ORNL/TM-11983

ornl

OAK RIDGE NATIONAL LABORATORY

MARTIN MARIETTA

Communication Performance of the Intel Touchstone DELTA Mesh

Thomas H. Dunigan

MANAGED BY
MARTIN MARIETTA ENERGY SYSTEMS, INC.
FOR THE UNITED STATES
DEPARTMENT OF ENERGY

ORNL/TM-11983

Engineering Physics and Mathematics Division

Mathematical Sciences Section

COMMUNICATION PERFORMANCE OF THE INTEL TOUCHSTONE DELTA MESH

Thomas H. Dunigan

Mathematical Sciences Section
Oak Ridge National Laboratory
P.O. Box 2008, Bldg. 6012
Oak Ridge, TN 37831-6367
thd@ornl.gov

Date Published: January 1992

Research was supported by the Applied Mathematical Sciences Research Program of the Office of Energy Research, U.S. Department of Energy.

Prepared by the
Oak Ridge National Laboratory
Oak Ridge, Tennessee 37831
managed by
Martin Marietta Energy Systems, Inc.
for the
U.S. DEPARTMENT OF ENERGY
under Contract No. DE-AC05-84OR21400

Contents

COMMUNICATION PERFORMANCE OF THE INTEL TOUCHSTONE DELTA MESH

Thomas H. Dunigan

Abstract

The communication performance of the i860-based Intel DELTA mesh supercomputer is compared with the Intel iPSC/860 hypercube and the Ncube 6400 hypercube. Single and multiple hop communicationbandwidth and latencies are measured. Concurrent communication speeds and speed under network load are also measured. File I/O performance of the mesh-attached Concurrent File System is measured.

Keywords: **mesh communication, hypercube communication.**

1. Introduction

In conjunction with a consortium of institutions led by Cal Tech, Intel has developed a massively parallel, distributed memory parallel processor called the Touchstone DELTA system. The DELTA processors are interconnected by a mesh, rather than the hypercube topology used in Intel's earlier parallel systems (iPSC/1, iPSC/2, and iPSC/860). The Intel i860 processor is used as the computing element in the DELTA, the same processor used in the iPSC/860 hypercube. This report describes the DELTA system and contrasts its performance with the iPSC/860 hypercube and the Ncube 6400 hypercube.

The DELTA project is a prototype to demonstrate that a mesh topology using byte-wide communication channels is a more cost-effective utilization of "communication wires" than the bit-wide hypercube topology. That is, given a fixed number of wires, can a more effective parallel processor be constructed using more wires among fewer adjacent processors versus using those wires to provide more direct connectivity? Since the DELTA and iPSC/860 use the same processor, measuring and comparing communication performance between these two parallel processors should provide some answers.

The mesh has some potential advantages over a hypercube topology. Though both topologies are extensible, in practice, commercial hypercubes have a fixed maximum dimension. For example, the largest iPSC/860 is seven dimensions or 128 processors. Hypercubes must be expanded in powers of two, which is often prohibitively expensive. Meshes can be expanded at linear costs by adding an additional row or column. Of course, the hypercube topology has advantages as well. The maximum distance between two processors in an n processor system is only $log_2 n$ for a hypercube, compared with $\sqrt{n}$ for the mesh. The lower connectivity of the mesh may lead to communication "hot spots" in the mesh or to slower aggregate communication operations such as barriers.

In the following section, the hardware specifications of the DELTA mesh are described. In section 3, the communication performance of the mesh is described and contrasted with hypercube communication performance. In section 4, the performance of the attached file system is measured. Section 5 calculates some communication metrics and contrasts the performance of two parallel applications on the mesh and on hypercubes.

2. DELTA configuration

The Intel Touchstone DELTA system is a mesh-connected parallel processor, consisting of 528 i860 compute nodes, 32 80386 I/O nodes, two 80386 network interface nodes, six services nodes, and two tape nodes [5]. Each compute node has 16 million bytes of memory and is connected to a Mesh Routing Chip (MRC) through a Mesh Interface Module (MIM). Each MRC channel is 8-bits wide and has a bandwidth of 65 million bytes/second (MB/s), but the FIFO's on the MIM have only a 26.7 MB/s data rate (Figure 2.1). (As of August, 1991, the peak data rate of the communication system was limited to 22 MB/s [6].) The largest mesh available to an application is 16×32.

