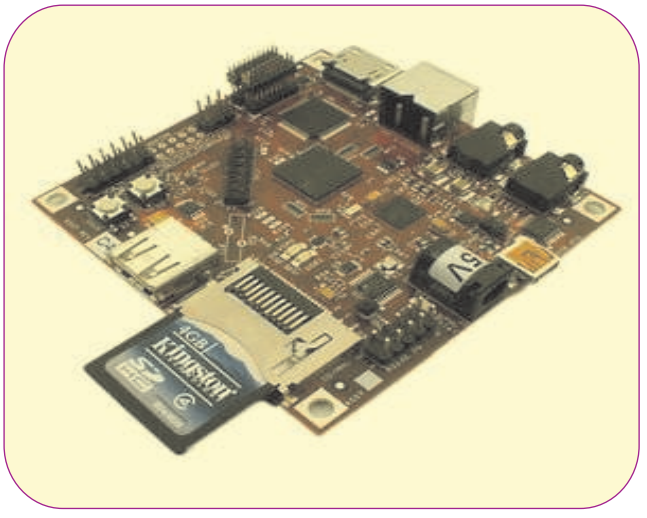
REVIEWS OF LITERATURE
ISSN: 2347-2723
IMPACT FACTOR : 3.3754(UIF)
VOLUME - 5 | ISSUE - 4 | NOVEMBER - 2017

PERFORMANCE EVALUATION OF MULTIPROCESSOR SERVER ARCHITECTURES FOR INFORMATION PROCESSING APPLICATION



Neelam Rajakishor
M.C.A & M. tech , Dept of Computer Science & Informatics, Kakatiya University Warangal.

ABSTRACT :

The internet begins to grow quickly in the recent years as a results user dependency as well as expectation enhanced. The increase bandwidth and sophisticated network infrastructure has also increased dramatically to stable various internet application services. One of the important applications is accessing information through internet by the end users. In hardware approach, one of the important issues is how to deal with the design of high performance server to attain the desirable performance. Traditional approaches toward the design of server architecture have targeted static contents that are usually network intensive. The present work is concerned with the design and development of a new multiprocessor architecture and to use it in a variety of information processing applications. The performance of the proposed architecture is tested for both; handling the unpredictable communication traffic and servicing a number of information retrieval queries in a variety of mode. The proposed scheme is also implemented on other similar multiprocessor network namely Hypercube (HC) and Star Crossed Cube (SCQ). The comparative simulation study shows that the proposed scheme provides more advantageous performance in term of task Scheduling on LCQ network. The other dynamic scheduling schemes such as Minimum Distance Scheduling (MDS) and Two Round Scheduling (TRS) are also implemented and tested on LCQ server. The Proposed Model that is LCQ network with the proposed scheduling scheme shows the superiority of the topology and considered as a better organizational model.

KEYWORDS : *Multiprocessor, LCQ, SCQ, MDS.*

INTRODUCTION

Computer system design has historically been more of an art than a science. The design of any system that is as complex as a present day computer system can not become a science until the effects of all variables upon the system become well understood. Take for instance the case of the aircraft industry. In its infancy it was merely a matter of luck who stayed up in the air and who came crashing down to the ground. This is in a sense true of all complex systems including computers. To understand the impact of design variables on the system, to make computer design a science, has been the major motivating factor in the performance evaluation of computer systems. Performance evaluation can help in system design in a number of ways: by measurement and analysis of existing systems, by projecting the impact of additions and expansions on system performance, and by predicting performance of new designs. The present work is concerned with the performance evaluation of a specific design for a specific set of applications: real time applications of multiprocessor computer systems. The real time environment imposes some challenging requirements on computer systems. These include high reliability, availability and

promptness of response. Multiprocessors can apparently satisfy all of these requirements successfully. These and other qualities, such as graceful degradation and ease of expansion, are reasons for turning to multiprocessor designs for real time applications.

