Parallel computer architectures for commodity computing and the Swiss-T1 machine

[1]Pierre Kuonen and [2]Ralf Gruber

[1]Computer Science Department, Pierre.Kuonen@epfl.ch
[2]Computer Services, Ralf.Gruber@epfl.ch
Swiss Federal Institute of Technology
CH-1015 Lausanne, Switzerland

Abstract: Commodity parallel computing becomes more and more popular, as the specialised supercomputing companies stop their activity. In this new concept, a user is confronted with a machine constructed with mass produced fully equipped workstations or PCs which are interconnected through some high speed, low latency network. In this paper, we discuss and compare the different network topologies that are used to cluster those computational units. Then we justify the choice made for the Swiss-T1 parallel commodity computer that will be installed at EPFL at the end of 1999 in the framework of the Swiss-Tx project. The final objective of this is to build a low cost, parallel commodity computer delivering one Teraflop/s by the year 2000.

Résumé: La disparition, au cours des dernières années, des principaux constructeurs de superordinateurs parallèles a favorisé l'émergence d'un nouveau concept de machine parallèle basé sur l'assemblage d'ordinateurs tout à fait standards, tels des PCs ou des stations travail, connectés au travers d'un réseau à haut débit et à latence faible. Dans cet article nous présentons et comparons différentes topologies envisagées pour réaliser de tels réseaux et nous justifions le choix fait pour la machine Swiss-T1 qui sera installée à l'EPFL fin 1999 dans le cadre du projet Swiss-Tx. L'objectif de ce projet est de réaliser, à l'horizon 2000, une machine parallèle de faible coût délivrant un Teraflop/s.

1. Introduction

Since 6 years, most of the supercomputer vendors have been taken over by personal computer manufacturers (Cray, Convex), have stopped supercomputing (Intel), or stopped their business (Thinking Machines, KSR). There is no manufacturer now remaining for whom the main business is supercomputing. Users of high performance parallel machines can now choose among Japanese vector machines (Cray/SGI, NEC, Fujitsu) and SMPs (SGI, IBM, Sun, Compaq, HP, Hitachi). Besides the high prices, the vector machines demand data structures different to those chosen in cache based computers as PCs or workstations and the SMP machines with their customised architectures often do not scale with the number of processors. These are major reasons why commodity parallel computing is now considered as an alternate road map towards high performance parallel computing. In this new approach, autonomous, high performance, shared memory computers are connected by an external high-speed network. Global communication between processors can be taken care of by message passing libraries such as MPI. Such parallel computers are often called message passing machines for which a user has to care about optimising the MPI implementation to become efficient on all those computer architectures. In contrary to vector machines, the use of commodity computers as computational units guarantees that local optimisation has not to be touched when porting a PC or workstation program to a commodity supercomputer.

In this paper we first discuss and compare the different popular network architectures chosen to cluster computational units with a special emphasis on circulant graphs. Secondly, we present the Swiss-Tx commodity parallel computer project [Brauss99, Dubois98] that aims at delivering a Teraflop machine by the year 2000.

2 Network architectures

The definition of efficient interconnection network topology is a major issue of parallel commodity computer designers. Since many years, several topologies have been studied and used to build parallel computers. For example, a few years ago, hypercube topology was largely used because of its apparently low diameter and its good mathematical properties. Besides the SGI Origin2000, this topology is not any more used today because of its lack of scalability. Topologies derived from trees are used in the IBM SP-2 and the fat-tree topology by the Compaq-Quadrics machines using a follow-up of the Meiko [Meiko94] network technology. On the other hand, there is a trend towards simple graphs such as grids (often used by Beowulfs) or the torus (SGI/Cray T3E). In the following paragraphs we will present the results of our studies concerning the topologies for interconnection networks realised in the framework of the Swiss-Tx project.

The graph theory is the main mathematical method applied in the field of interconnection networks. To well understand the content of this paper we start with a vocabulary.

- A graph is made of *edges* and *nodes*
- The *size* of the graph is the number N of nodes of the graph. It is directly related to the maximum computational power of the machine. Typically, each node of the graph will be occupied by a fixed number P of processors
- Two nodes are *adjacent* if they are the extremities of the same edge
- A *chain* between two nodes x and y is a list of k nodes $x_1,..,x_k$ such that two consecutive nodes x_i and x_{i+1}, $0<i<k$, are adjacent and such that $x_1=x$ et $x_k=y$
- A graph is *connex* if there exists a chain between each pair of nodes of the graph. In the following, we are only interested in connex graphs
- The *length* of a chain is the number of its edges
- The *distance* between two nodes is the length of the shortest chain between them
- The *diameter D* of a graph is the longest distance in the graph
- The *average diameter* (or *average distance*) *Dm* of a graph is defined as:

