Proceedings, Cray Users Group Conference, CUG 2005, May 16-19, Albuquerque, NM USA.
Extended version of the paper "R. Rabenseifner et al., Network Bandwidth Measurements and Ratio Analysis with the HPC Challenge Benchmark Suite (HPCC)" at EuroPVM/MPI 2005, Sep. 18-21, Sorrento, Italy, LNCS, Springer-Verlag, 2005.

Balance of HPC Systems Based on HPCC Benchmark Results

Rolf Rabenseifner

High-Performance Computing-Center (HLRS), University of Stuttgart,
Allmandring 30, D-70550 Stuttgart, Germany,
rabenseifner@hlrs.de, www.hlrs.de/people/rabenseifner/

Abstract. Based on results reported by the HPC Challenge benchmark suite (HPCC), the balance between computational speed, communication bandwidth, and memory bandwidth is analyzed for HPC systems from Cray, NEC, IBM, and other vendors, and clusters with various network interconnects. Strength and weakness of the communication interconnect is examined for three communication patterns. The HPCC suite was released to analyze the performance of high-performance computing architectures using several kernels to measure different memory and hardware access patterns comprising latency based measurements, memory streaming, inter-process communication and floating point computation. HPCC defines a set of benchmarks augmenting the High Performance Linpack used in the Top500 list. This paper describes the inter-process communication benchmarks of this suite. Based on the effective bandwidth benchmark, a special parallel random and natural ring communication benchmark has been developed for HPCC. Ping-Pong benchmarks on a set of process pairs can be used for further characterization of a system. This paper analyzes first results achieved with HPCC. The focus of this paper is on the balance between computational speed, memory bandwidth, and inter-node communication.

Keywords. HPCC, network bandwidth, effective bandwidth, Linpack, HPL, STREAM, DGEMM, PTRANS, FFTE, latency, benchmarking.

1 Introduction and Related Work

The HPC Challenge benchmark suite (HPCC) [5, 6] was designed to provide benchmark kernels that examine different aspects of the execution of real applications. The first aspect is benchmarking the system with different combinations of high and low temporal and spatial locality of the memory access. HPL (High Performance Linpack) [4], DGEMM [2, 3] PTRANS (parallel matrix transpose) [8], STREAM [1], FFTE (Fast Fourier Transform) [11], and RandomAccess are dedicated to this task. Other aspects are measuring basic parameters like achievable computational performance (again HPL), the bandwidth of the memory access (STREAM copy or triad), and latency and bandwidth of the inter-process communication based on ping-pong benchmarks and on parallel effective bandwidth benchmarks [7, 9].

This paper describes in Section 2 the latency and bandwidth benchmarks used in the HPCC suite. Section 3 analyzes bandwidth and latency measurements submitted to the HPCC web interface [5]. In Section 4, the ratio between computational performance, memory and inter-process bandwidth is analyzed to compare system architectures (and not only specific systems). In Section 5, the ratio analysis is extended to the whole set of benchmarks to compare the largest systems in the list and also different network types.

2 Latency and Bandwidth Benchmark

The latency and bandwidth benchmark measures two different communication patterns. First, it measures the single-process-pair latency and bandwidth, and second, it measures the parallel all-processes-in-a-ring latency and bandwidth.

For the first pattern, ping-pong communication is used on a pair of processes. Several different pairs of processes are used and the maximal latency and minimal bandwidth over all pairs is reported. While the ping-pong benchmark is executed on one process pair, all other processes are waiting in a blocking receive. To limit the total benchmark time used for this first pattern to 30 sec, only a subset of the set of possible pairs is used. The communication is implemented with MPI standard blocking send and receive.

In the second pattern, all processes are arranged in a ring topology and each process sends and receives a message from its left and its right neighbor in parallel. Two types of rings are reported: a naturally ordered ring (i.e., ordered by the process ranks in MPI_COMM_WORLD), and the geometric mean of the bandwidth of ten different randomly chosen process orderings in the ring. The communication is implemented (a) with MPI standard non-blocking receive and send, and (b) with two calls to MPI_Sendrecv for both directions in the ring. Always the fastest of both measurements are used. For latency or bandwidth measurement, each ring measurement is repeated 8 or 3 times – and for random ring with different patterns – and only the best result is chosen. With this type of parallel communication, the bandwidth per process is defined as total amount of message data divided by the number of processes and the maximal time needed in all processes. The latency is defined as the maximum time needed in all processes divided by the number of calls to MPI_Sendrecv (or MPI_Isend) in each process. This definition is similar to the definition with ping-pong, where the time is measured for the sequence of a *send* and a *recv*, and again *send* and *recv*, and then divided by 2. In the ring benchmark, the same pattern is done by all processes instead of a pair of processes. This benchmark is based on patterns studied in the effective bandwidth communication benchmark [7, 9].

For benchmarking latency and bandwidth, 8 byte and 2,000,000 byte long messages are used. The major results reported by this benchmark are:

- maximal ping pong latency,
- average latency of parallel communication in randomly ordered rings,
- minimal ping pong bandwidth,
- bandwidth per process in the naturally ordered ring,

- average bandwidth per process in randomly ordered rings.

Additionally reported values are the latency of the naturally ordered ring, and the remaining values in the set of minimum, maximum, and average of the ping-pong latency and bandwidth.

Especially the ring based benchmarks try to model the communication behavior of multi-dimensional domain-decomposition applications. The natural ring is similar to the message transfer pattern of a regular grid based application, but only in the first dimension (adequate ranking of the processes is assumed). The random ring fits to the other dimensions and to the communication pattern of unstructured grid based applications. The random ring and the bi-directional bi-section bandwidth benchmarks should report similar results because both are sending and receiving messages in parallel on each process between a bi-section (in the case of the ring, the group of the even and of the odd ranks form the two sections). Therefore, the following analysis is mainly focused on the random ring bandwidth.

3 Analysis of HPCC uploads

Fig. 1 is based on base-run uploads to the HPCC web-page. Therefore, the quality of the benchmarking, i.e., choosing the best compiler options and benchmark parameters was done by the independent institutions that submitted results. The authors have added two results from the NEC SX-6+ and some results for fewer number of processes on NEC SX-8 and Cray XT3 [12]. For IBM BlueGene, an additional optimized measurement is also shown in some of the figures. The measurements are sorted by the random ring bandwidth, except that all measurements belonging to some platform or network type are kept together at the position of their best bandwidth.

