International Journal of Scientific Engineering and Research (IJSER)
ISSN (Online): 2347-3878
Index Copernicus Value (2015): 62.86 | Impact Factor (2015): 3.791

Clustered Look Ahead Prefetching Mechanism In Near Memory Processor

[1]R. Meera, [2]T. Perarasi, [3]R. Ezhilarasi

[1,3]PG Student, Dept of ECE, Karpaga Vinayaga College of Engineering and Technology, Tamil Nadu, India
[2]Associate professor, Dept of ECE, Karpaga Vinayaga College of Engineering and Technology, Tamil Nadu, India

Abstract: ***Near memory processing is a recent solution to overcome the challenges imposed by memory wall. The solution can bridge the performance gaps between processing and memory speeds. Thus a light weight practical memory side prefecther is proposed to improve the performance of near memory processor. The proposed prefetcher exploits the system performance by bringing data from lower level memory before they are needed. The proposed clustered look ahead prefetching is used to predict the future data reference and determine which data are to be prefetched earlier based on the prediction. The proposed CLAP mechanism is fixed in a near memory processor which makes the system performance efficient and reduces the energy consumption to prolong the lifetime of the battery. Therefore the near memory processor reduces the delay and increase the speed of the DRAM system.***

Keywords: DRAM, CLAP, energy efficiency, memory side prefetcher, Near memory processor

1. Introduction

Energy efficiency is more critical in battery operated mobile systems. This become more apparent in high performance mobile system such as smart phone and tablet PCs. DRAM with a capacity of several gigabytes [1], [2]. DRAM is most popularly used as main memory because of high density and low cost. Multi core processors are popular in computer system owing to poor scalability of single core-processors. Programs are getting bigger and trending to larger DRAM accommodation for larger programs in main memory. However larger DRAM capacity is accommodate with higher amounts power and energy, thus increasing cooling cost and reducing the lifetime of the battery. By increasing power and energy consumption of DRAM it is focused on over fetching problem and static power consumption. A memory clustering traffic is proposed which focuses on the energy conservation of RAM, called Clustered Look Ahead Prefetching (CLAP) to reduce the activate / precharge and idle energy consumption of the system. The system memory is predicted using look-ahead prefetching (LAP). In CLAP, prefetching accesses are postponed until normal memory accesses are generated at data path. In this way they can increase the probability of row buffer hits and idle periods with first-ready, first-come, first serve (FR-FCFS). By using this memory scheduling technique it can reduce the number of row activation and idle power consumption. High Latency of off-chip memory accesses has been critical in thread performance. Inter thread memory contention, If not properly managed can have individual thread performance as well as overall system throughput which leads to system underutilization and thread starvation [11]. Previously proposed memory scheduling algorithms are biased in system performance. By using this approaches cannot provide the high fairness and system throughput at same time. Cache performance is discussed for system performance and energy consumption. Cache is a hardware or software components that stores data so that future request for that data can be served faster, the data stored in a cache might be the result of an earlier computation. DRAM has some problem is that the switch is not a perfect valve, so electrons often "leak" away which can cause the device to lose information. Near memory processing has been a topic of great interest among researchers [12], touted a solutions that can bridge the performance gaps between processing and memory speeds by bringing computation closer to where data resides. The challenges involved in integrating memory and logic had prevented them from becoming the mainstream processing paradigm. More recently, NMP systems are regaining a lot of interest [8,7,11], which enable the cost-effective integration of logic processors and memory. Therefore, several recent research studies [11], have proposed NPM systems incorporating simpler processors in logic layer of the memory stacks.

The proposed light weight, memory side prefetching techniques to improve the performance and energy efficient of CLAP based NPM systems that leverage their knowledge of the current state of the memory Systems to prefetch data from row buffers, thereby improving row buffer locality by over 40%. Over experimental results demonstrate that proposed prefetcher improve the performance by over 60%. In CLAP, prefetching accesses are postponed until normal memory accesses are generated at data path. In this way they can increase the probability of row buffer hits and idle periods with first-ready, first-come, first serve (FR-FCFS). By using this memory scheduling technique it can reduce the number of row activation and idle power consumption.

