Toward Dynamic Orchestration of Data/Power/Process Management for Hybrid Memory Based Systems

Eishi Arima
Technical University of Munich
Munich, Germany
eishi.arima@tum.de

Carsten Trinitis
Technical University of Munich
Munich, Germany
carsten.trinitis@tum.de

Martin Schulz
Technical University of Munich
Munich, Germany
schulzm@in.tum.de

ABSTRACT

The exponential growth of the transistor count on VLSI circuits, known as *Moore's law*, is slowing down, and the end of the technology scaling is predicted to be inevitable. As a consequence, computing system architectures are gradually shifting toward extremely heterogeneous designs consisting of multiple different hardware devices or accelerators in each component. As one example, over the past few years the industry has begun to support hybrid memory systems in their products based on emerging memory device technologies, most prominently HBM (High-Bandwidth Memory) and NVRAM (Non-Volatile RAM). This hardware trend has opened up new research opportunities in the system software and operating system area.

In this position paper, we focus on data, power and process management in hybrid memory based systems, with a particular focus on a *coordinated* and *dynamic* approach. This is based on our key insight, which is brought by our prior studies, that the on such systems memory access/utilization behavior as well as the memory management policy plays an important role for various optimizations, including power management and process (or job) scheduling. In this position paper, we clarify the problem, provide a high-level software architecture, and finally discuss the major challenges to realize it.

KEYWORDS

Hybrid Memory Systems, Data Placement, Power Management, Process Scheduling

FGBS '21, September 21–22, 2021, Online

https://doi.org/10.18420/fgbs2021h-03

1 INTRODUCTION

As VLSI technology scaling is threatening to come to an end, computing system architectures are gradually shifting towards heterogeneous designs. On one hand, CPU-GPU heterogeneous systems are now very commonly used in supercomputing centers, and the architecture in many cases is even more heterogeneous by adding various kinds of accelerators, FPGAs, AI processors [6], or in the near future even quantum computers. On the other hand, if we look at the memory system, several new memory devices are emerging, most notably HBM (High-Bandwidth Memory) [7] and NVRAM (Non-Volatile RAM). As these technologies have strengths and weaknesses in different aspects (e.g., bandwidth vs. capacity), hybrid main memory designs, which compose a main memory with multiple different memory devices, are a promising approach. One example are Intel Knights Landing based systems that have both 3D stacked DRAM modules and DIMM-attached DRAMs within each node [5]. Another example is the use of Intel Optane DC PMMs, an NVRAM solution that can be directly plugged into some of the DIMM slots in a node [4]. Aside from the actual hardware work, this architecture trend is opening up new system software and operating system research opportunities as it requires additional software efforts to fully exploit the performance/capabilities these systems can offer.

In this work, we target hybrid main memory based systems and focus on the needed data, power and process management on them. While the need for a sophisticated data management technique is obvious in order to achieve both high performance and large capacity at the same time on such systems [2], our prior studies have shown that memory access/utilization behavior, such as data footprint size, memory access pattern and memory access intensity, can become a key factor when optimizing power, resource allocations, process (or job) scheduling as well as any other optimization on such systems [1, 3]. This is caused by the fact that memory-related factors can impact performance more significantly than ever before, as such heterogeneous systems combine multiple memory technologies with different performance characteristics leading to heterogeneous

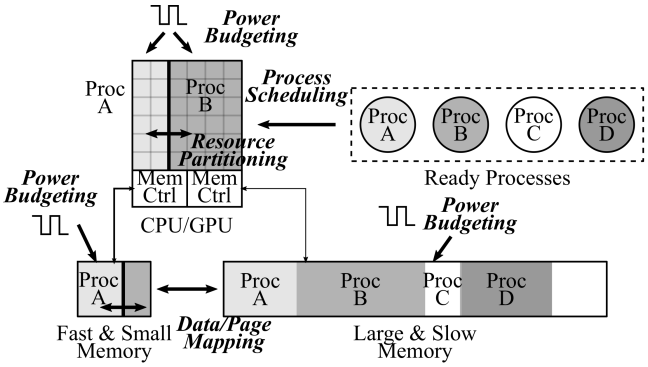

Figure 1: The Problem

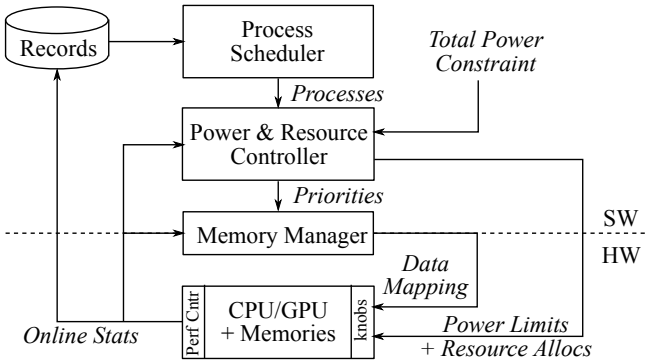

Figure 2: Overall Solution Workflow

and less predictable performance. This key insight motivates us to go towards co-optimizing data, power and process management to better control the on-node heterogeneity and with that to be able to achieve further performance or energy-efficiency improvements. In addition, as the memory access/utilization behavior is *dynamic* information, dynamic analyses/optimizations are also required for this purpose, which then also have to be suited for the operating system layer as well as have to be co-designed with the hardware side. In this paper, we provide a high-level software architecture to orchestrate such heterogeneous system management and discuss several research challenges/opportunities to realize it.