The increasing reliance on the internet as a ubiquitous medium for accessing information has compelled heavy load on the network resources. As a result, there are number of challenges in the quality of Web services. One of the desired qualities is fast accessing of information and hence small downloads time. To improve and meet the requirement of these services, efficient servers are designed. Exploiting parallelism is a necessity in the design of high performance computer system. In terms of hardware, this typically means providing multiple simultaneously active processors (nodes). In terms of software, it means structuring a program as a set of largely independent subtasks (load). Research is active in the direction of developing new multiprocessor architectures and scheduling the partitioned program onto it in order to achieve higher performance.

Single processor systems have achieved great speeds and have been used in high performance computing. However, this trend came to an end because there are physical and architectural bounds, particularly the physical limit of chip manufacturing. A parallel system in the form of internally linked processors is an alternative approach to increase computing power. Recently, advances in computer networks have created a new type of parallelism in the form of networked autonomous computers. Instead of putting everything in a single box i.e. tightly couple processors to memory; the Internet achieved a kind of parallelism by loosely connecting everything outside of the box. To obtain the best possible use of a high performance computer system with internal or external parallelism, it is necessary to understand the interaction between hardware and software parts of the system. Today numerous applications require more computing power than a sequential computer can offer. Parallel processing provides a cost effective solution to this problem by increasing the number of CPUs also known as Processing Elements (PEs) in a computer and by adding an efficient communication system between them. The work-load can now be shared between these connected processors. These interconnected processors results in much higher computing power and performance that could not be achieved with traditional single processor system. There are many applications of parallel processing. The most notable are VLSI design, engineering analysis (CAD/CAM), military applications and global climate modelling and forecasting etc. The parallel processing environment has the potential for providing significant computing power at a fraction of the cost of fourth-generation supercomputers. The technology of parallel processing is mature and can be exploited commercially. There is already significant research carried out on the development of tools and environment related to parallel processing.

DESIGN ISSUES OF SERVER

The design of information retrieval systems must address not only issues of look-and-feel but also of effective interaction. Dialog models based on relevance feedback and query reformulation explicitly address the ill-defined nature of information seeking by allowing users to learn from the repository and iteratively refine the information need. Systems need to support a number of interaction styles such as querying and browsing to accommodate the different kinds of search strategies users may need to use. Design strategies for retrieval systems need to pay particular attention how fast user's access information from the hardware. Therefore, effectiveness of information retrieval algorithm is also dependent on the physical network topology. Moreover, if they are bottlenecks in different network connection the performance is not increased and these additional connections simply waste the server resources. To overcome these bottlenecks enhancing the server performance has become the critical issue during the design and development of such systems. Besides, achieving additional computing power by incorporating more than one processing element, the performance of the server can be enhanced by running parallel algorithms for accessing information. Information analysis, user modelling and interaction modelling are some other methods that can be incorporated to improve the effectiveness of algorithm. A combination of parallel hardware with efficient software approach optimizes the system performance.

PERFORMANCE ISSUES OF SERVER

Improving the performance of servers has become a critical issue to cope with the increasing use of network based services. Due to increase in the demand of information transactions and scheduling overhead the existing approaches have some limitations. The critical nature of many online transactions require a high performance server since such servers are anticipated to be the bottleneck in hosting Internet based services. Multiprocessor approach is the most powerful and economical way to achieve desirable performance by incorporating efficient network topologies. Numerous interconnection topologies have been designed to achieve the desired performance. Nevertheless, the actual performance is far below the expectation of users when executing parallel applications on a particular multiprocessor network. The performance of multiprocessor interconnection topologies can be evaluated in terms of numerous topological parameters along with the routing techniques involved. These parameters along with suitable routing can increase the performance of multiprocessor architectures. The load balancing is also an important requirement which has a scope to improve the overall performance. Load balancing parameters namely Load Imbalance Factor (LIF's) as well as execution time are critical parameters for achieving high performance. The aim is to design and structured approach for running massively parallel application with multiple processing elements at backend. Therefore, efficient scheduling algorithms are designed that partitioned the task and schedule them onto the physical processors. Similarly, to manage the traffic a dynamically optimal solution is essential by means of programming model. In the present work, an idea is presented to overcome some of the problems of delay in accessing and downloading the information using multiprocessing technique. A new scalable network topology called Linear Crossed Cube (LCQ) multiprocessor is proposed to be used as server. Through simulation the performance of the proposed server is evaluated for both the load balancing and exchanging information simultaneously.

