Review Communication Middleware for Multiple Network Interface

Muhammad Farhan Sjaugi, Mohamed Othman, S. Napis

fhn@thinkparallel.org

Outline

- Introduction
- Problems
- Approaches to Increase Bandwidth in WAN
- Scenarios
- Related works on multiple network interfaces communication
- Conclusions
- Reference

Introduction

- Applications such as network-based multimedia storage, remote satellite observations, distributed data mining, distributed scientific simulations, and distributed geographic information systems are both **compute intensive**, requiring scalable high-processing power, and **data intensive**, demanding reliable, scalable high-bandwidth communication infrastructure for high-volume and high-speed data access.
- One of the main challenges in developing infrastructure services for Cluster and Grid computing is the heterogeneity in machine architectures, operating systems, and network resources.

Introduction (cont)

- This heterogeneity forces the services to be implemented at the middleware-level so that they will be easily ported to and utilized by different types of systems
- Grid computing aims to utilize available distributed software and hardware resources on a large scale spanning the whole nation and even the globe
- This requires the cluster and grid services to be scalable to efficiently utilize the available resources
- One solution for expandability of processing power and storage is to add mode nodes and storage units to the cluster or grid, similarly by adding more network interfaces and connections can increase the total communication bandwidth among the cluster nodes and grid components

Problems

- However, current network protocols, software, and APIs such as Sockets are designed for single physical network interface that is rising the need of protocols, and software services that can support multiple physical network interfaces on each node and provide transparent and efficient utilization of these interfaces
- The available software technologies are not yet adequate to seamlessly handle the variety of possible configurations of hardware components
- For example, the configurations of most cluster systems include two or more network interface cards (NIC) per node, but the applications cannot seamlessly utilize the NICs simultaneously

Problems (cont)

- Another problem that faces applications on clusters is the lack of information available to each application about the NIC utilization by the other applications
- For example, it could easily happen that one application starts using one network and then the next application binds its sockets to the same network even though the second network is free
- To satisfy the application's demand for higher bandwidth with the current technology, the existing networks have to be replaced by more advanced/faster networks, which is a costly and non-scalable solutions

Problem (cont'd)

- One of the major requirements of Grid computation is the availability of high bandwidth and low latency networks that could support huge data transfers efficiently.
- In an effort to provide high bandwidth and low latency solutions, many protocols, algorithms and techniques have been devised.
- Most of these applications, particularly the data-intensive applications, require large data transfers, which are mostly in the form of message-passing or file transfer.

Problem (cont'd)

- However, the current transport protocols such as TCP and UDP impose many limitations that limit the utilization of available bandwidths and reduce the performance of these applications.
- The bandwidth on WAN is affected by many factors that prevent applications from reaching their theoretical peak performances during transmission.

Limitation on Scalability of Bulk Data Transfers

- These limitations are identified as follows [1]:
  - Protocol Limitations
  - Network Limitations
  - Network Interface Card Limitations
  - Heterogeneity Limitations
  - Implementation Limitations
  - Other System Component Limitations

Approaches to Increase Bandwidth in WAN

- There are several approaches to increase bandwidth in WAN [1], includes:
  - Tuning TCP Protocol Parameters
    - This approach requires changes to the TCP protocol, which in many cases may not be accessible to the developer.
    - Most of these techniques, however, cannot overcome other limitations such as the network, NIC, processing power and others.
  - Using Multiple Parallel TCP Streams
    - Multiple logical connections are established to transmit data at a higher rate such that the available bandwidth is saturated quickly.
    - One advantage of this approach is that it can be implemented at the middleware level, thus making it highly portable.
    - However, using this technique on a single machine limits the bandwidth to the maximum available bandwidth on the NIC used, subject to the system components limitations.

Approaches to Increase Bandwidth in WAN

- Using the Stripping Technique at Different Levels of the Communication Protocol Stack
  - in the context of networks striping is used to describe the aggregation of multiple networks to achieve higher bandwidth, hence higher throughput.
  - The aim of the stripping algorithms is to distribute varying sized transmission units into multiple available connections.
  - In addition, the striping algorithms try to achieve load balancing for multiple networks and fairness in serving packets and frames coming from higher layers.
  - The technique does not guarantee utilizing peak bandwidth within each network interface used.