The diagram consists of three bands: 1) the ping-pong and random ring latencies, 2) the minimal ping-pong, natural ring, and random ring bandwidth-bars together with a background curve showing the accumulated Linpack (HPL) performance, and 3) the ratios *natural ring* to *ping-pong*, *random ring* to *ping-pong*, and additionally *random ring* to *natural ring.*

The systems on the upper part of the figure have a random ring bandwidth less than 300 MB/s, the systems on the lower part are between 400 MB/s and 1.5 GB/s. Concentrating on the lower part, one can see that all systems show a degradation for larger CPU counts. Cray and NEC systems are clusters of SMP nodes. The random ring bandwidth benchmark uses mainly inter-node connections whereas the natural ring bandwidth uses only one inter-node connection in both directions and all other connections are inside of the SMP nodes. Therefore one can see a significant difference between the random ring and the natural ring bandwidth. One exception is the multi-threaded measurement on a NEC SX-6+ (0.5625 GHz); here, all three bandwidth values are nearly equivalent because on each SMP, only one MPI process is running. The ratio natural ring to ping-ping bandwidth varies between 0.4 and 1.0, random ring to ping-pong between 0.1 and 0.45, and random to natural ring between 0.1 and 0.7. With the IBM High

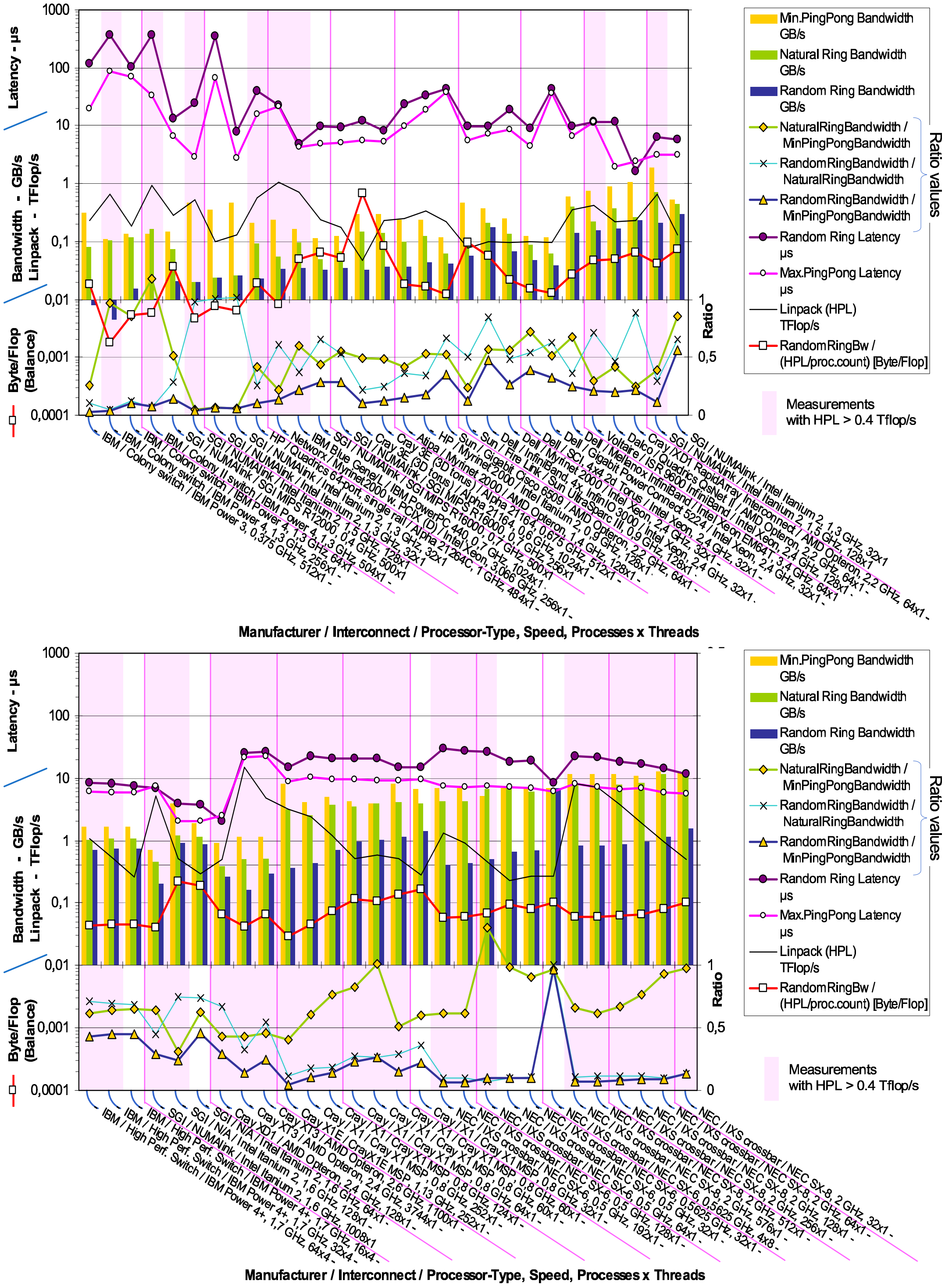


Fig. 1. Base runs of the HPC Challenge bandwidth and latency benchmarks, Status June 27, 2005.

