2011 IEEE International Symposium on Network Computing and Applications

Evaluation of an Alternative for Increasing Switch Radix

Juan A. Villar, Francisco J. Andújar, José L. Sánchez, Francisco J. Alfaro
University of Castilla-La Mancha, Spain
Email: {juanan,fandujar,jsanchez,falfaro}@dsi.uclm.es

José Duato
Technical University of Valencia, Spain
Email: jduato@disca.upv.es

***Abstract*—In large switch-based interconnection networks, increasing the switch radix results in a decrease in the total number of network components. In this paper we evaluate an interesting strategy for building high-radix switches going beyond the integration scale bounds. This approach is independent of the evolution of single-chip switches and will remain valid as integration scale keeps evolving. Simulation results show that with a correct internal switch design, this kind of switches achieves almost the same performance as single-chip switches with the same radix, which would be unfeasible with current integration scale.**

I. Introduction

Often, the interconnection network is the subsystem that a more custom design requires. For instance, Tianhe-1A supercomputer [1], number one in the Top500 (Nov.11), is composed of standard processors and a custom network that removes the interconnect bottleneck, and significantly contributes to the high global performance of Tianhe-1A.

The design of switches with a high number of ports is an attractive option to improve the performance and reduce the cost of the interconnection network, specially for large multiprocessor systems. In any case, switch size constraints are mainly determined by the current integration scale and package pin count.

It is possible to build high-radix switches by combining several low-radix switches. This strategy makes possible to eventually overtake integration technology and dramatically shorten the time-to-market. Note that this will remain valid as integration technology continues evolving.

An important consequence of this strategy for building larger switches is that the resulting switch will no longer be homogeneous. Switch performance will vary depending on the internal configuration. The internal switches interconnection can become a potential bottleneck if they have to support most of the traffic handled in the switch. Therefore, it is essential to minimize the impact of this bottleneck, otherwise the latency on the network will be increased. Thus, the switch-level connection pattern[1] (SCP) becomes an important design decision in the construction of this kind of switches. An arbitrary pattern may produce a significant performance degradation. Consequently, it is necessary to determine the most convenient pattern for internal connections in order to obtain the best switch performance.

The paper is organized as follows: In Section II details about this kind of high-radix switches are given. Section III provides the performance evaluation. Section IV reviews the related work. Some conclusions are outlined in Section V.

II. High-Radix Switches by Combining Low-Radix Switches

As mentioned above, it is possible to build high-radix switches by combining several low-radix switches. This strategy makes possible to eventually overtake integration technology and dramatically shorten the time-to-market. Note that this will remain valid as integration technology continues evolving.

This strategy opens a series of new issues that must be addressed so that it would be implemented in an efficient manner. In the next sections, we briefly overview these issues that we have analyzed in [2] in a more formal way. Nevertheless, in this paper we provide an experimental quantification of their influence on network performance.

Definition 1. *A Combined switch, or simply C-switch, is a switch formed by several smaller interconnected switches (internal switches). The ports being offered by a C-switch are obtained from free ports of its internal switches after they are interconnected.*

An interesting case is that where C-switches are built from only two identical internal switches. This subclass of C-switches still offers an important increase in the number of ports while the interconnection between the two internal switches is the simplest one.

Definition 2. *A Twin switch, or simply T-switch, is a switch formed by two identical smaller interconnected switches. The*

This work was supported by the Spanish MEC and MICINN as well as European Comission FEDER funds, under Grants "Consolider Ingenio-2010 CSD2006-00046" and "TIN2009-14475-C04", respectively; it was also partly supported by Junta de Comunidades de Castilla-La Mancha under grants "PEII11-0229-2343" and "POII10-0289–724".

[1]We distinguish between *network-level connection pattern* and *switch-level connection pattern*. The former is the traditional interconnection pattern connecting switch-based networks (e.g., *butterfly* permutation in multistage interconnection networks); and the latter refers to how the ports of this kind of high-radix switch ports are mapped to the ports of the internal switches.

978-0-7695-4489-2/11 $26.00 © 2011 IEEE
DOI 10.1109/NCA.2011.56

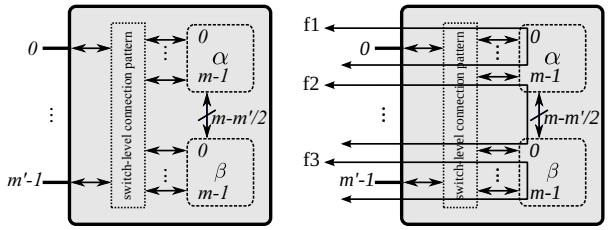

Figure 1. T-switch: (a) basic block diagram, and (b) common case.

