Exploring Stacked Main Memory Architecture for 3D GPGPUs

Yuang Zhang[1, 3], Li Li[1*], Axel Jantsch[2], Zhonghai Lu[3], Minglun Gao[1], Yuxiang Fu[1], Hongbing Pan[1]

[1] Institute of VLSI Design, LAPEM, Nanjing University, 210046, Nanjing, China
[2] Institute of Computer Technology, Vienna University of Technology, 1040 Vienna, Austria
[3] Department of Electronic Systems, KTH-Royal Institute of Technology, 16440 Kista, Stockholm, Sweden
* Email: lili@nju.edu.cn

Abstract

The tremendous number of threads on general purpose graphic processing units (GPGPUs) poses significant challenges on memory architecture design. 3D stacked main memory architecture atop GPGPU is a potential approach to provide high data communication bandwidth and low access latency to meet the requirement of GPGPUs. In this paper, we explore the performance of 3D GPGPUs with stacked main memory. The experimental results show that the 3D stacked GPGPU can provide up to 124.1% and on average 55.8% performance improvement compared to a 2D GPGPU scheme.

1. Introduction

Graphics processing units (GPUs) [1] were originally designed for graphics applications. The introduction of hardware and software support has allowed GPUs to become a viable platform for throughput-oriented general purpose computing, which is known as general purpose graphics processing units (GPGPU).

To unleash the computing power of GPGPUs, the massive concurrent threads require a huge memory bandwidth. Our experiments on a variety of GPGPU applications reveal that, by low overhead multi-thread switching, the parallel single instruction multiple thread (SIMT) execution model [2] can only partially hide the long latency of memory access. As the number of processing unit increases in GPGPUs, more and more data is required from the memory subsystem, increasing the pressure on the I/O infrastructure. Memory bandwidth and memory latency are among the major bottlenecks for modern GPGPUs.

Three dimensional integrated circuit (3D IC) [3] is a promising solution to provide low latency, high bandwidth interconnections between the stacked computing and memory resource layers by dense vertical vias. The vertical through-silicon vias (TSVs) [4] are the enabling technology for 3D integration. The width of a TSV channel can be up to thousands of bits. Hence, 3D integration is a way to mitigate the memory challenge in future GPGPUs.

In this work, we investigate the performance potential of 3D GPGPUs with stacked main memory. The key contributions include:

1. We perform experiments to reveal memory access latency impact on the performance of the GPGPU. Our study shows that the long memory latency may not be hidden efficiently by the SIMT execution model. There exists an optimization space for the memory system which depends on the characteristics of applications.
2. We propose a stacked memory architecture for 3D GPGPUs. The experimental results on the memory sensitive applications show that the 3D GPGPU can provide up to 124.1% and on average 55.8% performance improvement compared to a 2D GPGPU design.

The rest of this paper is organized as follows. Section 2 introduces the simulation methodology we use and the GPGPU benchmark feature analysis. Section 3 discusses the 3D GPGPU with stacked memory dies. In Section 4, we present and analyze the simulation results. Related work is discussed in Section 5. Finally, Section 6 concludes this paper.

2. GPGPU application feature analysis

We run the simulation on the GPGPU-Sim full system simulator [5]. CACTI-3DD [6] is used to get the detailed access latency for the 2D and 3D DRAM main memory respectively. We use application instruction per cycle (IPC) to evaluate the performance. IPC is a performance metric for multicore systems running parallel workloads that considers the variations of the execution times of different threads. We take the standard benchmarks, including the benchmark suit in GPGPU-Sim, Rodinia [7], and Parboil [8] for the experiments.

The GPGPU consists of a collection of data parallel cores, labeled SM (streaming multiprocessor). Each SM is equipped with tens of small processing cores. The SMs are connected by an interconnect network to multiple memory modules. The L2 caches are shared by all the SMs. At each L2 bank, there is a memory controller (MC) which connects the off-chip DRAM. An L2 cache and a corresponding MC together are called a *memory segment (MS)* in this paper.

