Tarsonis: Dynamic Page Sizing for Disaggregated Memory in Datacenters

Cheng-Hung He
UC San Diego

Jefferson Chien
UC San Diego

Heidi Cheng
UC San Diego

Abstract

Your Abstract Text Goes Here. Just a few facts. Whet our appetites.

1 Introduction

We are able to trace implementation of segments and pages to the earliest days of operating systems. Operating system developers and hardware designers worked together to increase the efficiency of virtual memory. In the past decades, researchers have urged for larger page sizes for its spatial locality, smaller page table, easier virtual memory translation, etc...[6] However, some recent research has indicated that larger page size might not be the most suitable solution in all use cases.[1]. Taking a step forward, we are inspired by many datacenter specific system designs that optimize system implementations for datacenter's special workload and requirements. Also, the rising awareness of microsecond scale latecy and optimization and increasing demands for customizability and flexibility motivated us to design a system that maintains the convenience brought by operating system abstraction for virtual memory while providing effcient memory mapping specifically for disaggregated memory in the datacenters.

Current mainstream operating systems support certain range of page sizes. This is, however, very limited by the underlying hardware architecture as well.[3]. In past research works, we have found some attempts to resolve this issue. Systems such as Kona uses dedicated hardware (FPGAs) to efficiently fetch cacheline size memory from disaggregated memory.[2] Within the Kona paper, Calciu et al. also argued that the standard 4KB page size introduced issues such as dirty data amplification, and proposed a new model to access data in a cache line granularity manner. We however, do not fully agree with this approach, and wish to find a software mechanism that can resolve the dynamically changing page size address translation issue. Further inspired by the thriving centralized managed datacenter architectures[5][4], we came up with an interesting question, if systems like Shenango and ghOSt can leverage centralized management for low tail latency while maintaining sufficient throughput, can do we the same for virtual memory management?

In this paper, we want to explore possibilities for implementing Tarsonis, a userspace dynamic page sizing delegation system for disaggregated memory in datacenters. We have three goals: (1) Argue that varying page sizes is a MUST for modern datacenter workload (2) Explore the possibilities of software mechanism for varying page sizes (3) Design a standardize and intuitive application programming interface for flexible and customizable page sizes for userspace applications (4) Propose an automatic page resizing alorithm that can detect changing page sizing needs and complete necessary modifications with low overheads (5) Resolve virtual memory - physical memory mapping issues to offer compatibility to existing software. We are aware that these goals may not be feasible to complete in the duration of six weeks. Therefore, we want to separate tasks into stages shown in Table 1, so we can trace our progress and make adjustments in necessary.

References

[1] Calciu, I., Imran, M. T., Puddu, I., Kashyap, S., Maruf, H. A., Mutlu, O., and Kolli, A. Rethinking software runtimes for disaggregated memory. In *Proceedings of the 26th ACM International Conference on Architectural Support for Programming Languages and Operating Systems* (New York, NY, USA, 2021), ASPLOS '21, Association for Computing Machinery, p. 79–92.

[2] Calciu, I., Imran, M. T., Puddu, I., Kashyap, S., Maruf, H. A., Mutlu, O., and Kolli, A. Rethinking software runtimes for disaggregated memory. In *Proceedings of the 26th*

Date	Task	Description
Week 5	Workload verifying	Find real workloads (or synthetic ones if not found) to demonstrate how different different applications have different optimized page sizes
Week 7	Software approach investigation	Investigate whether it is possible to achieve varying page sizes with software approach, and if not, propose ways to emulate this experiment to urge hardware designers for such feature in the hardware
Week 9	Implement page addressing	Find ways to implement traditional virtual memory - physical memory mapping.

Table 1: Milestones

ACM International Conference on Architectural Support for Programming Languages and Operating Systems (New York, NY, USA, 2021), ASPLOS '21, Association for Computing Machinery, p. 79–92.

[3] CHEN, R. What are the page sizes used by windows on various processors? https://devblogs.microsoft.com/oldnewthing/20210510-00/?p=105200, May 2021. Accessed: 2022-10-17.

[4] HUMPHRIES, J. T., NATU, N., CHAUGULE, A., WEISSE, O., RHODEN, B., DON, J., RIZZO, L., ROMBAKH, O., TURNER, P., AND KOZYRAKIS, C. *GhOSt: Fast amp; Flexible User-Space Delegation of Linux Scheduling*. Association for Computing Machinery, New York, NY, USA, 2021, p. 588–604.

[5] OUSTERHOUT, A., FRIED, J., BEHRENS, J., BELAY, A., AND BALAKRISHNAN, H. Shenango: Achieving high cpu efficiency for latency-sensitive datacenter workloads. In *Proceedings of the 16th USENIX Conference on Networked Systems Design and Implementation* (USA, 2019), NSDI'19, USENIX Association, p. 361–377.

[6] WEISBERG, P., AND WISEMAN, Y. Virtual memory systems should use larger pages. In *ITCS 2015* (2015).