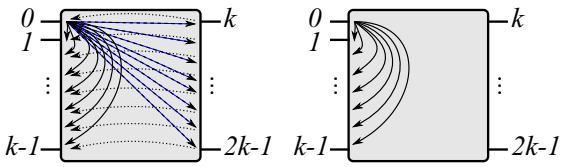

Figure 2. Connections required for uniform traffic pattern on each T-switch of (a) Type A, and (b) Type B.

ports being offered by a T-switch are obtained from free ports of its two internal switches after they are interconnected.

For instance, a m'-port switch consisting of two identical m-port switches ($m'/2 < m < m'$) can internally interconnect each other using $m - m'/2$ ports, using the remaining ports for external connections (Figure 1a), where internal switches are denoted as α and β. Although T-switches seem to be simple, there are significant challenges in its design. Two of them stand out especially: to obtain the appropriate SCP of internal subnetwork, and to determine the adequate number of ports used to interconnect α and β switches.

The SCP has an important influence on packet latency. Time to cross the T-switch will be minimal when only one internal (α or β) switch is used (flows f1 and f3 in Figure 1b). The bad case is obtained when both internal switches are used for every path crossing the T-switch (f2 in Figure 1b). To obtain the best case is not trivial and an in-depth study is required, in which several factors, e.g., network topology, routing and traffic pattern must be considered. In such situations, the main objective in the SCP design of T-switches is to minimize the use of the ports interconnecting internal switches.

Regarding the second challenge, the number of internal ports must be that which avoids the creation of the internal bottleneck between the α and β switches. Obviously, this number and the SCP have a clear interdependence.

A. C-Switch Configuration Methodology

Our methodology for determining the optimal SCP of C-switches consists of the following main steps [2]:

1) Network paths analysis. The purpose of this step is to determine the connections required in each C-switch at network level and the number of times all these connections are used taking into account all the possible paths used by the packets.
2) Switch classification. Depending on the network characteristics and load conditions, few or many different C-switch configurations could be obtained. In this step, C-switches are grouped according to their connection requirements, and so, several types of C-switch will be distinguished.
3) Switch-level connection pattern. From connection requirements and given the number of internal switches forming the C-switch, this last step consists in finding the optimal configuration for each class of C-switch.

The first two steps of this methodology are independent of the internal C-switch structure. Obviously, the third step depends on the structure of the C-switch.

Notice that although the methodology is general and can be applied considering different C-switches and interconnection networks, the C-switch configuration depends on the particular network properties. Therefore, in order to apply this methodology, network type, topology, routing algorithm and traffic pattern must be specified.

B. Methodology Application to a Particular Case

We must specify the network characteristics (i.e., topology, routing, and traffic), and then, we apply the methodology. We have chosen the k-ary n-tree BMIN network [3], a subclass of fat-trees, which is one of the most common topologies today in the largest supercomputers on the Top500 list. Regarding C-switches, we have used T-switches. We are considering T-switches for purely illustrative purpose that makes the formal development simpler and easier to understand, but the approach is general enough to be apply to any C-switch.

The routing algorithm is DESTRO [4], which is a deterministic routing algorithm for fat-trees. We have selected DESTRO because of its interesting features: simple hardware implementation, reduced routing time and in-order packet delivery. Moreover, it is able to evenly balance network traffic and reduce network contention. We have considered uniform traffic pattern because it is commonly used in network performance evaluations.

In [2] we apply the methodology to the mentioned network characteristics. As result, we obtain the optimal configuration for all the T-switches in the network. The results of the three steps are below:

1) Network paths analysis. We can obtain this information distinguishing three different cases, which correspond with forward, backward, and turnaround connections in a BMIN. Figure 2 shows the paths passing through the T-switch using the input port zero in the ascending and turnaround phases. The same paths are obtained for the remaining input ports, but they are not shown

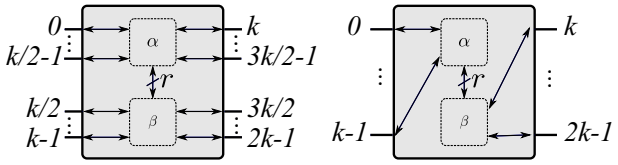

Figure 3. Optimal switch-level connection pattern for each T-switches of (a) Type A, and (b) Type B.

in the figure for the sake of clarity. The paths in the descending phase are independent of the other phases.

