

Cluster Interconnect Analysis and Comparison White Paper



Overview.....	3
The Cluster Interconnect Functions.....	3
Cluster Executive Components	4
Cluster I/O.....	5
Cluster Networking	5
Reliable Datagrams	5
Application-Specific Traffic.....	5
Configuration Assumptions	6
Hardware Configuration.....	6
Network Topology.....	6
Storage Topology.....	6
Operating System and Application Software Configuration.....	6
The Device Request Dispatcher.....	6
The Cluster File System.....	8
The Cluster Alias Subsystem.....	10
The Network File System	10
Cluster Interconnect Characteristics	10
Memory Channel	11
Local Area Network (LAN)	11
Factors to Consider	12
Throughput and Latency.....	12
Cluster Member Transition.....	14
Apache Web Server.....	15
AIM VII Data	17
Oracle 9i RAC.....	18
Data Stress Workload.....	19
Data File Creation Workload.....	20
Data Upload Workload	21
Data Warehouse Workload	22
Factors to Consider Summary	23
Analyzing Interconnect Traffic	23
Case Study: An In-house Production Cluster.....	24

Summary	27
For more information.....	28

Overview

TruCluster Server Version 5.1A and higher supports Memory Channel or local area network (LAN) as the cluster interconnect. A cluster must have a dedicated cluster interconnect to which all cluster members are connected. This interconnect serves as a private communication channel between cluster members. This white paper provides you with guidelines to understand and choose the appropriate cluster interconnect technology for your needs. The option to choose a cluster interconnect offers you a flexible range of potential cluster configurations. However, as with any choice, the most appropriate solution depends upon a variety of factors. These factors include:

- Application characteristics
- Cluster availability requirements
- Cluster performance requirements
- Cost

Memory Channel does exhibit an advantage with regard to latency-sensitive, cluster-aware applications such as Oracle 9i RAC. However, Memory Channel does not enjoy the commodity pricing of LAN and is more expensive. For the majority of application mixes, a LAN interconnect solution is a viable alternative to a Memory Channel solution.

This white paper describes the characteristics of the cluster interconnect options, and provides information to allow you to make a more informed decision about which cluster interconnect technology is best for a particular cluster deployment. This white paper is divided into six sections:

- Cluster interconnect functions
- Configuration assumptions
- Cluster interconnect characteristics
- Factors to consider
- Analyzing cluster interconnect traffic
- Concluding summary

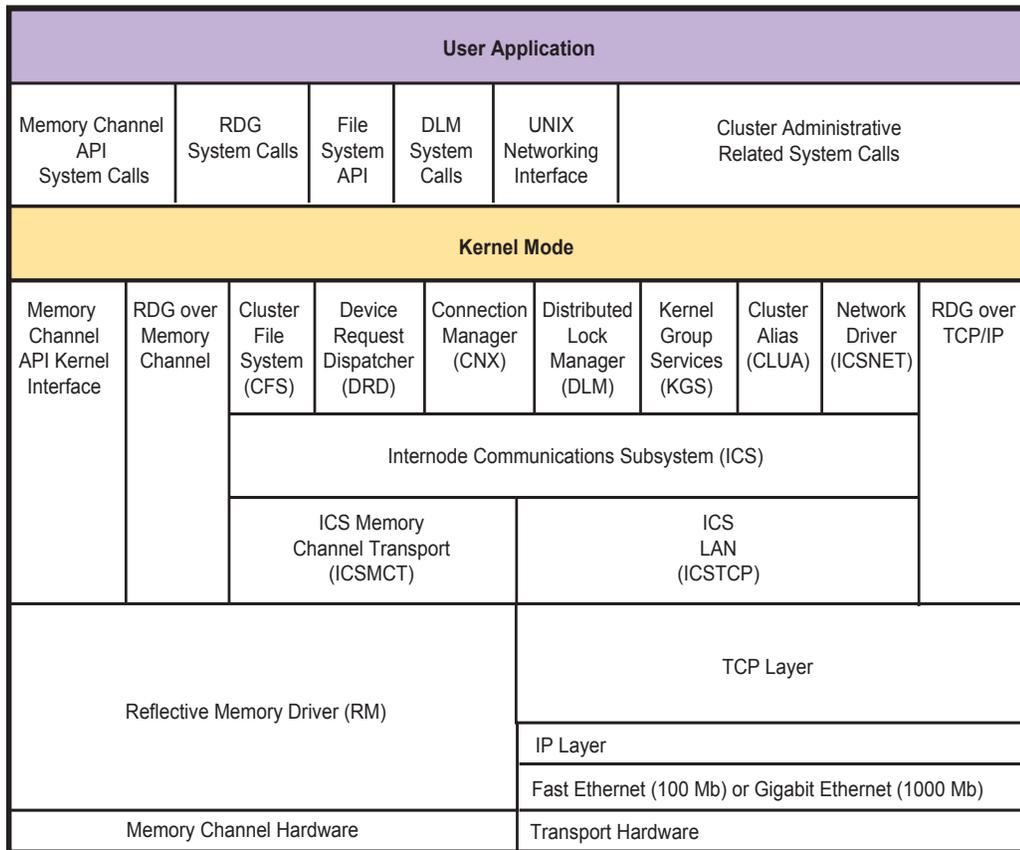
The Cluster Interconnect Functions

Understanding how the cluster interconnect is used will help determine which cluster interconnect technology is appropriate for a particular cluster deployment. The cluster interconnect is normally used for five high-level functions:

- Cluster Executive components
- Cluster I/O
- Cluster Networking
- Reliable Datagrams
- Application-specific traffic

This section describes each of these functions, and explains how each function impacts cluster performance. Figure 1 provides a conceptual view of the cluster interconnect architecture.

Figure 1. Cluster interconnect architecture



Cluster Executive Components

The Cluster Executive comprises the three cluster subsystems, based on the internode communications subsystem (ICS), that supply the core functionality supporting execution in a cluster environment:

- Quorum and membership (connection manager - CNX)
- Synchronization and locking (distributed lock manager - DLM)
- Subsystem coordination mechanisms (Kernel Group Services – KGS)

Message traffic across the cluster interconnect, attributed to the Cluster Executive components, increases during resource failover, cluster reconfiguration, and membership transition. The type of traffic created by these subsystems is latency-sensitive, and critical to the functional operation of the cluster. Latency-sensitive transmissions can be impacted by high-bandwidth traffic; therefore, it is important to minimize high-bandwidth communication across the cluster interconnect whenever possible.

The distributed lock manager (DLM) supports a user API. If an application makes heavy use of this API, the resultant cluster interconnect traffic can be significant. However, the DLM traffic created by the TruCluster Server software is minimal.

Cluster I/O

The TruCluster Server software presents a unified view of file systems and storage devices across all cluster members. Access to the file systems and storage devices can be either local or remote, depending upon the cluster configuration. All remote file system and storage device access must be transferred across the cluster interconnect. Remote access to the file systems and storage devices is provided by the cluster file system (CFS) and device request dispatcher (DRD).

The traffic created by remote access to file systems and storage devices can consume a significant amount of the cluster interconnect bandwidth, and thereby cause cluster-wide performance degradation. Localizing access to file system and storage devices bypasses the cluster interconnect for system I/O, which can improve cluster-wide application performance.

Cluster Networking

The TruCluster Server software provides flexible management of network services. These network services are provided in a highly available nature through the cluster alias subsystem (clua). A cluster alias is an IP address that makes some or all of the systems in a cluster look like a single system to Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) applications.

The bandwidth consumed by the cluster alias subsystem depends upon the degree to which it is being used. Under normal conditions this is limited to the tunneling of initial service requests. TCP service requests typically have minimal impact on the performance of the cluster interconnect; UDP service requests may have a larger impact. The network file system (NFS) is an example of a service which, when using UDP, may result in increased cluster interconnect traffic.

Improper network configuration and certain failure conditions may also result in the cluster alias subsystem forwarding packets over the cluster interconnect. This forwarding can significantly increase the traffic across the cluster interconnect, which can impact the overall performance of the cluster.

Reliable Datagrams

The Reliable Datagram Subsystem (RDG) presents a set of inter-process communication (IPC) interfaces. These interfaces can be used to communicate locally or remotely via the cluster interconnect. For remote IPC, RDG bypasses ICS and works directly with the cluster interconnect technology.

For example, in a Memory Channel cluster RDG manages its own message passing protocol over Memory Channel. In a LAN cluster, RDG layers its message-passing protocol over TCP/IP sockets directly to the cluster interconnect network interface card.

Application-Specific Traffic

The TruCluster Server software provides a private IP network that tunnels packets through the cluster interconnect. This private IP network uses the internode communication subsystem's network driver (icsnet). Applications can use this private IP network to perform member-to-member communication using the standard socket interface.

Many of the cluster-aware applications delivered as part of the Tru64 UNIX operating system use this private IP network. The overall impact of these applications on the cluster interconnect should be minimal. However, if a third-party application uses this network heavily, the IP traffic created can be significant and can cause clusterwide performance degradation.

Configuration Assumptions

There are a number of configuration considerations regarding hardware, operating system and application software that can minimize the impact of the cluster interconnect on performance. This section discusses the common configuration issues of which to be aware.

Hardware Configuration

Hardware configuration is an important component for overall application performance. Without carefully configured hardware, the cluster interconnect can wind up being used as an I/O tunnel. This is the expected and supported behavior for availability reasons in failure situations; however, it is not the ideal configuration for normal operations, and can lead to cluster-wide performance degradation. The two hardware configuration factors with the largest impact on cluster interconnect performance are network and storage topologies.