ORIGINAL CONTRIBUTION

The complete work as presented in this thesis can be divided into three parts. The first part is concerned with the design and development of a low cost multiprocessor architecture. A new multiprocessor topology called Linear Crossed Cube (LCQ) network is proposed. It includes the advantages of hypercube topology such as symmetry, smaller diameter, good connectivity and high bisection width. It is a highly scalable architecture that consists of constant node degree independent to the network size. The thesis also proposed a scheduling scheme that can maintain the efficient utilization of all the processing elements available in the network for various types of load. To minimize the balancing time, a novel dynamic scheduling scheme named as Optimal Multi-Step Scheduling (OMSS) scheme has been proposed and implemented on LCQ network. The OMSS scheme schedules the tasks on LCQ network whereas the load is divided into approximately equal number of packets to achieve load balancing in the system. The OMSS scheme is also implemented on other standard reported multiprocessors architectures and a comparative study is carried out which shows a minimum balancing time on the LCQ network.

CUBE BASED NETWORK

The cube based architectures are widely used networks in parallel systems. They have good topological properties such as symmetry, scalability and possess a rich interconnection topology. The Binary hypercube or n-cube has been one of the famous interconnection networks used in the design of parallel systems [Saad and Schultz, 1998]. Besides having a number of desirable features, the major drawback of hypercube is the difficulty of its VLSI layout. The minimum number of tracks for VLSI layout of an n-cube hypercube a one dimensional implementation has an order of network size which results more difficulties to design and fabricate the nodes of the hypercube. Several variations of hypercube architecture are designed to improve it further. Some examples of hypercube variants are the Folded Hypercube (FH), Dual Cube (DC), Folded Dual Cube (FDC), Meta Cube (MC), Folded Met cube (FMC), Crossed Cube (CC), Folded Cross Cube (FCC) and Star Crossed Cube (SCQ). Similarly, modified hypercube architecture named the necklace hypercube which has a good scalability and efficient VLSI layout has been reported.

LINEARLY EXTENSIBLE NETWORK

The Linearly Extensible Networks is another class of multiprocessor architectures which reduces some of the drawbacks of HC architectures. Several researches have been carried out in the design of linear type architectures which are considered less complex and easily extensible. Some examples are Linearly Extensible Tree (LET), Linear Array (LA), Ring, Star Graph and Torus Ring Network etc. The complexity of these networks is lesser as they do not have exponential expansion. Besides the scalability, other parameters to evaluate the performance of such networks are degree, number of nodes, diameter, bisection width and fault tolerance. Selection of a better interconnection network may have several applications with lesser complexities and improved power-efficiency. One such modern application is network on chip (NoC) paradigm where different cores are embedded with appropriate connectivity. Some examples may include mesh, torus, star, etc,. Besides their lesser complexity linear architectures severally suffer from the drawbacks of high connectivity and symmetry.

OBJECTIVES OF THE STUDY

1. To understand the awareness of multiprocessor server architectures.
2. To examine the linear cross cube and cube connected cycle and INS processing Net works.

METHODOLOGY

The study is basically depends upon secondary source. Basic data for the study is collected from the various books, journals, internet, and websites.

DESIGN ISSUES OF INTERCONNECTION NETWORKS (INS)

DIMENSION AND SIZE OF NETWORK It should be decided how many processing element (PE's) are there in the network and what the dimensionality of the network is i.e. with how many neighbours, each processor is connected.

