasbon)

UNIVERSITI TEKNOLOGI PETRONAS  
TRONOH, PERAK

May 2012

## CERTIFICATION OF ORIGINALITY

This is to certify that I am responsible for the work submitted in this project, that the original work is my own except as specified in the reference and acknowledgements, and that the original work contained herein has not been undertaken or done by unspecified sources or persons.

---

(TEH PEY SI)

## **ABSTRACT**

Green computing has been generally practiced in almost all kind of fields especially in the recent years as environmental sustainability is getting more important. High power consumption increases the carbon emission which is adverse to the environment. This project focuses on applying green computing in query processing specifically for distributed database in healthcare industry. The information about a patient is stored in the database of the hospital the patient visited. However, currently this information is not being shared among hospitals which are crucial for diagnosis purpose. Hence, the objective of this project is to model the process of data retrieval from database distributed at different hospitals by using different query processing strategies and analyzes the energy consumption to access data from these distributed databases. Two strategies are used to retrieve the distributed data during simulation which are complete replication and horizontal fragmentation. Based on the analyzed result from the simulation, the identified energy-efficient strategy is complete replication which consumed lower power consumption by enabling local access to data stored in distributed database.

## **ACKNOWLEDGEMENT**

First and foremost, I would like to express my greatest gratitude to my project supervisor, Miss Rozana for her professional assistance and guidance throughout the development of this project. Her encouragement and support played a big part to ensure the success of this project

Other than that, I would also like to thanks my lab tutor, Mr Jamal for teaching me Microsoft Visual Studio and guided me during the simulation of this project. This gratitude also dedicated to my friends and family who have given all their support to motivate me.

Last but not least, precious thanks to the committee of Final Year Project of Computer Information Sciences (CIS) department of Universiti Teknologi PETRONAS (UTP) for the guidelines and assistance provided throughout the semester.

## TABLE OF CONTENTS

ABSTRACT .....	iv
ACKNOWLEDGEMENT .....	v
TABLE OF CONTENTS .....	vi
LIST OF FIGURES .....	ixx
LIST OF TABLES .....	ixx
LIST OF EQUATIONS .....	ixx
CHAPTER 1 .....	1
INTRODUCTION .....	1
1.1 BACKGROUND OF STUDY .....	1
1.2 PROBLEM STATEMENT .....	2
1.3 OBJECTIVES .....	3
1.4 SCOPE OF STUDY .....	3
1.5 RELEVANCY OF THE PROJECT .....	4
1.6 FEASIBILITY OF PROJECT WITHIN THE SCOPE AND TIME FRAME .....	4
CHAPTER 2 .....	6
LITERATURE REVIEW .....	6
2.1 INTRODUCTION .....	6
2.2 SOLUTION FOR DATA SHARING AMONG HEALTHCARE INSTITUTION .....	6
2.3 INTRODUCTION ABOUT DISTRIBUTED DATABASE .....	9
2.3.1 Literature Review on Distributed Database Query Processing ..	10
2.3.2 Relationship between Distributed Database and Query Processing Strategy .....	11
2.4 INTRODUCTION ABOUT GREEN COMPUTING .....	12
2.4.1 Green Computing Approaches .....	12
2.4.2 Green Computing Case Study .....	13
2.5 INTRODUCTION ABOUT ENERGY EFFICIENCY .....	13
2.5.1 Energy Efficiency Benchmark .....	14

CHAPTER 3	.....	15
METHODOLOGY	.....	15
3.1	RESEARCH METHODOLOY .....	15
3.2	PROJECT ACTIVITIES.....	16
3.2.1	Planning .....	16
3.2.2	Design .....	16
3.2.3	Testing.....	20
3.2.4	Simulation.....	20
3.2.5	Computation.....	20
3.3	QUERY PROCESSING STRATEGY .....	21
3.3.1	Data Localization Layer.....	21
3.3.2	Global Optimization Layer .....	22
3.4	TOOLS OR HARDWARE REQUIRED.....	22
3.4.1	Hardware Required .....	22
3.4.2	Software Required .....	23
3.5	PROJECT TIMELINE..... <b>Error! Bookmark not defined.</b>	
3.5.1	Gantt Chart.....	23
3.5.2	Key Milestones .....	24
CHAPTER 4	.....	25
RESULTS AND DISCUSSION	.....	25
4.1	DESIGN OF DATABASE .....	25
4.2	DEVELOPMNT OF DATABASE.....	26
4.3	DATA GENERATION.....	26
4.4	DATA ALLOCATION STRATEGY.....	28
4.4.1	Horizontal Fragmentation .....	29
4.4.2	Complete Replication.....	32
4.5	SIMULATION OF DATA RETRIEVAL PROCESS.....	33
4.5.1	Simulation of Data Retrieval Process Using Horizontal Fragmentation Strategy .....	33
4.5.2	Simulation of Data Retrieval Process Using Complete Replication Strategy.....	33
4.6	SIMULATION RESULT ANALYSIS.....	36
CHAPTER 5	.....	41
CONCLUSIONS AND RECOMMENDATIONS	.....	41

5.1	RELEVANCY TO OBJECTIVE.....	41
5.2	SUGGESTED FUTURE WORK FOR EXPANSION AND CONTINUATION .....	42
REFERENCES	.....	43
APPENDICES		



## LIST OF FIGURES

Figure 2.1 Illustration of Distributed Database System.....	8
Figure 3.1 Prototyping Methodology Development Cycle.....	15
Figure 3.2 System Architecture .....	17
Figure 3.3 Data Relational Model.....	18
Figure 3.4 Simulation Process Design .....	19
Figure 3.5 Gantt Chart .....	23
Figure 3.6 Key Milestones.....	24
Figure 4.1 Distributed Database Design .....	24
Figure 4.2 Sample Data of Table “Patient” .....	25
Figure 4.3 Sample Data of Table “Hospital” .....	26
Figure 4.4 Sample Data of Table “Doctor” .....	26
Figure 4.5 Sample Data of Table “PatientVisit” .....	27
Figure 4.6 Average Power Consumption (watt) to Access Each Table.....	37
Figure 4.7 Power Consumption for Horizontal Fragmentation and Complete Replication ...	37

## LIST OF TABLES

Table 4.1 Power Consumption for Simulation Using Both Query Processing Strategies .....	36
---	----

## LIST OF EQUATIONS

Equation 4.1 Average.....	35
Equation 4.2 Weighted Average .....	35

# CHAPTER 1

## INTRODUCTION

### 1.1 BACKGROUND OF STUDY

When a patient visits a hospital or clinic for the first time, personal information such as name, identification number, age, birth date, housing address and medical information are recorded and stored. Some medical institutions store these data physically by keeping them in hard copy such as paper files while some store the data electronically by entering them into the computer and save in database.

As people nowadays often travel around the country from a place to another so one might fall sick and visit a medical institution at any time anywhere. Thus there is possibility that a patient may visit a medical institution which is not the one he usually visits. Patient information such as blood type and medical history is important for diagnosis purpose especially when there is an emergency. As a result, it is crucial to ensure that the information about a patient is able to be accessed and retrieved at a minimum time among different medical institutions. In other words, the efficiency of the patient data retrieval process leads to higher efficiency in patient diagnosis process.

However, currently in Malaysia, the hospitals either public or private are not sharing their patients' information among each other. Some of the hospitals are still using the old way by storing their patient data in paper files while some store the data in their own database that is not connected to others. This type of data storing method is known as centralized database. Since the importance of accessibility of patient information among hospitals has been highlighted above, it is recommended for these hospitals to implement distributed database to store their patient information at multiple physical locations.

This project intends to find out the most energy-efficient query processing strategy to retrieve medical information from the distributed database in order to enhance the data retrieval process for more efficient and effective diagnosis.

## **1.2 PROBLEM STATEMENT**

### **1.2.1 Problem Identification**

In support to the project topic, several problems have been identified and listed as follow:

1. Inaccessibility of patient information among hospitals which is crucial for diagnosis purpose.
2. Inefficiency in retrieving patient information slows down the diagnosis process.
3. No query processing strategy has been identified as the most energy-efficient way to access data from distributed database.
4. Inefficient query processing strategy leads to high power consumption.
5. High power consumption increases carbon emission to the environment
6. High power consumption increases the operational cost of medical institution.
7. Inefficient resource allocation to access medical data from database.
8. Healthcare industry has not generally applied green computing like other industries.

The problems stated as above will affect the operation of healthcare industry especially the efficiency of diagnosis as patient information is not accessible among different medical institutions. Another important problem is inefficiency of data retrieval from different medical institutions lead to inefficient diagnosis process by slowing down the time taken to diagnose a patient's condition. Besides, up to date, an energy-efficient query processing strategy is required to access the data from distributed database which lowers the power consumption.

### **1.2.2 Significance of Project**

This project aims to help healthcare industry in their data retrieval process so that it takes lesser time to retrieve patient information from other medical institution when needed. Based on the problems stated above, the significance of this project lies between the efficiency of data retrieval process for diagnosis purpose and energy conservation which reduce the adverse effects to the environment.

### **1.3 OBJECTIVES**

This project aims to achieve several goals as follow:

1. To model the data retrieval process of patient information from medical institution at dispersed locations by developing a distributed database.
2. To analyse the power consumption by each different query processing strategies to retrieve data from distributed database.
3. To identify the most energy-efficient query processing strategy to retrieve data stored in distributed database.

### **1.4 SCOPE OF STUDY**

In order to model the data retrieval process of patient information from distributed database, several scopes of study has been identified as follow:

1. To study on distributed database
  - Do research and studies about distributed database and query processing strategy to process data from distributed database
2. Simulation of data retrieval process from distributed database and record the power consumption
  - In order to set up the simulation, two databases will be created at different networks to model the distributed database. Each query processing strategy will be tested to retrieve the data stored at different networks and the power consumption will be computed by a power meter.
3. Analysis of the simulation
  - The query processing strategy that consumes the least amount of power will be identified.

## **1.5 RELEVANCY OF THE PROJECT**

In healthcare industry, the efficiency of data retrieval process and accuracy of data is very vital during patient's diagnosis especially when there is an emergency that any delay will cause fatal. Thus, in need of achieving high efficiency in data retrieval, this project is believed to be highly relevant to healthcare industry. Apart from that, an efficient query processing strategy not only able to increase the performance of data retrieval, it is also able to reduce the energy consumption during the retrieving process. Reduction in power consumption leads to lower operational cost which in turns means that more money can be saved for other purposes such as buy in more medical equipment, upgrade the medical machines and others. On the other hand, conserve the environment is everyone responsibility including all kinds of industries. The implementation of an energy-efficient query processing strategy helps healthcare industry to fulfill their social responsibility to conserve the environment by reducing the energy emission. This project is able to create a win-win situation to both the healthcare institutions as well as our precious nature.

## **1.6 FEASIBILITY OF PROJECT WITHIN THE SCOPE AND TIME FRAME**

In terms of scope and time frame, this project is feasible to be completed at the end of the day. However, there are several concerns that might affect the progress of the project especially on the technology area as stated below.

### **1.6.1 Technical Feasibility**

This project used MySQL as the main programming language to set up the distributed database on the local network and also to be used for data retrieval. The limited knowledge and time constraint made this project more challenging to be completed before the deadline. However, this risk is possible to be reduced to the minimum as there are many open source development tutorials about MySQL such as from books and Internet.

### **1.6.2 Operational Feasibility**

Operational feasibility is a measure of user acceptance about the solution provided whether it works as the user required or not. This is one of the main concerns during the project development. In order to ensure the operational feasibility of the project that the solution provided is realistic and working, an observation and analysis on how the existing healthcare institution store and retrieve their patients' information will be conducted. With more knowledge about how the real operating healthcare institutions manage their data storing and retrieving, it is believed that the project can achieve its operational feasibility.

### **1.6.3 Time Frame Feasibility**

Since the size of this project is small, it is believed that this project is able to be completed within the time frame given which is 28 weeks. In order to achieve the time frame feasibility, a Gantt chart has been developed to plan and monitor the progress of the project starting from initiation until the completion. There are several key milestones that have been identified throughout the development process. One of the key milestones is the simulation of the data retrieval process from distributed database. This phase takes up the longest time frame in this project. With the productivity of the system developer, it is believed that the project is feasible to be finished within the time frame.

## **CHAPTER 2**

### **LITERATURE REVIEW**

#### **2.1 INTRODUCTION**

Before the development of this project starts, it is important to understand the current data management system deployed by healthcare institutions in the country. There are about 150 government hospitals in Malaysia. Based on understanding from an employee who is currently working with a government hospital in East Malaysia, not all of the public hospitals store and manage their data electronically as some are still using the traditional way of managing the information in paper files format. For the hospitals that store their data electronically, they have their own database at their local network which known as centralized database that does not support sharing of data.

This chapter is going to cover several aspects such as, literature review about the data sharing systems available in the market that solves the problem as proposed earlier to enable sharing of data among hospitals, how does distributed database allows accessibility of data among hospitals, relationship between power consumption and query processing strategies as well as how does reduction in power consumption contribute to green computing.

#### **2.2 SOLUTION FOR DATA SHARING AMONG HEALTHCARE INSTITUTION**

The importance of sharing information among the healthcare providers can be seen in an article entitled [1] "*Fiona Caldicott to lead review into sharing of health information*", posted by the Guardian professional on 17<sup>th</sup> of February 2011. According to the Fiona (2011), information about patient needs to be shared among hospital to provide the best care and to promote excellent research. However, it needs to have a balance between

protecting patients' health care information and sharing to improve patient care to ensure the confidentiality of information.

As stated above the importance of data sharing among healthcare providers, now we will look at the solution available in the market that allows data sharing. One of the solutions is Healthcare Data Management (HDM) provided by [2] BridgeHead Software. According to the company's website, HDM solution allows healthcare data to be stored efficiently, fully protected and could be shared among other hospitals, making the data accessible to people that need it for the delivery of quality patient care. BridgeHead's HDM solution provides hospitals the ability to store all their data in one place efficiently and intelligently. On top of that, BridgeHead's HDM solution also enables hospitals to share their clinical data and administrative data among departments or with other hospitals. This can be achieved by having the feature of web service enabled, access control, authentication as well as encryption to protect the data.

According to an article on posted by [3] Marianne Kolbasuk McGee on 8<sup>th</sup> of June 2012, in United States there are eight Health Information Exchanges (HIE) to help the U.S. health organizations to share data in the name of lower costs and better patient care. The eight most established HIE in U.S. includes Indiana Health Information Exchange, New England Health Exchange Network, Michiana Health Information Network, Colorado Regional Health Information Organization, Greater Houston's Health Connect, Health Bridge, Maine Health Info Net and Care Continuity Consortium. The feature that all of the eight HIEs have in common is they provide a platform for the participating hospitals to access and exchange data which is exceptionally helpful during an emergency situation.

The solutions as discussed above are not applicable in Malaysia thus there is an urge to find a way or medium to enable accessibility of data among Malaysian hospitals for better patient care. The next content will discuss about the solution proposed by this project which is distributed database and how can healthcare institutions implement it for data sharing purpose.



### 2.3 INTRODUCTION ABOUT DISTRIBUTED DATABASE

Lately, distributed database have become an important area of information processing and IT experts foresee that their important will continue to grow. According to a research paper entitled [4] “*Distributed Databases Fundamentals and Research*” written by Dr H. Hakimzadeh from Department of Computer and Information Sciences, Indiana University South Bend, a distributed database (DBB) is a collection of multiple, logically interrelated databases distributed over a computer network. In the book entitled [5] “*Distributed Databases Principles & Systems*” by Stefano Ceri and Giuseppe Pelagatti, the definition of distributed database emphasizes two equally important aspects of a distributed database which is distribution and logical correlation. Distribution refers to the fact that the data are not resident at the same site (processor) so that people can distinguish a distributed database from a single centralized database. On the other hand, logical correlation means that the data have some properties which tie them together so that it is distinguishable from a set of local database which are resident at different sites of a computer network.

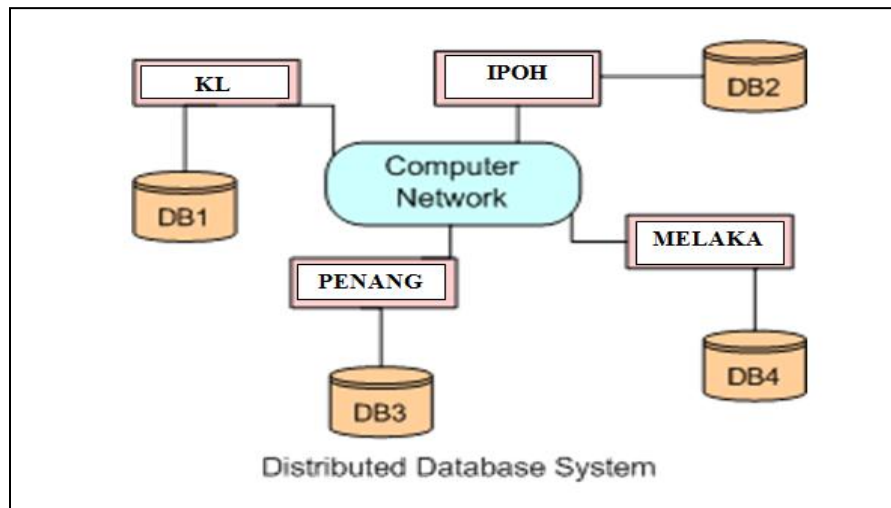


Figure 2.1: Illustration of Distributed Database System

Figure 2.1 illustrates the definition of distributed database in the context of this project. The illustration above assumes that a medical institution has several branches in Malaysia at Kuala Lumpur, Ipoh, Penang and Melaka with each branch has their own database but all connected to the same computer network. In this case, each branch update and maintain

their own database but the data is able to be shared depends on the requirement of the business.

Distributed database works hand in hand with a software known as distributed database management system (DDBMS) to manage the data. Thomas and Carolyn (2010) said that distributed database management system is defined as a software system that permits the management of the distributed database and makes the distribution transparent to users in their book entitled [6] “*Database Systems – A Practical Approach to Design, Implementation and Management*”. Thomas and Carolyn (2010) also said that a distributed database management system (DDBMS) consists of a single logical database that is split into a number of fragments. Each fragment is stored on one or more computers under the control of a separate database management system with the computers connected by a communications network. Each site is capable of independently processing user requests that require access to local data and is also capable of processing data stored on other computers in the network.

According to [7] “*Database System Concepts*” by Korth and Sudarshan, a distributed database management system consists of a single logical database that is split into a number of fragments. Each fragment is stored on one or more computers under the control of a separate database management system with the computers connected by a communication network. In this context, we will discuss about two main types of fragmentation which is horizontal and vertical fragmentation.

Korth and Sudarshan stated that horizontal fragmentation consists of a subset of the tuples of a relation by grouping together the tuples in a relation that are collectively used by important transactions. A horizontal fragment is produced by specifying a predicate that performs a restriction on the tuples in the relation. On the other hand, vertical fragmentation is defined as a fragment that consists of a subset of the attributes of a relation by grouping together the attributes in a relation that are used jointly by important transactions. The difference between horizontal and vertical fragmentation is that content of the subset where horizontal fragment contains tuples of relation while vertical fragment contains attributes of relation.