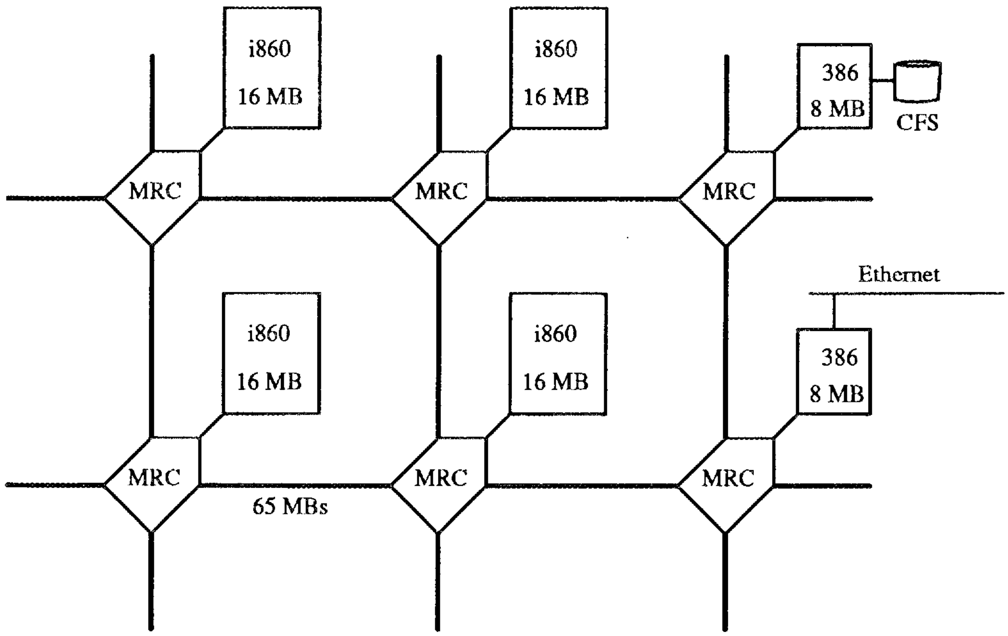

Figure 2.1: *DELTA mesh and nodes.*

The programming environment, attached file system (Concurrent File System, CFS), and network interface nodes are the same as that used on the Intel iPSC/860 [3]. At the time of the tests, the DELTA operating system was NX 3.3, X012 transmittal. The computational performance of a single i860 node is the same as the iPSC/860 node performance reported in [3]. The main difference between the DELTA and iPSC/860 is the number of nodes, 512 versus 128, and the architecture and speed of the communications. The iPSC/860 has bit-serial channels with a peak data rate of 2.8 MB/s connected in a hypercube network. Thus the iPSC/860 has more but slower channels than the DELTA.

For comparison, the following sections include performance data from the iPSC/860 and the Ncube 6400 hypercubes. The hypercube configurations and details of the benchmarks are described in [3].

3. Communication Performance

In this section, we analyze the communication performance of the DELTA mesh, first looking at adjacent node performance, then at communication to more distant nodes. A simple echo test is used, where a message is sent and echoed back by the receiver. The sender measures the round-trip time for 1000 iterations. Additional tests are performed with an artificial load on the communication channel (contention) and with multiple senders (concurrency).

3.1. Node-to-node communication

Figure 3.1 shows the data rate for two adjacent nodes echoing messages of various message lengths. The data rate increases linearly with message sizes from 8 to 8192 bytes, with the DELTA reaching a peak of about 13.1 MB/s for a message size of 100,000 bytes. For large message sizes, the data rate is over a factor of four greater than the iPSC/860 data rate. Also shown in the figure are the

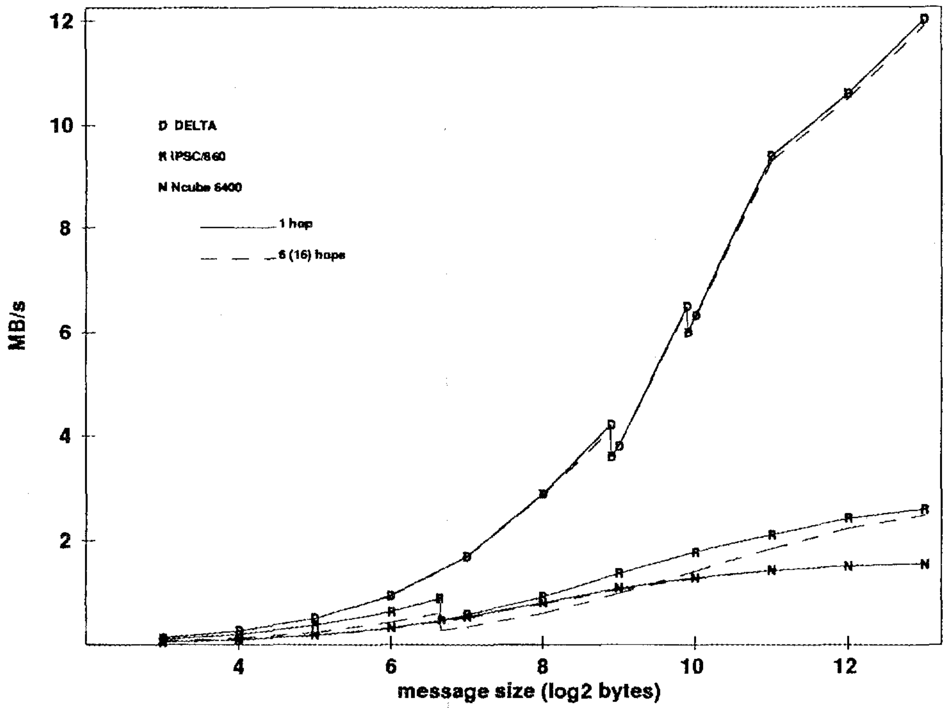

Figure 3.1: *Echo test one- and six-hop data rates.*

data rates for two non-adjacent nodes in a 64-node ensemble, where the distance between the two nodes is 16 hops for an 8×8 mesh and 6 hops for a dimension-6 hypercube. The multi-hop data rates for the DELTA and Ncube are nearly the same as their respective one-hop rates. (The discontinuities in the Intel data rates will be explained later.)

The extra time required for a multi-hop message is more clearly seen if we look at the latency for small messages (Figure 3.2). Though the bandwidth between nodes has increased on the DELTA in comparison to the iPSC/860, the latency has remained about the same. The latency is dominated by house-keeping chores (argument checking, context switch on interrupt, etc.) on the i860 on both the sending and receiving nodes. In a separate study ([2]), the time to handle the time-slice interrupt on the iPSC/860 was about 50 microseconds, which suggests that interrupt context switch overhead could be the dominant factor in message latency.

