Optimisation Of Extended Generalised Fat Tree Topologies

Adamantini Peratikou¹, Mo Adda¹

¹University of Portsmouth, School of Computing
Portsmouth , PO1 3HE
United Kingdom.
Adamantini.Peratikou@port.ac.uk,
Mo.Adda@port.ac.uk

Abstract. Extended generalised fat tree (XGFT) are interconnection networks with bidirectional multistage properties, (BMN) which can be extended or scaled to accommodate different system sizes and requirements. However, these extended topologies do not address power consumption and traffic constraints. In this paper, we extract a sub-set of the generalised fat tree topologies that are power consumption and performance aware. We called this sub-set optimised OXGFT. The cost which is proportional to the relative power is the objective function that is minimised based on the traffic constraints to maintain a lower delay and a higher throughput. The simulation results show that the extracted OXGFT topologies perform well under various load conditions.

Keywords: Fat-tree, Extended Generalised Fat tree, Optimisation, Interconnections, High Performance Architectures

1 Introduction

Fat tree networks [1] were proposed as binary tree based topologies. The only difference is that the processors of the fat tree are located to the leaves of a binary tree (figure 1), and the fact that moving upwards to the root of the tree the communication links increase and therefore the communication bandwidth increases as well. K-ary n tree architectures were later proposed with the difference that the upward links are quicker by a factor k than the downward links in order to achieve a non-changing bisection bandwidth. Fat trees have a constraint that when implementing them the switch port rates become too high near the root of the tree, thus the use of switches with the same radix and port speed is inevitable.

The most suitable candidate for the fat tree topologies is k-ary n-trees as it allows the switches to be configured at all levels in a similarly way as illustrated in figure 1 (c). In fat trees the system size depends on the degree of the switch such as $p=2^n$, where n is the height of the level, while in K-ary n-trees the descendants of the node are at index positions therefore $p=k^n$.

Fig. 1. a) 4-level fat tree b) binary 4 tree[2] c) K-ary n-tree with $k = 3$ and $n = 3$

Topologies based on fat tree that do not have full bisectional bandwidth are called extended fat trees such as m-ary trees [2].

While fat tree class topologies include some interesting characteristics they also experience some known issues such as the bottleneck caused from the limited availability of paths as in some cases only a single path exists. Extended generalised fat tree or XGFT was proposed by [2] as an optimisation of the standard fat tree topologies. Extended generalised fat tree or XGFT [3] on the other

Fig. 2. a) XGFT configuration of $(2,6,6,4,0)$ with 36 processors and b) XGFT $(3,4,3,5,2,2,2)$ with 60 processors [3]

hand, unlike k-ary n-tree, are interconnections that can be extended, or scaled to accommodate different system sizes and requirements. Switches in various stages of the network have different number of bi-directional ports. Like k-ary

n-tree, extended generalised fat trees can be regenerated recursively to accommodate a larger system, and the connectivity along with the links used, depends on the configuration requirements. Figure 2 illustrates two examples of XGFT each with different number of routing switches and number of leaf nodes. Each different stage of routing switches from top switch to the bottom switch is considered to have different levels of switches, with each level consisting different sub-trees.

The simulation results reported in [3], illustrated in figure 3, showed that better performance was achieved with higher number of Turn back channels. However, the extension of the routing algorithm proposed in XFGT [4] does not provide any performance enhancements, thus the added complexity that is introduced in the configuration of XGFT is unnecessary. This can be proved by our future work where we propose a different generalisation. The addressing

Switch type and the number of TB-channels (TBC)	Routing Algorithm	Max. Average Throughput [%]	Max. Average Latency [time slots]
Dual with 1 TBC (DUAL/1)	TB	8.18	415.0
Dual with 1 TBC (DUAL/1)	TBWP	17.8	196.7
Dual with 2 TBCs (DUAL/2)	TBWP	20.2	186.6
Dual with 3 TBCs (DUAL/3)	TBWP	22.3	175.4
Mega with no TBCs (MEGA)	TB	23.1	145.6