Switch	CPU	Proc. Speed GHz	Number of MPI processes x threads	**Random Ring Bandw. GB/s**	Ping-Pong Bandw. GB/s	Rand. Ring Lat. µs	Ping-Pong Lat. µs	**HPL Linpack** accumulated Gflop/s	**per process Gflop/s**	**Balance: Communi./ Comput. byte/kflop**
IBM Colony	IBM Power 4	1.3	256x1	**0.0046**	0.108	374	87	654	**2.55**	**1.8**
Quadrics switch	Alpha 21264B	1.0	484x1	**0.023**	0.280	40	16	618	**1.28**	**17.8**
Myrinet 2000	Intel Xeon 3	3.066	256x1	**0.032**	0.241	22	22	1030	**4.02**	**8.1**
Sun Fire Link	Ultra Sparc III	0.9	128x1	**0.056**	0.468	9	5	75	**0.59**	**94.5**
Infiniband	Intel Xeon	2.46	128x1	**0.156**	0.738	12	12	413	**3.23**	**48.2**
SGI Numalink	Intel Itanium 2	1.56	128x1	**0.211**	1.8	6	3	639	**4.99**	**42.2**
SGI Altix 3700 Bx2	Intel Itanium 2	1.6	128x1	**0.897**	3.8	4	2	521	**4.07**	**220.**
Infiniband, 4x, InfinIO 3000	Intel Xeon	2.4	32x1	**0.178**	0.374	10	7	101	**3.17**	**56.3**
Myrinet 2000	Intel Xeon	2.4	32x1	**0.066**	0.245	19	9	97	**3.03**	**21.7**
SCI, 4x4 2d Torus	Intel Xeon	2.4	32x1	**0.048**	0.121	9	4	100	**3.13**	**15.2**
Gigabit Ethernet, PowerConnect 5224	Intel Xeon	2.4	32x1	**0.038**	0.117	42	37	97	**3.02**	**12.5**
NEC SX-6 IXS	NEC SX-6	0.5	192x1	**0.398**	6.8	30	7	1327	**6.91**	**57.5**
NEC SX-6 IXS	NEC SX-6	0.5	128x1	**0.429**	6.9	27	7	905	**7.07**	**60.7**
NEC SX-6 IXS	NEC SX-6	0.5	64x1	**0.487**	5.2	26	7	457	**7.14**	**68.1**
NEC SX-6 IXS	NEC SX-6	0.5	32x1	**0.661**	6.9	18	7	228	**7.14**	**92.6**
NEC SX-6+ IXS	NEC SX-6+	0.5625	32x1	**0.672**	6.8	19	7	268	**8.37**	**80.3**
NEC SX-6+ IXS$^+$)	NEC SX-6+	0.5625	4x8	**6.759**	7.0	8	6	(268)	**(66.96)**	**(100.9)**
IBM HPS	IBM Power 4+	1.7	64x4	**0.724**	1.7	8	6	1074	**16.79**	**43.1**
IBM HPS	IBM Power 4+	1.7	32x4	**0.747**	1.7	8	6	532	**16.62**	**45.0**
Cray X1	Cray X1 MSP	0.8	252x1	**0.429**	4.0	22	10	2385	**9.46**	**45.3**
Cray X1	Cray X1 MSP	0.8	124x1	**0.709**	4.9	20	10	1205	**9.72**	**72.9**
Cray X1	Cray X1 MSP	0.8	120x1	**0.830**	3.7	20	10	1061	**8.84**	**93.9**
Cray X1	Cray X1 MSP	0.8	64x1	**0.941**	4.2	20	9	522	**8.15**	**115.4**
Cray X1	Cray X1 MSP	0.8	60x1	**1.033**	3.9	21	9	578	**9.63**	**107.3**

Table 1. Comparison of bandwidth and latency on HPCC entries with more than 0.4 Tflop/s with three exceptions: For SGI Numalink, only MPT 1.10 values are shown, the older MPT 1.8-1 values are omitted, and for Sun Fire and NEC SX-6, smaller systems are reported because on larger systems, HPCC results are not yet available, and the Dell Xeon cluster is included for network comparison. Note, that each thread is running on a *CPU*, but the communication and the second HPL value are measured with *MPI processes*.
$^+$) This row is based on an additional measurement with the communication benchmark software. The HPL value of this row is taken from the previous row because there isn't a benchmark value available and significant differences between single- and multi-threaded HPL execution are not expected. The last two columns are based on this HPL value.

Performance Switch (HPS), the reported random ring bandwidth values (0.72-0.75 GB/s) are nearly independent from the number of processes (64 to 256 [1.07 Tflop/s]), while the Cray X1 shows a degradation from 1.03 GB/s with 60 MSPs (0.58 Tflop/s) to 0,43 GB/s with 252 MSPs (2.38 Tflop/s). For some systems, the random ring latency and performance is summarized in Tab. 1.

For the bandwidth values, the achievable percentage on the random ring from the ping-pong varies between 4 % and 48 % with one exception: If only one (but multi-threaded) MPI process is running on each SMP node of a NEC SX-6+, random ring and ping-pong bandwidth are nearly the same. For the latency values, the ratio ping-pong to random varies between 0.23 and 0.99. On only a few systems, the ping-pong latency and the random ring latency are similar (e.g., on Infiniband, IBM HPS, NEC SX-6+ multithreaded).

These examples not only show the communication performance of different network types, but also that the ping-pong values are not enough for a comparison. The ring based benchmark results are needed to analyze these interconnects.

