Load Unbalance in k-ary n-cube Networks

J. Miguel-Alonso[1*], J.A. Gregorio[2*], V. Puente[2*], F. Vallejo[2*] and R. Beivide[2*]

[1] The University of the Basque Country, Department of Computer Architecture and Technology, P.O. Box 649, 20080 San Sebastián, Spain
miguel@si.ehu.es
[2] University of Cantabria, Computer Architecture Group, ETSIIT, Av. de Los Castros s/n, 39005 Santander, Spain,
{jagm,vpuente,fernando,mon}@atc.unican.es

Abstract. This paper studies the effect that HOL (Head-of-Line) blocking in the packet injection queue has on the performance of bidirectional k-ary n-cubes, for values of k over a certain threshold (around 20). The HOL blocking causes an unbalanced use of the channels corresponding to the two directions of bidirectional links, which is responsible for a drop in the network throughput and a rise in the network delay. Simulation results show that this anomaly only appears in those rings where most injections are performed (normally, those in the X axis), and that the elimination of the HOL blocking in the injection queue enables the network to sustain peak throughput after saturation.

1 Introduction

The performance of the interconnection network of a parallel computer has a great impact in the system's performance as a whole. *K*-ary *n*-cubes, are the most common direct interconnection network topologies, encompassing rings, meshes, and tori. A central element of this kind of networks is the packet router that injects packets from (and delivers packets to) the compute node to which it is connected, and also routes packets coming from other routers which have to be delivered to other nodes.

The architecture of the router has a fundamental impact on the execution time of applications running in the parallel machine [4],[7]. Typical design objectives are to keep it simple (to reduce cycle time) while getting as much functionality as possible. Simplicity leads to the use of input transit queues and injection queues with FIFO policy. It is well known, however, that this policy has negative effects, HOL (Head-of-Line) blocking among the most harmful of them. Several works have dealt with ways of reducing this effect on transit queues [5]. However, HOL blocking in the injection, to the best of our knowledge, has not been reported in the literature, and as this paper will prove it also has a negative impact on performance that severely affects the scalability of this kind of networks. In fact, HOL at the network interface is one of the key reasons why performance suddenly drops when the network surpasses

* This paper has been done with the support of the Ministerio de Ciencia y Tecnología, Spain, under grants TIC2001-0591-C02-01 and TIC2001-0591-C02-02, and also by the Diputación Foral de Gipuzkoa under grant OF-758/2003.

its saturation point. As we increment the applied load, before reaching the saturation point, the network accepts the entire offered load. However, sometimes when this maximum point is reached, the accepted load falls *below* the levels reached before saturation—instead of staying at that maximum level.

This anomaly shows up in bidirectional links as an uneven usage (unbalance) between channels going in opposite directions, but only on those rings of the k-ary n-cubes were most injections are performed—typically, those in the X axis—and only for ring sizes over a certain threshold (around 20 nodes per ring). This threshold depends on many characteristics of the network (such as deadlock-avoidance policy, number of virtual channels, etc.) so it must be taken as a reference point, not as an absolute value. For different values of k we have observed variations in the way this unbalance materializes. In some cases, channels X+ are full during most of the simulation time while X- are almost empty (it may well happen the other way around). In some others there are oscillations. In none of the cases studied we have observed this phenomenon disappearing spontaneously. To avoid this anomaly, it is necessary to eliminate the HOL blocking from the injection queue. In the simulation (with uniform traffic) we achieve this by dropping the packets that are blocking the queue. A realistic implementation would require the use of non-FIFO queues, or the implementation of separate injectors: one per direction.

The rest of this paper is organized as follows. In Section 2 we describe the anomaly and the context where it arises. Section 3 presents its effect in rings (1-cubes). Section 4 is devoted to describing the causes of the unbalance and possible measures to avoid it. Section 5 studies the load unbalance in 2D and 3D tori. Section 6 analyzes the appearance of unbalance when running actual applications. Finally, in Section 7, the main conclusions of the work are summarized.

