FifthGen White Paper

The Fifth Gen Binary Tree Cluster Computer (BTC-100X)

Part One: Product Overview

Thomas O. Jones

August 9, 2012

Fifth Generation Computer Corporation
445 Park Avenue, 9th Floor
New York, NY 10022
FGC Document No. WP-17

August 8, 2012

Part One: Product Overview

Fifth Gen Binary Tree Cluster Computer (BTC-100X)

A Binary Tree Computer System means a computer system of nodes connected in a binary tree configuration. A binary tree configuration means an arrangement of nodes where each node has a single parent and two children nodes, except the root node, which has no parent, and the leaf nodes, which have no children.

In order to visualize a Binary Tree Computer System, see Figure 1 below.

DIAGRAM OF A CLUSTER COMPUTER CONNECTED IN A BINARY TREE CONFGURATION

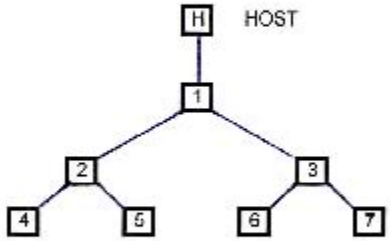


Figure 1. 8 Processors connected in a binary tree configuration

Many Improvements

Since the early days of binary tree computer systems many improvements have been implemented in order to eliminate the bottleneck at the root of the tree. One such improvement was the invention of virtual channels by William Dally, sponsored by DARPA.

Processor and memory speeds have also been dramatically increased.

Why Use a Binary Tree Cluster Computer?

We first ask the question: what advantages are to be gained by organizing a cluster of computers into a binary tree configuration?

The simple answer is SPEED of communications in solving certain types of problems.

But before you buy a cluster of computers or assemble your cluster, you must first determine if the problem you are working on is one of those that can benefit from a multiprocessor-based solution.

This report assumes you have made that decision.

A Computer Cluster connected in a binary tree configuration means an arrangement of nodes wherein each node has a single parent and two children nodes, except the root node, which has no parent, and the leaf nodes, which have no children. Each node is referred to as a Processing Element (PE) which comprises a processor and memory and a means for connecting the PEs to its neighbors. For example, a node could be a Blade computer module.

Unbalanced or asymmetrical trees are consistent with the architecture (see Exhibit A, US Patent No. 6,000,024, top of Column 6).

Diagram 1 on the following page illustrates a complete binary tree of processing elements connected by a network of bus controllers (or Host Channel Adapters). The advantage of this approach is that all communications among the processing elements and the Front End (Host) computer are handled by the communication device, leaving the processors free for computation.

This elegant and simple improvement over the prior art was based on a sophisticated understanding of binary tree computing and its practical use in pattern matching and searching applications.

For example, you will see from the discussion of the fat tree switch below that clusters connected in a binary tree configuration offer congestion-free message passing that is an essential performance requirement for certain mission-critical applications.

As described in a later section of this report, the nodes take the form of a network, with data (messages) moving either up the tree or down the tree. Routing is simple. Messages go up the tree until the target node is reachable on a down route. However, this approach will result in congestion near the root processor.

Charles E. Leiserson was the first to design a solution to this problem by extending the concept of a Clos network from telecommunications to high performance computing communications based on “fat tree” networks.” We discuss his solution in detail later in this report.

Applications

- Big Data Analytics
- Speech Recognition
- Pattern Matching
- Search Engines

Licensing Program

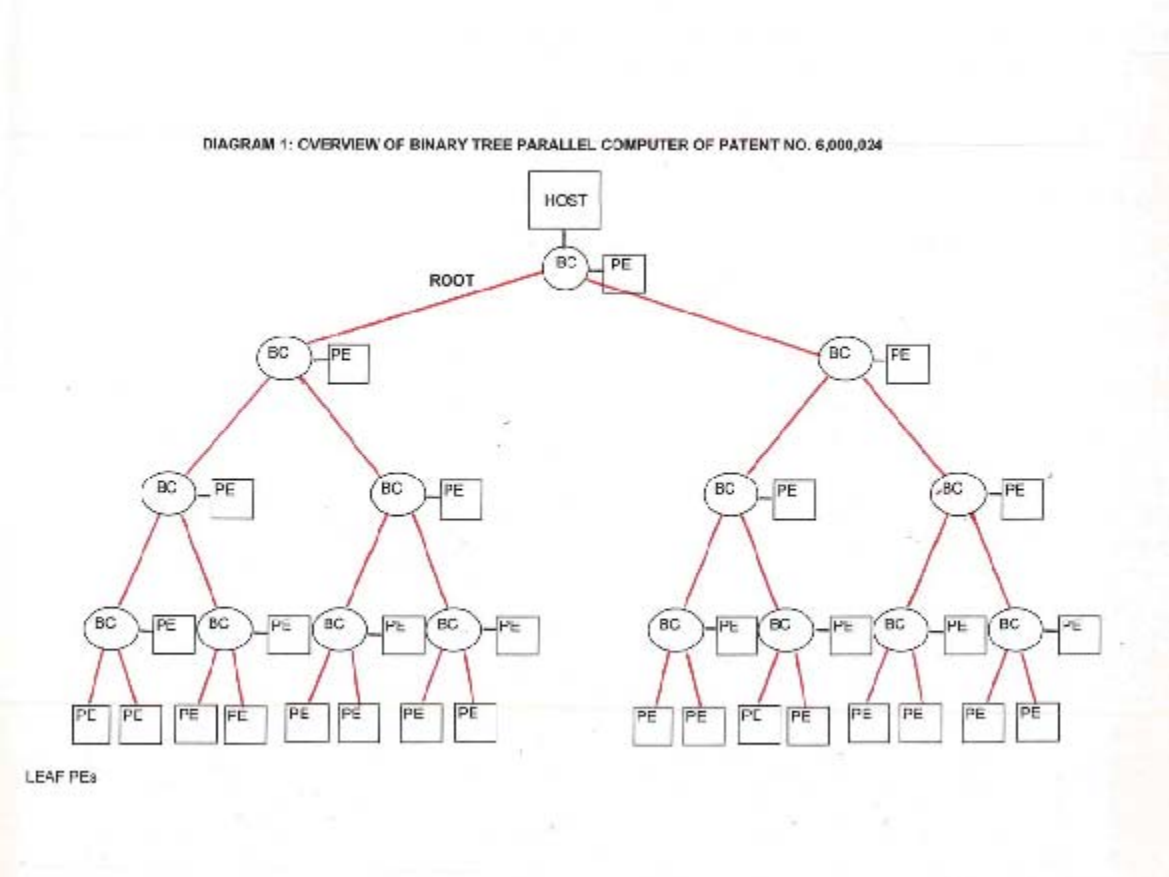
Individual or Vendor Licenses are available at reasonable prices.