Network Topology

A TruCluster environment should be configured so that all members are connected to all available networks to which they need to provide application access. The cluster interconnect is not meant to be a general-purpose router, and configuring default routes through the cluster interconnect can cause excessive forwarding to occur. Excessive forwarding can degrade cluster-wide performance by unnecessarily consuming cluster interconnect bandwidth. In addition, it is important to note cluster-wide performance can also be impacted by unbalanced bandwidth between the external network and the private cluster interconnect network. To avoid cluster interconnect saturation, HP recommends the internal cluster interconnect network should have at least as much bandwidth available as the external network. For example, if the external network type is Gigabit Ethernet (1000 Mb/sec), the internal cluster interconnect network should also be Gigabit Ethernet.

Storage Topology

Storage topologies such as Fibre Channel provide the capabilities to easily connect all cluster members to a common storage pool. Members are assumed to have direct access to the storage devices to which they need to provide application access. If a member does not have direct access to a storage device, all I/O to that device will be tunneled through the cluster interconnect to a member that does have direct access to the storage device. This can significantly impact latency-sensitive cluster services and can cause cluster-wide performance degradation. For more information about hardware configuration, see the Related Documentation section.

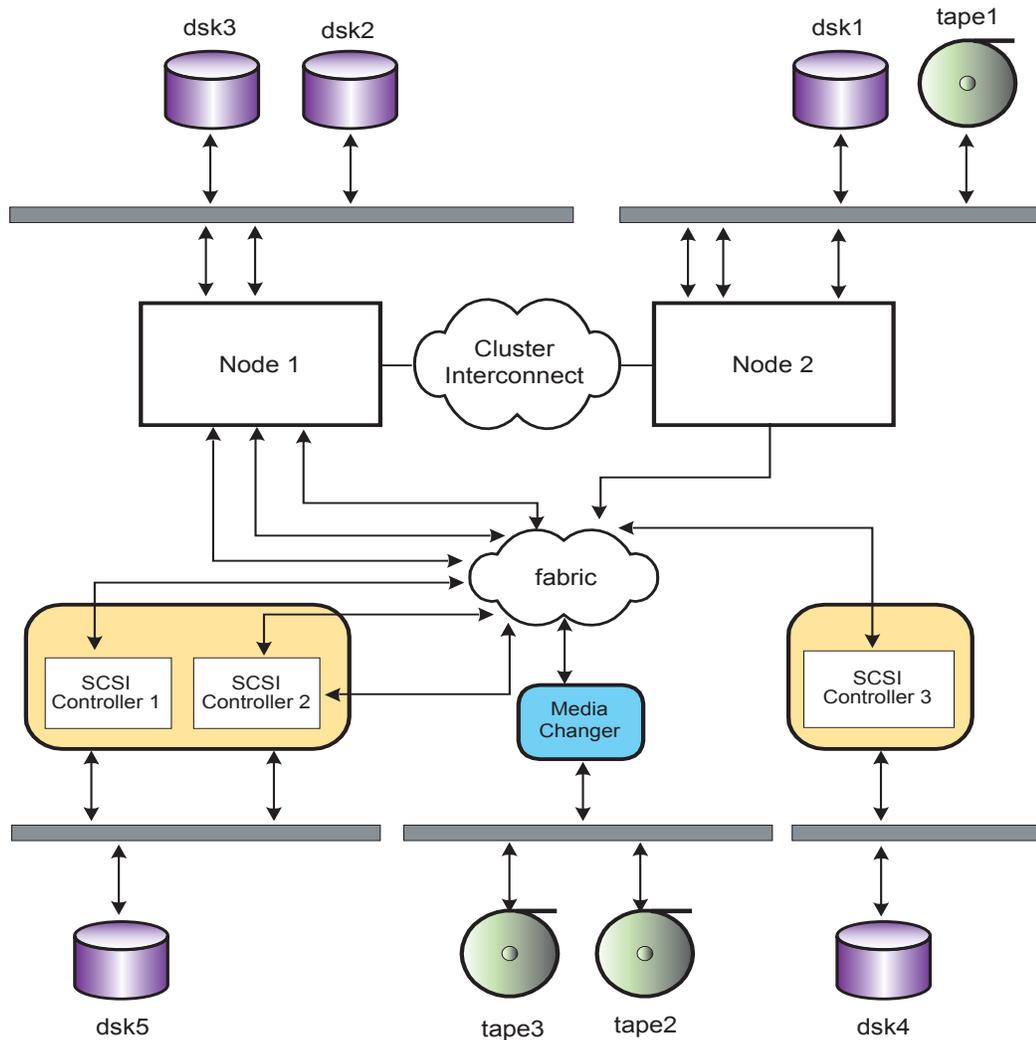
Operating System and Application Software Configuration

There are a number of operating system and application-specific configuration steps that you can take to maximize application performance. In general, try to use a principle of processing locality whenever possible. In this context, processing locality means keeping applications as close as possible to the system resources that they need to access. Some critical operating system components to keep in mind when considering application performance and processing locality are: the cluster file system (CFS), the device request dispatcher (DRD), the cluster alias subsystem (CLUA), and the Network file system (NFS).

The Device Request Dispatcher

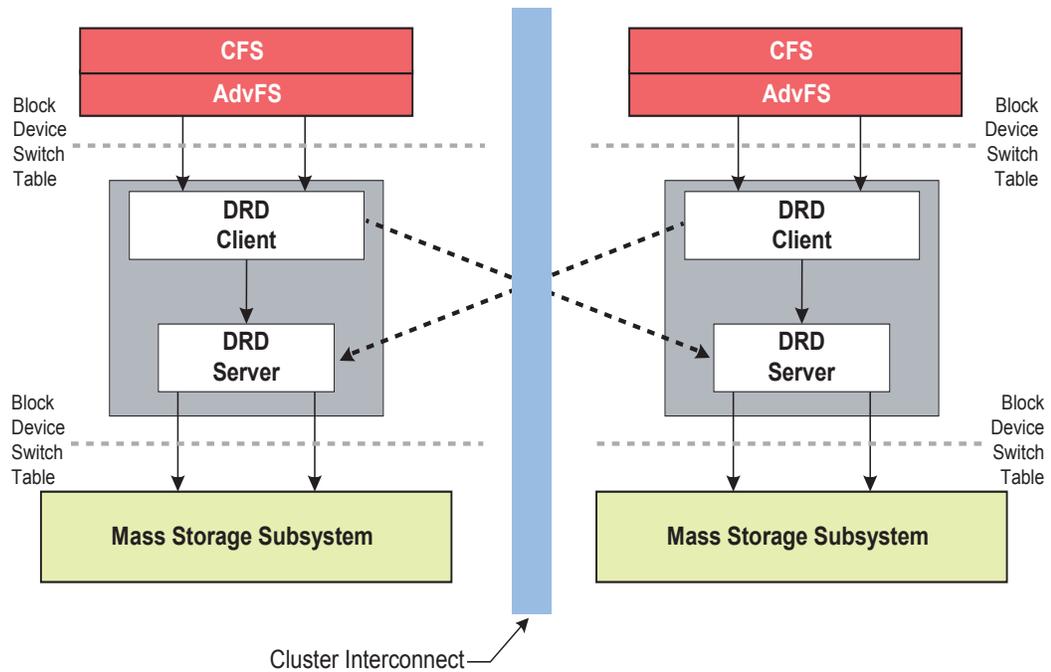
In a TruCluster Server, DRD provides transparent access to all mass-storage devices. DRD makes physical disk and tape storage available to all cluster members, regardless of where the storage is physically located in the cluster. A member does not need to be directly attached to the bus on which a device resides to access storage on that device. For example, in Figure 2 both Node1 and Node2 have access to all storage devices.

Figure 2. Unified device name access diagram



When necessary, DRD uses a client/server model. Devices in a cluster are either single-served or direct access I/O (DAIO) devices. A single-served device, such as a tape device, supports access from only a single member: the server of that device. For a single-served device, DRD will tunnel all client I/O over the cluster interconnect to the server. The server then performs I/O to the storage device. A DAIO device supports simultaneous access from multiple cluster members. DAIO devices on a shared bus are served by all cluster members on that bus. Whether or not a device is a DAIO device is determined by its attributes derived from SCSI inquiry commands. Figure 3 shows the relationship between DRD clients and DRD servers.

Figure 3. DRD client/server diagram



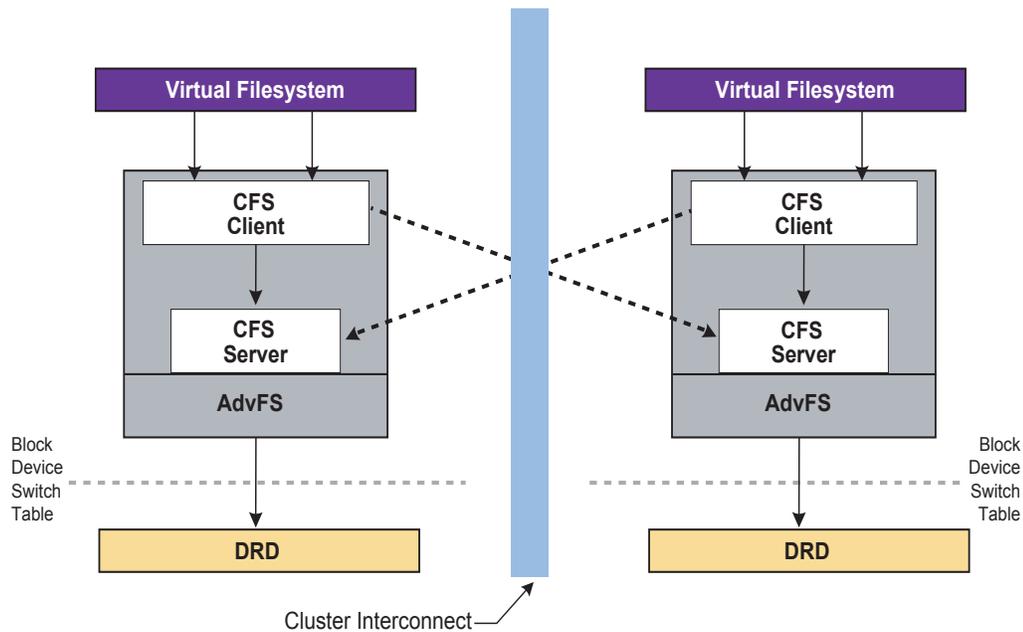
For more information regarding the device request dispatcher and mass-storage configuration, see the Related Documentation section.