Fragmentation is commonly used in distributed database to partition the database into disjoint fragments with each fragment assigned to one site. Since data is stored close to where it is most frequently used, the efficiency to retrieve the data is higher than storing in centralized database. Besides that, fragmentation provides higher data security as data that are not required by local applications will not be available to unauthorized users.

In this context, the project focused on fragmentation and query optimization strategy in distributed database to achieve energy efficiency in query processing which will be further discussed in methodology later.

### **2.3.1 Literature Review on Distributed Database Query Processing**

According to [8] "*Query Processing*" by Elmasri and Navathe, query processing involves parsing, validating, optimizing and executing a query as shown in Appendix 1. The aims of query processing are to transform a query written in a high-level language typically SQL into a correct and efficient execution strategy expressed in a low-level language, and to execute the strategy to retrieve the required data. One of the important aspects of query processing is query optimization which involves the activity of choosing an efficient execution strategy for processing a query.

A distributed database offers the advantages of increased data reliability and faster access to data compared to centralized database. However, one of the important problems is the efficient processing in a distributed system. Alan R. Hevner and S.Bing Yao in [9] "*Query Processing in Distributed Database System*" stated that, accessing data that are stored at separate computers differs in two important ways from accessing data from a centralized computer. The necessary transmission of data over communication lines introduces substantial time delays. The database management system must consider these facts and derive an effective arrangement of local data processing and transmission in order to process distributed queries. This arrangement of data processing and data transmission is known as a distribution strategy for a query.

Hevner and Yao set the main objectives of their paper to achieve minimization of response time and total time through optimization of distribution strategies. According to them, a

query optimization algorithm is an algorithm that derives a distribution strategy for a given query. Hevner and Yao have discussed about several optimization algorithms in their research paper such as algorithm PARALLEL, SERIAL strategy and general algorithm. Each of these strategies has their own strengths and weaknesses in response time minimization.

### **2.3.2 Relationship between Distributed Database and Query Processing Strategy**

Michael L. Rupley stated in his journal entitled [10] *“Introduction to Query Processing and Optimization”*, a query is a vehicle for instructing a database management system (DBMS) to update or retrieve specific data to from the physically stored medium. According to [11] *“Wiktionary”*, query processing is defined as the process of executing a specific set of instructions for extracting particular data from database. We can see that there is a significant relationship between query processing and database, it applies to any types of database either centralized database or distributed database.

Pankti Doshi and Vijay Raisinghani pointed out their research paper entitled [12] *“Review of Dynamic Query Optimization Strategies in Distributed Database”*, the performance of a distributed database depends on how fast and efficiently data can be retrieved by query from multiple sites. Faster retrieval of data in a distributed database system is a complex problem since multiple sites are involved. Several factors impact the performance of distributed query processing. These factors are selection of appropriate site (when same data is replicated at multiple sites), order of operation (such as select, project and join) and selection of join method (such as semi join, natural join, equi join etc). Due to the large number of factors involved, there could be multiple executions plans for a single query. Each plan is associated with a cost and the objective of a distributed query optimizer is to find a plan with the lowest possible cost. The execution cost is expressed as a sum of I/O, CPU and communication cost.

According to Pankti Doshi and Vijay Raisinghani as well, query processing in a distributed database requires transfer of data from one computer to another through a communication network. Query at a given site might require data from remote sites. The complexity and

cost increases with the increasing number of relations in the query. Thus, a query optimization is very much important in order to achieve energy efficiency as well as reduce operational cost as targeted in this project.

In conclusion, query processing optimization plays an important role not only to reduce the energy emissions which adversely affect the environment but also helps to reduce the operational cost of a business.

## **2.4 INTRODUCTION ABOUT GREEN COMPUTING**

Green computing is known as green IT or ICT sustainability which refers to environmentally sustainable computing of IT. In the article [13] “*Harnessing Green IT: Principles and Practices*” by San Murugesan, the field of green computing is defined as the study of practice of designing, manufacturing, using and disposing of computers, servers and associated subsystem such as monitors, printers, storage devices and networking and communication system efficiently and effectively with minimal or no impact on the environment. In spite of the long definition of green computing, green computing can be generally applied in anywhere. According to [14] “*Greenelectronics*”, green computing is very much related to movements like reducing the use of environmentally hazardous materials like CFCs, promoting the use of recyclable materials, minimizing the use of non-biodegradable components and encouraging the use of sustainable resources.

### **2.4.1 Green Computing Approaches**

The rapid growth of innovations in technology brings forth various ways on how green computing has great benefits not only to the environment, but also to the consumer, business and country. Rajguru P.V, Nayak S.K and More D.S in their research paper entitled [15] “*Solution for Green Computing*” stated that there are four main categories of green computing approaches. One of them is green use which reduces the energy consumption of computers and other information systems as well as using them in an environmentally sound manner. The other category is green disposal by refurbishing and reusing old computers then properly recycle unwanted computers or other electronic

equipment. Green design refers to designing energy-efficient and environmentally sound components, computers, servers, cooling equipment and data centers. Lastly, green manufacturing encourages manufacture of electronic components, computers and other associated subsystems with minimal impact on the environment.

This project intends to focus on applying green computing in query processing for distributed data, in a simpler way it means green computing in data center. As discussed before about the four types of green computing approach, application of green computing in data centers falls in one of the categories called green design. According to [16] “*Wikipedia*”, data center facilities are heavy consumers of energy which account for about 1.1% to 1.5 % of the world’s total energy used in 2010. The U.S Department of Energy estimates that data center facilities consume up to 100 to 200 times more energy than standard office buildings. Energy efficient data center design should help to better utilize a data center’s space as well as increase performance and efficiency.

#### **2.4.2 Green Computing Case Study**

Kunaciilan Nallappan, the product marketing manager of Citrix Systems commented on [17] “The Star”, Monday July 21, 2008, Citrix has conducted a green IT poll with the attendees of the Citrix App Delivery Conference 2008 in Malaysia, Singapore. The findings revealed that 51% of Malaysian companies have taken green computing into consideration in their IT initiatives in the past year. The US-based firm said that nearly all of the Malaysian respondents (91%) viewed virtualization as a key way for their company to go green in their IT initiatives while 70% viewed green IT as both a cost saving measure and part of their corporate social responsibility.

### **2.5 INTRODUCTION ABOUT ENERGY EFFICIENCY**

According to the research paper [18] “*Rethinking Query Processing for Energy Efficiency: Slowing Down to Win the Race*” by Willia Lang, Ramakrishnan and Jignesh M.Patel, energy management has become a critical aspect in the design and operation of database management systems. The emergence of this new paradigm as an optimization goal is driven by several factors. One of them is because of the tremendous amounts of energy

consumed by a server which is 61B kilowatt-hours in 2006 and doubling by 2011. In addition, the energy component of the total cost of ownership for servers is high and growing rapidly. Besides that, some typical servers are over provisioned to meet peak demands; as a result, they are idle or underutilized most of the time. When servers are idle or nearly idle, they tend to consume energy that is disproportional to their utilization which is more than 50% of its peak power. With these rising energy costs and energy-inefficient server deployments, it is clear that there is a need to consider energy efficiency as a first class operational goal.

### **2.5.1 Energy Efficiency Benchmark**

As discussed earlier, green computing emphasizes on the importance of energy consumption to achieve efficiency in business processes. Suzanne Rivoire, Mehul A. Shah and others pointed out in their paper entitled [19] *“Joule Sort: A Balanced Energy-Efficiency Benchmark”* that an energy-efficiency benchmark known as Joule Sort is proposed to drive the design of energy-efficient system. Joule Sort incorporates total energy which is a combination of power consumption and performance. Joule Sort is an I/O-centric benchmark that measures the energy efficiency of system at peak use by allowing comparison of energy efficiency of a variety of disparate system configurations.

According to Dimitris Tsirogiannis, Stavros Harizopoulos and Mehul A. Shah in [20] *“Analyzing the Energy Efficiency of a Database Server”*, energy efficiency is defined as the ratio of useful work done to the energy used which is the same as the ratio of performance to power (Energy Efficiency = Work Done/Energy). As database software is rich in tunable parameters from system level constants to query planning and execution, these parameters can potentially affect the energy efficiency. Besides that, the energy efficiency of the database also affects its performance in various ways such as access methods, compressions, join algorithms as well as complex queries and join orderings.

# CHAPTER 3

## METHODOLOGY

### 3.1 RESEARCH METHODOLODY

In order to complete this project, a lot of researches have been conducted to collect information and knowledge about application of green computing in distributed database data retrieval for medical data. There are various types of source for the research which includes books, journal and online material.

The methodology that is used to develop this project is prototyping methodology. The reason being prototyping methodology is selected is because of the time constraint to complete this project which is less than seven months. Besides that, prototyping methodology allows continuous improvement throughout the development process which increases the quality of the deliverables. Since this is a research-based project which focuses on the research elements of green computing, prototyping methodology is chosen as it allows incomplete versions of the software program being developed.

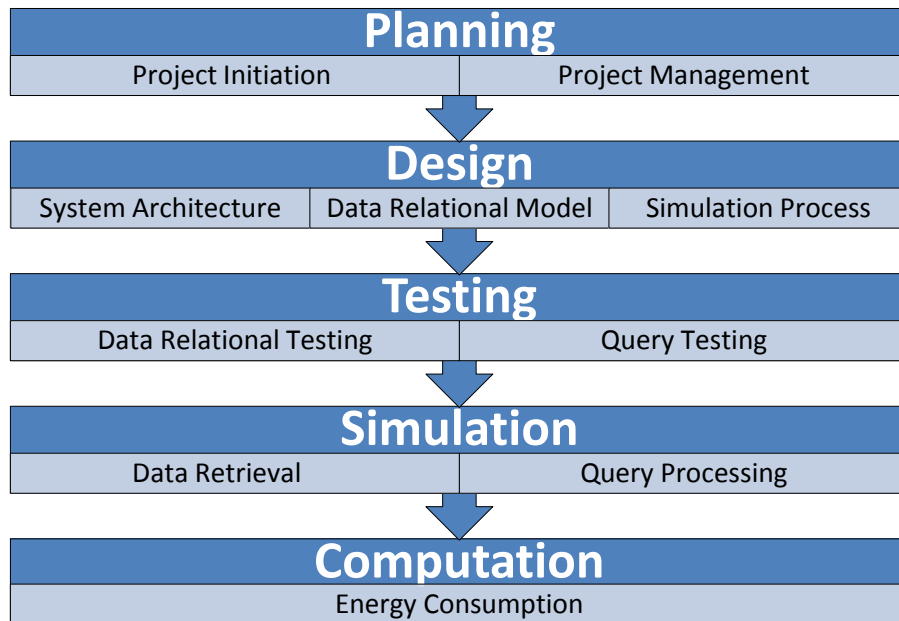


Figure 3.1: Prototyping Methodology Development Cycle



## **3.2 PROJECT ACTIVITIES**

Under prototyping methodology, the project can be divided into four main stages as below:

- Planning
- Design
- Testing
- Simulation
- Computation

### **3.2.1 Planning**

During planning phase, it is important to understand the deliverables of the project and the time given to completion. It mainly involves two processes that are project initiation and project management.

- i. Project initiation
  - Gather requirement and information from user about the project
  - Analyse feasibility of the project to the healthcare industry on medical data
- ii. Project management
  - Create work plan and Gantt chart
  - Create flow chart for project development process

### **3.2.2 Design**

At this stage, the design of the simulation is the main element to the process of retrieving medical data from distributed databases. Several processes are involved during the design phase such as:

- Include the specified requirement in the prototype
- Develop the model for simulation purpose (eg: relationship between extracted tables in UML diagram)
- Build the prototype based on the user requirement
- Develop distributed database system architecture
- Develop simulation process

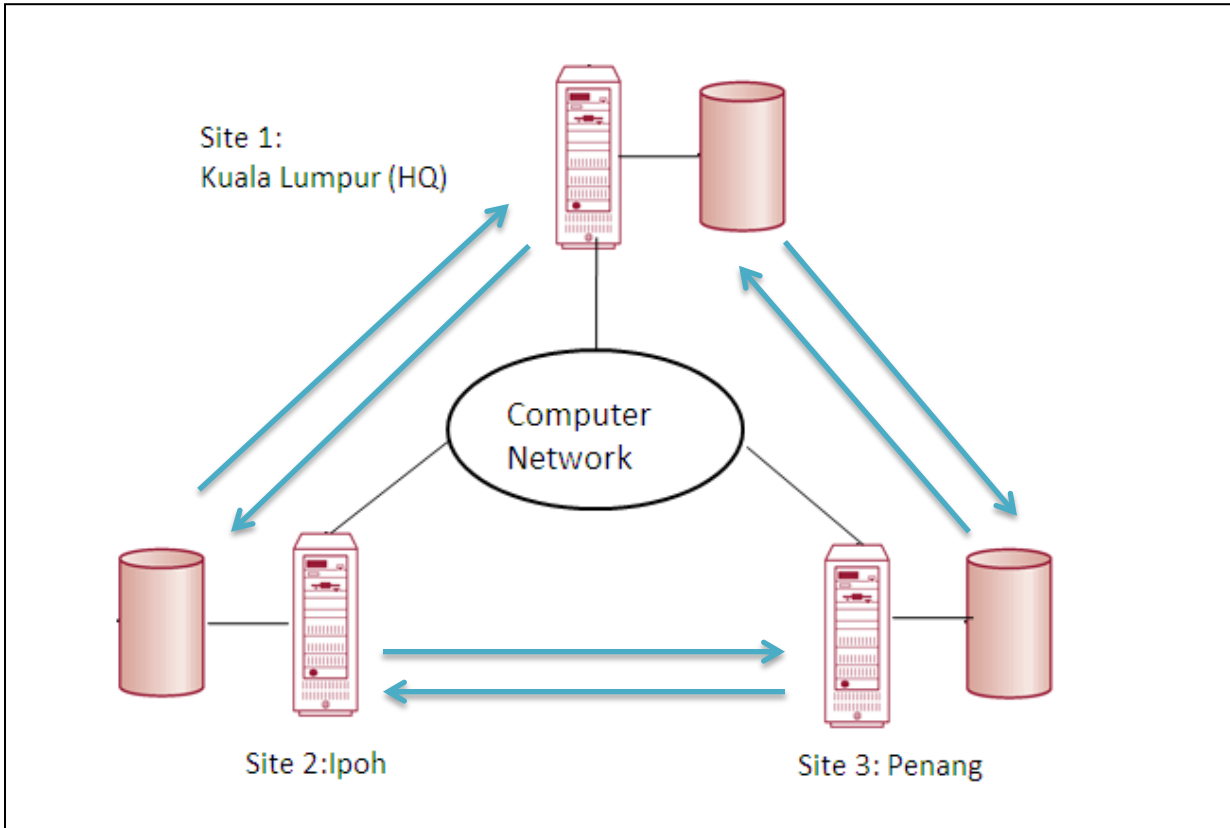


Figure 3.2 System Architecture

Figure 3.2 above shows the system architecture of the distributed database that will be set up on a local computer network for data retrieval simulation. There are three databases involved in this set up, each labelled with the location of the hospitals starting from DB1 to DB3. Each of these databases has different IP address which indicates the dispersion of the database at different locations. Each database contains the same tables that will be discussed in next content.

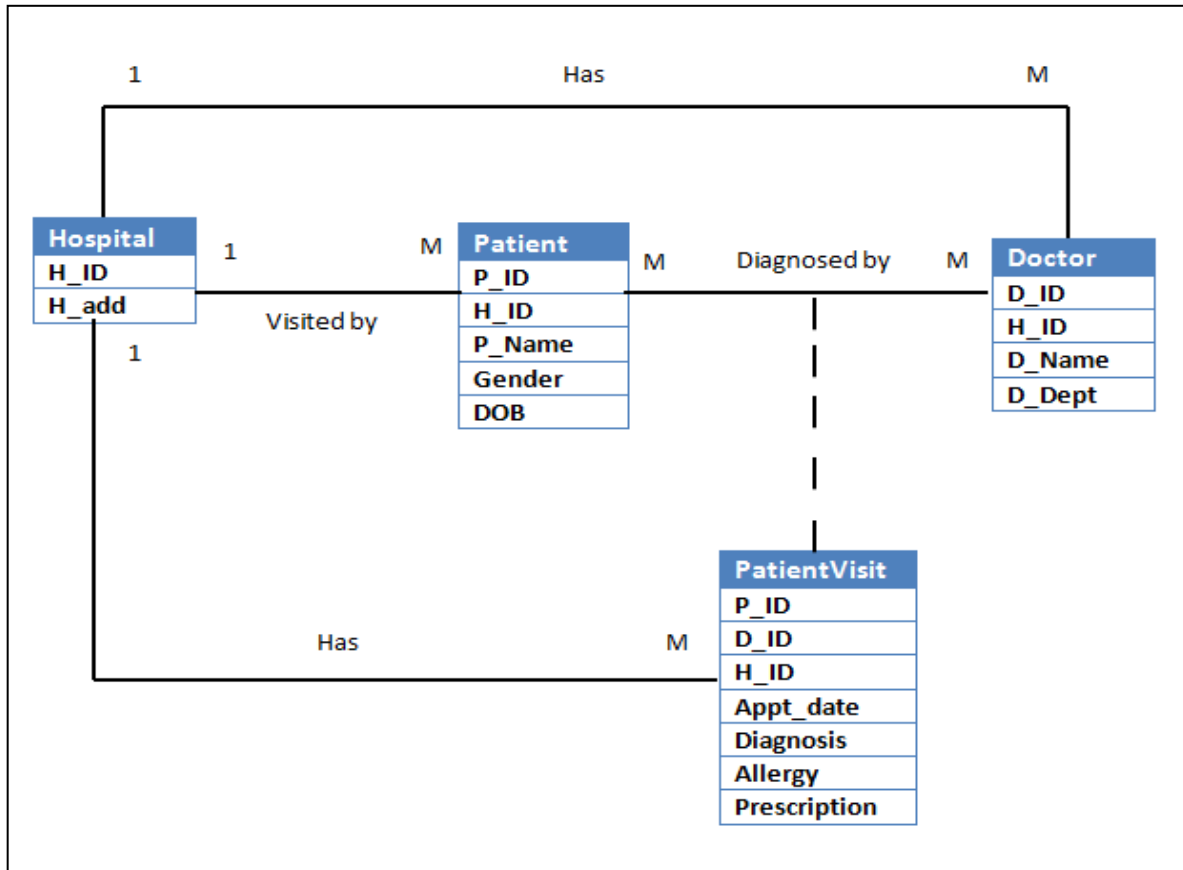


Figure 3.3 Data Relational Model

Figure 3.3 shows the relationship between different tables in the data retrieval process. There are four tables involved in the process such as ‘Hospital’, ‘Doctor’, ‘Patient’, and ‘PatientVisit’. The details of the fields in each table are shown below:

- 1) Hospital (**H\_ID**, H\_add)
- 2) Doctor (**D\_ID**, **H\_ID**, D\_Name, D\_Dept)
- 3) Patient (**P\_ID**, **H\_ID**, P\_Name, Gender, DOB, Hosp\_add)
- 4) PatientVisit (**P\_ID**, **D\_ID**, **H\_ID**, Appt\_date, Diagnosis, Allergy, Prescription)

These tables are located at each site in their own database and are able to be retrieved from one site to another site. The tables are logically interrelated with each other and their relationships are depicted as below:

- i. A ‘Hospital’ is visited by many ‘Patient’.
- ii. Many ‘Patient’ are diagnosed many ‘Doctor’.