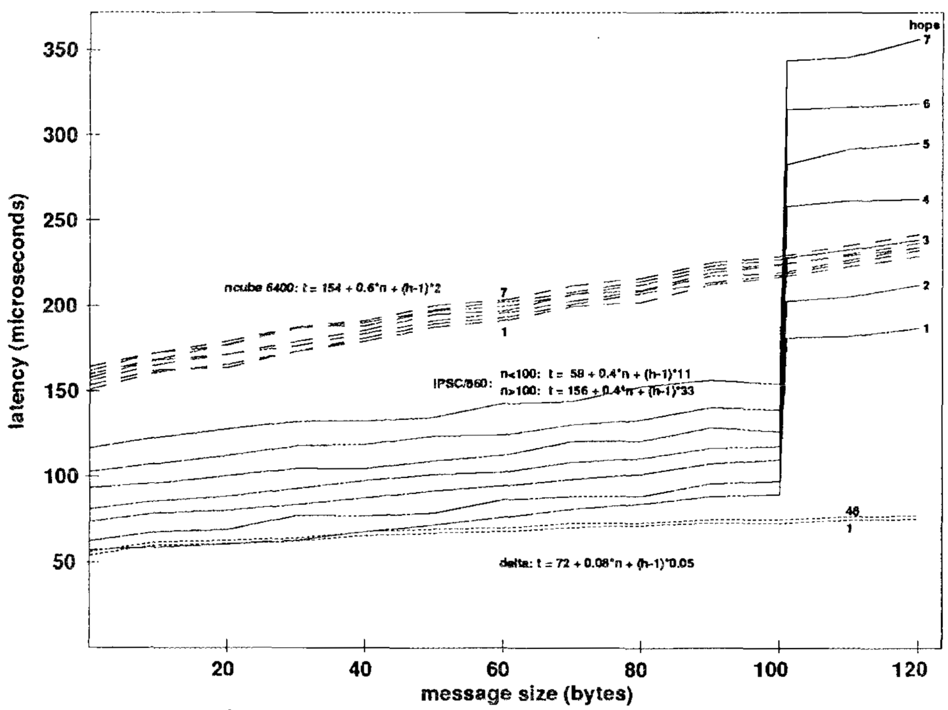

Figure 3.2: *Echo test message latency.*

For the hypercubes and the mesh, communication time, T, grows linearly with the size of the message (N). T is comprised of a startup time, α, transport time (β) proportional to N, and a per-hop delay, γ.

$$T = \alpha + \beta N + (h - 1)\gamma$$

Using a linear least-squares fit to our measured communication times for various message sizes and hops, we can estimate the communication coefficients (Table

3.1).

Coefficients of Communication microseconds			
	DELTA	iPSC/860	N6400
Startup (α)	72	136(75)	154
Byte transfer (β)	0.08	0.4	0.6
Hop penalty (γ)	0.05	33(11)	2

Table 3.1: *Least-squares estimates of communication coefficients.*

Though communication time is on the average linear, Figures 3.1 and 3.2 illustrate some discontinuities in the iPSC/860 and DELTA communication times at some specific message sizes. The iPSC/860 uses a buffer-request/reply protocol for messages larger than 100 bytes. The DELTA communication protocol breaks messages larger than 476 bytes into 476-byte segments, and for messages longer than 6 segments, a buffer-request/reply protocol is employed as well. Figure 3.3 illustrates the DELTA discontinuities in more detail. If the application can

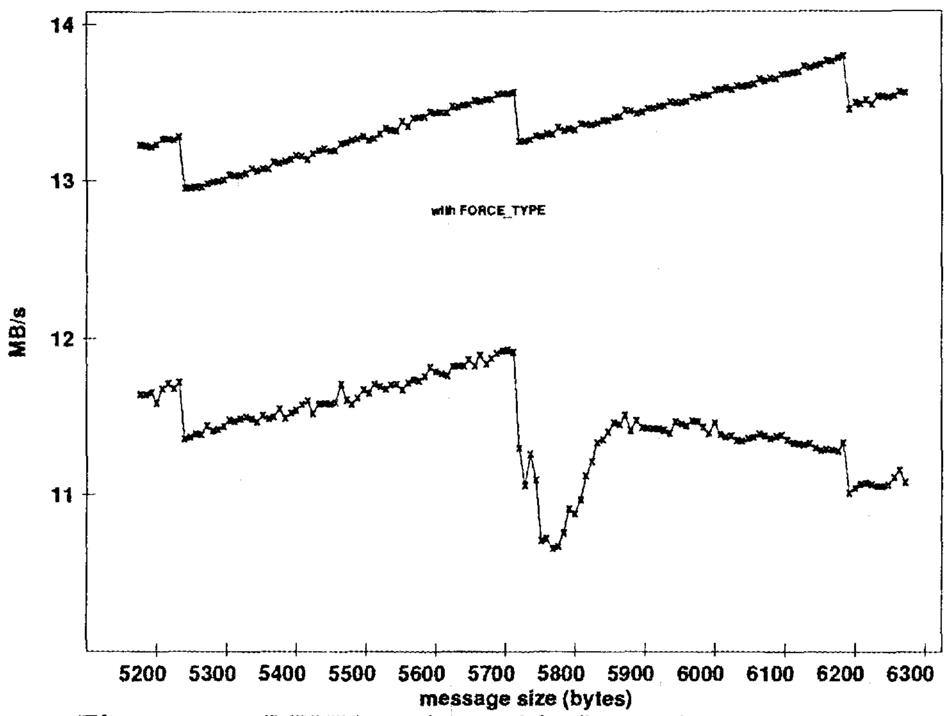

Figure 3.3: *DELTA packet and buffering discontinuities.*

guarantee a receive will be pending, the buffer-request can be eliminated with the FORCE_TYPE option. (If a receive is not pending, the message will be lost.)

Using FORCE_TYPE, the DELTA data rate reaches 17 Megabytes per second for 100,000 byte messages.

Even though the communication performance of the DELTA is generally better than the iPSC/860, the hypercube topology performs some communication primitives faster than the mesh. For example, using Intel's *gsync()*, barrier synchronization time grows with the number of nodes for the mesh, but only as the *log* of the number of nodes for the hypercube (Table 3.2).

Barrier Synchronization microseconds			
nodes	DELTA	iPSC/860	N6400
2	109	111	196
4	248	234	375
8	473	381	375
16	923	546	569
32	1816	692	762
64	3587	847	970

Table 3.2: *Barrier synchronization times.*

3.2. Contention

All of the communication data rates that we have reported have been measured on idle systems. In actual applications, other message traffic may compete for the communication channels, either from the application itself or from applications in other sub-cubes (sub-meshes). Other sub-cubes may need to use another sub-cube's communication channels to reach the host processor, I/O processor, or other service nodes. The iPSC/860, DELTA, and Ncube 6400 use circuit-switching to manage the communication channels. When a message is to be sent, a header packet is sent to reserve the channels required. When this "circuit" is established, the message is transmitted, and an end-of-message indicator releases the channels.