The Cluster File System

In a TruCluster Server, CFS provides transparent access to file systems that are located anywhere on the cluster. Users and applications are presented a single-system image for file system access.

For most of its operations, CFS implements a client/server model, with each file system served by a particular cluster member. Each file system is subsequently presented to other cluster members, creating a one-to-many relationship between CFS servers and CFS clients. Figure 4 shows the client/server relationship between CFS servers and CFS clients.

Figure 4. CFS client/server diagram



For optimum performance, it is recommended that CFS file systems be designed with locality in mind. For example, assume that a cluster member is the only member in the cluster accessing a particular file system. Furthermore, assume that another member who is not accessing the file system is the server. All writes to the file system will be tunneled over the cluster interconnect. While this is necessary in terms of high availability, turning the cluster interconnect into an alternate data path for system I/O will adversely affect cluster performance.

A performance enhancement for AdvFS file systems is direct access cached reads. Direct access cached reads allow CFS clients to read directly from storage simultaneously on behalf of multiple cluster members. Direct access cached reads assume direct access to the storage. Without direct access to the storage CFS cannot take advantage of this feature.

In addition applications can bypass the CFS server for writes if the file is opened for direct I/O. When direct I/O is enabled for a file, data I/O is direct to the storage. In a cluster, this means any cluster member can perform concurrent writes to a particular file. That is, regardless of which member originates the I/O request, I/O to a file does not go through the cluster interconnect to the CFS server. If direct I/O is enabled for a file, the application with the file open must perform its own inter-node synchronization. Oracle 9i RAC, utilizing its cache fusion feature, is an example of application which takes advantage of the direct I/O feature. To utilize direct I/O, direct access to the storage is required.

In short, if a cluster member is running an application that performs extensive writes to a particular file system, that member should have direct access to the storage volumes associated with the file system, and, if the application does not utilize the direct I/O feature, should be the CFS server for the file system. Use `cfsmgr(8)` to co-locate the CFS server with an application as part of its cluster application availability (CAA) action script.

For more information regarding file system configuration, see the Related Documentation section.

The Cluster Alias Subsystem

A cluster alias is an IP address that makes some or all of the systems in a cluster look like a single system to Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) applications. Cluster aliases free clients from having to connect to specific cluster members for services, and can provide a higher degree of availability. Just as clients can request a variety of services from a single host, clients can request a variety of services from a cluster alias.

Some common uses for a cluster alias (such as `telnet`, `ftp`, and Web hosting) typically make only small communications demands on the cluster interconnect. For such applications, the amount of data sent to the cluster alias subsystem is generally far outweighed by the amount of data returned to clients from the cluster. Only the incoming data packets have the possibility of needing to traverse the cluster interconnect. All outgoing packets are sent directly to the external network client and bypass the cluster interconnect.

By default, the cluster interconnect is not configured to be an alternate route to reach external networks. Configuring the cluster alias subsystem to use the cluster interconnect as an alternate route to reach external networks is not recommended. This type of forwarding can cause cluster-wide performance degradation.

The cluster interconnect can also be used as a default route when all external network interfaces on a cluster member fail. In the event of complete external interface failure, on a cluster member, a default route through the cluster interconnect can be transparently set up by the cluster alias subsystem. To maintain optimal performance, minimize this type of forwarding whenever possible.

For Oracle installations, HP recommends that each Oracle instance allocate a separate cluster alias to avoid cross-node and interconnect traffic.

For more information regarding the configuration of the cluster alias subsystem, see the Related Documentation section.

The Network File System

The Network file system (NFS) is a commonly used operating system component, which can consume a significant amount of the cluster interconnect bandwidth. While reads from the served disks do not cause much interconnect traffic (only the read request itself potentially traverses the cluster interconnect), disk writes through NFS can result in interconnect traffic. In this case, the incoming data that might need to be delivered over the cluster interconnect is comprised of disk blocks.

TruCluster Server Version 5.1A introduced a feature that can reduce the NFS write traffic. For the purposes of NFS serving, you can assign alternate cluster aliases to subsets of cluster members. This allows a selected set of cluster members to be identified as the NFS servers, which reduces the average number of inbound packets that must be sent over the interconnect to reach that connection's server process. It has been shown that this type of configuration can significantly reduce traffic over the cluster interconnect.

For more information regarding the configuration of the network file system, see the Related Documentation section.

Cluster Interconnect Characteristics

This section describes the characteristics of the Memory Channel and the local area network (LAN) cluster interconnects.

Both Memory Channel and LAN cluster interconnect technologies provide a high-performance cluster solution. The Memory Channel hardware is only available through HP, and has few options; the LAN cluster interconnect technology provides a variety of price/performance options. Both Memory

Channel and LAN support up to eight cluster members, and support no single point of failure configurations.

While Memory Channel exhibits a performance advantage, for most application mixes a LAN cluster interconnect solution provides a viable, cost-effective alternative to Memory Channel.

Memory Channel

Memory Channel is a high-speed, low latency communications device. Unlike network message passing, communication is accomplished across a memory bus that is shared among the cluster members. It is not memory, but a bus or channel into the memory of another system. Memory Channel devices can map up to 512 MB of shared memory per Memory Channel adapter. Memory Channel was the sole cluster interconnect technology when the TruCluster Server product first shipped. It is a mature high performance technology and is available only from HP.

In a two-member cluster, the members' Memory Channel adapters can be cabled together directly, operating in virtual hub mode. When more than two members are connected, a Memory Channel hub is required. Memory Channel hubs support up to eight slots, thereby supporting clusters with as many as eight members. Memory Channel can be configured with a single rail or dual rail.

Only members participating in the same cluster can be connected to a given Memory Channel hub. This provides the private communication network for the cluster. If there are enough PCI slots each member can host two Memory Channel adapters configured as a failover pair. The adapters must be connected to separate hubs. This provides a cluster interconnect with no single point of failure (SPOF). The maximum separation distance between Memory Channel adapters is 6000 meters.

Within the cluster, the Internode Communication Subsystem (ICS) layers a message-passing algorithm onto shared memory space. While the actual delivered throughput varies depending on the circumstances, it is reasonable to view ICS on each member as capable of delivering approximately 72 MB/s to the cluster component services across the Memory Channel.

The Memory Channel Application Programming Interface (MC API) routines support an environment based on the inter-cluster member-shared memory attributes of the Memory Channel adapter. Applications can directly map this shared memory area through the MC API. The MC API allows the application to directly access this shared memory area and implement a shared-memory programming model within the application. It is important to note that this API is hardware-specific and only works over the Memory Channel.

Local Area Network (LAN)

TruCluster Server Version 5.1A and 5.1B both support a specialized LAN configuration as the cluster interconnect. A LAN cluster interconnect is composed of industry standard networking components. In a LAN cluster, TCP sockets are used as the intermember communication mechanism for cluster services between cluster members.

The LAN cluster interconnect provides a variety of price/performance options, allowing you to select the right configuration for your current or future needs. At the low end, a Fast Ethernet (100 Mb/sec) solution may be appropriate if the interconnect use is minimal. At the high end, a Gigabit Ethernet (1000 Mb/sec) solution offers performance comparable to Memory Channel for many configurations.

While the wire speed of each connection in a LAN cluster currently can be as high as 1 Gb, the actual throughput of the cluster interconnect will vary with different types of loads. Loads consisting of large transfers result in higher total data throughput while smaller transfers may limit the total data throughput.

In addition to throughput, another performance characteristic to be aware of is latency. Latency in this context is the roundtrip time associated with sending messages over the cluster interconnect. The

latencies associated with the LAN cluster interconnect are larger than that of the Memory Channel cluster interconnect. While this may be an issue for some configurations, for most applications LAN presents a viable alternative to Memory Channel.

The network used as the cluster interconnect must be private to the cluster members. More specifically, the adapter(s) designated on each system for cluster interconnect use must be connected only to adapters similarly designated on other cluster members. All nodes in the cluster must be able to communicate directly to all the other members' interconnect adapters through this private network. (For example, a configuration that requires TCP/IP routing of the interconnect communications is not supported). Within this private network, the TCP/IP hop count limit between any two member nodes is three. That is, no path must require the packets to go through more than two switches.

The network communication characteristics of the various adapters must match. For example, you cannot mix 100 Mbps adapters with 1000 Mbps adapters. Switches and hubs must also match the adapters that are connected to them.

In a typical cluster configuration, Fast Ethernet (100 Mb/sec) or Gigabit Ethernet (1000 Mb/sec) network interface cards are connected directly for a two member configuration; network switches for configurations are used in configurations with three or more members. The adapters used as the cluster interconnect can be part of a NetRAIN set. This allows multiple physical adapters to be tied together in configurations more resilient to the effects of a failed adapter or an unplugged wire. (The cluster configuration documentation describes how to configure fully redundant LAN hardware). The use of NetRAIN sets does not have to be consistent; some nodes can use a single adapter as the cluster interconnect while others use NetRAIN sets for redundancy.

Factors to Consider

When considering which cluster interconnect technology best suits a given cluster deployment, consider these factors:

- Cluster performance requirements
- Cluster availability requirements
- Application characteristics

This section provides data that characterizes the raw capabilities of Gigabit Ethernet (1000 Mb/sec) and Memory Channel cluster interconnect technologies, outlines cluster member transition times, and provides three application examples.