$$\sum d_{ij} / N^2 - N$$

  where d_{ij} is the distance between the node i and j, N is the size of the graph (by definition, $d_{ii}=0$). The average diameter influences the transfer time between arbitrary nodes and the time used to broadcast information. In any case we will try to minimise these values with respect to the size and the degree of the graph
- The *bisectional width BiW* is the smallest number of edges we have to cut in order to separate the graph in two parts of the same number of nodes (plus or minus one). It is an informal measure of the available bandwidth between the two half of the machine. Usually we will try to keep this value as high as possible
- The *degree d* of the graph imposes the number of network communication (NC) ports we must have on each node. Usually this number is dictated by the status of the used technology. In any case, the price of the machine increases with the number of needed NC ports
- A graph is *regular* if all the nodes have the same degree. In order to avoid special-case nodes, regular graphs are our favourite candidates. Special case nodes can become “hot–spots” and increase the contention phenomena
- A *topology* is a class of graphs
- A topology is *rigid* if for any given size N and degree d there exists only a few (<<N) graphs of degree d and of size smaller or equal to N. An example of a very rigid topology is the hypercube. There exits only one hypercube with a given degree d and the size of this hypercube must be 2^d. Therefore, following the above definition, for any d and N there exists at most one hypercube of degree d and of size smaller or equal to N. To build computers of any power we need to have a great liberty on the choice of the size of the graph. Therefore we try to avoid rigid topologies

To summarise, the ideal topology should have the following characteristics*: It should be regular and not rigid, we need a low average diameter, a low diameter and a high bisectional width for a large size and a small degree.*

Our objective is to compare different topologies. To do so, we need to define clearly what parameters of which graphs should be compared. As it can be seen in Figure 1, the processing power of a machine is characterised by the number *N* of processing units (PU) and the number P of processors per node, and the network by the degree *d*, the diameter *D*, the average diameter *Dm*, the bisectional width *BiW*, and the cost. In our study, we will compare graphs with the same number of PUs. In others words, the problem is to decide, for a given computing power, which are the topologies having the best connection characteristics.

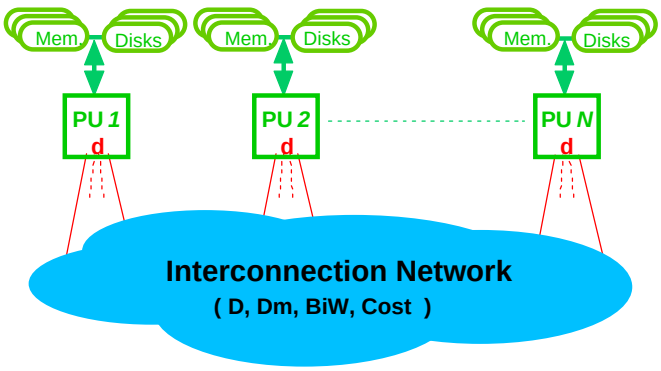

Figure:1 General representation of a commodity parallel computer

Circulant graphs

In [Kuonen99] we analyse and propose the so-called K-Ring topology for the network of the Swiss-Tx machines. We show that this topology has better characteristics than the most currently used (hypercube, torus, fat-tree,..) ones. In the following section we are going to enlarge our study in order to extend our analysis to a topology called circulant graphs [Boesch84] that includes the K-Rings.

- A circulant graph : $C_N<a_1,a_2,..,a_K>$ with $0<a_1<a_2<..<a_K<(N+1)/2$ is a graph of size N where the nodes are numbered from 0 to N-1 and such that the node i is linked to the nodes $i\pm a_1$, $i\pm a_2$,..,$(i\pm a_K)$ mod N.

Circulant graphs are regular graphs, but they can be non-connex (example $C_{12}<2,4>$). It has been demonstrated [Boesch84] that a circulant graph is connex if $gcd(a_1,a_2,..,a_k,N)=1$.

The condition "$\exists$ a_i, a_j such that $gcd(a_i,a_j)=1$" implies that $gcd(a_1,a_2,..,a_k,N)=1$, but the reverse is not true. A simple example is: gcd(6,10,15)=1 but gcd(6,10)=2, gcd(6,15)=3 and gcd(10,15)=5. If we impose that $a_1=1$, we obtain: $\forall$ i, $gcd(1,a_i)=1$ and the corresponding graph is connex. Even if the class of circulant graphs having $a_1=1$ does not contain all the connex circulant graphs, we will restrict our analysis to circulant graphs having $a_1=1$ i.e. to $C_N<1,a_2,..,a_K>$.

It has to be noted that K-Rings are circulant graphs $C_N<a_1,a_2,..,a_K>$ such that $\forall i$, $gcd(a_i,N)=1$. Consequently, K-Rings are included in $C_N<1,a_2,..,a_K>$ (see [Kuonen95] for details).

Fat-trees

Fat-tree topology is used to build multi-stage networks. In these networks some nodes of the graph are computing nodes (PU) while other nodes are switching nodes. More details on the fat-tree topology can be found in [Leiserson85].

