Resource Management Design in 3D-Stacked Multicore Systems for Improving Energy Efficiency

Tiansheng Zhang, Ayse K. Coskun
Electrical and Computer Engineering Department, Boston University, Boston, MA, USA
E-mail: {tszhang, acoskun}@bu.edu

Technology scaling and increasing power densities have led to a transition from single-core to multi-core processors, and the trend is now moving towards many-core architectures. Hundreds of millions of transistors can now be integrated on a single chip, however, they cannot be fully exploited due to interconnect/memory latency, power consumption, and yield related challenges. 3D integration is an emerging technology aiming to overcome the limitations faced by traditional (2D) design. 3D stacking enables heterogeneous integration of different technologies, increases transistor density per chip footprint, and improves system performance. In 3D systems, multiple layers are stacked on top of each other and interconnected by through-silicon-vias (TSVs). In this way, it is possible to improve yield (owing to the smaller chip area) and also tackle the latency, bandwidth, and power challenges of interconnects. In chip multiprocessors (CMPs), whose performance is typically limited by the memory access bottleneck, increased communication and memory access bandwidths are distinguishing advantages of 3D stacking. Nevertheless, the benefits offered by 3D integration may be strongly limited without an efficient management of the available resources.

Previous architecture research on 3D stacked systems focuses either on stacking memory layers on top of core logic to boost memory bandwidth or on augmenting the capabilities of planar CMPs including additional logic layers [1], [2]. In both cases, each layer has a different layout, which induces high design cost. A modular design approach on the other hand, dramatically simplifies the chip design process and reduces non recurring engineering (NRE) costs by creating a portfolio of architectures using the same mask set for manufacturing each layer. In addition, homogeneity of the system allows using the same testing protocol for each die within the stack, leading to pre-bond testability without any additional effort for test engineers. Most of the prior work on 3D stacked systems with homogeneous layers exploits the performance or energy efficiency benefits of 3D systems by considering fixed, homogeneous computational and memory resources for cores [3]. Heterogeneous multicore design, however, can bring substantial benefits in reducing energy consumption and cost. This is because applications have varying resource requirements (e.g., in terms of their cache use), which can be addressed by combining cores with different architectural resources in a single chip.

Resource pooling, which enables sharing architectural components of a core with other cores, allows implementing *flexible heterogeneity* in a homogeneous multicore system. In 2D multicore systems, resource pooling among the cores and assisting scheduling techniques have been proposed in prior work [4], [5]. However, the efficiency of resource pooling in 2D systems is limited by the large latency of accessing remote shared resources in the horizontal direction; thus, resource pooling in 2D is not scalable to a large number of cores. 3D stacked systems enable efficient resource pooling among different layers, owing to the short communication latency achieved by vertically stacking and connecting poolable resources using TSVs. A recent technique proposes pooling performance-critical microarchitectural resources such as register files in a 3D system [6]. Their work, however, does not address the cache requirements of applications. The significance of memory latency in determining application performance motivates investigating resource pooling of the caches, which can provide additional low-cost heterogeneity of resources among the cores and bring substantial energy efficiency improvements.

Our research targets improving the performance and energy efficiency of 3D stacked systems by pooling on-chip memory resources among homogeneous logic layers. We investigate on memory resource pooling in 3D stacked systems and the development of resource management policies. Our experiments focus on both embedded system design and high performance system design.

We introduce a novel 3D-CMP architecture based on the integration of homogeneous layers to augment the system performance with minimal design cost compared to conventional planar IC design [7]. The proposed system uses shared memory communication and TSVs to transfer data among the layers. We study performance, power, and thermal characteristics of the proposed architecture. On the 3D-CMP system, we propose a resource pooling technique to optimize memory access latency, allowing cores to leverage the available memory resources on remote layers for minimizing memory contention. The results demonstrate that through utilizing memory resource pooling we can achieve as much as 48.9% performance improvement when the memory on local layer is fully stressed, as shown in Figure 1. We also develop 4 applications to show the potential speedup of 3D CMP and memory resource pooling in 3D systems. To the best of our knowledge our system is the first fabricated CMP that combines multiple homogeneous layers in a modular fashion to increase system performance and apply memory resource pooling. Our results also demonstrate the low power consumption and reliable thermal profiles of the proposed 3D-CMP.

For a broader class of 3D multicore systems, we propose a novel 3D cache resource pooling (3D-CRP) architecture and a runtime management policy [8]. Our cache pooling architecture requires minimal additional circuitry and architectural modifications in comparison to static cache resources. Leverag-

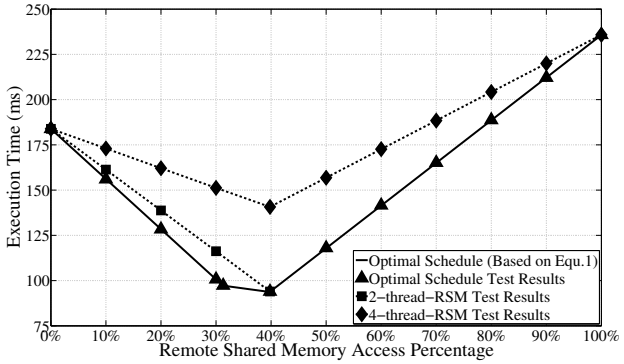
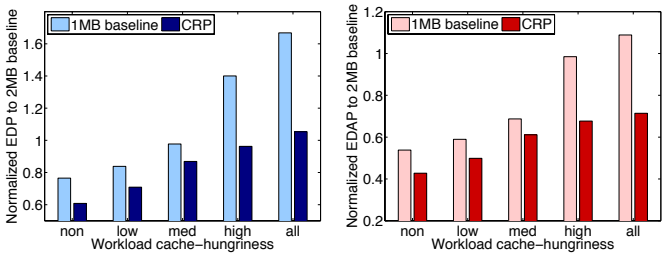