SYMMETRY OF THE NETWORK It is important to consider whether the network is symmetric or not i.e., whether all processors are connected with same number of processing elements or the processing elements of corners or edges have different number of adjacent elements.

MESSAGE SIZE What is a message size? How much data a processor can send in one time unit?

DATA TRANSFER TIME How long does it take for a message to reach to another processor? Whether this time is a function of link distance between two processors or it depends upon the number of nodes coming in between.

STARTUP TIME what is the time required to initiate the communication process?

LINEAR CROSSED CUBE (LCQ) NETWORK

DESIGN AND ANALYSIS: The Proposed LCQ network is an undirected graph and grows linearly in a cube like pattern. The network itself is defined through connection functions in a manner similar to that of cube connection. Let r be the set of designated processor of R thus, r = {Pi}, $0 \leq i \leq N-1$ The Link functions L1 and L2 define the mapping from r to R as: L1 (Pi) = P (i+2) Mod N; Pi in r L2 (Pi) = P (i+3) Mod N ..... (2) The two functions L1 and L2 indicate the links between various processors in the network. Let Q be a set of N identical processors, represented as Q= {P0, P1, P2..... PN-1} The total number of processors in the network is given by N = ... (3) where, n is the depth of the network. For different depth, network having 1, 3, 5, 7..... and 2, 4, 6, 8, 10..... processors are possible. The link between different nodes in a network of 8 processors could be obtained as:

P0 = P(0+2) Mod N = P2 ; P0 = P(0+3) Mod N = P3
P1 = P(1+2) Mod N = P3 ; P1 = P(1+3) Mod N = P4
P2 = P(2+2) Mod N = P4 ; P2 = P(2+3) Mod N = P5
P3 = P(3+2) Mod N = P5 ; P3 = P(3+3) Mod N = P6
P4 = P(4+2) Mod N = P6 ; P4 = P(4+3) Mod N = P7
P5 = P(5+2) Mod N = P7 ; P5 = P(5+3) Mod N = P0

P6 = P(6+2) Mod N = P0 ; P6 = P(6+3) Mod N = P1
P7 = P(7+2) Mod N = P1 ; P7 = P(7+3) Mod N = P2

In order to define the link functions, we denote each processor in a set of K processors as Pin, n being the level/depth in LCQ where the processor Pi resides.

Number of Nodes (N) The number of nodes in a multiprocessor system performs a vital role to examine the performance of a multiprocessor network. The lesser number of nodes means complexity of system is lesser as well as the system is considered more economical. Therefore, number of nodes should be optimal. The LCQ network has number of nodes which is $N = \sum k^{n} =_{1} K$ whereas, the number of nodes in the hypercube and crossed cube networks is 2^{N}.

Diameter (D)

The measure of the maximum inter-node distance in the network is known as diameter of a network. This property is important to determine the distance involved in communication which ultimately enhances the performance of multiprocessor systems. In simple words the maximum shortest path between source and destination node is called diameter of a network. The diameter of a network is bound to increase as the size grows unless there is no limit on the number of links. In case of LCQ network, it is observed that the diameter does not always increase with the addition of a layer of processors.

CUBE CONNECTED CYCLE (CCC)

The CCC architecture is an attractive parallel computation network suitable for VLSI implementation while preserving all the desired features of hypercube. The CCC is constructed from the n- dimensional hypercube by replacing each node in hypercube with a ring containing n node. Each node in a ring then connects to a distinct node to one of the n dimensions. The advantage of the cube- connected cycles is that node's degree is always 3, independent of the value of n. This architecture is modified from hypercube i.e. a 3-cube is modified to form a 3-cube-connected cycles (CCC) restricted the node degree to 3. The idea is to replace the corner nodes (vertices) of the 3-cube with a ring of 3-nodes. In general one can construct k-cube-connected cycles from a k-cube with n=2k rings nodes.