2 General description of the anomaly and its context

We have observed that, in k-ary n-cube networks with bidirectional links, when the applied load exceeds the saturation limit of the network for the corresponding traffic pattern (we have limited our studies to uniform traffic), the occupation of network resources is not balanced between both directions. This anomaly shows up in any of the mentioned networks, and it originates mainly in the structure of the packet router. Figure 1 describes our basic router organization, showing the usual hardware modules: crossbar, buffers, arbitration logic, synchronization, etc.

The design of a router has to maximize the use of the network resources avoiding communication anomalies such as packet deadlock, livelock and starvation. In our experiments, we use a router that has two or three virtual channels (FIFO queues) per input link to support fully Adaptive Bubble Routing (ABR) [7]. When using ABR, a subset of the total virtual channels is configured as a safe (or escape) virtual network [3] in which packet deadlock never occurs. The remaining virtual channels are configured as a fully adaptive virtual network. Packets move under two different policies. In the adaptive virtual network the injection and transit of packets are regulated by both Virtual Cut-Through flow control (VCT) and minimal adaptive routing. In the safe virtual network the injection of packets is regulated by Bubble Flow Control (BFC), a mechanism for avoiding packet deadlock in topologies based either on a

single ring or on a set of rings. In the case of topologies composed of a set of rings, these rings must be visited under Dimension Order Routing (DOR) and packets traveling from one ring to another inside the safe network are considered as new injections. Packets in transit inside a safe ring are regulated by VCT. Packets can move freely from the safe to the adaptive network, but the change of packets from the adaptive to the safe network is regulated by BFC.

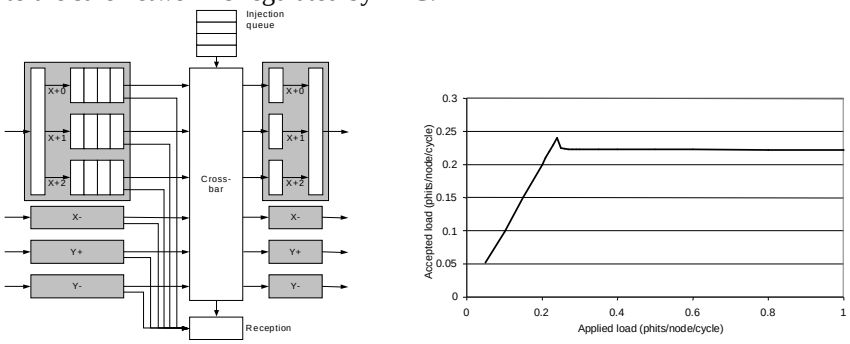

Fig. 1. Left: router model for a k-ary 2-cube. In this case, each physical channel is shared by three virtual channels (0 or Escape, 1 and 2). Each virtual channel has its own input queue to store packets in transit. Right: Accepted vs. applied load in a 30-ary 1-cube. Uniform traffic.

This interconnection network is very similar to the one being used in IBM's BlueGene/L supercomputer [1], in topology (a 3D torus), flow-control, deadlock-avoidance mechanism, etc. We have observed, however, that the mentioned unbalanced use of resources also appears when utilizing different deadlock-avoidance policies, such as the classic use of virtual channels proposed by Dally [2].

3 Load unbalance in rings

Let us consider a ring (k-ary 1-cube) with nodes such as those represented in Fig. 1. Each router is connected to its neighbors via two physical links: X+ (for packets moving from left to right) and X- (for packets moving from right to left). The bandwidth of each physical channel is one phit per cycle. Additionally, the router is connected (via an interface) with a computing element. For the rest of this section, we will consider these additional characteristics of the network: a) Each physical channel is shared between two virtual channels. We will name the VCs as follows: X+0, X+1, X-0 and X-1. Channels 0 are Escape channels. Channels 1 are adaptive; although there is no possible adaptation in a ring (there is only one way to reach the destination), the bubble restriction to inject does not apply in these channels; b) Packets are of 16 phits; c) Each VC has a transit queue of 8 packets (128 phits). The injection queue capacity is also of 8 packets; d) Traffic pattern is uniform.

