Bandwidth Bottleneck in Network-on-Chip for High-Throughput Processors

Jiho Kim, Sanghun Cho, Minsoo Rhu
KAIST
{jihokim,shcho1118,mrhu}@kaist.ac.kr

Ali Bakhoda
Microsoft
Ali.Bakhoda@Microsoft.com

Tor M. Aamodt
University of British Columbia
aamodt@ece.ubc.ca

John Kim
KAIST
jjk12@kaist.edu

ABSTRACT

A critical component of high-throughput processors such as GPGPUs is the network-on-chip (NoC) that interconnects the cores and the memory partitions together. Different NoC architectures for throughput processors have been proposed but they have often been based on similar principles as multicore (or CPU) NoC, including emphasis on bisection bandwidth and the traffic pattern. In this work, we identify how such prior approaches are not necessarily applicable to NoC in throughput processors. We identify how different bandwidth bottlenecks can be created in high-throughput processors and argue NoC design for throughput processors need to be re-evaluated.

CCS CONCEPTS

• **Computer systems organization → Interconnection architectures**.

KEYWORDS

GPGPU, network-on-chip, bisection bandwidth

ACM Reference Format:
Jiho Kim, Sanghun Cho, Minsoo Rhu, Ali Bakhoda, Tor M. Aamodt, and John Kim. 2020. Bandwidth Bottleneck in Network-on-Chip for High-Throughput Processors. In *Proceedings of the 2020 International Conference on Parallel Architectures and Compilation Techniques (PACT '20), October 3–7, 2020, Virtual Event, GA, USA.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3410463.3414673

1 INTRODUCTION

In this work, we explore network-on-chip (NoC) design in high-throughput processors or accelerators. In particular, we re-visit the design of NoC for high-throughput processors, such as GPGPUs. Prior work [2] identified how the communication pattern in GPGPUs can cause a bottleneck and limit overall performance. The *many-to-few-to-many* traffic pattern with *many* cores communicating with *few* memory controllers (or memory partitions) that send data back to the *many* cores [2] – with the few memory controllers becoming the bottleneck for overall performance. Since memory access for high-throughput workloads are often read requests where the read request size is relatively small but the reply (or the data) is larger, the *reply bandwidth* ($BW_{NoC-MEM}$ in Figure 1(b))) was identified as a key bottleneck in NoC for throughput processors.

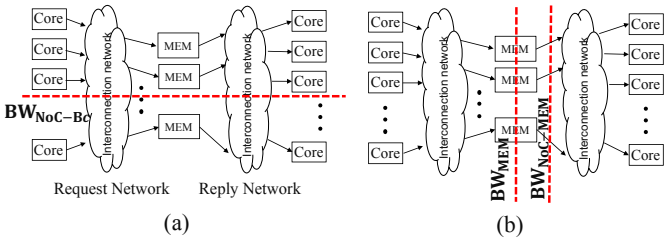


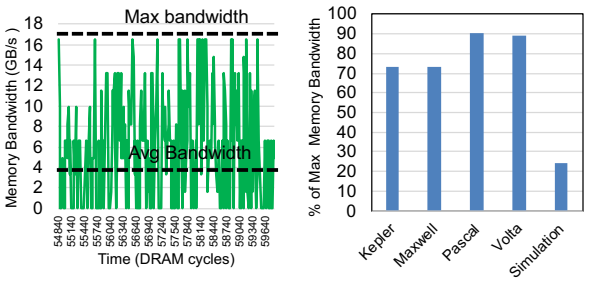
Figure 1: Communication pattern of many-to-few-to-many in throughput processors, showing (a) the bisection bandwidth (BW_{NoC-B_c}) and the (b) memory bandwidth (BW_{MEM}) and the interface bandwidth ($BW_{NoC-MEM}$). *MEM* are memory partitions, includes L2 and the memory controller.

However, baseline NoC of throughput processors attempted to provide a balance between memory bandwidth and NoC *bisection* bandwidth (BW_{NoC-B_c} in Figure 1(a))) to achieve cost-effective NoC. While bisection bandwidth is a critical component in any interconnection network design [3], the bisection bandwidth is only an important metric if the nodes (i.e., *Core*'s for the request network and *MEM*'s for the reply network) are injecting sufficient bandwidth to *saturate* the bisection bandwidth. With communication occurring from the cores to the memory nodes (and vice-versa), the *terminal* or *interface* bandwidth from the nodes (in particular, for the reply network $BW_{NoC-MEM}$ in Figure 1(b)) becomes the bottleneck. Thus, *insufficient interface bandwidth can fundamentally limit the bandwidth of memory (and L2) bandwidth and thus, the bisection bandwidth is not necessarily the main bottleneck.*

2 BANDWIDTH MEASUREMENTS

In this section, we provide some bandwidth/performance measurements on real GPUs and a simulator to understand the bandwidth bottleneck. Since congestion builds up at the NoC-MEM interface of the reply network, the reply bandwidth bottleneck impacts not only the reply network but backpressure propagates to impact memory bandwidth utilization and the request network [2]. Using the same 2D mesh "throughput-effective" baseline simulation configuration as [2] with GPGPU-sim [1], we plot the memory

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
PACT '20, October 3–7, 2020, Virtual Event, GA, USA
© 2020 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-8075-1/20/10.
https://doi.org/10.1145/3410463.3414673



(a) (b)

Figure 2: (a) Memory channel utilization fluctuation from backpressure from the reply bandwidth bottleneck using simulation and (b) memory utilization comparison across different real GPUs.

utilization of a single memory channel over time when executing a memory intensive synthetic workload (i.e., STREAM workload) in Figure 2(a). The memory bandwidth reaches maximum bandwidth but the bandwidth cannot be sustained as the average memory bandwidth utilization is significantly lower – i.e., less than 25% memory channel utilization is achieved on average. As the reply bandwidth interface congestion forms and queues between the memory system and the NoC fills up, the memory can no longer be serviced or is "blocked" and results in low utilization.

