Inter Module Communication in Distributed Data Base and Distributed Computing System

Lovely Arya

Research Scholar

Dravidian University

KUPPAM, ANDHRA PRADESH

Abstract

In DDBS systems Data & operation allocation are both closely interrelated & highly dependent on each other DCS & DDBS are will be compared in the research with reference to task allocation Models, Algorithms, Issues and Tools. General models and objective function are basic platform for research in the area of task allocation. Major issues in DCS will be explored in this research, while in DDBS the main issues will be highlighted. The characteristics of DDBS like distributed data, distributed operations from Inter Module Communication, query tree and result file are mentioned as tools will be taken in this field of research

Introduction

Distributed computing systems such as a network of heterogeneous workstations or PCS become an attractive alternative to expensive, massively parallel machines . But to exploit effective parallelism or distributed system , the tasks (more processors have more tasks) must be properly allocated to the processors. Multiple tasks if not managed properly would lead to the degradation of overall system DCS and DDBS differ in the resources to be shared. DCS share hard disks and printer etc. while DDBS share distributed databases ,where the data as well as operations on these data items are equally important. Importance of distributed databases and more specifically task (operation) based distributed database designs in design development is gaining importance. The DDBS defines and makes it possible to identify the tasks handled as well as distributed nodes for executing these tasks . In distributed computers system the computers may be homogenous or heterogeneous, connected together for some common application execution .

Task Allocation Of Distributed System

A task is a set of modules and module (in task)executes on one of the processing nodes (therefore executes on every one of the processors at that node) and communicates with some other modules of the task by inter module communication(IMC).A task is a program or a part of a program in execution. It is one of the important computational activities that takes place at the same time and /or at different locations In contrast to a task , a job is a whole, usually sequential ,program in execution. Task allocation is an

essential phase in distributed system & distributed software design. In DCS the software application is called a task and is a set of cooperating modules . For achieving a fast response time from such systems , an efficient assignment of the application tasks to the processors is imperative. The general assignment problem is known to be NP – hard ,except in a few special cases with strict assumptions. A task is represented by a task graph. As the position of modules in task graph represents its precedence of execution ,it is called a task precedence graph(TPG).The communication among its modules can also be shown by a graph known as task interaction graph(TIG).These two graphs can be represented separately or can be combined into one.

Pros And Cons Of Distributed Processing

Resource sharing: Since a computer can request a service from another computer by sending an appropriate request to it over the communication network, hardware and software resources can be shared among computers. For example, a printer, a compiler, a text processor, or a database at a computer can be shared with remote computers.

Enhanced Performance: A distributed computing system is capable of providing rapid response time and higher system throughput. This ability is mainly due to the fact that many tasks can be concurrently executed at different computers. Moreover, distributed systems can employ a load distributing technique to improve response time. In load distributing tasks at heavily loaded computers are transferred to lightly loaded computers, thereby reducing the time tasks wait before receiving service.

Improved reliability and availability: A distributed computing system provides improved reliability and availability because a few components of the system can fail without affecting the availability of the rest of the system. Also, through the replication of data (e.g., files and directories) and services, distributed systems can be made fault tolerant. Services are processes that provide functionality (e.g., a file service provides file system management; a mail service provides an electronic mail facility).

Modular expandability: Distributed computing systems are inherently; amenable to modular expansion because new hardware and software resources can be easily added without replacing the existing resources.

Operation Of Ddb(Distributed Data Base)

DDB system is group of distributed computers on which a database is portioned and saved so that for processing user query all look as single unit. A DDBS system consists of a set of data objects each of which is assigned a value. A data object represents the smallest unit of the database accessible to transactions. All the user requests for access to database are handled by the database management system. The basic units of user activity in database systems are transactions .Each transaction represents the basic mechanism for identifying and defining a set of logically associated operations. Each transaction has time factor associated with it. Data (object) and operation (task) allocation are among the key issues during designing a distributed database. Data allocation defines what data is stored and at what nodes(with replication) while operation allocation states where

accessing and processing of operations(select ,join, project etc.)will take place. Each node has its own (NDS) network directory structure which is small and stable relatively giving details(index) of data stored at that node. Both data and operation allocation are interdependent problems and must be solved simultaneously. The optimal set of all the data copies and their optimal allocation depends on processing schedules of all retrieval and updates accessing these data(ie task allocation).However the processing schedules depend on where data sets & their copies are located (ie data allocation)hence it is circular problem and to be effective a DDB design approach must comprehensively treat both data & operation allocation as a unified whole.