We have studied the performance of this network by injecting a variable load (measured in phits/node/cycle) of traffic and determining the actual load delivered by the network. Data for a 30-node ring is plotted in Fig.1 (right). All applied load is delivered until the saturation point is reached. After that point, a sharp drop in the accepted load is observed (the magnitude of this drop also depends on many design

parameters). One cause of this drop is that, when reaching the saturation point, network resources (queues) are not used in a balanced way. Channels moving packets to the right (X+) may be saturated while queues in the X- direction are almost empty. Of course, it may happen the other way around (X+ empty while X- full).

Fig. 2 plots the occupation of queues for different ring sizes (10, 20, 30 and 100), all with the characteristics stated before. We have performed simulations under a heavy applied load (always beyond saturation point) of uniform traffic. The occupation of the queues of one node in the ring has been sampled every 2Kcycles. In the case of a 10-node ring, occupation is always very low throughout the simulated time. The average distance traversed by packets when going from source to destination is very short and therefore saturation is reached before queues are fully used.

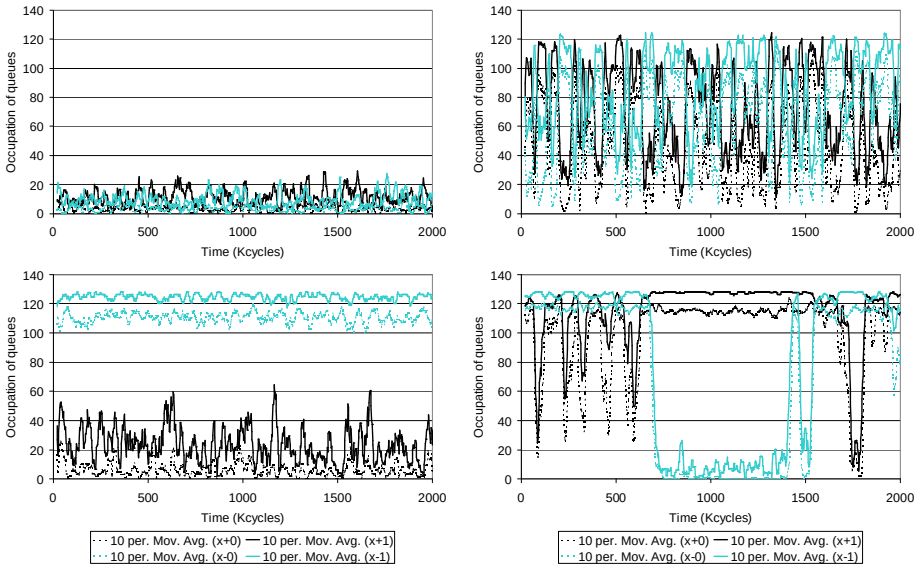

Fig. 2. Queue occupation in rings of 10 (top-left), 20 (top-right), 30 (bottom-left) and 100 (bottom-right) nodes; uniform load at maximum level. Measurements were taken every 2000 simulated cycles; graphs represent the moving average of them. Possible values are between 0 (totally empty queue) and 128 (queue full with 8 packets, 128 phits).

For larger rings, the behavior of the network drastically changes. For a 20-node ring, we observe a fast alternation of the unbalance: for a short time X+ is full while X- is empty but then the situation reverses and X+ is empty while X- is full, for a few cycles before returning to the previous situation. For a 30-node ring this alternation does not happen: almost immediately the unbalance appears and stays for all the simulation time: channels X- are saturated, while channels X+ have an occupation of about 30 phits, 1/4 of their capacity. The graph for a 100-node ring shows yet another scenario, which is actually a combination of the previous two: for 500 Kcycles channels X+ utilization oscillate, but below saturation, while X- are saturated; then, for about 1000 Kcycles, X+ are saturated while X- are less occupied. The last (rightmost) part of the graph shows how the unbalance then reverses. At any rate, in none of the

experiments performed with rings over 20 nodes have we seen the anomaly disappearing spontaneously.