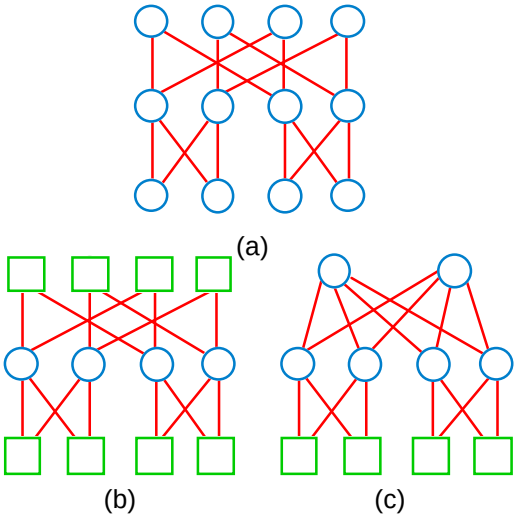

Figure 2: Fat-tree of size 12 and the corresponding interconnection networks

Figure 2(a) presents a fat-tree of size 12. Figures 2(b) and 2(c) presents the two possible solutions for building an interconnection network starting from the fat-tree represented in 2(a). Squares represent computing nodes while circles represent 4x4 crossbar switches. Only computing nodes contain processors. In this paper we will assume that interconnection networks are built using the solution 2(c). This choice is motivated by the fact that fat-trees were designed for maximising the bisectional width. Solution 2(c) leads to a better bisectional width with respect to the number of computing nodes and it is very close from the solution used by Meiko [Meiko94].

As it appears in Figure 2, fat-trees are not regular graphs. Indeed switching nodes have a degree that is the double of the one of computing nodes. In order to compare this topology with a regular one, we have to decide which degree we assume for fat-trees. In order to be fair in our comparison, we based our choice on the degree of the computing nodes. Indeed this degree determines how many NC ports must be present on the PUs. With this hypothesis the graphs presented in Figure 2 have a degree of 2.

Grids, toruses and hypercubes

Toruses are periodic grids. Since their topologies are well known, we only remind that a torus of dimension K is a regular graph of degree 2K.

As our objective is to build the interconnection network of a parallel computer we are not interested by multi-graph (graphs that can have more than one edge between two nodes). More precisely, we consider a multi-graph to be equivalent to the graph obtained by replacing any multiple edges by one edge. With such a definition hypercubes are special case of toruses (toruses of dimension K and of size 2^K).

In the following sections we will compare the characteristics of toruses, fat-trees, and circulant graphs $C_N<1,a_2,..,a_K>$ for the same size and the same degree. Our comparisons are limited to the degrees 4, 6, 8 and 10 because, on the one hand, the degree of toruses and circulant graphs must be an even integer and, on the other hand, circulant graphs and toruses of degree 2 are simple rings.

Comparison of the characteristics of torus, fat-trees and circulant graphs

On figures 3 and 4 are presented the comparisons of the measured values of the diameter, the average diameter and the bisectional width for degrees 4, 6, 8 and 10. For degree 10 the values are obtained with an approximate formula, since the measured values were not available for circulant graphs. These results show that:

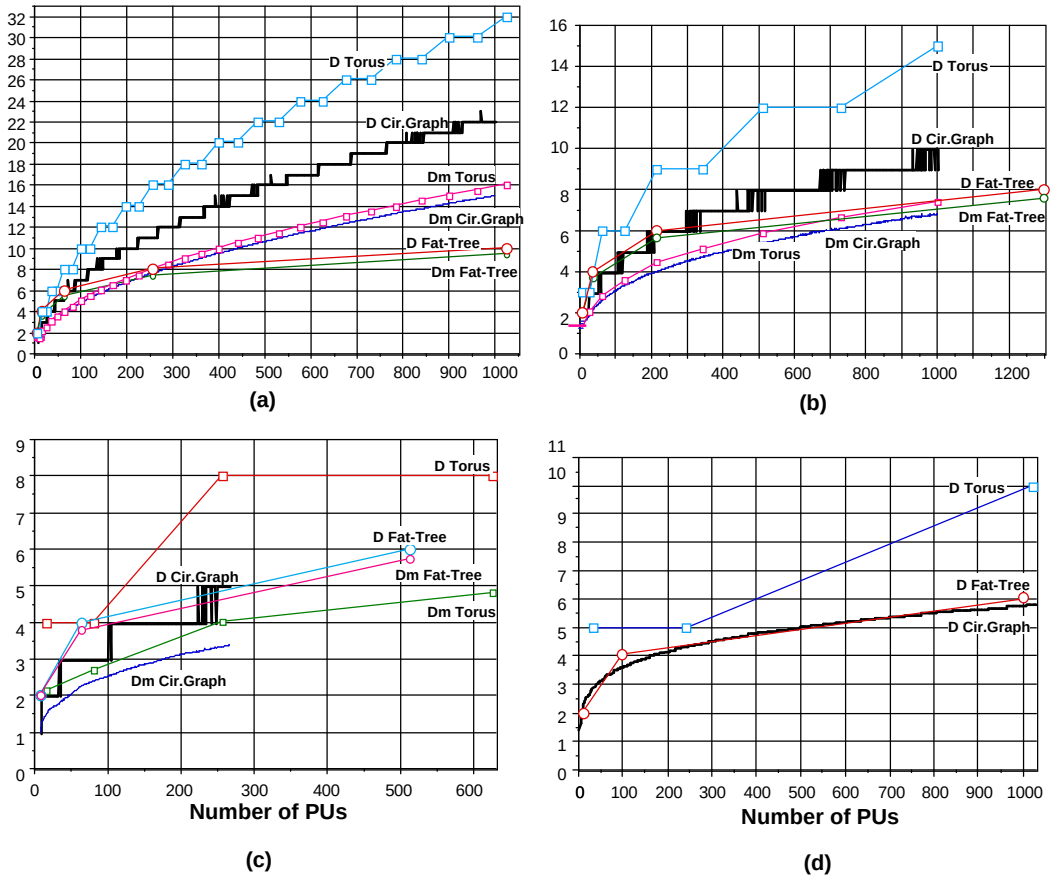

Figure 3: Comparison of the diameter (D) and average diameter (Dm) of toruses, fat-trees and circulant graphs for degrees 4 (a), 6 (b), 8 (c) and 10 (d)

