ETH zürich

ETH Library

Cost/performance tradeoffs in network interconnects for clusters of commodity PCs

Report

Author(s):
Kurmann, Christian; Rauch, Felix; Stricker, Thomas M.

Publication date:
2003

Permanent link:
https://doi.org/10.3929/ethz-a-006665816

Rights / license:
In Copyright - Non-Commercial Use Permitted

Originally published in:
CS technical report 391

This page was generated automatically upon download from the ETH Zurich Research Collection.
For more information, please consult the Terms of use.

Cost/Performance Tradeoffs in Network Interconnects for Clusters of Commodity PCs

Christian Kurmann Felix Rauch Thomas M. Stricker
Laboratory for Computer Systems
ETH - Swiss Institute of Technology
CH-8092 Zürich
Switzerland
{kurmann,rauch,tomstr}@inf.ethz.ch

CS Technical Report #391

This is an extension of a similar paper from
Workshop on Communication Architecture for Clusters (CAC '03)
in conjunction with
International Parallel and Distributed Processing Symposium (IPDPS),
April 2003, Nice, France.

Abstract

The definition of a commodity component is quite obvious when it comes to the PC as a basic compute engine and building block for clusters of PCs. Looking at the options for a more or less performant interconnect between those compute nodes it is much less obvious which interconnect still qualifies as commodity and which not. We are trying to answer this question based on an in-depth analysis of a few common more or less expensive interconnects on the market. Our measurements and observations are based on the experience of architecting, procuring and installing Xibalba, a 128 node - 192 processor versatile cluster for a variety of research applications in the CS department of ETH Zurich.

We define our unique way to measure the performance of an interconnect and use our performance characterization to find the best cost performance point for networks in PC clusters. Since our work is tied to the purchase of a machine at fair market value we can also reliably comment on cost performance of the four types of interconnects we considered. We analyze the reason for performance and non-performance for different Fast Ethernet architectures with a set of micro-benchmarks and conclude our study with performance numbers of some applications. Thus, the reader gets an idea about the impact of the interconnect on the overall application performance in commodity PC clusters.

Keywords: Clusters of commodity PC, Ethernet, Myrinet, switch performance, application performance, full bisection bandwidth, all-to-all communication.

1 Introduction

1.1 PC Clusters Built from Commodity Components

Several authors have pointed out the architectural principle for constructing high performance systems out of widely available commodity components about a decade ago. The literature on the Beowulf [1], the Hyglac and Loki parallel workstations projects provide a good overview on the topic and an almost complete list of credits to these early projects is given in [17].

Microprocessor based computer systems leverage from a high volume to be competitive in computational speed and price. Such volumes can only be sustained if the node architectures are similar to the architecture of PCs and workstations. It remains an open question of whether this law of commoditization also holds for cluster interconnects. So far the prediction of a commodity one-for-all-needs network has not quite materialized and the market is still split between regular Ethernet and a few dedicated high performance interconnects. Initially the first networking technology expected to become the universal standard was ATM/Sonet, but at this time Ethernets using the TCP/IP protocols seem to be a more viable candidate for the role of the universal interconnect.

PC Clusters are the successors to massively parallel computers. Networks for massively parallel computers are a well researched topic. It would be well beyond the scope of this paper to give a complete survey, but the two fundamental approaches can be mentioned easily:

- Networks for parallel computers should be scalable to a large number of nodes and should provide full bisection bandwidth across any arbitrary bisection of the parallel machine. As a tradeoff the performance of a single link in such a network could be a secondary concern. The best example of such a network is the fat tree used in the Thinking Machines CM-5 [11].
- Networks for parallel computers should be designed around a sophisticated tradeoff of technology factors (i.e. best possible pin counts, clock speeds) and the links should be as fast as possible, allowing only

simple networks like tori or hierarchical rings. Representatives of this line of research are the Cosmic cube project [5] or the Hector project [19].

Still after many research papers dedicated to this topic, it appears to us that the essential question is still open and needs to be re-addressed in the light of commodity clusters incorporating the technology factors of commodity cluster interconnects.

In related studies, [8] compares different networking technologies for parallel computing, focusing on system software aspects that balance the network load on different local area networks used in parallel. The requirements for a high performance compute cluster for a successful integration into a larger scale computational grid is nicely described in [15]. Some interconnects and protocols are analyzed, but the work focuses more on the communication of two single nodes within and outside the cluster, rather than traffic patterns requiring full bisection bandwidth. A communication cost model is presented in [12], characterizing the key communication resources for parallel applications in high performance networks of workstations. After a close examination of our networking architecture and our network model the performance results of this study could lead directly to the determination of their "gap" and "bulk gap" parameters for performance predictions of algorithms, whose communication system response can be defined as LogP model parameters.

In addition to the popular PC clusters there are several new platforms for wide area distributed computing. Those platforms use the regular Internet as an interconnect between the compute nodes and are therefore rather limited to embarrassingly parallel tasks at this time.

The difference between widely distributed and cluster computing is clearly in the design and the implementation of an interconnect network. In both cases standard networking technologies can be used, but in the first case of widely distributed or grid based computing the networking resources must be taken as they are and explored [7] while in the latter case of clusters the network is designed and managed at a certain capacity.

The rest of our workshop contribution is organized as follows: In Section 3 we show how to build a full bisection cluster network with Fast Ethernet using commodity networking equipment and show why this is difficult. Section 4 explains our evaluation principle and discusses how to read the performance results. In Section 5 we attempt to characterize a fairly expensive central switch that was said to provide full bisection bandwidth but did hopelessly fall short of our expectations. After a presentation of the benchmarking results the vendor replaced the switch against a model with higher performance that comes close to delivering full bisection bandwidth. The performance comparison in Section 6 discusses the performance and the cost performance ratios of the different networks by using an all-to-all personalized communication micro-benchmark. Section 7 finally describes the applications we regularly use in our cluster and discusses the relevance or the irrelevance of a full bisection network to real applications. The quite surprising results and experiences provided by the design process, the installation and the evaluation by the micro-benchmarks are presented for a conclusion in Section 8.

2 The Xibalba Cluster: Concept, Design and Implementation

During the past five years many research groups of the department of computer science at ETH Zurich have related some of their research to the cluster of PCs platform by working on the software technologies and the design of such systems, by parallelizing their database systems to run on such clusters, by investigating the scheduling of tasks and work flows in scientific computation on clusters or simply by bringing the important application of large scale car traffic simulation to clusters of PCs.

With many researchers interested in clusters, the major challenges for a departmental cluster was to provide a common infrastructure to be shared by the different research groups accommodating their different requirements.

2.1 A Common, Shared Infrastructure

As it became clear that several groups needed a cluster for their research the issues of the minimal size and the required architectural characteristics were raised. Despite the fact that the communications requirement for all application codes involved was within a narrow range, each group aimed for the largest cluster they could afford to experimentally prove the scalability of their ideas to large systems. It became apparent that a clever sharing concept for this research infrastructure would result in access to a much larger system and open a unique opportunity for a quite special research prototype.

2.2 Mode of Operation for Research in Computer Science

Many uses of computers in computational science just ask for readily and cheaply available compute cycles, that can be provided by any infrastructure, regardless whether it is operated by the research group itself, by a university computing facility or by a national supercomputer center.

The requirements of computer science researchers are quite different. In many computer science research projects the compute platform itself, including its hardware and software, is part of the experiment and needs to be controlled by the researchers. Such a mode of operation is largely incompatible with the setup of a supercomputer center that provides access on a "per job" and not on a "per machine" basis. Furthermore the planned research in parallel databases requires a powerful I/O system in each cluster node, which is usually not available in clusters designed for scientific computing. The concept and the design of the Xibalba cluster addresses those issues. With this concept all four participating research groups could bring their hardware and software requirements into the project. The resulting system remains highly flexible after its installation and the groups are welcome to contribute new, additional system software including new operating systems along with their experiments.

2.3 The Xibalba Hardware Concept

The core of the Xibalba cluster is made of 128 Dual 1 GHz Pentium III compute nodes. Since the database users can not make use of an additional processor, only half of the nodes are equipped with dual processors and the memory is kept at 512 MByte per processor in all the nodes at this time. Still for the node architecture a powerful Intel STL2 dual-processing server board with ServerWorks Serverset III LE chipset was chosen to provide a memory system with excellent characteristics using cost effective standard PC133 SDRAM memory. A 64 bit/66 MHz PCI bus provides maximal I/O throughput for existing and future high speed communication with Gigabit Ethernet or Myrinet PCI adapters. Each node is equipped with two Intel PRO/100+ Fast Ethernet controllers to attach to two separate networks for data and control traffic.

The installation process and the operational experience up to present have shown that the provision of this dual network offers considerable advantages. Even when the cluster is fully loaded and communicating over the data network, NFS mounts, remote shell logins and cluster monitoring software still work fine and allow control transfers at reasonable speed without interfering with user data communication. Furthermore using both networks in parallel for replicating operating system images and the huge data sets of replicated large databases to the entire cluster allows a nearly 100% improvement in speed as we will show in Section 7. We will discuss Xibalba's network options in more detail in Section 3.