4 Causes of the unbalance and measures to avoid it

We hypothesize that the cause of this unbalance is the Head-of-Line blocking of packets in the injection queue of the router. Packets injected in the network from a computing element (the only source/sink of traffic) have to go through the injection queue associated to the network interface, as shown in Fig. 1. Let us imagine that the packet in the head of the injection queue has to be routed through X+ (because the routing algorithm orders this) and this channel is saturated (actually, both X+0 and X+1 are saturated). The packet must wait. However, if the next packet in the queue has to be routed through X-, it must (unnecessarily) wait. Under uniform traffic, the probability of this situation arising is the same for X+ blocking X- as for X- blocking X+, so the traffic pattern is not the cause of the anomaly.

However, as soon as a certain unbalance happens (which is something we can expect with uniform traffic), a positive feedback effect exists that provokes instability. The packet at the head of the injection queue is willing to enter into the saturated channel and is waiting (forcing the next packets in queue to wait too). Meanwhile, the other (less busy) channel does not receive additional traffic, so part of its load will decrease as packets are consumed. As soon as the saturated channel has room for another packet, the one waiting uses it—so the channel is saturated again. Thus, one channel stays fully occupied, while the other one is almost empty. Packets traveling through the almost empty channel will, when finally injected, rapidly reach their destination, while the others will slowly traverse a sequence of full queues. The effect of the unbalance caused by the HOL blocking will be increasingly worse.

If HOL blocking is the cause of the anomaly, the solution to eliminate it should be avoiding this blocking. For example, if a non-FIFO queue policy is used, some packets may get ahead of others that are blocked [5] and there is no cause for the unbalance to appear. Obviously, this solution increases the complexity of the packet router. Another possible solution is to have more than one injection queue in the network interface: one per direction. A preliminary routing decision is performed at this interface, by putting the packet in the corresponding queue. All the packets stored in one of these queues will follow the same direction, so there is no HOL blocking. This solution, in addition to increasing the complexity of the interface, requires sufficiently long injection queues; otherwise, the HOL blocking would reappear in higher levels when queues are full.

A trivial solution would be to simply drop the packet causing the HOL blocking (the one at the head of the injection queue that cannot be injected). From a practical point of view this is not a valid solution, because it will force higher levels in the communication protocols (or even applications) taking care of recovering lost packets, and this will drastically reduce performance. However, it is perfectly applicable when dealing with a simulation and with synthetic workloads.

5 Studies with 2D and 3D tori

A 2D torus (k-ary 2-cube) consists of a set of rings in the X dimension (X-rings) and another set in the Y dimension (Y-rings). In a simulation study of this network, we have observed that queue usage in the X-rings is almost identical to that of rings (section 3). However, unbalance does not appear in Y-rings. This is due to the policy followed to make packets advance: for adaptive channels, it is not compulsory to inject in an X-ring, but there is a certain preference to do so (X-rings are tested for availability before Y-rings). In the Escape channels packets advance following dimensional order routing (DOR), so under saturation (when Escape channels are more heavily used) most packets will necessarily be injected into an X-ring. Furthermore, Escape channels in Y-rings have more injectors, because they accept traffic from X-rings; thus, the probability of the HOL blocking happening is lower than in X-rings. Fig. 3 plots the queue occupation in an X-ring and in a Y-ring of a 30x30 torus. As we stated before, the graph for the X-ring looks like the one for the 30-node ring. However, curves for the Y-ring show a different, but very reasonable behavior: adaptive channels Y+1 and Y-1 are heavily (and equally) used. Escape channels Y+0 and Y-0 are less heavily (but equally) used.