4 Balance of Communication to Computation

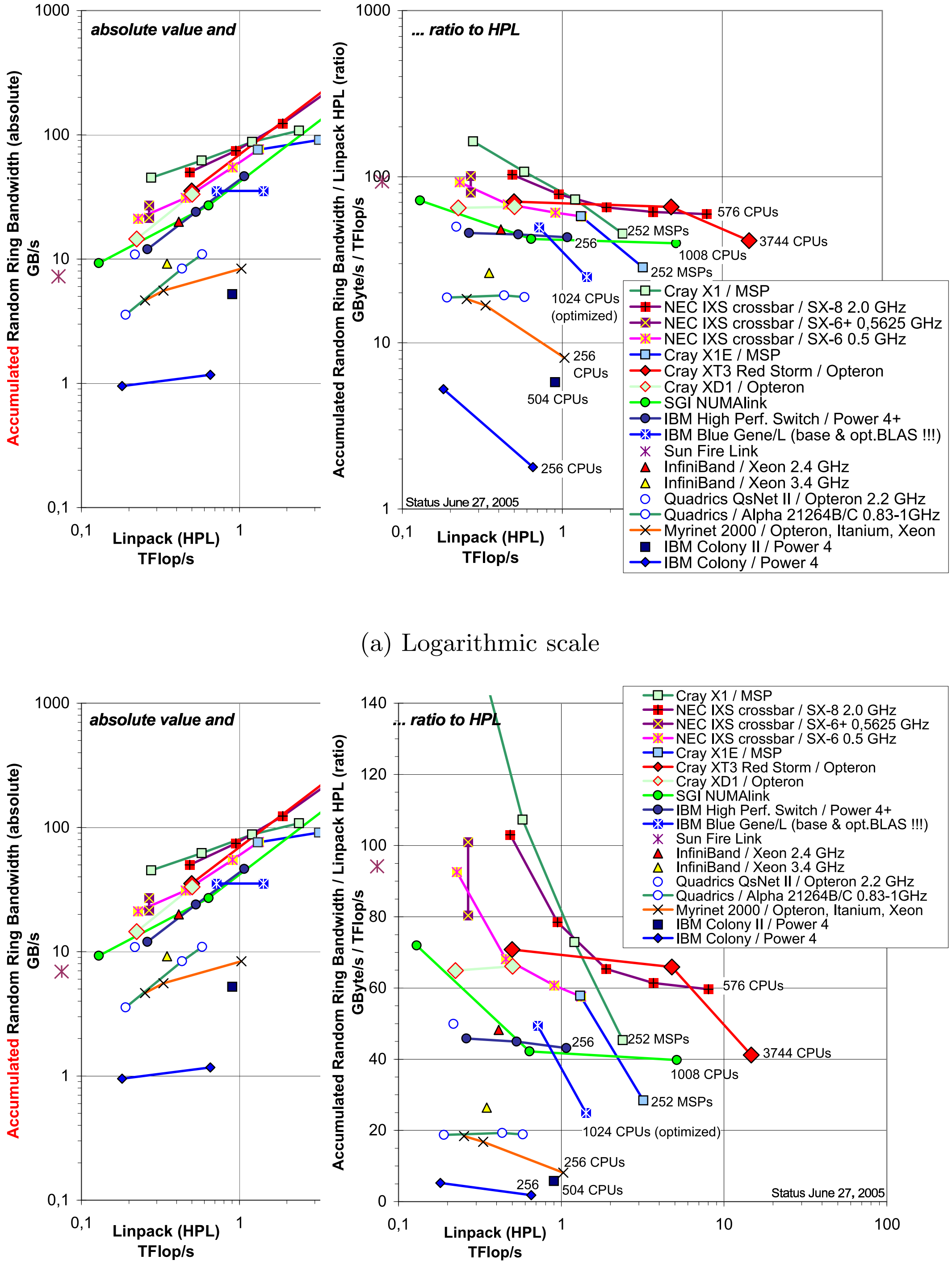
For multi-purpose HPC systems, the balance of processor speed, along with memory, communication, and I/O bandwidth is important. In this section, we analyze the ratio of inter-node communication bandwidth to the computational speed. To characterize the communication bandwidth between SMP nodes, we use the random ring bandwidth, because for a large number of SMP nodes, most MPI processes will communicate with MPI processes on other SMP nodes. This means, with 8 or more SMP nodes, the random ring bandwidth reports the available inter-node communication bandwidth per MPI process. To characterize the computational speed, we use the HPL Linpack benchmark value divided by the number of MPI processes, because HPL can achieve nearly peak on cache-based and on vector systems, and with single- and multi-threaded execution. The ratio of the random ring bandwidth to the HPL divided by the MPI process count expresses the communication-computation balance in byte/flop (see in Fig. 1) or byte/kflop (used in Tab. 1).

Although the balance is calculated based on MPI processes, its value should be in principle independent of the programming model, i.e., whether each SMP node is used with several single-threaded MPI processes, or some (or one) multi-threaded MPI processes, as long as the number of MPI processes on each SMP node is large enough that they altogether are able to saturate the inter-node network [10].

Table 1 shows that the balance is quite different. Currently, the HPCC table lacks of the information, how many network adapters are used on each SMP nodes, i.e., the balance may be different if a system is measured with exactly the same interconnect and processors but with a smaller or larger amount of network adapters per SMP node.

On the reported installations, the balance values start with 1.8 / 8.1 / 17.8 B/kflop on IBM Colony, Myrinet 2000 and Quadrics respectively. SGI Numalink, IBM High Performance Switch, Infiniband, and the largest Cray X1 configuration have a balance between 40 and 50 B/kflop. High balance values are observed on Cray XD1, Sun Fire Link (but only with 0.59 Gflops per MPI process), NEC SX-6 and on Cray X1 and X1E. The best values for large systems are for Cray XT3 and NEC SX-8 (see also Fig. 2).

For NEC SX-6, the two different programming models *single-* and *multi-threaded* execution were used. With the single-threaded execution, 25 % of the random ring connections involve only intra-node communications. Therefore only 0.504 GB/s (75 % from 0.672 GB/s) represent the inter-node communication bandwidth per CPU. The inter-node bandwidth per node (with 8 CPUs) is therefore 4.02 GB/s respectively. The balance of inter-node communication to computation is characterized by the reduced value 60.2 byte/kflop. With multi-threaded execution, all communication is done by the master-threads and is inter-node communication. Therefore, the random ring bandwidth is measured per node. It is significantly better with the multi-threaded application programming scheme (6.759 GB/s) than with single-threaded (4.02 GB/s). Implications on optimal programming models are discussed in [10].



(a) Logarithmic scale

(b) Linear scale (right diagram)

Fig. 2. Accumulated random ring bandwidth versus HPL Linpack performance.