2. Survey

1. Moving computation Near memory has become more practical because of 3D stacking technology. This article discusses in memory map reduce in the context of near memory processor (NMP). The author considers two NMP architecture: one that exploits Hybrid memory cube devices and one that does not. They examine the benefits of different NMP approaches and quantify the potential for improvement for an important emerging big data workload. Ware house scale computing is dominated by systems using commodity hardware to execute workloads, processing large amounts of data distributed among many disles. Several frame works, such as Map Reduce have

Licensed Under Creative Commons Attribution CC BY

emerged in recent years to facilitate the management and development of big data workloads. These systems rely on disles for most data accesses, but placing a large fraction of the data in memory is a growing trend. Spark is a new Map Reduce framework the help programmers tag objects and cache them in DRAM main memory, A primary reason for the resurgence of interest in near data computing (NDC) is the recent emergence of 3D stacked memory and logic products such as Microns Hybrid Memory cache (HMC). Such devices enable collocation of processing and memory in a single package, without impacting the manufacturing process for individual DRAM dies and logic dies. The high bandwidth between the logic and memory dies with through silicon vias can enable significant speedups for memory bound applications.

2. Traditionally, an open-page policy has been used for uni-processor systems and it has worked well because of spatial and temporal locality in the access stream. In future multi-core processors, the possibly independent access streams of each core are interleaved, thus destroying the available locality and significantly under-utilizing the contents of the row buffer. In this work, they attempt to improve row-buffer utilization for future multi-core systems. Main memory has always been a major performance and power bottleneck for compute systems. The problem is exacerbated by a recent combination of several factors - growing core counts for CMPs , slow increase in pin count and pin bandwidth of DRAM devices and microprocessor and increasing clock frequencies of cores and DRAM devices. Power consumption and DRAM latencies are serious concerns in modern chip-multiprocessor (CMP or multi-core) based computer systems. The management of the DRAM row buffer can significantly impact both power consumption and latency. Modern DRAM systems read data from cell arrays and populate a row buffer as large as 8 KB on a memory request. But only a small fraction of these bits are ever returned back to the CPU. This ends up wasting energy and time to read (and subsequently write back) bits which are used rarely. Power consumed by memory has increased substantially and datacenters now spend up to 30% of the total power consumption of a blade (motherboard) in DRAM memory alone. Given the memory industry's focus on costper bit and device density, power density in DRAM devices is also problematic. Further, modern and future DRAM systems will see a much smaller degree of locality in the access stream because requests from many cores will be interleaved at a few memory controllers. In systems that create memory pools shared by many processors locality in the access stream is all but destroyed. Main memory has always been a major performance and power bottleneck for compute systems. The problem is exacerbated by a recent combination of several factors - growing core counts for CMPs slow increase in pin count and pin bandwidth of DRAM devices and microprocessor and increasing clock frequencies of cores and DRAM devices. Power consumed by memory has increased substantially and datacenters now spend up to 30% of the total power consumption of a blade (motherboard) in DRAM memory alone. Given the memory industry's focus on costper- bit and device density, power density in DRAM devices is also problematic. Further, modern and future DRAM systems will see a much smaller degree of locality in the access stream because requests from many cores will be interleaved at a few memory controllers. In systems that create memory pools shared by many processors locality in the access stream is all but destroyed.

3.Cluster-based network servers in which a front-end directs incoming requests to one of a number of back-ends. Consider content-based request distribution: the front-end uses the content requested, in addition to information about the load on the back-end nodes, to choose which back-end will handle this request. Content-based request distribution can improve locality in the back-ends' main memory caches, increase secondary storage scalability by partitioning the server's database, and provide the ability to employ back-end nodes that are specialized for certain types of requests. Network servers based on clusters of commodity workstations or PCs connected by high-speed LANs combine cutting-edge performance and low cost.