2) Switch classification. Figure 2 shows the two types of T-switches are identified: all the T-switches are Type A (Figure 2a) with the exception of those placed at the last-stage in the BMIN. Thus, the T-switches of Type B (Figure 2b) correspond with the last-stage T-switches.
3) Switch configuration. In this case, it is not possible to have a global optimal SCP and two different SCPs are needed (Figure 3). One of them (Figure 3a) is an optimal SCP for the T-switches of Type A, and the other (Figure 3b) is an optimal SCP for T-switches of Type B in the network.

Once the particular configuration has been set up in each switch, it is interesting to know how the network behaves when it is stressed by traffic. We will study this behavior in the following section.

III. Performance Evaluation

The evaluation has been driven by simulation and focused on the two key issues discussed before: switch-level connection pattern and bandwidth of interconnection between internal switches.

A. Simulation Model

To perform this evaluation, we have chosen a representative case: the k-ary n-tree topology connecting 4096 end-nodes; DESTRO deterministic routing algorithm; and uniform traffic pattern.

We have simulated 32-port T-switches that are formed by two internal 24-port switches such that 16 ports are used for external ports and 8 ports for interconnecting the two internal switches. Other assumptions are an input queued switch design with virtual output queuing at switch level; 1 GByte/s full-duplex link bandwidth; round-robin scheduler per output port; 2 KBytes packet size; every queue buffer has capacity of 64 packets, among others.

B. Case Studies

We have estimated the performance of the network described in Section III-A, considering T-switches with the SCP that has been proved in [2] to be the optimal for T-switches under the simulated conditions (this network will be referred to as O-BMIN in the figures). Then, we have compared the performance of this network with that offered by an equivalent network considering single-chip switches with the same radix as T-switches. This network will be referred to as U-BMIN and it represents the upper performance threshold.

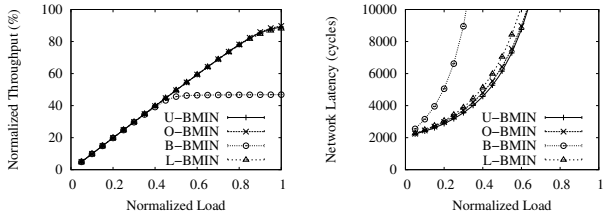

Figure 4. (a) Throughput and (b) latency for the different configurations.

In order to show the importance of the SCP we also consider another case which represents an arbitrary/bad SCP, labeled as B-BMIN in the figures. O-BMIN and B-BMIN networks use T-switches with the optimal SCP and an arbitrary/bad SCP, respectively. The optimal SCPs for every type can be seen in Figure 3. Additionally, Figure 3b is considered as the arbitrary/bad SCP for all the $k \times k$ T-switches in the network. All the T-switches consider r internal ports, respectively. Also, we include a fourth case: a network connecting the same number of end-nodes, but with single-chip switches, which would be feasible with the current integration scale, namely L-BMIN.

C. Simulation Results

We have run two different tests. The first test has the aim of measuring the performance gap between the aforementioned case studies.

The figures of this section show the average values and the confidence intervals at 95% confidence level of thirty different simulations performed at a given input load.

Figure 4 shows throughput and latency results for all the network configurations. Some interesting details can be observed:

- When optimal switch-level connection pattern is considered, the T-switch based network (O-BMIN) is able to offer similar performance to high-radix single-switch based network (U-BMIN). Therefore, combined switches are confirmed as a good alternative to build high-radix switches.
- However, the B-BMIN network reaches the saturation point at only 40% of normalized network load. The different performance obtained is due to the fact that in these T-switches much of the traffic needs an extra hop with the bad SCP. So, it is also confirmed that it is essential to determine in each case the optimal switch-level connection pattern for combined switches.
- When U-BMIN and L-BMIN are compared, we have noticed that latency of L-BMIN is higher than latency of U-BMIN (e.g., 14% at the 50% of network load). This increase is due to L-BMIN has one more stage than U-BMIN, so the paths are longer.

- Also, we can see that the number of ports devoted for interconnecting the internal α and β switches does not produce any performance degradation.

We have performed a second test that measures how the number of internal ports influences on the performance.

In Figure 5 we can see the network throughput results when network load is at 80% for all network configurations (except L-BMIN). In order to perform the second test, we have instrumented the simulator to vary the number of ports for interconnecting the internal switches from 1 to 16, while the number of external ports remained the same.

