BODCA: Heterogeneous CPU-GPU computing system with Bandwidth-Optimized DRAM cache design

Sungji Choi and Won Woo Ro
School of Electrical and Electronic Engineering, Yonsei University
{choi.sungji, wro}@yonsei.ac.kr

Abstract

Heterogeneous CPU-GPU computing is important due to multi-core CPU and GPU. In heterogeneous CPU-GPU computing, cache coherence and unified memory spaces allow CPU and GPU cores to access the same memories without memory copy overheads. This paper focuses on the potential of the large-scale DRAM cache design in a heterogeneous computing simulation, considering the tradeoff of DRAM cache design. Also, this paper analyzes the improved latency and utilization of the system bandwidth of DRAM cache design.

We propose a Bandwidth Optimized DRAM cache (BODCA), which is similar to Alloy Cache but has a dynamic bandwidth assignment technique. On average, BODCA shows 5.1% IPC improvement and 7.3% increment in the utilization of the memory system bandwidth compared to the Alloy Cache.

Keywords: DRAM cache, Heterogeneous computing, Dynamic bandwidth assignment technique

1. Introduction

Running complex neural networks in data centers leads to demands for high memory bandwidth. 3D-stacked DRAM technology can solve these demands. 3D-DRAM, such as hybrid memory cube(HMC) and high-bandwidth memory(HBM), provides 4x-8x high bandwidth at similar DRAM access latency [1]. As a commodity product, Intel's Knights Landing(KNL) [2] is a representative of using 3D-DRAM, HBM. Intel's KNL takes advantage of HBM in a cache mode or a flat mode. In the cache mode, HBM is a hardware-managed cache (DRAM cache), placed in the package, between last-level cache and the main memory. Otherwise, in the flat mode, HBM is a part of the main memory and visible to OS.

In this paper, we focus on the cache mode of Intel's Knight Landing and analyze the effect of the DRAM cache. However, one challenge exists in using such a large cache. The challenge is handling with meta-tags of DRAM cache. Because DRAM cache stores metadata, the tag of data is much larger than conventional. Many researchers solve this problem by placing a tag store in a DRAM cache [3,4].

However, this approach generates DRAM access twice(One for tag access, one for data access). For example, in a 64-byte cache line, Loh-Hill Cache [3] col-locates 3 ways for tags and 28ways for data in the same 2KB row buffer. This means that Loh-Hill Cache does serial accesses similar to conventional caches. This method degrades the total performance because the DRAM cache is much slower than conventional caches. Therefore, the key point is researchers have to focus on optimizing latency first and then hit rate second.

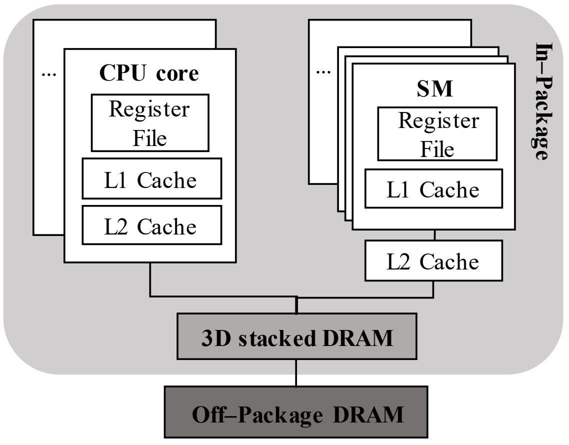

Figure 1: A baseline of heterogeneous CPU-GPU computing system with 3D stacked DRAM

Compared with Loh-Hill Cache, the Alloy Cache is a latency-optimized DRAM cache[4]. Alloy Cache is a direct-mapped cache and consists of TAD(tag and data). In the a 64-byte cache line, Alloy Cache locates 28 data lines of 72-byte TAD(8-byte tag and 64-byte data) in 2KB row buffer. In Alloy Cache, accessing tag and data at once is a big factor in reducing latency. Because of this advantage, we use our DRAM cache architecture, BODCA, as Alloy Cache with a 128-byte cache line. Moreover, we implement a dynamic bandwidth assignment technique [6] in BODCA and analyze our new cache architecture in a heterogeneous CPU-GPU computing simulation.