E-Mail your request to Tom Jones at tojones@fifthgen.com.

Graphic Illustration

We have included a graphic illustration of the Fifth Gen Binary Tree Cluster Computer below.

The bus controllers (BCs) at each node perform functions similar to those performed by Host Channel Adapters in an Infiniband network.



Examples of Commercial Applications using Computer Clusters interconnected in a Binary Tree Configuration

In order to illustrate the power of binary tree computer systems, we have selected two examples of successful commercial systems whose architecture includes a cluster of computers connected in a binary tree configuration. These descriptions are intended to illustrate the diverse examples of large computer clusters connected in a binary tree configuration. Fifth Generation Computer Corporation was not involved in the Blue Gene project.

AT&T BT-100 System

On October 13, 1986, AT&T and Fifth Generation Computer Corporation, AT&T's sole subcontractor on the project, were awarded a contract from the Defense Advanced Research Projects Agency (DARPA) "to develop prototypes of a computer that can recognize speech and

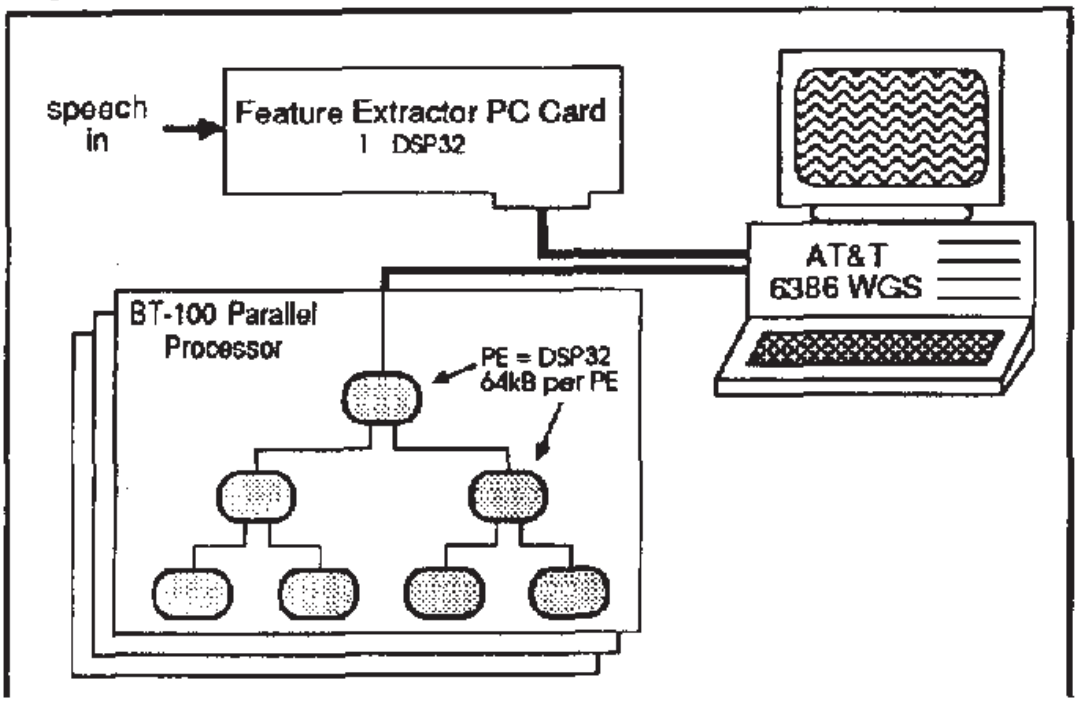
images and do other complex pattern-matching tasks in a fraction of the time of today's best computers."

The AT&T Press Release included the following additional technical information:

> "To demonstrate the power of its parallel-processing architecture, the AT&T machines will run speech-processing algorithms currently under 'development at AT&T \Bell Laboratories. The Dado architecture links processing nodes in a so-called binary tree architecture in which each node communicates only to a parent node" and two "descendants," thus forming a "family tree" of related processors."

The prototypes that the two companies developed were named the AT&T BT-100 Systems which is depicted in the Diagram below.

Figure 2: Parallel Processor for Speech Recognition



On Tuesday, March 3, 1992 AT&T announced "it will deploy voice-recognition technology nationwide to automate many long-distance calls now handled by operators."

As planning and deployment took place, AT&T later announced that "when completed, the phase-out (of jobs) is expected to save the company at least $900 million a year as new technology is deployed and management layers are reduced."

IBM Blue Gene Systems Utilize Binary Tree Networks

Inside every IBM Blue Gene System is an embedded Binary Tree Computer System which IBM calls, the "Collective Network."

The IBM Blue Gene System was jointly developed with Lawrence Livermore National Laboratories (LLNL) under a government contract.

In the April 2005 article, entitled "Into the Wild Blue Yonder with BlueGene/L," Mark Seager, is quoted (at the bottom of page 2):

> "Another difference between BlueGene/L and other platforms is that it has not one but three interconnects for applications: a 3D torus network, a binary-tree (combining and broadcasting) network, and a barrier network."
>
> "BlueGene/L's binary-tree network is useful for low-latency global operations that share data and synchronize programs. This interconnect determines how a highly parallel computer program "talks" to all the nodes quickly and efficiently. "Different ways exist to deliver a message to a large number of nodes," says Seager. "In a binary-tree network, one node talks to two neighbors, those two talk to two of their neighbors, and so on. Getting the message out to 65,536 nodes is a very efficient process, taking only 16 tree operations, or hops."
>
> "The binary tree can operate in broadcast mode to replicate information across the machine or in combining mode to gather data distributed across the machine into a single location. Both broadcast and combining modes are used in operations performed millions of times in real scientific applications. **In BlueGene/L, the binary-tree interconnect is implemented in the hardware rather than in the software, making those hops extremely fast. Performing those operations in the hardware, says Seager, is a huge leap forward in making BlueGene/L scalable and fast."**

In an IBM research report, entitled "Overview of the Blue Gene/L system architecture," authored by Alan. Gara et al, in the IBM Journal of Research and Development (Vol. 49, No. 2/3, 2005 Special Issue on Blue Gene), the authors refer to the tree network as the "Collective Network."