The 128 nodes in 2-unit cases are mounted in 8 racks with a ninth rack for the communication facilities and a console. The console is connected to a switch that pools together the keyboard, video and mouse signals (KVM) of all nodes and enables administrators to work with each node directly. The quite costly KVM network for console and video is rarely provided in commodity clusters running either Linux or Windows, but is highly recommended for multi-boot installations that change operating system installation frequently. It also speeds up diagnostic work as hardware (mostly disk) failures occur.

For the research in database systems some special consideration was given to secondary storage in Xibalba. Each node includes two fast 7'200 RPM IBM Deskstar 75 GXP ATA disk drives and two 10'000 RPM IBM Ultrastar 36LZX SCSI disk drives with a cumulative capacity of 100 GByte to provide a distributed and reliable storage for operating systems, scratch space and replicated data files of very large databases at an optimal cost-performance ratio. The total storage of the cluster is over 10 Terabytes.

2.4 The Xibalba Software

The different groups working on the Xibalba cluster rely on vastly different operating system installations with different middleware packages. The parallel databases research group uses Windows 2000 and the SQL Server database management system provided by Microsoft Corporation in a research agreement. Other groups use different Linux distributions, which can be quite specifically configured to their requirements. A small service operating system based on Linux is maintained by the cluster architecture group for diagnostics and maintenance. The same group uses an additional Linux setup with experimental communication system software for benchmarking. For installation and administration purposes the cluster can be booted over the network by a combination of PXE[1] and bootp into a minimalistic Linux installation including diagnostic and administrative tools that runs completely in a ramdisk and can be used to repair broken file systems or install new hard disk drives.

To support the different needs of its users, the Xibalba cluster is conceived as a multi-boot system, that can designate the OS of each node individually. The boot process of every node is centrally configured and can start any operating system supported. In addition to the preinstalled operating systems for immediate use the cluster is equipped with spare partitions on the disks and Dolly [14], a specialized software distribution tool that uses Xibalba's powerful networking infrastructure to distribute entire new software installations to any number of nodes within minutes. With these software tools Xibalba can host experiments that involve complete installations including system, middleware and application software. Note that the performance of two fast Ethernets with bonded channels match about the speed of a disk storing a copy of the incoming data stream during a partition cast.

3 Xibalba Network Options

3.1 Networks for Clusters

For the optimal cost/performance tradeoff the interprocessor communication facility is the most critical part of a cluster. The networks of Xibalba are based on commodity 100 MBit/s Fast Ethernet interconnects, like in most Beowulf class systems. Before inexpensive single backplane networking switches became readily available several different topologies were proposed for Beowulf clusters [16]. As a major difference to most other Beowulf clusters, Xibalba has two Fast Ethernet networks as specified below.

For dedicated networks in parallel computing several high speed interconnect technologies were developed, e.g. with Myrinet. A nice comparison between two such technologies and a traditional supercomputer network is given in [9]. Myrinet with its very low latency and high bandwidth was considered for Xibalba but initially rejected due to its high cost. The expense was not justifiable to the database experts as their database management middleware was not instrumented for high speed communication at all and therefore a high performance network appeared useless to them. In the mean time the traffic simulation group solicited funding to equip a 32 node sub-cluster with Myrinet 2000.

For a brief introduction we give a broad overview of the networking technologies considered and implemented in Xibalba at this point in the evolution of cluster technology (i.e. in the year 2002). The fractions of the network cost relative to the total cost of the cluster are as show in Table 1.

[1] Preboot execution environment

Cluster Network Technology	Cost Ratio Nodes:Network
High Perf. Myrinet	65% : 35%
High Perf. Shared Myrinet	70% : 30%
Full Bisection ER16	80% : 20%
Reduced Bisection E7	87% : 13%
Maintenance Ethernet	96% : 4%

Table 1: Cost ratios (nodes versus network) for different cluster networks in Xibalba.

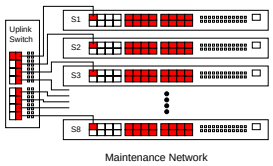

Figure 1: The maintenance network configuration has a very limited bisection bandwidth since 16 nodes (connected by Fast Ethernet) share a single Fast Ethernet uplink.

Maintenance Network For the purpose of separating maintenance and operating system traffic from application traffic we designed a cheap secondary network for the Xibalba cluster in order to supplement the primary network. This network uses 100BaseT technology but is most reduced in its topology and the performance of the components used. The topology follows the physical design. At the price-performance point of 128 nodes, the cost of this network is only 4% of the cluster. This type of network is installed in addition to the primary data network. It is implemented by eight 24-port Enterasys Vertical Horizon VH-2402S Fast Ethernet switches, which are interconnected further by a Fast Ethernet switch of the central communication facilities at ETH Zurich (see Figure 1).

Full Bisection Ethernet The primary data network targeted at in our 128 node Xibalba cluster design is specified to sustain full-speed non-blocking, full-duplex communication on all ports simultaneously. Several networking product vendors offered their switches which shall comply to this specification. This network was first implemented by a large central Enterasys Matrix E7 network switch, including four 6H302-48 line-cards providing 48 Fast Ethernet ports each. We will explain the problem with this equipment and the reason for providing more ports than what seemed required in Section 5. Due to the many limitations of the Matrix E7, the switch was upgraded to an Enterasys X-Pedition ER16 Switch Router with seven ER16-TX-24 switching modules (24 port 100Base-TX) and an ER16-8 Gigabit uplink switching module (8 port 1000Base-SX). At the price-performance point of 128 nodes, the cost of this network is 20% of the cluster while the E7 solution amounts 13%.

High Performance Network As a representative of the expensive networks we are considering Myrinet 2000. A technical introduction is given in [3]. The important difference to the previous networking concepts is the emphasis on expensive network interfaces and low cost high speed switches. At the price-performance point of 128 nodes the overall cost of Myrinet is about 35% of the total cost of the cluster. In our cluster a 32 dual processor node subsection is equipped with Myrinet allowing to use the network either solely with just one processor or in a shared dual configuration. The second option leads to a network cost *per processor* ratio of 30% (see Shared Myrinet in Tables 1 and 2).

The relative costs of the network components are as shown in Table 2. The main difference between a dedicated high performance network and an Ethernet lies in the cost ratio of the switches versus the interface cards. While interface cards for Ethernet are nearly for free because already built on the main boards the cost lies in the switches. For Myrinet it is the other way round. The switches can be built very simple as the intelligence is in the interface hardware which is therefore much more expensive than Ethernet adapters.

Cluster Network Technology	Cost Ratio Switch:Cable:Interf.
High Perf. Myrinet	24% : 9% : 67%
High Perf. Sahred Myrinet	24% : 9% : 67%
Full Bisection ER-16	92% : 2% : 5%
Red. Bisection E7	87% : 4% : 9%
Maintenance Ethernet	65% : 5% : 30%

Table 2: Cost ratios (switch versus cabling versus interface) for different cluster networks in Xibalba.

All our cost calculations are for cost/performance evaluation only and are given relative to the total cost of the 128 node Xibalba cluster which amounts to about US$ 500'000. As we work in a country with exceptionally high wages and high cost of graduate students labor, the design of the Xibalba cluster was advertised in a public bidding process. The winning bid was by DALCO Inc., a local contractor that happily assumed the responsibility for systems integration and installation in the machine room of our university.

3.2 Full Bisection Bandwidth

Interconnect networks of most regular computing structures are characterized by their bisection bandwidth. In the discussion of bisection bandwidth the worst case performance critical bisection of the network is determined (according to the topology) and addressed. For the measurement the nodes are paired in such a way that all the communication must go across the links on the most critical bisection cut. If the network access provides simultaneous full duplex links the communication between the pair of nodes must also be addressed as full duplex, i.e. must go simultaneously in both directions.

A network is said to have a full or—in somewhat more precise terms—a fully scalable bisection bandwidth, if it

can sustain the full network access bandwidth of every node across the most critical bisection while all nodes communicate simultaneously. For a 100BaseT network this means that every node must send and receive data at the same time with 100 MBit/s. Network topologies with full or scalable bisection include the full fat tree, the hypercube and the full crossbar central switches. The mesh, the torus and the plain/skimmed tree network configurations do not offer scalable bisection bandwidth in general, but for some cases some full bisection communication might be achievable for machines up to a certain fixed size.

3.3 Cluster Networks with Full Bisection Bandwidth

In switch based high performance networks like Myrinet the Clos network used for their switches can readily sustain scalable bisection bandwidth up to 128 nodes at almost linear cost per port. After that scaling beyond 128 nodes will face some growing switch costs per port as multiple switches have to be cascaded into a larger network. Still full bisection bandwidth is doable for high performance networking in larger machines.