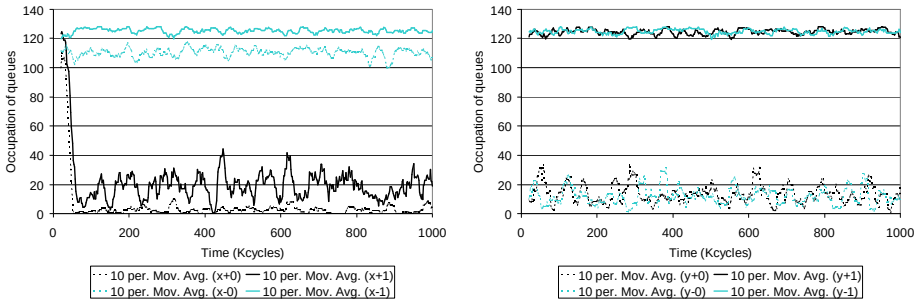

Fig. 3. Queue occupation in a 30x30 torus for the X-rings (left) and Y-rings (right) under maximum uniform load. Measurements were taken every 2000 simulation cycles; graphs represent the moving average of them. Possible values are between 0 (totally empty queue) and 128 (queue full with 8 messages, 128 phits).

What happens now with network performance? Fig. 4 (left) shows accepted vs. offered load for four variants of 30x30 torus with and without using any mechanism to avoid HOL blocking. We observe that using two adaptive VCs (2CVA) improves performance (compared with just one (1CVA)). However, the unbalance in the X-rings does cause drops in performance in both cases—although in the case "2CVA" this happens when applying a slightly higher load. We also observe that the elimination of HOL blocking allows a high level of accepted load to be reached (close to the theoretical maximum) and, what most importantly, maintained under applied loads above the saturation point. Even more noteworthy is the impact that this unbalance has on the average delay experienced by packets when traversing the network, as well as on the standard deviation of this delay. Fig. 4 (right) shows that, avoiding HOL blocking, std. dev. of packet delay drops to about 1/3 of its original value. It is easy to understand that the existence of one saturated path while the other is almost empty causes a great dispersion of the number of cycles that packets require to reach their

respective destinations: while some packets experience an empty network, others have to compete to pass through a collection of saturated channels.

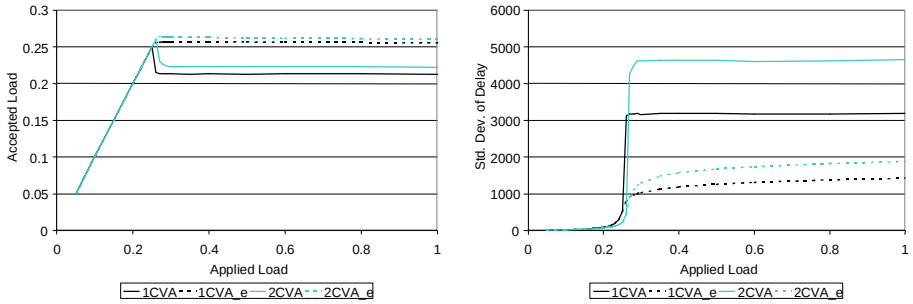

Fig. 4. Left: Accepted vs. applied (uniform) load in a 30x30 torus for 1 and 2 VCs (in addition to the Escape VC) and with/without avoiding HOL blocking. Right: standard deviation of the average packet delay in this network.

For 3D tori the unbalance also appears in the X-rings when their sizes surpass the threshold already stated (around 20). An obvious consequence is that this phenomenon can only be observed in very large networks (around 8000 nodes). As an example, the previous 30x30 torus (900 nodes) suffers from this effect, while a similar 3D torus (10x10x10, or 1000 nodes) does not. In addition to the better topological characteristics (in terms of network bisection bandwidth, average distance, and so on) this may be another reason to make 3D torus the most suitable topology.

6 Unbalance using actual applications

This section proves that the unbalance is not a consequence of using a synthetic uniform load. This unbalance is also present under real applications traffic load. Using a simulator of multiprocessor systems [6] integrated with the SICOSYS simulator of interconnection networks [8], we have performed an execution-driven simulation of the Radix application (part of the SPLASH-2 benchmark suite). Fig. 5 shows the results of measuring each 5000 cycles queue occupation of each virtual channel's transit queues. Packet length is 20 phits; queue capacity is 4 packets.

