Network Survivability Simulation of a Commercially Deployed Dynamic Routing System Protocol

Abdur Chowdhury[1,2], Ophir Frieder[1], Paul Luse[2], Peng-Jun Wan[1]

{abdur, wan, ophir}@cs.iit.edu, pluse@iitri.org
Department of Computer Science
Illinois Institute of Technology[1] and
IIT Research Institute[2]

Abstract. With the ever-increasing demands on server applications, many new server services are distributed in nature. We evaluated one hundred deployed systems and found that over a one-year period, thirteen percent of the hardware failures were network related. To provide end-user services, the server clusters must guarantee server-to-server communication in the presence of network failures. In prior work, we described a protocol to provide proactive dynamic routing for server clusters architectures. We now present a network survivability simulation of the Dynamic Routing System (DRS) protocol. We show that with the DRS the probability of success for server-to-server communication converges to 1 as N grows for a fixed number of failures. The DRS's proactive routing policy performs better than traditional routing systems by fixing network problems before they effect application communication.

INTRODUCTION

Traditional supercomputers are becoming scarce and distributed server clusters are becoming the solution of choice. These smaller computers are coupled by networks to achieve the same objective at a substantially lower cost. The Berkley NOW (Network Of Workstations) project was one of the first projects pushing this solution [2]. PVM (Parallel Virtual Machine) [3] and MPI (Message Passing Interface) [4] libraries provide messaging and synchronization constructs that are needed for distributed parallel computing with NOW solutions. Projects like Beowulf [5] for Linux are continuing the distributed computing approach. All of these approaches have one common resource, the network. While the network is very important, no strong push has been made to provide fault tolerance for network failures in a server cluster solution.

We developed a network routing algorithm to provide fault-tolerance for server-to-server communication by proactively monitoring network communication links between servers. This is different from reactive routing techniques [6] that wait for a failure to occur and then react by finding an alternative route. Our proactive algorithm constantly looks for errors via continuous ICMP echo requests. When a failure is identified, a new route is selected around the failed portion of the network. This new

J. Rolim et al. (Eds.): IPDPS 2000 Workshops, LNCS 1800, pp. 1281-1285, 2000.

route is often found in the time of a TCP retransmit, so server applications are unaware that a network failure has occurred.

Our algorithm, the Dynamic Routing System (DRS) [1], improves reliability by providing a second network interface card for each server thus providing an alternate method of physical communications in the case of hardware failure. The DRS works by frequent link checks between all pairs of nodes to determine if the link between pairs of computers is valid. This algorithm uses the redundant network link between two servers to provide multiple communication channels. When one link fails, the second direct link is checked and used. However, if no link exists, a broadcast is made to identify whether or not some other server is able to act as a router to create a new path between the sender and the proposed recipient. Our algorithm discovers the failure before server-to-server communication is affected. The essential goal of our algorithm is to hide network failures from distributed applications.

The DRS was deployed in 27 local voice mail server clusters by MCI WorldCom, each cluster contains between 8 and 12 servers. Thus understanding the reliability supported is not only of theoretical interest but of practical interest as well. In prior work [1] we showed that, over a one year period, 13% of hardware failures for 100 compute servers were network related, i.e., network interface cards, hubs, etc. This likelihood of failure provides motivation to improve the resilience of server clusters where services need to be guaranteed. We show that, with the DRS, the probability of success of server-to-server communication converges to 1 as N grows for a fixed number of failures. The proactive routing policy of the DRS performs better than traditional routing systems by fixing network problems before they effect the server-to-server communication.

DRS ALGORITHM

RIP [7], OSPF [8], EGP and BGP [9] are routing solutions to many different routing problems, however, they do not address the needs of a high availability server cluster environment [11]. Their primary goal is to provide routing updates to other routers on the network to find alternative routes to the same network. The general design goal is based on reactively rerouting when a specified timeout period has been reached. So if a destination network does not respond to a route query, after some time quantum, it is considered down and a new route is sought after.

The DRS works with IP networks unlike some telecommunication approaches using specialized hardware [10] and improves fault tolerance via proactive failure recognition and the use of a redundant network. Thus, each computer has two network interface cards connected to two separate networks. It is the task of the DRS routing demons to monitor the connections between two servers. If a failure occurs, the demons set up new point-to-point routes around the problem before network applications are aware that a problem occurred.

