Enhanced Partition Aware Engine For Efficient Load Balancing Computing Using Fluid Queue Model

A. Saranya and R. Lakshmi

1PG student - Department of CSE, K.L.N.College of Engineering, Pottapalayam, Sivagangai 630 612 INDIA.
2Associate professor- Department of CSE, K.L.N.College of Engineering, Pottapalayam, Sivagangai 630 612 INDIA.

ABSTRACT

The performance of parallel graph computation systems are greatly affected by the quality of graph partition. The two main factors which are used to measure graph partition quality are balance factor and edge cut ratio. However, according to the practical study on Giraph, these systems only optimize for remote messages and cannot efficiently handle the growing workload of local message processing even though a high quality graph partition is used. PAGE process both local and remote message process in a unified way and use DCCM, to provide sufficient message process units to concurrently handle current workload but it leads to imbalance workload. This paper present a technique called fluid queue model, it is used along with graph computation engine called partition aware graph computation engine (PAGE). To improve efficient Load balancing and to support computation tasks with different partitioning qualities. Fluid queues have been widely used in the buffer occupancy distribution and the performance evaluation of high-speed communication networks.

KEYWORDS: Fluid queues model, PAGE, buffer occupancy, load balancing, giraph.

INTRODUCTION

Graphs are an important part of everyday life. Enormous big graphs are common nowadays. Well-known examples include web graphs, social networks etc. Now a day’s graphs have trillions of edges and nodes. It also used to represent relationship between various objects [8]. It provides lots of important applications, such as linkage analysis, pattern matching and community detection. With this unexpected development of a large graphs and parallel processing, diverse applications becomes the important for graph computing paradigm for large scale graph analysis. Various parallel graph computation systems have been introduced, such as Pregel, GPS, Giraph and Graph-Lab.

Most of the graph processing software concentrates more on remote message processing and leave out local messages. When the local communication cost increases unexpectedly, which straight forwardly leads to the downgrade of overall performance. This unusual result implies the local message processing becomes a bottleneck in the system and takes over the overall cost when the workload of local message processing increases suddenly. PAGE use dual message processor to handle both the messages equally and DCCM to concurrently handle current work load.

In a fluid queue model a separable commodity arrives at a storage capacity where it is stored in a buffer and slowly released. In a standard queuing system it is considered that, individual jobs or customers arriving at service facility, wait, then get service and depart. For such models calculate the number of customers or jobs in
the system and describe the experience of individual customers. In disparity a fluid queue model is used in request where individual customer is so small that can hardly be notable. It is then easier to have a continuous stream of work that stream into the system instead of jobs or customers.

They are used to represent the systems where some quantity gathers or is exhausted, gradually over time, subject to some unsystematic environment. For fluid queues models, buffer content can be study at any time t, which has finite or infinite capacity of work in the system and it cannot be negative.

2 Related Works:

Bean et al [1] proposed fluid flow model for maintenance of power generation systems. This model can be used as a decision making tool in the development of long-term profitability, operating strategies and short term changes. It is used a tool to inform strategic decision making on electricity market bidding prices.

Huang et al [2] proposed two-level stochastic fluid tandem queuing model for burst impact analysis. To model the burstiness of a packet stream at different time scales a novel method called two-level Markov On-Off source model is used. To reveal the impact of traffic burstiness at two levels on the queue lengths in a tandem queue system. The burst of both levels have linear impact on the average queue size throughout the entire tandem queue network. Fluid queueing model, which is dedicated to deal with the correlation structure in network traffic and classical queueing theory, which focuses on the packet level burstiness.

Xiangfeng et al [3] proposed fluid approximation of point-queue model. To prove point-queue model is actually a queuing model with a server and a buffer with infinite capacity, a fluid approximation (FA) model is proposed to interpret the original point-queue model. The arrival to the server is time-dependent with constant time delay, which is the free flow time of the link, while the service rate is constant whose value is the link capacity. With the different ensemble server utilization, the outflow is time-dependent, too. With the first-order Taylor expansion, the Gronwall’s inequality is used to prove the non-negativity of the queue length in the FA model based on some mild assumptions.

Vijayashree et al [4] proposed fluid queue modulated by a single server queuing model subject to catastrophes under steady state conditions. A closed form expression for the buffer occupancy distribution is obtained using continued fraction methodology in the transformed domain. Fluid models are appropriate in the field of telecommunication for modelling the network traffic where individual units of arrival have less impact on the performance of the network. Such models characterize the traffic as a continuous stream with a parameterized flow rate. It provides in-depth analysis of the physical model for the practitioners.