Ethernet based networks with full bisection can be constructed from either single backplane switching solutions or fat trees using small 8-way switches with up-links to a central backplane that are 8–10 times the speed of the basic links. Both kinds of networks were considered for the primary data network of the Xibalba cluster, but finally the single backplane solution was given preference. The single switch solution can scale up to about 512 nodes for basic 100BaseT connections. Fat trees can scale up to somewhat larger configurations, even without any exploding costs in practice when multiple up-links and a moderate number of duplicated backbone switches are used.

4 Evaluation Principles and Microbenchmarks

We intend to design and evaluate the performance of an interconnect for specific communication patterns, that can be represented as micro-benchmarks. The primary goal of looking at isolated communication primitives is to gain architectural insights into the bottlenecks. Despite the primitive function we can relate these benchmarks to some common communication patterns in real application code quite easily.

4.1 Bandwidth versus Latency

There is a lot of emphasis on communication bandwidth in this study and the aspect of latency is not considered much. The commodity system architect can not do much about certain latency components in the system e.g. the PCI bus arbitration latency. A common wisdom says that additional bandwidth can be purchased easily while latency is given by the laws of nature (or maybe better by the boundary conditions of systems engineering). In the light of this background there are many more interesting cost performance tradeoffs for additional bandwidth than there are for lower latency.

We understand the several order of magnitude difference of latency between a high performance interconnect, using a flit level worm-hole routing scheme in the switches and a communication co-processor at the endpoints vs. the commodity Ethernet that uses store-and-forward routing and a simple host interface using delayed interrupt processing due to coalescing of interrupts. Still many applications are affected by the granularity of communication instead of pure latency and can therefore be reprogrammed accordingly to communicate in large blocks or use mechanisms of latency tolerance.

4.2 Communication Patterns Requiring Full Bisection Bandwidth

Communication patterns requiring full speed communication across the critical bisections are relatively rare and can be avoided in many cases by clever parallel programming or with probabilistic algorithms for large data sets (e.g. with sample sort) [2]. The most important parallel algorithm requiring full bisections are computations in a bitonic sorting network or an FFT butterfly network. The most common communication pattern limited by critical bisection is all-to-all personalized communication.

4.2.1 All-to-all Personalized Communication

The all-to-all personalized communication (AAPC) step is frequently encountered in parallel programs. In an AAPC step, each processor sends a block of distinct data to every other processor. The AAPC step occurs in multi-dimensional convolutions (e.g. FFTs) and in array transposes where only one dimension of the array is distributed [18]. Transforming a two-dimensional 4096×4096 HDTV video image to Fourier space and back for filtering at 30 frames per second would require 60 GFlops/s sustained performance and can certainly only be done with an entire PC cluster or a big array of dedicated DSPs. Transforming a $128 \times 128 \times 128$ grid for a particle mesh Ewald force calculation in a molecular dynamics simulation code at 1000 time-steps per second costs 70 GFlops/s for the FFTs alone, not including any additional work for force calculations or total energy evaluations. In addition to the GFlops/s those applications require a GBytes/s communication performance.

A simple message passing AAPC program:

```
parallel algorithm AAPC
1  for i = 1 to NumberOfProcessors − 1 do
2     NBSendMsg(Destination_i, DataBlock_i)
3  for j = 1 to NumberOfProcessors − 1 do
4     NBReceiveMsg(Source_j, DataBlock_j)
```

We assume that `NBSendMessage()` and `NBReceiveMessage()` are buffered, non-blocking primitives offered by the message passing library. Still this simple program will cause congestion, loss of packets and TCP retransmissions for any larger machine using simple Ethernet networks. We modify the algorithm to proceed in phases carefully controlling the congestion in each phase.

4.2.2 Congestion Controlled AAPC as a Micro-Benchmark

A phased AAPC algorithm as described below can be devised and can achieve optimal aggregate bandwidth once the different phases are carefully separated. Phase separation can be maintained by globally synchronizing the entire machine after each phase is completed. This strategy adds some overhead for synchronizations and might require additional communication resources and/or dedicated hardware mechanisms, but it makes sure that no communication resources are wasted due to inefficient scheduling and due to unnecessary congestion.

A simple algorithm for AAPC proceeds as follows: In the first phase every node sends data to its next higher neighbor and receives data from its next lower neighbor. In the next phase, every node sends data to its next but one higher neighbor and receives data from its next but one lower neighbor and so on. In the last phase every node sends data to its next lower neighbor and receives data from its next higher neighbor. A pseudo code representation of our implementation looks as follows. Each node n_{self} runs the all-to-all algorithm:

parallel algorithm *all-to-all*
1 **for** $i = 1$ **to** $n - 1$ **do**
2 concurrently send data to node $n_{(self+i) \bmod n}$ and receive data from node $n_{(self-i) \bmod n}$
3 wait for barrier

For the evaluation of the AAPC performance we try to minimize the congestion in each phase. For every phase each node has a fixed communication partner to send to and to receive from. The patterns can be symmetric (same node to send to and receive from) or asymmetric (different nodes). Since phases are synchronized across the entire machine the duration and the final throughput is determined by the slowest connection of a phase. In the most common case of a balanced AAPC the same amount of data is sent/received by each node in each phase. In such an AAPC a lower performance is an indication of congestion due to a particular communication pattern.

In the simple algorithm above the logical communication distance increases with each phase of the algorithm. The physical distance between the communicating nodes depends on the mapping of node numbers to communication ports and of the topology of the network. An AAPC contains many different communication patterns and a large number of network properties are exercised. We study a simple linear mapping between node numbers and switch ports that are grouped on different interface modules plugged into a switching backplane. Therefore the next neighbor communication stays mostly within a switching module while some long range communications traverses the module boundaries and the backplanes.

We show the detailed result of an AAPC benchmark in two different graphical representations. First we look at performance vs. communication distance to check for limitations in inter-module communication in switched Ethernet or for decencies on other topological features in the network. The amount of congestion vs. distance is graphed in a Figure like Figure 2.

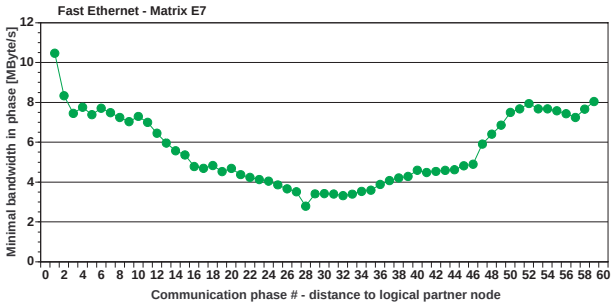

Figure 2: All-to-all performance for different communication phases for the Enterasys Matrix E7 Switch.

To look for hot-spots we time each connection between each source-destination pair one-by-one. For an AAPC of 64 nodes we have 4096 distinct source-destination pairs. To show the impact of congestion to each route in the network we graph a histogram according to the different communication speeds (see Figure 3). The number and the properties of slow communication can lead easily to the identification of hot-spots in the network but the histogram of connections alone does not contain the information necessary to determine the overall execution time and overall performance. Due to global synchronization the performance of the phase is determined by the slowest connection in the phase and therefore a separate histogram captures the distribution of phases according to their speed. The execution time of the phases is truly cumulative and therefore the weighted average of all phase speeds is the total performance of the AAPC, provided that the synchronization overhead is negligible. For large data blocks the barrier synchronization can be neglected, for small data blocks congestion is not a limiting issue and the barriers are omitted. The amount of connections with congestion and the number of phases with congestion is outlined in figures like Figure 3.

Many other communication patterns are subsets of AAPC e.g. some next neighbor patterns or some bitonic sorting exchanges. The corresponding performance data about a particular interconnect under test can be easily derived from the detailed performance characterization of the machine under AAPC load.

Detailed timing data of the different communication phases on all the nodes is gathered to allow careful analysis of the capabilities of the underlying networking architecture or switching technology. The all-to-all benchmark is integrated into the *switchbench* benchmark package [10].

5 Analyzing a Non-Performing Ethernet Switch

5.1 Different Network Configurations

As announced in Section 1 and described in Section 3 we have three networks installed in the Xibalba cluster: A full bisection primary data network implemented by a large central Ethernet switch, a secondary maintenance

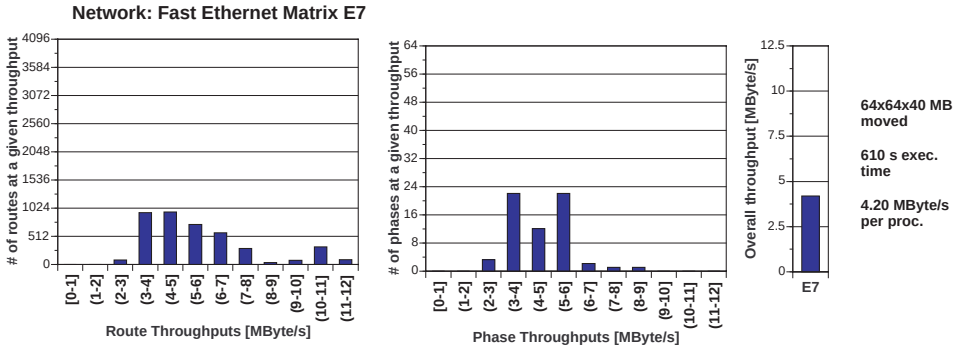

Figure 3: Histograms of transfer rates for the 64^2 routes (left), the transfer rates of the 64 phases (right) and the overall performance in a phased AAPC algorithm on a 64 processor cluster for the E7 network configuration.

network implemented by 8 small switches mounted in the 8 racks—including an up-link switch that interconnects these 8 switches by a Fast Ethernet link—and a Myrinet 2000 network in a part of the cluster. For these tests TCP/IP and the socket interface was chosen as the software API.