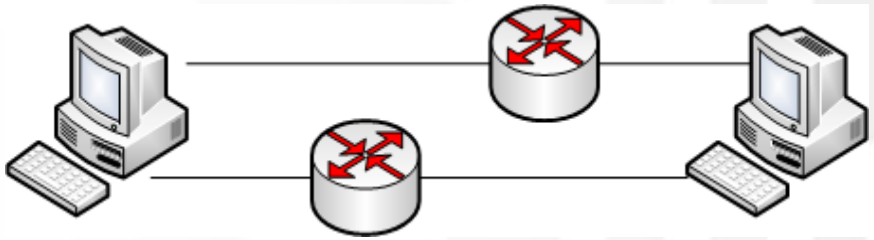
Approaches to Increase Bandwidth in WAN

- Using UDP-Based Techniques
  - Many approaches moved towards using UDP to avoid the limitations of TCP.
  - UDP-based protocols provide efficient data transfers. These techniques can be implemented at the middleware level, thus providing high portability.
  - However, this approach requires developing mechanisms for flow control and reliability, which impose some overhead.
  - On the other hand, others tried to avoid these costly mechanisms by confining the solutions to dedicated high bandwidth networks or quality of service (QoS) enabled networks.

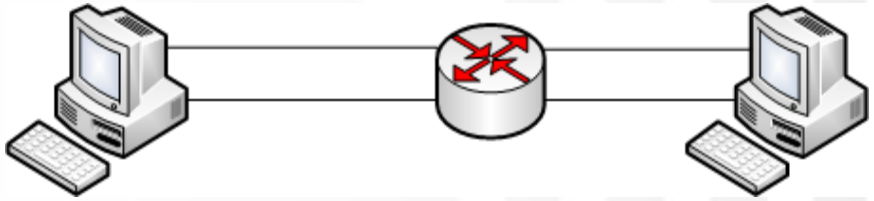
Approaches to Increase Bandwidth in WAN

- Providing alternative transport protocols
  - The main advantage here is that the new protocols are not restricted to any limitations imposed by the standard protocols.
  - Thus, they can be designed to provide the most efficient mechanisms for data transfer.
  - However, this creates a problem of compatibility since all participating parties must support the new protocol.

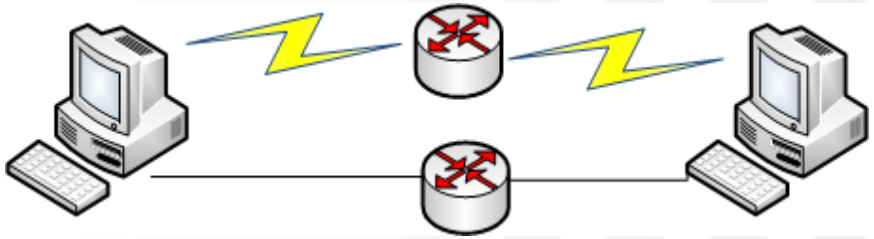
Scenarios (multiple interfaces in local area network)



Multiple homogeneous networks: Machines connected through multiple homogeneous interfaces and networks

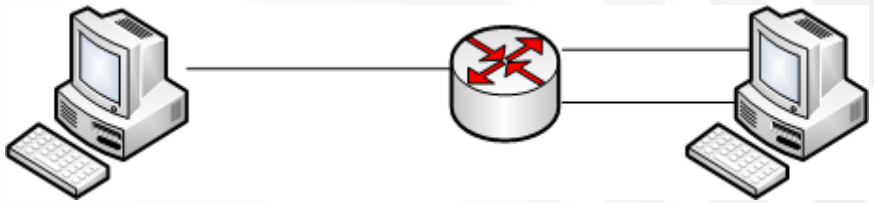


Multiple homogeneous interfaces and single networks: Machines connected through multiple homogeneous interfaces and one network

