Exploiting Content Similarity to Improve Memory Performance in Exascale Systems

Scott Levy*[1], Kurt B. Ferreira[2], Patrick G. Bridges[1], Dorian Arnold[1], and David Fiala[3]

[1] Department of Computer Science, University of New Mexico
[2] Scalable System Software, Sandia National Laboratories†
[3] Department of Computer Science, North Carolina State University

1 Background & Motivation

As we consider building the next-generation of extreme-scale systems, many of the biggest challenges are related to memory characteristics. In particular, overcoming challenges related to resilience and memory bandwidth will require innovative strategies for improving the performance of main memory.

DRAM ECC failures are one of the most frequently observed sources of node failure in large scale distributed systems. [6] As node counts continue to grow, traditional checkpoint/restart will no longer be sufficient to efficiently recover from these errors. [4] Moreover, power concerns may exacerbate this problem as we consider deploying low voltage memory chips that are more prone to error. Given that the frequency of memory errors is likely to remain a problem for the foreseeable future, we need to explore ways in which we can prevent memory errors from leading to node failures.

Over the past decade, processor clock rates have plateaued. Nonetheless, the computational power of individual processors have maintained their rapid pace of growth by including more cores per processor. However, the rate at which the number of cores per processor is growing over time is outstripping the rate at which memory access speeds are increasing. As a result, fully exploiting the increasingly powerful multicore processors that will compose future extreme-scale systems requires novel strategies for supplying their constituent cores with sufficient data to keep them highly utilized.

2 Our Position

We believe that there is compelling evidence that exploiting similarities found in the contents of system memory is a promising research approach for improving the resilience and memory bandwidth of extreme scale systems. To facilitate these improvements, we propose infrastructure that collects and maintains metadata about the similarities/redundancies within the contents of main memory as those contents change. For the purposes of this paper, two pages of memory are *similar* if each can be created by applying a small patch to the other. *See e.g.,* [5].

Information about similarities in system memory has been used for more than a decade in virtualization [2], [8], [5] and more recently in HPC [1] to reduce memory consumption. The data collected in these efforts suggests that there is significant similarity within the main memory of a single node. For some applications, including some HPC applications, this similarity may exceed 50%. [5], [1] We believe that this information could be used to aid in recovery from ECC DRAM errors. When an error is detected and a machine check exception is raised, the system software[1] can use the proposed similarity infrastructure to determine whether the failed portion of memory is similar to an uncorrupted block of memory. If a suitable memory block is

*corresponding author: slevy@cs.unm.edu

†Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

[1] In principle, similarity detection and the error correction described here could be implemented entirely in hardware.

identified, its contents can be used to reconstruct the corrupted block of memory (possibly at a different physical address). By exploiting similarities in main memory, we can prevent some ECC DRAM errors from leading to node failure. Additionally, knowing that two pages in main memory are similar may enable us to accelerate checkpoint computation by eliminatng the need to include pairs of similar pages in the checkpoint.

Content similarity in main memory could also be used to improve memory bandwidth. To improve cache peformance and reduce demands on DRAM bandwidth, we may be able to use content similarity information to satisfy a memory request by retrieving a similar page that is already resident in cache. Similarly, in NUMA architectures, we may be able to use content similarity information to acquire requested data from a similar page that is closer to the processor servicing the request.

At the outset, improving resilience and memory bandwidth are two of the most compelling applications of this approach. However, given our proposed infrastructure for collecting and maintaining similarity information, there may be many other ways in which this information could be exploited.

3 Assessment

- CHALLENGES ADDRESSED: initially, our proposed approach addresses the challenges of resilience and memory bandwidth. However, we believe that this approach holds promise in other areas as well (*see e.g.,* [1], reducing the memory footprint may reduce the fraction of system power consumed by memory)
- MATURITY: although content similarity in main memory has not yet been exploited to address resilience or memory bandwidth, significant research into memory de-duplication, *see e.g.*, [2], [8], [5] and [1], indicates that the fundamental approach underlying this approach is promising.
- UNIQUENESS: the challenges of resilience and memory bandwidth are most acutely felt in extreme scale systems. Although improving memory bandwidth and, to a lesser extent, resilience is important in other computing environments, the need to overcome these challenges in smaller systems is less urgent than in the next-generation of extreme-scale systems.
- NOVELTY: memory de-duplication is an established research area. However, we are aware of no other efforts to apply the approach to solve other problems, including resilience and memory bandwidth.
- APPLICABILITY: as noted above, neither of the principal solution we advocate in this paper are likely to have much applicability to other research areas. Nonetheless, with the development of infrastructure to collect and maintain similarity metadata, there may be opportunities to solve problems that are more generally applicable to other computing environments.
- EFFORT: the established body of memory de-duplication literature provides significant guidance on the development of the key parts of the infrastructure. Using the proposed infrastructure to assess the resilience benefit of similarity information should be entail a comparatively modest effort. Assessing the impact on memory bandwidth may be a more intensive research effort as an efficient, scalable implementation will likely require consideration of new hardware functions.