The limited performance of the primary network as it was first installed gives us a good picture how a fully switched off-the-shelf backplane switch with reduced bisection bandwidth would operate with 128 nodes. In this Section we focus on this reduced bisection network and study the limitations introduced by its design. Still as we paid for a full bisection network this configuration was upgraded to full bisection.

We distinguish between some switch connectivity patterns as shown in Figure 4. Those patterns were instrumental to characterize the bottlenecks in the line modules of the Matrix E7 switch. As shown in Figure 5, a Matrix E7 switching module consists of two ASICs that provide 24 ports each. These two ASICs communicate over an internal bus at full speed. But as measured later in the benchmark results each ASIC provides barely enough bandwidth for 16 ports. Such bottlenecks are fairly typical for equipment that is optimized for general LAN use.

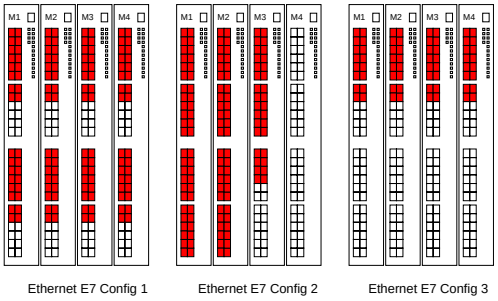

Figure 4: Different network setups and switch utilizations of the primary network for the benchmark tests.

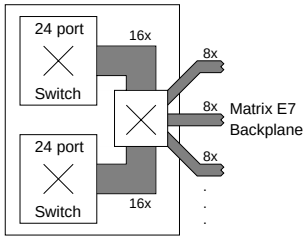

Figure 5: Matrix E7 module: Each module consists of two internal 24 port switches that are able to communicate at full speed to each other or with reduced speed to the E7 backplane.

Ethernet E7 (configuration 1) This switch configuration was the configuration we run on with the E7. Each ASIC was populated with 16 machines only to achieve a better bisection communication.

Ethernet E7 (configuration 2) This configuration uses all the ports provided by a module and uses only three modules with 48 ports each.

Ethernet E7 (configuration 3) To further test the communication between switching modules we setup a configuration where just one ASIC is used per module which results in 16 nodes per module.

5.2 Performance Measurements

Pairwise Traffic Tests

We first use a pairwise traffic generator to analyze all kinds of different bisections of the E7. In this micro-benchmark a set of machines communicates in pairs. All pairs send and receive a large amount of data between the two nodes, in parallel and at full duplex. The pairs in a set N of n machines are separated at an arbitrary distance of node ids, called a stride t, and with wrap around, so that the node pairs $(i, i + t \bmod n)$, where $0 \leq i < n$, $n \bmod 2 = 0$, communicate with each other. The stride parameter t allows to test different bisectional communication patterns, thereby varying the amount of data crossing a well defined bisection line.

The switch consists of a rack with a switching backplane as well as single switching modules providing 48 ports each. To consider this architecture we divide the study in two parts: intra-module and inter-module communication.

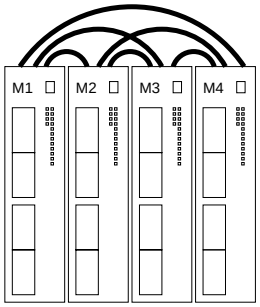

Figure 6: Matrix E7 and ER16 backplane: All line modules are connected with all other modules. The backplane itself should not be a limitation to the total communication bandwidth.

Performance Matrix E7		
Communication Partners (from, to)	Nr of Nodes	Transfer Rate [MByte/s]
Intra-Module comm.	7+7	11.2
(ASIC 1,ASIC 1)	8+8	10.5
Pairs: (1,2)(3,4)..	9+9	9.7
..(23,24)	12+12	**7.8**
Intra-Module comm.	14+14	11.3
(ASIC 1,ASIC 2)	15+15	10.7
Pairs: (1,25)(2,26)..	16+16	10.4
..(24,48)	24+24	**7.8**
Inter-Module comm.	7+7	11.3
(Module 1,Module 2)	8+8	10.2
Pairs: (1,49)(2,50)..	12+12	6.9
..(48,96)	48+48	**2.2**

Table 3: The results of the pairwise tests for different number of pairs with the reduced bisection network (Matrix E7 conf. 2, i.e. all ports used). We measure intra- and inter-module communication.

To measure the *intra-module* capability of a switch module we first generate pairwise traffic within the first internal ASIC. The upper part of Table 3 shows the achieved bandwidth that drops for the reduced bisection network as soon as more then 16 ports communicate. The aggregate bandwidth limitation of an E7 ASIC seems to be 2×1600 MBit/s. If the number of pairwise partners is increased to two ASICs (48 ports, middle part of Table 3) the performance is the same as in the intra-ASIC case, which means that the intra-module communication bandwidth is not reduced below the limitation of a single ASIC. A limited usage of the switching modules to no more than 16 nodes per ASIC enables full bisection bandwidth within a module and was the reason for using the data network configuration 1 with the E7 switch in the Xibalba cluster.

The *inter-module* communication was measured by pairwise traffic between each port of the first module to each port of the second module with the reduced bisection E7 network. The lower part of Table 3 shows the results which reveal a drastic drop in bandwidth for more than 8 ports communicating to their partners. The aggregate bandwidth limitation for inter-module communication seems to be limited to 2×800 MBit/s. While every module of the switch is directly connected with each other module, these connections are not capable of transferring data between more than 8 ports communicating at full speed (see Figure 6). This is a severe limitation down to just $1/6$ of the specified bandwidth. Only very reduced bisection bandwidth is possible as soon as the number of nodes reaches 8 per module.

All-to-all Communication Tests

The all-to-all communication tests show the limitations of a reduced bisection network over a full bisection network with a slightly more realistic workload.

The reduced bisection network implemented by the E7 configuration 1 performs very well for 32 nodes by over 10 MByte/s for all communication steps and provides nearly bisection bandwidth. But going up to 64 nodes an inter-module communication limitation of the E7 switch reduces the resulting total bandwidth significantly.

For the E7 configuration 2 we have again the ASIC limitation inside a module resulting in a sustained bandwidth of 6 MByte/s for all patterns. We have less inter-module communication here, therefore this limitation does not carry weight.

The interesting test with network configuration 3 shows the inter-module limitation quite clearly. The more communication paths cross the module boundary, the more the bandwidth drops. As soon as the inter-module communication limit is reached the bandwidth continuously stays at 2 MByte/s for some phases.

Network	Execution Time
Ethernet E7 Config 1	610 s
Ethernet E7 Config 2	711 s
Ethernet E7 Config 3	468 s

Table 4: Execution times for an all-to-all test with 64 nodes on different network setups for the E7 matrix switch. Each node transfers and receives a 40 MByte message to/from its partner (in every step).

The overall execution times for these tests are given in Table 4. As implied by the bandwidth results the Ethernet E7 configuration 2 needs roughly 16% more time than the configuration 1 where the underpopulated configuration 3 results in 23% better performance.

Streaming Tests for a Data Grid Environment

To test the suitability of the switches for a data grid or meta-cluster environment, we implemented a third series of tests. In these environments it is of great importance that a large amount of data can be streamed into, out of or between clusters. Either because multiple clusters are virtually connected to a larger meta-cluster and many processes of a distributed computation on the nodes need to exchange data, or because the computation requires the processes on the nodes of the cluster to read or write large files from or to cluster-external storage servers. In both cases a performing switch is expected to stream data with the full wire-speed of its external data connection.

In this experiment we used a second cluster consisting of 16 nodes similar to the Xibalba dual CPU nodes. The nodes of the second cluster are interconnected by a Gigabit Ethernet network and a Cabletron Smart Switch Router 8600. The switches of the two clusters are interconnected with two fiber optic cables that are pooled into a smart trunk offering a maximal bandwidth of 2 GBit/s. For the test we had an increasing number of clients on the nodes of the second cluster that were sending data streams over their Gigabit Ethernet interface and the smart trunk to nodes in the Xibalba cluster. While each node of the second cluster sent four data streams concurrently over its Gigabit Ethernet interface, all of the Xibalba nodes received one single data stream over their Fast Ethernet interface. The tests were conducted with the E7 switch in configuration 1 and the ER16 switch. For the tests with the limited bisection E7 switch we used two setups: In the first setup, all receiving nodes were on the same switching module, while in the second setup the nodes were on two different switching modules.