CROSSED CUBE (CC)

The Crossed Cube (CC) has the same node and link complexity as the hypercube and has most of its desirable properties including regularity, recursive structure, partition ability, strong connectivity and ability to simulate other architectures. Its diameter is only half of the diameter of the hypercube. Mean distance between vertices is smaller and it can simulate a hypercube through dilation 2 embedding. The basic properties of the CC, optimal routing and broadcasting algorithms are developed. The CC is derived from a hypercube by Changing the way of connection of some hypercube links. The diameter of CC is almost half of that of its corresponding hypercube. The n-dimensional cross cube denoted by CCn, is the labelled graph defined inductively as follows. CC1 is K2, the complete graph on two vertices with labels 0 and1 as show in Figure 2.3. Specially, the diameter of an n-dimensional Cross Cube is $\lceil n+1/2 \rceil$ and the diameter of an n- dimensional hypercube is n. However, the CC makes no improvement in the hardware cost compared to the hypercube. The node connectivity of CC is n with 2n and it has a bisection width 2 n-1.

OPTIMAL MULTI-STEP SCHEDULING ALGORITHM (OMSS)

The basic approach in MDS is to optimize the load balancing among processors under the constraint of the need to keep message path lengths to one hop and thus satisfying the minimum distance property. Migration from donor processor is done to the directly connected acceptors. Thus, for every donor there is a set of Minimum Distance Acceptors (MDA). A new scheme has been proposed for solving load balancing problem with unpredictable load estimates. The proposed algorithm works as an extension of MDS and named as Optimal Multi-Step Scheduling Algorithm (OMSS). It is dynamic in the sense that no prior knowledge of the load is assumed. Decision of migration is taken on the fly based on the current system utilization. Adequate number of

nodes with multiple paths are available that may be considered as donor and acceptor nodes. Selection of nodes for task migration is challenging and directly affects the communication cost. In order to reduce the communication cost, the OMMS Algorithm selects directly connected nodes at first step for the purpose of task migration. To avoid greater communication overhead, the algorithm does not select nodes with multiple hopes. Once directly connected processors are exhausted, the OMMS Algorithm identifies highly imbalanced nodes irrespective of their connectivity. After identification it searches those paths between these nodes which require only a single intermediate node that could be involved for task migration.

PARALLEL QUERY PROCESSING

There are a variety of approaches wherein parallelism will help an IR server to procedure queries much faster. The two most desired strategies are index partitioning and replication. Assume that we have a complete n index server. Following the regular terminology we allude to these servers as nodes. By producing n-simulations of the index and allocating every model to a detached node, we are able to recognize an n-fold improve of the search engine's service rate without disturbing the time needed to procedure one query. This kind of parallelism is known as inter-query parallelism as numerous queries could be processed in parallel. Nevertheless every single query is prepared sequentially. On the other hand, we can separate the index into n-components and every node performs merely by itself on compact section of the index.

CONCLUSION

A variety of information retrieval queries are executed and the execution time is evaluated. Simulation results are evaluated and compared with different types of systems. In particular, queries are executed on Multi-Process uniprocessor system as well as on multiprocessor systems with different number of nodes. The comparative study shows that the LCQ server with eight processor outperforms with the proposed information retrieval algorithm. Comparisons of the LCQ multiprocessor network and its inherent qualities along with lesser number of processors reveal that this network is reasonably comparable and economical with the existing multiprocessor networks. From the simulation studies, it has been found that the proposed dynamic scheduling scheme is performing better on LCQ network in comparison to other dynamic scheduling schemes. Therefore, it may be concluded that for unpredictable and bursty data the proposed architecture with the proposed schemes results in better performance. Hence, due to its various characteristics and performance parameters, the LCQ architecture could be used as server in the networking. When LCQ server is used to compare the retrieval of the information with a uniprocessor server, it is found that the LCQ type server reduces the resource download time when proposed information retrieval algorithm is applied.