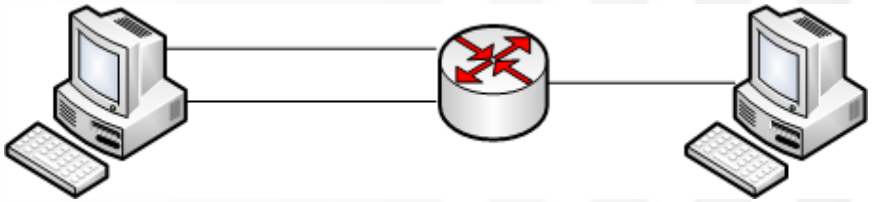


Multiple homogeneous networks and interfaces: Machines are connected through wired and wireless networks

Scenarios (multiple interfaces in local area network)

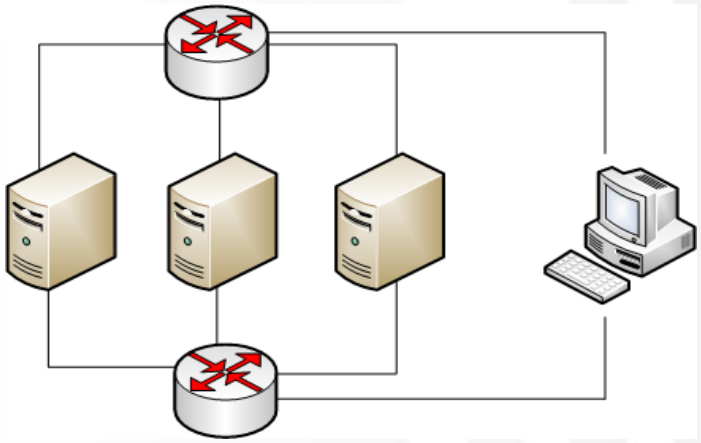


Multiple homogeneous interfaces, heterogeneous number of interfaces, and single homogeneous networks

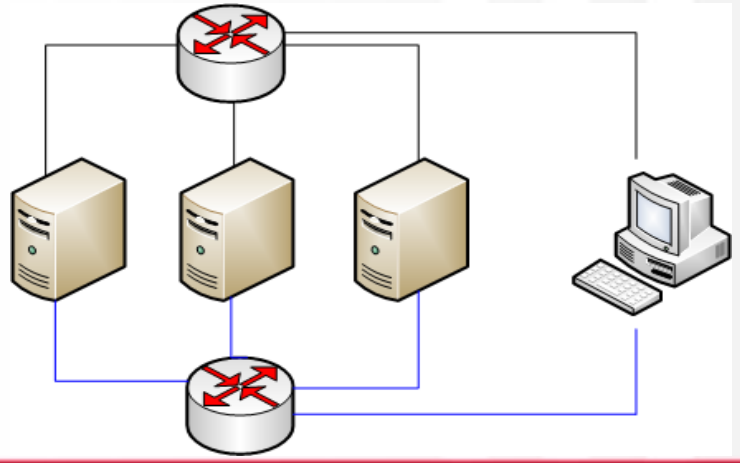


Multiple heterogeneous interfaces, heterogeneous number of interfaces, and single network with heterogeneous links

Scenarios (multiple interfaces in system area networks network)