- iii. A 'Hospital' has many 'Doctor'.
- iv. During diagnosis, a 'Doctor' has access to 'PatientVisit'.
- v. A 'Hospital' has many 'PatientVisit'.

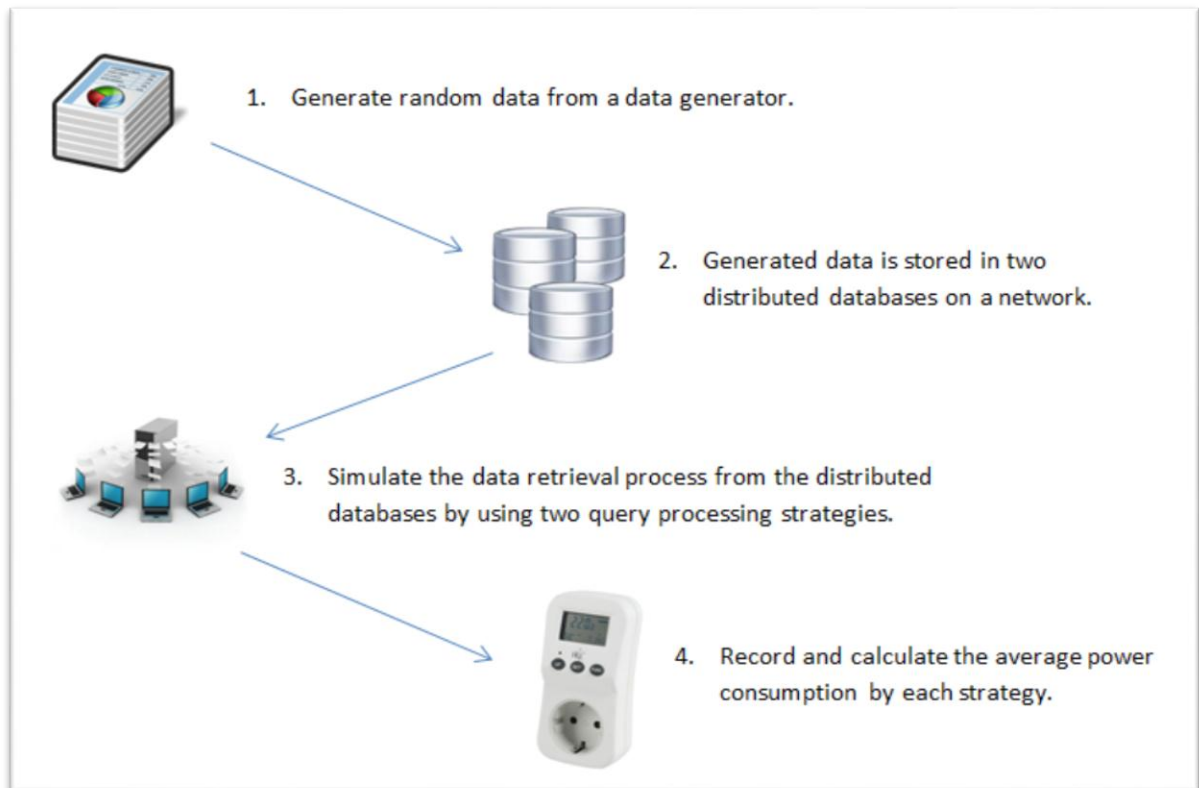


Figure 3.4 Simulation Process Design

Figure 3.4 is the design of the simulation process which involves four main steps as below:

- i. First of all, generates random medical data from a data generator.
- ii. Secondly, store the generated medical data in the distributed database that has been set up.
- iii. Thirdly, conduct the simulation of the data retrieval process from each database by using different query processing strategies.
- iv. Compute the energy consumption by using power consumption calculator.

### **3.2.3 Testing**

The third stage in this project development cycle is testing. Testing comes before the simulation process in this project as we need to ensure the reliability of the data in the database and also the accuracy of the query before the simulation of the data retrieval process are carried out. There are two testing involved in this phase such as:

- Data relational testing to ensure the logical relationship between different tables in the database.
- Query testing to ensure the accuracy of the query before execution during the simulation process.

### **3.2.4 Simulation**

This is the stage where the simulation will be carried out to identify the most energy-efficient query processing strategy to retrieve medical data from distributed database. There are two processes happened at this stage such as:

- Simulation of data retrieval process by requesting for medical data from the distributed database
- Simulation of query processing by using different query processing strategies to retrieve the requested medical data from the distributed database.
- Computation of power consumption by each query processing strategy.

### **3.2.5 Computation**

During this phase, there is only one process involved which is the calculation of power consumption by each query processing strategy during the simulation process. The power consumption calculated for each data retrieval process is based on the result from power meter which reads the power consumption during the simulation. Based on the calculation which averages all the readings from power meter, the query processing strategy with the lowest power consumption will be identified as the most energy-efficient query strategy.

### 3.3 QUERY PROCESSING STRATEGY

This project focuses on an important aspect of query processing which is query optimization. Query optimization aims to achieve minimum resource usage by reducing the total execution time of the query. Generally there are two techniques for query optimization where the first one uses heuristic rules to order the operations in a query and the other one compares different query processing strategies based on the relative costs then selects the one with minimum resource usage. Decision about which techniques to use depends on several factors such as the layer at which data is located, types of fragmentation being used and others. In this context, only two layers of the distributed database architecture will be discussed which are data localization and global optimization layers. Below are some of the distributed database query optimization strategies that can be used for these two layers:

#### 3.3.1 Data Localization Layer

##### a) Reduction techniques

- generates simpler and optimized query
  - Reduction for primary horizontal fragmentation
    - Reduction with Selection operation (eliminates operation with empty intermediate relation)
    - Reduction for Join operation (allows Join operation to be commuted with Union operation)
  - Reduction for vertical fragmentation
    - Removes vertical fragment that have no attributes in common with the projection attributes except the relation key
  - Reduction for derived horizontal fragmentation
    - Allows Join and Union operations to be commuted with the knowledge that some of the partial joins are redundant

##### b) Distributed Joins

- Replaces Joins with combinations of Semijoins to reduce the size of operand relation

- Reduces processing times by reducing the amount of data transferred between databases

### 3.3.2 Global Optimization Layer

#### a) R\* Algorithm

- Uses static query optimization
- Optimization algorithm is based on an exhaustive search of all join orderings, join methods and the access paths for each relation

#### b) SDD-1 Algorithm

- Modifies Semijoin operator to reduce the cardinality of the join operands
- Concentrates on minimizing size of the message

As discussed before, the result of the simulation is the most energy-efficient query processing strategy to retrieve medical data from distributed database. Below are the equations that will be used to compute the average power consumption by each strategy during the simulation:

i. 
$$\text{Average} = \frac{\sum A}{N}$$
 where A: power consumption for each query  
N: number of query execution

ii. 
$$\text{Weighted Average} = 100\% \times \frac{\sum B}{\text{Total of B}}$$
 where B: Total in average of all tables

## 3.4 TOOLS OR HARDWARE REQUIRED

### 3.4.1 Hardware required

- Energy consumption calculator
  - To calculate the power consumption by each query processing strategy in every simulation
- Personal computer

- To execute query in order to retrieve data from each database

### 3.4.2 Software required

i. PuTTY

- To connect to the database

ii. Microsoft Visual Studio

- To simulate the process of remote access to data that stored in other database by propagating the data request to all the available servers

## 3.5 PROJECT TIMELINE

### 3.5.1 Gantt Chart

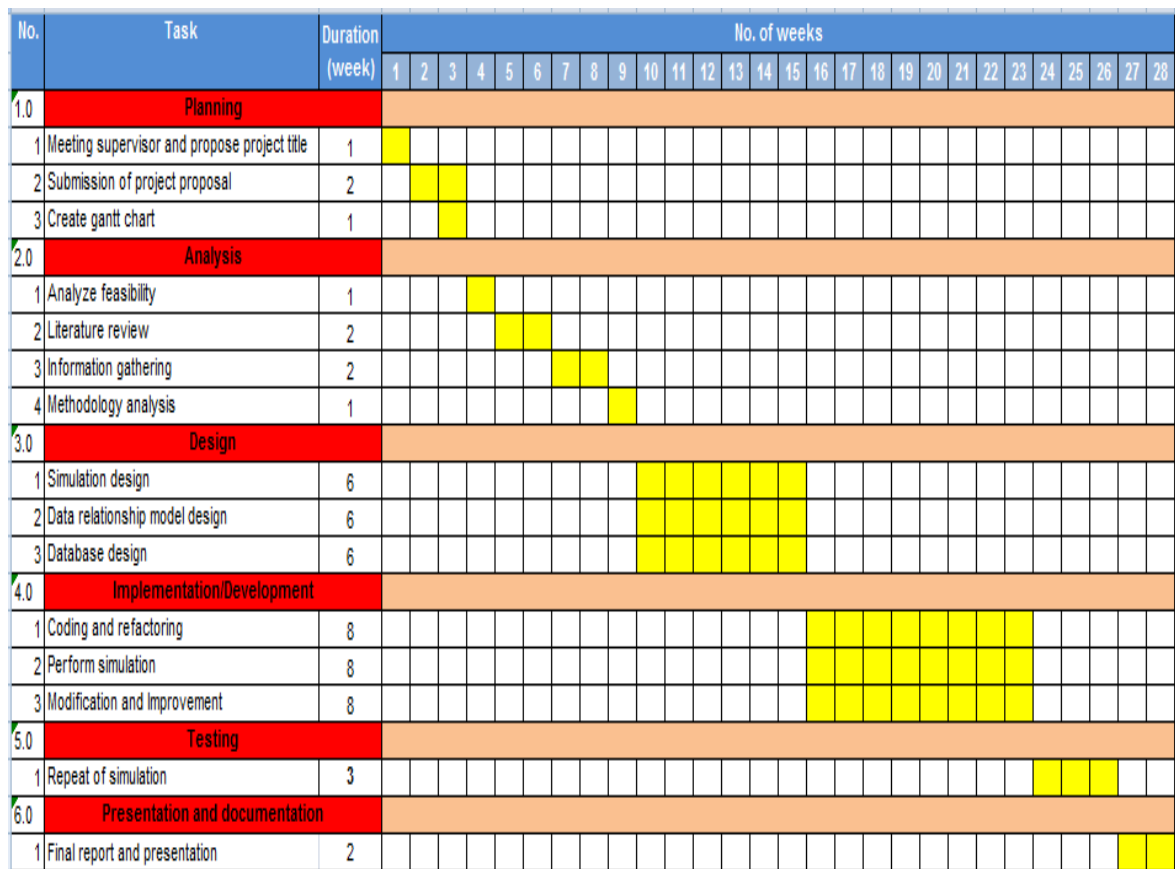


Figure 3.5 Gantt Chart

Referring to Figure 3.5, this project took about 28 weeks to complete which involved four main stages as discussed earlier.



### 3.5.2 KEY MILESTONE

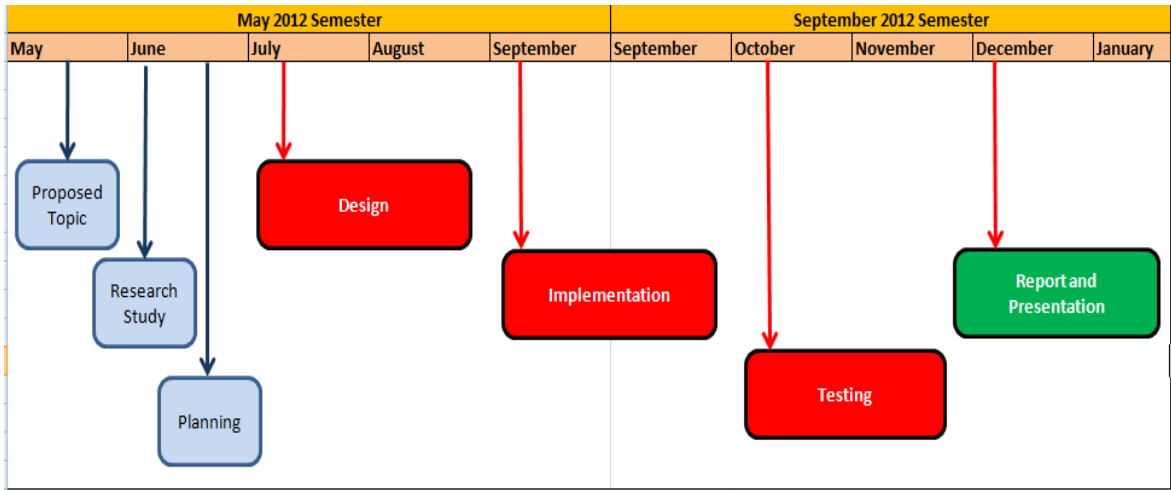


Figure 3.6 Key Milestones

Figure 3.6 shows the key milestones of the project during the development process. The three key milestones are design phase, implementation phase and testing phase. Design phase is critical as it involved the development of a lot of designs as shown above such as system architecture design, simulation design and data relational model. The failure to develop appropriate and accurate design may bring down the success of the project as the project development is based on the designs.

Implementation phase is where the simulation is conducted. This is the most critical phase throughout the project development. At this stage, the simulation of medical data retrieval process is carried out for several times while each time using different query processing strategies to find out which is the most energy-efficient way. This process also involved the calculation of the energy consumption by each query processing strategy to retrieve the medical data from database. Testing phase involved the replication of the simulation process to ensure the accuracy of the computation of the energy consumption by each query processing strategies.

## CHAPTER 4

### RESULTS AND DISCUSSION

#### 4.1 DESIGN OF DATABASE

There are two strategies to design a distributed database such as top-down approach and bottom-up approach. In this project, the second approach is used for the design of the database which is bottom-up approach. Bottom-up approach is used to connect the dispersed database at different locations to solve common tasks; in this context the common task is to retrieve patients' information from hospitals located at different places. Below is the design of the database for the simulation.

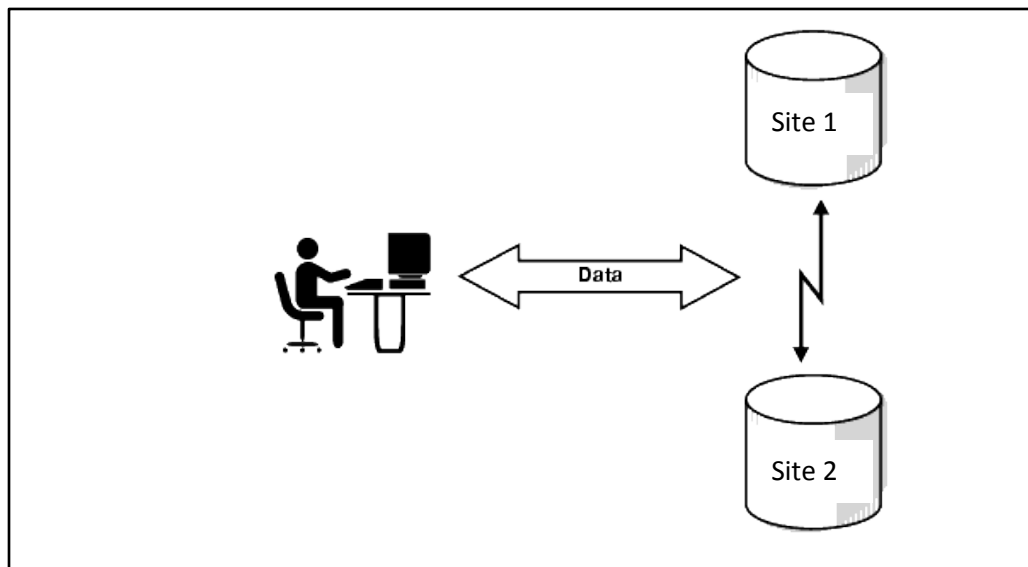


Figure 4.1 Distributed Database Design

The Figure 4.1 above illustrates the interaction between the user request for data and two connected distributed databases. The databases are named as "Site 1" and "Site 2" to represent the different physical locations on the network for example, the actual location of "Site 1" is Kuala Lumpur while the actual location of "Site 2" is Ipoh.

## 4.2 DEVELOPMENT OF DATABASE

The design of distributed database as illustrated in Figure 4.1 is developed by an administrator of UTP local network where site 1 on IP address of 192.168.113.251 and site 2 located on IP address of 192.168.113.155. The software that used to connect to the database on the local network is PuTTY which uses Linux as the programming language and supports MySQL.

## 4.3 DATA GENERATION

As shown in Figure 3.3 in Methodology, there are four tables involved in the relationship between each data such as patient personal information, doctor information, hospital information and patient medical information. For each of these tables, the data are being generated by using an online random data generator, <http://www.generatedata.com>. There are a few assumptions made during the generation of test data as stated below:

- 1) The generated data are dummy data that were generated on a random basis; thus the types of prescriptions and diseases are those that are more common.
- 2) Assume that the data generated are real and true.

Figure 4.2 shows the sample data generated for table “Patient” with 1000 rows of data.

1	P_ID	H_ID	P_Name	Gender	DOB
2	P1000	Site 2	Carissa S. French	Female	2002-12-18
3	P1003	Site 2	Kitra S. Petty	Female	2012-02-26
4	P1004	Site 1	Astra G. Britt	Female	1939-01-21
5	P1005	Site 1	Aubrey P. Baker	Female	2008-05-25
6	P1007	Site 2	Chastity O. Jensen	Female	1966-09-13
7	P1008	Site 1	Eve O. Franklin	Male	1939-11-04
8	P1009	Site 1	Shelly L. Frost	Male	1952-08-11
9	P1010	Site 1	Hiram I. Wong	Male	1953-07-04
10	P1011	Site 1	Fleur B. Duke	Female	1914-03-23
11	P1015	Site 1	Kevyn H. Washington	Female	1965-04-26
12	P1016	Site 2	Wendy I. Ferrell	Male	1981-04-13
13	P1018	Site 1	Melvin X. Hardin	Female	1971-01-05
14	P1019	Site 1	Macon X. Nunez	Male	1954-03-12
15	P1022	Site 1	Miranda X. Melton	Female	1926-12-05
16	P1023	Site 2	Blaine T. Harvey	Female	1959-06-17
17	P1026	Site 2	Josephine I. Fry	Male	1945-07-03
18	P1027	Site 2	Madison G. Beach	Female	1999-12-08

Figure 4.2 Sample Data of Table “Patient”

Figure 4.3 shows the sample data generated for table “Hospital” with 2 rows of data.