REFERENCES:

1. Mallach, E.G., "Analysis of a Multiprocessor Guidance Computer", Ph.D. Thesis, Dept. of Aero. and Astro., M.I.T., June 1969, Also available as C.S. Draper Lab. Report T-515.
2. Gordon, M. I., Thies, W., and Amarasinghe, S. (2006, October). Exploiting coarse-grained task, data, and pipeline parallelism in stream programs. In ACM SIGOPS Operating Systems Review (Vol. 40, No. 5, PP. 151-162).
3. Pancake, C. M. (1996). Is parallelism for you? Computational Science & Engineering, IEEE, 3(2), 18-37.
4. Balram, N., Belo, C., and Moura, J.M.F. (1988). In Proceedings of ACM/IEEE International Conference on Supercomputing, PP. 247-257.
5. Tiwari, R., Sharma, M., and Mehta, K. K. (2015). International Journal of Advanced Research in Computer Science and Software Engineering, Vol. 5, No. 6, PP. 730-743, 2015.
6. Mishra, D., & Mishra, A. (2008). Design issues in client-server software maintenance. ACM SIGSOFT Software Engineering Notes, 33(5), 6.
7. Hellerstein, J. M., Stonebraker, M., & Hamilton, J. (2007). Architecture of a database system. Now Publishers Inc.
8. Tripathy, M., & Tripathy, C. R. (2011). On a Virtual Shared Memory Cluster System with VirtualMachines. International Journal of Computer and Electrical Engineering, 3(6), 754.

9. Gopal, T. V., Nataraj, N. K., Ramamurthy, C., & Sankaranarayanan, V. (1996). Load balancing in heterogenous distributed systems. Microelectronics Reliability, 36(9), 1279-1286.
10. Galindo, I., Almeida, F., & Badía-Contelles, J. M. (2008). Dynamic load balancing on dedicated heterogeneous systems. In Recent Advances in Parallel Virtual Machine and Message Passing Interface (pp. 64-74). Springer Berlin Heidelberg.
11. Haroon, M., & Husain, M. (2013). Analysis of a Dynamic Load Balancing in Multiprocessor System. International Journal of Computer Science engineering and Information Technology Research, 3(1).
12. Patel, A., Kusalik, A., & McCrosky, C. (2000). Areaefficient VLSI layouts for binary hypercubes. Computers, IEEE Transactions on, 49(2), 160-169.
13. Li, Y., & Peng, S. (2000, December). Dual-cubes: a new interconnection network for high-performance computer clusters. In Proceedings of the 2000 international computer symposium, workshop on computer architecture (pp. 51-57).
14. PEl-Amawy, A., & Latifi, S. (1991),properties and performance of folded hypercubes. Parallel and Distributed Systems, IEEE Transactions on, 2(1), 31-42.
15. Efe et al., 1991] Efe, K. (1991)A variation on the hypercube with lower diameter. Computers, IEEE Transactions on, 40(11), 1312-1316.
16. Monemizadeh, M., & Sarbazi-Azad, H. (2005, March). The necklace-hypercube: a well scalable hypercube-based interconnection network for multiprocessors. In Proceedings of the 2005 ACM symposium on Applied computing (pp. 729-733). ACM.
17. Rafiq, M. Q., Kumar, P., & Gupta, J. P. (1995). A Novel TreeStructured Multiprocessor Network. In Proceedings of International Conference of on Robotics Vision and Parallel Processing for Automation, Malaysia (Vol. 2, pp. 576-585).
18. Akers, S. B., Harel, D., & Krishnamurthy, B. (1994, June). The star graph: An attractive alternative to the n-cube. In Interconnection networks for highperformance parallel computers (pp. 145-152). IEEE Computer Society Press.
19. Kwak, J. W., & Jhon, C. S. (2007). Torus Ring: improving performance of interconnection network by modifying hierarchical ring. Parallel Computing,33(1), 2-20.