FAIRNESS CATEGORIZATION POLICY OF QUEUING THEORY FOR GEOGRAPHIC INFORMATION SYSTEM JOB SCHEDULING

KHEOH HOOI LENG

UNIVERSITI SAINS MALAYSIA

2013

FAIRNESS CATEGORIZATION POLICY OF QUEUING THEORY FOR GEOGRAPHIC INFORMATION SYSTEM JOB SCHEDULING

by

KHEOH HOOI LENG

Thesis submitted in fulfilment of the requirements for the Degree of Master of Science

May 2013

ACKNOWLEDGEMENTS

In the journey of pursuing Masters, I have gained a lot of rewarding experiences from both academically and personally. Thus, I wish to take this opportunity to express my appreciations towards a few individuals who have played important roles in my research life.

Initially, I would like to convey my sincere gratitude to my supervisor, Associate Professor Dr. Chan Huah Yong. He has given this thesis his utmost attention by providing many constructive and valuable suggestions to my research work. Besides, his assistance and supervision as well as inspiration throughout the course of my research will never be forgotten.

Next, I wish to show my heartfelt appreciation to School of Computer Sciences, Universiti Sains Malaysia for providing a conducive environment during the course of my research. Additionally, seminars that are conducted by the expertise have been carried out from time to time could definitely help in my research studies.

Meanwhile, I would like to thank my lab mates in Grid Computing Lab for their advices and information sharing throughout the course of this research. Finally, I wish to sincerely thank my family and friends especially my beloved one for their encouragement and support. Without them, I would never be able to undertake this journey.

TABLE OF CONTENTS

CHAPTER 3 RESEARCH METHODOLOGY

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

λ	Average arrival rate
μ	Average service rate
e	Exponential
T_n	Each job processing time
X_n	I/O time for transferring jobs from queues to machines
L_n	I/O time for loading jobs within the machines for execution
R_n	I/O time for sending jobs back to client as output
BST	I/O time for transferring a batch of jobs from queues to machines
LT	I/O time for loading jobs within the machines for execution
RT	I/O time used for sending jobs back to client as output
E_n	Job execution time
ET	Job execution time
$job_{granularity}$	Job granularity
N	Mean number of customers in the system
$E(B)$	Mean service time
$E(L^q)$	Mean number of customers in the queue

$E(N)$	Mean number of customers in the system
$E(S)$	Mean number of customers in the server
$E(T)$	Mean turnaround time
$E(W)$	Mean waiting time
c	Number of cores in the machine
$OptBS$	Optimum number of jobs that fit in a batch
$P_n(t)$	Probability
L^q	Queue length
$\sum$	Summation
t	The interval 0 to t
$total\_CPU\_time$	Total CPU processing time
$total\_IO\_time$	Total I/O time
n	Total number of arrivals in the interval 0 to t
N	Total number of jobs
S_N	Total time used for a list of jobs processing
T	Total waiting time
ρ	Traffic intensity / occupancy
W	Waiting time

LIST OF ABBREVIATIONS

ACO	Ant Colony Optimization
API	Application Programming Interface
BLBD	Based on Load Balancing and Demand
CGI	Common Gateway Interaction
CPU	Central Processing Unit
DCOM	Distributed Component Object Model
ESRI	Environmental System Research Institute
FCFS	First-Come First-Served
FCQS	Fair Categorized Queue Scheduling
FIFO	First-In First-Out
FPFS	Fit Processors First Served
FPMPFS	Fit Processors Most Processors First Served
FPLPFS	Fit Processors Least Processors First Served
GA	Genetic Algorithm
GIS	Geographic Information System
GML	Geography Markup Language
GPS	Global Positioning System

GRS	Grid Resource Scheduler
GSA	Genetic Simulated Annealing
GUI	Graphic User Interface
HTML	HyperText Markup Language
I/O	Input / Output
JDK	Java Development Kit
JSch	Java Secure Channel
JSP	Java Server Pages
LCFS	Last-Come First-Served
LIM	Load Information Manager
LSF	Load Sharing Facility
LSLIB	Load Sharing Library
MCT	Minimum Completion Time
MET	Minimum Execution Time
MySQL	My Structured Query Language
MQMM	Multiple Queues Multiple Machines
MQSM	Multiple Queues Single Machine
OLB	Opportunistic Load Balancing
OS	Operating System

PBS Pro	Portable Batch System Professional Edition
PSO	Particle Swarm Optimization
QoS	Quality of Service
RES	Remote Execution Server
SA	Simulated Annealing
SIG	Spatial Information Grid
SIRO	Service In Random Order
SPT	Shortest Processing Time
SQMM	Single Queue Multiple Machine
SQSM	Single Queue Single Machine
TS	Tabu Search
3D	Three-Dimensional
2D	Two-Dimensional