Fig. 3. a) Simulation results of XGFT with TB(Turn back routing) and TBWP (turn back when possible. [3]

used is based on the space encoding for TB (Turn Back) and TBWP (Turn back when possible) routing algorithms [3]. The addresses are of integer vectors which specify the routing path, both the source address and the destination address are attached into the packet header to ensure correct shortest path calculations (figure 4). Two routing options exist, the up routing and the down routing. In the up routing, a packet is routed in the upwards direction until it finds the common routing switch ancestor or reaches the root switch of the destination node. The common ancestor switch can be found, by comparing the destination address with the source address carried along with the packet at each switch stage of the network. Once the ancestor switch is reached, then the down routing checks the destination address to determine the proper port to route the packet through to reach its destination. This routing is entirely deterministic, whereas the up routing is adaptive

Fig. 4. Packet structure [3]

2 Optimal Configuration

To find the optimal configuration among the space of all the endless XGFTs topologies, a simulator was developed that uses the power consumption as the objective functions with traffic constraints. That simulator takes the number of processors and runs a set of constrain in order to find the optimum configuration of both sub-trees and routing switches to be set to achieve the higher performance possible with the lower cost.

2.1 Objective Function

The objective function is proportional to the power consumption. By minimising the cost of the architecture, one can minimise the power associated with it. The connectivity cost for each level depends on the number of ports, the number of sub-trees, and the number of routing switches. Overall the cost of the XGFT of level n can be expressed as

$$Cost = \sum_{i=1}^n R_i^T \times L_i^2 \quad (1)$$

where R_i^T is the total number of routing switches at level i and L_i is the total number of ports per routing switch at level i. The total number of routing switches and ports per routing switch at level i, which determines the complexity of the level and hence the total complexity of the network, can be defined by the following two equations:

$$R_i^T = R_i \prod_{j=i+1}^n S_j \quad (2) \quad L_i = S_i + \frac{R_{i+1}}{R_i} \quad (3)$$

Where S_i is the number of sub-trees, and is the number of routing switches per sub-tree at level i.

2.2 Constrains

The cost equation 1 is minimised subject to several constraints that ensure a high performance for the topology. This is achieved by setting the number of routing switches and ports per switch and per level to an adequate number that guarantees an overall minimum latency. The total number of processors is set as a constraint among the sub-trees which is defined in the following equation.

$$P = \prod_{i=1}^n S_i \rightarrow P = S_i \times S_{i+1} \times \dots \times S_n \quad (4)$$

The connectivity constrains can be illustrated in the two following equations. The ratio between routing switches of different levels has to be a positive integer as it defines the number of ports per routing switch. The number of the routing switches per sub-tree has to increase from leafs to the root to satisfy the connectivity requirements of a fat tree.

$$\frac{R_{i+1}}{R_i} \in \mathbb{Z}^+ \quad (5) \quad R_i \leq R_{i+1} \leq R_{i+2} \leq \dots \leq R_n \quad (6)$$

The number of sub-trees per level is the most important constrain. The sub-trees and the routing switches are related to make sure that the numbers of ports per level are adequate enough to fully connect the number of sub-trees per level and hence minimise the delay in the network. This equation is based on queuing theory and can be simplified into:

$$S_i \leq \frac{R_{i+1}}{\prod_{j=1}^n S_j} \quad (7)$$

One can also include another constraint to relate to the current technology which requires the maximum number of ports supported by a given switch at any level.

3 Performance Analysis

For the purpose of this research two simulators were implemented. The first one in C++ called m: Z-node that can simulate multiple fat tree topologies with multiple groups of levels and sub-trees along with the option of adjusting the properties of the channel links, routing, and applications patterns, and the second one called SimOpt in VB and Excel that takes a set of constraints and produces an optimal topology based on equation 1. Two configurations of XGFT [4] with different processors and number of levels were compared to their optimal versions obtained from our optimisation simulator. Figure 2 (a) illustrates a two-level configuration of XGFT, which consists of 36 processing elements, divided into groups of 6, with each group connecting to an ancestor switch. The total number of ancestor switches for the first and second levels are 6 and 4 respectively. Figure 2 (b) shows a three level configuration with 60 processors.