1	H_ID	H_add
2	Site 1	Kuala Lumpur
3	Site 2	Ipoh

Figure 4.3 Sample Data of Table “Hospital”

Figure 4.4 shows the sample data generated for table “Doctor” with 200 rows of data.

1	D_ID	H_ID	D_Name	D_Dept
2	D102	Site 2	Cheryl Khor Seng Yin	Nephrology
3	D104	Site 2	Yeap Jing Kuan	Oncology
4	D105	Site 2	Kayathiri A/P Kumar	Rheumatology
5	D106	Site 1	Brielle U. Ortiz	Anaesthetics
6	D108	Site 1	Lois K. Raymond	Haematology
7	D110	Site 1	Lo Kah Huat	Cardiology
8	D111	Site 1	Nabila Faeqa binti Muhammad	Microbiology
9	D114	Site 1	Harihalal A/L Rajesh	Diagnostic Imaging
10	D116	Site 1	Serena Tan Leng Yee	Microbiology
11	D118	Site 2	Kavitta Kaur	Accident and Emergency
12	D120	Site 1	Abdul Arif bin Afiq	Ophthalmology
13	D121	Site 2	Yasir U. Heath	Oncology
14	D122	Site 2	Steven V. Turner	Diagnostic Imaging
15	D125	Site 2	Robin J. Rasmussen	Neurology
16	D130	Site 2	Roanna U. Rose	Cardiology
17	D131	Site 2	Chancellor E. Riley	Microbiology
18	D133	Site 2	Renee O. Baldwin	Ear nose and throat
19	D134	Site 2	Shafira L. Vargas	Anaesthetics
20	D135	Site 2	Gwendolyn F. Pollard	Microbiology

Figure 4.4 Sample Data of Table “Doctor”

Figure 4.5 shows the sample data generated for table “PatientVisit” with 1000 rows of data.

1	P_ID	D_ID	H_ID	Appt_Date	Diagnosis	Allergies	Prescription
2	P1001	D101	Site 2	2012-07-30	Cardiovascular	Skin allergies, Medication allergies	Antibiotics, Antivirals, Pain Relief, Prenatal Vitamins
3	P1002	D102	Site 2	2013-09-05	Asthma	Skin allergies	Contraceptives, Prenatal Vitamins
4	P1004	D104	Site 1	2003-08-14	SARS, Diabetes, Prostate Caner, Depression, Bacterial Vaginosis	Respiratory allergies	Insulin, Contraceptives, Prenatal Vitamins, Antifungals, Growth Hormone, Antivirals
5	P1005	D105	Site 1	2009-02-04	Depression ,Diabetes	None, Anaphylaxis	Antibiotics, Antifungals, Insulin, Miscellaneous
6	P1006	D106	Site 2	2011-10-24	H1N1, Bronchitis, Migraine, Arthritis, Influenza	Environmental allergies	Miscellaneous, Antifungals
7	P1008	D108	Site 1	2012-05-04	Asthma, Depression, Osteoporosis, Arthritis	None	Prenatal Vitamins, Antifungals, Insulin, Contraceptives
8	P1009	D109	Site 1	2011-09-24	Bacterial Vaginosis, Osteoporosis, Alzheimer's, H1N1	Anaphylaxis	Antibiotics, Insulin, Growth Hormone, Pain Relief, Antifungals, Prenatal Vitamins
9	P1010	D110	Site 1	2002-11-14	Asthma, Cervical caner, Osteoporosis	Respiratory allergies	Pain Relief, Contraceptives, Antifungals, Miscellaneous, Prenatal Vitamins, Antivirals, Growth Hormone
10	P1011	D111	Site 1	2005-01-21	H1N1, Osteoporosis, Cervical caner, Hepatitis, Alzheimer's	None	Growth Hormone, Antifungals, Insulin, Prenatal Vitamins, Pain Relief, Miscellaneous, Antibiotics
11	P1012	D112	Site 2	2011-04-23	Asthma, Breast Cancer, Bronchitis	Skin allergies, Medication allergies	Pain Relief, Prenatal Vitamins

Figure 4.5 Sample Data of Table “PatientVisit”

#### 4.4 DATA ALLOCATION STRATEGY

After the data has been generated, the data are being entered into the database by using MySQL “Insert” statement. Next, the data is allocated to the database based on two strategies which is fragmentation and replication. Fragmentation and replication are used to create two scenarios which are different in the way of how client access the data. Both strategies used will then be compared in terms of the difference in power consumption during data retrieval process. The details of the scenarios are showed below.

## Scenarios:

- i. Fragmentation (Horizontal fragmentation)
  - To model remote access of data from client (Site 2) to server (Site 1)
  - Site 1 holds fragments that are created based on the location of data (Site 1 or Site 2)
  
- ii. Replication (Complete replication)
  - To model local access of data at each site (Site 1 and Site 2)
  - Each site holds a complete copy of all information another site e.g., Site 1 has all the information stored in Site 2 and vice versa.

### **4.4.1 Horizontal Fragmentation**

There are four types of fragmentation to fragment data in distributed database while in this project horizontal fragmentation is used to fragment the data. Horizontal fragmentation is used as it groups together the tuples in a relation that are used by important transaction which is retrieval transaction in this context. A horizontal fragment is produced by using the Selection operation to group together the tuples that have common property. The common characteristics in the healthcare information generated above are all of them have a field named "H\_ID" which determine the location of the data either Site 1 or Site 2. In this simulation, Site 2 acts as the client side who request for data while Site 1 is the server who reply the request by returning the requested data. Thus, the fragments are created at site 1 to hold all the information. There are two main types of fragments for data which are fragmented based on the location of the data either Site 1 or Site 2 as depicted below. In total, there will be six fragments for the three tables that have been created in the database. The details of the fragmentation are showed below.

- i. 6 fragments
  - Fragment 1: Table "Patient" fragmented by hospital address ( $S_1$ )
  - Fragment 2: Table "Patient" fragmented by hospital address ( $S_2$ )
  - Fragment 3: Table "Doctor" fragmented by hospital address ( $S_1$ )

Fragment 4: Table “Doctor” fragmented by hospital address ( $S_2$ )

Fragment 5: Table “PatientVisit” fragmented by hospital address ( $S_1$ )

Fragment 6: Table “PatientVisit” fragmented by hospital address ( $S_2$ )

ii. 1 transaction

Transaction 1: Access only

iii. 2 sites

$S_1$ : Site 1 (Server)

$S_2$ : Site 2 (Client)

Below are SQL statements to create the fragments based on the location of the data ( $S_1$  or  $S_2$ ).

Fragment 1: Table “Patient” fragmented by hospital address ( $S_1$ )

$F_1: \sigma_{H\_ID = 'Site 1'}(PATIENT)$

Fragment 2: Table “Patient” fragmented by hospital address ( $S_2$ )

$F_2: \sigma_{H\_ID = 'Site 2'}(PATIENT)$

Fragment 3: Table “Doctor” fragmented by hospital address ( $S_1$ )

$F_3: \sigma_{H\_ID = 'Site 1'}(DOCTOR)$

Fragment 4: Table “Doctor” fragmented by hospital address ( $S_2$ )

$F_4: \sigma_{H\_ID = 'Site 2'}(DOCTOR)$

Fragment 5: Table “PatientVisit” fragmented by hospital address ( $S_1$ )

$F_5: \sigma_{H\_ID = 'Site 1'}(PATIENTVISIT)$

Fragment 6: Table “PatientVisit” fragmented by hospital address ( $S_2$ )

$F_6: \sigma_{H\_ID = 'Site 2'}(PATIENTVISIT)$

Sample data of fragment stored in Site 1 for each tables:

- i. Fragment 1: Table “Patient”

P_ID	H_ID	P_Name	Gender	DOB
P1004	Site 1	Astra G. Britt	Female	1939-01-21
P1005	Site 1	Aubrey P. Baker	Female	2008-05-25

- ii. Fragment 3: Table “Doctor”

D_ID	H_ID	D_Name	D_Dept
D106	Site 1	Brielle U. Ortiz	Anaesthetics
D108	Site 1	Lois K. Raymond	Haematology

- iii. Fragment 5: Table “PatientVisit”

P_ID	D_ID	H_ID	Appt_Date	Diagnosis	Allergies	Prescription
P1004	D104	Site 1	2003-08-14	SARS, Diabetes, Prostate Caner, Depression, Bacterial Vaginosi	Respiratory allergies	Insulin, Contraceptives, Prenatal Vitamins, Antifungals, Growth Hormone, Antivirals
P1005	D105	Site 1	2009-02-04	Depression ,Diabetes	None, Anaphylaxis	Antibiotics, Antifungals, Insulin, Miscellaneous



Sample data of fragment stored in Site 2 for each tables:

- i. Fragment 2: Table “Patient”

P_ID	H_ID	P_Name	Gender	DOB
P1000	Site 2	Carissa S. French	Female	2002-12-18
P1003	Site 2	Kitra S. Petty	Female	2012-02-26

- ii. Fragment 4: Table “Doctor”

D_ID	H_ID	D_Name	D_Dept
D102	Site 2	Cheryl Khor Seng Yin	Nephrology
D104	Site 2	Yeap Jing Kuan	Oncology

- iii. Fragment 6: Table “PatientVisit”

P_ID	D_ID	H_ID	Appt_Date	Diagnosis	Allergies	Prescription
P1001	D101	Site 2	2012-07-30	Cardiovascular	Skin allergies, Medication allergies	Antibiotics, Antivirals, Pain Relief, Prenatal Vitamins
P1002	D102	Site 2	2013-09-05	Asthma	Skin allergies	Contraceptives, Prenatal Vitamins

#### 4.4.2 Complete Replication

Another strategy that is used to model the local access of patients’ information at client side (Site 2) is replication. There are two types of replication such as complete replication and selective replication. In this project, complete replication is used to maintain a full and complete copy of database at each distributed site which is Site 1 and Site 2. This strategy required the copies of data to be updated frequently in order to keep the data up to date. This is a cost to the business which will be discussed later.

There are several methods can be used to replicate data in a distributed database. This project is using multi-master replication where all the distributed sites are masters. Multi-master replication allows retrieval or update of data by every master in the group.

## **4.5 SIMULATION OF DATA RETRIEVAL PROCESS**

This section will be discussed about the simulation of healthcare information from a distributed database in two types of access, one is local access by using complete replication strategy while another one is remote access by using horizontal fragmentation strategy.

### **4.5.1 Simulation of Data Retrieval Process Using Horizontal Fragmentation Strategy**

Horizontal fragmentation is used to simulate the remote access of client (Site 2) to the data stored in server (Site 1). Assume that client side (Site 2) only stores patient information which is located at its site where field named "H\_ID"=Site 2. This simulation only involved retrieval of data from single table. The flow of the simulation processes are described as below.

1. Execution of query to retrieve data from each table at client site (Site 2).
2. Record the power consumption from the power meter when each query is executed.

To model the process of remotely access the data stored in another site, Microsoft Visual Studio is used to connect to the database. When a query requests for information is executed at one site which is known as client site (eg: Site 2), it will multicast the request to all available servers (Site 1 and Site 2). If Site 2 does not have the requested information and other server (Site 1) has it, the server (Site 1) will return the information to the client site (Site 2).

Remote access happens when a query executed at client side (Site 2) but the result is returned by another server located at different physical location (Site 1). Remote access allows retrieval of distributed data from a database at another site without being physically execute the query at the site.

Below are the queries executed at Site 2 to remotely access data stored in each fragments located at Site 1.

### **Transaction 1: Retrieval of patient information**

#### **Retrieve patient information where “H\_ID” =Site 1**

Fragment 1: Table “Patient” fragmented by hospital address (S<sub>1</sub>)

F<sub>1</sub>:  $\sigma_{H\_ID = 'Site 1'}(PATIENT)$

SQL statement: SELECT \* FROM F1;

#### **Retrieve patient information where “H\_ID” =Site 2**

Fragment 2: Table “Patient” fragmented by hospital address (S<sub>2</sub>)

F<sub>2</sub>:  $\sigma_{H\_ID = 'Site 2'}(PATIENT)$

SQL statement: SELECT \* FROM F2;

### **Transaction 2: Retrieval of doctor information**

#### **Retrieve doctor information where “H\_ID” =Site 1**

Fragment 3: Table “Doctor” fragmented by hospital address (S<sub>1</sub>)

F<sub>3</sub>:  $\sigma_{H\_ID = 'Site 1'}(DOCTOR)$

SQL statement: SELECT \* FROM F3;

#### **Retrieve doctor information where “H\_ID” =Site 2**

Fragment 4: Table “Doctor” fragmented by hospital address (S<sub>2</sub>)

F<sub>4</sub>:  $\sigma_{H\_ID = 'Site 2'}(DOCTOR)$

SQL statement: SELECT \* FROM F4;

### **Transaction 3: Retrieval of patient medical information**

#### **Retrieve patient medical information where “H\_ID” =Site 1**

Fragment 5: Table “PatientVisit” fragmented by hospital address (S<sub>1</sub>)

F<sub>5</sub>:  $\sigma_{H\_ID = 'Site 1'}(PATIENTVISIT)$

SQL statement: SELECT \* FROM F5;

## **Retrieve patient medical information where “H\_ID” =Site 2**

Fragment 6: Table “PatientVisit” fragmented by hospital address (S<sub>2</sub>)

F<sub>6</sub>:  $\sigma_{H\_ID = 'Site 2'}(PATIENTVISIT)$

SQL statement: SELECT \* FROM F<sub>6</sub>;

### **4.5.2 Simulation of Data Retrieval Process Using Complete Replication Strategy**

Complete replication is used as a strategy that allows local access to the data stored in local database. In this simulation, the query is executed at Site 1. Assume that Site 1 contains a complete copy of information of itself as well as information stored at Site 2. This simulation involves retrieval of data from single table only.

The flow of the simulation processes are described as below.

1. Execution of query to retrieve data from each table at local server (Site 1).
2. Record the power consumption from the power meter when each query is executed.

Below are queries executed at Site 1 to locally access the replica of all data stored in its local server.

#### **Transaction 1: Retrieval of doctor information**

##### **Scenario 1: Retrieve doctor information at Site 1**

SQL statement: SELECT \* FROM DOCTOR WHERE H\_ID='Site 1';

##### **Scenario 2: Retrieve doctor information at Site 2**

SQL statement: SELECT \* FROM DOCTOR WHERE H\_ID='Site 2';

#### **Transaction 2 :Retrievalof patient information**

##### **Scenario 1: Retrieve patient information at Site 1**

SQL statement: SELECT \* FROM PATIENT WHERE H\_ID='Site 1';

### Scenario 2: Retrieve patient information at Site 2

SQL statement: SELECT \* FROM PATIENT WHERE H\_ID ='Site 2';

### Transaction 3 :Retrievalof patient medical information

#### Scenario 1: Retrieve patient medical information at Site 1

SQL statement: SELECT \* FROM PATIENTVISIT WHERE H\_ID='Site 1';

#### Scenario 2: Retrieve patient medical information at Site 2

SQL statement: SELECT \* FROM PATIENTVISIT WHERE H\_ID ='Site 2';

## 4.6 SIMULATION RESULT ANALYSIS

The simulation of the data retrieval process is started after the data has been fragmented and replicated as discussed earlier. Below are some important information about the specification of hardware and software used during the simulation.

Hardware / Software	Specification
i. Dell Inspiron 1420 Model	Intel Core 2 Duo CPU @ 2GHz, 2Gb RAM
ii. LAN bandwidth	Maximum 100Mb/s
iii. Server (boinc and nativeboinc)	Intel Xeon CPU X5550 @2.67GHz

The power consumption captured by the power meter during the execution of query for the simulation above are tabulated into table to calculate the average as well as weighted average. Two equations that are used to calculate the average and weighted average are:

i. Equation 4.1

$$\text{Average} = \frac{\sum A}{N} \quad \text{where A: power consumption for each query}$$

N: number of query execution

ii. Equation 4.2

$$\text{Weighted Average} = 100\% \times \frac{\sum B}{\text{Total of B}}$$

where B: Total in average of all tables

Below is the table that shows the power consumption by each query processing strategy and the calculation of average as well as weighted average.

<b>No. of query execution(N)</b>	<b>Table accessed</b>	<b>Power Consumption for horizontal fragmentation in watt (A)</b>	<b>Power Consumption for complete replication in watt (A)</b>
1	Patient	39.4	29.1
	Doctor	36.5	28.7
	PatientVisit	39.6	30.4
2	Patient	38.6	29.1
	Doctor	36.9	28.7
	PatientVisit	42.3	30.7
3	Patient	40.2	29.2
	Doctor	37.7	28.5
	PatientVisit	41.6	31.7
4	Patient	39.8	30.5
	Doctor	37.2	29.3
	PatientVisit	43.1	32.1
5	Patient	41.2	29.6
	Doctor	37.5	28.1
	PatientVisit	42.6	31.9
Average ( $\sum A/N$ )	Patient	39.8	29.5
	Doctor	37.2	28.6
	PatientVisit	41.8	31.4
Total power in average (B)		39.6	29.8
Percentage of total power consumption (B)		57%	43%

Table 4.1: Power Consumption for Simulation Using Both Query Processing Strategies

The average power consumption to access each table as computed in Table 4.1 is tabulated into graph form as shown below.

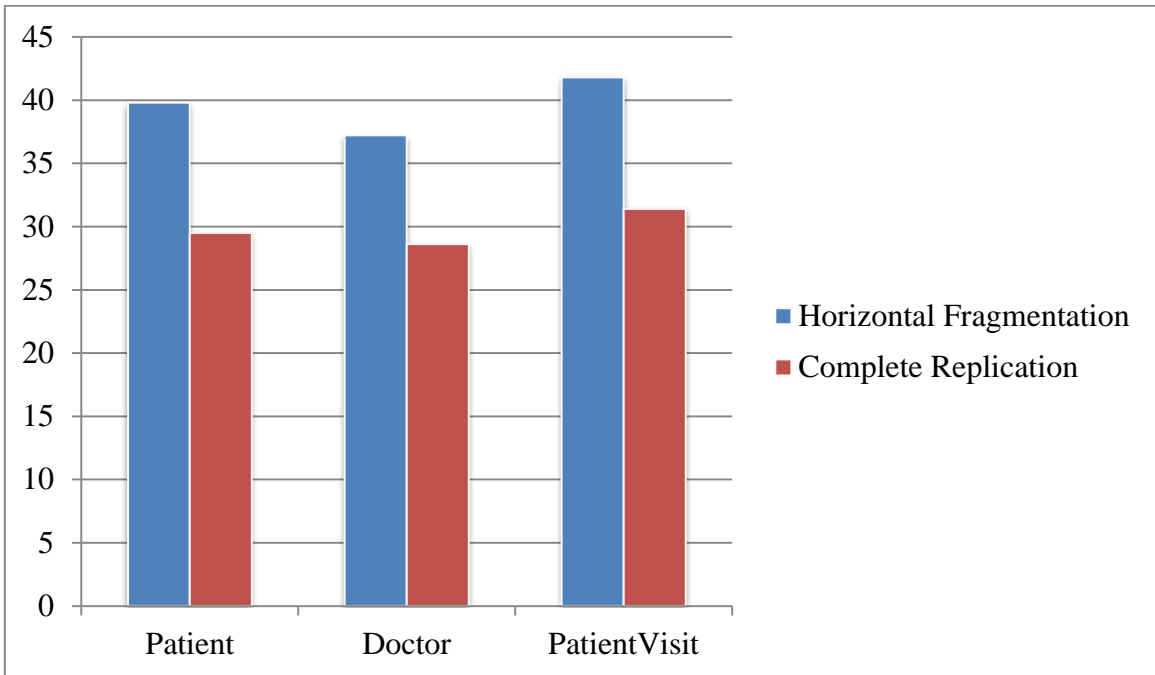


Figure 4.6 Average Power Consumption (watt) to Access Each Table

The weighted average from Table 4.1 is tabulated into graph form as shown below.