DASAR PENGKATEGORIAN KEADILAN BAGI TEORI BARIS GILIR UNTUK PENJADUALAN KERJA SISTEM MAKLUMAT GEOGRAFI

ABSTRAK

Sistem Maklumat Geografi (GIS) ialah satu aplikasi intensif dalam pengkomputeran dan data yang berurusan dengan pemprosesan sejumlah besar data spatial dan persembahan imej tiga dimensi (3D) bagi lokasi-lokasi. Selain kerja-kerja penyelidikan yang dijalankan dari segi pemprosesan data atau imej bagi aplikasi GIS, penjadualan beban kerja GIS juga boleh dikaji untuk meningkatkan prestasi aplikasi-aplikasi GIS. Tesis ini mencadangkan satu algoritma penjadual kerja bernama Fair Categorized Queue Scheduling (FCQS) yang dapat mengedarkan kerja-kerja GIS dengan cekapnya. Teori baris gilir digunakan dalam FCQS untuk proses penjadualan kerja manakala ketibaan kerja-kerja GIS diedarkan mengikut taburan Poisson. Setiap kategori kerja-kerja dilayan berdasarkan asas Datang-Dahulu Dilayan-Dahulu (FCFS) dengan konfigurasi Barisan Berbilang Mesin Berbilang (MQMM). Eksperimen melalui simulasi telah dijalankan untuk menilai prestasi FCQS dan konfigurasi-konfigurasi beratur yang lain seperti Barisan Tunggal Mesin Tunggal / Berbilang (SQSM / SQMM) dan Barisan Berbilang Mesin Tunggal / Berbilang (MQSM / MQMM). Keputusan membuktikan bahawa FCQS mencapai daya pemprosesan yang tertinggi dengan sebanyak 24 kerja atau 72.727% lebih daripada konfigurasi Barisan Tunggal Mesin Tunggal yang mempunyai daya pemprosesan terendah. Tambahan pula, jumlah masa pemindahan Input / Output (IO) dapat dikurangkan sehingga sebanyak 49.194% dengan menggunakan pemprosesan kerja berbilang berbanding dengan pemprosesan kerja tunggal untuk kerja-kerja yang berjenis kecil. Purata masa pemulihan dan masa menunggu yang lebih rendah juga dapat dicapai secara serentak.

Akhirnya, pengoptimuman sumber grid telah ditingkatkan dengan ketara iaitu mengurangkan jumlah kerja menunggu ke 28.261% dibandingkan dengan yang tertinggi 52.174%.

FAIRNESS CATEGORIZATION POLICY OF QUEUING THEORY FOR GEOGRAPHIC INFORMATION SYSTEM JOB SCHEDULING

ABSTRACT

Geographic Information System (GIS) is a compute-intensive plus data-intensive application that deals with substantial amount of spatial data processing and rendering of three-dimensional (3D) images of the locations. Besides research work on data or image processing part of GIS applications, scheduling of GIS workload can be further studied to improve the performance of GIS applications. In this regards, this thesis proposes an algorithm of job scheduler named Fair Categorized Queue Scheduling (FCQS) which distributes jobs of GIS applications efficiently. Queuing theory is applied in FCQS for job scheduling processes meanwhile the GIS job arrivals are distributed according to Poisson distribution. Each category of jobs is served along with First-Come First-Served (FCFS) basic using the Multiple Queues Multiple Machines (MQMM) configuration. The experiment through simulation has been carried out to evaluate the performance of FCQS and other queue configurations such as Single Queue Single / Multiple Machine(s) (SQSM / SQMM) and Multiple Queues Single / Multiple Machines(s) (MQSM / MQMM). The results proved that the FCQS algorithm achieved the highest throughput with 24 jobs or 72.727% more than the lowest throughput of SQSM. Additionally, the total Input / Output (IO) transferring time can be reduced up to 49.194% by using multiple jobs processing compared to single job processing within small jobs, attaining lower average turnaround time and waiting time simultaneously. Last but not least, the optimization of grid resources has been significantly improved by decreasing total pending jobs to 28.261% instead of the highest 52.174%.

CHAPTER 1

INTRODUCTION

1.1 Introduction

Grid enables the virtualization of distributed computer and data resources to create a single system image, granting users and applications seamless access to its powerful computational capabilities. Grid computing can be considered as an economical super-computing (Al-Khannak & Bitzer, 2007; Foster & Kesselman, 1999) that facilitates communication across heterogeneous and geographically dispersed environment besides optimizing computing and data resources, pooling them for large capacity workloads. In addition, grid system has five features that include heterogeneity, scalability, adaptability, structure unpredictability and multi-level domain management (Xiaosheng et al., 2009). Therefore, grid computing is an ideal option for providing massive data storage, intensive computational processes and computing resources that are obligatory criteria in Geographic Information System (GIS) applications.

Nowadays, GIS is growing dramatically with the advancement of computer technologies and facilities of Internet. GIS is a compute-intensive plus data-intensive application. Its compute-intensive tasks are rendering three-dimensional (3D) images of the locations meanwhile data-intensive tasks are dealing with voluminous information processing of GIS data. Lots of GIS applications exist in the market to fulfill user demands such as the Google Earth and the Google Maps by Google, the Bing Maps by Microsoft and the ArcGIS by Environmental System Research

Institute (ESRI). GIS applications are not only used to search locations around the world but also to illustrate the available routes for travelling and location finders. Moreover, they have the capability to show some buildings around the world in detailed 3D structures instead of two-dimensional (2D) structures.