This paper makes the following contributions :

(1) We re-configure DRAM cache design in a 128-byte cache line. By coordinating the cache line size of CPU and GPU, we can do heterogeneous CPU-GPU computing. Also, we implement DRAM cache design in the heterogeneous CPU-GPU computing system, as shown in Figure 1. We analyze the performance potential of DRAM cache design in the heterogeneous CPU-GPU computing system.

(2) We propose a bandwidth optimized DRAM cache, named BODCA, which dramatically improves the overall performance of heterogeneous CPU-GPU computing system and maximizes the system bandwidth. We implement the dynamic bandwidth assignment technique in BODCA and this technique is related to the DRAM cache access rate.

Authorized licensed use limited to: Raytheon Technologies Corporation. Downloaded on December 18,2020 at 16:04:57 UTC from IEEE Xplore. Restrictions apply.

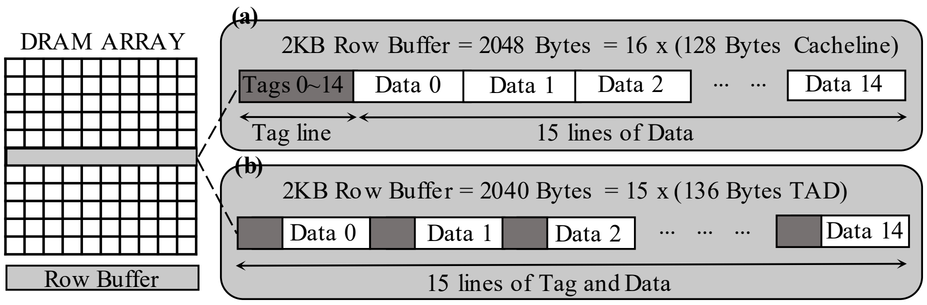

Figure 2: DRAM cache organization with a 128-byte cache line (a) Loh-Hill Cache, (b) Alloy Cache

2. Background and Motivation

In this section, we introduce heterogeneous CPU-GPU computing and reconstructed DRAM cache with a 128-byte cache line. Also, we show our motivation for the dynamic assignment technique of the DRAM cache.

2-1. Heterogeneous CPU-GPU computing

In a discrete CPU-GPU system, CPU executes the application and transfers parallel tasks to the GPU. In this system, the GPU cannot do any parallel tasks without copies of data and instruction from the CPU. However, it becomes common to see CPU and GPU integrated into the same package. Intel and AMD have shown the improved performance of heterogeneous CPU-GPU computing systems compared to the discrete CPU-GPU system. Designing CPU and GPU into a single die has a huge advantage of allowing processors to avoid memory copy overheads [5]. By sharing virtual address space, CPU and GPU share the same memory system.

We investigate the performance potential of DRAM cache in a heterogeneous CPU-GPU computing system. IPC(Instructions Per Cycles) in Figure 3 is normalized to the IPC of heterogeneous CPU-GPU computing system without DRAM cache, Alloy Cache. With a heterogeneous CPU-GPU computing, the average IPC of system with Alloy Cache is 1.08 times higher than the IPC of system without the Alloy Cache.

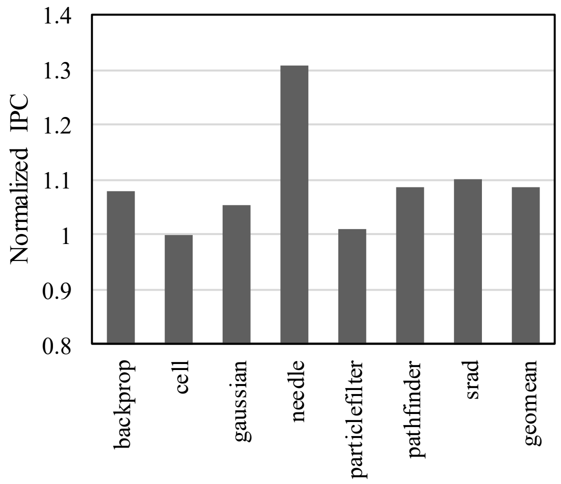

Figure 3: Performance potential of DRAM cache in heterogeneous CPU-GPU computing system

2-2. DRAM cache

In this paper, we set the cache line of DRAM cache as 128 bytes because of cache coherency between CPU and GPU. By setting CPU cache line to GPU cache line, 128 bytes, both CPU and GPU can share virtual address space and same memory system. Also, for explanation of BODCA, we implement Loh-Hill Cache and Alloy Cache in Figure2.