The results of the experiments are shown in Figure 7 and show quite clearly that the very limited inter-module bandwidth of the E7 switch reduces the transfered aggregate bandwidth drastically. When using only 16 ports per switch module, the aggregate bandwidth scales much better and reaches the much better values of the full bisection ER16 switch. These tests reveal the limited suitability of the E7 switch for data intensive grid environments.

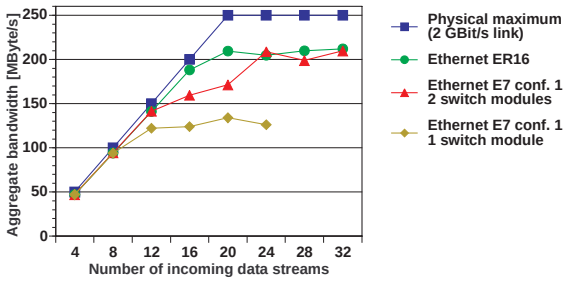

Figure 7: Aggregate bandwidths for increasing numbers of data streams transmitted into the Xibalba cluster.

5.3 Vendor Promises vs. Reality

The outcome of this switch evaluation seems disappointing. It is well known that data sheets sometimes do not reflect the performance of the real hardware implementation. Confronted with our test results the representative of the vendor readily checked with engineering and admitted that there is an inter-module communication limitation in the line modules of the Matrix E7. The local representative also stated that marketing inflates the total bandwidth numbers to take into account that in a “normal” network setting not all the users on a switch will communicate with all other users on the switch and that we are the first customers that have a problem with this limitation. The system integrator relied on the data sheets of the vendor and was rather puzzled by those explanation of his network equipment supplier.

Still during the renegotiation of the acceptance criteria the vendor has offered to upgrade the network to full bisection bandwidth for all nodes by exchanging the Matrix E7 by the X-Pedition ER16 which is referred to as the full bisection network in this paper.

A simple evaluation of the new ER16 switch with the pairwise test resulted in full performance (see upper part of Table 5). The all-to-all communication test revealed that the AAPC transfers between hosts on different switching modules of the full bisection network interestingly resulted in variable performance numbers, depending on the pattern of used ports and the communication phase of the all-to-all communication benchmark. A test with 30 nodes resulted in an average communication bandwidth of 11.3 MByte/s, a test with 60 nodes still with a very good average communication bandwidth of 10.5 MByte/s.

To further investigate the performance degradation in all-to-all tests on the full bisection network, we did various tests with full-duplex pairwise communications and varying patterns (see Figure 8). A strange anomaly can be shown with pattern 1 where the performance drops from 11.3 MByte/s to 8.5 MByte/s (see lower part of Table 5). Pairs of two hosts communicate full-duplex: The first 12 ports of module 1 (hosts 101-112) communicate with the

Performance X-Pedition ER16		
Communication Partners (from, to)	Nr of Nodes	Transfer Rate [MByte/s]
Intra-Module comm. (Module 1) Pairs: (1,2)(3,4).. ..(23,24)	12+12	11.3
Inter-Module comm. (Module 1, Module 2) Pairs: (1,25)(2,26).. ..(24,48)	24+24	11.3
Mixed comm. (patt. 1) (Module 1, Module 2) Pairs: (1,37)(2,38).. ..(12,48) and (13,14) (15,16)..(23,24)	24+12	8.5
Mixed comm. (patt. 2) (Module 1, Module 2) Pairs: (1,25)(2,26).. ..(12,36) and (13,14) (15,16)..(23,24)	24+12	10.7
Mixed comm. (patt. 3) (Module 1, Module 2) Pairs: (1,25)(2,26).. ..(24,48)	24+24	11.3
Mixed comm. (patt. 4) (Module 1, Module 2) Pairs: (1,37)(2,38).. ..(12,48) and (13,25) (14,26)..(24,36)	24+24	11.3

Table 5: The results of the pairwise tests for different number of ports for intra and inter-module communication with the full bisection network of the X-Pedition ER16 switch. The patterns for the mixed communication tests are shown in Figure 8.

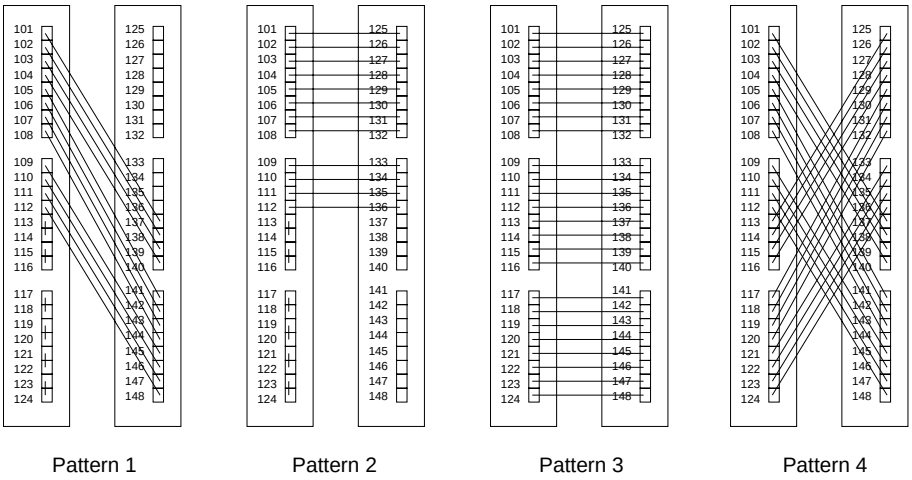

Figure 8: Test pattern used to evaluate the ER16 switch. Pattern 1 leads to reduced bisection in the full bisection network whereas patterns 2–4 work well.

second 12 ports of module 2 (hosts 137-148). The second 12 ports of module 1 (hosts 113-124) communicate with each other (intra-module). From the detailed measurement results it seems that the intra-module communication of the hosts 113-124 slows down the intra-module communication while the pairs 113-114, 115-116 ... 123-124 operate at nearly full speed. Secondly it is interesting that when not all the inter-module pairs communicate with the second 12 ports of the second module the achieved bandwidth is better and reaches 10.7 MByte/s (pattern 2 in Figure 8) or even full performance (pattern 3 in Figure 8). A full inter-node-communication pattern as depicted in pattern 4 of Figure 8 achieves the full bandwidth with all pairs.

A repetition of the streaming tests from Section 5.2 with the ER16 switch revealed a much higher streaming capacity into to cluster then with a single E7 switching module. The capacity of the ER16 is also slightly higher than the E7 with two switching modules (see Figure 7).

6 Performance of the AAPC Micro-benchmark

In this chapter we compare the performance of the AAPC micro-benchmark as presented in Section 4.2.2 on the four principal networks presented in Chapter 3, namely the cheap maintenance network, the full crossbar networks with the inexpensive E7 switch, the expensive ER16 switch and the specialized high performance interconnect with the Myrinet switch. For Ethernet TCP/IP and the socket interface was chosen as API, for Myrinet MPICH-GM was used instead. The all-to-all communication tests show the different performance figures of the networks with a slightly more realistic workload than the isolated pairwise test presented in Section 5.2.

Figure 10 shows the minimal bandwidth achieved by each single communication phase of an all-to-all communication for 60 nodes with the three Fast Ethernet based networks at the bottom. The upper part of the Figure also depicts the performance for the Myrinet network on 30 nodes in single and dual node processor configurations respectively.

Looking at the numbers for the maintenance network in Figure 10, we see the expected sharp drop in performance where all nodes of the cluster attempt to communicate over the highly limited bisection of a single 100 MBit/s link. The bandwidth is slightly higher when a limited amount of intra-switch communication occurs in phases 1–15 and 44–59. The reduced bisection network of the E7 in configuration 1 shows the inter-module bandwidth bottleneck of the switch clearly when more than 8 nodes communicate to another switch module in phases with offsets 9–51. The X-Pedition ER16 full bisection network performs very well by over 10 MByte/s in average.

Looking at the results for the high performance Myrinet interconnect in the 30 nodes case in Figure 10 (note the different axis on the right side) we remark the very uniform performance of the highly symmetrical switch architecture. The performance of the 60 processor case (dual processor nodes) is roughly halved since the two processors of a node share a single network adapter and the

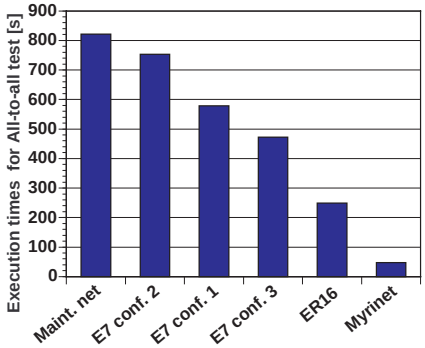

Figure 9: Execution times of AAPC benchmark on 60 nodes with all investigated networks.