A program was developed to measure the effect of contention on the data rate of a communication channel. For the hypercubes, the program has node 0 continuously send messages to node 7. The messages from node 0 to node 7 pass through node 1 and node 3. The amount of load (measured as a percentage of the total available bandwidth of a single channel) presented by node 0 is varied by selecting various messages sizes. With a communication load from node 0 to node 7, node 1 then sends a stream of messages to node 3. Node 3 measures the data rate of messages arriving from node 1 under various loads and for various

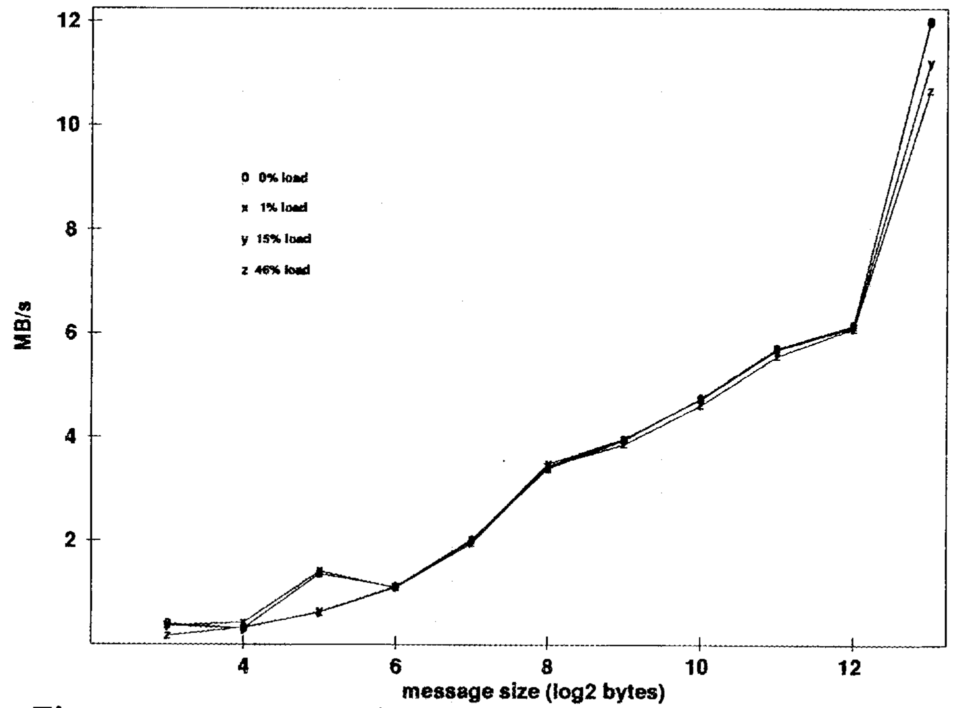

Figure 3.4: *Data rates for the DELTA with channel contention.*

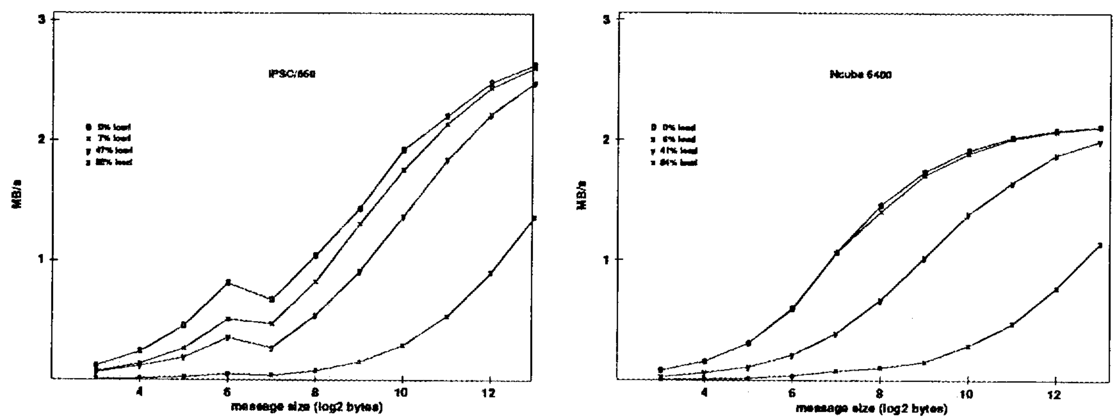

Figure 3.5: *Data rates for the iPSC/860 and Ncube 6400 with channel contention.*

message sizes. For the DELTA mesh, nodes 1 and 2 communicate in the presence of load from nodes 0 and 3 in a 1×4 mesh. The mesh (Figure 3.4) shows little effect from contention under the tested loads, whereas both hypercubes (Figure 3.5) exhibit the expected behavior, as the load from node 0 to 7 increases, the data rate from node 1 to node 3 decreases. The effect of contention can vary from run to run and can slow down an application. Since the DELTA mesh has fewer channels between nodes than the hypercube architecture, one would expect increased contention for the mesh channels. The higher data rates of the DELTA mesh channels will reduce the effect of contention, but communication "hot spots" may still develop for some mesh applications.