Dr. Alan Gara and three other scientists had been recruited from Columbia University in 1998. He was later named as the Chief Architect of Blue Gene by IBM.

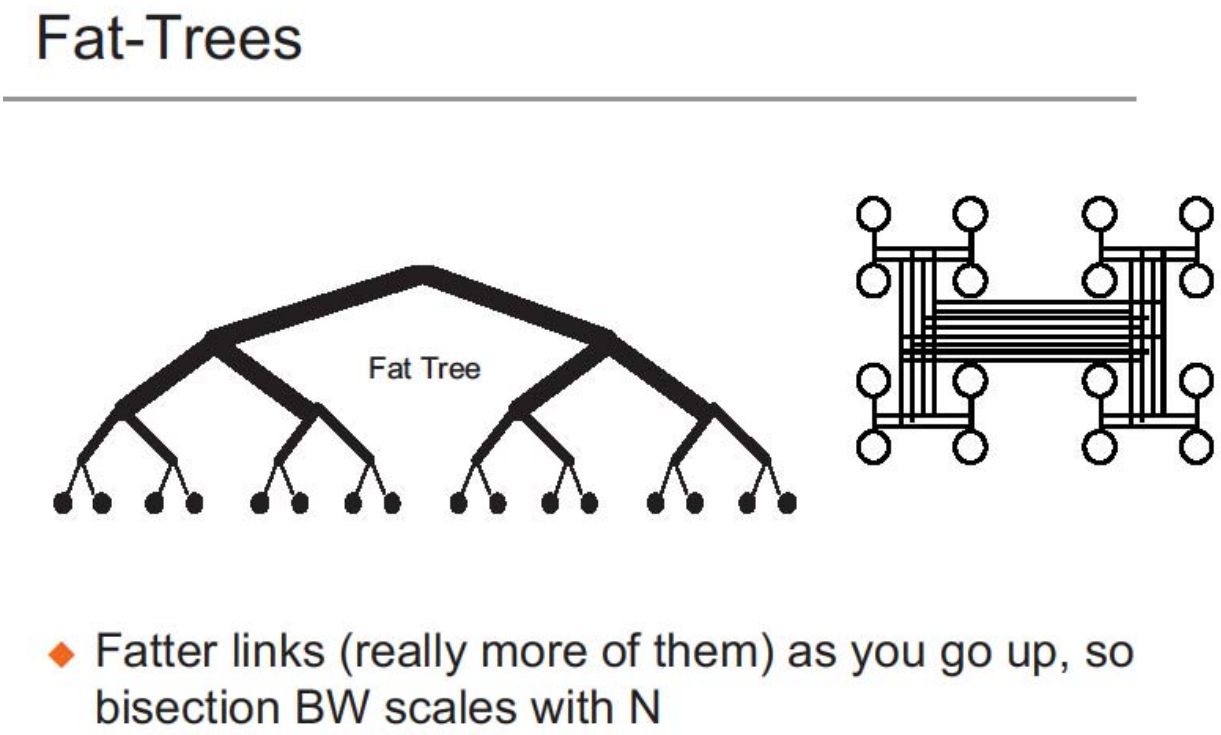
In describing the collective network, Alan Gara calls it a "tremendous improvement:"

> "Arithmetic and logical hardware (ALU) is built into the collective network to support integer reduction operations including min, max, sum, bitwise logical OR, bitwise logical AND, and bitwise logical XOR…**The latency of the collective network is typically at least ten to 100 times less than the network latency of typical supercomputers,** allowing for efficient global operation, even at the scale of the largest BG/L machine.
>
> "The collective network is also used for global broadcast of data, rather than transmitting it around on rings on the torus. For one to all communications, this is a **tremendous**

improvement from a software point of view over the nearest-neighbor 3D torus network."

Two Views of Computer Clusters Connected in a Binary Tree Configuration

The Diagram on the left below depicts the fat-tree topology. One can see that a Fat Tree is a Binary Tree with the addition of higher bandwidth to the links connecting processors at the higher levels of the tree in order to maintain constant bandwidth and eliminate congestion. The Diagram on the right below depicts the same arrangement in the form of an "H-Tree" layout.



The Non-Blocking Fat-Tree Switch

In a Mellanox White Paper the author states:

> "High performance computing clusters typically utilize Clos networks, more commonly known as "Fat Tree" or Constant Bisectional Bandwidth (CBB) networks to construct large node count non-blocking switch configurations.
>
> "The concept of a CBB or Fat-Tree topology is based on the seminal 1953 work on non-blocking switch networks by Charles Clos. This paper describes how a non-blocking switch network can be constructed using a minimal number of basic switch building blocks. Clos networks, developed for the telephone network, have been successfully used in packet based networks, and to first order, provide non-blocking switch configurations.

Leiserson's Explanation of Fat-Tree Interconnection Networks

Charles E. Leiserson was the first to recognize this and extended the concept of a Clos network from telecommunications to high performance computing communications based on "fat tree" networks."

Leiserson's 1985 article, "Fat-Trees: Universal Networks for Hardware-Efficient Supercomputing" defined a fat-tree computer network. On the second page of his report, he states at the beginning of the second paragraph:

> "The intuitive model for parallel computation that we use is a parallel computation engine composed of a set of processors interconnected by a routing network. The processors share no common memory, and thus they must communicate through the routing network, using messages."

Leiserson continues his explanation at the beginning of the third paragraph:

> "A fat-tree FT is a routing network based on a complete binary tree. A set P of n processors is located at the leaves of the fat-tree. Each edge of **the underlying tree** corresponds to two channels of the fat-tree: one from parent to child, the other from child to parent."

The Fat-Tree Network Provides the Interconnecting Links for the Underlying Binary Tree

Leiserson states in the fifth paragraph on page two:

"Routing in the fat-tree is basically easy since every message has a unique path in the underlying binary tree."

Thus, in a High Performance Computer Cluster interconnected by a fat tree switch, **there are two trees**: one is the underlying binary tree configuration of processing elements as nodes and the other is the fat-tree routing network which includes a set of switch nodes.
The Non-Blocking Fat-Tree switch is now the predominant method of interconnecting processor nodes within high performance computer clusters.

The Fat Tree Switch is a Set of Tiered Switches.