GIS applications can be occupied in assorted fields as their databases comprise a wide range of information embracing geographic, social, political, environmental and demographic (Do-Hyun et al., 2001). The connection authority of GIS is that maps can be drawn from databases and spatial data can be referenced from the maps (Chengda et al., 2001; Do-Hyun et al., 2001). Furthermore, GIS applications assists in organizing and analyzing spatial information then output the relevant results which are useful to the users. For examples, the time critical applications such as emergency service decision support, military command and control requires high performance GIS and visualization applications such as flight simulators need GIS to represent actual terrain as well (Fang et al., 2007).

As a result, scheduler that is part of computational grid is essential in GIS applications. A quality scheduler plays an important role in dispatching jobs to the resources meanwhile enabling GIS applications to perform well and satisfy user requests at any time. There are varieties of optimization solutions presented for existing scheduling methods to be fitted in different kind of applications. Therefore, the goal of this thesis is to illustrate a more data load and visualization scheduler that is designed mainly for GIS applications. This scheduler is aimed to balance the job loads and perform effectively in computational resources environment. Moreover, it is capable to accomplish its requirements even when dealing with voluminous user requests for spatial information simultaneously.

1.2 Background

Formerly, job scheduling is handled manually by the grid administrators. Hence, acquaintance staffs in the grid environment are required to oversee the overall procedures from configuration until job submission by utilizing their job scheduling policy. Though, these days jobs scheduling is done by a scheduler. A scheduler is responsible to perform the scheduling tasks whereby it decides and delegates the jobs to the resources among varieties of possible jobs by adapting its own scheduling policy. Therefore, the reveal of schedulers could certainly ease grid administrators in arranging jobs to the resources.

Basically, there are few basic scheduling algorithms available and applied in current scheduler such as First-Come First-Served (FCFS) or First-In First-Out (FIFO), round robin, shortest job first, largest job first, preemptive algorithm, priority based scheduling, multi-level queues, multiple processor scheduling and real time scheduling (Jerry, 2009; Siddesh & Srinivasa, 2011). All of these basic scheduling algorithms have their advantages and disadvantages. Nonetheless, the types of scheduling algorithms to be employed are totally depending on the types of applications. Each kind of applications has its own fundamental requirements to be achieved that are diverse from other kind of applications.

Job scheduling becomes more complicated when it is utilized in data-intensive and compute-intensive applications such as GIS applications. It is challenging as these applications have to deal with extreme volume of data and processing load from time to time as well as satisfy the user requirements. In order to address the challenges, there are couples of scheduling issues and evaluation criteria that need to be taken into considerations while evaluating the scheduler's performance. For instance, these criteria are utilization, fairness, overhead, throughput, turnaround time, waiting time, response time and residence time (Insup & Dianna, 2003; Tanenbaum, 2005).

Consequently, job scheduler is mandatory in grid environment for scheduling the incoming jobs submitted from application such as GIS application. A quality job scheduler can accomplish the basic requirements of the GIS applications by delegating jobs to well-fit each machine capability.

1.3 Research Problem

In order to cope with numerous user requests concurrently, GIS applications need a competent job scheduler that is capable of scheduling GIS jobs well and gratifying user demands periodically. Therefore, the connection with grid computing to gain huge data storage and demanding resources is essential. However, the involvement of GIS applications in distributed environment together with well-fitted scheduler has its own challenges.

Therefore, several problems encountered and intended to be solved in this research are as follow:

- How to optimize resources usage meanwhile well-balanced its workload in distributed environment;
- How to prioritize fairness among GIS jobs upon their arrivals;

- How to decrease I/O time while scheduling GIS jobs and increase overall throughput at the same time.

1.4 Research Objectives

The purposes of this thesis are to study and discover a superior way for managing issues in scheduling GIS workload. The objectives of this research are as mentioned below:

- To investigate the nature of resources by running applications mainly GIS that are data-intensive in storing all the information of locations and compute-intensive in rendering 3D images of related locations in order to optimize resources usage and well-balanced the workloads;
- To build and evaluate a high scalable job scheduler for GIS applications using queuing theory with fairness categorization policy;
- To reduce job I/O time by exploiting multiple jobs processing meanwhile improving the throughput of GIS applications.

1.5 Importance and Significance of the Research

In this technology era, most of the individuals own an assortment of mobile devices such as smart phones, tablets and laptops. Essentially, these mobile devices are embedded with GIS applications or Global Positioning System (GPS) to ease users in getting location and routes information. As a result, there is a high possibility whereby GIS applications have to deal with massive user requests for geographical information at the same time. Thus, an effective scheduler is vital in scheduling all the incoming user requests, ensuring that the GIS applications able to cope with numerous user requests and response to them in the shortest period of time.