The DRS runs on every node in the server array. Each DRS demon is configured to monitor hosts on the networks and executes a two stage run process. In the first phase, the communications links between the local host and all other hosts that is it has been configured to monitor are checked. These checks are accomplished using

the ICMP (Internet Control Message Protocol) [13] echo request. Host "A" sends an ICMP echo request to host "B" via the first network. If the echo is returned, the DRS can assume that the hub, wiring, network interface card, device driver, network protocol stack and host kernel, are operational. The DRS continues to test all known hosts on all known networks in the same manner.

Each demon keeps track of which hosts to monitor and the state that they are in (i.e., "up", "down"). If a failure occurs, the DRS demon must determine a new route of communication between host "A" and "B". The DRS demon loops through a cycle of monitoring communication links, answering requests, and fixing problems as they occur, for the life of the server cluster. The DRS algorithm avoids routing loops and other issues involved in distributed routing. For a detailed presentation and proof of correctness see [1].

DRS PROACTIVE COST

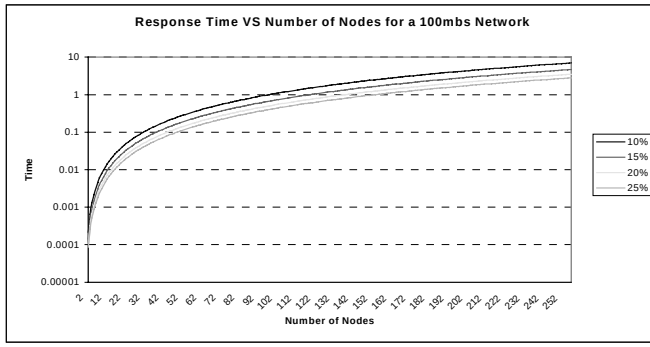

Figure 1: 100Mb Network Performance

The DRS's proactive monitoring of network links comes at a cost of network bandwidth. To find errors before they effect network communication, the links must be checked frequently. If the links were not checked frequently, the DRS would become equivalent to a reactive routing protocol. As the number of nodes increase, the bandwidth required to support the frequent checks likewise increases. In Figure 1, we present the maximum number of servers in the cluster that the DRS supports given a requirement for error resolution in X time units and the percentage of network bandwidth useable by the DRS. As show in Figure 1, ninety hosts are supported in less than 1 second with only 10% of the bandwidth usage.

NETWORK SURVIVABILITY ANALYSIS

We now present a conditional probability model like [12] to quantitatively evaluate networking systems with a given number of network failures occurring at any given instance. This model yields the probability of success, independent of time, of a system with N nodes and f failures.

We assume that in a system with N nodes, there are exactly 2N interface connections and two non-meshed back planes, each with equal probability of failure,

say q, for $0 \le q \le 1$. Therefore, the probability of 2 failures in any system will be q^2, the probability of 3 failures will be q^3, and the probability of f failures will be q^f. It follows that $\lim_{f \to \infty} q^f = 0$. Therefore, the probability of multiple failures in a system decreases exponentially.

Now we develop the equation for the probability of success by counting the number of possible failure combinations for a system with N nodes and f failures. We represent this number by the combinatorial function F(N,f), with

$$F(N,f) = \binom{2N}{f-2} + 2 \cdot \left[\binom{2N}{f-1} - \binom{2N-2}{f-1} \right] + \binom{2N-2}{f-2} + 2 \cdot \left[\binom{2N-4}{f-3} + \binom{N-2}{f-N} \cdot 2^{2N-f-2} \right] + \binom{2N-4}{f-2}$$

Because the total number of combinations in a networking system is $\binom{2N+2}{f}$, the probability of success can be written, as shown in Equation 1.

$$P[\text{Success}] = \frac{\binom{2N+2}{f} - F(N,f)}{\binom{2N+2}{f}}$$

Equation 1: Probability of Success

By graphing Equation 1 for fixed values of f, it is evident that as the number of nodes in a system increases, the probability of that system maintaining a successful connection between any two nodes at any given time will approach 1 using the DRS. More specifically, for f=2 the P[S] surpasses 0.99 at 18 nodes. For f=3 the P[S] surpasses 0.99 at 32 nodes, and for f=4 the P[S] surpasses 0.99 at 45 nodes. Given that $\lim_{f \to \infty} q^f = 0$ and that $\lim_{N \to \infty} P[S] = 1$, a system implementing the DRS has a high probability of resilience to network failure, as show in Figure 2.