The main purpose of the tiers of switches is to provide increased bandwidth for the links between processors at each level above the leaf level of the underlying binary tree configuration.

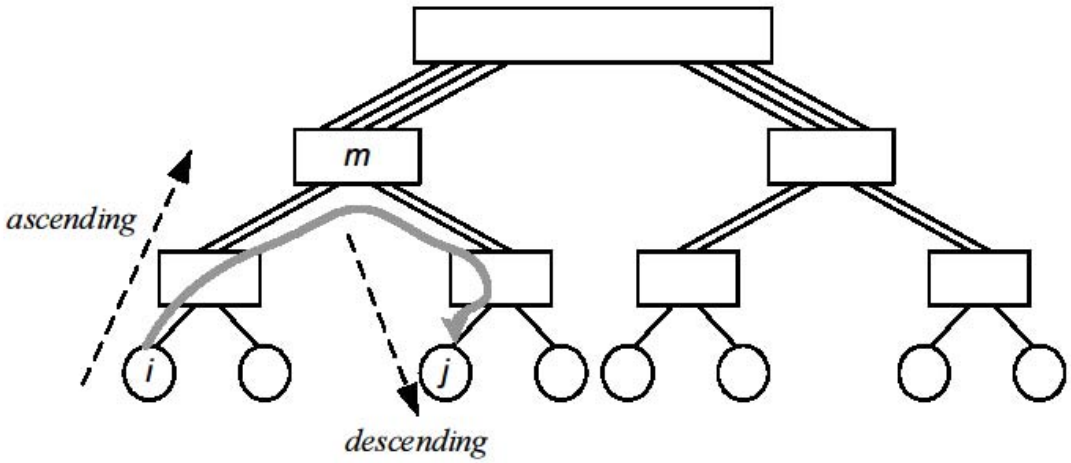


Figure 2: A routing example in a fat-tree.

Visualizing the Underlying Binary Tree Configuration of Processors

The switch ports at the leaves of the fat-tree represent all of the processor nodes in the underlying binary tree configuration. In order for one processor to communicate with the processor at another specific port, the links are very specifically defined by the routing tables and follow the topology of a binary tree.

To trace these links in a switched network, you need to see the routing tables. The user sees only the Port receptacles for connecting computers to the switch nodes. The Server Vendors Pre-set the Routing Tables at the Factory

The InfiniBand Architecture

As published on the Webopedia website, Infiniband architecture is defined as the following:

"Both an I/O architecture and a specification for the transmission of data between processors and I/O devices that has been gradually replacing the PCI bus in high-end servers and PCs. Instead of sending data in parallel, which is what PCI does, InfiniBand sends data in serial and can carry multiple channels of data at the same time in a multiplexing signal. The principles of InfiniBand mirror those of mainframe computer systems that are inherently channel-based systems. InfiniBand channels are created by attaching host channel adapters (HCAs) and target channel adapters (TCAs) through InfiniBand switches. HCAs are I/O engines located within a server. TCAs enable remote storage and network connectivity into the InfiniBand interconnect infrastructure, called a *fabric*. InfiniBand architecture is capable of supporting tens of thousands of nodes in a single subnet.

"InfiniBand is a trademarked term. The technology is a result of the merger of two competing designs -- *Future I/O*, which was developed by Compaq, IBM and Hewlett-Packard, and *Next Generation I/O*, which was developed by Intel, Microsoft and Sun Microsystems. *InfiniBand* was previously called *System I/O*.

InfiniBand transmission rates begin at 2.5GBps."

The Evolution of Infiniband, 2001-2006

The evolution of Infiniband from an I/O connectivity standard to a Distributed Computing standard is well known in the industry. Infiniband provides both a backplane solution "in the box" and an external interconnect. In 2006 the Remote Direct Memory Access (RDMA) function became a standard feature.

As you will see in the following section, RDMA is a standard part of FifthGen's parallel computing system architecture.

Fifth Generation Computer Corporation (Fifth Gen) Patents

Fifth Generation owns US Patent 6,000,024 entitled "PARALLEL COMPUTING SYSTEM" and European Patent No. 1145129. An additional European Patent is pending.

The SPMD Mode of Operation

In a Cluster Computing System Connected in a a binary tree configuration, the major application programming construct is "divide-and-conquer" in which the identical program is executed simultaneously in each Processing Element (PE) but on different data sets. The multiple executions of this single procedure are forced to converge and synchronize at the completion of the individual procedures. This synchronization scheme is known as barrier synchronization.

This mode of operation is referred to as SPMD**,** an acronym that stands for Same Program Multiple Data.

First Published Use of the term SPMD (Single Program, Multiple Data Stream)

In the January 1987 issue of COMPUTER magazine, Professor Salvatore J. Stolfo, the leader of the DADO project (which was partially funded by DARPA) described some early results in an article entitled, "Initial Performance of the DADO2 Prototype," attached as Exhibit D to this report.

> "On December 5, 1985 a 1023-processor parallel machine named DADO2 was successfully demonstrated at Columbia University. DADO2 is the fourth prototype, but the first large-scale prototype of a class of machines called DADO.
>
> "DADO was first proposed in 1980 as a special-purpose parallel computer attached to a conventional host processor and designed to accelerate a particular class of artificial intelligence rule-based programming paradigms called *production systems.*

Later in the same article, Professor Stolfo elaborated on the SPMD mode of operation:

> "DADO2 provides parallel remote procedure invocation in the style of SIMD processing. The procedures are stored locally within the PEs, operate autonomously, and, therefore, may take different amounts of time to complete. Machine level instructions are not broadcast and executed in lock-step. This mode of operation may be regarded as SPMD (for single program, multiple data stream) execution.

In 2007, an article was published as part of the 2007 IEEE International Conference on Cluster Computing entitled, "Efficient Offloading of Collective Communications in Large-Scale Systems," by Sancho, Kerbyson and Barker. In the introduction, the authors stated the following:

> "The single program multiple data (SPMD) programming model is generally the preferred programming model in parallel scientific applications as they usually exhibit a rich degree of data parallelism."

Programming and Running Jobs on the Fifth Gen BTC-100X: BIG DATA EXAMPLE

Using standard tools, each Processor is downloaded a slimmed down version of the Linux Operating System. Then your application program is downloaded to each processor.

Dividing the Data

Using another standard tool, the data that you wish to analyze is equally divided across all of the processing nodes. You now have created a true distributed MPP database ranging from a single server to many thousands of servers with full linear scalability.