1. Toruses always have the worst diameter.
2. Fat-trees appear to have the best diameter but the difference with circulant graphs is decreasing with increasing degree.
3. The average diameter of fat-trees are very close to the diameter, as a consequence, for degrees greater than 4 and a size smaller than 1000, the average diameter of circulant graphs is smaller than the one of fat-trees.
4. For a number of PUs up to 1000 the diameter of circulant graphs is smaller or equivalent to the one of fat-tree as soon as the degree is greater than 6.
5. Fat-trees always have the best bisectional width, toruses the worst ones, and the bisectional width of circulant graphs is very erratic.

Based on these results we can discard the toruses that always have the worst diameter and bisectional width. Small degree fat-trees seem to be the best choice even if the difference with circulant graphs is not spectacular. Nevertheless, the drawback of fat-tree is that they are extremely rigid. We have the following properties:

- The number of fat-trees of a given degree d and of size $\leq N$ is equal to $\lceil \mathrm{Log}_d(N) \rceil$. For d=8 and N=1000 this number is equal to 3.
- Performant circular graphs can be found for any number of PUs.

In order to decide whether or not fat-trees is a better choice than circulant graphs we are going to study how to build a communication network using these topologies.

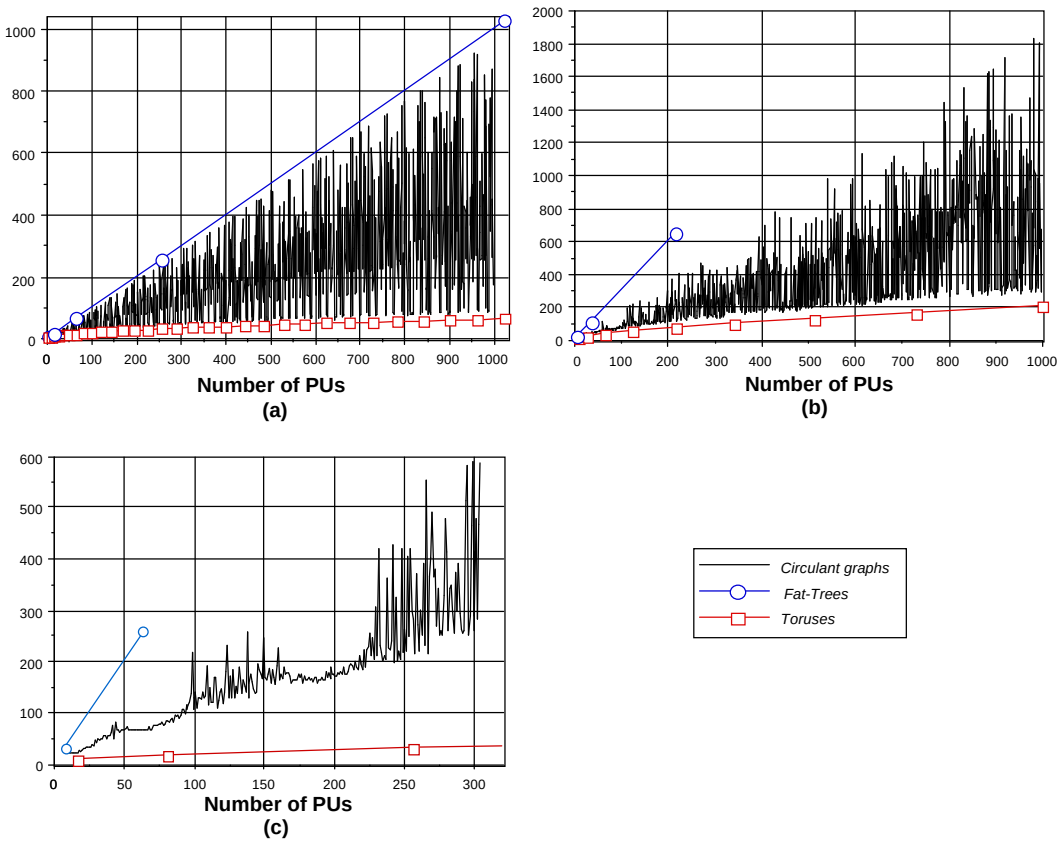

Figure 4: Comparison of the bisectional width of toruses, fat-trees and circulant graphs for degrees 4 (a), 6 (b) and 8 (d).