Memory Bandwidth in GPUs: We evaluate the same STREAM benchmark on real systems and the results are shown in Figure 2(b). Average memory utilization is measured across different generations of NVIDIA GPU architectures and all system measurements exceed 70% utilization and some reach as high as 90%. In comparison, the simulator evaluation is significantly limited to approximately only 20% – thus, the NoC bandwidth provisioned insufficiently in the simulations. This is not a simulator limitation but a bandwidth/configuration issue in assumptions made when utilizing the simulator.

L2 Bandwidth: In addition to memory bandwidth, another bandwidth that NoC impacts is the L2 bandwidth since L2 is placed near the memory controllers and interconnected through the NoC from the cores. We measured the L2 bandwidth on real systems by executing a synthetic workload with memory accesses that results in approximately 100% L2 hit rate and access all L2 partitions. L1 is bypassed to ensure all accesses traverse the NoC. Figure 3 plots L2 bandwidth as the number of CTAs allocated to each core (or SM) increases for two different GPU architectures, Maxwell and Volta. The L2 bandwidth continues to increase as the number of CTAs increases and having more than 2 (or 4) CTAs results in L2 bandwidth *exceeding* the maximum system memory bandwidth for both GPUs. While memory bandwidth is critical, the NoC bandwidth needs to be provisioned sufficiently to ensure that L2 bandwidth is not limited by the NoC bandwidth in simulations.

Traffic pattern: Traffic pattern has significant impact on the performance of a topology or routing algorithm [3]. The dominant traffic pattern in throughput processors is the *many-to-few-to-many* but while such traffic or communication pattern does occur, bandwidth dominates the traffic pattern in throughput processors – i.e., the bandwidth (i.e., memory bandwidth or L2 bandwidth) determines the traffic, not necessarily the number of nodes. This is critical to ensure that the system is properly balanced.

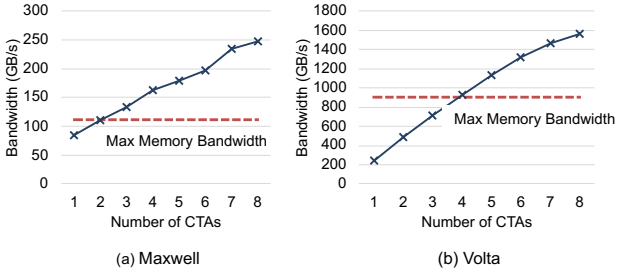


Figure 3: L2 bandwidth measured on (a) Maxwell and (b) Volta GPU systems as the number of CTAs allocated to each core increases.

Based on these preliminary measurements, the following observations can be made.

(1) NoC does not bottleneck the memory bandwidth in real system since high memory utilization can be observed.
(2) L2 bandwidth is significantly higher than memory bandwidth and the NoC must support L2 bandwidth.
(3) Simulations-based studies for high-throughput processors need to provide sufficient bandwidth to ensure that the NoC does not limit either the L2 or memory bandwidth.

Thus, insufficient interface bandwidth between memory partition and the NoC can fundamentally limit the bandwidth of memory (and L2) bandwidth. Another approach to view the problem is using a simple bottleneck analysis. Since the cores, the NoC, and the memory system are connected in series, *"the maximum throughput of K sub-systems in series is the* **minimum** *of the subsystem throughput"* [4]. Thus, even if high memory bandwidth (BW_{MEM}) or bisection bandwidth is provided, the system cannot sustain such high bandwidth since the aggregate terminal bandwidth ($BW_{NoC-MEM}$) becomes the bottleneck among the components.

Impact on NoC Design: Based on the observations that high interface bandwidth is needed, cost-effective NoC architecture and topology needs to be re-visited. Some prior work assumed a flat network topology (e.g., mesh, torus, etc.) for the NoC in throughput processors. However, while such topologies can provide good scalability and high bisection bandwidth, it can be challenging to provide high terminal bandwidth that is needed for the throughput processors. In comparison, a hierarchical NoC organization can better match the bandwidth demands of the throughput processors.

3 SUMMARY

Bandwidth is an important performance metric in any interconnection networks. However, in addition to bisection bandwidth, the terminal or the interface bandwidth in throughput processors need to be properly provisioned to ensure that NoC of throughput processors do not become the bottleneck.

REFERENCES

[1] Ali Bakhoda et al. 2009. Analyzing CUDA workloads using a detailed GPU simulator. In *ISPASS*. IEEE, 163–174.
[2] Ali Bakhoda et al. 2010. Throughput-Effective On-Chip Networks for Manycore Accelerators. In *MICRO (MICRO '43)*. Atlanta, GA, 421–432.
[3] William Dally and Brian Towles. 2003. *Principles and Practices of Interconnection Networks.* Morgan Kaufmann Publishers Inc.
[4] Mark D. Hill. 2019. Three Other Models of Computer System Performance. *CoRR* abs/1901.02926 (2019). arXiv:1901.02926 http://arxiv.org/abs/1901.02926