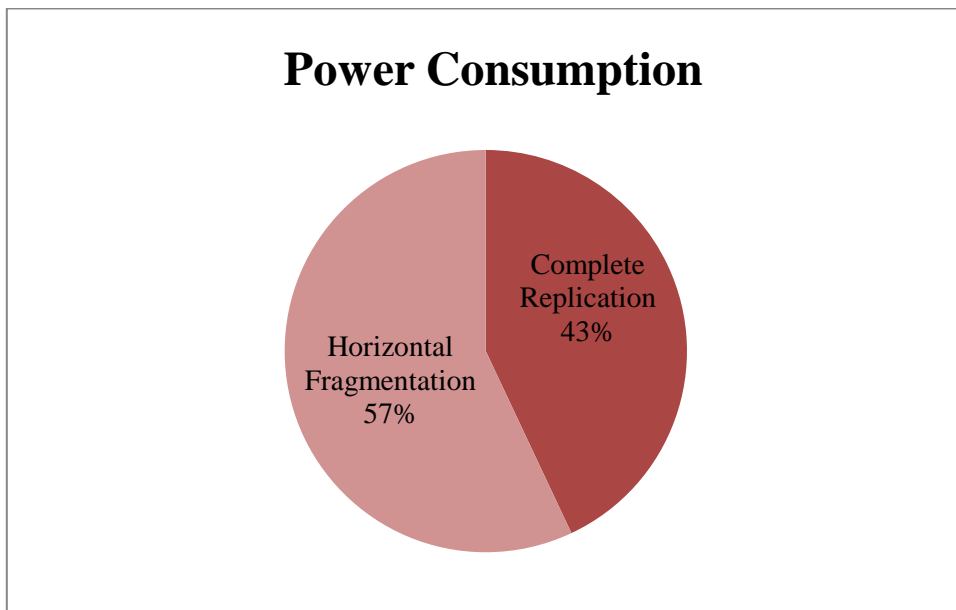


Figure 4.7 Power Consumption of Horizontal Fragmentation and Complete Replication

Table 4.1 shows the power consumption for both strategies during the simulation. Based on the calculation, the average of power consumption by remote access (horizontal fragmentation strategy) to patients' information during the simulation is 39.6 watt while local access (complete replication strategy) to patients' information averagely consumed 29.8 watt of power. The difference in power consumption between two strategies is about 10 watt. It can be clearly seen that local access which used complete replication as the query processing strategy consume lesser energy as compared to that of remote access which used horizontal fragmentation.

Looking at Figure 4.7 which tabulates the data form Table 4.1 into chart form shows that horizontal fragmentation strategy used up 57% of the total power consumption during the simulation while complete replication strategy only used 43% of the total power consumption. The difference of 14% in the power consumption between two strategies is equivalent to 10 watt of power. In terms of energy, 10 watt is equivalent to 10 joules per second. 10 joules seems to be a small amount, however in a long run and running on a huge amount of data, a savings in 10 joules can make a big difference to the environment as well as the business operational cost.

However, this set of result is only acceptable to certain extent only referring to the specification of hardware and software used in the simulation as discussed earlier. In real life, each healthcare institution uses different type of computer models with varying processing speed, thus the power consumption during data retrieval process may varies as well. Other than that, the LAN bandwidth is also important to contribute to the power used to access data especially for remote access. This is because remote access requires transfer of data among two sites located at different physical location which consume more power during the transfer process.

As stated earlier the objective of this project is to model the data retrieval process in distributed database by using different query processing strategies and analyze the result in order to identify the energy-efficient query processing strategy. Based on the simulation result which used complete replication and horizontal fragmentation to model the local access and remote access, it can be said that complete replication that enables local access



to the data stored in distributed database consume lesser energy as compared to remote access. Lesser energy consumption also means higher efficiency in query performance as lesser time is taken to execute a query and get reply from the server.

Other than being energy-efficient, complete replication also provides the benefits of improved local database performance and availability of data. When requested information is not found in the local database, replication enables client to access the replica of other database in its own server locally to search for the information needed. This minimizes the network traffic of the database and achieves maximum performance in a shorter response time.

Despite the lower power consumption to retrieve replica of distributed data on a local database by complete replication strategy, there are several costs need to be considered before implementation. Complete replication is about maintaining a complete copy of the database at each site. The storage cost and communication costs for update are expensive and vary depending on the size of data and servers. Before this strategy is implemented, one should consider the requirement of healthcare institution to make sure that their data is up to date so that the requestor gets the most updated information for better patient treatment. However, it will cost a lot in terms of money and time to the healthcare providers to update their replicas frequently.

In a nutshell, complete replication is an energy-efficient query processing strategy which allows local access to the replica of database of other sites at its own database. Besides being energy-efficient, complete replication reduce the time taken to retrieve the requested data from a distributed database. Nonetheless, the cost of replication to store and maintain the updated as well as complete copy of the distributed database at each site in the local database is expensive for healthcare institution. Thus, the healthcare institutions have to make a balance between the reductions in power consumption and cost allocation to implement a complete replication.

## **CHAPTER 5**

### **CONCLUSIONS AND RECOMMENDATIONS**

#### **5.1 RELEVANCY TO OBJECTIVE**

On top of that, with regards to the objective of implementing green computing in healthcare industry, the target market of this project is healthcare institutions which have more than one branch in a country and store their data by using distributed database. Energy-efficient query processing strategy is important to healthcare industry as reduction in power consumption can as well reduce the negative impact to the environment which is a part of their corporate social responsibility. Furthermore, energy-efficient query processing strategy is able to reduce the operational cost of the business by consuming lesser power. In this case, it can be seen that the implementation of green computing in healthcare industry not only benefits the institution but also take care of the environment which is in line with the objective of green computing proposed earlier.

The objective of this project is to identify the most energy-efficient query processing strategy to retrieve medical data from distributed database. In order to achieve the specified goal, a simulation that model the data retrieval process has been carried out to calculate power consumption for each query processing strategy used. In terms of relevancy, it can be concluded that it is highly relevant to the proposed objectives. This is because all the processes from planning to computation the final stage are designed to find out the energy-efficient strategy to access healthcare information at minimum power consumption and time taken.

In short, the project does follow the objectives as specified and this can be clearly seen from the deliverable of the project which is the identified energy-efficient query processing strategy (complete replication) for medical data stored in distributed database based on the computation result of power consumption for each simulation.

## **5.2 SUGGESTED FUTURE WORK FOR EXPANSION AND CONTINUATION**

According to research, an IT process particularly data center or server consume about 23% of the total energy demand. Since the objective of the project is to reduce the energy consumption of query processing, it is suggested that the energy consumption should be reduced to a minimum of 15% which will effectively reduce the relative cost as well.

Other than that, the simulation for this project only covers single table queries. However in real life situation, healthcare industries might require more complex queries to access the data from the distributed database. Thus it is recommended to extend this research to multi-table queries to make this research more realistic. Add on to that, this simulation model the data retrieval process from table which only contains a maximum of 1000 data. It is good if this research can expand the scope to a higher amount of data to suit the real life situation in a healthcare institution which normally has more than 10,000 of patients.

Due to time constraint, the scope of the project has been narrowed down to a particular industry which is healthcare industry. As for future work to expand and continue, it will be great if the research can be expanded to other industries which require high efficiency in data retrieval process such as banking industry. With the expansion of the project on other industries, the advantages of green computing not only benefit the industry but our precious mother nature as well.

In conclusion, there is still room of improvement for this project to continue in the future. This can be achieved by expanding the scope of the project to become more real-life based and also expand the research to other industry where green computing can help them to accomplish energy efficiency in their data retrieval process.

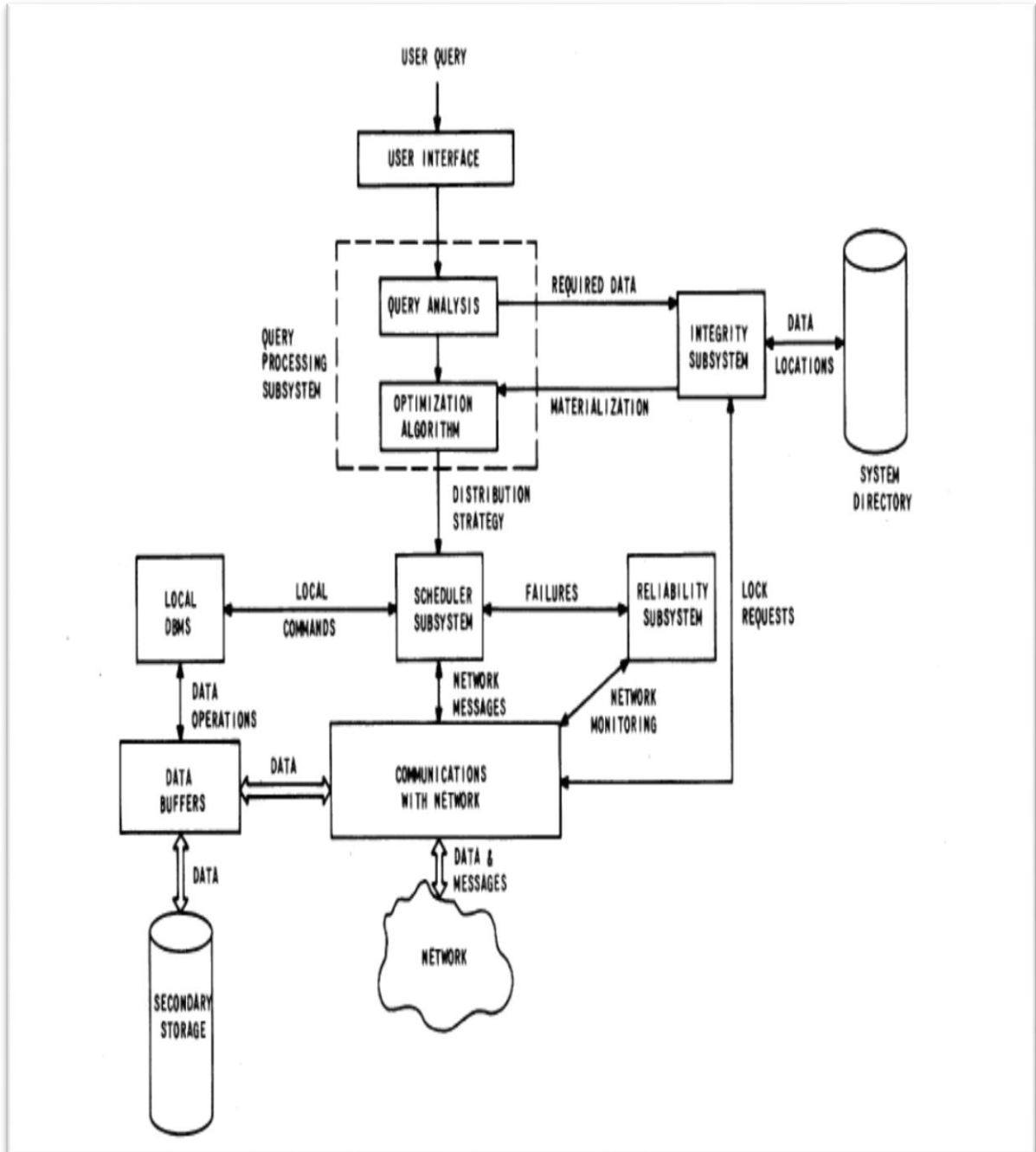
## REFERENCES

- [1] “*Fiona Caldicott to lead review into sharing of health information*”  
Retrieved 21<sup>th</sup> November 2012 from: [www.guardian.co.uk](http://www.guardian.co.uk)
- [2] Health Data Management  
Retrieved 21<sup>th</sup> November 2012 from: [www.bridgeheadsoftware.com](http://www.bridgeheadsoftware.com)
- [3] “*8 Health Information Exchange Leading the Way*”  
Retrieved 21<sup>th</sup> November 2012 from: [www.informationweek.com](http://www.informationweek.com)
- [4] Dr H. Hakimzadeh, “*Distributed Databases Fundamentals and Research*”, Department of Computer and Information Sciences, Indiana University South Bend
- [5] Stefano Ceri & Giuseppe Pelagatti (1985): *Distributed Databases Principles & Systems*: McGraw-Hill Book Company Publisher
- [6] Thomas Connolly & Carolyn Begg (2010): *Database Systems – A Practical Approach to Design, Implementation and Management*: Pearson Education International Publisher
- [7] Korth and Sudarshan (2010): “*Database System Concepts*: McGraw-Hill Book Company Publisher
- [8] R. Elmasri and S.B. Navathe (2004): “*Principle of Database Query Processing*”  
4<sup>th</sup> Edition: Addison-Wesley Publisher
- [9] Alan R. Hevner and S. Bing Yao, *Query Processing in Distributed Database System*, IEEE Transactions on Software Engineering , Vol. SE-5, No.3
- [10] Michael L. Rupley, Jr., *Introduction to Query Processing and Optimization*, Indiana University at South Bend
- [11] Query Processing  
Retrieved 21<sup>th</sup> July 2012 from: [www.en.wiktionary.org](http://www.en.wiktionary.org)

- [12] PanktiDoshi& Vijay Raisinghani, “*Review of Dynamic Query Optimization Strategies in Distributed Database*”, Department of Computer Science and Information Technology, Mukesh Patel School of Technology Managemnt and Engineering, NMIMS Deemed-to-be University
- [13] San Murugesan (2008): “*Harnessing Green IT: Principles and Practices*”, Volume 10, Issue 1
- [14] Green Computing: What is Green Computing?  
Retrieved 21<sup>th</sup> June 2012  
from:<http://greenelectronics.com/FAQRetrieve.aspx?ID=31973&Q=>
- [15] Rajguru P.V, Nayak S.K and More D.S, *Solution for Green Computing*, Department of Computer Science and IT, Adarsh college, Hingoli (Maharashtra), India
- [16] Data Center Power Consumption  
Retrieved 21<sup>th</sup> November 2012 from:[www.bridgeheadsoftware.com](http://www.bridgeheadsoftware.com)
- [17] The Star, “*Green computing as a way to reduce IT operation costs*”, Monday, 21<sup>st</sup> of July 2008
- [18] Willis Lang, RamakrishnanKandhan, JigneshM.Patel, *Rethinking Query Processing for Energy Efficiency: Slowing Down to Win the Race*, Computer Sciences Department, University of Wisconsin, Madison
- [19] Suzanne Rivoire, MehulA.Shah, Parthasarathy and Christos Kozyrakis, *JouleSort: A Balanced Energy-Efficiency Benchmark*, Stanford University
- [20] DimitrisTsirogiannis, Stavros Harizopoulos, Mehul A. Shah, *Analyzing the Energy Efficiency of a Database Server*, University of Toronto

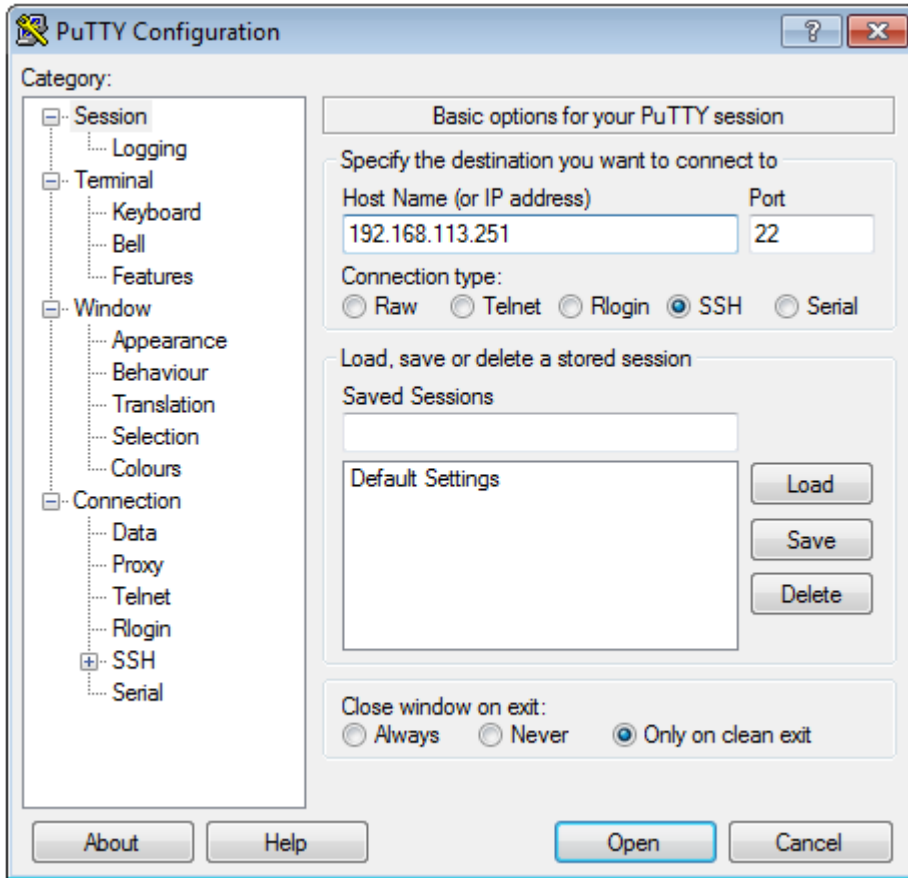
# APPENDIX 1

## Query processing in a distributed database system



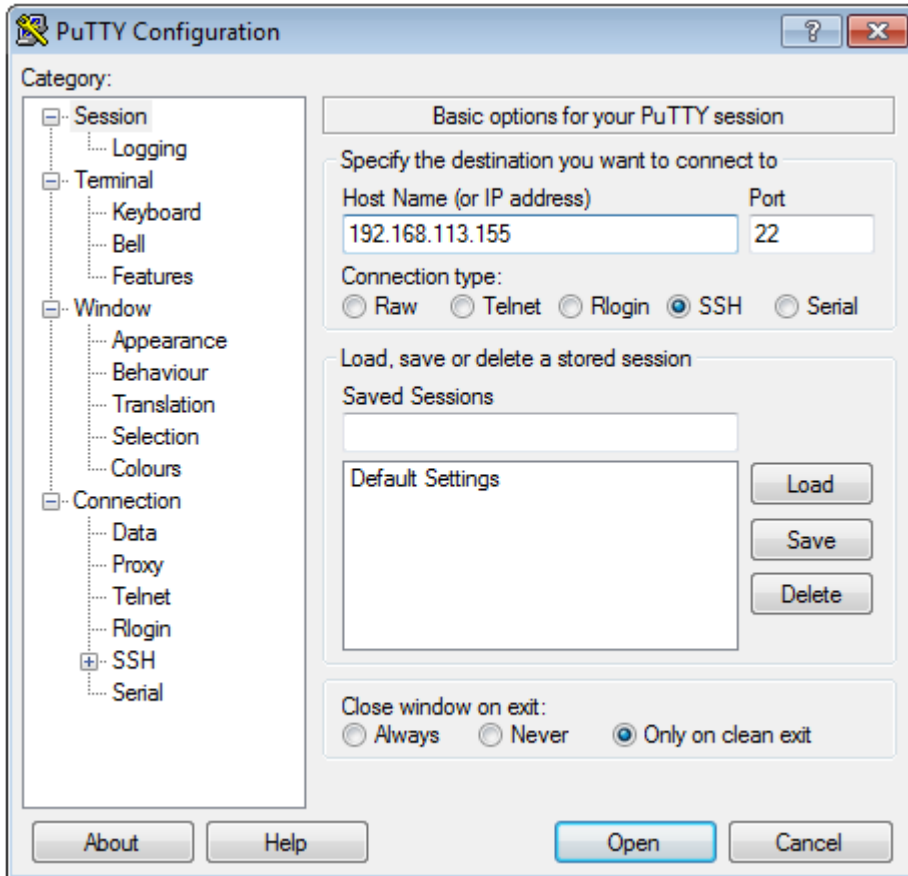
## APPENDIX 2

PuTTY configuration to connect database (Site 1: IP Address 192.168.113.251)



