Modeling Serverized UNICOS on the T3E Architecture

Bruce Schneider, Cray Research, Inc., System Performance and Analysis Section, Eagan, Minnesota

Introduction

The UNICOS operating system is being reorganized into a microkernel based modular distributed operating system called UNICOS/mk. This serverized UNICOS is intended to support the T3E massively parallel system. Running a distributed operating system on a distributed architecture offers many complex performance challenges. This presentation details three areas of the system we are modeling to gain a better understanding of their associated performance. The three areas of focus are: T3E interconnect, SCX channel and UNICOS/mk I/O.

T3E Hardware Network Interconnection

One of the lowest levels of communication is the T3E hardware interconnection network. This interconnection network provides communication among the PE's. The network is a 3D torus with possible partial planes. The speed of this interconnect could have an impact on a number of performance concerns. We created a model of the network interconnection with the purpose of looking at a number of potential concerns.

One area of concern was if the network overhead would affect the possible placement of UNICOS/mk servers. Portions of the distributed operating system can be spread across multiple PE's. This implies that there is a potential for large system communication occurring between PE's. There was a concern about the overhead of communicating between servers residing on various PE's and if the placement of these servers on individual PE's would have an impact upon performance. Related to server placement was the concern if the SCX I/O location could be affected due to network overhead.

The message load generated by the operating system for communicating among the PE's was of interest. The message load on the interconnect was looked at by generating a number of message scenarios where PE's sent control packets to other PE's. Messages were sent from: a single PE to all PE's, all PE's to a single PE, all PE's to all PE's, all PE's to the nearest neighbor PE, all PE's to a PE neighbor two hops away, and etc.

One outcome of the model was to verify the hardware routing table information and optimization. Actual routing table information was used in the model and by running the model, blips would have occurred if the tables were not correct.

Another area of use for this model was as a low level component of the UNICOS/mk I/O model. Since I/O involves a large number of PE to PE communication, the network overhead should be factored in.

The preliminary results of this modeling showed that the hardware message latencies are not a factor in sever or SCX placement. The overhead of the network is minimal no matter the distance or number of hops associated with the message. Also, the system interprocessor communication calls do not significantly impact network load. The amount of PE to PE communication occurring occupies a very small percentage of the total network bandwidth available. The available bandwidth on the network is more than capable of handling data transfers along with the PE to PE system communication.

SCX Channel

The SCX channel is based on a pair of unidirectional, counter-rotating rings. SCX performance is of major concern since it is the channel by which all communication from the mainframe to peripherals is made. The SCX channel provides a great deal of flexibility, both in how systems are connected, and how data transfers are performed. This same flexibility, however, significantly complicates the performance model. Multiple transfers, involving multiple clients, can be simultaneously active on the same channel. The performance of a given transfer is affected not only by the topology and other transfers active on the channel, but by several factors specific to the transfer in question: which node is mastering the transfer, the speeds of the client ports involved, and whether the transfer is being performed using reads or writes.

The model we are creating is analyzing the performance based upon: traffic patterns and routing, transfer size, client port speeds and client service rates. The goal is to make sure the SCX channel is capable of handling all the data and communication traffic needs of any possible customer workload. The model should also be able to aid in determining the appropriate SCX configuration for various sizes and numbers of peripherals.

UNICOS/mk I/O

On the software side, I/O is one of the most critical performance characteristics of a system. Because of this, we are

modeling the software I/O path to look for potential system I/O bottlenecks. The goal is to investigate I/O performance and scaling on a distributed operating system. As the number of PE's increase, can the current I/O implementation and the NC1 filesystem support the I/O demands? We have the basic model completed and are running software I/O design alternatives through the model to look for ways to improve I/O performance.

The model uses I/O path trace information from the UNICOS/mk running on the T3D MPP. We validated the model by comparing an actual T3D I/O test run to a modeled run. The modeled results showed an error of less than 10% compared to the actual run. This low margin gave us confidence in the ability to do design prediction. We then took the T3D timings and extrapolated to the T3E. From there, we have run a series of "what if's" related to design ideas. Example I/O design ideas include: having all the I/O servers on one PE, splitting the I/O servers onto multiple PE's and creating multiple I/O servers. We have looked at performance if some of the I/O communication overhead was decreased. In all cases we have been able to predict performance of design ideas without having to actually spend man-months implementing code.

Summary

Putting a serverized UNICOS operating system onto a massively parallel system presents many challenges related to performance. The above examples are just some of the areas we have focused on. We are working to build the highest possible performance into the design of the system by modeling key aspects of the system.