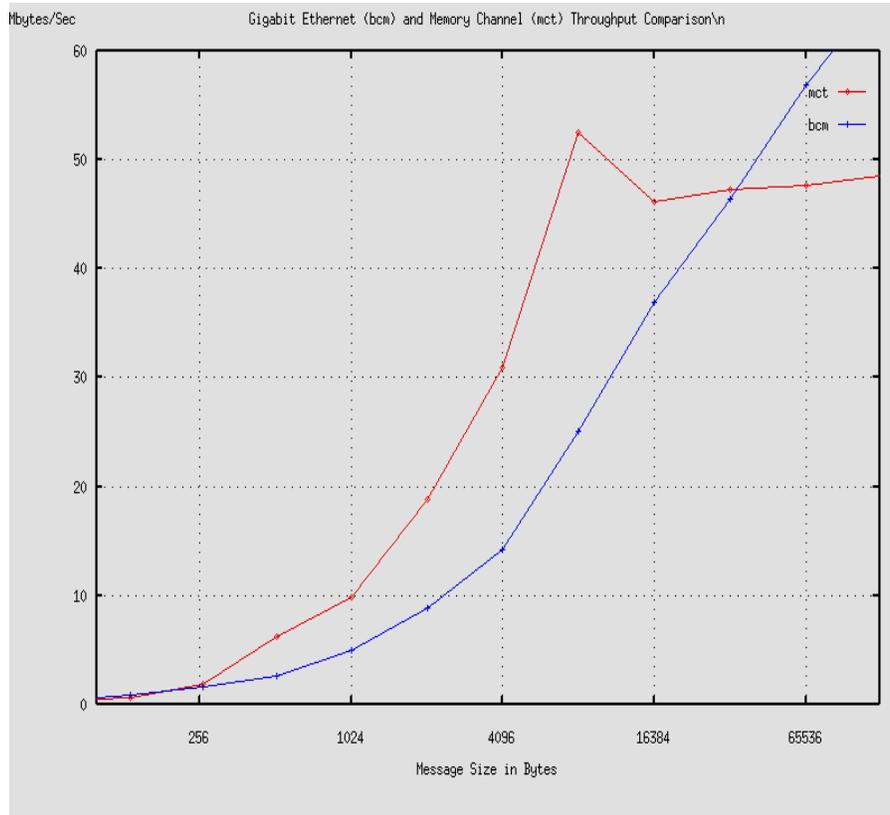
Throughput and Latency

This section presents the raw throughput and latency differences between Memory Channel and Gigabit Ethernet. The data was generated using in-house tools and a custom instrumented kernel. The configuration consists of two ES45 AlphaServers running TruCluster Server Version 5.1B. We use a DEGXA-SA Gigabit Ethernet network interface card.

For historical reasons, data transferred over the cluster interconnect can be fragmented into multiple transfers by the cluster subsystems. For this reason we examine only data ranging between 0 and 128 KB.

Throughput represents the amount of message data that can be passed across the cluster interconnect in a given period of time. The data represents direct measurements of the roundtrip message time of synchronous ICS messages sent from a single ICS client to a single ICS server. Figure 5 shows the median throughput for messages ranging in size from 64 bytes to 128 KB.

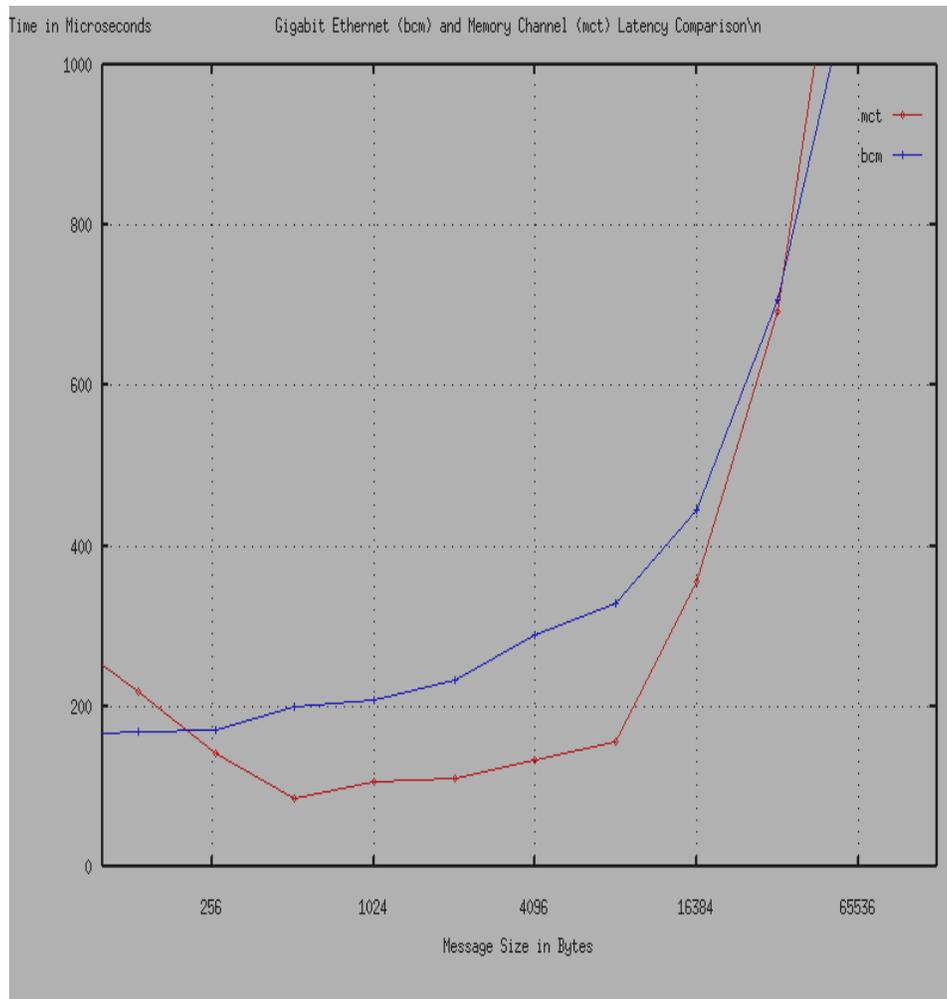
Figure 5. Differences in throughput between Gigabit Ethernet and Memory Channel



For data sizes above 32 KB, the LAN throughput exceeds that of Memory Channel. For data sizes below 32 KB, Memory Channel throughput exceeds that of LAN.

Latency represents the roundtrip time required for a message to be sent and received across the cluster interconnect. The same configuration as that used in the throughput example was used. Direct measurement of the roundtrip message time was made synchronously from a single ICS client to a single ICS server. Figure 6 shows the median latency for messages ranging in size from 64 bytes to 128 KB.

Figure 6. Gigabit Ethernet and Memory Channel latency comparison



For data sizes 32 KB and smaller, Memory Channel provides lower latency. For most applications this difference will not significantly impact application performance or availability. For an application where the difference in latencies in the hundreds of microseconds range makes a difference, Memory Channel clearly has the advantage.

While throughput and latency are major factors to consider, for a large number of application mixes it is unlikely that the differences between Memory Channel and Gigabit Ethernet will significantly impact the overall application performance.

Cluster Member Transition

Cluster member transition occurs when members leave or join the cluster. A cluster member may expectedly leave the cluster as a result of a user entering a shutdown, reboot, or halt command. Additionally, a cluster member may unexpectedly leave the cluster as the result of a panic, machine check, equipment failure, or loss of power.

During cluster member transition, a sequence of events takes place that is required for the cluster to maintain operation. These events cause applications, such as NFS and other network services, to be unavailable for a period of time.

Cluster member transition occurs in two independent phases. The first phase is transition detection; the second is transition action. For operations such as a cluster member joining or expectedly leaving, there is minimal difference between a Memory Channel cluster and a LAN cluster. In the case of an unexpected cluster member transition, the Memory Channel cluster is faster in the detection phase of the transition. Memory Channel has circuitry built into it that almost immediately detects the loss of a member; a LAN cluster relies on TCP keep alive timeouts to detect this event. This results in a longer overall failover time for a LAN cluster.

To address this issue, the Tru64 UNIX group published the *Tuning the Cluster Transition Time Best Practices* document. This Best Practice describes how to tune a TruCluster Server to maximize client availability during cluster member transitions. You can access this Best Practice from:

http://h30097.www3.hp.com/docs/best_practices/BP_CLU/TITLE.HTM

The Best Practices document addresses the differences in cluster member transition between Memory Channel and LAN, but it does not address all aspects of node transition and application failover. For more information on these topics see the previously mentioned Best Practices document and the *Cluster Highly Available Applications* manual. See the Related Documentation section for links to these documents and others.

Apache Web Server

This example demonstrates that some types of multi-instance applications consume minimal cluster interconnect bandwidth and perform almost as well on either of the available cluster interconnect technologies.

The Apache Benchmark is a tool for benchmarking the performance of the Apache HyperText Transfer Protocol (HTTP) server. It does this by giving you an indication of how many requests per second your Apache installation can serve. The Apache Benchmark tool was used to measure the difference between the two cluster interconnects.