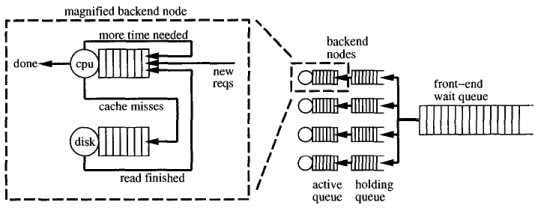

Figure 1: Cluster simulation model

In most current cluster servers the frontend distributes requests to back-end nodes without regard to the type of service or the content requested. That is, all back-end nodes are considered equally capable of serving a given request and the only factor guiding the request distribution is the current load of the backend nodes. With content-based request distribution, the frontend takes into account both the service/content requested and the current load on the back-end nodes when deciding which back-end node should serve a given request. The potential advantages of content-based request distribution are: (1) increased performance due to improved hit rates in the back-end's main memory caches, (2) increased secondary storage scalability due to the ability to partition the server's database over the different back-end nodes, and (3) the ability to employ back-end nodes that are specialized for certain types of requests (e.g., audio and video).

4. The long latency and serialization caused by atomic operations have a significant impact on performance. The data communication is not overlapped with the main computation, which reduces execution efficiency. The inefficiency comes from the data movement between where they are stored and where they are processed. Machine learning (ML) workloads have become an important class of applications these days. ML provides an effective way to model relationships in physical, biological, and social systems, and thus is actively used in a variety of application domains such as molecular models, disease propagation, and social network analysis. Parallel machine learning workloads have become prevalent in numerous application domains. Many of these workloads

Licensed Under Creative Commons Attribution CC BY

are iterative convergent, allowing different threads to compute in an asynchronous manner, relaxing certain read after- write data dependencies to use stale values. While considerable effort has been devoted to reducing the communication latency between nodes by utilizing asynchronous parallelism, inefficient utilization of relaxed consistency models within a single node have caused parallel implementations to have low execution efficiency. As more data becomes available for many tasks, ML is expected to be applied to more domains, thereby making efficient execution of ML workloads on architectures increasingly important. In ML, the key phase is learning; ML inductively learns a model by examining the patterns in massive amounts of data. This requires significant amounts of computation and thus easily takes several days or even months with sequential execution on a single node. As such, most of ML work loads are parallelized to be executed on large-scale nodes, and prior work has mainly focused on reducing the synchronization cost among multiple nodes by utilizing asynchronous parallelism. While a considerable amount of efforts has been focused on inter-node synchronization in large-scale nodes, there has been little focus on the performance of ML workloads on a single node.

3. Clap Mechanism

Look Ahead Prefetching technique is proposed to track the future instructions which are known as program counter and prefetching technique is called as Look Ahead program counter. To generate a prefetching addressed from a previous memory reference address with a stride limits the prefetching to a single loop forward iteration. While in LAPC prefetching the prefetch address is created using previous memory reference and time valve. The prefetchimg data are useless when branch is incorrectly predicted which causes unnecessary memory traffic and cache pollution. In this existing work larger and faster DRAM systems were demanded due to increase in program sizes and popular thread level parallelism. This trend however increases the power and energy consumption of DRAM. In this paper it is based on data prefetching scheme for memory traffic clustering which can achieve a large improvement in the row buffers and power down mode utilization for system performance.

Stride prefetching is implemented normally by comparing addresses used by memory instruction. The stride prefetching requires the address of previous memory access to be recorded with the last detected stride called reference prediction table (RPT). This is used to maintain the information regarding to the recently used load instructions. Prefetching request queue (PRQ) is responsible for clustering prefetching requests generated by RPT. A prefetching request is filtered out to avoid duplicate prefetching for same data.

