Advanced Computer Organization (Paper review) Low-Overhead Network-on-Chip Support for Location-Oblivious Task Placement

Aleksandr Salo

November 26, 2015

Overview

Paper discusses the approach for ensuring global euqality of service (EoS) for systems that have many processing cores with a network-on-chip (NoC) that provides access to shared resources such as main memory and on-chip caches. Since locally-fair arbitration in multi-stage NoC can lead to globally unfair access to shared resources and impact system-level performance, authors design probabilistic arbitration combined with distance-based weights to achieve EoS and overcome the limitation of round-robin arbiter.

The motivation for assuring the EoS is the following: as the number of cores integrated in a chip increases, efficiently scheduling the task on to the many cores becomes increasingly difficult since the scheduling complexity increases super-linearly with the number of cores [2]. One of several aspects of many-core processors that makes optimal scheduling difficult is network-on-chip (NoC) topology. In a multi-hop NoC, the access latency and bandwidth to shared resource such as memory controllers can vary depending on the location of the core. [1] Thus system-level performance can be impacted by the placement the tasks in a multi- and many-core system.

Major points of the article:

1. Round Robin arbitration while providing locally fair arbitration, does not provide globally fair arbitration.
2. Age based arbitration provides global fairness, but too complex to implement for NoC.
3. Probabilistic distance based arbitration (PDBA) with weights, assigned based on the number of hops, approximate age arbitration well enough while providing feasibility of the implementation.
4. Weights should be non-linear: i.e. $w = C^h$ where C is the contention degree or the number of packets contending for the same output port, because linear weight result in the scheme which is similar to Round Robin.
5. Finally, if the traffic is low, then it doesn't matter at all which scheme to use; since Round Robin is much simpler author propose hybrid arbiter that uses Round Robin when the traffic is low and PDBA when traffic is high.

1 Round-Robin is not Globally Fair

With a round-robin (RR) arbitration policy, the processor closest to the destination gets the most bandwidth - 1/2 of the available bandwidth. The processor two hops away gets half of the bandwidth into the intermediate router, for a total of 1/2 * 1/2 = 1/4 of the available bandwidth. As a result, a core that is 7 hops away from the target core receives only 1/64 of the available bandwidth into the target, a factor of 32 times less than that of the core that is next to the target.

Thus although round-robin arbitration provides local fairness at each router, it does not provide any global fairness across all routers. This is an important problem, because reducing the variation in bandwidth is critical for application performance as applications are scaled to higher processor counts. [1]

2 Approximating Age with the Distance (hop count)

To avoid the complexity with age-based arbitration authors are using information already present in the packet, such as source node, current node, or destination node to approximate the packet's age. That approach works because the package latency (T) is directly proportional to the total hop count (H) from source to destination:

$$\begin{aligned} T &= T_h + T_s + T_w + T_c \\ &= Ht_r + T_s + Ht_w + Ht_q \\ &= H(t_r + t_w + t_q) + T_s \end{aligned}$$

Where T_s is the serialization time which is constant, t_r is the per-hop router latency, t_w per-hop wire delay and t_q is per-hop quening delay.

3 Probabilistic distance based arbitration

For sorted priority-based arbitration [2] such as age-based arbitration, arbitration is done deterministically (given a set of input requests and the switch's current state, the output grants are always deterministically assigned) based on the relative age of the requests. Starvation is inherently prevented by age-based arbitration.

However, since authors want to use *priority based on hop count*, livelock and fairness issues are problematic because packets with a lower priority (i.e. a lower hop count) can continually lose arbitration due to a constant stream of newly injected traffic with higher priority. To overcome this problem while still using hop count as the weight, authors propose probabilistic arbiter which probabilistically determines its output based on the weights of input requests. [1] The probability are simply obtained by calculating the fraction the request's weight in the total weight of all the requests:

$$P(g_i) = \frac{w_i}{\sum_{j=1}^{m} w_j}$$

where g is a grant for request i with weight w_i.

By doing so a request in probabilistic arbitration will not starve indefinitely, since the probability that a request will not be granted converges to 0 as the number of arbitrations tried increases.

4 Using Non-Linear Weights

Authors show how linear weights (linearly proportional to the hop count of a packet in probabilistic arbiter—i.e. $w = h_x + h_y$ where h_x and h_y represent the hop count from source to destination in each dimension) cannot provide EoS since farther nodes will be serviced linearly instead of geometrically.

For example, for two packets that are separated by x hop count, the linear weights for the two packets will be w and $w - x$, assuming both packets have the same destination. The probability of each packet winning an arbitration is $\frac{w}{w2-x}$ and $\frac{w-x}{w2-x}$, respectively. For large values of $w(w >> x)$ or for small values of x, the probability of each packet winning the is approximately $1/2$. Thus, the result of probabilistic arbitration with linear weight is very similar to round-robin arbitration.

To account for the exponential proportion of service (as described in section 1), authors propose to use non-linear weights as follows: while traversing in the x-dimension, $w = 2^{h_x}$ is used and when traversing in the y-dimension, $w = 2^{h_x} C^{h_y}$ is used, where $C = 3$ if the destination located on the edge of the 2D mesh network and $C = 4$, for all other destination.

In other words, when traveling in the y-dimension, x-dimension is included as well to properly prioritize packets that have traversed longer overall distance. However, the y-dimension weight is not included while traversing the x-dimension since when a packet only needs to traverse the x-dimension, a packet that needs to traverse both the x and the y-dimension will be unfairly biased. With this metric, authors show, the weight of each packet remains constant or fixed throughout the network.

Additionally, authors investigate the effect of using static vs. dynamic values for hop count h and contention degree C.

5 Hybrid Arbiter

Authors make an observation that a complex arbiter (such as PDBA) only impacts the performance at a high load or when contention occurs in the network. Thus, at a low load or a near zero load, simple arbitration can be used with minimal impact on performance or fairness since there is little or no contention in the network. In that case authors use round-robin as a simple arbiter.

Furthermore, some of the steps required in the complex arbitration can be done in the previous cycles while waiting in the queue and hiding some of the latency in the complex arbitration; additionally weights could be pre-calculated and stored in separate buffer. That reduces the latency since weights computations requires random number generating and scaling appropriately.

Conclusion

I really liked the paper because it conveyed its ideas (using probabilistic distance-based arbiter when the traffic load is high) clearly and in precise manner. Motivation (importance of scheduling with many multi-cores and NoC) was provided along with the compelling test suite which demonstrates the benefits of using PDBA for various traffic patterns (uniform, tornado etc).

References

[1] Kim, Gwangsun, et al. "Low-overhead network-on-chip support for location-oblivious task placement." Computers, IEEE Transactions on 63.6 (2014): 1487-1500.

[2] A. Fedorova, "Say yes to dumb operating systems and smart applications," 2008, WACI-VI.

[3] D. Stiliadis and A. Varma, "Design and analysis of frame-based fair queueing: A new traffic scheduling algorithm for packet-switched networks," in Proc.ACMSIGMETRICSConf. (SIGMETRICS'96), 1996, pp. 104–115.