Week 2 Progress

Tim Russell

Department of Electrical and Computer Engineering
University of Maine
Orono, USA
tgrussell@eiu.edu

For this week, my work focused almost solely on reading various journal articles to develop sufficient background knowledge of my problem topic, namely the development of an energy-optimal online page replacement algorithm.

Liang, et. al [3] study page replacement algorithms for non-uniform access caches. This extends the work of Belady, as his seminal optimal online algorithm, MIN, presupposes a uniform access cost when choosing a victim page. In the event of cache miss, MIN chooses as a victim the page whose next reference is furthest in the future. This has been proven to minimize cache misses, and in the case of uniform cache access this sole criterion maximizes cache performance. Liang, et. al study a common case in which cache access is not uniform, i.e. in distributed storage systems. In such architectures, the heterogeneous, hierarchical style of memory and storage distribution dictate that different cache management policies be instituted. The analog between the weighted caching problem and the minimal cost maximum flow problem is investigated, and the memory requirements of the implementation of such an approach are shown to be prohibitively large. The authors go on to development MIN-d and MIN-cod, two online algorithms and HD-cod, an offline algorithm to optimize non-uniform cache access.

Zhu and Zhou [4] examine the use of energy-intelligent cache management policies in disk energy management. Though their work focuses on maximizing disk energy savings without regard to memory energy management, the authors' work is extensive and their approach is especially useful in our context. Dynamic programming is used in the development of an optimal energy-aware cache management policy, an approach that I have begun to explore in my own work. Their analysis of their own dynamic programming-developed algorithm mirrors my analysis of another recent algorithms, specifically that such an approach looks attractive but may provide time complexity that is both prohibitively large and difficult to accurately estimate. Two offline algorithms, PA-LRU and PB-LRU, are introduced and then evaluated quantitatively to determine their effectiveness. Both algorithms impressively outperform other common page replacement algorithms in terms of both energy savings and disk performance, showing that a trade off between the two is not necessarily required. Again, though the work is most focused on disk energy savings, several important concepts are used in their offline algorithms; partitioning the cache into regions of varying priority, dynamic programming, and greedy algorithm development.

Chen and Zhang [2] also investigate energy savings in disks through the use of aggressive cache policies. They show that by buffering disks requests, a cache management policy can cause bursty disk access, so that I/O can be bundled together. In such a scheme, the disk is kept from transitioning between power modes and operates almost exclusively in either the low-power or high-power mode. After the bursty accesses, disks spend longer amounts of time idling, allowing for more effective use of energy saving low power modes. These authors implement their work in the Linux kernel and use some Linux-specific hardware and software resources, making it a less-portable solution, but their use of partitioning schemes in the cache (into various priority regions) is interesting.

Albers, et. al [1] investigate the general caching problem to determine approximate solutions to the problem, one that has been shown to be NP-hard. Their work is focused, as conventional, on increasing memory performance, not decreasing power consumption. Several important results, however, are obtained, including the complexity categories of specific examples of the general caching problem, the Bit Model and the Fault Model. Unfortunately, my work seems to be exhibited by what is dubbed the Cost Model, and the authors do not examine the complexity of this model of caching.

References

[1] Arora S. Khanna S. Albers, S. Page replacement for general caching problems. In *Proceedings of the tenth annual ACM-SIAM Symposium (need to finish)*. Need to complete this reference; I couldn't access it from the ACM website.

[2] Zhang X. Chen, F. Caching for bursts (c-burst): Let hard disks sleep well and work energetically. In *International Symposium on Low Power Electronics and Design*, Bangalore, India, August 2008. Indian Institute of Science.

[3] Chen L. Jiang S. Zhang Xiaodong Z. Liang, S. Cost-aware caching algorithms for distributed storage servers. volume 4731 of *Lecture Notes in Computer Science*, pages 373–387. Springer, Berlin / Heidelberg, GE, 2007. ISBN 978-3-540-75141-0.

[4] Zhou Y. Zhu, Q. Power-aware storage cache management. *IEEE Transactions on Computers*, 54(5):587–602, May 2005. DOI: 10.1109/TC.2005.82.