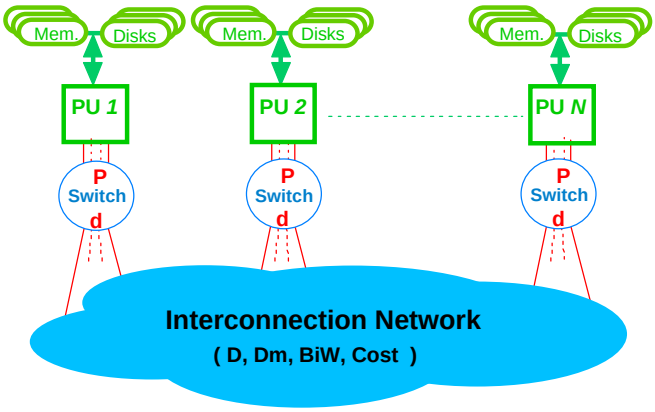

Figure 5: General representation of a parallel computer built using crossbar switches

Communication networks and crossbar switches

Recent developments of high-speed crossbar switches have opened new possibilities for the design and the realisation of interconnection networks. Figure 5 shows the general schema of a communication network built using crossbar switches. In the case of Swiss-Tx machines, the available technology is a 12x12 crossbar switch, called T-NET, designed by the company Supercomputing Systems AG (SCS). The objective of the Swiss-Tx project is to build a parallel machine having a peak performance of up to 1 Teraflop/s. Today's technology can provide processors of a peak performance of 1 Gflop/s (such as the DEC-Alpha 21264). For a parallel one Teraflop/s computer, 1000 one Gflop/s processors have to be interconnected using the high bandwidth, low latency 12x12 T-NET switches. We make the assumptions that we need one link per processor. This assumption leads to the following possibilities:

- P=2 processors by PU, a topology of degree d=10 and a size of N=500
- P=4 processors by PU, a topology of degree d=8 and a size of N=250
- P=6 processors by PU, a topology of degree d=6 and a size of N=167
- P=8 processors by PU, a topology of degree d=4 and a size of N=125

All these situations can be realised with circulant graphs; no one can exactly be realised with fat-tree topology.

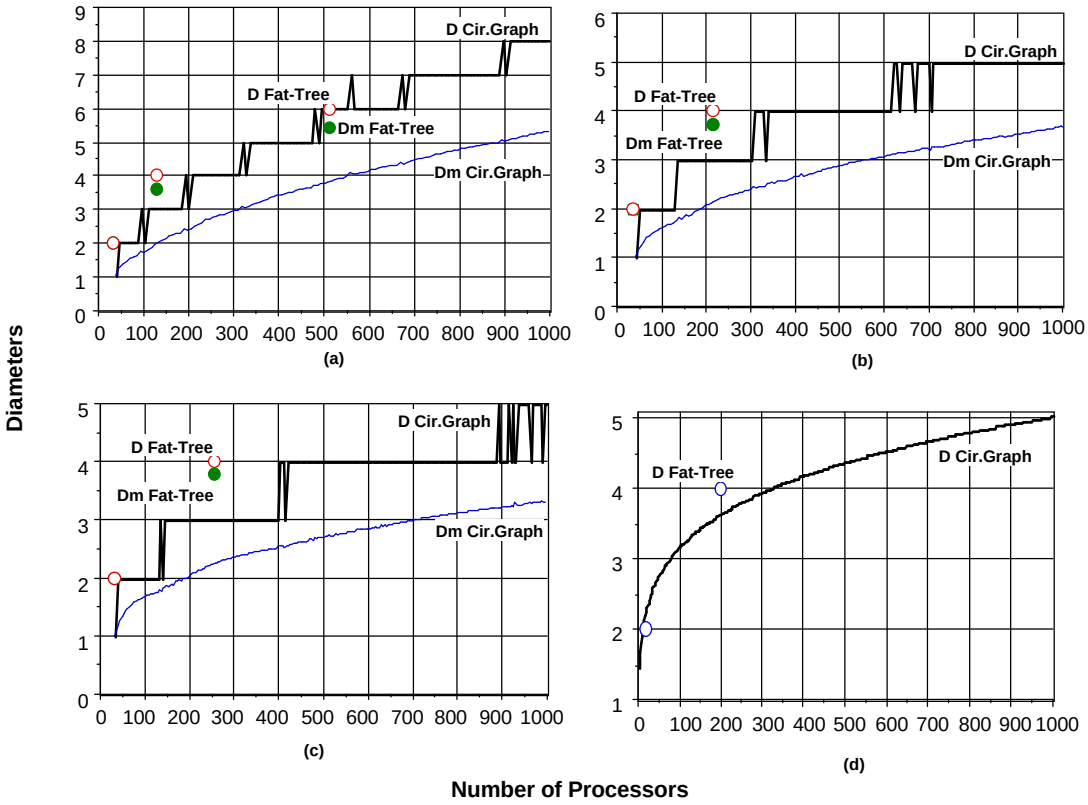

Figure 6: Diameter (D) and average diameter (Dm) of interconnection networks built using crossbar switches for degrees of 4 (a), 6 (b), 8(c) and 10 (d).

Figure 6 compares the diameters of possible solutions using fat-trees and circulant graphs. For degrees 4, 6 and 8 the results are measured values, for degree 10 results are based on an approximate formula. Possible solutions using fat-trees are indicated with a circle. It clearly appears that circulant graphs always have a diameter smaller or equal to the one of fat-trees. Nevertheless, fat-trees have a better bisectional width.