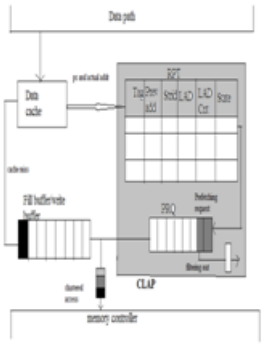

Figure 2: CLAP overall architecture

4. Near Memory Processor

The NMP has a large, high-bandwidth, multi-bank local memory area that it directly manages. they call it as Scratchpad. The NMP has support for vector load and store operations that move data between memory and scratchpad. Data can be moved from the scratchpad to the vector unit in the NMP using multiple concurrent lanes, thus providing high bandwidth access to vector operands. Vector gathers and scatter instructions are also available. To support streaming efficiently, the NMP supports very low overhead producer-consumer synchronization between concurrent threads. Specifically, each addressable unit in the scratchpad (byte) has a full/empty bit, with logic that can block a thread that attempts to read an empty word or write a full word. Since the scratchpad is large, it is impractical to save and restore it upon context switch. Thus, the scratchpad is not part of a thread context — the thread context includes only a small number of scalar and control registers. The scratchpad is accessed with virtual addresses and is shared by all the threads running on the NMP. Although using virtual addresses slightly increases scratchpad access time, the overhead is modest if data is processed using long vectors, as address translation is performed only once per vector operand access in the scratchpad. Such virtualization has the added benefit that scratchpad storage associated with threads that are inactive for a long period of time can be lazily paged out (into main memory) and brought back on demand when accessed. The NMP also includes instructions for bit processing like those of the Cray machines. In particular, it has a bit matrix register that is used for data permuting instructions such as bit matrix multiply. The bit matrix register is also virtualized, so that it does not have to be saved and restored on context switch.

5. Simulation Results

The simulation result is shown in Modelsim 6.2c and synthesized using ISE design suite 14.5 is shown below.

Licensed Under Creative Commons Attribution CC BY

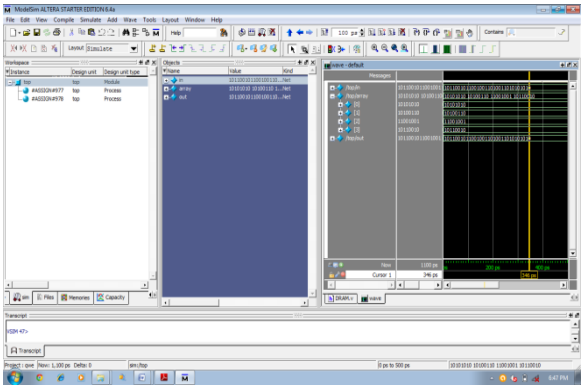

Figure 3: Simulation results for memory processor

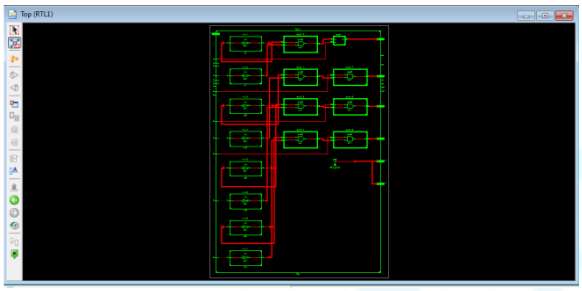

Figure 4: RTL schematic viewer

Figure 5: Technology schematic viewer

Modelsim is offered in multiple editions such as Modelsim PE, ModelSim SE, and Modelsim Xe. Modelsim SE offers high-performance and advanced debugging capabilities, while Modelsim PE is the entry-level simulator for hobbyists and students. Xilinx ISE is a design environment for FPGA products from Xilinx, and is tightly-coupled to the architecture of such chips, and cannot be used with FPGA products from other vendors. The Xilinx ISE is primarily used for circuit synthesis and design, while ISIM or the ModelSim logic simulator is used for system-level testing.