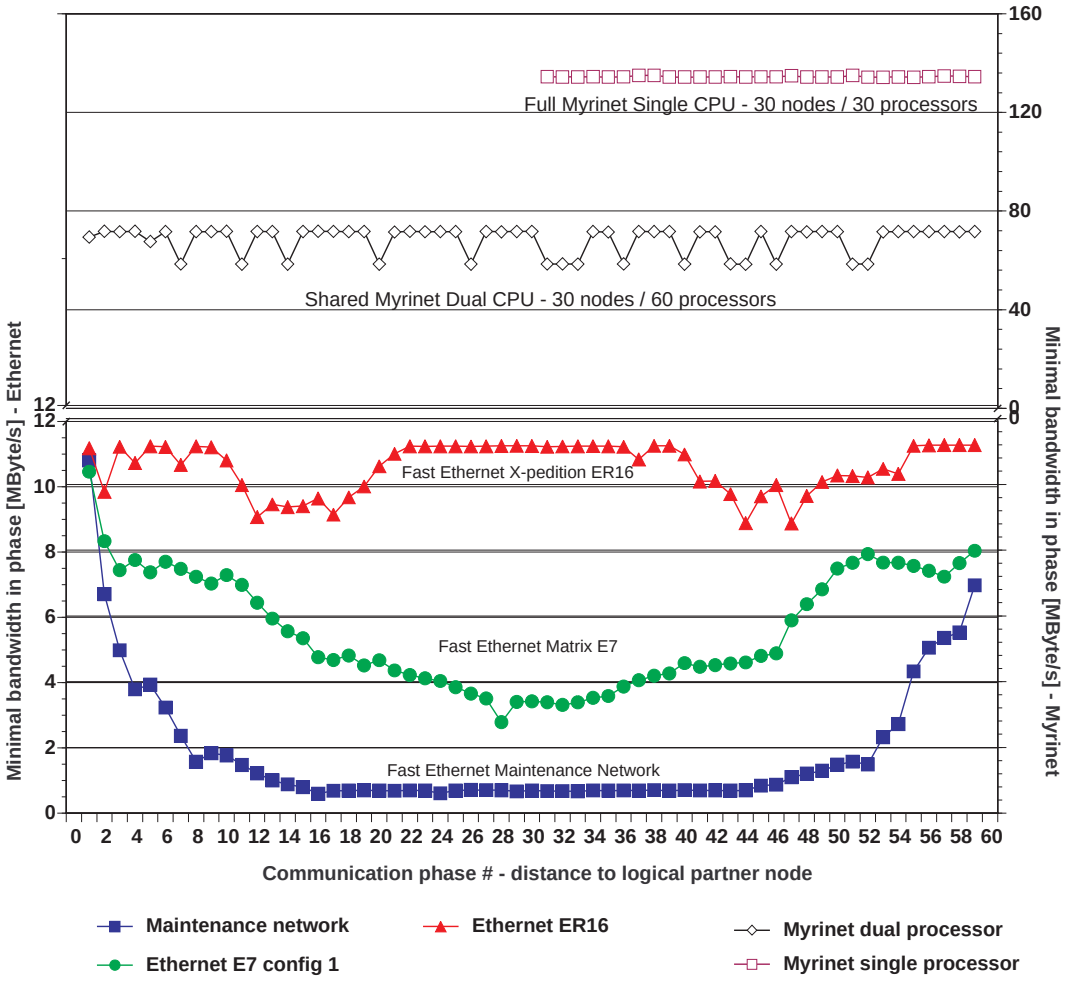

Figure 10: All-to-all minimal per-node bandwidth for different communication phases for all the discussed network architectures.

bandwidth of a single network link. We attribute the irregular performance variations to a measurement uncertainty in an SMP environment with shared resources.

We instrumented an AAPC implementation based on congestion-free communication phases separated by global barriers to record the performance of each individual transfer (see Section 4.2.2). There are P^2 of these transfers with P processors. Figure 11 shows histograms of transfer rates over all 64×64 routes and all the communication phases of the algorithm on 64 processors. The Figure includes also overall throughput and execution time numbers. The histograms show that the maintenance network has an extremely limited performance on almost all routes and phases. On the reduced bisection network of the E7 reduced bisection bandwidth switch the performance for the routes varies considerably. The routes performance varies depending on intra-module or inter-module communication. The network of the ER16 offers very well performance on almost all routes. Since every single route with bad performance reduces the performance of a whole phase, there are slightly more phases with reduced performance than routes. Looking at the Myrinet interconnect shared between two processors we note a high percentage of routes with slightly reduced performance due to the resource sharing between the two processors. With single processors per node the performance of the Myrinet network achieves almost a perfect distribution of high route and phase bandwidths.

Figure 9 compares the execution times of all the AAPC benchmarks from Sections 5 and 6.

7 Performance of Application Benchmarks and Applications

In Section 6 we determined significant differences in performance for the different switched Ethernet and Myrinet configurations. While these differences are quite interesting for the architects of a cluster they might not matter much for the typical application user of the Xibalba cluster. We can certainly confirm that the amount of communication of the parallel database project using Windows NT and SQL Server is well below the limitations

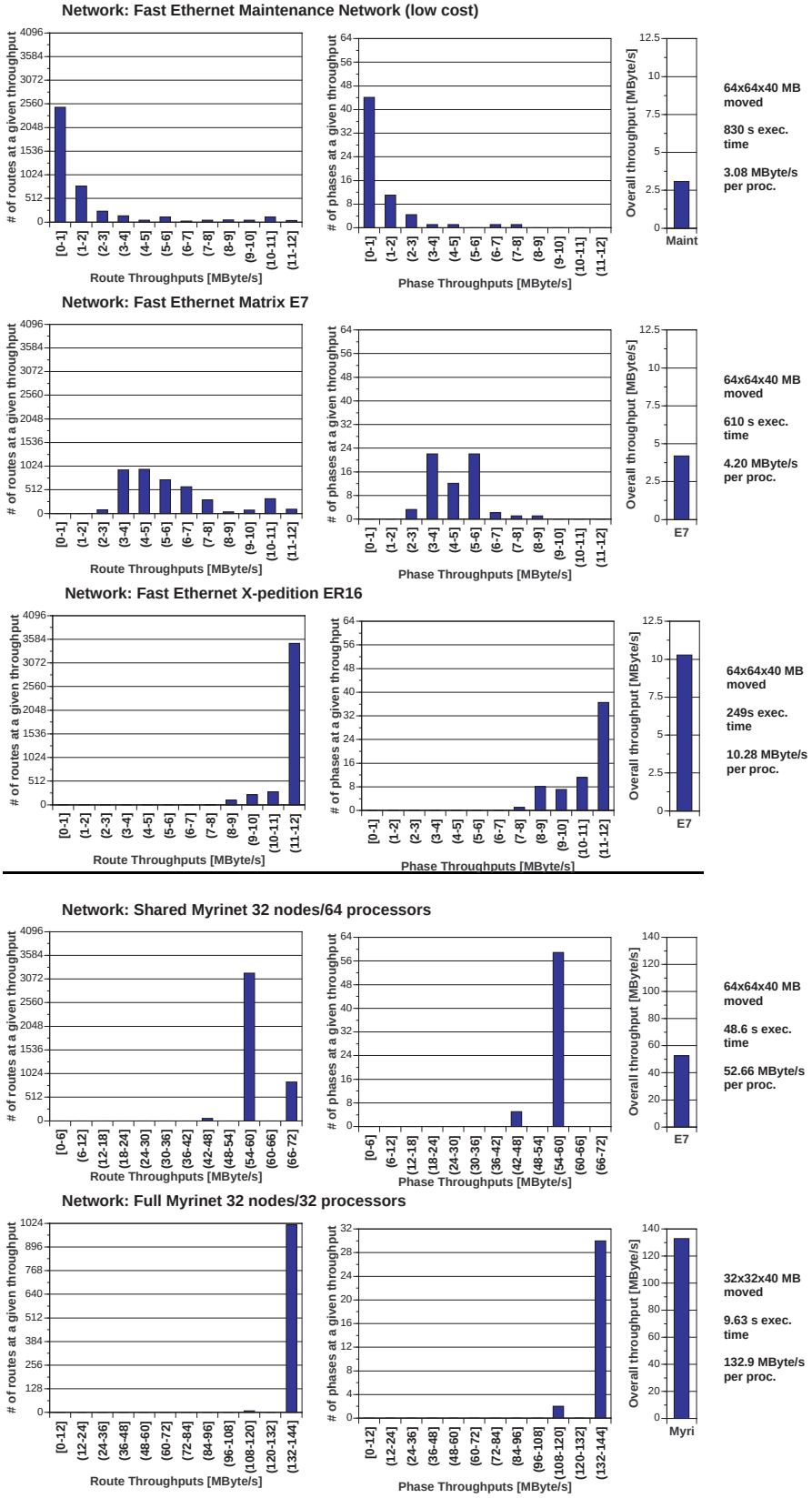

Figure 11: Comparison of the four networks with regard to transfer rates for the 64^2 routes (left), the 64 phases (center) and the overall throughput per processor (right).

we found due to software overheads, but how about other applications?

We measured three different kinds of application benchmarks to find out if the differences in networking performance really matter. The codes represent a communication bound workload, a mostly compute bound workload and a mixed workload respectively. The application programs are as follows. An additional benchmark measures the switches' capabilities to stream data from a high-speed link into a cluster connected to them, thereby testing their suitability for a data grid application.