### APPENDIX 3

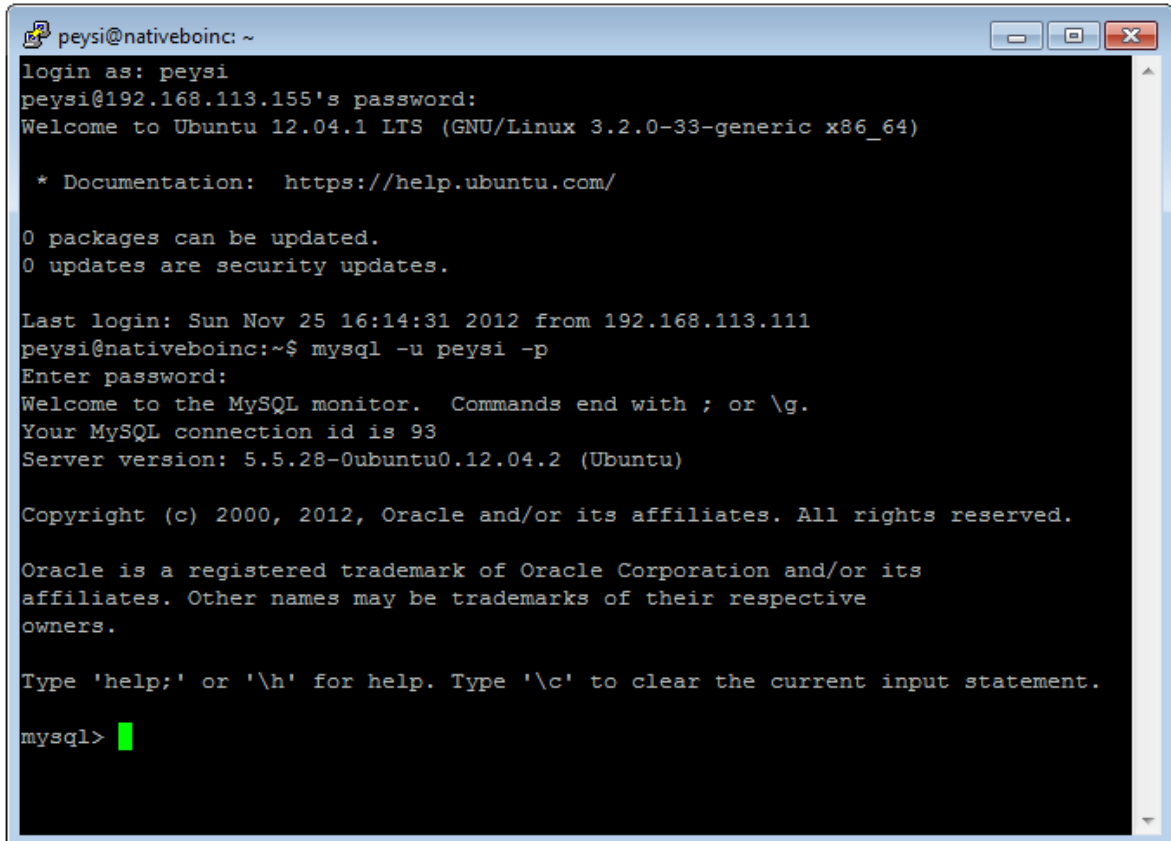
PuTTY configuration to connect database (Site 2: IP Address 192.168.113.155)





## APPENDIX 4

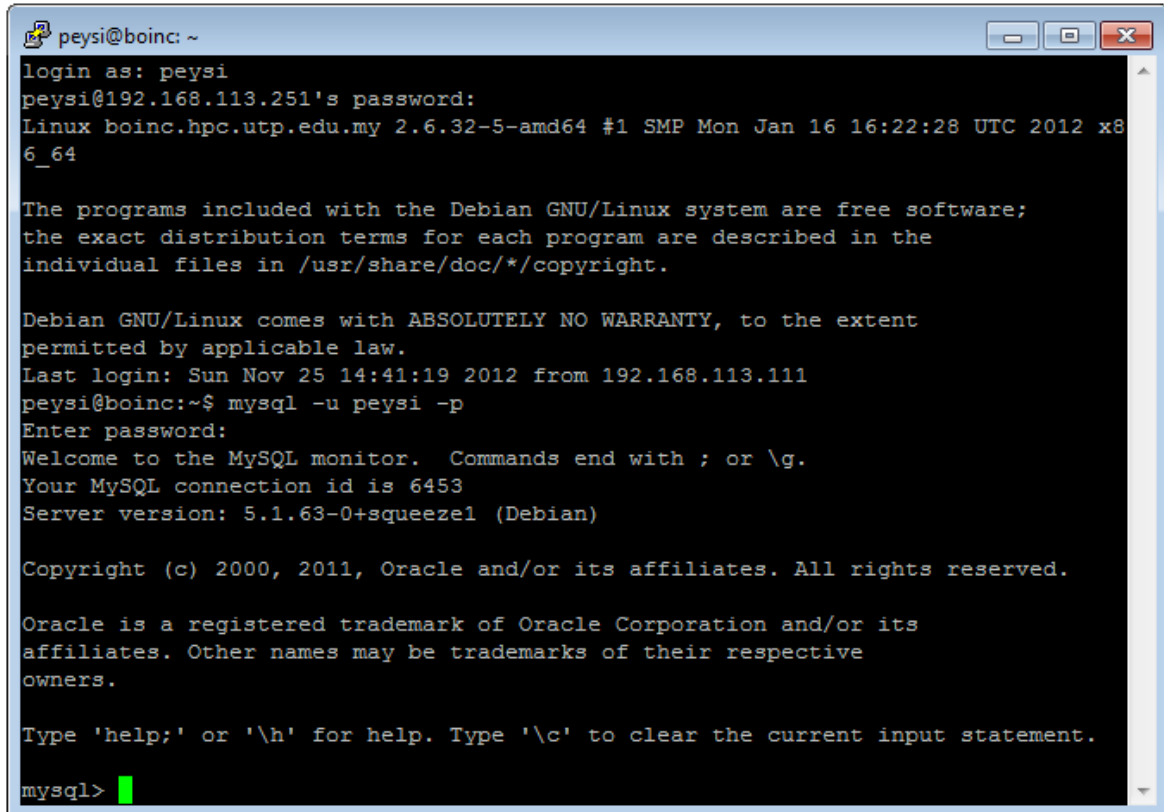
Connected to database (Site 1: IP Address 192.168.113.251)



```
peysi@nativeboinc: ~  
login as: peysi  
peysi@192.168.113.155's password:  
Welcome to Ubuntu 12.04.1 LTS (GNU/Linux 3.2.0-33-generic x86_64)  
  
* Documentation:  https://help.ubuntu.com/  
  
0 packages can be updated.  
0 updates are security updates.  
  
Last login: Sun Nov 25 16:14:31 2012 from 192.168.113.111  
peysi@nativeboinc:~$ mysql -u peysi -p  
Enter password:  
Welcome to the MySQL monitor.  Commands end with ; or \g.  
Your MySQL connection id is 93  
Server version: 5.5.28-0ubuntu0.12.04.2 (Ubuntu)  
  
Copyright (c) 2000, 2012, Oracle and/or its affiliates. All rights reserved.  
  
Oracle is a registered trademark of Oracle Corporation and/or its  
affiliates. Other names may be trademarks of their respective  
owners.  
  
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.  
mysql>
```

## APPENDIX 5

Connected to database (Site 2: IP Address 192.168.113.155)



```
peysi@boinc: ~
login as: peysi
peysi@192.168.113.251's password:
Linux boinc.hpc.utp.edu.my 2.6.32-5-amd64 #1 SMP Mon Jan 16 16:22:28 UTC 2012 x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sun Nov 25 14:41:19 2012 from 192.168.113.111
peysi@boinc:~$ mysql -u peysi -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 6453
Server version: 5.1.63-0+squeezel (Debian)

Copyright (c) 2000, 2011, Oracle and/or its affiliates. All rights reserved.

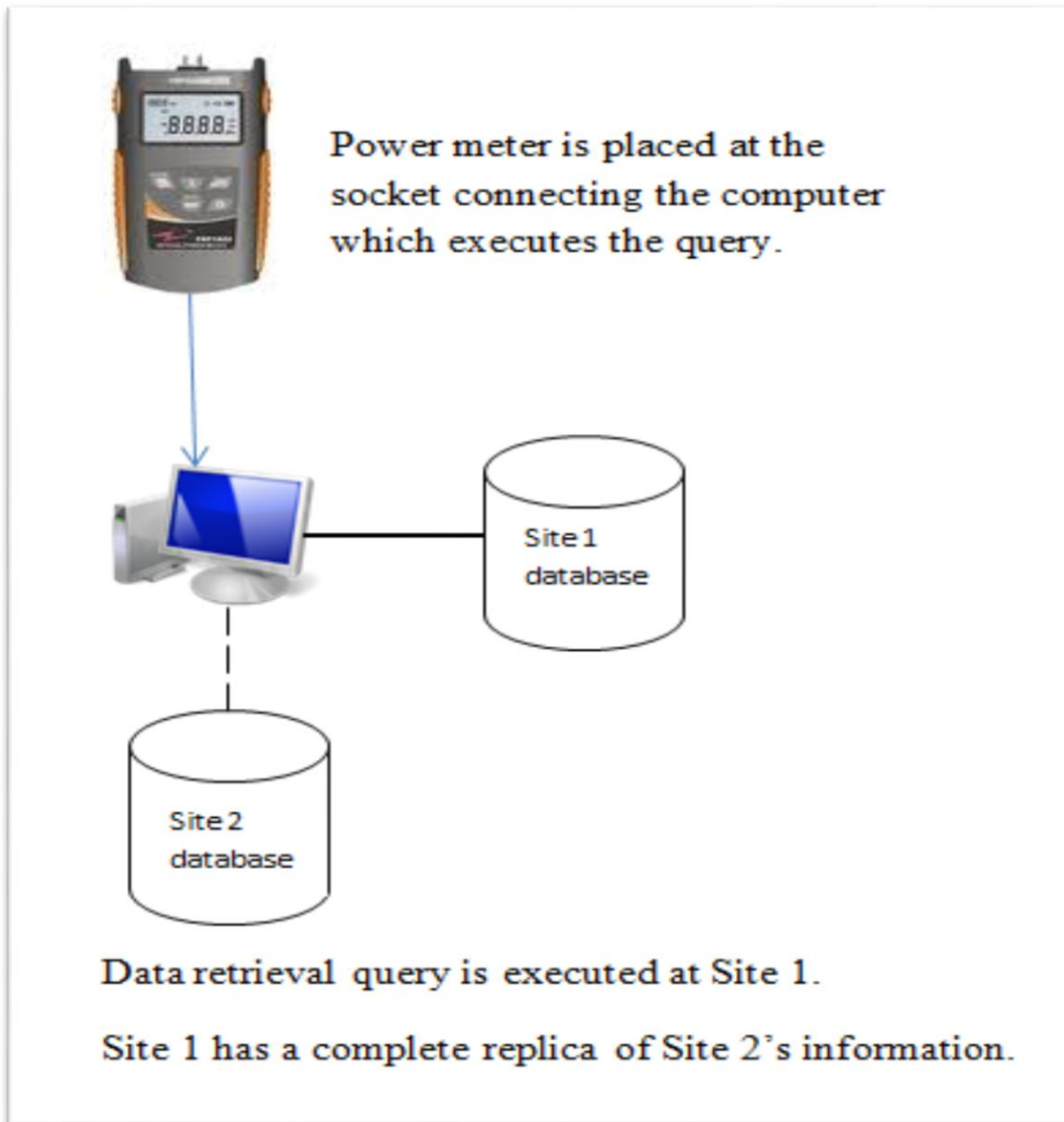
Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

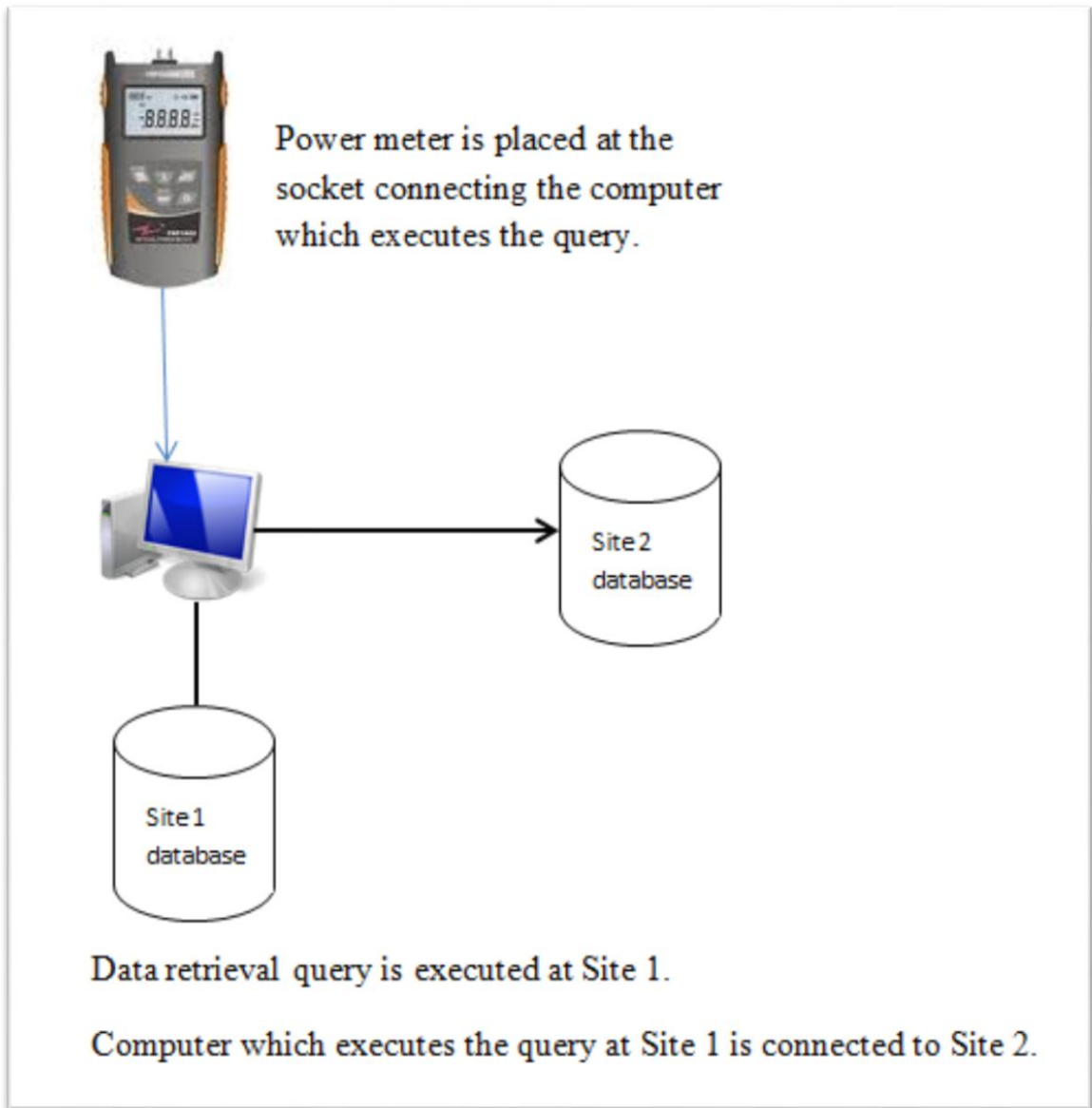
## APPENDIX 6

Architecture of simulation using complete replication strategy to execute query either at Site 1 or Site 2 to retrieve data from local server.



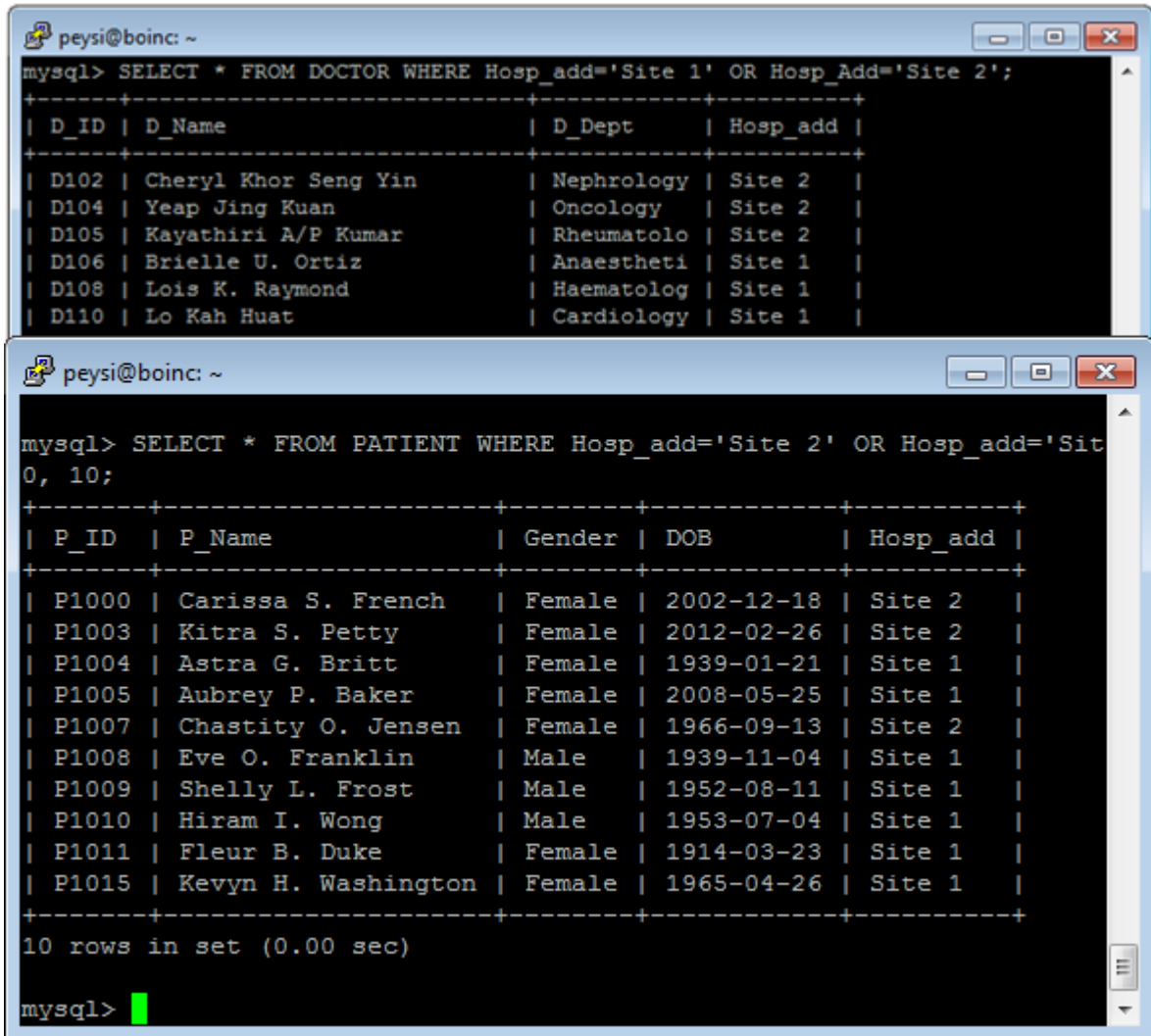
## APPENDIX 7

Architecture of simulation using horizontal fragmentation strategy to execute query at Site 1 to retrieve data from Site 2.



## APPENDIX 8

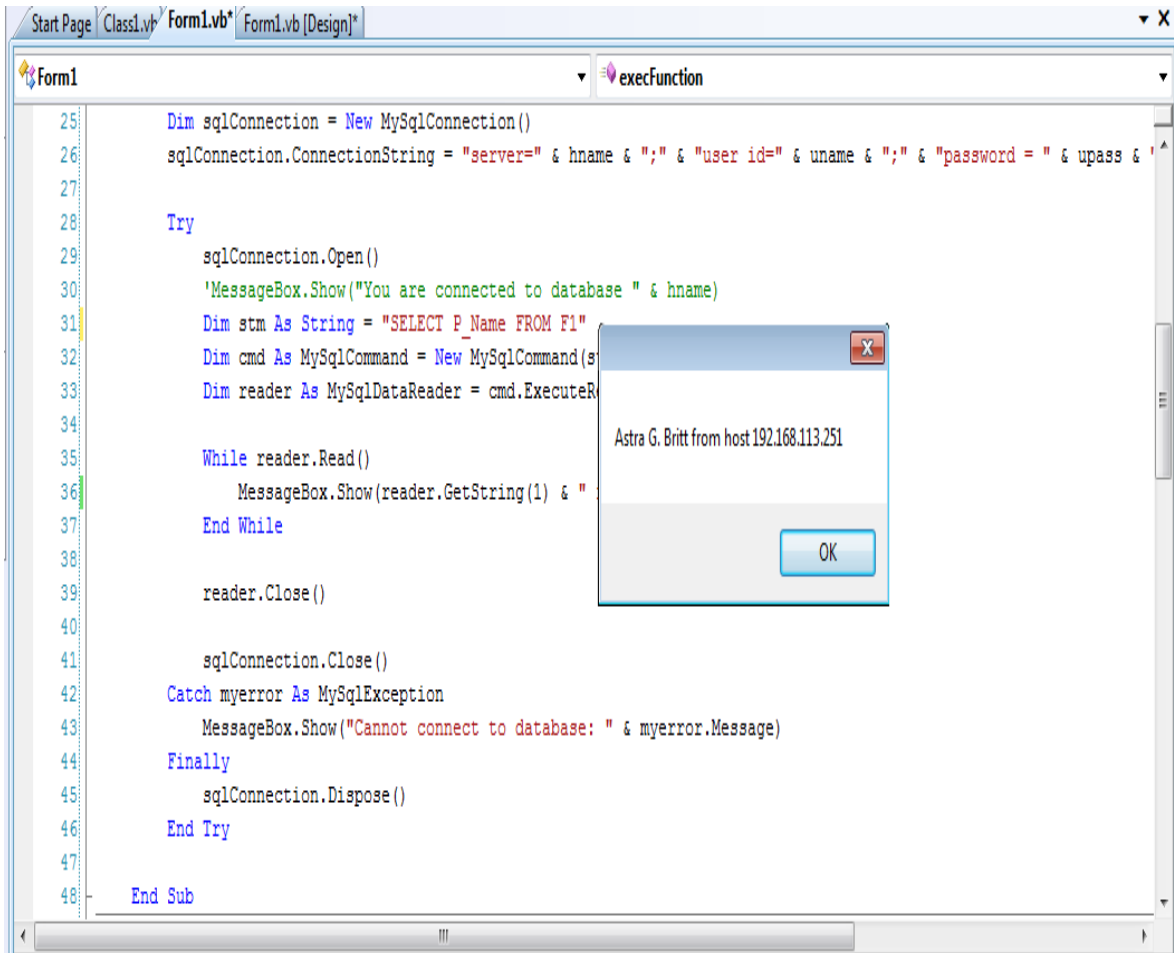
Screen shot of result from simulation using complete replication strategy.



