Open access Journal International Journal of Emerging Trends in Science and Technology

Significantly Load Divination and Multiple Processor Support for Manipulation of Resources

Authors

Imran Mulla[1], Karunakar Kothapelli[2]

[1]CSE Department, MLR Institute of Technology, Dundigal, Hyderabad, Telangana, India
Emraan.*mulla@rediffmail.com*

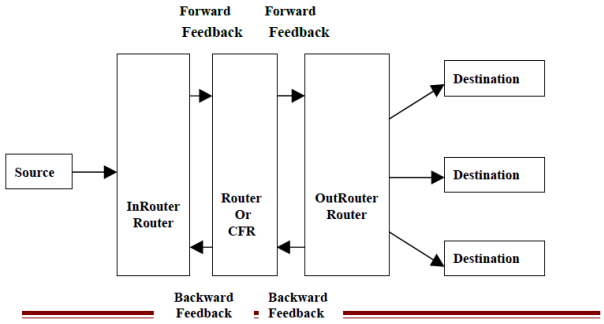
[2]Assistant Professor, CSE Department, MLR Institute of Technology, Dundigal, Hyderabad, Telangana, India.
Email: *karunakar524@gmail.com*

Abstract:
It is generally difficult for a system to ensure load symmetry and resource constraints at a time. When we deal with the resources such as CPU and memory in the task related with real time it is very complicated process of manipulation of resources in multiple processor systems. In simple word resource scheduling and manipulation is a key problem in multiprocessor system. To solve this problem a phenomenon of load symmetry and divination is discussed to integrate and manage resources in multiprocessor system in this paper. This phenomenon significantly predicts the wait time before a packet to be processed based on the load of the processor and the timestamp of the packet. We use simulation experiment to present that the phenomenon can decrease the possibility of packets delay and increase the concurrency of flows that the system can serve easily. It also deduces about migration of packet between processors to guarantee that the packets can be processed before time. This phenomenon increases utilization of multiple processors and also build up the capacity of system to face difficulties occurred in packet migration between processors.
Keywords: *JAVA, Swing(JFC), Eclipse 3.3*

1. INTRODUCTION

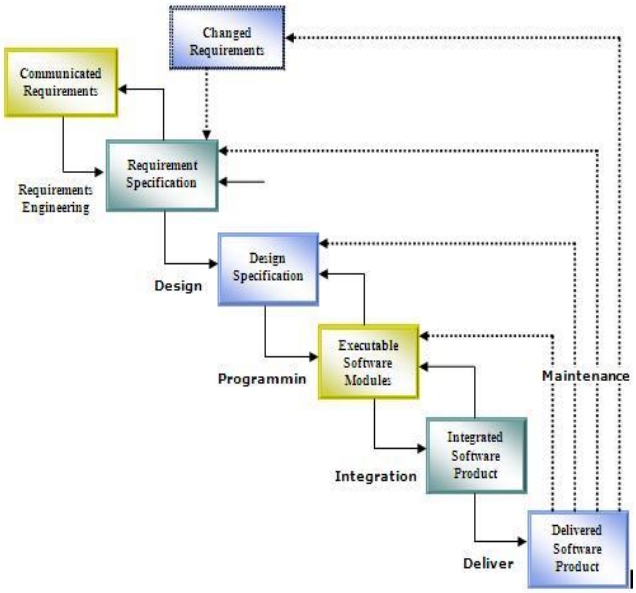
Resource scheduling and management is a key problem in multiprocessor system. Each processor in the system may handle different type of tasks, the need of the resources such as CPU, memory and I/O may be different, so that some processors may not have adequate resources to execute tasks while other processor may not take full use of resources. A system without load balancing policy will suffer from resources wasting and cannot handle more tasks to exert potential of the system. In order to solve this problem, a number of load balancing policies have been proposed, primarily considering the usage of CPU and memory and the combination of resources and current research has shown that the data communications will also affect the performance of load balancing schemes, so that a load balancing policy has to consider a number of factors to adapt system environment becoming more and more complex. With the development of processor technology, multi core framework is gradual stable and mature, more and more fields 1 This work was supported in part by High-Tech Research and Development Plan of China under Grant No.2008AA01A317 begin to take multi core framework as application solutions. But the resource management will be more complex if a system uses multi core processor to accomplish applications, because

multi core will increase flexibility as well as complexity for system to manage resource.