Challenges Of Distributed Data

A distributed data base environment complicates the systems organization and the resulting data base operations. A decision must first be made about the allocation of the files to the various locations, or nodes, of the data base network. A particular file could be kept in some unique, central place; alternatively allocating the various file portions to several different nodes could partition it. Finally, the file or certain file portion could be replicated by number of messages and request circulating from node to node assuming that the file portions of interest at a particular site are locally stored. When the data files are replicated, the message traffic between nodes may be substantially reduced. In fact, a tradeoff exists between the extra storage used by the data replication and the increased speed of operations resulting from the reduced communications load between the nodes. In a distributed database system the need for the basic physical and logical data independence is expanded to include also location and replica transparency. Location transparency implies that the user programs are independent of the particular location of the files, while replica transparency extends the transparency to the use of an arbitrary number of copies. Once a particular file environment is created, procedures must be available for executing the various transactions and furnishing results to the requesting parties. A given transaction might be run locally; alternatively, various remote points might be asked to carry out the operations followed by the routing of responses to the originating points.

Distributed Systems and Parallel Computing

Parallel computing systems aim for improved performance by employing multiple processors to execute a single application. They come in two flavors: shared-memory systems and distributed memory systems. The former use multiple processors that share a single bus and memory subsystem. The latter are distributed systems in the sense of the systems that we are discussing here and use independent computing nodes connected via a network (i.e., a multi computer). Despite the promise of improved performance, parallel programming remains difficult and if care is not taken performance may end up decreasing rather than increasing.

Distributed systems in context The study of distributed systems is closely related to two other fields: Networking and Operating Systems. The relationship to networking should

be pretty obvious, distributed systems rely on networks to connect the individual computers together. There is a fine and fuzzy line between when one talks about developing networks and developing distributed systems. As we will discuss later the development (and study) of distributed systems concerns itself with the issues that arise when systems are built out of interconnected networked components, rather than the details of communication and networking protocols. The relationship to operating systems may be less clear.

Conclusion

In this paper, we have discussed several issues and challenges involved in providing about Task allocation is an essential phase in distributed system & distributed software design. In DCS the software application is called a task and is a set of cooperating modules with the help of Inter Module Communication to achieve a fast response time from such systems. As efficient assignment of the application tasks to the processors is imperative. The general assignment problem is known to be Non Deterministic Polynomial – hard, except in a few special cases with strict assumptions

Reference

- T. Urhan, M. J. Franklin, and L. Amsaleg, "Cost based query scrambling for initial delays," ACM-SIGMOD International Conference on Management of Data, 1998.
- R. von Behren, J. Condit, F. Zhou, G. C. Necula, and E. Brewer, "Capriccio: Scalable threads for internet services," in Proceedings of the Ninteenth Symposium on Operating System Principles (SOSP-19), Lake George, New York, October 2003.
- M. Welsh, D. Culler, and E. Brewer, "Seda: An architecture for well- conditioned, scalable internet services," in Proceedings of the 18th Symposium on Operating Systems Principles (SOSP-18), Banff, Canada, October 2001. C. Zou and B. Salzberg, "On-line eorganization of sparsely-populated b+trees," pp. 115–124, 1996.
- Procedures Manual for the Consultative Committee for Space Data Systems. CCSDS A00.0-Y-7.4. Yellow Book. Issue 7.4 (interim update). Washington, D.C.: CCSDS, January 2001.