3.3. Concurrent Communication

The message-passing performance of a node may be improved by utilizing more than one of its communication channels at the same time. A fan-in test was constructed to measure the aggregate data rate of a single node when one or more of its nearest neighbors are sending it messages. Figure 3.6 shows the aggregate receive data rate for various size messages when 1, 2, 3, and 4 mesh neighbors send

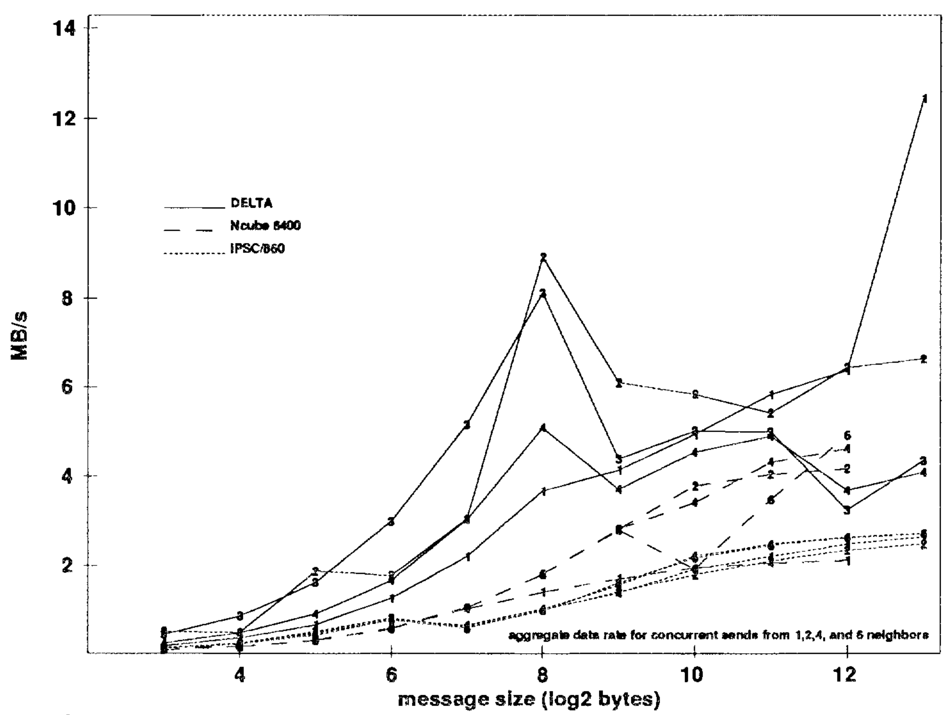

Figure 3.6: *Aggregate receive data rates for concurrent sends.*

concurrently, contrasted with 1, 2, 4, and 6 nearest hypercube neighbors sending concurrently [3]. The iPSC/860 shows only a slight improvement in data rate as more neighbors send messages. The DELTA shows improvement for message sizes smaller than the 476-byte segment size, but for larger messages, the interleaving of segments results in no apparent improvement in data rate. Even though the Intel channels for the iPSC/860 and DELTA can operate concurrently, only one channel can use the node FIFO at a time, thus limiting the amount of concurrent communication to a single node. In contrast, the data rate measured by the receiving Ncube 6400 node increases markedly as additional nearest neighbors transmit to it concurrently [3].

In summary, the communication performance of the DELTA mesh provides fewer but higher bandwidth channels between adjacent nodes than the iPSC/860 hypercube. Both Intel communications systems have about the same latency for small messages, but the wider mesh channels provide nearly six times the bandwidth. The high bandwidth and fast routing of the DELTA mesh further help reduce contention. Although there are more hops in a mesh than for a hypercube with the same number of nodes, the multi-hop penalty on the DELTA mesh is much smaller than that of the iPSC/860. So small, in fact, an application can treat the mesh nodes as if they were all "adjacent" nodes.

4. File System Performance

The DELTA system provides 32 I/O nodes supporting the Concurrent File System (CFS). Files are striped across the drives, under the control of the user. The I/O nodes are directly connected to the mesh, but otherwise the hardware and software is the same as the iPSC/860 CFS. Figure 4.1 compares the read throughput of the DELTA CFS with earlier iPSC/860 results [3]. The iPSC/860 configuration had only 10 I/O nodes. As the figure illustrates, the CFS performance for the DELTA when using 10 or less I/O nodes differs little from the iPSC/860 performance. Even though the DELTA communication is faster, the communication rate from the compute nodes to the I/O nodes is only a small percent of the total I/O performance data rate (which includes disk latency, SCSI bus data rate, and file-system software overhead). Using all 32 I/O nodes, the aggregate date rate for 32 compute nodes each reading their own 16 megabyte files is 11 MB/s. For both the iPSC/860 and DELTA, throughput decreases for this simple read test when the number of compute nodes exceeds the number of I/O nodes, mainly because of thrashing within the I/O node buffers.

DELTA provides two 80386-based network nodes, each with 4 megabytes of memory and an Ethernet interface; the same hardware/software that is used on

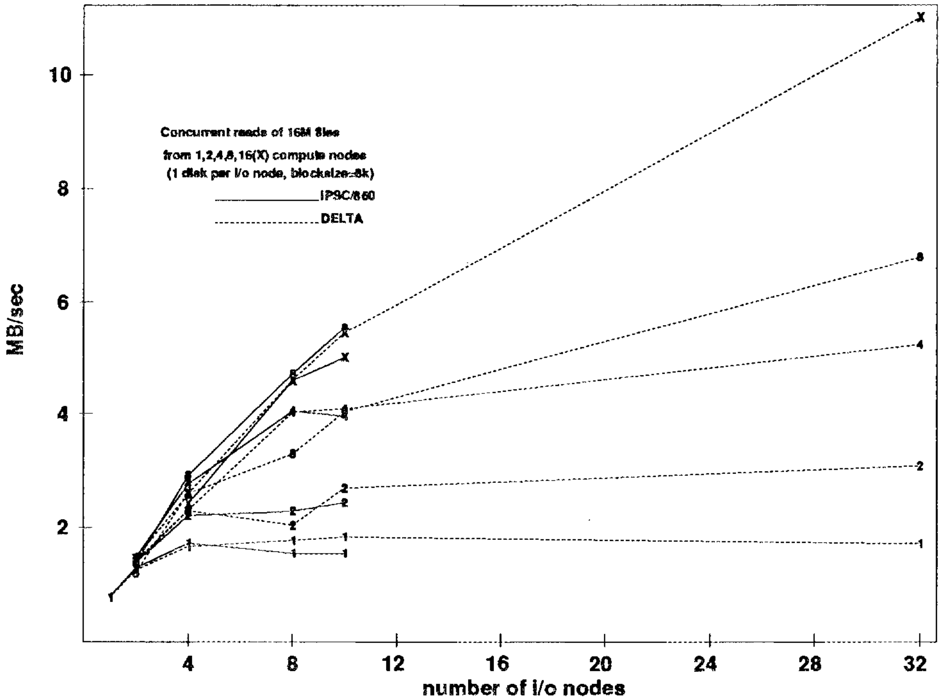

Figure 4.1: *CFS read throughput.*

the iPSC/860 network nodes. The TCP data rate for one of these Ethernet nodes is roughly the same as for the iPSC/860 [3], ranging from 40 KB/s (8-byte message) to 304 KB/s (4096-byte messages). As with CFS, the mesh data rate is only a small factor in the Ethernet date rates.