2 PROBLEM AND SOLUTION OVERVIEW

2.1 Problem Description

Figure 1 illustrates the problem we are aiming at solving with the presented approach. As described above, we target computing systems with hybrid main memories that consist of a fast (but small) memory area and a large (but slow) memory region. All of the data utilized on the system are stored on either of them (if not swapped out to the disk). Depending on the data management policy utilized on the system, some of the data can be stored on both of the memories, i.e., in this case the fast (but small) memory is used as an inclusive cache. The data management between them can be conducted by software or hardware, depending on the system. On our test system, we choose single or multiple process(es) from the set of ready processes. Within this study we assume all processes are multi-threaded, i.e., we consider multi-threaded and multi-programmed environment and optimize the on-node co-scheduling. At the same time, for the selected processes, we optimize the core resource allocations as well as the fast (but small) memory allocations — for the latter, we assume that a partitioning or prioritizing feature is supported. Furthermore, we consider also a component-wise power budget controlling by using power capping like in Intel's RAPL, where available, or DVFS (Dynamic Voltage and Frequency Scaling).

2.2 Solution Overview and Major Challenges

As we state above, we dynamically optimize data, power, and process management in a coordinated fashion and Figure 2 describes the overall software architecture to realize our approach. As shown in the figure, our architecture follows a top-down approach, and the details and the major challenges are as follows:

2.2.1 Process Scheduler. The process scheduler sits at the top of the entire optimization stack and chooses one or more process(es) to simultaneously execute on the system. For this purpose, it utilizes the statistics gained during previous runs and uses it to model and evaluate the interference impact for arbitrary process combinations. The interference impact is a function of resource, power and data allocations, and these are optimized for the co-run combinations in the records, thanks to the optimization functions in the other components.

The major challenge here is to construct the models and online-training methodologies to evaluate the interference impact by using statistical histories. The major difference from other scheduling studies is that we pay explicit attention to the memory access/utilization behavior and its access patterns, which is especially critical for hybrid memory based systems.

2.2.2 Power and Resource Controller. This software component optimizes the power budget and the hardware resource allocations for a given set of processes under a given total power constraint. At the same time, it also decides the data allocation priority on the fast (but small) memory for each

process, which is ultimately enabled by the memory manager. For these optimizations, the component utilizes the online statistics collected via the performance counters (and also any other process statistics monitored by the operating system).

The major challenge in this component is developing the methodology to optimize these hardware setups, which should be based on performance and power modeling as a function of collected statistics. In addition, a control theory based optimization is another promising direction. Clarifying a necessary and sufficient set of statistics for the optimization will also be an important direction as it will be different from that for traditional systems composed of monolithic main memories.

2.2.3 Memory Manager. This component is responsible for the data or page management across the multiple different memories. Depending on the system, this part can be fully software, fully hardware or a mixture of them. In this study, we assume this component supports the allocation priority setting feature (e.g., partitioning) for each running process. Regardless of the policy, this part should utilize the hardware statistics to optimize the data allocation. The major challenge in this component is developing an intelligent data/page allocation policy, and according to our prior study [2], an access-pattern-aware optimization will be a good option for this purpose.

2.2.4 Hardware Knobs and Performance Counters. In many cases this is a hardware component (although software components can exist as well in this role) and works as sensors/actuators within the entire optimization loop. In particular, the sensor side is a very important software/hardware interface that significantly contributes to the quality of the optimization. However, if it could provide additional information instead of just counting the number of events on a hardware component, it would be even more beneficial for optimizations like ours. For this reason, revisiting the design of this component is a good software/hardware co-design research opportunity, and we are confident that applying our pattern characterization technique using Bloom filters [2] to this will be a good option.

3 CONCLUSION

In this paper, we targeted hybrid memory based systems and focused on the orchestration of data, power and process management on them. We introduced a high-level software architecture and also discussed the major challenges and the future directions. Further, if we extend it to longer-term challenges, one promising option is to cover the heterogeneity in other components, especially in processing units consisting of CPUs, GPUs and other types of accelerators.

REFERENCES

[1] Arima, E., Hanawa, T., Trinitis, C., and Schulz, M. Footprint-aware power capping for hybrid memory based systems. In *ISC High Performance* (2020), Springer, pp. 347–369.

[2] Arima, E., and Schulz, M. Pattern-aware staging for hybrid memory systems. In *ISC High Performance* (2020), Springer, pp. 474–495.

[3] Arima, E. and Trinitis, C. A Case for Co-scheduling for Hybrid Memory Based Systems, 2019. ICPP, Poster Session.

[4] Izraelevitz, J., et al. Basic performance measurements of the intel optane dc persistent memory module, 2019.

[5] Jeffers, J., et al. *Intel Xeon Phi Processor High Performance Programming: Knights Landing Edition.* Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2016.

[6] Schulz, M., Kranzlmüller, D., Schulz, L. B., Trinitis, C., and Weidendorfer, J. On the inevitability of integrated hpc systems and how they will change hpc system operations. In *HEART* (2021).

[7] Standard, J. High Bandwidth Memory (HBM) DRAM. *JESD235* (2013).