Fig. 1: Test results of different memory resource pooling schedules [7]. *4-thread-RSM*, where RSM stands for remote shared memory, makes all 4 cores access the remote shared memory; *2-thread-RSM* issues 2 cores to access remote shared memory at a time; *Optimal schedule* is the optimal way to schedule remote shared memory accesses.

ing this resource pooling design, we are able to achieve higher performance, lower power, and smaller chip area compared to 3D stacked systems with static cache resources. We also evaluate the proposed method on 3D systems with and without DRAM stacking.

In 3D systems with resource pooling, it is necessary to design policies that are aware of the application cache requirements as well as the memory access rate. In other words, the runtime management policy should manage the utilization of the 3D poolable resources according to the characteristics of workloads that are running on the system, while considering the interplay between performance and energy. To address this need, we design an integrated cache management and job allocation policy that maximizes the energy efficiency of 3D systems for dynamically changing workloads [8], [9]. Our policy predicts the resource requirements by collecting the performance characteristics of each workload at runtime. We then allocate jobs with contrasting cache usage in adjacent layers and determine the most energy-efficient cache size for each application. Our experimental results demonstrate that, using minimal architectural modifications complemented with an intelligent management policy, 3D stacked systems achieve higher energy efficiency through cache resource pooling. For a 4-core low-power system, 3D cache resource pooling reduces system energy-delay-product (EDP) by up to 38.9% and system energy-delay-area-product (EDAP) by 36.1% on average compared to using fixed cache sizes, as shown in Figure 2 (a) and (b) respectively.

Job allocation plays a more significant role as the number of per-layer cores and the number of layers increase. When there are multiple cores on one layer in the 3D system, we call all the cores vertically stacked as a column. And we propose an inter-column job reallocation to balance the cache usage and memory access rate among columns. In order to investigate the scalability of our policy, we evaluate our runtime policy on a larger 3D systemw with 3D stacked DRAM. Figure 3 shows an example of our job allocation policy in a 16-core 3D system with DRAM stacking. This system has 1 DRAM layer at the bottom and 4 logic layers with 4 cores on each layer, where each core has 1MB private L2 cache. The results show that our technique provides 19.7% EDP reduction and 43.5%



(a) Normalized EDP (b) Normalized EDAP

Fig. 2: EDP and EDAP of low-power 3D system with cache resource pooling and its 3D baseline with 1MB static caches, normalized to the 3D baseline with 2MB static caches. [8]

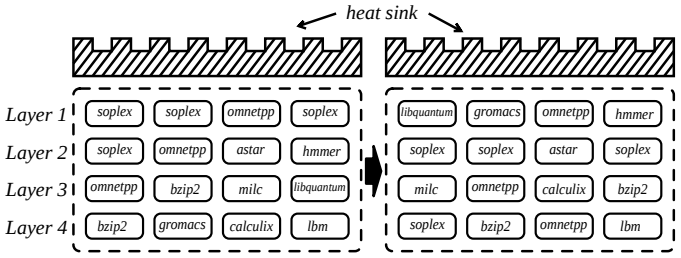


Fig. 3: A job allocation example of a 3D-CMP system with 4 layers and each layer has 4 cores.

EDAP reduction in comparison to a 3D baseline system with 2MB L2 caches. Compared to only using the cache resource pooling method, the performance and energy-efficiency are further improved by 12.3% and 7.8% separately when the memory access rate is also balanced among columns.

The future directions include improving the scalability of the cache resource pooling architecture and integrating our proposed techniques with heterogeneous memory systems.

REFERENCES

[1] D. H. Kim *et al.*, "3D-MAPS: 3D massively parallel processor with stacked memory," in *IEEE International Solid-State Circuits Conference (ISSCC)*, pp. 188–190, 2012.

[2] G. H. Loh, "A modular 3D processor for flexible product design and technology migration," in *Proceedings of the 5th Conference on Computing Frontiers*, 2008.

[3] A. K. Coskun, J. L. Ayala, D. Atienza, T. S. Rosing, and Y. Leblebici, "Dynamic thermal management in 3D multicore architectures," in *DATE*, pp. 1410–1415, 2009.

[4] S. Zhuravlev, S. Blagodurov, and A. Fedorova, "Addressing shared resource contention in multicore processors via scheduling," in *Proceedings of Architectural Support for Programming Languages and Operating Systems*, 2010.

[5] J. Martinez and E. Ipek, "Dynamic multicore resource management: A machine learning approach," *IEEE Micro*, vol. 29, no. 5, pp. 8–17, 2009.

[6] H. Homayoun *et al.*, "Dynamically heterogeneous cores through 3D resource pooling," in *HPCA*, pp. 1–12, 2012.

[7] T. Zhang *et al.*, "3D-MMC: A modular 3D multi-core architecture with efficient resource pooling," in *DATE*, 2013.

[8] J. Meng, T. Zhang, and A. K. Coskun, "Dynamic cache pooling for improving energy efficiency in 3D stacked multicore processors," in *VLSI-SoC*, 2013.

[9] T. Zhang, J. Meng, and A. K. Coskun, "Dynamic cache pooling in 3d multicore processors with dram stacking," *ACM Journal on Emerging Technologies in Computing System*, 2015.