Choosing the Result

Using a combination of standard examples and your proprietary application code, you choose the methodology for choosing and reporting the results of your analysis.

The Fifth Gen Binary Tree Cluster Computer (BTC-100X)

Part Two: Technical Discussion

Thomas O. Jones

August 9, 2012

Fifth Generation Computer Corporation
445 Park Avenue, 9th Floor
New York, NY 10022
FGC Document No. WP-18

The Fifth Gen Binary Tree Cluster Computer
(BTC-100X)

Part Two: Technical Discussion

The Fifth Gen BTC-100X is protected by US Patent 6,000,024 and 17 Euro Zone Patents

The Abstract of US Patent 6,000,024

A binary tree computer system connected to a host computer that includes N bus controllers connected in a binary tree configuration in which each bus controller except those at the extremes of the tree are connected to left and right child bus controllers, where N is an integer. One of the bus controllers is a root bus controller that connects the binary tree to the host computer. Each of the bus controllers has an associated processing element attached thereto and two processing elements are connected to each of the bus controllers at the extremes of the binary tree. Each of the processing elements includes a microprocessor and an associated memory. Each of the bus controllers includes, for each of the processing elements connected thereto, a buffered interface connecting the processing element to the bus controller for transmitting instructions and data between the bus controller and the processing element, and for writing and reading information into and from the memory of the processing element without involving the microprocessor.

The ‘024 Patent

The Parallel computing system described in the “024 patent uses a binary tree interconnection scheme. The ‘024 invention includes a number of innovations one of which is the placement of a Bus Controller Module (Bus Controller) at each of the processing nodes with the exception of the leaf processing nodes, which have connections to the bus controllers one level above.

The invention portrayed in the 024 patent is much more than just “the addition of a bus controller at each of the processing nodes.”

The Bus Controller Module (BCM) is a unique advance in the design of parallel computing systems. The BCM technology is introduced in the ‘024 patent by James Maddox, the sole inventor. James Maddox has years of experience as a computer design engineer. His earlier inventions, while at the Philco Corporation, included the first use of transistors in a computer system and the first design of a completely transistorized computer system in the US delivered to the National Security Agency (NSA) in 1955.

The processing elements are the computing workhorses of the system. The BCM included at each node works in conjunction with the processor at each node to allow the processors to concentrate on computing while the BCM (Bus Controller) manages communications up and down the tree.

The controller performs other functions to make the parallel processing system more efficient, such as interpreting and executing instructions.” See Claim 7.

In one embodiment, the inventor configured a version whereby he was able to eliminate the controllers at the leaf processors by combining the functions into the controller one level above so that each processor has an associated controller. He determined that the controllers at this extreme

level could handle the communications for both the processing element at its level and its two children because there is no further communication down the tree by the PE's at the individual leaves.

The Bus Controller also include means for generating a signal when it is not ready to send information up or down the tree to cause all the bus controllers in the path of the information to pause until the bus controller generating such signal ceases to do so. (See Claims 4 and 8.)

Reduced Involvement of the Microprocessor

- Data and programs can be transferred from/to the Host directly into memory without the involvement of the PE's microprocessor.
- Function calls are implemented with software logic in the Tree-Bus Control nodes rather than in the Processor element microprocessors as in the '201 invention.

Nomenclature is very important. FGC's earlier patents, referred to Processing Elements or PEs. The computer industry also uses the term, "nodes," to refer to PEs. However, there is no standard definition of a PE or a Node.

The Parallel Processor of the '024 patent

From the DETAILED DESCRIPTION Section of US Patent 6,000,024:

"Referring to FIG. 1 of the drawings, the binary tree computing system of the invention interconnects a number of Processor Elements (PEs) with each other and with a host computer over a binary-tree bus. The tree-bus of the system illustrated in FIG. 1 consists of three bus control nodes designated as nodes BC1, BC2 and BC3 respectively. The three nodes connect seven PE's, designated as nodes PE1 through PE7, to the host. All of the PEs are identical and in the illustrated embodiment consist of a microprocessor, such as an IBM PowerPC 603e microprocessor, and associated RAM memory with a bridge circuit interconnecting the two."

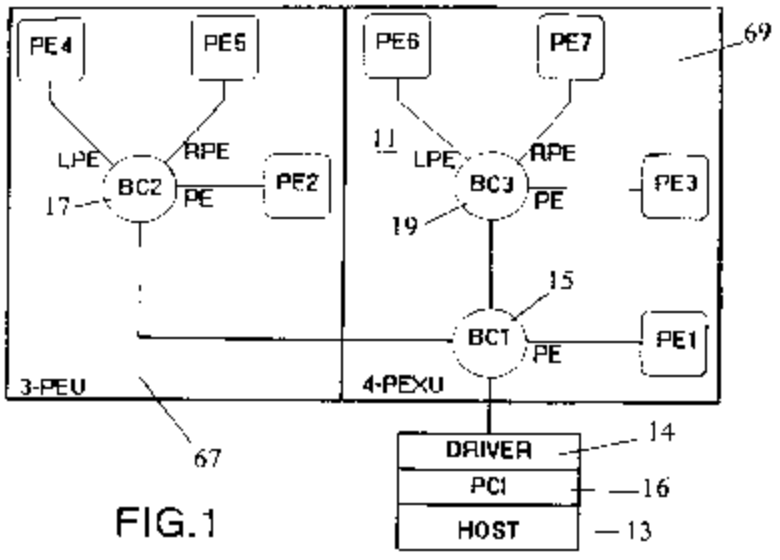


FIG.1

“The system of FIG. 1 illustrates the basic structure of the binary tree parallel computer system of the invention. Each node BCx is connected upstream to a parent node, except for the root node BC1, which is connected to the host. Each node BCx is also connected downstream to its own PE and either to two child nodes BCx, or in the case of the nodes at the extremes of the tree, to right and left leaf PEs.”

The Bus Controller Modules (Bus Controllers)

A Bus Controller is a modular device for “interconnecting a number of Processor Elements (PEs) with each other and with a host computer over a binary tree-bus“(From the Patent, DETAILED DESCRIPTION).

“each of the bus controllers includes, for each processing element connected thereto, a buffered interface connecting the processing element to the bus controller for transmitting instructions and data between the bus controller and the connected processing element. Importantly, each bus controller further includes means for respectively writing and reading information into and from the memory of the connected processing element without involving the microprocessor of the processing element.” (From the SUMMARY OF THE INVENTION).