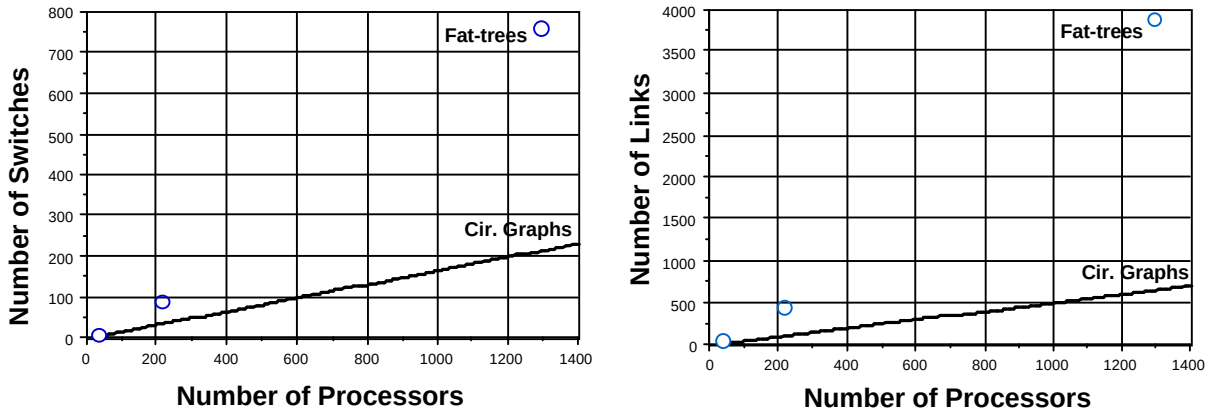

Figure 7: Cost of interconnection networks of degree 6 built using 12x12 crossbar switches

At this stage of our analysis it is still difficult to choose between circulant graphs and fat-trees. Circulant graphs are much more flexible, have most of the time a better diameter and always have a better average diameter whereas fat-trees have a better bisectional width. The last criterion we have to analyse is the cost of the network.

The cost of a massively parallel machine is roughly composed of the cost of the PUs and the cost of the communication network. Since N*P is chosen to be the same in all the cases, the cost for the PUs is the same for fat-trees and circulant graphs. The cost of the network is directly related to the number of switches and links necessary to build it. Fat-trees have the following property: if we use nxn crossbars to realise the switch nodes and the computing nodes then the degree of the fat-tree must be n/2 (see Figure 2). Therefore, if the available technology is 12x12 crossbars, we are limited to a degree of 6.

Figure 7 shows the cost of fat-trees and circulant graphs of degree 6 for a number of processors up to 1400. The needed links and crossbars increase more rapidly for fat-trees than for circulant graphs. Let us compare possible solutions that are the closest to 1000 processors:

- Fat-Tree: 1296 processors, a diameter of 6, an average diameter of 5.63, a bisectional width of 648, 756 crossbars and 3888 links.
- Circulant graph: 1002 processors, a diameter of 5, an average diameter of 3.68, a bisectional width of 96, 167 crossbars and 501 interconnection links.

Of course the solution with a fat-tree has a much better bisectional width. But two remarks can be made:

1. The high bisectional width has its price. Thus, the ratios of the number of switches and links between the fat-tree solution and the circulant graph solution are 4.53 and 7.77, respectively.
2. The number of processors in a fat-tree must be a power of 6. Using circulant graphs, the number of processors must be dividable by 6. Thus, there always exists a solution close to a given number of processors.

Because of their flexibility and of their erratic behaviour of the bisectional width, circulant graphs can always exhibit solutions having a good bisectional width for a number of processors close to a given value. In our case we can mention two possibilities:

1. 1260 processors, a diameter of 6, an average diameter of 3.98, a bisectional width of 362, 210 crossbars and 630 links.
2. 1356 processors, a diameter of 6, an average diameter of 4.07, a bisectional width of 394, 226 crossbars and 678 links.

Nevertheless, these two solutions still have a quite low bisectional width compared with the one for the fat-tree solution. But, as using circulant graphs we are not limited to a degree 6, we can also build solutions using a circulant graph of degree 8 (with 4 processors on each switch). Doing this we find the following solution:

- 1192 processors, a diameter of 5, an average diameter of 3.47, a bisectional width of 588, 298 crossbars and 1192 links.

In comparison with the solution of 1296 processors using a fat-tree, this solution exhibits a better diameter and average diameter, uses 2.5 times less crossbars, 3,26 less links and achieves a bisectional width per processor of 0.49 which almost the same that the fat-tree solution (0.5).

Nevertheless, if the only important parameter is the bisectional width, regardless any other considerations and particularly the cost, we can design the interconnection network using circulant graphs of degree 10 with two processors per PU. Below are the characteristics of some examples of these possible solutions:

- 1000 processors, a diameter of 5, an average diameter of 3.54, a bisectional width of 570, 500 crossbars and 2500 links.
- 1020 processors, a diameter of 5, an average diameter of 3.57, a bisectional width of 642, 510 crossbars and 2550 links.
- 1080 processors, a diameter of 5, an average diameter of 3.61, a bisectional width of 778, 540 crossbars and 2700 links.