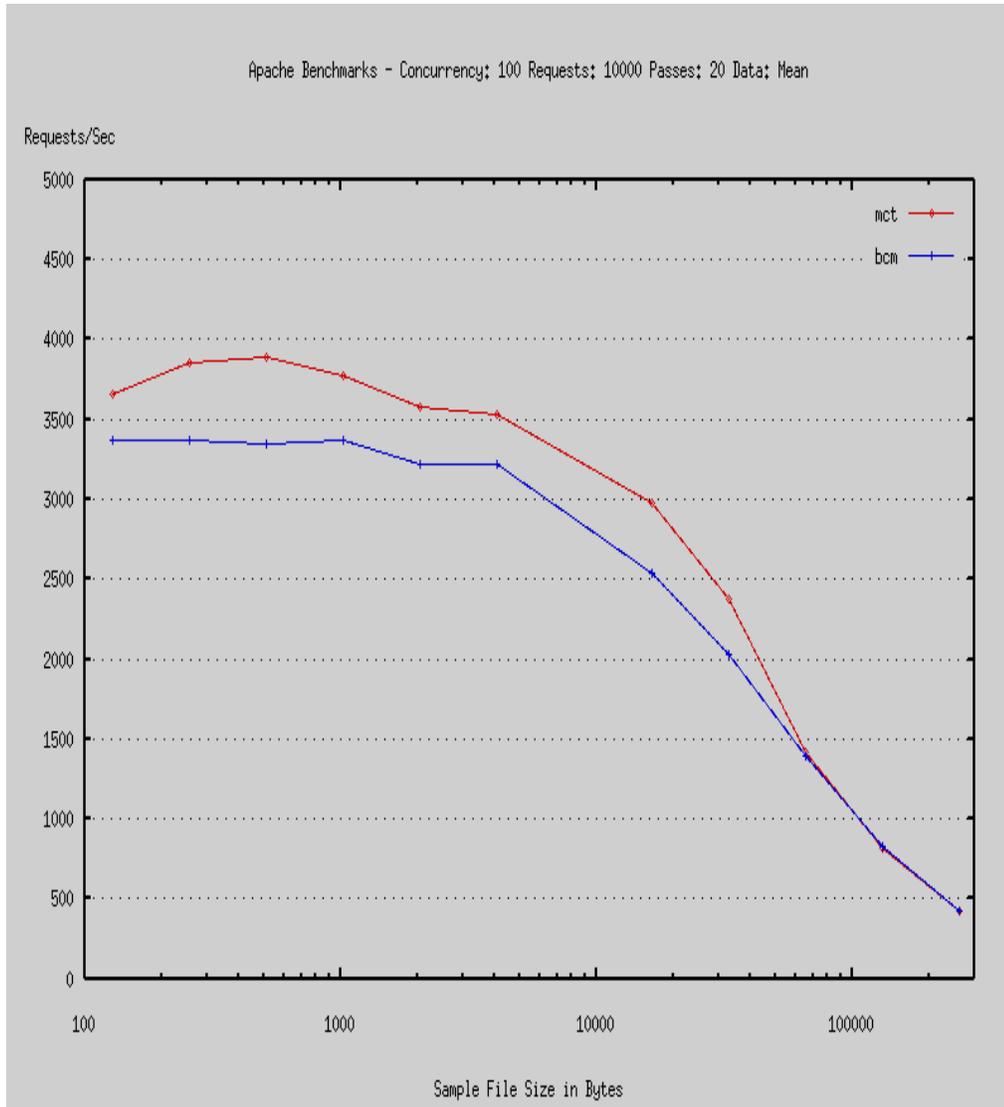
For the examples that follow, a two-member ES45 cluster was configured with Tru64 UNIX Version 5.1B. Apache Version 1.3.27 was installed and configured on the cluster. A standalone ES45 was also installed with Tru64 UNIX Version 5.1B. The cluster and the standalone system were then networked using a private gigabit network.

A copy of the Apache Benchmark program was placed on the client systems and a set of HTTP documents were made available on the server. These files ranged in size from 128 bytes to 256 KB.

For each file size the Apache Benchmark program was run 20 times. It was first configured with Memory Channel as the cluster interconnect, and then configured to use a DEGXA-SA Gigabit Ethernet cluster interconnect.

Figure 7 shows the number of requests per second achieved using the Apache Benchmark tools for 10,000 requests with a concurrency of 100 processes for each file size.

Figure 7. Apache benchmarks (number of requests per second)



As shown in Figure 7, Memory Channel and LAN are very close in request per second performance with Memory Channel achieving a maximum of 15 percent better performance for some file sizes, but under 10 percent for most file sizes.

Figure 8 shows the number of kilobytes per second transfers that were achieved using the Apache Benchmark tools for 10,000 requests with a concurrency of 100 processes for each file's sizes.

Figure 8. Apache benchmarks (kilobytes per second)

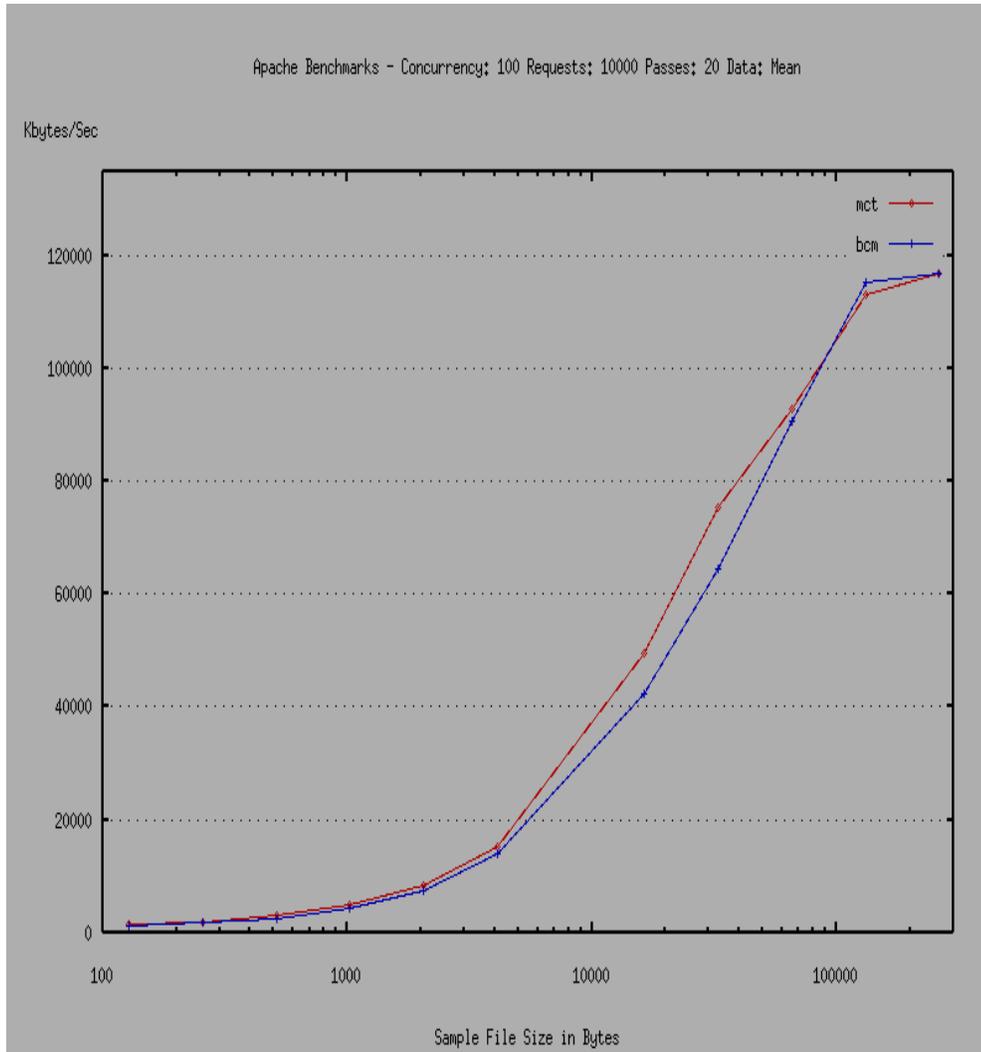


Figure 8 shows that Memory Channel and LAN are very close in throughput performance, with Memory Channel achieving a maximum of 15 percent better performance for some file sizes but under 10 percent for most file sizes.

AIM VII Data

This example demonstrates that an application mix requiring minimal cluster interconnect bandwidth will perform almost as well on any of the cluster interconnect technologies available.

The AIM Multi-user Benchmark Suite VII, developed by AIM Technology, is a multithreaded system exerciser designed to test system performance. The suite can be run on a range of systems, including compute servers, file servers, and multi-user systems. The suite includes four standard application mixes that measure the performance of the entire system:

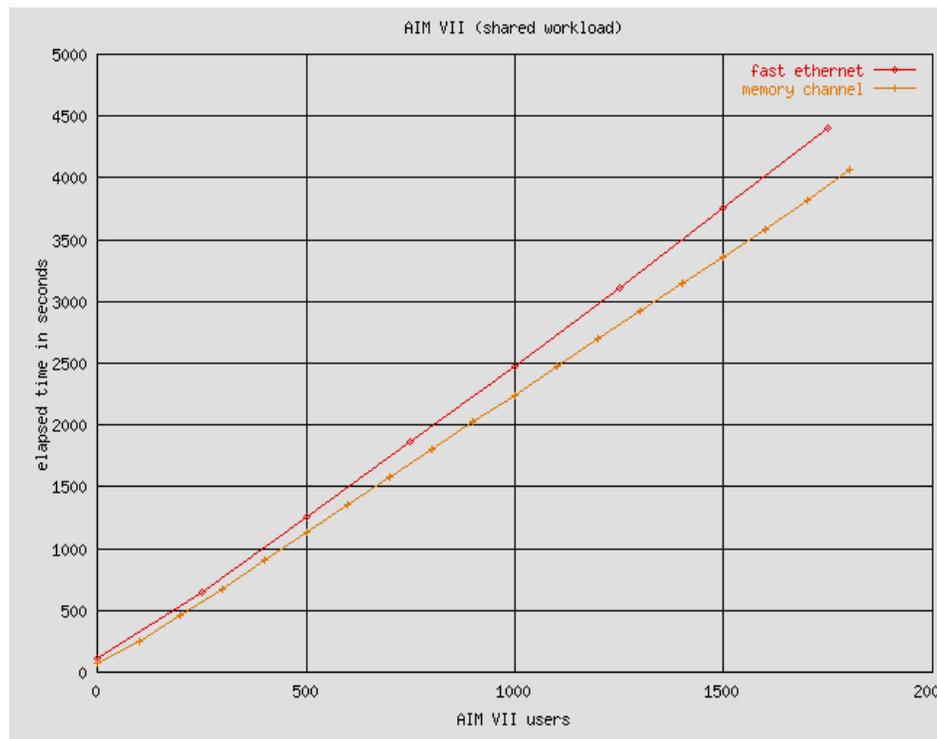
- Multi-user/Shared System
- Compute Server

- Large Database
- File Server

The Multi-user/Shared System suite was used to measure the difference between the two cluster interconnects. This suite includes office automation, word processing, spreadsheet, electronic mail, database, payroll, and data processing applications. It simulates a large number of concurrent users using small amounts of memory while performing heavy tasking, medium integer, light floating point, medium file I/O, shell routines, and string routines.