The Bus Controllers act as buffered repeaters that transfer Function Calls and data from the Host Computer to the selected PE(s), and data with its Fault Message from the selected PE to the Host Computer. The PIOs transfer data between their respective PEs and the binary tree-bus in compliance with Function Call requirements, and <u>arbitrate access of the RAM associated with the respective PE between the needs of the PE’s microprocessor and the binary-tree bus.”</u>

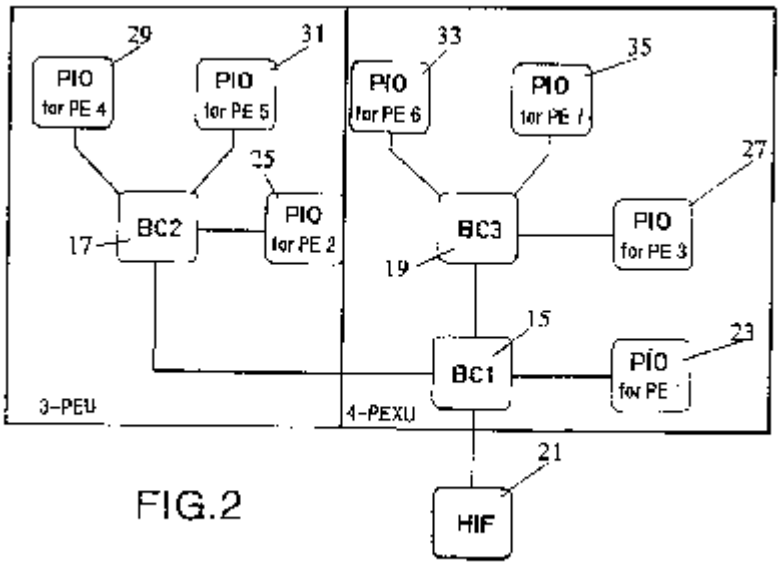


FIG.2

The Binary-tree Bus

"Referring to FIG. 2 of the drawings, the binary-tree bus of the embodiment of FIG. 1 includes a host interface (HIF), three bus control nodes and seven processor input/outputs (PIOs), one for each PE. The HIF interfaces the binary tree-bus with the connection bus of the Host Computer. In the illustrated embodiment of the invention, this bus is preferably a PCI Bus, although others could be used." Figure 2 is an illustration of the Binary-tree bus architecture. The inventor did not include the Processing Elements (PE's) in order to simplify the illustration.

On the following page, we have included Figure 2 (not the above Figure 2) from the FGC MP-6 Design Specification to show a complete Block Diagram of the Preferred Embodiment of the '024 patent.

"Bus Control Nodes (BCx) operate as repeaters to act collectively as a bucket brigade to transfer data between the HIF and the PEs, and PEs to PEs. Referring to the timing diagram of FIG. 6, this is accomplished with the use of the "Ready for Data" RFD and "New Data Present" (NDP) signals. As illustrated in FIG. 6, data are transferred through the BCx (DATA IN and DATA OUT) on consecutive clock cycles until RFD (IN) is unasserted by the next down-stream BC in the chain indicating that it is not ready to accept another data transfer. This causes the BCx to unassert RFD (OUT) to cause the data transfer to pause-up stream. In this case, RFD (IN) is reasserted such that data transfer may resume. BCx retains data until the transfer process continues so that there is no loss of data when the flow of transfers in momentarily interrupted."