6. Conclusion

In this paper, the proposed light-weight, practical memory-side prefetcher designs, which improve the performance and energy efficiency of PIM systems. The near memory processing is used as a recent solution to overcome the challenges imposed by memory wall. The solution can bridge the performance gaps between processing and memory speeds. Thus proposed prefetcher exploit the system performance by bringing data from lower level memory before they are needed. Thus the proposed clustered look ahead prefetching is used for predicting the future data reference and determine which data are to be prefetched earlier based on the prediction. Therefore the proposed CLAP mechanism is fixed in near memory processors which makes the system performance efficient and reduce the energy consumption to prolong the lifetime of the battery. Therefore the near memory processor reduces the delay and increase the speed of the DRAM system. Thus the power consumption and DRAM latencies are serious concerns in modern chip multiprocessor (CMP or multicore) based computer systems. Thus the Prefetching techniques are applied to the memory processor to improve the memory system performance.

Reference

[1] Jedec standard jesd235a. High Bandwidth Memory(HBM) 2 DRAM, 2016
[2] J. Ahn, S. Hong, S. Yoo, O. Mutlu, and K. Choi. A scalable processing-in-memory accelerator for parallel graph processing. In Proceedings of the 42Nd Annual International Symposium on Computer Architecture, ISCA '15, pages 105–117, New York, NY, USA, 2015.ACM
[3] S. S. Baghsorkhi, I. Gelado, M. Delahaye, and W.-m. W. Hwu. Efficient performance evaluation of memory hierarchy for highly multithreaded graphics processors. In PPoPP, pages 23–34, New York, NY, USA, 2012. ACM
[4] A. Bakhoda, G. L. Yuan, W. W. L. Fung, H. Wong, and T. M. Aamodt. Analyzing CUDA workloads using a detailed GPU simulator. In ISPASS, pages 163–174. IEEE Computer Society, 2009
[5] K. Chandrasekar, C. Weis, B. Akesson, N. Wehn, and K. Goossens. System and circuit level power modeling of energy-efficient 3d stacked wide i/o drams. In Proceedings of the Conference on Design, Automation and Test in Europe, pages 236–241, San Jose, CA, USA, 2013. EDA Consortium
[6] K. Chandrasekar, C. Weis, Y. Li, S. Goossens, M. Jung, O. Naji, B. Akesson, N. Wehn, and K. Goossens. Drampower: Open-source dram power and energy estimation tool
[7] D. Chang, G. Byun, H. Kim, M. Ahn, S. Ryu, N. Kim, and M. Schulte. Reevaluating the latency claims of 3d stacked memories. In Design Automation Conference (ASP-DAC), 2013 18th Asia and South Pacific, pages 657 662, Jan 2013
[8] S. Che, B. M. Beckmann, S. K. Reinhardt, and K. Skadron. Pannotia: Understanding irregular GPGPU graph applications. In IISWC, pages 185–195. IEEE Computer Society, 2013
[9] S. M. Khan, Y. Tian, and D. A. Jimenez. Sampling dead block prediction for last-level caches. In MICRO, pages 175–186, Washington, DC, USA, 2010. IEEE Computer Society
[10] J. Lee, N. B. Lakshminarayana, H. Kim, and R. W.Vuduc. Many-thread aware prefetching mechanisms for GPGPU applications. In MICRO, pages 213–224. IEEE Computer Society, 2010
[11] D. Li, M. Rhu, D. R. Johnson, M. O'Connor, M. Erez, D. Burger, D. S. Fussell, and S. W. Redder. Priority-based cache allocation in throughput processors. In *HPCA*, pages 89–100. IEEE, 2015
[12] X. Zhuang and H.-H. Lee. Reducing cache pollution via dynamic data prefetch filtering. In IEEE Trans. Comput., volume 56, January 2007
[13] H. Wong, M.-M. Papadopoulou, M. Sadooghi-Alvandi, and A. Moshovos. Demystifying GPU micro architecture through micro benchmarking. In ISPASS, pages 235–246. IEEE Computer Society, 2010

Licensed Under Creative Commons Attribution CC BY