ECE5658: Operating Systems Design

Paper Reading Assignment

2019712410 Jeon Hyung-Joon

Regularities Considered Harmful: Forcing Randomness to Memory Accesses to Reduce Row Buffer Conflicts for Multi-Core, Multi-Bank Systems

Summary:

The objective of this paper is to propose M3 (Mcube, Multi-core Multi-bank Memory allocator), the kernel-level memory allocator for multi-core, multi-bank systems. It has two features. One is that it uses the so-called "memory container". Such system object dedicates multiple banks of memory to a core as much as possible to achieve memory bank parallelism. Another one is that page frame allocation is performed in a random manner. The pages are allocated randomly across banks so that each thread's access pattern is randomized. From this, the frequency of cases where multiple cores access the same bank simultaneously is reduced. Using a tool created by the developers, M3 was implemented and observations were made. Experiments show that page allocation via M3 module is significantly better in terms of performance compared to that of regular page allocation and access pattern. The paper analyzes that this is due to reduction of interference between threads' accesses on the row buffer in memory banks.

Strengths, Weaknesses, and Suggestions for Improvement:

According to the paper, this is one of the first papers that proposed a page allocation scheme that utilizes the concept of randomness to reduce row-buffer collisions. It is clearly shown that their mechanism can lessen time complexity in memory allocation, at least informally in average cases. However, such expectations can be doubted when massive chunks of consecutive memory are to be allocated by different threads, especially when the total amount of memory allocated eventually reaches the maximum capacity. Plus, formal investigation on worst-case time complexity seems insufficient.

To address the weaknesses, as planned by the authors of this paper, their algorithm can be put to more formal analysis for worst-case time complexity. In addition, issues like fragmentation and lock contention, which can almost certainly occur when memory usage amount is near maximum capacity, can be diagnosed and resolved with further improvements. Finally, aside from just random memory allocations, making use of memory isolation techniques can further remove inter-core row-buffer conflicts to improve memory allocation performance.