```
peysi@boinc: ~  
mysql> SELECT * FROM DOCTOR WHERE Hosp_add='Site 1' OR Hosp_Add='Site 2';  
+-----+-----+-----+-----+  
| D_ID | D_Name                | D_Dept   | Hosp_add |  
+-----+-----+-----+-----+  
| D102 | Cheryl Khor Seng Yin  | Nephrology | Site 2 |  
| D104 | Yeap Jing Kuan        | Oncology   | Site 2 |  
| D105 | Kayathiri A/P Kumar   | Rheumatolo | Site 2 |  
| D106 | Brielle U. Ortiz      | Anaestheti | Site 1 |  
| D108 | Lois K. Raymond       | Haematolog | Site 1 |  
| D110 | Lo Kah Huat           | Cardiology | Site 1 |  
+-----+-----+-----+-----+  
peysi@boinc: ~  
mysql> SELECT * FROM PATIENT WHERE Hosp_add='Site 2' OR Hosp_add='Site 1'  
0, 10;  
+-----+-----+-----+-----+-----+  
| P_ID | P_Name                | Gender | DOB        | Hosp_add |  
+-----+-----+-----+-----+-----+  
| P1000 | Carissa S. French     | Female | 2002-12-18 | Site 2 |  
| P1003 | Kitra S. Petty        | Female | 2012-02-26 | Site 2 |  
| P1004 | Astra G. Britt        | Female | 1939-01-21 | Site 1 |  
| P1005 | Aubrey P. Baker       | Female | 2008-05-25 | Site 1 |  
| P1007 | Chastity O. Jensen    | Female | 1966-09-13 | Site 2 |  
| P1008 | Eve O. Franklin       | Male   | 1939-11-04 | Site 1 |  
| P1009 | Shelly L. Frost       | Male   | 1952-08-11 | Site 1 |  
| P1010 | Hiram I. Wong         | Male   | 1953-07-04 | Site 1 |  
| P1011 | Fleur B. Duke         | Female | 1914-03-23 | Site 1 |  
| P1015 | Kevyn H. Washington   | Female | 1965-04-26 | Site 1 |  
+-----+-----+-----+-----+-----+  
10 rows in set (0.00 sec)  
mysql>
```

## APPENDIX 9

Screen shot of result during simulation using horizontal fragmentation strategy (access from Site 2: 192.168.113.155 to fragment F1 located at Site 1: 192.168.113.251).