We compare the IPCs of a 2D GPGPU (GPGPU-basic) with that of a GPGPU which has a perfect memory system (GPGPU-perfect_mem). Figure 1 shows the basic 2D GPGPU. There are four MSs at the corner nodes.

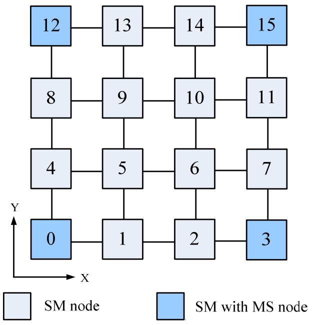

Figure 1. Layout of mesh-based 2D GPGPU

The GPGPU-perfect_mem has a similar topology as the GPGPU-basic. Perfect memory means that the L1 cache misses are filled immediately. The performance of the GPGPU-perfect_mem reveals the optimization potential of the memory system for the related applications.

In Figure 2, we show the normalized comparison of the GPGPU-perfect_mem over the GPGPU-basic.

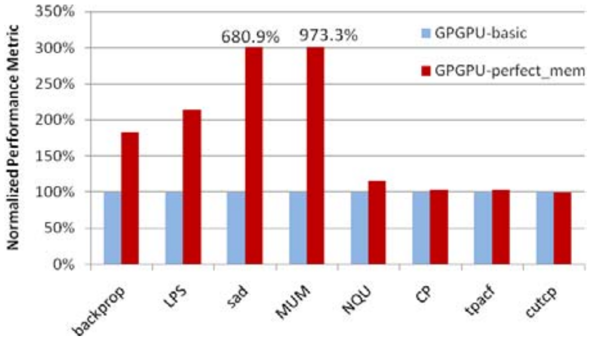

Figure 2. Normalized performance comparasion of GPGPU-perfect_mem over GPGPU-basic

We can find from Figure 2 that the GPGPU applications can be categorized into two groups. The right most four cases (*NQU*, *CP*, *tpacf* and *cutcp*) are memory insensitive. The memory access latency has very little impact on the overall performance. The left most applications (*backprop*, *LPS*, *sad* and *MUM*) show that if the memory access latency of GPGPU system is reduced, the overall performance can be largely enhanced. In these cases, the memory access latency is not hidden efficiently by the SIMT scheme of the GPGPU. In this paper, we focus on applications which are sensitive to the memory access latency (including *backprop*, *gaussian*, *lud*, *nw*, *srad_v2*, *AES*, *BFS*, *LPS*, *MUM*, *RAY*, *LIB*, *histo*, *sad*, and *stencil*). Evidently there is an optimization space for the memory system. Hence, we propose a 3D stacked memory architecture for GPGPUs.

3. 3D GPGPU with stacked main memory

The GPGPU application threads are grouped into cooperative thread arrays (CTAs). Threads from one CTA can make progress while threads from another CTA are waiting for the data fetching results. For a given number of threads per CTA, allowing more CTAs to run on an SM core provides additional memory latency tolerance, though it may imply increasing register and shared memory resource usage. However, if a compute kernel is memory-intensive, completely filling up all CTA slots may reduce performance by increasing contention in the interconnection network and DRAM controllers. Hence, to decrease the memory latency has two fold benefits, first, the demanded on-chip resource by SIMT, e.g. registers, to hide memory latency can be decreased; second, the on-chip network and memory controller contention can be mitigated, which can help to decrease the related power consumption.

The number of on chip memory controllers is limited by the port count constraint of the 2D chip. 3D stacked memory can release the port constraint by integrating the memory layer on top of the SM layer with TSVs. We propose a 3D GPGPU by stacking main memory, which is shown in Figure 3.