Figure 9 shows the difference in the elapsed time required to execute the benchmark suite. Fast Ethernet (100 Mb/sec) and Memory Channel were used for this example.

Figure 9. AIM VII elapsed execution time



As shown in Figure 9, the difference between Fast Ethernet (100 Mb/sec) and Memory Channel are within 10 percent. We provide this example to show that there are application mixes where a high-performance cluster interconnect yields little advantage. A two-member cold failover cluster is an example of a deployment where the cluster interconnect latency and bandwidth have little impact on cluster-wide application performance.

Oracle 9i RAC

Oracle 9i RAC is an application with features that exploit the distributed nature of a TruCluster Server, and can fully utilize its resources. Oracle 9i RAC's Cache Fusion architecture exploits the low-latency, high bandwidth cluster interconnect technologies to maintain cluster-wide database cache coherency. This eliminates the need to perform disk I/O for internode synchronization. In a TruCluster Server environment, Oracle 9i RAC utilizes the inter-process communication interfaces of the Reliable Datagram Subsystem (RDG).

In addition, Oracle 9i RAC automatically takes advantage of CFS/AdvFS direct I/O. When direct I/O is enabled for a file, data I/O is direct to the storage; the system software does no data caching for the file at the file-system level. In a cluster, this arrangement supports concurrent direct I/O on the file from any member in the cluster. That is, regardless of which member originates the I/O request, I/O to a file does not go through the cluster interconnect to the CFS server.

For more information about RDG and AdvFS direct I/O, see the Related Documentation section.

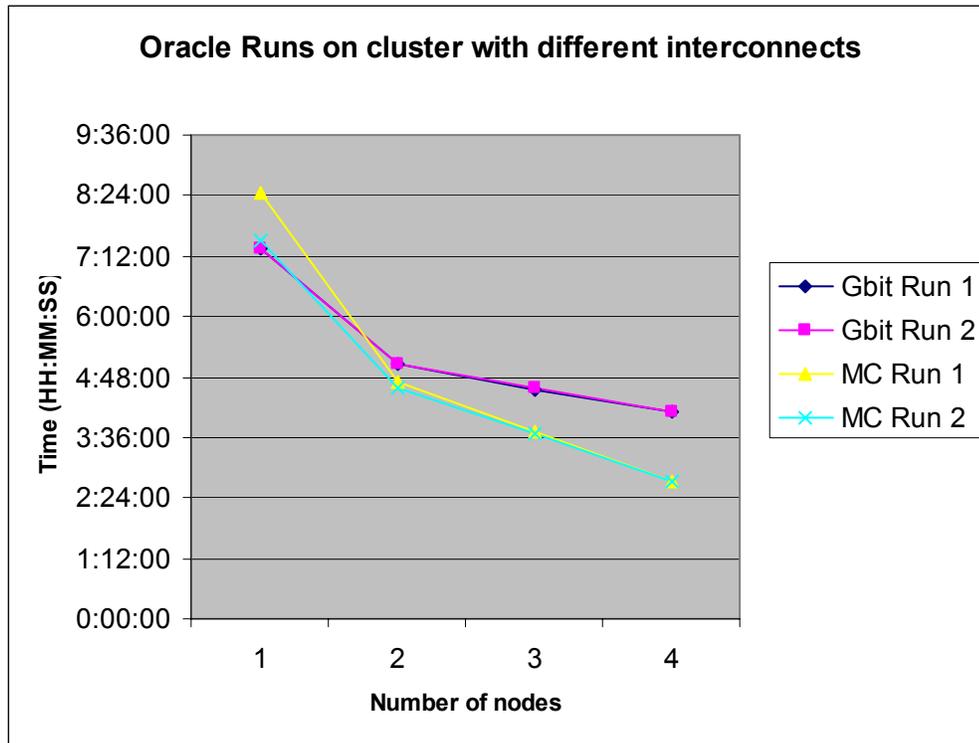
In this section we present performance characteristics of four different Oracle workloads: data stress, data upload, file create, and data warehouse. Each workload differs in the way it stresses Oracle, and thereby differs in how the cluster interconnect impacts overall application performance. Each experiment was conducted using four symmetrically configured ES47 servers running TruCluster Server Version 5.1B. All systems were connected via a Fibre Channel fabric to all storage devices.

Data Stress Workload

In this workload, Oracle 9i RAC reads in two large database tables, scrubs them using Oracle stored procedures, and merges them writing out a third table. In a TruCluster Server environment Oracle runs in multi-instance mode. This means that an instance of Oracle runs on each node in the cluster. During the multi-instance execution, this benchmark heavily utilizes the Cache Fusion Architecture, which transfers large amounts of data over the cluster interconnect via RDG.

Figure 10 shows that the Memory Channel cluster exhibits an advantage in both bandwidth and latency over Gigabit Ethernet, resulting in an increase in overall application performance. As the number of nodes in the cluster increases, so does the performance disparity.

Figure 10. Data stress workload

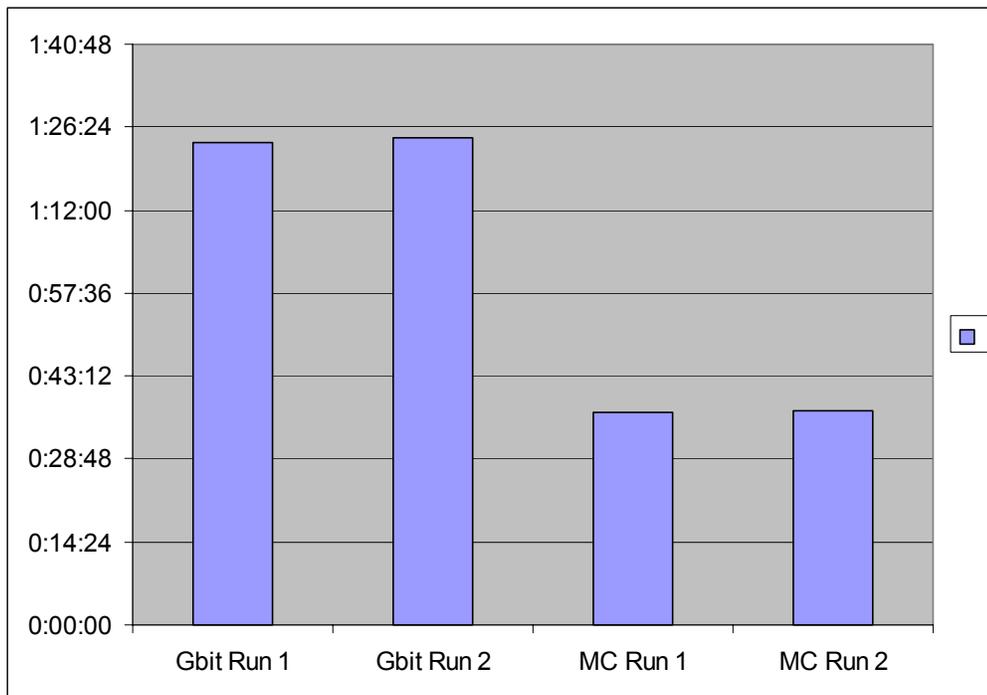


Data File Creation Workload

For this workload, Oracle data files were created on 30 file systems from one node while all the CFS servers of these file systems were distributed across the other three nodes in the cluster (ten each). Oracle hands over file create requests to the operating system (Tru64 UNIX). Since file create involves meta data changes, Tru64 UNIX uses the cluster interconnect to create the files on the file system through the node serving the file system. The performance of this workload is heavily dependent upon the cluster interconnect.

In Figure 11 note that the Gigabit Ethernet cluster took approximately twice the time of Memory Channel to complete this test.

Figure 11. Data file creation workload

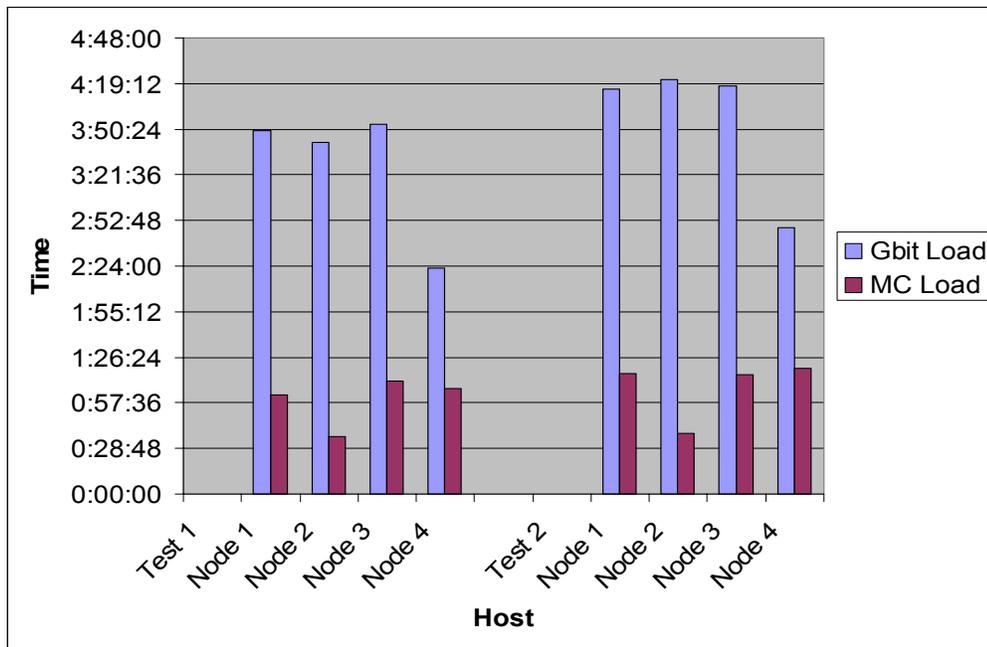


Data Upload Workload

In this test, a C program generates data into named pipes and Oracle SQL loader loads the data from the pipes into the database. We used all four nodes in the cluster to do this activity.