A system using multi core processor has to design framework deliberately to distribute kinds of resources between cores in order to extract the capacity of processing of the system. So resource management on multi core processor is a complex issue, and resource scheduling between multi core processors will be more challenging. In this paper, we will research a special system consisting of two multi core processors which are connected by a bus and embedded on one server board. Because each processor controls independent resources such as cores to process VOD flows, load balancing schemes and resource management can be used in this system to make multic ore processors cooperate when the cores are not enough for processor to complete real-time tasks on time. Specifically, there is timestamp for each packet of flows, so packets should be processed and sent out with real-time constraint. And the rate of flows changes over time, the total rate of flows can changes dramatically in the short time when a processor serves lots of flows, so that the load of each processor of the system will change sharply.

When the load on one processor cannot be completed before timestamp, then the packets will be delayed, however, the CPU resource will be wasted when the load is low on the processor. In order to increase processors utilization and improve the real-time characteristic and the concurrency of the system, we present a load balancing policy which can integrate the resources managed by each processor, so that the system can process more flows simultaneously and guarantee real-time of packets. The reset of the paper is organized as follows. In section 2, the related work is briefly reviewed. Section 3 introduces the system model, and describes the problem in the system. In section 4, a policy is proposed to schedule the resource in the system. In section 5, we introduce a simulation experiment to verify the performance of the system. Section 6 concludes the paper.



2. EXISTING SYSTEM

1) As a result of its strict adherence to end-to-end congestion control, the current Internet suffers from two maladies:

2) Congestion collapse from undelivered packets, and unfair allocations of bandwidth between competing traffic flows.

3) The first malady — congestion collapse from undelivered packets — arises when packets that are dropped before reaching their ultimate continually consume bandwidth destinations.

4) The impact of emerging streaming media traffic on traditional data traffic is of growing concern in the Internet community. Streaming media traffic is unresponsive to the congestion in a network, and it can aggravate congestion collapse and unfair bandwidth allocation.

3. PROPOSED SYSTEM

To address the maladies of congestion collapse we introduce and investigate a novel Internet traffic control protocol called *Congestion Free Router* (CFR). The basic principle of CFR is to compare, at the borders of a network, the rates at which packets from each application flow are entering and leaving the network. If a flow's packets are entering the network faster than they are leaving it, then the network is likely buffering or, worse yet, discarding the flow's packets. In other words, the network is receiving more packets than it is capable of handling. CFR prevents this scenario by "patrolling" the network's borders, ensuring that each flow's packets do not enter the network at a rate greater than they are able to leave the network. This patrolling prevents congestion collapse from undelivered packets.

4. CONCLUSIONS

In this paper a load symmetry policy using on multiprocessor platform is presented, and introduce a pipeline frame to arrange the multicore in the processor to fully exert the capacity of the processor. On this basis, we give a load balancing policy considering the real-time constraint and integrating the resources of processors, and the policy can effectively balance the load between processors so that increase the processor utilization and improve the concurrency and stability of flows that the system serves. The experiment result proves that the load balancing policy can enhance the capacity of the system to resist jitter and the performance improvement on packets delay and concurrency of flows is significant.

REFERENCES

1. F. Douglis and J. Ousterhout, "Transparent process migration: design alternatives and the sprite implementation, Software Practice & Experience", vol.21 no.8, pp.757-785, Aug. 1991.
2. Harchol-Baker and A. Downey, "Exploiting process lifetime distributions for dynamic load balancing", *ACM Transactions on Computer Systems (TOCS)*, vol.15 no.3, Aug. 1997
3. C.-C. Hui and S.T. Chanson, "Improved strategies for dynamic load balancing", *IEEE Concurrency*, vol.7, no.3, pp.58-67, Jul-Sep 1999.
4. Acharya and S. Setia, "Availability and Utility of Idle Memory in Workstation Clusters", *Proc. ACM SIGMETRICS Conf. Measuring and Modeling of Computer Systems*, pp. 35-46, May 1999.
5. X. Zhang, Y. Qu and L. Xiao, "Improving Distributed Workload Performance by Sharing Both CPU and Memory Resources," *Proc. 20th Int"l Conf. Distributed Computing Systems, (ICDCS "2000)*, pp. 233-241, Apr. 2000.
6. L. Xiao, X. Zhang and Y. Qu, "Effective Load Sharing on Heterogeneous Networks of Workstations", *Proc. 2000 Int"l Parallel and Distributed Processing Symp., (IPDPS"2000)*, pp. 431-438, May 2000.

AUTHOR NAME

Imran Mulla Currently Pursuing Master of Technology in CSE from MLR Institute of Technology, Dundigal, Hyderabad, Telangana, India

Karunakar Kothapelli B.Tech, M.Tech CSE He is currently working in the Department of Computer Science and Engineering, MLRIT, Telangana, India. He is having 02 years of teaching experience**.** His research interesting areas are Ad-Hoc and sensor networks.