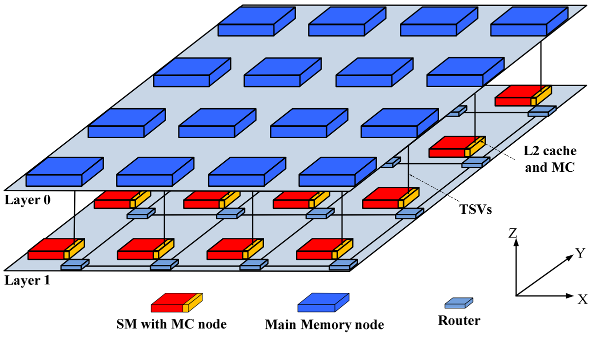

Figure 3. Stacking main memory architecture for 3D GPGPU

The SMs, L2 caches and memory controllers (MCs) are in layer 1, the DRAM memory arrays are in layer 0 and connected to memory controllers by the vertical TSVs.

We use static XYZ routing algorithm that the packets route by the X direction first, then by the Y direction and by the Z direction at last. The link and the router latency are both 1 cycle. For example, if an SM at node 0 sends a read request to the memory at node 15, the request goes along the following route 0-1-2-3-7-11-15 (the reply takes the route of 15-14-13-12-8-4-0), and the total round zero-load access latency is 30 cycles. The zero-load latency presents the ideal latency of packets traversing the network with no contentions. We assume the probabilities of SMs to access MSs as equal. The average zero-load latency for all the SMs is calculated in (1).

$$L_{average\_zero-latency} = \left(\sum_{i=0}^{N_{SM}-1} \sum_{j=0}^{N_{MC}-1} L_{SM_i\_MC_j} \right) \Big/ N_{MC} \qquad (1)$$

In Equation (1), the $L_{SM_i\_MC_j}$ is the zero-load latency between SM at node *i* and MC at node *j*. N_{SM} and N_{MC} are the number of SMs and MCs.

Table 1 lists the individual and average zero-load latency between all the SM nodes and the MC nodes for 2D and 3D GPGPUs in cycles.

Table 1. Zero-load latency for 2D and 3D GPGPUs

	2D	3D		2D	3D
SM0	72	288	SM8	72	256
SM1	72	256	SM9	72	224
SM2	72	256	SM10	72	224
SM3	72	288	SM11	72	256
SM4	72	256	SM12	72	288
SM5	72	224	SM13	72	256
SM6	72	224	SM14	72	256
SM7	72	256	SM15	72	288
Average			288 (2D)		256 (3D)

From Table 1, we can find that the 3D GPGPU has average (288-256)/256=12.5% less communication latency. If we count in the contention of the network, L2 caches and memory controllers, the ratio of memory access latency between 3D and 2D GPGPUs is further decreased.

Hence, the 3D GPGPU should generally outperform the 2D GPGPU. For real life applications, the data accessing pattern may not follow the uniform distribution pattern. In Section 4, we evaluate the performance of the presented 3D GPGPU with stacked main memory.

4. Experiment results and analysis

The configurations of the 2D and 3D GPGPUs are shown in Table 2.

Table 2. 2D and 3D GPGPU system configuration

SM	16
Warp Size	32
SIMD Width	32
Threads/SM	1024
Registers/SM	32768
Shared Memory/SM	48K
L2 Cache	2D: 4 bank×128KB 3D: 16 banks×32KB 8-way assoc., 128B lines
MC No.	2D: 4; 3D: 16
TSV No.	144 per node
Memory Timing	2D: t_{CL}=13, t_{RP}=7, t_{RC}=24, t_{RAS}=17, t_{RCD}=9, t_{RRD}=3 3D: t_{CL}=8, t_{RP}=7, t_{RC}=22, t_{RAS}=15, t_{RCD}=9, t_{RRD}=1
Network	4×4 mesh topology, 1-cycle link latency

The performance improvement of the 3D GPGPU with stacked main memory compared to that of the 2D GPGPU is shown in Figure 4.