According to the constraints and the objective function discussed above, both configurations do not satisfy the requirements for high performance based on the given number of processors. Their optimised versions for the same number of processors and levels are shown in Figure 5. However, for 60 processors the optimum configuration pays a smaller price for power consumption as illustrated in figure 6 compared to the non-optimised shown in figure 3, at the achievement of better performance, as we will demonstrate later.

Fig. 5. a) Optimal configuration for 36 processors. b) Optimal configuration for 60 processors.

Fig. 6. Power reduction with 60 processors in XGFT and OXGFT.

3.1 Discussion Of the Optimal Configuration of XGFT

The optimal configuration for 36 processors with two levels, according to our assumptions, consists of a total of 12 switches for the first level, with each switch connected to 3 processors, and 3 switches for the second level (figure 5 (a)).

XGFT configuration and optimal configuration on 36 processors were tested under various offered traffic load. Both the configurations performed similarly on a load of 5% to 50%, with the optimum configuration achieving slightly lower message delay of -1.00 to -2.00 nanoseconds compared to XGFT (figure 7 (a)). When the load increases to 60% the difference in throughput between the two configurations becomes noticeable (Figure 7 (b)), and the message delay becomes significantly higher in XGFT. This is due to the lack of ports interconnections to service all the backed traffic at each level.

Fig. 7. a) Message delay on various input rates b) Throughput under on various input rates

Figure 8 indicates that the message delay is lower under all traffic patterns in the optimum configuration with the exception of Round Robin where both configurations have equal values. The throughput is equal on both cases except on bit reversal traffic where in the OXGFT configuration is slightly higher.

The XGFT configuration (figure 2 (b)) against optimum XGFT configuration on 60 processors was also tested, the configuration of the optimum XGFT consists of 20, 30 and 6 switches for levels 1, 2 and 3 respectively (figure 5 (b)). While the XGFT configuration consists of the switching elements illustrated in figure 2 (b).

Fig. 8. a) Message delay on 36 processors under different applications. b) Throughput on 60 processors under various applications.

Under various normalised loads (figure 9) , it is identified that even on higher number of processors the optimum configuration still overcomes the XGFT configuration in all the different loads. The difference in the message delay between the two configurations is even higher. The message delay obtained in XGFT shows a significant increase in Complement traffic while the OXGFT (optimum) follows amore constant pattern against all traffics with an increase towards hotspot,

Fig. 9. a) Message delay under various normalised input rates on 60 processors b) Throughput under various normalised input rates on 60 processors

Based on the simulation results illustrated in figure 10, it is identified that the optimum configuration performs significantly better on 60 processors than the non-optimised XGFT. The throughput is higher in the optimum configuration under most of the traffic patterns, with the exception of hotspot traffic in which both configuration performed similarly (figure 10(b)). The message delay obtained in the optimum configuration follows a straight pattern with an increase towards transpose. While in XGFT the message delay increases enormously under complement and transpose traffic (figure 10 (a)).

Fig. 10. a) Message delay on 60 processors under various applications b) Throughput on 60 processor configuration under various applications

4 Conclusion

In this paper we have extracted an optimal configuration from a set of endless extended generalised fat tree topologies. The impact of this paper is the identification of an optimised configuration that will give a high performance structure at the expense of small power consumption. From the results obtained it has been verified that the optimum configuration has great performance rewards for various traffic patterns and loads. This paper is presented as an introduction of the optimisation of extended fat tree topologies, with the aim to exploit the optimisation of all fat tree class topologies in future work.

References

1. Leiserson C.E. Fat-trees: Universal networks for hardware efficient supercomputing. //IEEE transactions on Computers. 1985 C-34(10):892901.
2. Minkenberg C., Ronald P, Luijten, Germn Rodrguez.On the optimum switch radix in fat tree networks. //IEEE 12th International Conference on High Performance Switching and Routing.2011. P.44-51
3. Kariniemi H., Nurmi J.Performance Evaluation and Implementation of two Adaptive Routing Algorithms for XGFT Networks. //In Computing and Informatics. 2004. Vol. 23, no. 5-6 p. 415-435.
4. Kariniemi H.On-Line Reconfigurable Extended Generalised Fat tree XGFT Network-on-Chip for Multiprocessor System-On-Chip.//Tampere University of technology Publication 614.2006. ISBN 952-15-1746-8.