5. Summary

The Intel DELTA mesh provides improved communication performance over the Intel iPSC/860 hypercube. The DELTA mesh provides wider and faster communication channels between nodes, plus faster routing hardware, but the reduced connectivity of the mesh slows some communication primitives such as barriers. The message startup times are nearly identical, being dominated by software overhead and the speed of the i860. Table 5.1 summarizes the communication and computational performance of the DELTA machine. The data rates represent the 8192-byte transfer speeds, and the megaflops rate is calculated from a five operation expression [3]. The 8-byte transfer time is based on the 8-byte, one-hop, echo times. The structure of a parallel algorithm will be dictated by the amount of memory available on a node, the host-to-node communication speed, and the ratio of communication speed to computation speed. As can be seen from the table, the DELTA and iPSC/860 have roughly equivalent communication-to-computation ratios. (The ratio was calculated using the 8-byte transfer and multiply times.) For larger messages, the DELTA would show a more balanced ratio than the iPSC/860.

Figures of Merit			
	DELTA	iPSC/860	N6400
Data rate (MB/s)	11.9	2.6	1.6
Megaflops	18	18	2.5
8-byte transfer time (μs)	62	80	161
8-byte multiply time (μs)	0.08	0.08	1.5
Comm./Comp.	775	1000	107

Table 5.1: *Summary performance metrics.*

To compare the performance of the DELTA machine to the earlier machines, in an application involving both communication and computation, we solved a 1000 $\times$ 1000 linear system of equations (C double precision) using Cholesky factorization on 16 nodes. The DELTA ran at 30.1 Megaflops compared with 22.2 Megaflops from the iPSC/860 (the Ncube 6400 was 5.3 Megaflops). These results are consistent with the LINPACK results reported in [1]. The LINPACK peak performance (measured by solving the largest linear system the memory can support) for 128 nodes was 3.6 Gigaflops for the DELTA (16 Megabytes of memory per node) versus 1.9 Gigaflops for the iPSC/860 (8 Megabytes) and 0.24 Megaflops for the Ncube 6400 (4 Megabytes) [1]. Using all available nodes, the peak LINPACK was 13.9 Gigaflops for the 512-node DELTA versus 1.9 Gigaflops for the 1024-node Ncube. (The maximum number of nodes for an iPSC/860 is 128.)

To measure the performance of all of the DELTA subsystems (computation, communication, and I/O), we ran the FORTRAN SLALOM benchmark (version 1) [4] on a 64-node mesh. SLALOM on the DELTA ran at 205 Megaflops on a 64-node mesh, as compared with 135 Megaflops for a 64-node iPSC/860 and 16 Megaflops for a 64-node Ncube 6400. Since the compute nodes are identical on the iPSC/860 and the DELTA, the improved performance results from the faster communications. The contribution of a larger CFS had little effect on the SLALOM performance. So the faster mesh communication of the DELTA results in improved application performance.

6. Acknowledgements

Special thanks are given to the members of the Concurrent Supercomputing Consortium for their cooperation and contribution of time and access to the Touchstone DELTA system.

7. References

[1] J. Dongarra. Performance of various computers using standard linear equations software. Technical report, University of Tennessee, January 1991. CS-89-85.

[2] T. H. Dunigan. Hypercube clock synchronization. Technical report, Oak Ridge National Laboratory, 1991. ORNL/TM-11744.

[3] T. H. Dunigan. Performance of the Intel iPSC/860 and Ncube 6400 hypercubes. Technical report, Oak Ridge National Laboratory, 1991. ORNL/TM-11491.

[4] John Gustafson, Diane Rover, Stephen Elbert, and Michael Carter. The design of a scalable, fixed-time computer benchmark. Technical report, Ames Laboratory, 1990.

[5] Intel. *A Touchstone DELTA System Description.* Intel, Portland, Oregon, 1991. Intel Advanced Information.

[6] Greg Regnier. Intel touchstone delta message passing performance. Technical report, Intel, 1991.

ORNL/TM-11983

INTERNAL DISTRIBUTION

1. B. R. Appleton
2–3. T. S. Darland
4. J. J. Dongarra
5–9. T. H. Dunigan
10. G. A. Geist
11. M. R. Leuze
12. C. E. Oliver
13. R. T. Primm
14–18. S. A. Raby
19–23. R. F. Sincovec
24–28. R. C. Ward
29. P. H. Worley
30. Central Research Library
31. ORNL Patent Office
32. K-25 Plant Library
33. Y-12 Technical Library / Document Reference Station
34. Laboratory Records - RC
35–36. Laboratory Records Department