As shown in Figure 2(a), we configure Loh-Hill Cache to locate tags and data in the same 2KB row buffer. We consider 2KB row buffer to satisfy with 8-byte tag-stores of each data line and 128-byte data lines. So, we locate a 128-byte tag store and 15 data lines of 128 bytes in a 2KB row buffer.

Also, as shown in Figure 2(b), we configure Alloy Cache consisted of TAD(8-byte tag and 128-byte data) in a 2KB row buffer. The difference between tag overhead in Loh-Hill Cache and Alloy Cache is only 8bytes, very little, compared to a 2KB row buffer. We use this design as BODCA.

2-3. Dynamic bandwidth assignment technique

We observe that the ratio of requests served in DDR4 is so small that the memory system bandwidth is not fully utilized. As shown in Figure 4, the average ratio of CPU requests served in DDR4 is 12.2% and the average ratio of GPU requests served in DDR4 is 0.47%. This result means that the complexity of DRAM cache is very high, especially in handling GPU requests

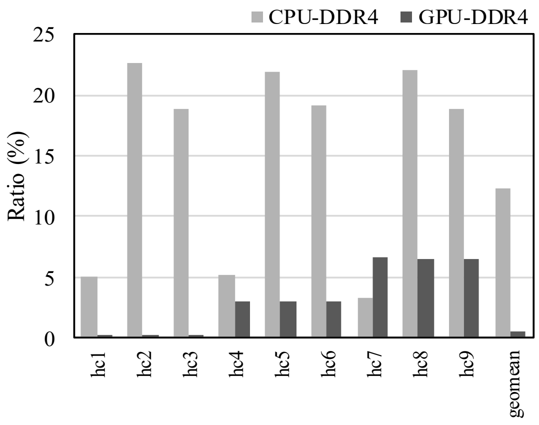

Figure 4: The average ratio of CPU/GPU requests served in DDR4

Authorized licensed use limited to: Raytheon Technologies Corporation. Downloaded on December 18,2020 at 16:04:57 UTC from IEEE Xplore. Restrictions apply.

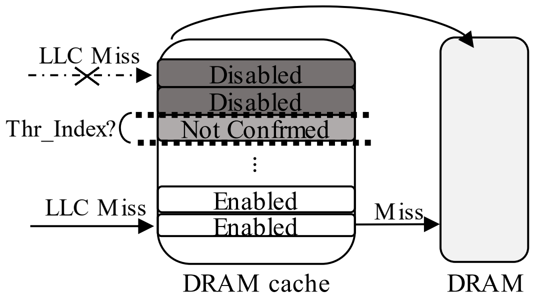

Figure 5: Dynamic Thr_index choosing technique by DRAM cache access rate.

To utilize memory system bandwidth properly, we implement a dynamic bandwidth assignment technique [6] in BODCA. Figure 5 shows a method of distributing memory accesses between in-package DRAM cache and off-package DRAM. This method dynamically determines the threshold index of the disabled sets by the DRAM cache access rate(1).

$$\text{DRAM cache access rate}=\frac{\text{Total DRAM cache accesses}}{\text{Total memory accesses}} \quad (1)$$

If the DRAM cache access rate is lower than 80%, the number of disabled sets is reduced, allowing the system to utilize more bandwidth of DRAM cache than off-package DRAM. In this case, the threshold index is lowered. Conversely, if the DRAM cache access rate is higher than 80%, the threshold index and the number of disabled sets are increased, so that the system can utilize efficiently the bandwidth of off-package DRAM.

3. Experiments And Results

3-1. System Configuration

We use gem5-gpu [7], to simulate the heterogeneous CPU-GPU computing system. As shown in Table 1, we model our system processors as 8 out-of-order CPU cores and 16 Fermi GPU cores. In this system, each CPU core has private L1 and L2 caches. However, GPU cores share L2 cache for cache coherence between GPU cores. Also, we configure the cache line size of CPU and GPU as 128bytes to share virtual address space and same memory sytem. For memory system configuration, we integrate Ramulator [8], an open-source for DRAM systems, to gem5-gpu. With gem5-gpu integrated with Ramulator, we simulate the configured system in a cache mode.

3-2. Workloads

We select memory-intensive workloads from the SPEC CPU2006 suite [9] and Rodinia [10]. Specifically, we use only-CPU workloads from SPEC CPU2006 and CPU-GPU workloads from Rodinia. As we model our heterogeneous computing system with 8 CPU cores, we place 7 CPU cores for only-CPU workloads and 1 CPU core for CPU-GPU workloads. So, we mix memory-intensive workloads from SPEC CPU2006 and Rodinia into Table 2.