We can observe that the performance for the 3D GPGPU outperforms the 2D GPGPU. The main reasons are that for the 3D GPGPU, the memory bandwidth is enlarged and the contention on memory controllers is decreased which lead to smaller memory access latency. The *BFS* demonstrates the most significant performance improvement which is 124.1%. The performance increases by 55.8% on average.

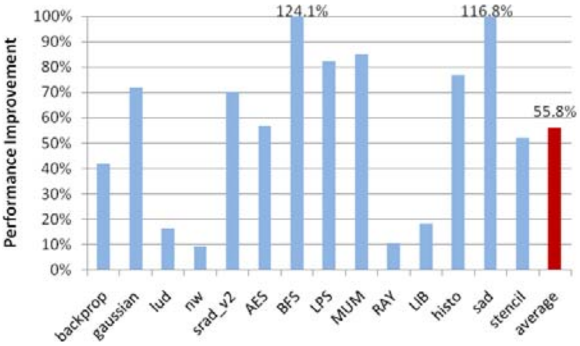

Figure 4. The performance improvement of 3D GPGPU with stacked memory

Figure 5 shows the detailed IPCs of *MUM* for 2D and 3D GPGPUs. The X axis shows the total execution time (in cycles), and the Y axis is the IPC for each of the 16 SMs in the GPGPU. The darker color denotes a greater IPC. We find that 3D GPGPU shows much better IPCs.

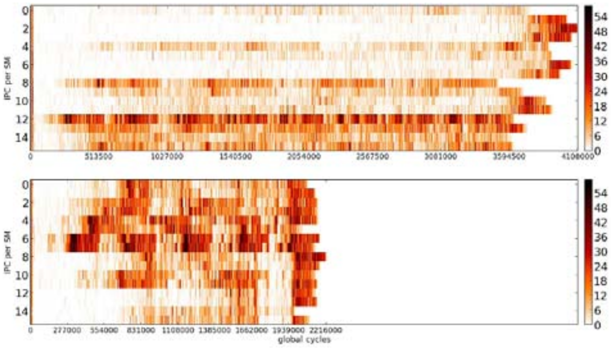

Figure 5. Individual IPC for 2D and 3D GPGPUs

5. Related work

For 3D integration, research that explores memory architectures for 3D GPUs is emerging. However, there are few works that study 3D GPGPUs with stacked main memory. Al Maashri et al. [9] study the performance of 3D stacked L1 caches which include SRAM and MRAM on the GPU. By modifying the organization of caches, and partitioning these caches into multiple layers, the hit rate gets higher while maintaining a reasonable access time. Zhao et al. [10] design a reconfigurable "3D +

2.5D" GPU system. The DRAM memory is 3D stacked memory, whereas the GPU and DRAM are integrated through a interposer (2.5D). Two reconfiguration mechanisms are proposed to optimize the GPU system energy efficiency and throughput.

By leveraging vertical TSVs, several studies show the performance benefits of 3D stacked main memory for CPU based chip multiprocessors (CMPs). Loh [11] explores an aggressive 3D DRAM organization that makes use of the wide die-to-die bandwidth for 3D CMPs. Meng et al. [12] present a framework to model on-chip DRAM main memory and analyze the performance, power, and temperature tradeoffs of 3D CMPs. A runtime optimization policy is proposed to maximize performance while maintaining power and thermal constraints. In [13], Meng and Coskun propose a memory management scheme targeting applications with spatial variations in DRAM accesses for 3D CMPs.

6. Conclusions

In this paper, we conduct a quantitative analysis on the impact of memory access latency for GPGPUs. The experimental results reveal that many applications are constrained by the memory latency. Thus, we present a 3D GPGPU with stacked DRAM main memory. We demonstrate that 3D GPGPU architecture can dramatically promote the performance for memory sensitive applications. The experimental results show that the 3D stacked GPGPU can provide up to 124.1% and on average 55.8% performance improvement compared to a 2D GPGPU scheme. The results of this paper emphasize the importance of considering 3D technology in placement of main memory for future GPGPUs.