7.1 Dolly

Dolly is our partition multicast utility program for system administration on clusters as described in [14]. Dolly is a small program that distributes large amounts of data to many nodes in a cluster in a highly efficient way. It is mostly used to install new operating systems in partitions on the hard disk drives of clusters or replicating database images with maximal performance. In short, it sends the partition data from a master in a virtual multi-drop-chain over TCP/IP to the first participating node in a cluster, which writes the data to the local hard disk drive and forwards it concurrently to the next participating node and so on. With Fast Ethernet or dual Fast Ethernet as networking technology and fast hard disk drives the nodes in the Xibalba cluster are capable of saturating their network interfaces for sending as well as receiving data at the same time. For the purpose of this benchmark Dolly does not access the local hard disk drives but sends dummy data through its multi-drop-chain.

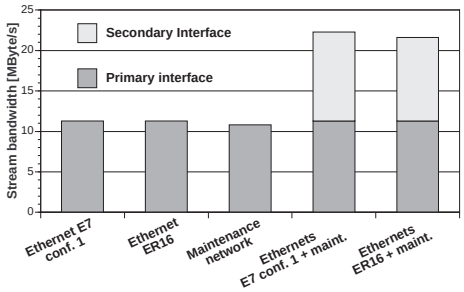

Figure 12: Average bandwidth of Dolly stream through 30 nodes with the full bisection Ethernet with the configuration 1 and maintenance network alone as well as combined.

The application is communication bound, but its communication pattern is limited to a few high speed connections to the nearest neighbors of each node. Because of this communication pattern there is only very limited data traffic over any bisection for reasonable configurations.

In Figure 12 we measure a dolly partition broadcast for a distribution to 60 nodes in parallel and examine the data distribution over the maintenance network, the reduced bisection switch E7 and the full bisection switch ER16. As expected from the results in Section 5.2, Dolly is able to use the full bandwidth on the E7 and ER16 switches. The maintenance network is able to handle nearly the same stream bandwidth as the central switches since its minimal cost switches do not run into any bandwidth limitation for 16 connected nodes. Furthermore, we combine the maintenance network with the different data networks using Dolly's capability to send data over both interfaces to double the throughput as seen in Figure 12.

For the data distribution application Dolly, the cheap maintenance network offers roughly the same performance as the expensive full bisection Ethernet with its large central switch and therefore the maintenance network is a cost effective investment to double Dolly's performance.

7.2 HPL

HPL (High Performance Linpack [6]) is a popular benchmark suite to evaluate the computational capabilities of supercomputers and clusters. The results of that benchmark are published semi-annually in the *Top500* list of the worlds most powerful computers [13]. The benchmark involves solving a system of linear equations.

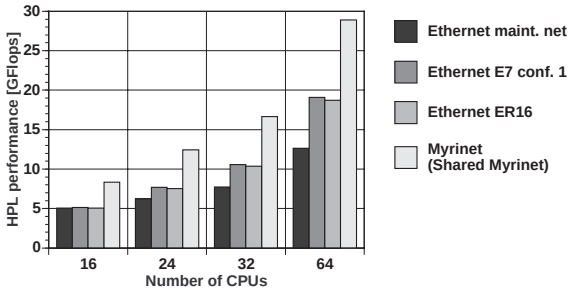

Figure 13: Performance comparison of cheap, medium and expensive networking architectures for the HPL benchmark in GFlops.

The results of the benchmark depend only moderately on the performance of the underlying communication network and the tasks executing at the different nodes of the cluster are mostly compute bound. The communication pattern involves broadcasting panels of columns, which can be done by six different broadcasting algorithms. We used the broadcasting algorithms "Increasing-ring" and "Increasing-2-ring(modified)" as they gave the best performance. Despite the broadcasting of data in the computation, the communication is mostly between near neighbor nodes in any time-step and does not seem to require a high bisection bandwidth.

We examine the results of the HPL benchmark which was run on 16, 24, 32 and 64 processors with the high performance Myrinet network (in dual node configuration), the full bisection ER16, the reduced bisection E7 configuration 1 and the maintenance network. The results of the benchmark are shown in Figure 13.

The HPL benchmark was not tuned for maximal performance on the Xibalba cluster, as every node uses 50 MByte of memory during all the experiments. The results are fine to compare the different networking architectures against eachother, but should not be used to compare the performance of the Xibalba cluster with other clusters (a Top500 test with optimal parameters resulted in approximately 60 GFlops on Xibalba).

The results of the HPL benchmark on 16 nodes reveal no difference between the Ethernet architectures. When using 16 nodes, all the nodes are directly connected to the

same switching module/switch in all cases and are therefore practically identical. As more nodes are used the limited bisection bandwidth of the maintenance and the E7 network become more important factors. The same holds for the additional latency due to the stacked switches in the maintenance network. The latter low cost architecture shows clearly worse performance than the fully switched Ethernets networks with single central switches. The two Ethernet networks E7 and ER16 respectively achieve about the same performance, the only difference being the slightly higher latency of the ER16 due to its more sophisticated higher level switching features. Myrinet with its much higher bandwidth and lower latency surpassed all the Ethernet network architectures by more than 50%.

Since the MFlop counts of the HPL Benchmarks are well documented we can calculate a price/performance ratio for the different networking architectures based of the cost of $818 per port for the full bisection Ethernet ER16, $480 per port for the reduced bisection E7 and $145 per port for the maintenance Ethernet. For the different machine sizes the price per MFlop is roughly constant at about $1.60 for the E7 and $2.40 for the ER16 configuration and varies from about $0.50 on 16 nodes to about $0.75 on 64 nodes for the maintenance network. Therefore adding a better network increases the performance, but definitely reduces the price performance ratio of a machine regarding the execution of HPL.

Mostly due to its much better latency Myrinet performs best in all the tests and scales nearly perfectly up to 64 nodes. The optimal scaling also holds for the 64 processor configuration where two CPUs share a motherboard with a single network adapter. Due to the most cost effective dual CPU configuration the cost performance ratio there is about $1.80 per MFlop with 2x$900 per node allocated to the interconnect.

7.3 QTPlan

QTPlan is a parallel program to model queuing in traffic micro-simulations [4]. In our benchmark the application simulated 6 hours of real-time traffic in Switzerland. The input comprised 50'000 and 990'000 automobiles respectively, on their way through a two lane tunnel of the single highway passage to the southern part of Switzerland. The road map is space partitioned in order to minimize the number of connections between the processor nodes. The 50 K cars case is a testing scenario for traffic jams, since it assume cold rain in the north and all vehicles drive to the southern part of the country. The crossing of the partitions around the actual traffic bottleneck translate into a network bottleneck between the two machines that hold these partitions. The 990 K cars scenario is simulating an more balanced everyday scenario when most of the cars on the way to and from work all over Switzerland.

The QTPlan simulation has computing as well as communication intensive parts. The communication involves mostly small data packets at a fine granularity and requires a low latency interconnect. The performance results of the QTPlan application are shown in Figure 14.

The execution times of the application tests are taken from a single test run with 64 processors. The tests indicate a better performance on the Ethernet networks with central switches for small amounts of cars compared to the weaker maintenance network. The higher bandwidth and lower latency of the high performance Myrinet network results in nearly halved execution times. With a small working set there are less cars in each space partition (and therefore in each node) and the ratio of computation versus communication drops. With a greater proportion of communication the factor network becomes more relevant resulting in runtime that doubled on the minimal cost maintenance network. With many cars in the simulation, the performance using the maintenance network is only 12% below the performance using the reduced bisection Ethernet, which in turn is also only 12% below the performance figure with the full bisection Ethernet. The more expensive Myrinet network reduced the execution time by another 8%. With large working-sets, the amount of local computation increases in every space partition resulting in a higher computation versus communication ratio. The contribution of the factor network to the total runtime decreases and results in a smaller difference in runtime between the two networking architectures. Another factor that contributes to the higher runtime on the cheap network is the increased average latency because of the stacked switches.

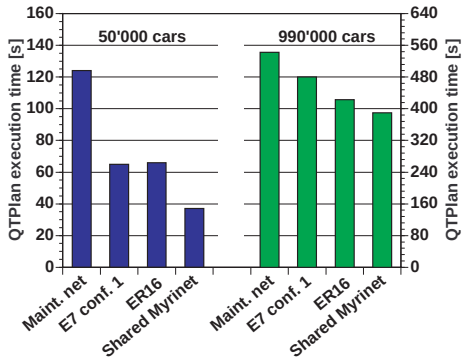

Figure 14: Runtime of QTPlan traffic simulations for 50'000 and 990'000 cars on 64 nodes/processors in seconds.

The cost per performance analysis is more difficult with this benchmark since we do not have GFlops numbers for QTPlan. The compute performance of this application is measured in the simulation to real time ratio. A simple evaluation of the execution times shows that the full bisection network is at the same speed as the reduced bisection network, but both switched networks are a significant improvement over the minimal maintenance network. Therefore investing into a reasonable network does definitely help QTPlan. Myrinet can improve the results in the same amount as the ER16 improves the performance over the maintenance network.