Table 1: Baseline System Configuration

Processor & Memory Subsystem	
CPU	
Cores	OoO, 8cores(x86) @3GHz
L1 I, D Cache	32KB, 4-way (private)
L2 Cache	512KB, 16-way (private)
GPU	
Cores	16 Fermi SMs@700MHz
L1 I, D Cache	32KB, 4-way (private)
L2 Cache	4096KB, 16-way (shared)

Memory system	
3D stacked DRAM (In-Package)	
Channel	4channel, 128MB per channel
Rank	1 ranks per channel
Bank	8 banks per rank
Bus Frequency	500MHz
Bus Width	128 bits per channel
tCL-tRCD-tRP	7-7-7 (ns)
tBURST	2 (ns)
DRAM (Off-Package)	
Channel	1 channel
Rank	1 ranks per channel
Bank	8 banks per rank
Bus Frequency	1200MHz(DDR 2.4GHz)
Bus Width	64 bits per channel
tCL-tRCD-tRP	16-16-16 (ns)
tBURST	4 (ns)

Table 2:Mixed SPEC CPU2006 + Rodinia workloads

Name	SPEC CPU2006	Rodinia
hc1	milc x 7	needle
hc2	mcf x4 , lbm x3	needle
hc3	lbm x4, milc x4	needle
hc4	milc x 7	backprop
hc5	mcf x4 , lbm x3	backprop
hc6	lbm x4, milc x4	backprop
hc7	milc x 7	particlefilter
hc8	mcf x4 , lbm x3	particlefilter
hc9	lbm x4, milc x4	particlefilter

3-3. Performance Results

In our experiments, we compare Alloy Cache and our proposed DRAM cache, BODCA. In Figure 6, we analyze BODCA's memory requests in DDR4. The average ratio of CPU requests served in DDR4 is 28.7% and the average ratio of GPU requests served in DDR4 is 26.3%. On average, BODCA shows improved ratios in the DDR4 with 2.34x CPU request ratio and 56.3x GPU request ratio, compared to Alloy Cache.

Also, in Figure 7 and Figure 8, we analyze the benefit of BODCA in terms of performance and system memory bandwidth utilization. Figure 7 and Figure 8 is normalized to IPC and bandwidth utilization of heterogeneous CPU-GPU computing system without DRAM cache. As shown in Figure 7, BODCA has average 1.67x normalized IPC and Alloy Cache has average 1.59x normalized IPC. Moreover, as shown in Figure 8, BODCA uses average 6.28x normalized system memory bandwidth and Alloy

Authorized licensed use limited to: Raytheon Technologies Corporation. Downloaded on December 18,2020 at 16:04:57 UTC from IEEE Xplore. Restrictions apply.

Cache uses average 5.85x normalized system memory bandwidth. As a result, BODCA has benefits of 5.1% higher performance and 7.3% larger system memory bandwidth utilization than Alloy Cache.