1.6 Research Scope

The main concern of this thesis is limited to the GIS job scheduling and its overall processing performance. Also, the load demands of GIS applications that deal with voluminous user requests simultaneously will be taken into considerations in order to turn out a high throughput GIS application. The scopes of this research are as follows:

- A framework of GIS which is data-intensive and compute-intensive, performing well in resources management and workload allocation;
- An application of high scalable GIS utilizing job scheduler in distributed environment with enhanced performance characteristic and load demands;
- A job scheduler that is designed to suit GIS applications, prioritizing fairness between each category of GIS jobs while reducing I/O time and enhancing throughput;
- A simulated environment that is created to evaluate our proposed methodology.

1.7 Contribution

The contribution of this thesis is to present a novel way of scheduling jobs using queuing theory for GIS applications to grid resources in an improved performance characteristic and load balancing manner. The proposed scheduling algorithm is configured in Multiple Queues Multiple Machines (MQMM) environment with the combination of FCFS and priority services that prioritizes fairness of each job category while scheduling. It includes the embedded feature of multiple jobs submission as well. Moreover, the usage of grid resources can provide vast capacity for geospatial data analysis and handling, facilitating heterogeneous data sharing. As a consequence, the proposed solution in scheduling GIS jobs can ease GIS applications that are data-intensive and compute-intensive in coping with voluminous user requests efficiently. The contributions of this thesis can be measured as follows:

- Identification of scheduling technique for scheduling GIS applications jobs with the introduction of categorization technique for GIS jobs.
- Identification of queues classification and machines assignments as well as queue discipline that prioritizes fairness for each category of GIS jobs with the introduction of multiple jobs submission feature for scheduling technique.

1.8 Organization of Thesis

This thesis consists of six chapters namely Introduction, Literature Review, Research Methodology, the Implementation, Result and Discussion as well as Conclusion and Future Work.

In this chapter, the overall idea of this thesis that includes the general introduction and backgrounds of our research work are presented. Next chapter, the literature review covers the overview of job scheduling methods. Several existing job scheduling techniques are discussed as well as their advantages and disadvantages. Also, several existing data storage and image rendering techniques that are used in GIS applications are elaborated. The overall picture of queuing theory that is to be adapted in our job scheduling algorithm is presented. Meanwhile, the study of software involved is described in related work too.

In chapter 3, an algorithm and a method which are utilized in implementing job scheduling methods for GIS applications are presented. Additionally, details of the algorithm are elaborated in this chapter as it is the core of this thesis. The newly proposed method for scheduling the GIS workload together with the explanations on how it can help in scheduling GIS jobs efficiently are presented. Also, the evaluation methods that will be used for evaluating the testing results are explained.

In chapter 4, the details design of the implemented algorithm is proposed. The details of hardware specifications and software needed in the implementation are discussed. In addition, the classes and functions involved in our algorithm are described briefly. Then, the steps of how this algorithm is able to help in job scheduling are illustrated in pseudo code.

In chapter 5, the analytical evaluation, testing scenarios and evaluation results as well as qualitative discussion for the proposed scheduling method are presented. Some criteria are taken into consideration in the experiment such as throughput, CPU processing time, I/O time, CPU idle time, average waiting time and average turnaround time. Also, the experiments and scenarios that have been carried out are discussed in details. Finally, the analysis of the obtained results is presented.

In the last chapter, the research work is concluded meanwhile suggesting the future direction of how the research can be carried on in future.

CHAPTER 2

LITERATURE REVIEW

2.1 Overview

This chapter reviews the existing research work that is relevant to our field of study. The importance of integrating grid computing and GIS applications is introduced throughout this chapter. In such a way, a job scheduler and its scheduling method play a crucial role in handling GIS application jobs. For this reason, a scheduling algorithm is proposed in this thesis by adapting queuing theory that is capable of prioritizing fairness of each job category in scheduling meanwhile balancing the load of resources. Additionally, several identified scheduling algorithms and their usage in variety fields are explained in order to provide a better understanding on their wellness and limitations.

Then, the section proceeds with existing data storage, image rendering techniques and job schedulers that are currently utilized or could be adapted in GIS applications. The pros and cons of the techniques are explained as well. On the other hand, the concept of queuing theory along with the exploitation of queuing theory in diverse fields or even in our daily life together with the involved software such as Blender and My Structured Query Language (MySQL) are presented in related work section. As a whole, the rational of using queuing theory in our job scheduling method are justified.

2.2 Grid Computing

Grid computing facilitates the allocation and integration of computing resources through networks. In addition, it creates a dynamic and multi-institutional environment for grid users, stimulating co-operation among users and organizations. Primarily, grid can be classified in proportion to its utilization purposes (Foster et al., 2001). There are some common categorizations of grid; for instance, data grid, sensor grid and computational grid.

A sensor grid incorporates a collection of sensors with grid to enable real-time sensor data collection and processing by sharing of resources. Meanwhile, a data grid provides users with data storage and data retrieval as well as facilitates modification and transfer of large amounts of geographically dispersed data. Alternatively, a computational grid is a hardware and software infrastructure that provides users with consistent, dependable, economical and pervasive access to high computational capabilities (Foster & Kesselman, 1999). It combines many computing resources into a network for executing computational tasks (Xhafa et al., 2010).