As Figure 12 shows, Gigabit Ethernet took three times as long as Memory Channel.

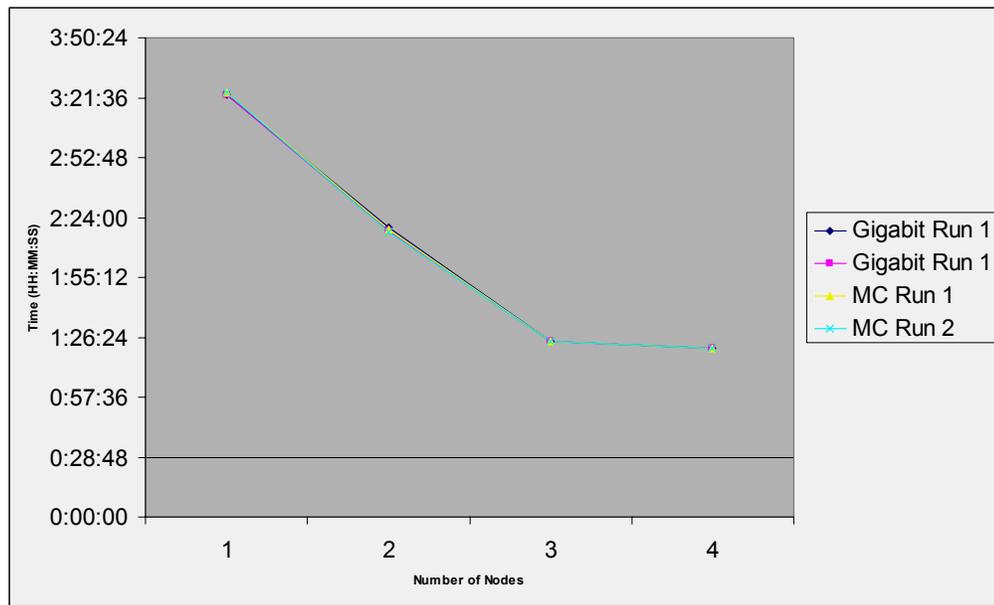
Figure 12. Data upload workload



Data Warehouse Workload

For this workload, an Oracle full table scan (read) is performed, ramping up from one node to four nodes. Due to Oracle's cache fusion and TruCluster Server direct I/O, this type of workload lends itself to parallelization and overall scan time is reduced by adding more nodes. Figure 13 shows the performance between Memory Channel and Gigabit Ethernet for this type of workload. Since cache fusion data exchange between the nodes is very limited in this workload, there is no difference between Gigabit Ethernet and Memory Channel.

Figure 13. Data warehouse workload



From the different types of workloads and their results, we can conclude that the type of workload determines the impact of the cluster interconnect on the overall application performance. Any workload that is heavily dependent on CFS has the largest performance impact. Further, workloads that perform excessive writes to storage require Oracle to perform a lot of inter-node synchronization. This type of synchronization traffic is latency-sensitive and the cluster interconnect technology can impact overall application performance.

In a distributed transaction oriented database executing high volumes of both reads and writes, the low-latency, high bandwidth Memory Channel cluster interconnect technology exhibits a clear advantage in overall application performance. For data warehouse workloads (reads only), either Memory Channel or Gigabit Ethernet provides a high-performance solution.

Factors to Consider Summary

A TruCluster Server environment is a generalized distributed computing platform designed to handle many types of application mixes. Both Memory Channel and LAN cluster interconnect technologies can be deployed to provide high-performance solutions.

The Memory Channel hardware was designed specifically to be a cluster interconnect, and as such presents some performance advantages over LAN; however, a LAN cluster provides a viable, cost-effective alternative to Memory Channel for a large number of TruCluster Server deployments.

Analyzing Interconnect Traffic

This white paper has discussed the importance of resource configuration within a cluster to obtain optimal performance. You can use `cfsstat(8)`, a command available with the TruCluster Server product, to measure and analyze the cluster interconnect usage of a give cluster deployment. You can use the statistics provided by `cfsstat(8)` to analyze both the cluster file system (CFS) and the internode communications subsystem (ICS). In this paper we limit our discussion about `cfsstat(8)` to ICS-related information.

ICS provides communication primitives to allow various TruCluster Server components to communicate through the cluster interconnect. ICS has a total of 16 communication channels with which messages can be transmitted across the cluster interconnect. You can use `cfsstat(8)` to retrieve statistics from the kernel about the traffic on each ICS channel.

`cfsstat(8)` statistics are broadly classified into server side and client side statistics. This command provides you with a lot of flexibility in how to display the statistics. You can view these statistics in count mode, bytes per second mode, operations per second mode, and several other modes. Typical `cfsstat(8)` statistics are on a per-channel basis and include: number of messages sent and received, size of the messages, and total data throughput. You can use `cfsstat(8)` to obtain a snapshot of the statistics at any given instant, or you can use it to continuously log the ICS activity reporting statistics at periodic intervals.

Collecting a set of statistics and analyzing them yields some interesting characteristics about the cluster's usage of the cluster interconnect. Given a target installation, you can use `cfsstat(8)` to optimize the configuration for better performance. For more information about how to use `cfsstat(8)`, see the `cfsstat(8)` reference page.

Case Study: An In-house Production Cluster

The Configuration Assumptions section discussed the importance of resource locality. Using `cfsstat(8)`, you can analyze resource locality issues by studying the amount of load being transmitted across the cluster interconnect. This section presents a case study of an in-house production cluster. The production cluster consists of a four-member Memory Channel cluster running TruCluster Server Version 5.1B. The goal of the experiment was to validate the use of `cfsstat(8)` based analysis to understand the cluster interconnect usage pattern of real-world applications. This TruCluster environment is being used primarily as an NFS server. Most users use this system for small file access, e-mail, and Web browsing.

The experiment involved collecting ICS channel usage statistics on an in-house production server. The logs were collected at periodic intervals over a period of two days. Figure 14 shows a sample session log on how the `cfsstat(8)` command was used to collect statistics on a live system. In Figure 14, the first column of the table is a list of packet sizes. The rest of the table lists how many packets were sent in each channel for any given size.

Figure 14. Sample cfsstat output

```

Window Edit Options Help
#cfsstat icschanhisttot
Total:
      BOOT  PRIO  PRI1  CFS  NET  CLSM  DRD  TUNL  CLUA  KEVM  CFSR  RBLD  KCH  DLM  TEST
0      0      0      0      0      0      0      0      0      0      0      0      0      0      0
1      0      0      0      0      0      0      0      0      0      0      0      0      0      0
2      0      0      0      0      0      0      0      0      0      0      0      0      0      0
4      0      183m  32k      0      22k      2      0      0      5m      224      0      0      0      0
8      39      0      141      0      0      0      480      0      0      0      0      0      0      0
16     9      0      116      0      0      0      5k      0      0      224      0      0      0      0
32     5      3m      150      375m  75k      0      4k      0      0      0      0      593k      0      0
64     4      4m      2      118m      1m      0      360      78m      0      0      0      979k      0      0
128    1      180m  238k  402m      4m      0      2k      126m  14m      2k      0      178k      0      0
256   251      55m      7k      1g      41k      0      99      2m      0      274k      0      91k      63m      0
512    0      127k      1k      7m      224      2      2k      954k      0      70      0      885k      0      0
1k     0      223k      0      2m      3k      0      286      1m      0      0      0      63k      0      0
2k     0      333k      0      2m      44      0      1k      99k      7k      0      0      2k      0      0
4k     0      642k      0      1m      28      0      0      217k      0      0      0      63k      0      0
8k     0      610k      1      23m      0      0      0      1m      0      0      0      0      0      0
16k    0      94k      0      1m      0      0      0      46k      0      0      0      17      0      0
32k    0      71k      0      845k      0      0      0      271k      0      0      0      0      0      0
64k    0      1m      0      3m      0      0      0      0      0      0      0      0      0      0
128k   0      0      0      0      0      0      0      0      0      0      0      0      0      0
256k   0      0      0      0      0      0      0      0      0      0      0      0      0      0
512k   0      0      0      0      0      0      0      0      0      0      0      0      0      0
1m     0      0      0      0      0      0      0      0      0      0      0      0      0      0
2m     0      0      0      0      0      0      0      0      0      0      0      0      0      0
4m     0      0      0      0      0      0      0      0      0      0      0      0      0      0
8m     0      0      0      0      0      0      0      0      0      0      0      0      0      0
16m    0      0      0      0      0      0      0      0      0      0      0      0      0      0
more   0      0      0      0      0      0      0      0      0      0      0      0      0      0
log.1: END

```

The data collected from Figure 14 was further processed to yield the results in Figures 15 and 16.