Dube and Altman [5] proposed goodput analysis of a fluid queue with selective discarding and a responsive bursty source. Analyze feedback system consisting of a finite buffer fluid queue and a responsive source. The source alternates between silence periods and active periods. At random epochs of times the source becomes ready to send a burst of fluid. The length of the bursts is independent and identically distributed with some general distribution. The queue employs a threshold discarding policy in the sense that only those bursts at whose commencement epoch the workload is less than some preset threshold are accepted. If the burst is rejected then the source backs off from sending. Using techniques from Volterra Integral Equations obtain an explicit characterization of the queue length distribution at commencement epochs of bursts from which we obtain an explicit characterization of the goodput ratio associated with such a feedback system.

3 Existing System:

Giraph, only optimize remote messages and cannot powerfully handle the increasing workload of local message processing even though a high quality graph partition is used. In PAGE’s worker, communication module is added with a new dual concurrent message processor. This message processor concurrently handles both remote and local messages in a same way, thus accelerating the message processing, then partition aware module is added to each worker to observe the partition related characters and adjust the concurrency of the message processor based on online workload, which leads to imbalance workload on server process.

4 Proposed System:

In communication networks, data generated by a source node are carrying to their destination by routing them via a switch, or sequence of intermediate nodes [6]. The information arriving at the intermediate node is buffered for transmission or service, the server generally being a processing unit or communication channel. In high-speed networks, the traffic is very bursty in nature.

It is used to study the buffer occupancy allocation in high-speed networks. These are natural choice for problems concerning continuous flow. For some queuing systems behavior of individuals and the flow consists of discrete entities is not important to identify the performance analysis, fluid queue models are use as approximate models.

These models estimated packet streams by flows of fluid. Fluid queues are of special inference of congestion controls mechanisms [7]; particularly the input process is change by the current value of the buffer content. Previous work on PAGE predominantly focused on queues in which a single server process for both the
messages [8]. In these models, based on whether the buffer occupancy is below or above, the traffic source is allowed to transmit various server processes to improve workload balance.

To efficiently balance the workload among message processes fluid queue model is used along with partition aware graph computation.

4.1 Partition Aware Graph Computation Engine:

PAGE- Partition Aware Graph computation Engine, is intended to maintain high performance and support different graph partition qualities by new cooperation methods and tuning mechanism.

Architecture:
(a) PAGE follows the master-worker pattern. Graph is divide and distributive among workers and stored in memory. The master is responsible for aggregating and coordinating.
(b) Equipped with an improved communication module and partition aware module.
Minimum edge cut ratio are usually in order to have minimum cost.
The following fig shows that messages are usually send with minimum edge cut ratio in order to reduce cost

![Fig. 1: Messages with minimum edge cut ratio](image)

Graph Algorithm Execution in PAGE:
The following algorithm are used in page technique.

Algorithm: Procedure of a Superstep in PAGE
1: DCCM reconfigures dual concurrent message processor parameter.
2: foreach active vertex v in partition P do
3: call vertex program of v;
4: send messages to the neighborhood of v;
5: /* monitor tracks related statistics in the background */
6: end foreach
7: synchronization barrier
8: monitor updates key metrics, and feeds to the DCCM

4.2 Dual Concurrent Message Processor:
It decouples the computation cost from the local communication cost, as the computation will not be stop by any communication operation. Hence, it compute both remote and local message in a unified way.
It has two types of concurrency
1. Concurrency in message processor
2. Concurrency at internal of remote and local message processor.

4.3 Dynamic Concurrency Control Model:
Partition aware module has two main components: DCCM and monitor. The monitor is used to maintain necessary information and provide this to DCCM. The DCCM create appropriate parameter through estimation and adjust the concurrency of dual message processor.
4.4 Fluid Queue Model:
Fluid flow model for the network of servers is used to design a suitable load-balancing that assures equilibrium of the queues. This mechanism will redirect the incoming message to most appropriate server and balance the overall system request load. This mechanism influence local balancing in order to achieve global balancing. This is carried out through a periodic interaction among the system nodes. It improve average number of requests served in a unit time and have better response time.

It assures equilibrium of the queues in a balanced request by using a fluid flow model for the network of servers. It redirects incoming messages to the most appropriate server, thus balancing the overall system requests load.

5. Comparative Study:
The response time is increased when PAGE is combined with fluid queue model and increase average number of request. Fluid queue model use many server process to handle the burst messages.

It also reduce the time taken to process the message, which leads to the reduction cost and time.

Fig. 3: Response time for PAGE

Fig. 4: Response time for fluid queue model
Conclusion:
Using PAGE with DCCM it leads to the burst of messages, to overcome this problem a new technique called fluid flow model is used along with PAGE. It aims at achieving load balancing in the network by removing local queue instability conditions through redistribution of potential excess traffic to the set of neighbors of the congested server. Finally it provides complete solution for load balancing in a cooperative, distributed environment.

REFERENCES