4 Related Work

The technique of de-duplication has been used in virtualization, [2], [8], [5], HPC systems [1] and in storage/backup applications. [10] However, to our knowledge, our proposed application of memory content similarlity to problems that are not directly related to data storage requirements is novel.

In Linux, the machine check exception handler attempts to absorb faults that occur in memory that is not owned by a running process or can be read from a backing store. [7] However, we are aware of little work that allows a system to withstand an ECC DRAM error without re-launching the affected applications.

Recent work on improving memory bandwidth demands has largely focused either compiler techniques for efficient cache reuse [3] or data compression techniques that allow more application data to be delivered to the processor in fewer cache lines. [9] We believe that our proposed approach to use information about memory content similarities to efficiently and dynamically leverage cache content is novel. Moreover, because our approach is entirely agnostic about data semantics it could, in principle, be used in conjunction with these techniques.

References

[1] Susmit Biswas, Bronis R. de Supinski, Martin Schulz, Diana Franklin, Timothy Sherwood, and Frederic T. Chong. Exploiting data similarity to reduce memory footprints. In *Proceedings of the 2011 IEEE International Parallel & Distributed Processing Symposium*, IPDPS '11, pages 152–163, Washington, DC, USA, 2011. IEEE Computer Society.

[2] Edouard Bugnion, Scott Devine, Kinshuk Govil, and Mendel Rosenblum. Disco: running commodity operating systems on scalable multiprocessors. *ACM Trans. Comput. Syst.*, 15(4):412–447, November 1997.

[3] Chen Ding and Ken Kennedy. Improving effective bandwidth through compiler enhancement of global cache reuse. *J. Parallel Distrib. Comput.*, 64(1):108–134, January 2004.

[4] Kurt Ferreira, Rolf Riesen, Jon Stearley, James H. Laros III, Ron Oldfield, Kevin Pedretti, Patrick Bridges, Dorian Arnold, and Ron Brightwell. Evaluating the viability of process replication reliability for exascale systems. In *Proceedings of the ACM/IEEE International Conference on High Performance Computing, Networking, Storage, and Analysis, (SC'11)*, Nov 2011.

[5] Diwaker Gupta, Sangmin Lee, Michael Vrable, Stefan Savage, Alex C. Snoeren, George Varghese, Geoffrey M. Voelker, and Amin Vahdat. Difference engine: harnessing memory redundancy in virtual machines. *Commun. ACM*, 53(10):85–93, October 2010.

[6] Andy A. Hwang, Ioan A. Stefanovici, and Bianca Schroeder. Cosmic rays don't strike twice: understanding the nature of DRAM errors and the implications for system design. In *Proceedings of the seventeenth international conference on Architectural Support for Programming Languages and Operating Systems*, ASPLOS '12, pages 111–122, New York, NY, USA, 2012. ACM.

[7] Andi Kleen. mcelog: memory error handling in user space. In *Proceedings of Linux Kongress 2010*, Nuremburg, Germany, September 2010.

[8] Carl A. Waldspurger. Memory resource management in vmware esx server. *SIGOPS Oper. Syst. Rev.*, 36(SI):181–194, December 2002.

[9] Jeremiah Willcock and Andrew Lumsdaine. Accelerating sparse matrix computations via data compression. In *Proceedings of the 20th annual international conference on Supercomputing*, ICS '06, pages 307–316, New York, NY, USA, 2006. ACM.

[10] Benjamin Zhu, Kai Li, and Hugo Patterson. Avoiding the disk bottleneck in the data domain deduplication file system. In *Proceedings of the 6th USENIX Conference on File and Storage Technologies*, FAST'08, pages 18:1–18:14, Berkeley, CA, USA, 2008. USENIX Association.