For these solutions the bisectional width per processor is 0.57, 0.63 and 0.72 respectively which is better than the solution using a fat-tree. Moreover they use significantly less crossbar switches and links.

As a conclusion we can say that the fat-tree is a topology especially designed for exclusively building very high performance interconnection network. The circulant graph topology allows to adapt the performance of the interconnection network to the user needs and, if needed, it allows to obtain performance equivalent or better to fat-tree for a lower cost. Moreover, the fat-tree topology is a very rigid topology, it cannot fully benefit from the increasing size of the crossbar switch technology, since the degree of a fat-tree must correspond to half of the crossbar size, and the number of PUs must be a power of the degree. With circulant graphs, networks of any numbers of PUs can be built.

For all those reasons we decided to use circulant graphs for building the interconnection network of the Swiss-Tx computers series. In the next section we present the architecture of the Swiss-T1 computer which is based on a circulant graph.

3. Swiss-T1 configuration

Hardware configuration

The first prototype Swiss-T1 machine will consist of 8 PUs connected using the T-NET 12x12 crossbar switches. Each PU consists of 4 dual processor, Alpha-based, DN20 servers, 2 links are used per server, and 8 links per PU. The four remaining links are used to connect the 8 PUs through the circulant graph: $C_8<1,3>$ (Fig.

8). The diameter is 2, the average diameter is 1.43 and the bisectional width is 8. It has to be noted that we obtain a K-Ring for this particular case. An efficient routing for communications between all PUs is given in Table 1. This table has to be read in the following manner: for a direct link between the nodes, the routing number is identical to the destination node, for an indirect connection, the number denotes the crossbar number through which the routing passes. In Fig. 8 the numbers on the links indicate the number of packages that have to be sent in both directions for such an all-to-all message passing operation. The routing table has been set up to well distribute the charge on the different links during an all-to-all global communication.

The architecture is completed by a frontend consisting of 2 dual processor DN20 servers. The frontend is connected to the computing nodes through the Gigabit Ethernet/Fast Ethernet switching system.

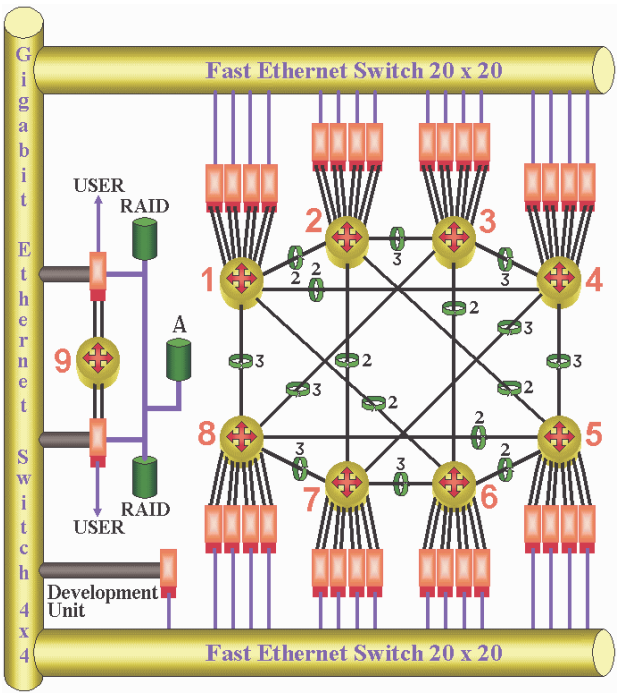

Figure 8: Swiss-T1 architecture based on Alpha 21264 dual-processors. The 8 PUs consist of 4 dual processor boxes. They are connected by a 12x12 crossbar switch, called T-NET. Each box, represented by a red rectangle, is connected by two links, the remaining 4 links connect to other crossbars. The diameter is D=2. For an all-to-all communication 3 bidirectional data exchanges are needed at most. The numbers on the links denote the number of these exchanges between these links. A Gigabit Ethernet/Fast Ethernet switching system directly connects the boxes to the frontend. An archive robot of 1 Tbytes and 2 RAID disks can be accessed by the 2 frontend boxes in a symmetric manner. There is a special development machine connected to the Fast Ethernet and the Gigabit Ethernet.

Table 1: Routing table for the Swiss-T1 machine

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
1	-	2	2	4	4	6	8	8
2	1	-	3	7	5	3	7	5
3	2	2	-	4	4	6	8	8
4	1	7	3	-	5	7	7	3
5	4	2	4	4	-	6	6	8
6	1	3	3	7	5	-	7	1
7	8	2	8	4	6	6	-	8
8	1	5	3	3	5	1	7	-

The detailed hardware and software specifications are:

- Swiss-T1 consists of 8 PUs each one including one 12x12 crossbar and 4 Alpha 21264 dual-processor boxes running at 667 MHz, giving 84 Gflops peak performance. One box has 1 Gbytes of main memory and 13 Gbytes of local disk space
- Each box is connected to the 12x12 crossbar by two bidirectional 100 Mbyte/s links through PCI adapters
- Four links interconnect the crossbars. The communication configuration is the circulant graph C_8<1,3>
- For an all-to-all communication, up to three messages in both directions have to be sent between crossbars if one follows the routing table given in Table 1
- There is one frontend node consisting of 2 Alpha 21264 dual-processor boxes running at 667 MHz. One box includes 2 Gbytes of main memory and 13 Gbytes of local disk space. During the installation phase it will

be connected only through the Gigabit Ethernet switch, in a second phase, it will be directly linked to the T-NET as well