Similarly, we can notice more relevant details:

- We can determine the minimum number of ports of internal switches devoted to interconnect them in order to avoid the creation of a new bottleneck. We can observe that the switch-level connection pattern is the key factor that influences the global performance, and the number of ports is not so relevant.
- When the number of ports is over four, the O-BMIN network achieves the same performance as the U-BMIN. However, when the B-BMIN network is considered, the internal bottleneck does not disappear until the number of internal ports is equal to the T-switch arity ($k = 16$).
- T-switches shown in Figure 4 have eight ports, but when the T-switch has the optimal SCP only four ports are enough to achieve the maximum performance. In such a case, these four extra ports could be used for external communication.

IV. Related Work

In this section we review of existing proposals in the literature of high-radix switches, which are mainly focused on solving the scaling problems.

The YARC is the high-radix switch used by the Cray BlackWidow [5]. Its internal organization is defined in [6] and is based on increasing the number of ports considering more thin channels instead of less wide channels.

The Partitioned Crossbar Input Queued [7] is a more recent proposal for the internal organization of high-radix switches, and deals with one of the main constraints in high-radix switch design, the excessive memory requirements.

Regarding the alternative for building high-radix switches using low-radix switches, the Oracles's Sun Blade 6048 Infiniband QDR Switched Network Express Module (NEM) [8] has already implemented this strategy, offering the ability to connect up to 12 dual-node blades in a single shelf.

On the other hand, the Dragonfly topology [9] uses a group of routers as a virtual router to increase the effective radix of the network.

To the best of our knowledge, there are currently no published formal studies for obtaining the optimal SCP of high-radix switches composed of several internal single-chip switches interconnected each other.

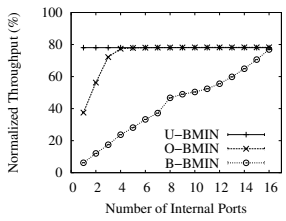

Figure 5. Network throughput in function of the number of internal ports.

V. Conclusions

In this paper we have described an interesting alternative for building high-radix switches. We have also obtained the optimal configuration for T-switches to avoid the internal bottleneck by applying our methodology to a particular network characteristics. To check the potential of this alternative, we have evaluated the performance of networks based on combined switches composed of two smaller identical switches. The results are similar to those obtained with the same networks, but using single-chip switches with the same number of ports as combined switches.

References

[1] J. J. Dongarra, H. W. Meuer, and E. Strohmaier, "TOP500 Supercomputer Sites," Nov. 2010.

[2] J. A. Villar, F. J. Andújar, F. J. Alfaro, J. L. Sánchez, and J. Duato, "An alternative for building high-radix switches: Formalization and configuration methodology," Department of Computing Systems. University of Castilla-La Mancha, Tech. Rep. DIAB-11-02-1, 2011, www.dsi.uclm.es/descargas/thecnicalreports/DIAB-11-02-1/diab-11-02-1.pdf.

[3] F. Petrini and M. Vanneschi, "K-ary n-trees: High performance networks for massively parallel architectures," TR., 1995.

[4] C. Gómez, F. Gilabert, M. E. Gómez, P. Lopez, and J. Duato, "Deterministic versus adaptive routing in fat-trees." Los Alamitos, CA, USA: IEEE Computer Society, 2007.

[5] S. Scott, D. Abts, J. Kim, and W. J. Dally, "The BlackWidow high-radix Clos network," *SIGARCH Comput. Archit. News*, vol. 34, no. 2, pp. 16–28, 2006.

[6] J. Kim, W. J. Dally, B. Towles, and A. K. Gupta, "Microarchitecture of a high-radix router," *SIGARCH Comput. Archit. News*, vol. 33, no. 2, pp. 420–431, 2005.

[7] G. Mora, J. Flich, J. Duato, P. López, E. Baydal, and O. Lysne, "Towards an efficient switch architecture for high-radix switches," in *ANCS '06: Proc. of the 2006 ACM/IEEE symposium on Architecture for networking and communications systems*. New York, NY, USA: ACM, 2006, pp. 11–20.

[8] "Sun datacenter Infiniband switch 36, Sun datacenter Infiniband switch 72, Sun datacenter Infiniband switch 648: Architecture and deployment," April 2010.

[9] J. Kim, W. J. Dally, S. Scott, and D. Abts, "Technology-driven, highly-scalable Dragonfly topology," in *ISCA '08: Proc. of the 35th Annual International Symposium on Computer Architecture*. Washington, DC, USA: IEEE Computer Society, 2008, pp. 77–88.