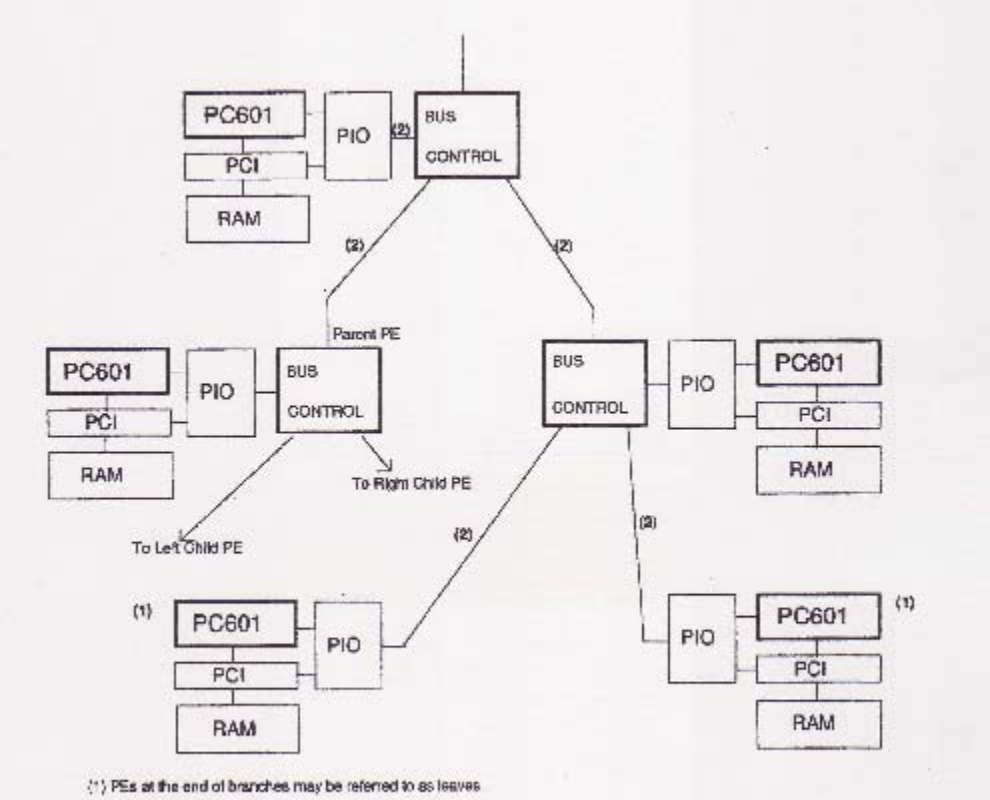
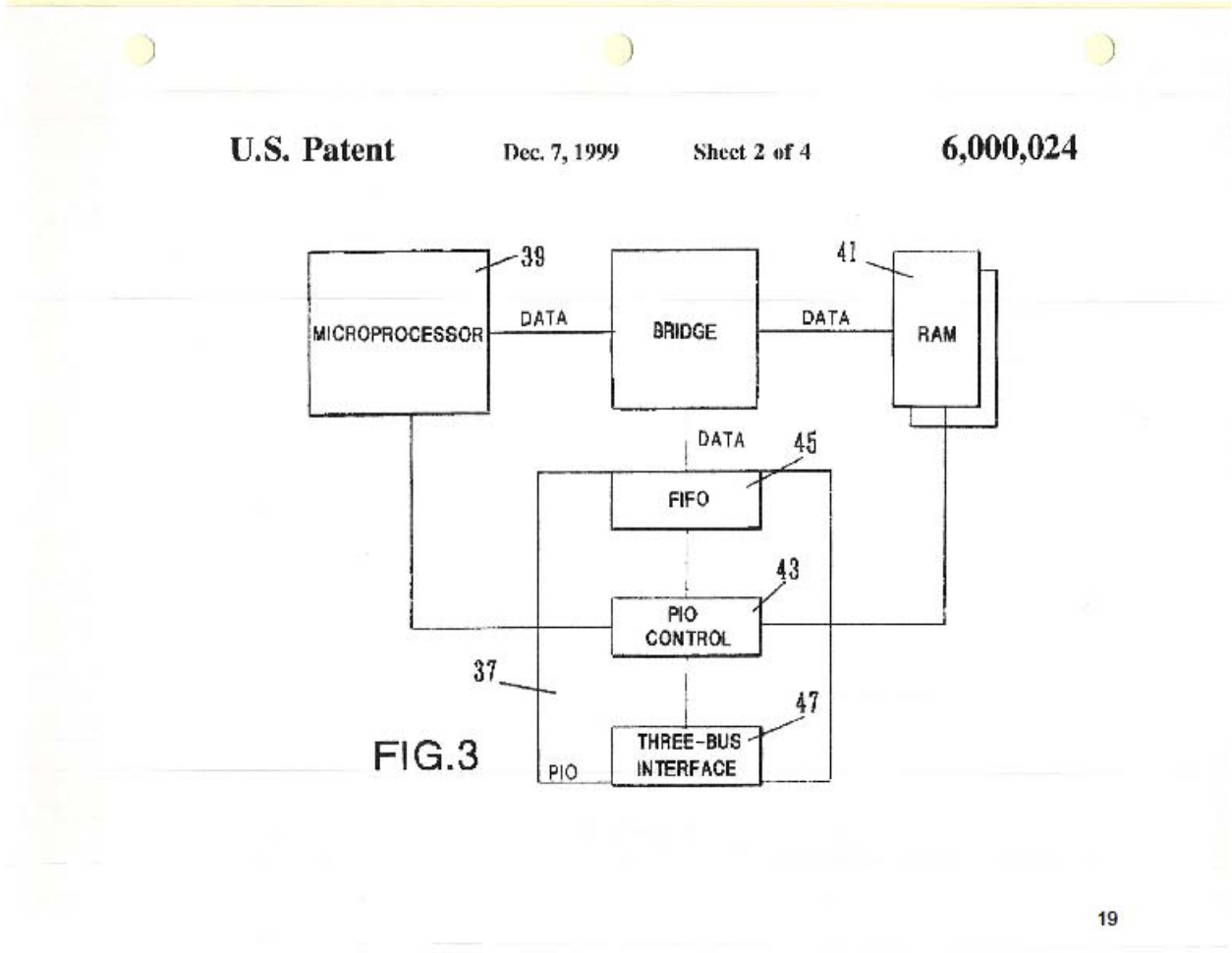


Figure 2 Block Diagram of Processor Elements

The Use of Function calls to further accelerate processing

"In the binary tree computer system of the illustrated embodiment of the invention, the host computer 13 generates instructions referred to as Function Calls to control the operation of the system. These Function Calls are generated by the host computer in the illustrated embodiment of the invention." (DETAILED DESCRIPTION OF THE '024 PATENT).

These function calls are implemented in the FPGA based Bus Controller. Use of these Function calls further offload processing from the microprocessors.



"Referring to FIG. 3 of the drawings, the PIO controls the read/writes of both the PE microprocessor and the binary-tree bus with the RAM of the PE. Data flows between the binary-tree bus and RAM under the control of the PIO. The PIO also includes a buffer that acts to accommodate delays in data transfer that may be caused by memory arbitration conflicts. The HIF contains a similar buffer for the same reason."

"The HIF, BCxs and PIOs are each preferably implemented with Field Programmable Gate Arrays (FPGAs)."

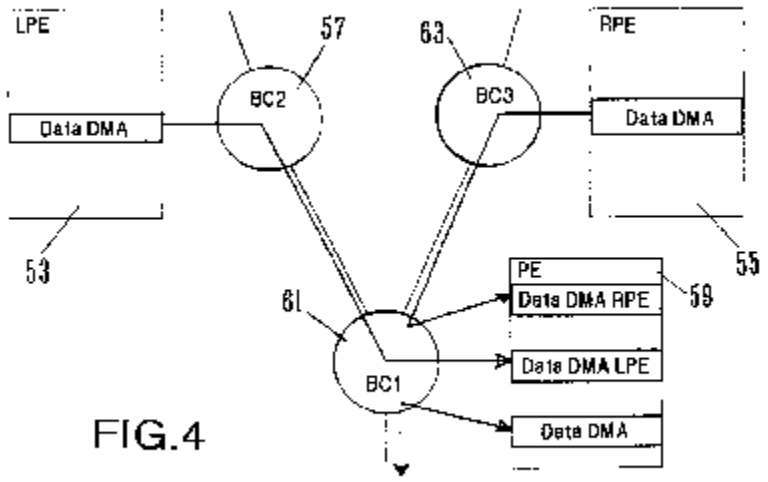


FIG.4

FIG. 4 illustrates an example of the Resolve Function Call. The "best results" are computed by the LPE 53 and the RPE 55 and are stored in their associated memory starting at address "Data DMA." The best results are then transferred from LPE 53's memory via BC2 57 to the memory of PE 59, starting at address data DMA LPE by BCI 61. Next the results are transferred from the memory of RPE 55 BC 63 to the memory of PE 59 starting at location Data DMA RPE by BC1 61. Finally, PE 59 selects the best results among those received from LPE 53 and RPE 5 and that which PE 59 itself computed and the results are stored in PE 59's memory starting at location Data DMA.