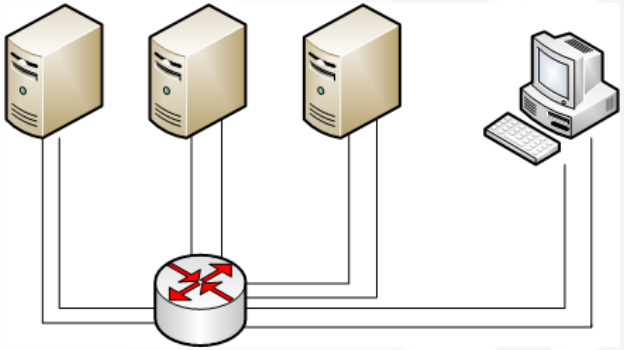


Multiple-Homogeneous-Network Switched Cluster

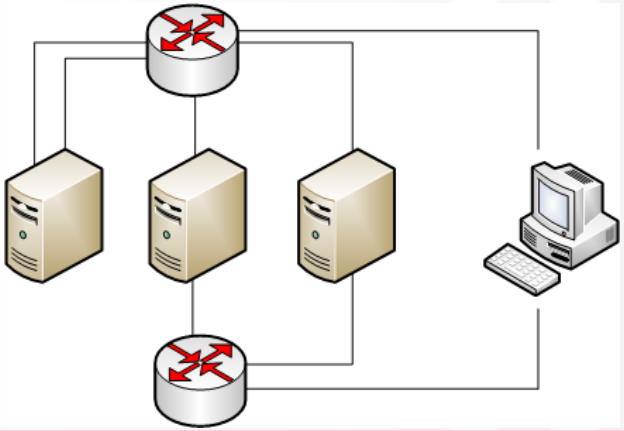


Multiple-Heterogeneous-Network Switched Cluster

Scenarios (multiple interfaces in system area networks network)

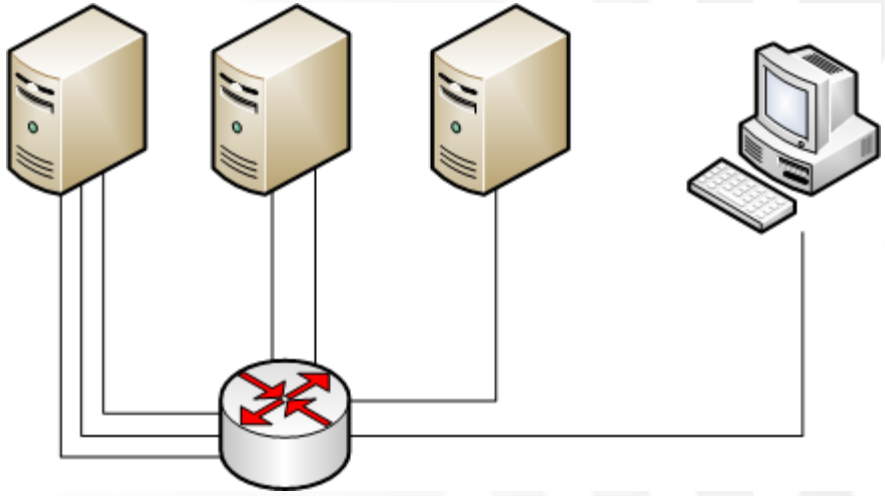


Multiple-Homogeneous-Interface switched cluster, all network interfaces are connected through single switch



Multiple-Heterogeneous-Network, switched cluster with the file system node connected through multiple interfaces

Scenarios (multiple interfaces in system area networks network)



Heterogeneous Switched Cluster. In Heterogeneous cluster, nodes may have different network needs and capacities. Multiple interfaces can be installed in high capacity machines