```
25 Dim sqlConnection = New MySqlConnection()
26 sqlConnection.ConnectionString = "server=" & hname & ";" & "user id=" & uname & ";" & "password = " & upass & "
27
28 Try
29     sqlConnection.Open()
30     'MessageBox.Show("You are connected to database " & hname)
31     Dim stm As String = "SELECT P_Name FROM F1"
32     Dim cmd As MySqlCommand = New MySqlCommand(stm, sqlConnection)
33     Dim reader As MySqlDataReader = cmd.ExecuteReader()
34
35     While reader.Read()
36         MessageBox.Show(reader.GetString(1) & " ")
37     End While
38
39     reader.Close()
40
41     sqlConnection.Close()
42 Catch myerror As MySqlException
43     MessageBox.Show("Cannot connect to database: " & myerror.Message)
44 Finally
45     sqlConnection.Dispose()
46 End Try
47
48 End Sub
```

Astra G. Britt from host 192.168.113.251

OK

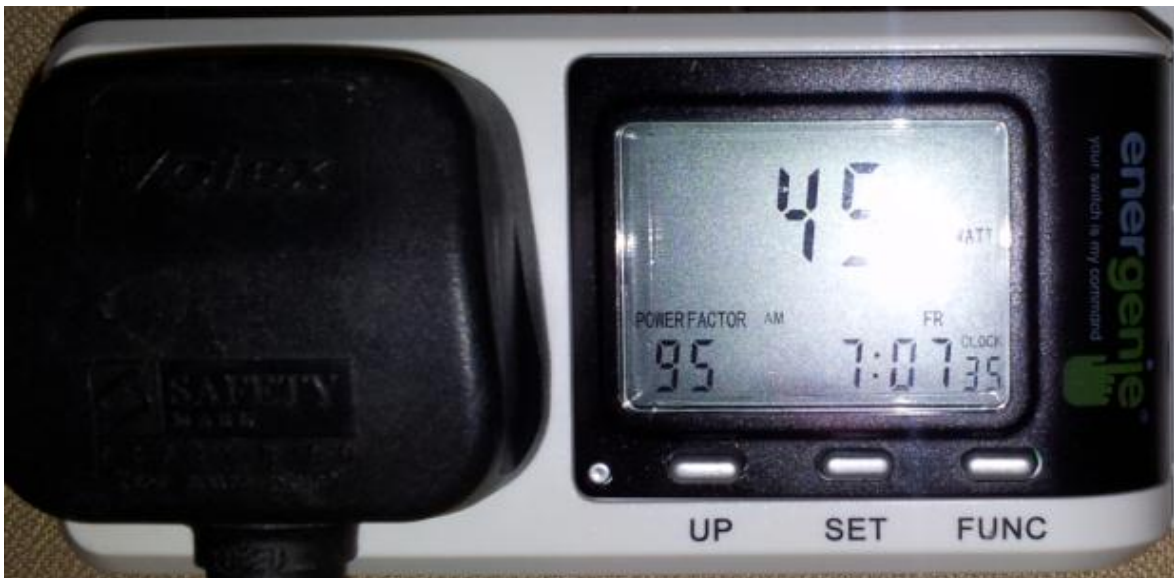
## APPENDIX 10

Readings from power meter before and after the execution of query to retrieve data.

Before query execution:



After query execution:



**APPENDIX 11**  
**TECHNICAL PAPER**



# Modeling and Analyzing Power Consumption in Query Processing for Distributed Database

Teh Pey Si

Department of Computer and Information Sciences,  
Universiti Teknologi PETRONAS,  
Bandar Seri Iskandar, Tronoh Perak, Malaysia  
peysi7@gmail.com

Rozana bt Kasbon

Department of Computer and Information Sciences,  
Universiti Teknologi PETRONAS,  
Bandar Seri Iskandar, Tronoh Perak, Malaysia  
rozank@petronas.com.my

**Abstract - Green computing has been generally practiced in almost all kind of fields especially in the recent years as environmental sustainability is getting more important. High power consumption increases the carbon emission which is adverse to the environment. This project focuses on applying green computing in query processing specifically for distributed database in healthcare industry. The information about a patient is stored in the database of the hospital the patient visited. However, currently this information is not being shared among hospitals which are crucial for diagnosis purpose. Hence, the objective of this project is to model the process of data retrieval from database distributed at different hospitals by using different query processing strategies and analyses the energy consumption to access these distributed databases. Based on the simulation result, the identified energy-efficient strategy is complete replication which consumed lesser power consumption by enabling local access to healthcare data stored in distributed database.**

## I. INTRODUCTION

### A. Background of Study

When a patient visits a hospital or clinic for the first time, personal information such as name, identification number, age, birth date, housing address and medical information are recorded and stored. Some medical institutions store these data physically by keeping them in hard copy such as paper files while some store the data electronically by entering them into the computer and save in database.

As people nowadays often travel around the country from a place to another so one might fall sick and visit a medical institution at any time anywhere. Thus there is possibility that a patient may visit a medical institution which is not the one he usually visits. Patient information such as blood type and medical history is important for diagnosis purpose especially when there is an emergency. As a result, it is crucial to ensure that the information about a patient is able to be accessed and retrieved at a minimum time among different medical institutions. In other words, the efficiency of the patient data retrieval process leads to higher efficiency in patient diagnosis process.

However, currently in Malaysia, the hospitals either public or private are not sharing their patients' information among each other. Some of the hospitals are still using the old way by storing their patient data in paper files while some store the data in their own database that is not connected to others. This type of data storing method is known as centralized database. Since the importance of accessibility of patient information among hospitals has been highlighted above, it is recommended for these hospitals to implement distributed database to store their patient information at multiple physical locations.

This project intends to find out the most energy-efficient query processing strategy to retrieve medical information from the distributed database in order to enhance the data retrieval process for more efficient and effective diagnosis.

### B. Problem Statement

In support to the project topic, several problems have been identified and listed as follow:

1. Inaccessibility of patient information among hospitals which is crucial for diagnosis purpose.
2. Inefficiency in retrieving patient information slows down the diagnosis process.
3. No query processing strategy has been identified as the most energy-efficient way to access data from distributed database.
4. Inefficient query processing strategy leads to high power consumption.
5. High power consumption increases carbon emission to the environment
6. High power consumption increases the operational cost of medical institution.
7. Inefficient resource allocation to access medical data from database.
8. Healthcare industry has not generally applied green computing like other industries.

### C. Objective

This project aims to achieve several goals as follow:

1. To model the data retrieval process of patient information from medical institution at dispersed locations by developing a distributed database.
2. To analyse the power consumption by each different query processing strategies to retrieve data from distributed database.
3. To identify the most energy-efficient query processing strategy to retrieve data stored in distributed database.

### D. Scope of Study

In order to model the data retrieval process of patient information from distributed database, several scopes of study has been identified as follow:

1. To study on distributed database
  - Do research and studies about distributed database and query processing strategy to process data from distributed database
2. Simulation of data retrieval process from distributed database and record the power consumption
  - In order to set up the simulation, two databases will be created at different networks to model the distributed database. Each query processing strategy will be tested to retrieve the data stored at

different networks and the power consumption will be computed by a power meter.

3. Analysis of the simulation
  - The query processing strategy that consumes the least amount of power will be identified.

## II. LITERATURE REVIEW

### A. *Solution for Data Sharing among Healthcare Institutions*

One of the solutions is Healthcare Data Management (HDM) provided by [2] BridgeHead Software. According to the company's website, HDM solution allows healthcare data to be stored efficiently, fully protected and could be shared among other hospitals, making the data accessible to people that need it for the delivery of quality patient care. BridgeHead's HDM solution provides hospitals the ability to store all their data in one place efficiently and intelligently. On top of that, BridgeHead's HDM solution also enables hospitals to share their clinical data and administrative data among departments or with other hospitals. This can be achieved by having the feature of web service enabled, access control, authentication as well as encryption to protect the data.

According to an article posted by [3] Marianne Kolbasuk McGee on 8<sup>th</sup> of June 2012, in United States there are eight Health Information Exchanges (HIE) to help the U.S. health organizations to share data in the name of lower costs and better patient care. The eight most established HIE in U.S. includes Indiana Health Information Exchange, New England Health Exchange Network, Michiana Health Information Network, Colorado Regional Health Information Organization, Greater Houston's Health Connect, Health Bridge, Maine Health Info Net and Care Continuity Consortium. The feature that all of the eight HIEs have in common is they provide a platform for the participating hospitals to access and exchange data which is exceptionally helpful during an emergency situation.

### B. *Relationship between Distributed Database and Query Processing*

According to a research paper entitled [4] "*Distributed Databases Fundamentals and Research*" written by Dr H. Hakimzadeh from Department of Computer and Information Sciences, Indiana University South Bend, a distributed database (DBB) is a collection of multiple, logically interrelated databases distributed over a computer network. In the book entitled [5] "*Distributed Databases Principles & Systems*" by Stefano Ceri and Giuseppe Pelagatti, the definition of distributed database emphasizes two equally important aspects of a distributed database which is distribution and logical correlation. Distribution refers to the fact that the data are not resident at the same site (processor) so that people can distinguish a distributed database from a single centralized database.

Pankti Doshi and Vijay Raisinghani pointed out their research paper entitled [12] "*Review of Dynamic Query Optimization Strategies in Distributed Database*", the performance of a distributed database depends on how fast and efficiently data can be retrieved by query from multiple sites. Faster retrieval of data in a distributed database system is a complex problem since multiple sites are involved.

Several factors impact the performance of distributed query processing. These factors are selection of appropriate site (when same data is replicated at multiple sites), order of operation (such as select, project and join) and selection of join method (such as semi join, natural join, equi join etc). Query processing in a distributed database requires transfer of data from one computer to another through a communication network. Query at a given site might require data from remote sites. The complexity and cost increases with the increasing number of relations in the query. Thus, a query optimization is very much important in order to achieve energy efficiency as well as reduce operational cost as targeted in this project.

### C. *Importance of Energy Efficiency*

According to the research paper [18] "*Rethinking Query Processing for Energy Efficiency: Slowing Down to Win the Race*" by Willia Lang, Ramakrishnan and Jignesh M. Patel, energy management has become a critical aspect in the design and operation of database management systems. The emergence of this new paradigm as an optimization goal is driven by several factors. One of them is because of the tremendous amounts of energy consumed by a server which is 61B kilowatt-hours in 2006 and doubling by 2011. In addition, the energy component of the total cost of ownership for servers is high and growing rapidly. Besides that, some typical servers are over provisioned to meet peak demands; as a result, they are idle or underutilized most of the time. When servers are idle or nearly idle, they tend to consume energy that is disproportional to their utilization which is more than 50% of its peak power. With these rising energy costs and energy-inefficient server deployments, it is clear that there is a need to consider energy efficiency as a first class operational goal.

### D. *Energy Efficiency Benchmark*

Suzanne Rivoire, Mehul A. Shah and others pointed out in their paper entitled [19] "*Joule Sort: A Balanced Energy-Efficiency Benchmark*" that an energy-efficiency benchmark known as Joule Sort is proposed to drive the design of energy-efficient system. Joule Sort incorporates total energy which is a combination of power consumption and performance. Joule Sort is an I/O-centric benchmark that measures the energy efficiency of system at peak use by allowing comparison of energy efficiency of a variety of disparate system configurations.

According to Dimitris Tsirogiannis, Stavros Harizopoulos and Mehul A. Shah in [20] "*Analyzing the Energy Efficiency of a Database Server*", energy efficiency is defined as the ratio of useful work done to the energy used which is the same as the ratio of performance to power (Energy Efficiency = Work Done/Energy). As database software is rich in tunable parameters from system level constants to query planning and execution, these parameters can potentially affect the energy efficiency. Besides that, the energy efficiency of the database also affects its performance in various ways such as access methods, compressions, join algorithms as well as complex queries and join orderings.

### III. METHODOLOGY

#### A. Research Methodology

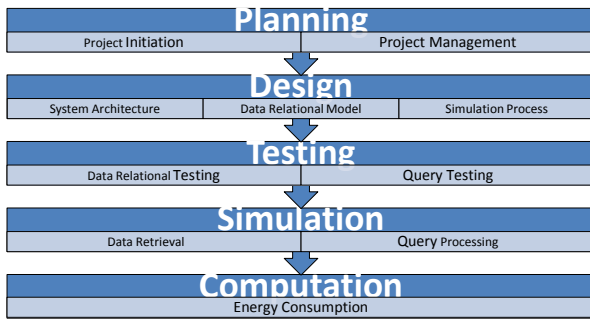


Figure 1: Prototyping Methodology Development Cycle

**Figure 1** shows the research methodology that is used for this project which is prototyping methodology. The reason being prototyping methodology is selected is because of the time constraint to complete this project which is less than seven months. Besides that, prototyping methodology allows continuous improvement throughout the development process which increases the quality of the deliverables. Since this is a research-based project which focuses on the research elements of green computing, prototyping methodology is chosen as it allows incomplete versions of the software program being developed.

#### B. System Architecture

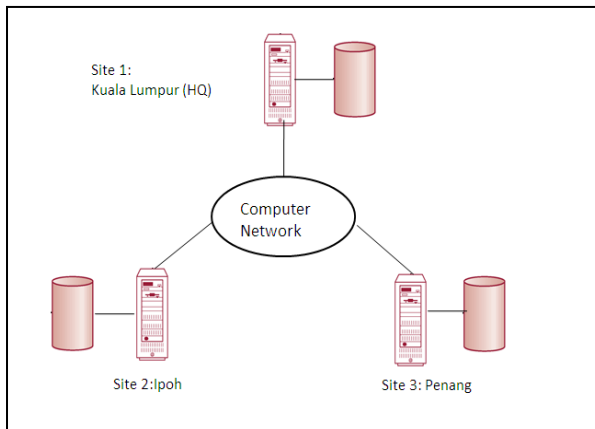


Figure 2: System Architecture

**Figure 2** above describes the architecture of the distributed database that will be set up for the simulation. Figure 3.2 above shows the system architecture of the distributed database that will be set up on a local computer network for data retrieval simulation. There are three databases involved in this set up, each labeled with the location of the hospitals starting from DB1 to DB3. Each of these databases has different IP address which indicates the dispersion of the database at different locations.

#### C. Simulation Process Design

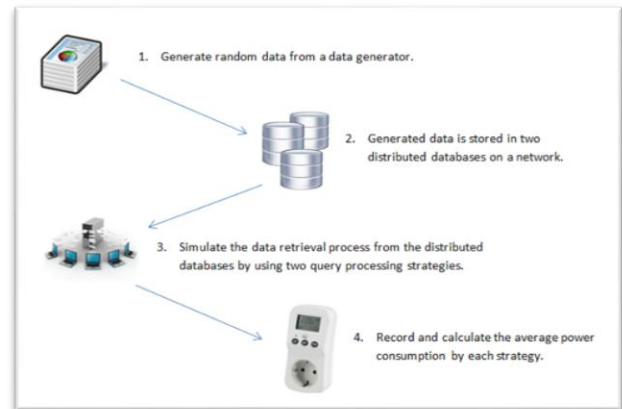


Figure 3: Simulation Process Design

**Figure 3** is the design of the simulation process which involves four main steps as below:

- i. First of all, generates random medical data from a data generator.
- ii. Secondly, store the generated medical data in the distributed database that has been set up.
- iii. Thirdly, conduct the simulation of the data retrieval process from each database by using different query processing strategies.
- iv. Compute the energy consumption by using power consumption calculator.

#### D. Hardware and Software required

- i. Energy consumption calculator
  - To calculate the power consumption by each query processing strategy in every simulation
- ii. PuTTY
  - To connect to the database
- iii. Microsoft Visual Studio
  - To simulate the process of remote access to data that stored in other database by propagating the data request to all the available servers

### IV. RESULT AND DISCUSSION

#### A. Simulation of Data Retrieval using Horizontal Fragmentation Strategy

Horizontal fragmentation is used to simulate the remote access of client (Site 2) to the data stored in server (Site 1). Assume that client side (Site 2) only stores patient information which is located at it site where field named "H\_ID"=Site 2. This simulation only involved retrieval of data from single table. The flow of the simulation processes are described as below.

1. Execution of query to retrieve data from each table at client site (Site 2).
2. Record the power consumption from the power meter when each query is executed.

To model the process of remotely access the data stored in another site, Microsoft Visual Studio is used to connect to the database. When a query requests for information is executed at one site which is known as client site (eg: Site 2), it will multicast the request to all available servers (Site 1 and Site 2). If Site 2 does not have the requested information

and other server (Site 1) has it, the server (Site 1) will return the information to the client site (Site 2).

Remote access happens when a query executed at client side (Site 2) but the result is returned by another server located at different physical location (Site 1). Remote access allows retrieval of distributed data from a database at another site without being physically execute the query at the site.

Below are the queries executed at Site 2 to remotely access data stored in each fragments located at Site 1.

**Transaction 1: Retrieval of patient information**

**Retrieve patient information where “H\_ID” = Site 1**

Fragment 1: Table “Patient” fragmented by hospital address (S<sub>1</sub>)

F<sub>1</sub>:  $\sigma_{H\_ID = 'Site 1'}(PATIENT)$   
SQL statement: SELECT \* FROM F1;

**Retrieve patient information where “H\_ID” = Site 2**

Fragment 2: Table “Patient” fragmented by hospital address (S<sub>2</sub>)

F<sub>2</sub>:  $\sigma_{H\_ID = 'Site 2'}(PATIENT)$   
SQL statement: SELECT \* FROM F2;

**Transaction 2 : Retrieval of doctor information**

**Retrieve doctor information where “H\_ID” = Site 1**

Fragment 3: Table “Doctor” fragmented by hospital address (S<sub>1</sub>)

F<sub>3</sub>:  $\sigma_{H\_ID = 'Site 1'}(DOCTOR)$   
SQL statement: SELECT \* FROM F3;

**Retrieve doctor information where “H\_ID” = Site 2**

Fragment 4: Table “Doctor” fragmented by hospital address (S<sub>2</sub>)

F<sub>4</sub>:  $\sigma_{H\_ID = 'Site 2'}(DOCTOR)$   
SQL statement: SELECT \* FROM F4;

**Transaction 3 : Retrieval of patient medical information**

**Retrieve patient medical information where “H\_ID” = Site 1**

Fragment 5: Table “PatientVisit” fragmented by hospital address (S<sub>1</sub>)

F<sub>5</sub>:  $\sigma_{H\_ID = 'Site 1'}(PATIENTVISIT)$   
SQL statement: SELECT \* FROM F5;

**Retrieve patient medical information where “H\_ID” = Site 2**

Fragment 6: Table “PatientVisit” fragmented by hospital address (S<sub>2</sub>)

F<sub>6</sub>:  $\sigma_{H\_ID = 'Site 2'}(PATIENTVISIT)$   
SQL statement: SELECT \* FROM F6;

**B. Simulation of Data Retrieval using Complete Replication Strategy**

Complete replication is used as a strategy that allows local access to the data stored in local database. In this simulation, the query is executed at Site 1. Assume that Site 1 contains a complete copy of information of itself as well as information stored at Site 2. This simulation involves retrieval of data from single table only.

The flow of the simulation processes are described as below.

1. Execution of query to retrieve data from each table at local server (Site 1).

2. Record the power consumption from the power meter when each query is executed.

Below are queries executed at Site 1 to locally access the replica of all data stored in its local server.

**Transaction 1: Retrieval of doctor information**

**Scenario 1: Retrieve doctor information at Site 1**

SQL statement: SELECT \* FROM DOCTOR WHERE H\_ID='Site 1';

**Scenario 2: Retrieve doctor information at Site 2**

SQL statement: SELECT \* FROM DOCTOR WHERE H\_ID ='Site 2';

**Transaction 2 : Retrieval of patient information**

**Scenario 1: Retrieve patient information at Site 1**

SQL statement: SELECT \* FROM PATIENT WHERE H\_ID='Site 1';

**Scenario 2: Retrieve patient information at Site 2**

SQL statement: SELECT \* FROM PATIENT WHERE H\_ID ='Site 2';

**Transaction 3 : Retrieval of patient medical information**

**Scenario 1: Retrieve patient medical information at Site 1**

SQL statement: SELECT \* FROM PATIENTVISIT WHERE H\_ID='Site 1';

**Scenario 2: Retrieve patient medical information at Site 2**

SQL statement: SELECT \* FROM PATIENTVISIT WHERE H\_ID ='Site 2';

**C. Simulation Result Analysis**

The simulation of the data retrieval process is started after the data has been fragmented and replicated as discussed earlier. Below are some important information about the specification of hardware and software used during the simulation.

Hardware / Software	Specification
i. Dell Inspiron 1420 Model	Intel Core 2 Duo CPU @ 2GHz, 2Gb RAM
ii. LAN bandwidth	Maximum 100Mb/s
iii. Server (boinc and nativeboinc)	Intel Xeon CPU X5550 @2.67GHz

The power consumption captured by the power meter during the execution of query for the simulation above are tabulated into table to calculate the average as well as weighted average. Two equations that are used to calculate the average and weighted average are:

i. Equation 1  
 $Average = \sum A/N$   
 where A: power consumption for each query  
 N: number of query execution

ii. Equation 2  
 $Weighted\ Average = 100\% \times \sum (B/Total\ of\ B)$   
 where B: Total in average of all tables

Below is the table that shows the power consumption by each query processing strategy and the calculation of average as well as weighted average.

No. of query execution (N)	Table accessed	Power Consumption for horizontal fragmentation in watt (A)	Power Consumption for complete replication in watt (A)
1	Patient	39.4	29.1
	Doctor	36.5	28.7
	PatientVisit	39.6	30.4
2	Patient	38.6	29.1
	Doctor	36.9	28.7
	PatientVisit	42.3	30.7
3	Patient	40.2	29.2
	Doctor	37.7	28.5
	PatientVisit	41.6	31.7
4	Patient	39.8	30.5
	Doctor	37.2	29.3
	PatientVisit	43.1	32.1
5	Patient	41.2	29.6
	Doctor	37.5	28.1
	PatientVisit	42.6	31.9
Average ( $\sum A/N$ )	Patient	39.8	29.5
	Doctor	37.2	28.6
	PatientVisit	41.8	31.4
Total power in average (B)		39.6	29.8
Percentage of total power consumption (B)		57%	43%

Table 1: Power Consumption for Simulation Using Both Query Processing Strategies

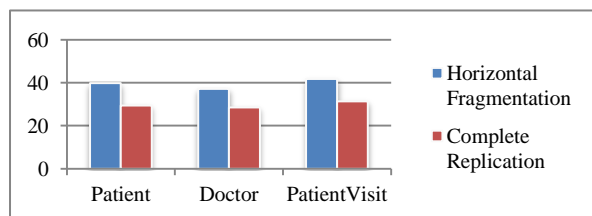


Figure 4: Average Power Consumption for each table

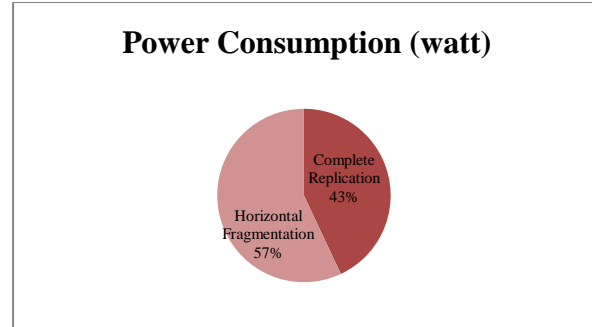


Figure 5: Weighted Average Power Consumption

Figure 4 shows the average power consumption used by each strategy to access different tables. These average power consumptions are translated into Figure 5 which shows the total power consumption used by each strategy during data retrieval.

Based on Table 1, the average of power consumption by remote access (horizontal fragmentation strategy) to patients' information during the simulation is 39.6 watt while local access (complete replication strategy) to patients' information averaged 29.8 watt of power. The difference in power consumption between two strategies is about 10 watt. It can be clearly seen that local access which used complete replication as the query processing strategy consumes lesser energy as compared to that of remote access which used horizontal fragmentation.

Figure 5 shows that horizontal fragmentation strategy used up 57% of the total power consumption during the simulation while complete replication strategy only used 43% of the total power consumption. The difference of 14% in the power consumption between two strategies is equivalent to 10 watt of power. In terms of energy, 10 watt is equivalent to 10 joules per second. 10 joules seems to be a small amount, however in a long run and running on a huge amount of data, a savings in 10 joules can make a big difference to the environment as well as the business operational cost.

Based on the simulation result which used complete replication and horizontal fragmentation to model the local access and remote access, it can be said that complete replication that enables local access to the data stored in distributed database consumes lesser energy as compared to remote access. Lesser energy consumption also means higher efficiency in query performance as lesser time is taken to execute a query and get reply from the server.

## V. CONCLUSION AND RECOMMENDATION

In conclusion, complete replication is an energy-efficient query processing strategy which allows local access to the replica of database of other sites at its own database. Besides being energy-efficient, complete replication reduces the time taken to retrieve the requested data from a distributed database. Nonetheless, the cost of replication to store and maintain the updated as well as complete copy of the distributed database at each site in the local database is expensive for healthcare institutions. Thus, the healthcare institutions have to make a balance between the reductions

in power consumption and cost allocation to implement a complete replication.

However, this set of result is acceptable to certain extent only referring to the specification of hardware and software used in the simulation as discussed earlier. In real life, each healthcare institution uses different type of computer models with varying processing speed, thus the power consumption during data retrieval process may varies as well.

It is recommended to extend this research to multi-table queries to make this research more realistic. Add on to that, this simulation model the data retrieval process from table which only contains a maximum of 1000 data. It is good if this research can expand the scope to a higher amount of data to suit the real life situation in a healthcare institution.

Besides that, it will be great if the research can be expanded to other industries which require high efficiency in data retrieval process such as banking industry. With the expansion of the project on other industries, the advantages of green computing not only benefit the industry but our precious mother nature as well.

#### ACKNOWLEDGEMENT

First and foremost, the writer would like to express my greatest gratitude to her project supervisor, Miss Rozana for her professional assistance and guidance throughout the development of this project. Her encouragement and support played a big part to ensure the success of this project. Other than that, she would also like to thanks her lab tutor, Mr Jamal for teaching her Microsoft Visual Studio and guided me during the simulation of this project. This gratitude also dedicated to her friends and family who have given all their support to motivate her. Last but not least, precious thanks to the committee of Final Year Project of Computer Information Sciences (CIS) department of UniversitiTeknologi PETRONAS (UTP) for the guidelines and assistance provided throughout the semester.

#### REFERENCES

[1] "Fiona Caldicott to lead review into sharing of health information"  
Retrieved 21<sup>th</sup> November 2012 from:  
[www.guardian.co.uk](http://www.guardian.co.uk)

[2] Health Data Management  
Retrieved 21<sup>th</sup> November 2012 from:  
[www.bridgeheadsoftware.com](http://www.bridgeheadsoftware.com)

[3] "8 Health Information Exchange Leading the Way"  
Retrieved 21<sup>th</sup> November 2012 from:  
[www.informationweek.com](http://www.informationweek.com)

[4] Dr H. Hakimzadeh, "Distributed Databases Fundamentals and Research", Department of Computer and Information Sciences, Indiana University South Bend

[5] Stefano Ceri & Giuseppe Pelagatti (1985): *Distributed Databases Principles & Systems*: McGraw-Hill Book Company Publisher

[6] Thomas Connolly & Carolyn Begg (2010): *Database Systems – A Practical Approach to Design, Implementation and Management*: Pearson Education International Publisher

[7] Korth and Sudarshan (2010): "Database System Concepts": McGraw-Hill Book Company Publisher

[8] R. Elmasri and S.B. Navathe (2004): "Principle of Database Query Processing" 4<sup>th</sup> Edition: Addison-Wesley Publisher

[9] Alan R. Hevner and S. Bing Yao, *Query Processing in Distributed Database System*, IEEE Transactions on Software Engineering, Vol. SE-5, No.3

[10] Michael L. Rupley, Jr., *Introduction to Query Processing and Optimization*, Indiana University at South Bend

[11] Query Processing  
Retrieved 21<sup>th</sup> July 2012 from: [www.en.wiktionary.org](http://www.en.wiktionary.org)

[12] Pankti Doshi & Vijay Raisinghani, "Review of Dynamic Query Optimization Strategies in Distributed Database", Department of Computer Science and Information Technology, Mukesh Patel School of Technology Management and Engineering, NMIMS Deemed-to-be University

[13] San Murugesan (2008): "Harnessing Green IT: Principles and Practices", Volume 10, Issue 1

[14] Green Computing: What is Green Computing?  
Retrieved 21<sup>th</sup> June 2012 from:  
<http://greenelectronics.com/FAQRetrieve.aspx?ID=31973&O=>

[15] Rajguru P.V, Nayak S.K and More D.S, *Solution for Green Computing*, Department of Computer Science and IT, Adarsh college, Hingoli (Maharashtra), India

[16] Data Center Power Consumption  
Retrieved 21<sup>th</sup> November 2012 from:  
[www.bridgeheadsoftware.com](http://www.bridgeheadsoftware.com)

[17] The Star, "Green computing as a way to reduce IT operation costs", Monday, 21<sup>st</sup> of July 2008

[18] Willis Lang, Ramakrishnan Kandhan, Jignesh M. Patel, *Rethinking Query Processing for Energy Efficiency: Slowing Down to Win the Race*, Computer Sciences Department, University of Wisconsin, Madison

[19] Suzanne Rivoire, Mehul A. Shah, Parthasarathy and Christos Kozyrakis, *JouleSort: A Balanced Energy-Efficiency Benchmark*, Stanford University

[20] Dimitris Tsirogiannis, Stavros Harizopoulos, Mehul A. Shah, *Analyzing the Energy Efficiency of a Database Server*, University of Toronto