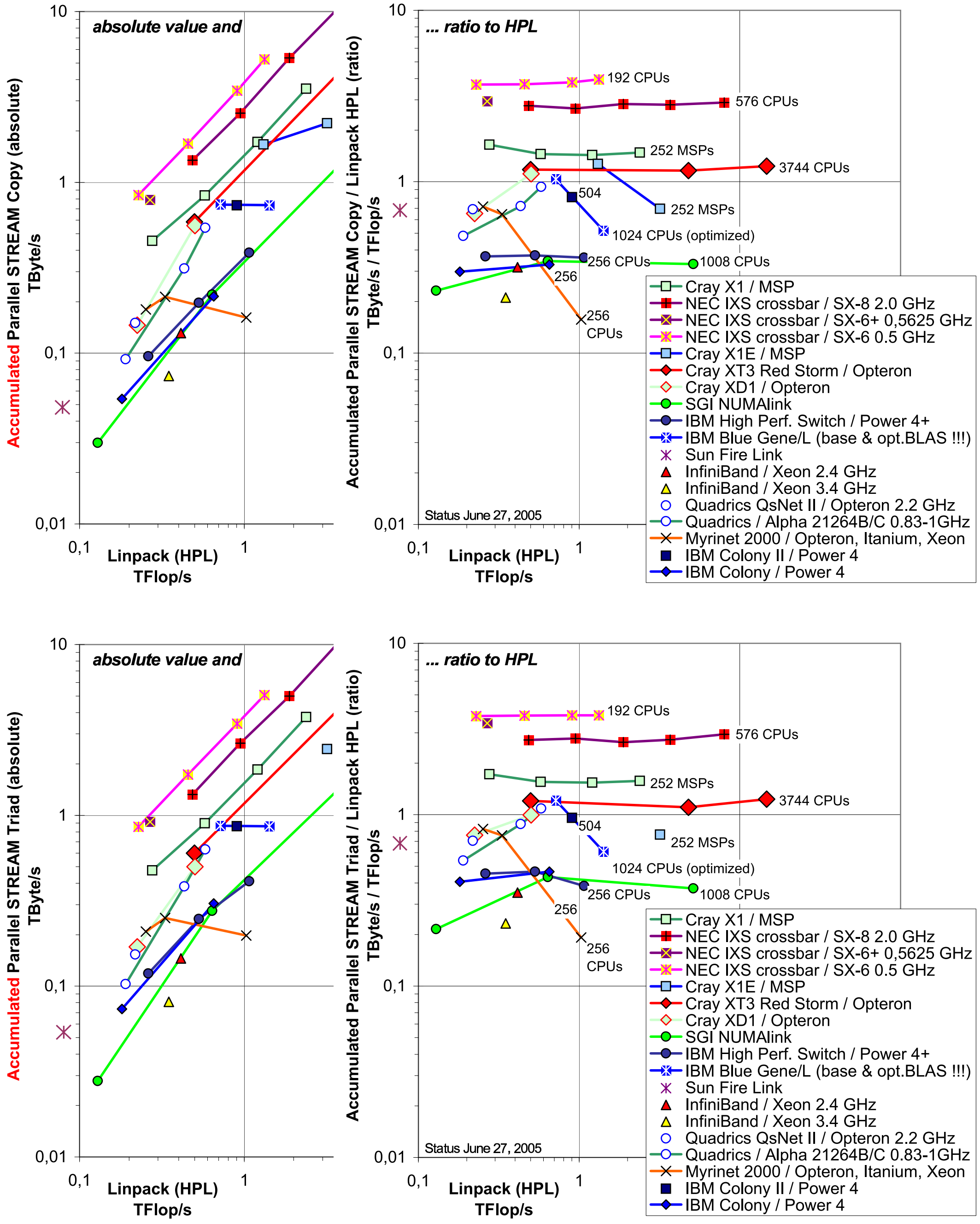


Fig. 3. Accumulated stream copy and triad bandwidth versus HPL Linpack performance.

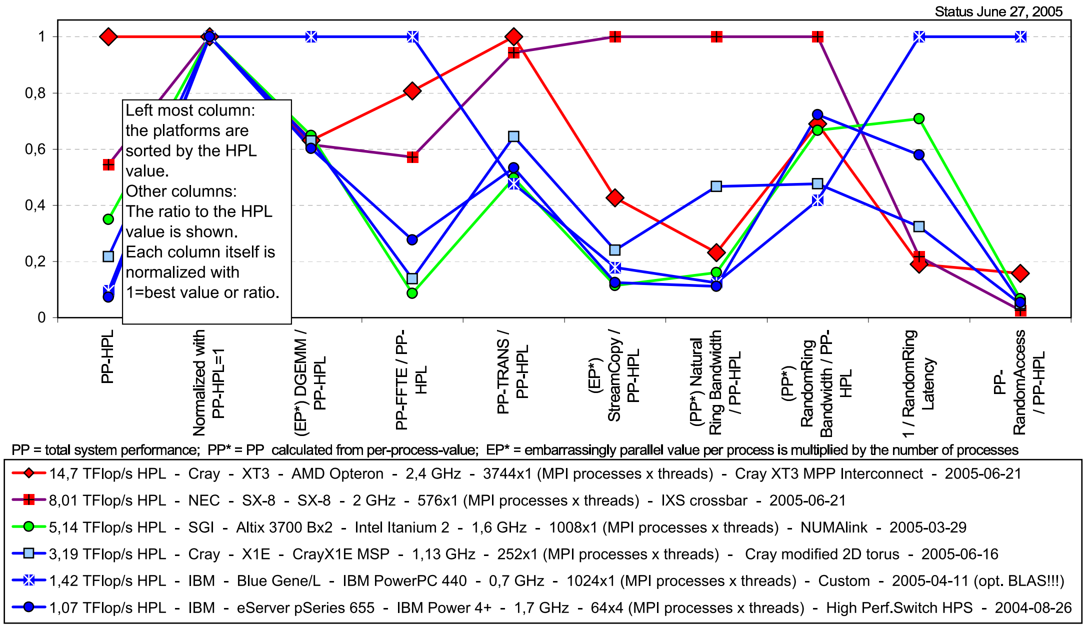


Fig. 4. Comparing the largest clusters in the HPCC list. Each system is normalized with its HPL value.

Fig. 2 shows the scaling of the accumulated random ring performance with the computational speed. For this, the HPCC random ring bandwidth was multiplied with the number of MPI processes. The computational speed is benchmarked with HPL. The left diagram shows absolute communication bandwidth, whereas the right diagram plots the ratio of communication to computation speed. Better scaling with the size of the system is expressed by horizontal or a less decreasing ratio curve. E.g., the Cray X1 and X1E curves show a stronger decrease than the NEC SX-6 or SX-8. Interpolation at 3 TFlop/s gives a ratio of 30 B/kflop on Cray X1E, 40 B/kflop on SGI Altix 700 Bx2, 62 B/kflop on NEC SX-8, and 67 B/kflop on Cray XT3.