EXTERNAL DISTRIBUTION

37. Cleve Ashcraft, Boeing Computer Services, P.O. Box 24346, M/S 7L-21, Seattle, WA 98124-0346
38. Donald M. Austin, 6196 EECS Bldg., University of Minnesota, 200 Union St., S.E., Minneapolis, MN 55455
39. Robert G. Babb, Oregon Graduate Institute, CSE Department, 19600 N.W. von Neumann Drive, Beaverton, OR 97006-1999
40. Lawrence J. Baker, Exxon Production Research Company, P.O. Box 2189, Houston, TX 77252-2189
41. Jesse L. Barlow, Department of Computer Science, Pennsylvania State University, University Park, PA 16802
42. Edward H. Barsis, Computer Science and Mathematics, P. O. Box 5800, Sandia National Laboratories, Albuquerque, NM 87185
43. Chris Bischof, Mathematics and Computer Science Division, Argonne National Laboratory, 9700 South Cass Avenue, Argonne, IL 60439
44. Ake Bjorck, Department of Mathematics, Linkoping University, S-581 83 Linkoping, Sweden
45. Roger W. Brockett, Wang Professor of Electrical Engineering and Computer Science, Division of Applied Sciences, Harvard University, Cambridge, MA 02138
46. James C. Browne, Department of Computer Science, University of Texas, Austin, TX 78712
47. Bill L. Buzbee, Scientific Computing Division, National Center for Atmospheric Research, P.O. Box 3000, Boulder, CO 80307
48. Donald A. Calahan, Department of Electrical and Computer Engineering, University of Michigan, Ann Arbor, MI 48109

49. John Cavallini, Acting Director, Scientific Computing Staff, Applied Mathematical Sciences, Office of Energy Research, U.S. Department of Energy, Washington, DC 20585

50. Ian Cavers, Department of Computer Science, University of British Columbia, Vancouver, British Columbia V6T 1W5, Canada

51. Tony Chan, Department of Mathematics, University of California, Los Angeles, 405 Hilgard Avenue, Los Angeles, CA 90024

52. Jagdish Chandra, Army Research Office, P.O. Box 12211, Research Triangle Park, NC 27709

53. Eleanor Chu, Department of Computer Science, University of Waterloo, Waterloo, Ontario, Canada N2L 3G1

54. Melvyn Ciment, National Science Foundation, 1800 G Street N.W., Washington, DC 20550

55. Tom Coleman, Department of Computer Science, Cornell University, Ithaca, NY 14853

56. Paul Concus, Mathematics and Computing, Lawrence Berkeley Laboratory, Berkeley, CA 94720

57. Andy Conn, IBM T. J. Watson Research Center, P.O. Box 218, Yorktown Heights, NY 10598

58. John M. Conroy, Supercomputer Research Center, 17100 Science Drive, Bowie, MD 20715-4300

59. Jane K. Cullum, IBM T. J. Watson Research Center, P.O. Box 218, Yorktown Heights, NY 10598

60. George Cybenko, Center for Supercomputing Research and Development, University of Illinois, 104 S. Wright Street, Urbana, IL 61801-2932

61. George J. Davis, Department of Mathematics, Georgia State University, Atlanta, GA 30303

62. Tim A. Davis, Computer and Information Sciences Department, 301 CSE, University of Florida, Gainesville, Florida 32611-2024

63. John J. Dorning, Department of Nuclear Engineering Physics, Thornton Hall, McCormick Road, University of Virginia, Charlottesville, VA 22901

64. Iain Duff, Numerical Analysis Group, Central Computing Department, Atlas Centre, Rutherford Appleton Laboratory, Didcot, Oxon OX11 0QX, England

65. Patricia Eberlein, Department of Computer Science, SUNY at Buffalo, Buffalo, NY 14260

66. Stanley Eisenstat, Department of Computer Science, Yale University, P.O. Box 2158 Yale Station, New Haven, CT 06520

67. Howard C. Elman, Computer Science Department, University of Maryland, College Park, MD 20742

68. Albert M. Erisman, Boeing Computer Services, P.O. Box 24346, M/S 7L-21, Seattle, WA 98124-0346

69. Geoffrey C. Fox, Northeast Parallel Architectures Center, 111 College Place, Syracuse University, Syracuse, NY 13244-4100

70. Paul O. Frederickson, NASA Ames Research Center, RIACS, M/S T045-1, Moffett Field, CA 94035

71. Robert E. Funderlic, Department of Computer Science, North Carolina State University, Raleigh, NC 27650

72. K. Gallivan, Computer Science Department, University of Illinois, Urbana, IL 61801

73. David M. Gay, Bell Laboratories, 600 Mountain Avenue, Murray Hill, NJ 07974

74. C. William Gear, Computer Science Department, University of Illinois, Urbana, IL 61801

75. W. Morven Gentleman, Division of Electrical Engineering, National Research Council, Building M-50, Room 344, Montreal Road, Ottawa, Ontario, Canada K1A 0R8

76. J. Alan George, Vice President, Academic and Provost, Needles Hall, University of Waterloo, Waterloo, Ontario, Canada N2L 3G1

77. John R. Gilbert, Xerox Palo Alto Research Center, 3333 Coyote Hill Road, Palo Alto CA 94304

78. Gene H. Golub, Department of Computer Science, Stanford University, Stanford, CA 94305

79. Joseph F. Grcar, Division 8331, Sandia National Laboratories, Livermore, CA 94550

80. John Gustafson, Ames Laboratory, Iowa State University, Ames, IA 50011

81. Michael T. Heath, National Center for Supercomputing Applications, 4157 Beckman Institute, University of Illinois, 405 North Mathews Avenue, Urbana, IL 61801-2300

82. Don E. Heller, Physics and Computer Science Department, Shell Development Co., P.O. Box 481, Houston, TX 77001

83. Charles J. Holland, Air Force Office of Scientific Research, Building 410, Bolling Air Force Base, Washington, DC 20332

84. Robert E. Huddleston, Computation Department, Lawrence Livermore National Laboratory, P.O. Box 808, Livermore, CA 94550

85. Ilse Ipsen, Department of Computer Science, Yale University, P.O. Box 2158 Yale Station, New Haven, CT 06520

86. Lennart Johnsson, Thinking Machines Inc., 245 First Street, Cambridge, MA 02142-1214

87. Harry Jordan, Department of Electrical and Computer Engineering, University of Colorado, Boulder, CO 80309

88. Malvyn H. Kalos, Cornell Theory Center, Engineering and Theory Center Bldg., Cornell University, Ithaca, NY 14853-3901