Scheduling that can be considered as a branch of computational grid plays an important role in delegating jobs to grid resources. The characteristics of grid make scheduling become more challenging in a computational grid environment (Tao et al., 2006). For this reason, many research works have been performed from jobs or resources perspectives in order to enhance the scheduling performances and balance the workload in resources. Assortment of existing scheduling algorithms together with its advantages and disadvantages are further discussed.

2.2.1 Resource Scheduling

It is challenging in finding an optimal scheduling solution that is capable to efficiently schedule computational jobs in a distributed heterogeneous computing environment. However, heuristics derived from the nature has illustrated a surprising degree of effectiveness and generality in handling the challenges (Abraham et al., 2000). Current scheduling techniques that utilize heuristics approach can be categorized into two types: static scheduling and dynamic scheduling.

Static scheduling algorithms are useful in predictive as well as post-mortem analyses, impact studies, overcoming dynamic scheduling problems and fulfilling several different requirements in a system (Braun et al., 2001; Ritchie & Levine, 2003). A comparison and summarization of 11 static heuristics algorithms such as Opportunistic Load Balancing (OLB), Minimum Execution Time (MET), Minimum Completion Time (MCT), Min-min, Max-min, Duplex, Genetic Algorithm (GA), Simulated Annealing (SA), Genetic Simulated Annealing (GSA), Tabu Search (TS) and A* for scheduling jobs in heterogeneous computing environment are provided by Braun et al. (Braun et al., 2001). Also, they discovered GA consistently performs better in comparisons to the other techniques in their given scenarios.

After surveying the existing optimization solutions, GA is recommended to be the best algorithm for decentralized scheduling similar to Braun et al. (Braun et al., 2001). In this regards, the authors have proposed a decentralized scheduling method based on GA and cooperation among scheduling agents (Pop et al., 2008). Moreover, Vivekanandan et al. presented a comparative study between Ant Colony Optimization (ACO) and Particle Swarm Optimization (PSO) for grid scheduling and concluded that PSO has better performance compared to ACO by using makespan as the performance measurement (Vivekanandan et al., 2010).

GA is an adaptive method that is based on genetic process of biological organisms in resolving schedule optimization problem. It is also proven that GA is competent to perform better against other scheduling techniques (Abraham et al., 2000). Thus, Abraham et al. addressed the hybridized usage of three nature's heuristics which are GA, SA and TS for scheduling jobs in computational grid environment (Abraham et al., 2000). In addition to the beneficial gained from GA, the hybridization of GA-SA algorithm provides a better convergence whereas GA-TS algorithm improves the search efficiency compared to purely GA.

On the other hand, dynamic scheduling can be grouped into two modes: immediate mode and batch mode heuristics. Tasks in immediate mode are mapped to the resources upon arrivals whereas the tasks in batch mode are needed to wait for a period of time for collecting into a batch first then only mapped to the resources (Maheswaran et al., 1999; Xhafa et al., 2010). Besides, Maheswaran et al. revealed the arrival rate of tasks as well as the structure of heterogeneity between tasks and resources are those two important aspects that need to be taken into consideration while choosing an ideal dynamic scheduling to fit in particular situation (Maheswaran et al., 1999).

Tao et al. portrayed an adaptive resource scheduling algorithm known as the Based on Load Balancing and Demand (BLBD) for computational grid environment (Tao et al., 2006). The novelty of the work is the BLBD considers both system load and personal resource requirements such as cost and Quality of Service (QoS) of node in order to dispatch jobs to the most appropriate grid node. Also, Ritchie and Levine suggested a simple local search procedure in combination with either static or dynamic heuristics algorithm in order to figure out the best known makespans for some benchmark problems in a short period of time, improving solutions significantly (Ritchie & Levine, 2003).

2.2.2 Job Scheduling

Most of the conventional job scheduling schemes in space-sharing schedulers utilize simple FCFS approach in scheduling the incoming jobs. However, there are some drawbacks in FCFS method such as resources fragmentation (Dmitry & Peter, 1999) and inefficiency in processors utilization (Kento et al., 1998). Therefore, Kento et al. proposed an approach called Fit Processors First Served (FPFS) to overcome the shortage of FCFS (Kento et al., 1998). The FPFS applies two sorting techniques either Fit Processors Most Processors First Served (FPMPFS) or Fit Processors Least Processors First Served (FPLPFS) then dispatches jobs that fit idle processors by searching them from the job queue.

Backfilling is a simple scheduling technique that is capable of enhancing the utilization of space-sharing schedulers effectively and decreasing average slowdowns for batch schedulers (Dmitry & Peter, 1999; Barry & Evgenia, 2002; Peter et al., 2000; Kento et al., 1998). It allows jobs to move ahead in the queue without delaying the start time of the first job in the job queue. Due to the advantages of backfilling as mentioned above, Barry and Evgenia described a job scheduling policy based on backfilling while making use of multiple job queues as well as different job priorities and job reservations for scheduling jobs on parallel system (Barry & Evgenia, 2002). After monitoring, the policy is able to outperform traditional backfilling consistently.