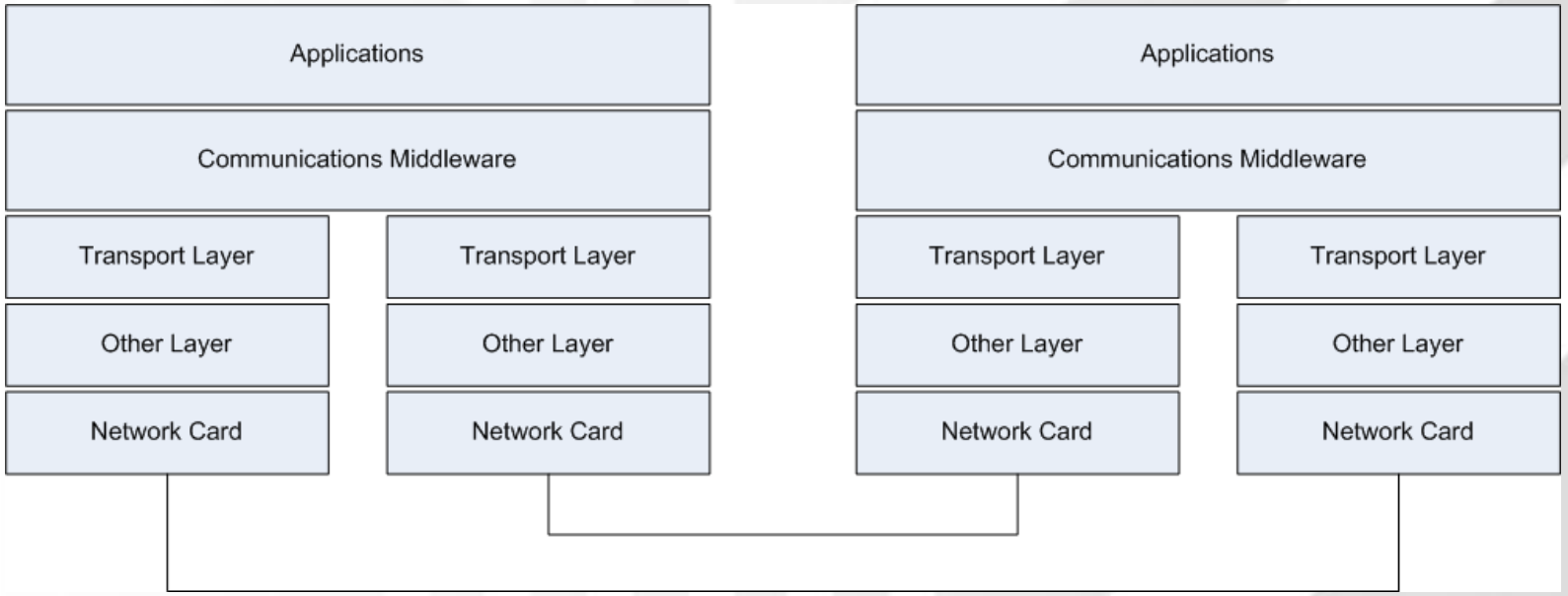
Related works on multiple network interfaces communication:

- MuniSocket [2] and MuniCluster [3]
- Concurrent-Multipath Transfer [4,5,6]
- Fault Tolerant Ethernet [7]

MuniSocket

- MuniSocket [2] provides parallel message fragmentation and reconstruction mechanisms in addition to load balancing.
- The main difference between MuniSocket and the standard Socket is that MuniSocket processes and transfers large user messages in parallel, fully utilizing the existing multiple network interconnects, while the standard Socket processes and transfers messages sequentially through a single network interface.
- MuniSocket has the potential of providing expandable bandwidth, load balancing, and fault-tolerance for data-intensive applications running on clusters or grids connected by multiple interconnection networks.

MuniSocket Architecture



MuniCluster Architecture

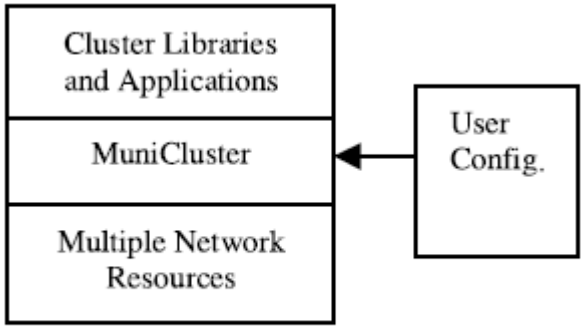


Fig. 1 The proposed model architecture.