Figure 15. Packet size distribution

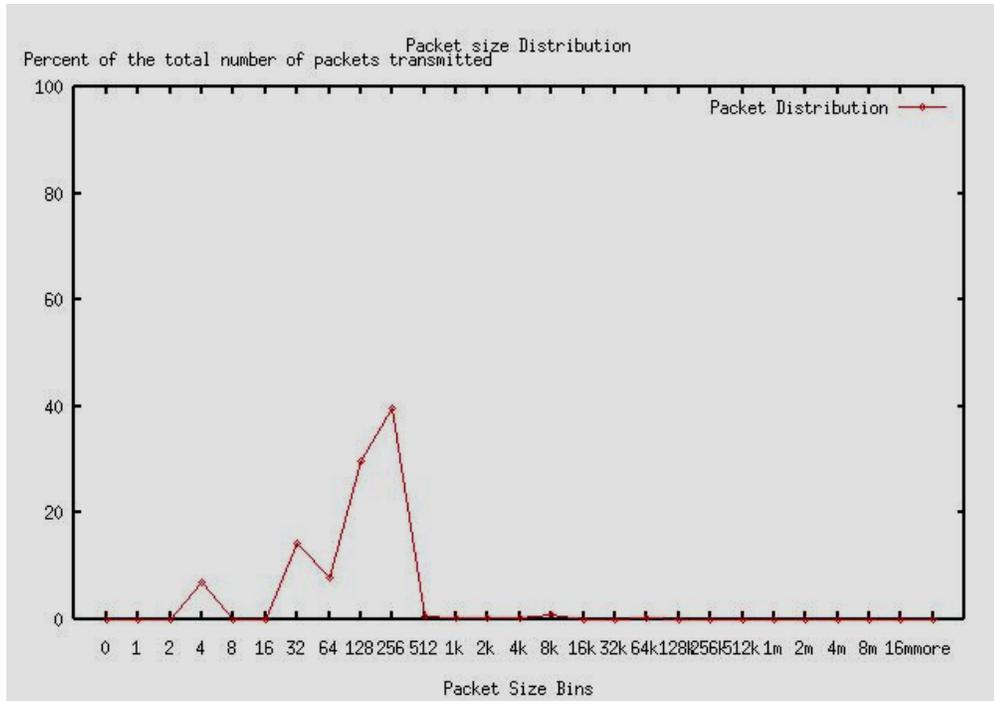


Figure 15 shows the percentage of the total number of packets (y axis) plotted against the size of the packets (x axis). For example, roughly 41 percent of all the packets transmitted were 256 bytes in size.

Figure 16. Channel traffic distribution

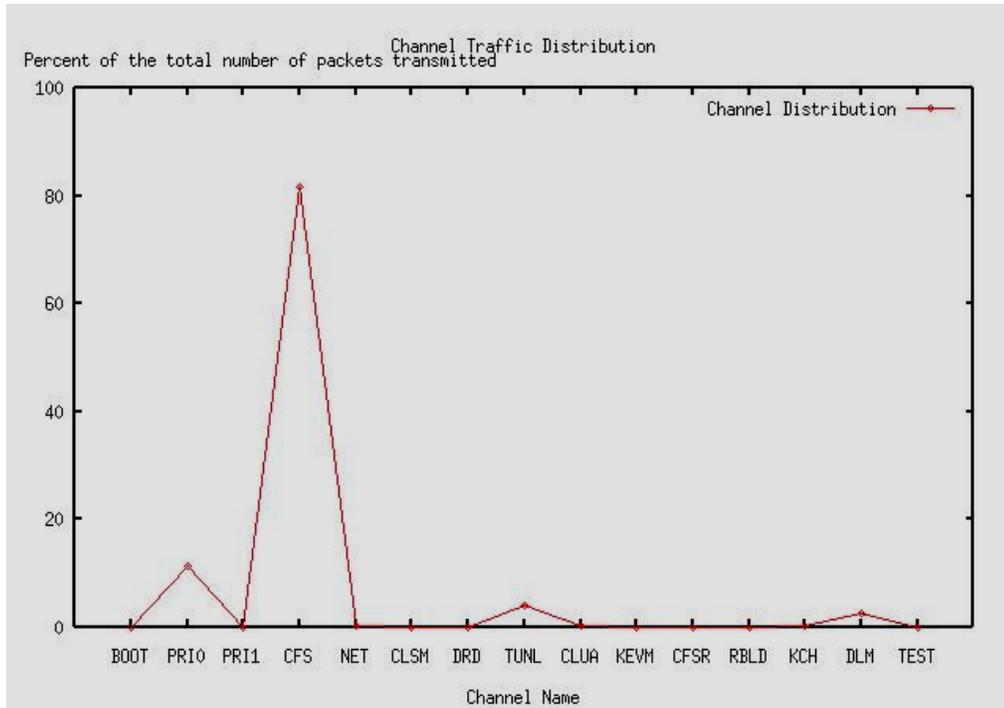


Figure 16 gives you an idea of how the channels are used in a typical NFS server scenario. It is clear from Figure 16 that most of the traffic is concentrated on the CFS channel. Combining our observations from both figures, it appears that in this scenario the cluster serves a load consisting of predominantly small-sized messages utilizing the CFS channel. This is what we would expect since most of the users use the production machines for mail, shared tool directories, and small file access.

It is clear from the discussion that `cfsstat(8)` is a useful tool in understanding and analyzing the usage of the cluster interconnect. However, it is noteworthy to mention that `cfsstat(8)` cannot be used in all scenarios. For example, Oracle 9i RAC uses RDG to perform internode communication. RDG bypasses the ICS layer through which `cfsstat(8)` collects statistics. Therefore, if an application bypasses ICS to perform internode communication, `cfsstat(8)` cannot be used to evaluate cluster interconnect traffic.

Summary

Deciding which interconnect is appropriate involves more than just considering the raw performance characteristics of the cluster interconnect technologies. To make an informed decision about the most appropriate cluster interconnect technology, you must consider the roles of the cluster interconnect, the performance requirements of the application mix, the price to performance ratio, and the budget.

Along with configuring the storage so that each system has direct connectivity, the key component in the overall picture is the type of applications that will be run on the cluster, and what demands those applications place on the cluster interconnect.

In general, high-performance, time-sensitive parallel applications that are cluster-aware, and make use of the distributed processing capabilities of a TruCluster Server, operate best with the high bandwidth and low latency characteristics of Memory Channel. However, a LAN cluster can provide a cost-effective alternative to Memory Channel for a large number of TruCluster application mixes.

For all practical purposes, most low to mid-range TruCluster Servers, depending upon the application mix, would execute equally well with either Memory Channel or LAN. For example, simpler style application failover clusters will often do well with an Ethernet-based LAN interconnect.

For more information

For general information about TruCluster Server Version 5.1A and 5.1B software configuration, TruCluster Server administration, and TruCluster Server hardware configuration, see the TruCluster Server documentation sets at the following URLs:

TruCluster Server Version 5.1A:

http://h30097.www3.hp.com/docs/pub_page/cluster51B_list.html

TruCluster Server Version 5.1B:

http://h30097.www3.hp.com/docs/pub_page/cluster51B_list.html

For updates to this documentation, see the following Tru64 UNIX technical updates:

Version 5.1A:

<http://h30097.www3.hp.com/docs/updates/TCR51A/TITLE.HTM>

Version 5.1B:

<http://h30097.www3.hp.com/docs/updates/TCR51B/TITLE.HTM>

For information on tuning the Cluster transition time, see the following Best Practice document:

http://h30097.www3.hp.com/docs/best_practices/BP_CLU/TITLE.HTM

For information on installing and configuring Fibre Channel storage in a TruCluster environment see the following Best Practice document:

http://h30097.www3.hp.com/docs/best_practices/BP_TCRFC/TITLE.HTM

For information on configuring a highly available NFS-mounted file system, see the following Best Practice document:

http://h30097.www3.hp.com/docs/best_practices/BP_NFSCDSL/TITLE.HTM

For information about tuning a Gigabit Ethernet configuration for optimal performance, see the following Best Practice document:

http://h30097.www3.hp.com/docs/best_practices/BP_GIGABIT/TITLE.HTM

TruCluster Server Version 5.1B *Hardware Configuration* manual:

http://h30097.www3.hp.com/docs/base_doc/DOCUMENTATION/V51B_HTML/ARHGWETE/TITLE.HTM

TruCluster Server Version 5.1B *Technical Overview* manual:

http://h30097.www3.hp.com/docs/base_doc/DOCUMENTATION/V51B_HTML/ARHGVETE/TITLE.HTM

TruCluster Server Version 5.1B *Cluster Administration* manual:

http://h30097.www3.hp.com/docs/base_doc/DOCUMENTATION/V51B_HTML/ARHGYETE/TITLE.HTM

TruCluster Server Version 5.1B *Highly Available Applications* manual:

http://h30097.www3.hp.com/docs/base_doc/DOCUMENTATION/V51B_HTML/ARHH0ETE/TITLE.HTM

Chapter 9, Optimizing CFS Performance, in the TruCluster Server *Cluster Administration* manual for Tru64 UNIX Version 5.1A:

http://h30097.www3.hp.com/docs/cluster_doc/cluster_51A/HTML/ARHGYDTE/TITLE.HTM

For more information about Oracle 9i RAC on Tru64 UNIX/TruCluster Server Version 5.x, see the following Oracle paper:

http://otn.oracle.com/products/oracle9i/pdf/Oracle_9i_on_Tru64_UNIX.pdf

© 2004 Hewlett-Packard Development Company, L.P. The information contained herein is subject to change without notice. The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HP shall not be liable for technical or editorial errors or omissions contained herein.

UNIX® and The Open Group™ are trademarks of The Open Group in the U.S. and/or other countries. All other product names mentioned herein may be trademarks of their respective owners.

05/2004