In backfilling, users are required to estimate the execution time for each submitted job. Previous researches have proved that inaccuracy in estimating job execution times may lead to better performance in backfilling scheduler (Peter et al., 2000). Thus, Dmitry and Peter suggested that existing backfilling schedulers could be improved by mutating estimated execution times systematically and sorting jobs by increasing job execution time upon arrivals (Dmitry & Peter, 1999). By adapting this method, the average slowdowns can be reduced effectively meanwhile avoiding starvation. Researchers continue their efforts in eliminating the bottlenecks of backfilling schedulers. Peter et al. showed that queue randomization and queue sorting could further improve the performance of current backfilling schedulers in comparisons to the adaption of queue sorting by job length alone with actual estimation of job execution times (Peter et al., 2000).

Furthermore, Jaspal et al. evaluated different scheduling optimizations such as backfilling and waiting queue sorting to identify the influential factors in forming a rational basis for designing and making best use of scheduling systems (Jaspal et al., 1996). Meanwhile, Siddesh and Srinivasa proposed a scheduler framework that known as Grid Resource Scheduler (GRS) by utilizing the combination of Job Rank-Backfilling policy as its job scheduling algorithm as well as best fit allocation model as its resource matching algorithm (Siddesh & Srinivasa, 2011). As a conclusion, the main issues that need to be considered in scheduling policy include the characteristic of job mix, efficiency of machine usage and fairness in scheduling.

2.3 GIS Applications

GIS has been known as “Intelligent Mapping” as it includes the combination of influential database features and location analysis of a digital map (Missouri Economic, 2007). GIS integrates hardware, software and data, obtaining layers of information such as building, soil, street, river and transportation that ties to locations on the earth then overlays them for analyzing and displaying purposes. Figure 2.1 below shows spatial data that are represented in a set of thematic layers to compose an informative map.

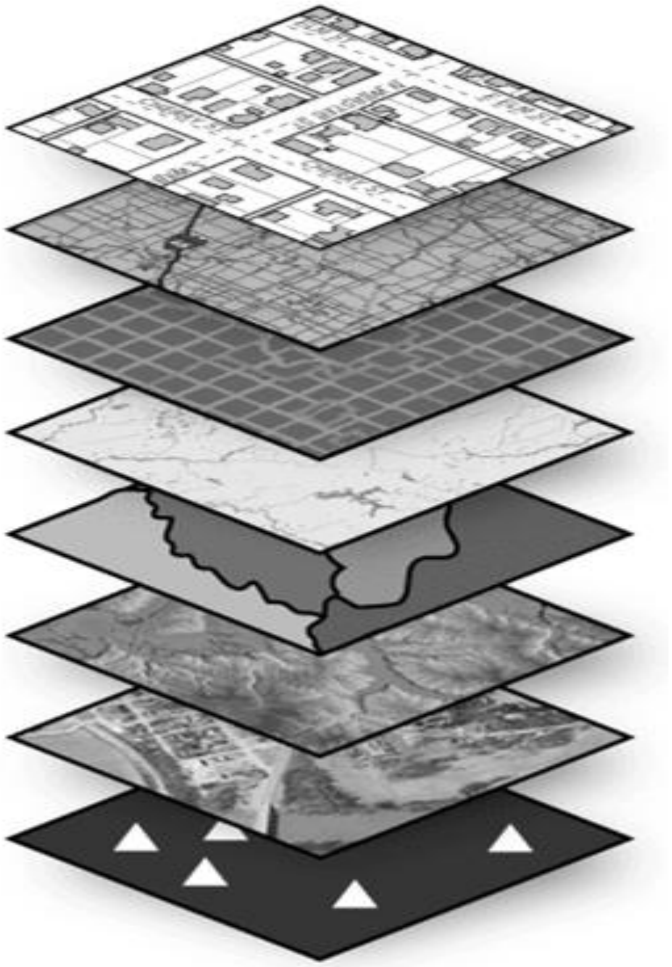

Figure 2.1: Thematic layers for representing GIS spatial (Indiana Geographic Information Council, 2007)

The rapid development of GIS through computer technology has brought forward higher demands towards GIS applications and GIS software. Researchers put efforts in researching new solutions for next generation GIS in order to handle the traditional GIS shortcomings such as inadequacy of spatial data handling and analysis capacity, difficulties in sharing of data and services (Jia et al., 2008), network traffic congestion and slow response rate to users.