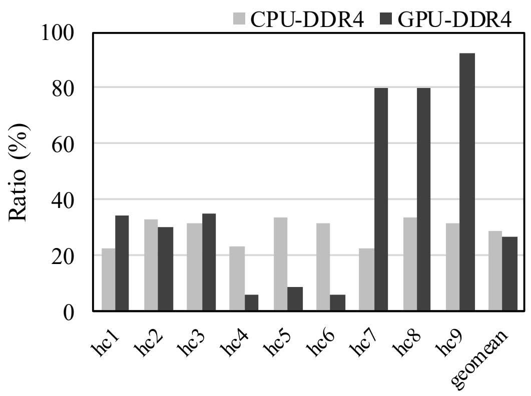

Figure 6: The ratio of CPU, GPU requests in DDR4 to the total CPU, GPU requests

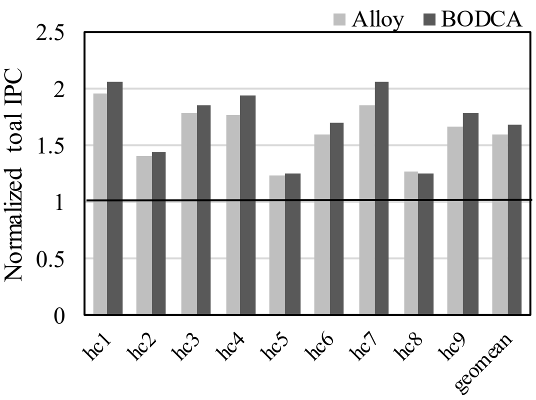

Figure 7: Normalized IPC of heterogeneous CPU-GPU computing with BODCA

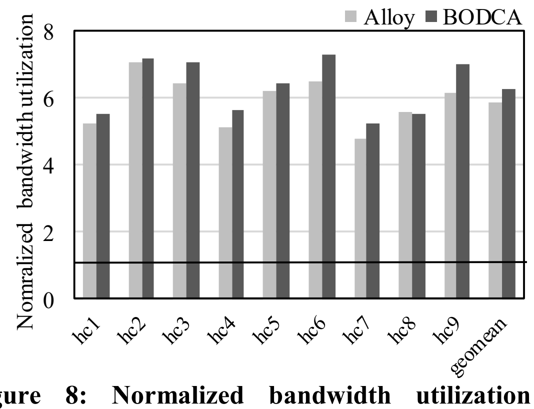

Figure 8: Normalized bandwidth utilization of heterogeneous CPU-GPU computing with BODCA

4. Conclusion

In this paper, we present a bandwidth optimized DRAM cache, named BODCA. BODCA achieves improved performance and maximizes the usage of the system memory bandwidth. In detail, BODCA uses a re-configured Alloy Cache architecture, which solves latency problem in searching tag stores, and a dynamic bandwidth assignment technique. In our experiment results, BODCA shows 5.1% higher IPC and 7.3% larger usage of the system memory bandwidth than Alloy Cache design.

5. Acknowledgment

This research was supported by the MOTIE(Ministry of Trade, Industry & Energy (No. 10080590, Technology Development of Unified Memory System for Heterogeneous System Architecture) and KSRC(Korea Semiconductor Research Consortium) support program for the development of the future semiconductor device.

References

[1] D. W. Chang, G. Byun, H. Kim and etc, “Reevaluating the latency claims of 3D stacked memories”, Design Automation Conference (ASP-DAC), 2013 18th Asia and South Pacific, pp. 657–662, 2013

[2] Avinash Sodani, “2015. Knights Landing (KNL): 2nd Generation Intel Xeon Phi Processor”, Hot-Chips, 2015. http://tinyurl.com/hotchips-2015-sodani

[3] G. H. Loh and M. D. Hill, “Efficiently enabling conventional block sizes for very large die-stacked DRAM caches”, In Proceedings of the 44th International Symposium on Microarchitecture, Dec 2011.

[4] M. Qureshi and G. H. Loh, “Fundamental latency trade-offs in architecting DRAM caches”, In Proceedings of the 45th International Symposium on Microarchitecture, Dec 2012.

[5] Hestness, J., Keckler, S. W., & Wood, D, “A comparative analysis of microarchitecture effects on CPU and GPU memory system behavior”, In Workload Characterization (IISWC), pp. 150-160, October 2014.

[6] Chiachen Chou and etc, “BATMAN: Maximizing Bandwidth Utilization of Hybrid Memory Systems”, Technical Report.School of Electrical and Computer Engineering, Georgia Institute of Technology.

[7] Power, J. et al, “gem5-gpu: A Heterogeneous CPU-GPU Simulator”, Computer Architecture Letters, 13, 1 2014.

[8] Y. Kim, W. Yang, and O. Mutlu, “Ramulator: A Fast and Extensible DRAM Simulator”, IEEE CAL, 2016

[9] John L. Henning, “SPEC CPU2006 Benchmark Descriptions”, SIGARCH Comput. Archit. News 34, 4 (Sept. 2006), 1–17. https://doi.org/10.1145/1186736.1186737

[10] Che, S., Boyer, M., Meng, J., Tarjan, D., Sheaffer, J. W., Lee, S. H., & Skadron, K., “Rodinia: A benchmark

Authorized licensed use limited to: Raytheon Technologies Corporation. Downloaded on December 18,2020 at 16:04:57 UTC from IEEE Xplore. Restrictions apply.

suite for heterogeneous computing", In Workload Characterization(IISWC), pp. 44-54, 2009 October.

Authorized licensed use limited to: Raytheon Technologies Corporation. Downloaded on December 18,2020 at 16:04:57 UTC from IEEE Xplore. Restrictions apply.