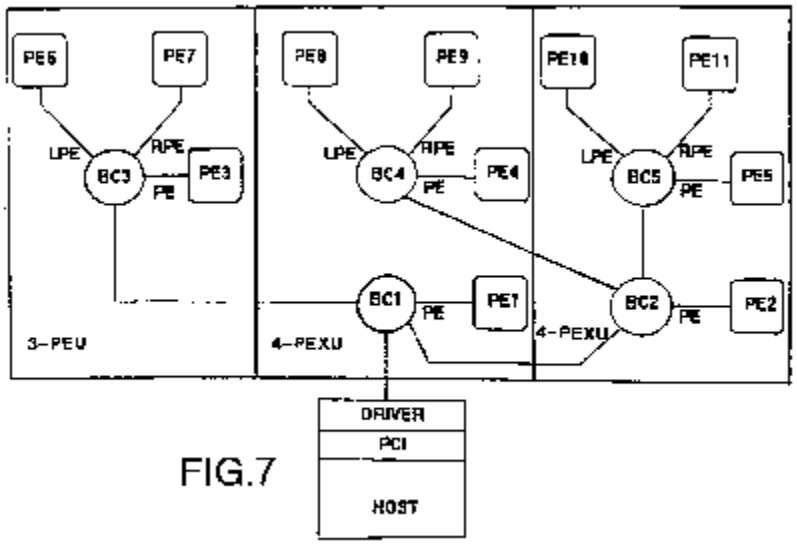


FIG.7

Scalability of the System by Expansion

“The smallest set of PEs in a system in accordance with the invention is three. In such a configuration, PE2 and PE3 are in the leaf positions. This configuration can be scaled to seven PEs by using a 3-PEU in conjunction with a four 4-PEXU expansion unit that conveniently can be disposed on a single printed circuit board. In this configuration PE4 through PE7 are leafs by virtue of being located at "leaf" physical locations on the printed circuit board. To scale-up from a three to seven PE system the BC of the 3-PEU is connected to the first BC of the 4-PEXU which is connected to the host interface.

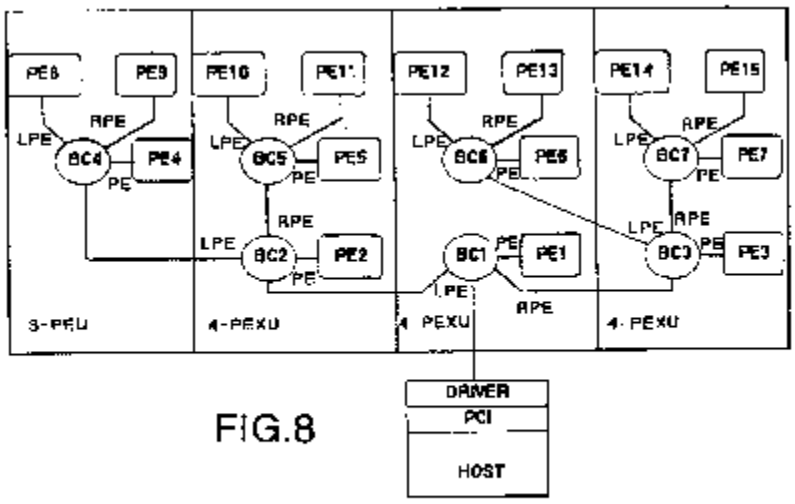


FIG.8

As illustrated in FIG. 7 of the drawings, by adding an additional 4-PEXU, the configuration can continue to be expanded to eleven PEs. This results in an asymmetrical tree which is permitted in the architecture. The eleven PE configuration can be expanded further into the fifteen PE configuration depicted in FIG. 8 by the addition of another 4-PEXU. This process can be continued indefinitely by adding further 4-PEXU's to construct a binary tree of the desired size. In general, a tree constructed in accordance with this aspect of the invention will be constructed from one 3-PEU and as many 4-PEXU's as necessary. Therefore the binary tree can be expanded indefinitely with only two printed circuit board types, i.e. 3-PEU and 4-PEXU. The 4-PEXU has optional connections from the first BC. For example, in FIGS. 1 and 7, BC1 is routed to BC2. In FIG. 8, BC1 is routed to BC3 on the same printed circuit board.

Visualizing the Underlying Binary Tree Network in a Fat-Tree Switched Network

In the ancient exact sciences such as mathematics and astronomy, diagrams always occupy center stage In modern science diagrams serve as a kind of illustration

On the following two pages we have included several diagrams and figures in order to illustrate the underlying binary tree network in a fat-tree non-blocking switch used in both Ethernet and Infiniband HPC configurations.

Figure 1 at the top of the page illustrates the familiar binary tree configuration of processor nodes.

Figure 2 illustrates that the processors appear to be in a linear configuration but, in fact, the connections are the same as in Figure 1 above.

Figure 3 illustrates the same 8 processors arranged in a line but connected in a binary tree configuration through two tiers of four-port switches in order to maintain constant bandwidth at all levels of the tree.

Figure 4 illustrates the addition of links in a fat tree at the higher levels of the tree.

The switch ports in Figure 3 appear as leaves on the tree but the connections to each other rely on the underlying binary tree configuration. For example, in order for one processor to communicate with the processor at another specific port, the links are very specifically defined by the routing tables and follow the topology of a binary tree as explained in Leiserson's original paper.

To trace these links in a switched network, you need to see the routing tables. See the article "A Multiple LID Routing Scheme for Fat-Tree-Based Infiniband Networks," Xuan-Yi Lin, National Univ. Taiwan, IEEE article 2004)

The switch is not one large cross bar switch. The underlying binary tree is implemented with 6 4-port crossbar switches.

The user sees only the Port receptacles for connecting computers to the nodes.

References

"Into the Wide Blue Yonder with BlueGene/L," LLNL internal magazine, April 2005. (Exhibit B in this book, pages 32-34).

"Scaling 10 Gb/s Clustering at Wire Speed." A Mellanox White Paper (Exhibit C in this book, pages 36-39).

"Fat-Trees: Universal Networks for Hardware-Efficient Supercomputing" Charles E. Leiserson, August 1985 IEEE 0018-9340/85/1000-0892 (Exhibit D in this book, pages 41-42)

"Fat-tree routing and node ordering providing contention free traffic for MPI global collectives," by Eitan Zahavi, Mellanox Technologies, LTD, Yokneam, Israel, *Journal of Parallel and Distributed Computing*, February 2012.

"InfiniBand Architecture." HP Technology Brief, 2006.