Therefore, a trend of GIS named web GIS is carried out to enable users in accessing GIS conveniently via Internet at anywhere. Jianyu et al. developed a high performance web GIS based on Microsoft COM / Distributed Component Object Model (DCOM) to provide map symbol dynamic editing and network publishing through Internet (Jianyu et al., 2004). Their web GIS system consists of three major aspects which are the three tier system architecture of the web GIS to improve system extensibility and scalability, the design and deployment of distributed GIS component to enable sharing of data and computing resources, GIS data compression and transferring together with its relative key techniques. Those aspects aim to reduce network flux, accomplish faster response to users and solve the weaknesses of traditional web GIS that are implemented by using the Common Gateway Interaction (CGI) such as less component interaction, less server-side efficiency and less client-side extensibility (Shifeng & Steve, 2003). However, this system has its drawback which it cannot be migrated to UNIX operating system as it is implemented under Windows operating system by using the Microsoft COM technique.

Furthermore, GIS web services act as software components to supply hosted GIS spatial data and features, enabling integration to the practical GIS application without maintaining its geo-data. Thus, a lot of GIS web services works have been explored (Xiaolin, 2005; Shengru et al., 2004; Shifeng & Steve, 2003). As a consequence, a service-oriented distributed web GIS platform that combines web service, servlet or the Java Server Pages (JSP) function, the GIS Application Programming Interfaces (APIs) based on the J2EE infrastructure framework is constructed (Xiaolin, 2005). The purpose of this web GIS platform is designed to provide dynamic GIS service components for publication of both raster and vector maps to internet, viewing through web browsers. Yet, the work is only focused on the system architecture for GIS platform. Therefore, a broad GIS web services need to be further concentrated in order to fulfill the requirements of the practical GIS applications in different professional domains.

Due to the changes of GIS software from functional processing module to component processing GIS or web GIS, JIA et al. stated a suitable GIS data model for process modeling and GIS application system that is able to support dynamic system architecture of process-oriented modeling analysis are essential for realizing flexible spatial analysis function (Jia et al., 2008). Thus, they proposed a resolution of rapid application development for distributed GIS software based on distributional GIS process modeling and integration with software component technology from the perspective of engineering technology.

2.3.1 GIS Data

On the whole, a GIS application comprises of data and image. Nonetheless, the traditional GIS facing problems in data part of GIS application, for instances, interoperability among heterogeneous GIS duplicating enormous spatial data and synchronization of spatial data in homogeneous distributed geographic environment. Hence, Yu and Guoqing carried out some initial researches on integrating computing resources with the Spatial Information Grid (SIG) platform to overcome interoperability of heterogeneous GIS platform (Yu & Guoqing, 2008). Basically, their approach consists of three steps which are service development, service gridify and service publishing. The results shown that the integration of grid and GIS based on SIG is effective in processing massive heterogeneous spatial data meanwhile improving the interoperability of spatial data and facilitating access across resources.

Besides, Jian et al. researched on synchronizing voluminous geographical data accurately and reliably in the homogenizing environment (Jian et al., 2008). Their proposed solution consists of three main methods. Initially, they constructed a message server queue for stable message delivery by adopting both push and pull model. Then, they developed the extended Geography Markup Language (GML) for wrapping geographical spatial data into small flexible and linkable unit, defining them in accordance to granularity level, relation and length of inner string. By making use of their combination methods into a solution, unnecessary data manipulation and unmanageable network corruption which happen in geographic data synchronization can be solved.

2.3.2 GIS Image

On the other hand, many research works have been carried out in the field of image rendering. Although the application of distributed computing such as parallel rendering and distributed rendering to variety image rendering techniques likes polygon rendering, volume rendering, ray-tracing, radiosity renderers and terrain rendering is no longer a new topic (Thomas, 1996), it might still be beneficial to GIS image processing part in improving its performance.

Therefore, Thu and John presented a distributed rendering system that is making use of polygon rendering and multiple hardware graphics accelerators as well as optimizing rendering performances over a sequence of frames (Thu & John, 1999). Also, Rangel-Kuoppa et al. proposed a 3D rendering system that distributes rendering tasks across a multi-agent platform by creating a virtual 3D environment for rendering remote graphical units of the 3D object (Rangel-Kuoppa et al., 2003). By making use of that, more complex 3D environments are allowed in distributed environments compared to centralizing 3D processing meanwhile overcoming the bottleneck. Next, Gooding et al. implemented a render farm in distributed rendering environment by utilizing cluster of computers mainly for rendering 3D images together with a web based GUI that eases users in submitting jobs (Gooding et al., 2006).

Additionally, Thomas summarized the details of parallel rendering encompassing issues in designing parallel rendering algorithms from the perspectives of task granularity, data decomposition, scalability and load balancing (Thomas, 1996). Discussion on variety types of parallelism with its usage in rendering applications has been carried out as well. More to the point, sorting algorithms have been explored widely for processing computer graphics. Molnar et al. stated a classification scheme based on where the sort happens from object coordinates to screen coordinates that believes to be the fundamental whenever geometry processing and rasterization are performed in parallel (Molnar et al., 1994). They classified parallel rendering approaches into sort-first, sort-middle and sort-last algorithm while analyzing their advantages versus disadvantages.