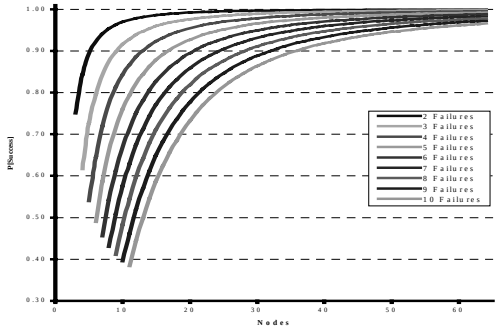

Figure 2: Convergence of P[Success] to 1

DRS Simulation

To validate our probability model, we have developed a computer simulation of a networking system with N nodes and f failures implementing the DRS algorithm. Given a specified number of iterations and a fixed f, the simulation output consists of randomly generated success probability values for f<N<64. The graph in Figure 3 displays the convergence of the simulation outputs to the actual equation values for two through ten network failures as we increase the number of iterations. The y-axis represents the mean absolute difference between the simulation output and the equation value for f<N<64. The x-axis represents the number of iterations in log10 scale. With 1,000 iterations, the mean absolute difference is less than 0.009 for each of the fixed f values, and as the number of iterations increases the mean absolute difference converges to zero. Therefore, the simulation results support the probability model of Equation 1 given in the prior section.

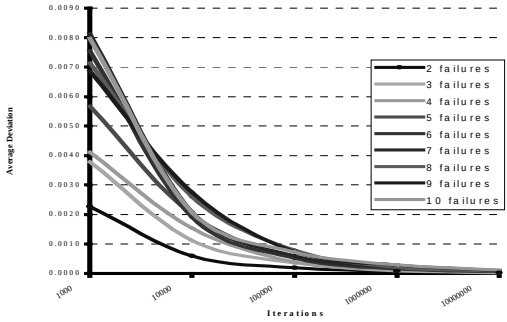

Figure 3: Convergence of Simulation Results to Equation Results

CONCLUSIONS

The DRS algorithm provides a reactive routing protocol for tightly coupled server clusters of the given topology. These server clusters do not have elaborate network topologies since server-to-server communication of the cluster is of concern. We provided a brief review of the DRS algorithm as described in detail in [1]. We provided a probability model to quantitatively evaluate the DRS algorithm resilience to network failures. The model gives a conditional failure probability of the entire system. Using Equation 1, we showed that the probability of success converges to 1 as N gets large for fixed values of f. More specifically, for f=2 the P[S] surpasses 0.99 at 18 nodes. For f=3 the P[S] surpasses 0.99 at 32 nodes, and for f=4 the P[S] surpasses 0.99 at 45 nodes. Given that $\lim_{f \to \infty} q^f = 0$ and that $\lim_{N \to \infty} P[S] = 1$, a system implementing the DRS has a high probability of resilience to network failure. To validate this model we present a validation simulation of the DRS algorithm.

References

1 A. Chowdhury, et. al., "Dynamic Routing System (DRS): Fault tolerance in network routing", Computer Networks And ISDN Systems (31) 1-2 (1999).
2 T.Anderson, et.al, "A Case for NOW (Networks of Workstations)", IEEE Micro 15 (1995).
3 J. Casas, et.al. "Adaptive Load Migration Systems for PVM". Supercomputing 94, November 1994.
4 M. Snir, et.al, "MPI: The Complete Reference", The MIT Press, 1996
5 C. Reschke, et. al, ‘‘A Design Study of Alternative Network Topologies for the Beowulf Parallel Workstation,”, IEEE High Performance Distributed Computing, 1996.
6 G. R. Ash, Dynamic Routing in Telecommunications Networks, McGraw Hill, 1998.
7 C. Hedrick, Request For Comment 1058, "Routing Information Protocol", 06/01/1988, http://ds.internic.net/ds/dspg2intdoc.html
8 J.Moy, Request For Comment 1583, "OSPF Version 2", 03/23/1994, http://ds.internic.net/ds/dspg2intdoc.html
9 K. Varadhan, Request For comment 1503, "BGP OSPF Interaction", 01/14/1993, http://ds.internic.net/ds/dspg2intdoc.html
10 B. R. Hurley, C. J. R. Seidl, and W. F. Sewell. "A Survey of Dynamic Routing Methods for Circuit-Switched Traffic". IEEE Communications Magazine, 25(9), September 1991.
11 S. Low, P. Varaiya, "Stability of a class of dynamic routing protocols (IGRP)". In IEEE Proceedings of the INFOCOM, volume 2, pages 610--616, March 1993.
18 R. Talbott, "Network Survivability Analysis", Fiber & Integrated Optics, Vol. 8, 1988.
19 J. Postel, "Internet Control Message Protocol (ICMP)", RFC 792, 1981