- Two 20x20 Fast Ethernet switches interconnect the upper and lower half of the computational boxes. They are connected to the frontend through the 4x4 Gigabit Ethernet switch
- A RAID disk system of 300 Gbytes is connected to the frontend boxes
- A one Terabytes archive robot system is connected to the frontend
- The remaining frontend crossbar links can be used to interconnect other units
- There is also a development box separated from the production machine

Software configuration

The different important software packages to be installed on the Swiss-T1 are (with * are marked the programs that will be available at T1 installation time):

Basic software in each dual processor box

*Digital Unix	Compaq	Operating system in each box
*F77/F90	Compaq	Fortran compilers
*HPF	Compaq	High performance Fortran
*C/C++	Compaq	C and C++ compilers
*DXML	Compaq	Digital math library in each box
*MPI	Compaq	SMP message passing interface from Compaq (only usable in a box)
*Posix threads	Compaq	Threading in a box
*OpenMP	Compaq	Multiprocessor usage in a box through directives
*KAP-F	Kuck Ass. Inc.	To parallelise a Fortran code in a multiprocessor box (preceeds OpenMP)
*KAP-C	Kuck Ass. Inc.	To parallelise a C program in a multiprocessor box (preceeds OpenMP)
Software to pass messages between the boxes and to use them in parallel		
*LSF	Platform/SIC-EPFL	Load Sharing Facility for resource management
MONITOR	SIC-EPFL	Monitoring of system parameters
*Totalview	Dolphin	Parallel debugger
*Paradyn	Madison/CSCS	Profiler to help parallelising programs
*MPI-1/FCI	SCS AG	Message passing interface between boxes running over TNET
MPI I/O	SCS/LSP-EPFL	Message passing interface for I/O
*MPICH	Argonne	Message passing interface running over Fast Ethernet
*PVM	UTK	Parallel virtual machine running over Fast Ethernet
*BLACS	UTK	Basic linear algebra subroutines
*ScaLAPACK	UTK	Linear algebra matrix solvers
NAG	NAG	Math library package
Ensight	Ensight/CSCS	4D visualisation
MEMCOM	SMR SA	Data management system for distributed architectures

Conclusions

The fat tree topology was especially designed to build a very high bandwidth network. As the fat-tree is a topology derived from trees, the diameter grows optimally with the logarithm of the number of PUs. The drawback of this topology is the extreme rigidity, the high cost, and the very high number of links.

We expect, but we have not prove yet, that the diameter of circulant graphs grows as $\sqrt[d/2]{N}$ which is, on a theoretical point of view, not as good as fat-trees. Nevertheless due to its great flexibility we can fully benefit from the numerous possibilities offered by the use of the crossbar technology. Consequently, in the practice, it is always possible to find a solution using circulant graphs which has better characteristics for a lower cost than the ones using fat-tree. Moreover by using circulant graph we can adapt the performance of the network to user needs. For a given number of processors and a given crossbar switch technology, we can choose the performance of the network. If, subsequently, the user needs to increase this performance we can increase the degree of the circulant graph without changing the number of processors. The opposite modification is also possible, we can

increase the number of processors without changing the degree of the circulant graph. This flexibility which is not possible with other topologies, allows us to optimise the ratio price/performance according the user needs.

Acknowledgements

We would like to thank Martin Frey for continuous interaction to define the Swiss-T1 architecture and to Mario Romano for the graphical representation of it. The Swiss-Tx project is a co-operation between EPFL, ETHZ, CSCS, Supercomputing Systems and Compaq. It is financed by CTI (Commission for Technology and Innovation at Bern).

References

[Boesch84] Boesch F., Tindell R., "Circulants and their Connectivities", Journal of Graph Theory, vol 8, p.487-499, 1984

[Brauss99] Brauss, S., Frey, M., Gunzinger, A., Lienhard, M. and Nemecek J., "Swiss-Tx Communication Libraries", HPCN'99 (Amsterdam) and this issue

[Dubois98] Dubois-Pèlerin, Y., Gruber, R. and Swiss-Tx Group: "Swiss-Tx, First experiences on the T0 system", EPFL, Supercomputing Review, 10 (1998) 19-23 and http://capawww.epfl.ch/

[Kuonen95] Kuonen, P. "The K-Ring", Proceedings of the European Research Seminar on Advances in Distributed Systems (ERSADS), April 1995 .

[Kuonen99] Kuonen P., Gruber R., A. de Vita, and Volgers P., "Parallel computer architectures for commodity computing" keynote speech at High Performance Computing and Networking (HPCN) Europe, Amsterdam, April 12-14, 1999

[Leiserson85] Leiserson, C.E. "Fat-Tree, "Universal network for hardware-efficient supercomputing", IEEE Transactions on Computers, C-34, No. 10 (1985) 892-901

[Meiko94] "Communication Network Overview", http://www.meiko.com/info/NetworlOverview/Network/Overview.html.