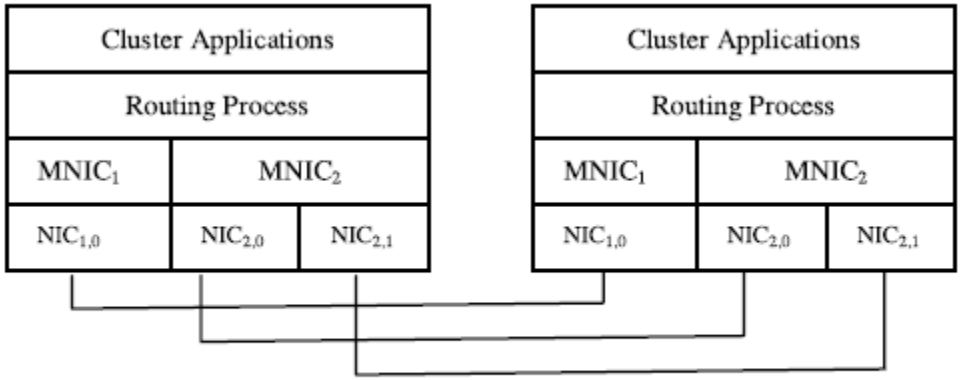


Fig. 2 Multiple network configurations.

Concurrent Multipath Transfer

- *Concurrent Multipath Transfer (CMT) [4,5,6] is the simultaneous transfer of new* data from a source host to a destination host via two or more end-to-end paths.
- CMT is used to increase throughput for a networked application.
- The current transport protocolworkhorses, TCP and UDP, are ignorant of multihoming;
- TCP allows binding to only one network address at each end of a connection.
- At the time TCP was designed, network interfaces were expensive components, and hence multihoming was beyond the ken of research.

Concurrent Multipath Transfer

- Two recent transport layer protocols, the Stream Control Transmission Protocol (SCTP) , and the Datagram Congestion Control Protocol (DCCP) support multihoming at the transport layer.
- The motivation for multihoming in DCCP is mobility, while SCTP is driven by a broader and more generic application base, which includes fault tolerance and mobility.
- A naıve form of CMT can be obtained by simply modifying the SCTP sender to transmit new data to all destinations.

Fault-Tolerant Ethernet

- The FTE [7] has been developed to address the following goals:
  - Multiple faults must be detected.
  - In the presence of multiple faults, any peer nodes should be able to communicate as long as a healthy physical link exists between them.
- consider two types of network architectures for realizing the fault-tolerance capability.
- One is based on the philosophy of redundancy, by which two or more independent networks are employed such that at least one of them continues to function in the presence of one or more network faults.

Fault-Tolerant Ethernet (cont'd)

- The other type of fault-tolerant network follows the philosophy of self-healing. It is structured as a single network with reliable multipath.

Failure Detection at FTE

- Two peer nodes, i and j, communicate with each other via FTE. Node i sends out a pair of diagnostic messages periodically, one along channel 1 and the other along channel 2
- Each message carries a sequence number. node j , once it has received one diagnostic message from node i on one channel, compares the sequence number of the message with that of the last message received from the other channel.
- If there exists a certain sequence number difference between the two messages from channels 1 and 2, node j declares a potential network failure.

Failure Detection on FTE (cont'd)

- To detect multiple faults, a node needs the state information about the health of the entire network.
- Specifically, each FTE node establishes and maintains a global view based on what it observed, called My View, and what other nodes observed, called Peers' View.
- My View is the network status detected by a node itself with received diagnostic messages.
- Peers' View is the network status information sent by the other nodes.
- With this information, the node can detect multiple faults

Failure Recovery on FTE

- The middleware-based FTE provides applications with network redundancy transparency.
- This is achieved by exposing to applications only one of the IP addresses associated with N-channel interfaces of a node.
- Each node keeps the destination IP address in the ARP table associated with a MAC address of the default channel (channel A).
- If the healthy channel of the destination node is not the default channel (channel B), the data packet (MAC address is ww) will not be received by the destination node, because the receiving NIC has a different MAC address (MAC address is zz).

Failure Recovery on FTE (cont'd)

- To solve this problem, each FTE node maintains a MAC Address Resolution Table (MART) that contains the N channel MAC addresses of every peer FTE node.
- The "NIC Switch" of the sender node resolves the destination MAC address of every frame sent down from the IP layer by looking up MART.