8 Conclusion

In this study we considered four networks as alternatives to connect the compute nodes of a PC cluster: (1) Myrinet 2000, a dedicated, high performance interconnect, (2) a high end Fast Ethernet based on a switch delivering full bisection bandwidth, (3) a lower cost Fast Ethernet based on a central switch but with reduced bisection bandwidth

and (4) a minimal cost Fast Ethernet designed as a secondary network for maintenance purposes.

Looking at the costs of these networks we see that depending on the performance requirement the total fraction of costs allocated to the network in a cluster can range from 4% for a lowest cost Fast Ethernet up to 35% for a multi Gigabit high performance interconnect. The intermediate cost of 20% is reached for a full bisection, Fast Ethernet using an expensive high end switch. It is also noted that there is an entirely different allocation of equipment costs between switches, cables and interface hardware in commodity networks (Fast Ethernet) and in the non-commodity high performance networks (Myrinet). With Ethernet the cost is typically in the switches while with dedicated high performance interconnects the high cost is in the sophisticated network adapters.

With the help of a congestion controlled all-to-all personalized communication (AAPC) micro-benchmark and some isolated pairwise communication patterns we managed to analyze the performance and the non-performance of different Fast Ethernet configurations built from off-the-shelf LAN communication equipment. For the performance analysis we developed two different views to see AAPC and state its performance. The first view gradually increases the logical communication distance in the different phases of the all-to-all communication patterns revealing strength and weaknesses of the switches in full speed inter-module communication. The second view uses histograms over all possible source-destination pairs and routes to track down congested routes that slow down particular communication patterns and lead to poor performance in AAPC.

The performance and the non-performance of our different Ethernet Switches gives a very interesting architectural insight, as we are trying to answer the question of whether commodity Ethernet components used in LAN networking can provide a fully scalable, full bisection interconnect cheaply and efficiently for a mid-sized PC cluster. But only with a fairly expensive and powerful switch the performance of near full bisection bandwidth on Fast Ethernet can be reached.

Looking at the cost performance ration it appears that the best cost performance tradeoffs are at low end of the Fast Ethernet interconnect built from low cost switches as used in our Xibalba maintenance network or then alternatively at the high end of the dedicated high speed interconnect. Spending large sums on a Fast Ethernet switch to achieve full bisection bandwidth seems to be unattractive. Our Xibalba cluster received such a network only due to unwarranted vendor claim that lead to free equipment upgrades in the end.

Unlike with microprocessors the technical requirements for interconnects in mainstream workstations on a LAN remains sufficiently different from a high performance PC cluster setting. Therefore using commodity component for a cluster interconnects stays a mixed success.

In addition to the raw performance figures measured by our micro-benchmarks we also attempted to gain an idea of the performance impact of the interconnect on a storage system utility for high speed disk cloning using partition cast, on the HPL benchmark used in the competition for the Top 500 list and on an applications we plan to use regularly on the cluster. The application is a vehicular traffic simulator developed and is used by some colleagues for large scale simulations of the automotive traffic through the few and narrow passage-ways across the Alps.

While the performance differences appear significant in the micro-benchmarks that exercise the communication system, the final performance impact on applications might be far less than expected. Our experiences with applications show that although good network performance can be very helpful to get high performance codes up quickly, most codes are not very sensitive to full bisection bandwidth in the end and can ultimately be rewritten and adapted to lower communication requirements. Therefore it seems that while full bisection Ethernet bandwidth is a nice feature to have, it appears not to be the most critical issue to the success of a Beowulf class system.

Acknowledgements

We would like to thank Nurhan Cetin and Kai Nagel for their help with the QTPlan application and the opportunity to use their application code for our experiments.

References

[1] D. J. Becker, T. Sterling, D. Savarese, J. E. Dorband, U. A. Ranawake, and C. V. Packer. Beowulf: A Parallel Workstation for Scientific Computation. In *Proceedings of International Conference on Parallel Processing*. CRC Press, Boca Raton, FL, USA, August 1995.

[2] Guy E. Blelloch, Charles E. Leiserson, Bruce M. Maggs, C. Greg Plaxton, Stephen J. Smith, and Marco Zagha. A comparison of sorting algorithms for the connection machine CM-2. In *ACM Symposium on Parallel Algorithms and Architectures*, pages 3–16, 1991.

[3] Nanette J. Boden, Robert E. Felderman, Alan E. Kulawik, Charles L. Seitz, Jakov N. Seizovic, and Wen-King Su. Myrinet — A Gigabit per Second Local Area Network. *IEEE-Micro*, 15(1):29–36, February 1995.

[4] Nurhan Cetin and Kai Nagel. Parallel Queue Model Approach to Traffic Microsimulations. In *Proceedings of Swiss Transportation Research Conference*, March 2002.

[5] William J. Dally. *A VLSI Architecture for Concurrent Data Structures*. PhD thesis, California Institute of Technology, Pasadena, 1986. Technical report 5209:TR:86.

[6] J. Dongarra, J. Bunch, C. Moler, and G. W. Stewart. Linpack users guide. SIAM, Philadelphia, PA, 1979.

[7] Ian Foster and Carl Kesselman, editors. *The Grid: Blueprint for a new Computing Infrastructure*. Morgan Kaufmann Publishers, Inc., San Francisco, USA, 1999.

[8] G. Hipper and D. Tavangarian. Advanced workstation cluster architectures for parallel computing. *Journal of Systems Architecture*, 44(3–4):207–226, December 1997.

[9] Ch. Kurmann and T. Stricker. A Comparison of Three Gigabit Technologies: SCI, Myrinet and SGI/Cray T3D. In *SCI Based Cluster Computing, H. Hellwagner and A. Reinefeld, eds.* Springer, Berlin, Spring 1999. An earlier version appeared in Proc. of the SCI Europe'98 Conference, EMMSEC'98, 28-30 Sept 1998, Bordeaux, France.

[10] Christian Kurmann. Switchbench benchmark. `http://www.cs.inf.ethz.ch/CoPs/sbench/`, June 2001.

[11] Charles E. Leiserson, Zahi S. Abuhamdeh, David C. Douglas, Carl R. Feynman, Mahesh N. Ganmukhi, Jeffrey V. Hill, W. Daniel Hillis, Bradley C. Kuszmaul, Margaret A. St. Pierre, David S. Wells, Monica C. Wong, Shaw-Wen Yang, and Robert Zak. The Network Architecture of the Connection Machine CM-5. *Journal of Parallel and Distributed Computing*, 33(2):145–58, March 1996.

[12] Richard P. Martin, Amin M. Vahdat, David E. Culler, and Thomas E. Anderson. Effects of Communication Latency, Overhead, and Bandwidth in a Cluster Architecture. *Computer architecture news*, 25(2):85–97, May 1997. 24th Annual International Symposium on Computer Architecture, ISCA '97.

[13] Hans Meuer, Erich Strohmaier, Jack Dongarra, and Horst D. Simon. TOP500 Supercomputer Sites. `http://www.top500.org/`.

[14] Felix Rauch, Christian Kurmann, and Thomas M. Stricker. Partition Cast — Modelling and Optimizing the Distribution of Large Data Sets on PC Clusters. In Arndt Bode, Thomas Ludwig, Wolfgang Karl, and Roland Wismüller, editors, *Lecture Notes in Computer Science 1900, Euro-Par 2000 Parallel Processing, 6th International Euro-Par Conference Munich*, Munich, Germany, August 2000. Springer. Also available as Technical Report 343, Department of Computer Science, ETH Zürich, `http://www.inf.ethz.ch/`.

[15] Alexander Reinefeld and Volker Lindenstruth. How to Build a High-Performance Compute Cluster for the Grid. In *Proceedings of Intl. Workshop on Metacomputing Systems and Applications (MSA'2001)*, pages 221–227, September 2001.

[16] Chance Reschke, Thomas Sterling, Daniel Ridge, Daniel Savarese, Donald Becker, and Phillip Merkey. A Design Study of Alternative Network Topologies for the Beowulf Parallel Workstation. In *Proceedings of the Fifth IEEE International Symposium on High Performance Distributed Computing*, pages 626–635. IEEE Comput. Soc. Press, Los Alamitos, CA, USA, 1996.

[17] Thomas L. Sterling, John Salmon, Donald J. Becker, and Daniel F. Savarese. *How to Build a Beowulf: A Guide to Implementation and Application of PC Clusters*. MIT Press, Cambridge, USA, 1999.

[18] T. Stricker and J. Hardwick. From AAPC Algorithms to High Performance Permutation Routing and Sorting. In *Proc. SPAA'96*, pages 200–203, Padua, Italy, June 1996. ACM.

[19] Michael Stumm, Zvonko Vranesic, Ron White, Ron Unrau, and Keith I. Farkas. Experiences with the Hector Multiprocessor. In *Proceedings of the Seventh International Parallel Processing Symposium*, April 1994.