5 Ratio-based analysis of all benchmarks

Fig. 3 compares the memory bandwidth with the computational speed analog to Fig. 2. The accumulated memory bandwidth is calculated as the product of the number of MPI processes with the embarrassingly parallel STREAM copy and triad HPCC result. There is a factor of about 100 between the best and the worst random ring ratio values in Fig. 2, but only a factor of 25 with the memory bandwidth rations in Fig. 3 (right diagram). But looking at the systems with the best memory and network scaling the differences in the memory scaling are more significant. E.g., while NEC SX-8 and Cray XT3 have both shown best network bandwidth ratios, here, NEC SX-8 provides 2.4 times more memory bandwidth

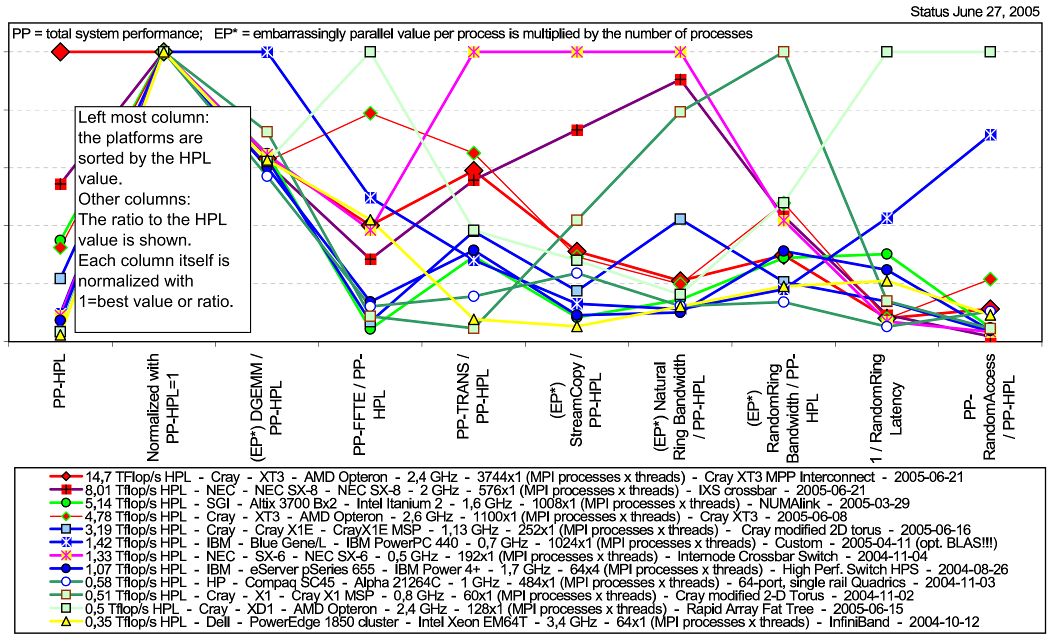


Fig. 5. Comparing additional clusters in the HPCC list. Each system is normalized with its HPL value.

per Tflop/s than the Cray XT3. The CPU counts also indicate that different numbers of CPUs are needed to achieve similar computational speed.

Figures 4–6 are comparing the systems based on several HPCC benchmarks. This analysis is similar to the current Kiviat diagram analysis on the HPCC web page [5], but it uses always embarrassingly parallel benchmark results instead of single process results, and it uses only accumulated global system values instead of per process values. If one wants to compare the balance of systems with quite different total system performance, this comparison can be done hardly on the basis of absolute performance numbers. Therefore in Fig. 4–6, all benchmark results (except of latency values) are normalized with the HPL system performance, i.e., divided by the HPL value. Only the left column can be used to compare the absolute performance of the systems. This normalization is also indicated by normalized HPL value in the second column that is per definition always 1. Each column itself is additionally divided by largest value in the column, i.e., the best value is always 1. The columns are sorted together to show influences: HPL and DGEMM are reporting performance with high temporal and spatial locality. FFT has a low spatial locality, and PTRANS a low temporal locality. FFT and PTRANS are strongly influenced by the memory bandwidth benchmark (EP STREAM copy) and the inter-process bandwidth benchmark (random ring). The two right-most columns are latency based: The reciprocal value of the random ring inter-process latency, and the Random Access benchmark ratio. Fig. 4 compares systems with more than 1 Tflop/s, Fig. 5 compares

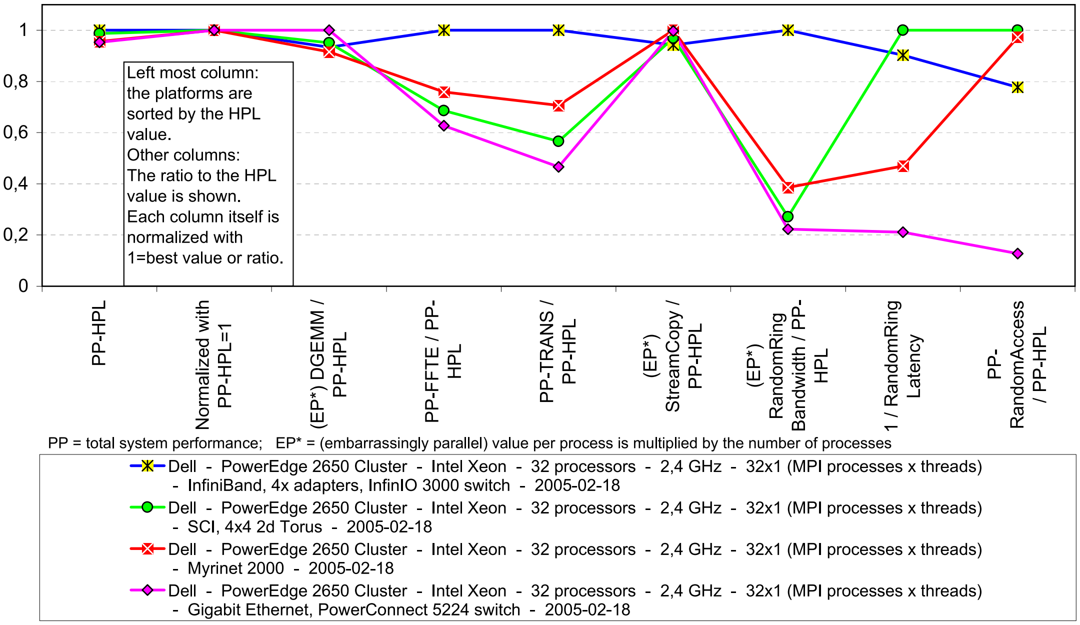


Fig. 6. Comparing different node interconnects.

a larger set of HPC systems. One must be aware that the optimal Cray XD1 (FFT) and Cray X1 (random ring) values need not to scale (see Fig. 7), resp., does not scale (see Fig. 2) up to the range of the presented multi-Tflop/s systems. Fig. 6 analyzes the four different networks on a Dell Intel Xeon cluster.

In Fig. 7, the Global Fast Fourier Transformation (FFTE) shows best results on Cray XD1, IBM BlueGene/L (base-run) and Cray XT3, followed by the vector systems from NEC.