Acknowledgments

This work is supported by the National Natural Science Foundation of China under Grant No. 61176024; Research Fund for the Doctoral Program of Higher Education of China under Grant No. 20120091110029; IUR Cooperative Innovation Foundation-Prospective Research Project of Jiangsu Province under Grant No.BY2013072-05; A Project Funded by the Priority Academic Program Development of Jiangsu Higher Education Institutions (PAPD).

References

[1] E. Lindholm, J. Nickolls, S. Oberman, and J. Montrym, NVIDIA Tesla: A unified graphics and computing architecture, *IEEE Micro*, volume 28, issue 2, pp. 39-55, 2008.

[2] Wang, P.-H.; Lo, C.-W.; Yang, C.-L. & Cheng, Y.-J., A cycle-level SIMT-GPU simulation framework, IEEE International Symposium on Performance Analysis of Systems and Software, pp. 114-115, 2012.

[3] Burns, J.; Carpenter, G.; Kursun, E.; Puri, R.; Warnock, J. & Scheuermann, M., Design, CAD and technology challenges for future processors: 3D perspectives, 48th ACM/EDAC/IEEE Design Automation Conference, pp. 212, June 2011.

[4] Van der Plas, G.; Limaye, P.; Loi, I.; Mercha, A.; Oprins, H.; et al., Design issues and considerations for low-cost 3-D TSV IC technology, IEEE Journal of Solid-State Circuits, volume 46, issue 1, pp. 293-307, 2011.

[5] Bakhoda, A.; Yuan, G.; Fung, W.; Wong, H. & Aamodt, T., Analyzing CUDA workloads using a detailed GPU simulator, IEEE International Symposium on Performance Analysis of Systems and Software, pp. 163-174, 2009.

[6] Chen, K.; Li, S.; Muralimanohar, N.; Ahn, J. H.; Brockman, J. & Jouppi, N., CACTI-3DD: architecture-level modeling for 3D die-stacked DRAM main memory, Design, Automation Test in Europe Conference Exhibition, pp. 33-38, 2012.

[7] Che, S.; Sheaffer, J.; Boyer, M.; Szafaryn, L.; Wang, L. & Skadron, K., A characterization of the Rodinia benchmark suite with comparison to contemporary CMP workloads, IEEE International Symposium on Workload Characterization, pp. 1-11, 2010.

[8] John A. Stratton, Christopher Rodrigues, I-Jui Sung, Nady Obeid, vLi-Wen Chang, Nasser Anssari, Geng Daniel Liu, Wen-mei W. Hwu, IMPACT Technical Report, IMPACT-12-01, University of Illinois, at Urbana-Champaign, March 2012.

[9] Al Maashri, A.; Sun, G.; Dong, X.; Narayanan, V. & Xie, Y., 3D GPU architecture using cache stacking: performance, cost, power and thermal analysis, IEEE International Conference on Computer Design, pp. 254-259, Oct. 2009.

[10] Zhao, J.; Sun, G.; Loh, G. H. & Xie, Y., Optimizing GPU energy efficiency with 3D die-stacking graphics memory and reconfigurable memory interface, ACM Transactions on Architecture and Code Optimization, volume 10, issue 4, Article No. 24, 2013.

[11] Loh, G., 3d-stacked memory architectures for multi-core processors, 35th Annual International Symposium on Computer Architecture, pp. 453-464, 2008.

[12] Meng, J.; Kawakami, K. & Coskun, A., Optimizing energy efficiency of 3-D multicore systems with stacked DRAM under power and thermal constraints, 49th ACM/IEEE Design Automation Conference, pp. 648-655, June 2012.

[13] Meng, J. & Coskun, A., Analysis and runtime management of 3D systems with stacked DRAM for boosting energy efficiency, Design, Automation Test in Europe Conference Exhibition, pp. 611-616, March 2012.