89. Hans Kaper, Mathematics and Computer Science Division, Argonne National Laboratory, 9700 South Cass Avenue, Bldg. 221, Argonne, IL 60439

90. Kenneth Kennedy, Department of Computer Science, Rice University, P.O. Box 1892, Houston, TX 77001

91. Thomas Kitchens, Department of Energy, Scientific Computing Staff, Office of Energy Research, ER-7, Office G-236 Germantown, Washington, DC 20585

92. Richard Lau, Office of Naval Research, 1030 E. Green Street, Pasadena, CA 91101

93. Alan J. Laub, Department of Electrical and Computer Engineering, University of California, Santa Barbara, CA 93106

94. Robert L. Launer, Army Research Office, P.O. Box 12211, Research Triangle Park, NC 27709

95. Charles Lawson, MS 301-490, Jet Propulsion Laboratory, 4800 Oak Grove Drive, Pasadena, CA 91109

96. James E. Leiss, Rt. 2, Box 142C, Broadway, VA 22815

97. John G. Lewis, Boeing Computer Services, P.O. Box 24346, M/S 7L-21, Seattle, WA 98124-0346

98. Robert F. Lucas, Supercomputer Research Center, 17100 Science Drive, Bowie, MD 20715-4300

99. Franklin Luk, Electrical Engineering Department, Cornell University, Ithaca, NY 14853

100. Paul C. Messina, Mail Code 158-79, California Institute of Technology, 1201 E. California Blvd., Pasadena, CA 91125

101. James McGraw, Lawrence Livermore National Laboratory, L-306, P.O. Box 808, Livermore, CA 94550

102. Neville Moray, Department of Mechanical and Industrial Engineering, University of Illinois, 1206 West Green Street, Urbana, IL 61801

103. Cleve Moler, The Mathworks, 325 Linfield Place, Menlo Park, CA 94025

104. Brent Morris, National Security Agency, Ft. George G. Meade, MD 20755

105. Dianne P. O'Leary, Computer Science Department, University of Maryland, College Park, MD 20742

106. James M. Ortega, Department of Applied Mathematics, Thornton Hall, University of Virginia, Charlottesville, VA 22901

107. Roy P. Pargas, Department of Computer Science, Clemson University, Clemson, SC 29634-1906

108. Beresford N. Parlett, Department of Mathematics, University of California, Berkeley, CA 94720

109. Merrell Patrick, Department of Computer Science, Duke University, Durham, NC 27706

110. Robert J. Plemmons, Departments of Mathematics and Computer Science, Box 7311, Wake Forest University Winston-Salem, NC 27109

111. Jesse Poore, Department of Computer Science, Ayres Hall, University of Tennessee, Knoxville, TN 37996-1301

112. Alex Pothen, Department of Computer Science, Pennsylvania State University, University Park, PA 16802

113. Yuanchang Qi, IBM European Petroleum Application Center, P.O. Box 585, N-4040 Hafrsfjord, Norway

114. Giuseppe Radicati, IBM European Center for Scientific and Engineering Computing, via del Giorgione 159, I-00147 Roma, Italy

115. John K. Reid, Numerical Analysis Group, Central Computing Department, Atlas Centre, Rutherford Appleton Laboratory, Didcot, Oxon OX11 0QX, England

116. Werner C. Rheinboldt, Department of Mathematics and Statistics, University of Pittsburgh, Pittsburgh, PA 15260

117. John R. Rice, Computer Science Department, Purdue University, West Lafayette, IN 47907

118. Donald J. Rose, Department of Computer Science, Duke University, Durham, NC 27706

119. Edward Rothberg, Department of Computer Science, Stanford University, Stanford, CA 94305

120. Joel Saltz, ICASE, MS 132C, NASA Langley Research Center, Hampton, VA 23665

121. Ahmed H. Sameh, Center for Supercomputing R&D, 1384 W. Springfield Avenue, University of Illinois, Urbana, IL 61801

122. Robert Schreiber, RIACS, Mail Stop 230-5, NASA Ames Research Center, Moffet Field, CA 94035

123. Martin H. Schultz, Department of Computer Science, Yale University, P.O. Box 2158 Yale Station, New Haven, CT 06520

124. David S. Scott, Intel Scientific Computers, 15201 N.W. Greenbrier Parkway, Beaverton, OR 97006

125. Andy Sherman, Department of Computer Science, Yale University, P.O. Box 2158 Yale Station, New Haven, CT 06520

126. Kermit Sigmon, Department of Mathematics, University of Florida, Gainesville, FL 32611

127. Horst Simon, Mail Stop T045-1, NASA Ames Research Center, Moffett Field, CA 94035

128. Danny C. Sorensen, Department of Mathematical Sciences, Rice University, P. O. Box 1892, Houston, TX 77251

129. G. W. Stewart, Computer Science Department, University of Maryland, College Park, MD 20742

130. Paul N. Swartztrauber, National Center for Atmospheric Research, P.O. Box 3000, Boulder, CO 80307

131. Robert G. Voigt, ICASE, MS 132-C, NASA Langley Research Center, Hampton, VA 23665

132. Phuong Vu, Cray Research, Inc., 655F Lone Oak Drive, Eagan, MN 55121

133. Mary F. Wheeler, Rice University, Department of Mathematical Sciences, P.O. Box 1892, Houston, TX 77251

134. Andrew B. White, Computing Division, Los Alamos National Laboratory, P.O. Box 1663, MS-265, Los Alamos, NM 87545

135. David Young, University of Texas, Center for Numerical Analysis, RLM 13.150, Austin, TX 78731

136. Office of Assistant Manager for Energy Research and Development, U.S. Department of Energy, Oak Ridge Operations Office, P.O. Box 2001 Oak Ridge, TN 37831-8600

137–146. Office of Scientific & Technical Information, P.O. Box 62, Oak Ridge, TN 37831