Besides, Greg et al. presented a sort-first algorithm that works with immediate-mode rendering while minimizing the network traffic for scalable display (Greg et al., 2000). Meanwhile, Mueller suggested a parallel graphics architecture using sort-first algorithm which takes advantages of frame-to-frame coherence of scenes in rendering high resolution images of complex model datasets (Mueller, 2000). Then, the dynamic pixel bucket partition that utilizes recursive sort-first partitioning algorithm in parallel rendering is projected (Huabing et al., 2003). It distributes rendering workload evenly among rendering units and minimizes bandwidths required, accomplishing rapid and high quality rendering of enormous data.

Lastly, Rudrajit et al. investigated a hybrid parallel polygon rendering that adapts both sort-first and sort-last approach (Rudrajit et al., 2000). The main difference of this algorithm compared to others is that it dynamically partitioning both 3D model and 2D image while balancing the rendering load among resources. However, the algorithm has a potential drawback in client-side processing as well as dynamic data management.

2.3.3 GIS Scheduling

Besides research work done on data or image processing part of GIS applications, scheduling of GIS workload also plays a crucial role in improving the overall performance of GIS applications. This is the reason why this thesis focuses on the scheduling part of GIS applications in distributed grid environment. Contemporary, there are few prominent and commonly used grid schedulers that could be utilized for scheduling GIS workload such as the Sun N1 Grid Engine, the Portable Batch System Professional Edition (PBS Pro) and the Load Sharing Facility (LSF). Even though these grid schedulers are based on a central job scheduler running on a computational node (Gaj et al., 2003), they are adapting different types of scheduling techniques.

The Sun N1 Grid Engine is a resource management and scheduling system for UNIX in grid environment developed by the Oracle Corporation (Oracle, 2010). It is superior in dynamically administering and allocating multiple shared resources. Essentially, the N1 Grid Engine scheduler comprises a waiting line with pending jobs as well as a technical scheduler that responsible to assign jobs to idle resources (Stosser et al., 2008). Firstly, a user will need to submit a job and declare the technical requirements of the job. Then, the scheduler will place the jobs in the waiting line of pending jobs after receiving the job requests. However, the position of a job in the waiting line is based on job's priority that set manually by grid engine system manager according to individual service policies (Stosser et al., 2008; Oracle, 2010).

Meanwhile, the PBS Pro is an enterprise-quality workload management solution and job scheduling system supported by Veridian Systems (Gaj et al., 2003; Bill et al., 2004). In spite of the fundamental workload management capabilities, PBS Pro includes the compute aspect of Grids such as advance reservation, cycle harvesting and peer scheduling (Bill et al., 2004). Additionally, PBS Pro permits the consideration of job dependencies during job submission process. For instance, user may specify to run job Y after job X completed. A job will be run once it has been submitted but it will be interrupted if it is preempted by the scheduler or cancelled by the user. Furthermore, user and administrator can revoke any allocation in PBS Pro regardless the job is queuing or running. Therefore, a preempted job could be suspended or even required to start over queuing process again.

Then, LSF is a commercial job management system from Platform Computing Corporation (IBM, 2012). It has been declared as one of the most widely used local resource management system (Costen et al., 1999; Gaj et al., 2003) besides supporting the largest number of architectures (Do-Hyeon & Kyung-Woo, 2006) nowadays. Basically, Load Information Manager (LIM) and Remote Execution Server (RES) are those two daemons run in Load Sharing Library (LSLIB) that form the clustering layer in every host. LIM in every host responsible to monitor its own load then send information to LIM in master node whereas RES allows job to be executed in the machine as well as provides remote execution services. If there is the case where LIM in master node is down then one of the other LIMs will take over the coordinating role until the real master node recovers. Hence, LSF provides a strong fault tolerance mechanism (Do-Hyeon & Kyung-Woo, 2006).

On top of LSLIB layer, there are set of utilities such as LSF Batch that allows jobs to be batched and scheduled from the information attained from LIMs, LSF Job-Scheduler extends LSF Batch capabilities by permitting event driven jobs to be scheduled and LSF Multi-Cluster (Xu, 2001) allows load sharing among clusters. Furthermore, LSF offers variety scheduling policies such as fair share scheduling, backfill scheduling and etc. (IBM, 2012) to ensure that resources are allocated in conjunction to individual service level agreements (SLAs). After the experiments carried out by Costen et al., they concluded LSF is good in avoiding the machines with high processor utilization but not avoiding paging (Costen et al., 1999). Lastly, they observed that LSF only does well in balancing load for serial jobs and not for parallel jobs in a cluster with busy condition.

After reviewing a few existing grid schedulers together with its pros and cons, a job scheduler plays a vital role in performing job scheduling and queuing based on the job characteristics, resource requirements as well as scheduling policies. Due to the importance of job scheduler, the study in this field can be further discovered in order to propose a better scheduling technique for scheduling GIS jobs. The proposed scheduling algorithm is developed by exploiting the concept of queuing theory that happens in our daily life. It prioritizes fairness of each job categories upon arrivals which is varies from other existing job schedulers. Therefore, user satisfaction could be achieved besides improving the overall performance of GIS applications.