Fig. 8 compares the accumulated natural ring bandwidth of different platforms in relation to their HPL values. The accumulated natural ring bandwidth depends strongly on the usage pattern: The two NEC SX-6+ measurements are done with different number of threads and MPI processes on each SMP node. The observed results are quite different because there is a high additional intra-node communication fraction if each CPU on an SMP node is running an independent MPI process, whereas with a multi-threaded execution, only inter-node communication is reported. Therefore, if one wants to know the approximately bidirectional bisection bandwidth of a system, one can only use the random ring bandwidth benchmark (and not the natural ring). The random ring and a bidirectional bisection bandwidth benchmarks perform similar communication patterns: Each process sends and receives a message at the same time, either in a ring, or in pairs of processes.

Fig. 9 presents the ratios between a global matrix transpose and HPL. This benchmark uses both intra-node and inter-node communication on clusters of SMP nodes and therefore results should be similar to a combination of the natural ring and random ring bandwidth measurements.

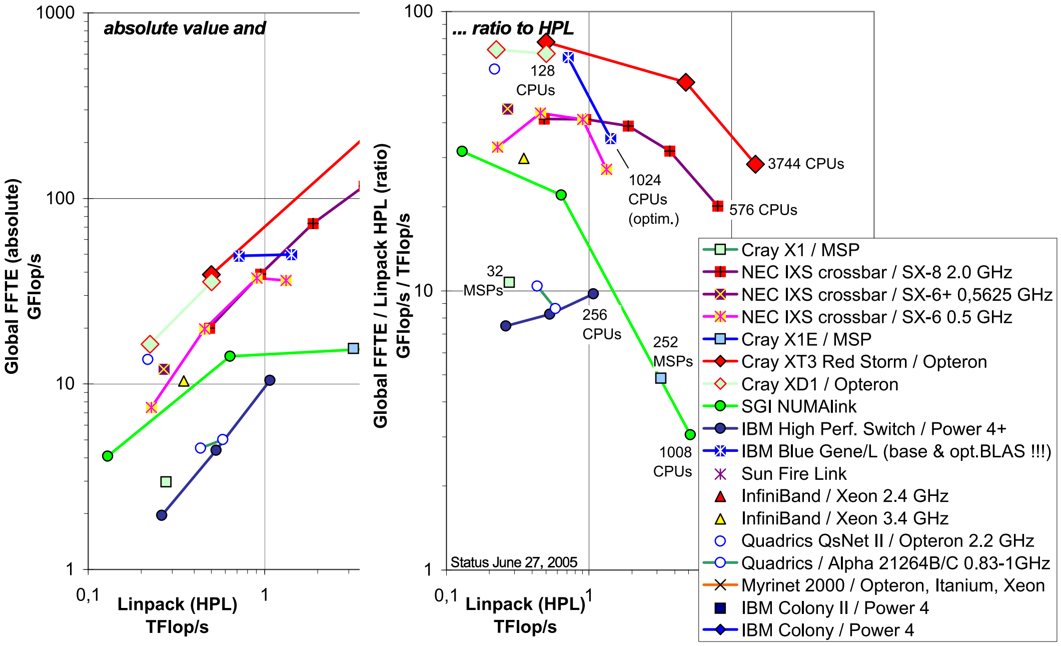


Fig. 7. Global Fast Fourier Transform versus HPL Linpack performance.

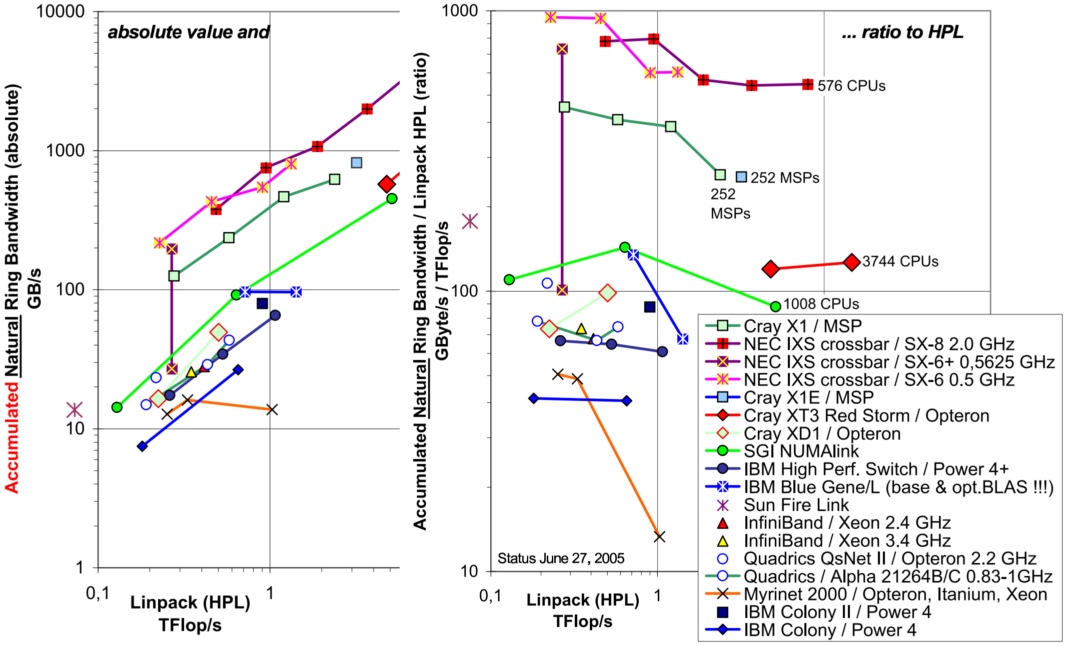


Fig. 8. Accumulated natural ring bandwidth versus HPL Linpack performance.