Fig. 5 shows the results for an application characterized by its uniform traffic pattern such us radix. Between cycles 1M and 3M channel X-1 (adaptive) is more heavily used than X+1. This effect is particularly visible around cycle 2.5M. Additionally, around cycle 6M we can observe the alternation phenomenon previously described for rings. The unbalance cannot be attributed to the characteristics of the Radix application (it is not true that nodes in Radix send more data towards one direction that towards the other), because for the time periods represented in the figure the application interchanges keys in a highly random, uniform way. Thus, this behavior confirms our hypothesis about the occurrence of the anomaly not only with synthetic traffic but also with actual applications.

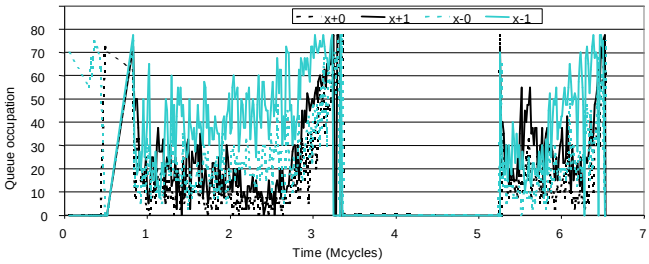

Fig. 5. Occupation of queues when running the Radix application on a 30-node ring with 512K integer keys. Data taken from an execution-driven simulation combining RSIM and SICOSYS.

7 Conclusions

We have identified and analyzed the load unbalance that appears in bidirectional rings of k-ary n-cubes when uniform traffic is applied, which is caused by the HOL blocking in the injection queues. We show that the elimination of this blocking allows throughput to be sustained at its peak level and, additionally, may reduce the standard deviation of network latency by up to 30%. This unbalance appears under different deadlock-avoidance techniques, in 1, 2 and 3-D networks, but only for sizes over 20 nodes per dimension, and only in the rings where most packet injections are performed (usually, the rings in the X axis). Finally, we have shown that the anomaly can also be present with traffic generated by real applications.

References

1. NR Adiga, GS Almasi, Y Aridor, M Bae, Rajkishore Barik, et al., “An Overview of the BlueGene/L Supercomputer”, Proc. of SuperComputing 2002, Baltimore, Nov. 16-22, 2002
2. W. J. Dally and C. L. Seitz, “Deadlock-free message routing in multiprocessor interconnection networks”, IEEE Transactions on Computers, vol. 36, no.5, pp. 547-553, May 1987
3. J. Duato. “A Necessary and Sufficient Condition for Deadlock-Free Routing in Cut-Through and Store-and-Forward Networks”. IEEE Trans. on Parallel and Distributed Systems, vol. 7, no. 8, pp. 841-854, 1996
4. J. Duato, S. Yalamanchili, and L. Ni, Interconnection networks. “An engineering approach”, IEEE Computer Society, 2003
5. G. L. Frazier and Y. Tamir, “Dynamically-Allocated Multi-Queue Buffers for VLSI Communication Switches”. IEEE Trans. on Computers, vol. 41, no.6, pp. 725-737, June 1992
6. V.S.Pai, P. Ranganathan, and S.V.Adve, "RSIM: An Execution-Driven Simulator for ILP-Based Shared-Memory Multiprocessors and Uniprocessors". IEEE TCCA New., Oct. 1997
7. V. Puente, C. Izu, R. Beivide, J.A. Gregorio, F. Vallejo and J.M. Prellezo, “The Adaptive Bubble Router”, J. of Parallel and Distributed Computing. Vol 61, no. 9, September 2001
8. V. Puente, J.A. Gregorio, R. Beivide, “SICOSYS: An Integrated Framework for studying Interconnection Network in Multiprocessor Systems”, Euromicro Workshop on Parallel and Distributed Processing, 2002