More solutions

- Channel Bonding [8]
- IPNMP (Solaris IP Network Multipathing) [9]
- Heterogeneous network utilization [10]
- Multirail [11]

Conclusions

- Typical application in Cluster and Grid computing are need both compute and data intensives.
- One solution for expandability of processing power and storage is to add mode nodes and storage units to the cluster or grid, similarly by adding more network interfaces and connections can increase the total communication bandwidth among the cluster nodes and grid components.
- However It also will increase the complexity of the Cluster and Grid systems, especially in the heterogeneous system where many possible configurations of hardware and software components.
- The available software technologies are not yet adequate to seamlessly handle the variety of possible configurations of hardware components.
- There are several solutions available to increase data communication performance in Cluster and Grid systems. One of the solution is by exploiting multiple network interfaces.

Conclusions

- Change the transport protocol in order to deal with multiple networks. However it is good for some systems, it has limitation since not all operating systems provide easy ways to access and change the transport protocol implementations
- Implementation as middleware APIs or sockets which is give transparency to the applications on how the communication is done. This will gives lots of flexibility to both homogeneous and heterogeneous systems to exploit multiple network interfaces for communication. However this APIs may be available in specific programming language only. A multi-platform programming language is a good choice to answer this issue

Conclusions

- There are several implementations in order to exploit multiple network interfaces, one of them is implementation at the middleware level
- The main advantage of middleware level implementation is it can gives transparency and flexibility to support data intensive Cluster and Grid applications

Reference

1. Mohamed, N., Al-Jaroodi, J., Jiang, H., and Swanson, D. 2006. *High-performance message striping over reliable transport protocols*, Journal of Supercomputer. 38, 3 (Dec. 2006), 261-278.
2. N. Mohamed, J. Al-Jaroodi, H. Jiang, S. Swanson, *A user-level socket layer over multiple physical network interfaces*, 14th International Conference on Parallel and Distributed Computing and Systems, Cambridge, USA, November 2002, pp. 810-815.
3. N. Mohamed, J. Aljaroodi and H. Jiang, *Configurable Communication Middleware for Clusters with Multiple Interconnections*, *IEICE Transaction of Information and Systems* (Special Issue on Hardware/Software Support for High Performance Scientific and Engineering Computing), Vol. E87-D, No. 7, pp.1657-1665, July 2004.
4. Preethi Natarajan, Nasif Ekiz, Janardhan Iyengar, Paul D. Amer and Randall Stewart, *Concurrent Multipath Transfer Using Transport Layer Multihoming: Introducing the Potentially-failed Destination State,* IFIP Networking 2008, Singapore, May 2008.
5. Preethi Natarajan, Janardhan Iyengar, Paul D. Amer and Randall Stewart, *Concurrent Multipath Transfer Using Transport Layer Multihoming: Performance Under Network Failures*, MILCOM, Washington D.C., October 2006.
6. Preethi Natarajan, Janardhan Iyengar, Paul D. Amer and Randall Stewart, *SCTP: An Innovative Transport Layer Protocol for the Web*, 15th International Conference on World Wide Web, Edinburgh, May 2006.
7. Song, S., Huang, J., Kappler, P., Freimark, R., and Kozlik, T. 2000. Fault-tolerant Ethernet middleware for IP-based process control networks. In *Proceedings of the 25th Annual IEEE Conference on Local Computer Networks* (November 08 - 10, 2000). LCN. IEEE Computer Society, Washington, DC, 116.
8. Beowulf Ethernet Channel Bonding web page, At http://www.beowulf.org/software/bonding.html, 2002
9. Solaris IP Network Multipathing (IPNMP) Product Document, Sun Microsystems Inc., http://docs.sun.com/db/dov/816-0850/6m7adiu4a?a=view
10. J.Kim and D.Lilja, *Exploiting Multiple Heterogeneous Networks to Reduce Communication Costs in Parallel Programs*, Proc. of the 6th Heterogeneous Computing Workshop (HCW'97), 1997.

TQ ☺