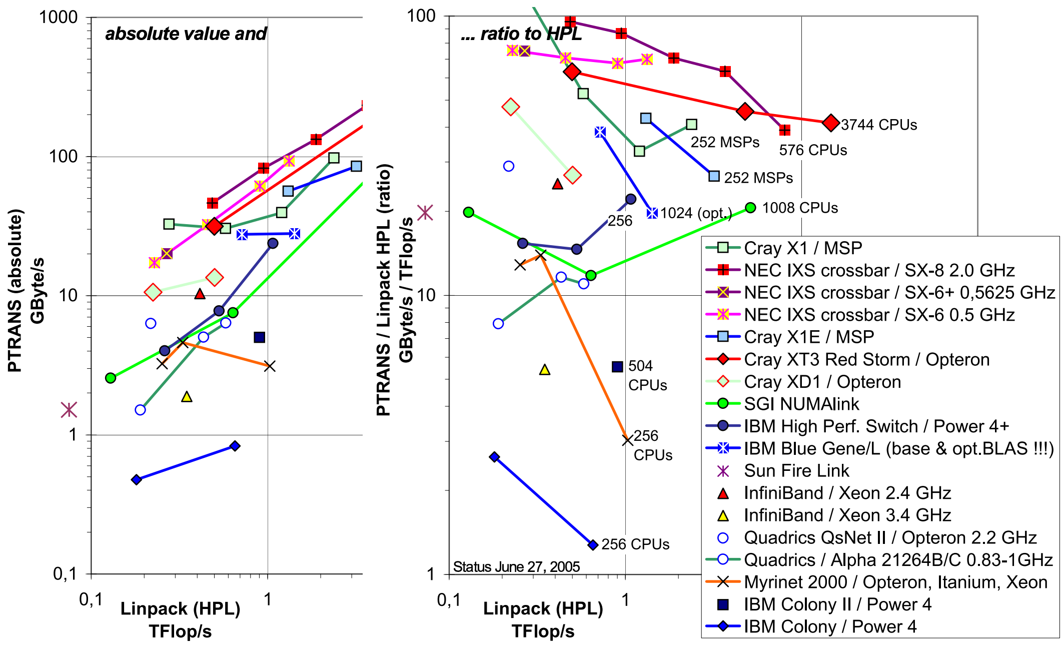


Fig. 9. PTRANS bandwidth versus HPL Linpack performance.

6 Conclusions

The HPC Challenge benchmark suite and the uploaded results from many HPC systems are a good basis for comparing the balance between computational speed, inter-process communication and memory bandwidth. The figures presented in this paper clearly show the strengths and weaknesses of various systems. One can see that several systems provide a similar balance between computational speed, network bandwidth and memory bandwidth, although hardware architectures vary between MPP concepts (e.g., Cray XT3, IBM BlueGene/L), clusters of vector SMP nodes (NEC SX-8, Cray X1E), constellations (IBM), and ccNUMA architectures (SGI). One can also see that the gap between best and worst balance ratios is more than 25. The number of CPUs needed to achieve similar accumulated performance and network and memory bandwidth is also quite different. Especially the curves in Figures 2–3 and 7–9 can be used for interpolation and to some extent also for extrapolation.

Outside of the scope of the HPCC database is the price-performance ratio. In this paper, most scaling was done on the basis of the HPL system performance. In a procurement, relating the performance data additionally to real costs will give additional hints on pros and cons of the systems.

7 Acknowledgments

The author would like to thank all persons and institutions that have uploaded data to the HPCC database, Jack Dongarra and Piotr Luszczek, for the invitation to Rolf Rabenseifner to include his effective bandwidth benchmark into the HPCC suite, Holger Berger, Sunil Tiyyagura and Nathan Wichmann for the HPCC results on the NEC SX-6+/SX-8 and Cray XT3 and helpful discussions on the HPCC analysis, David Koester for his helpful remarks on the HPCC Kiviat diagrams, and Gerrit Schulz and Michael Speck, student co-workers, who have implemented parts of the software.

References

1. John McCalpin. *STREAM: Sustainable Memory Bandwidth in High Performance Computing.* (`http://www.cs.virginia.edu/stream/`)
2. Jack J. Dongarra, Jeremy Du Croz, Sven Hammarling, Iain S. Duff: A set of level 3 basic linear algebra subprograms. *ACM Transactions on Mathematical Software (TOMS)*, 16(1):1–17, March 1990.
3. Jack J. Dongarra, Jeremy Du Croz, Sven Hammarling, Iain S. Duff: Algorithm 679; a set of level 3 basic linear algebra subprograms: model implementation and test programs. *ACM Transactions on Mathematical Software (TOMS)*, 16(1):18–28, March 1990.
4. Jack J. Dongarra, Piotr Luszczek, and Antoine Petitet: The LINPACK benchmark: Past, present, and future. *Concurrency nd Computation: Practice and Experience*, 15:1–18, 2003.
5. Jack Dongarra and Piotr Luszczek: *Introduction to the HPCChallenge Benchmark Suite.* Computer Science Department Tech Report 2005, UT-CS-05-544. (`http://icl.cs.utk.edu/hpcc/`).
6. Panel on HPC Challenge Benchmarks: An Expanded View of High End Computers. SC2004 November 12, 2004 (`http://www.netlib.org/utk/people/JackDongarra/SLIDES/hpcc-sc2004-panel.htm`).
7. Alice E. Koniges, Rolf Rabenseifner and Karl Solchenbach: *Benchmark Design for Characterization of Balanced High-Performance Architectures.* In IEEE Computer Society Press, proceedings of the 15th International Parallel and Distributed Processing Symposium (IPDPS'01), Workshop on Massively Parallel Processing (WMPP), April 23-27, 2001, San Francisco, USA, Vol. 3. In IEEE Computer Society Press (`http://www.computer.org/proceedings/`).
8. Parallel Kernels and Benchmarks (PARKBENCH) (`http://www.netlib.org/parkbench/`)
9. Rolf Rabenseifner and Alice E. Koniges: *Effective Communication and File-I/O Bandwidth Benchmarks.* In J. Dongarra and Yiannis Cotronis (Eds.), Recent Advances in Parallel Virtual Machine and Message Passing Interface, proceedings of the 8th European PVM/MPI Users' Group Meeting, EuroPVM/MPI 2001, Sep. 23-26. Santorini, Greece, pp 24-35.
10. Rolf Rabenseifner: *Hybrid Parallel Programming on HPC Platforms.* In proceedings of the Fifth European Workshop on OpenMP, EWOMP '03, Aachen, Germany, Sept. 22-26, 2003, pp 185-194
11. Daisuke Takahashi, Yasumasa Kanada: High-Performance Radix-2, 3 and 5 Parallel 1-D Complex FFT Algorithms for Distributed-Memory Parallel Computers. *Journal of Supercomputing*, 15(2):207–228, Feb. 2000.
12. Nathan Wichmann: *Cray and HPCC: Benchmark Developments and Results from Past Year.* Proceedings of CUG 2005, May 16-19, Albuquerque, NM, USA.