Paper Critique – Week 10

Name: PHAM CONG VINH Student ID: 2019711010

Traffic Management: A Holistic Approach to Memory Placement on NUMA Systems

In this paper, the author stated and argued about their statement that is, in modern NUMA system the delays caused by remote accessing is not the important source of the performance overhead but the "congestion on interconnect links and in memory controllers" is the main reason.

The proposed solution for this problem is combining an arsenal of techniques (replication, interleaving, co-location) in a way that beyond the locality oriented like previous solution, this resulted in their NUMA-aware algorithm for memory placement, called Carrefour. The author implemented their solution in Linux, evaluated and compared with stock kernel and other patches for Linux. Their results are pretty well as it speed-up up to 3.6 time and never hurt 4% of performance.

The strength of this paper is that they had observed and stated a compelling statement: modern NUMA system had overcame the delays by remote memory accessing and we should move out the focus on locality (placing memory page and thread on same core). Also, author justified the argument by experiment with convincing result.

The limitation of this paper? Even the experiment was conducted before 2013 (the time that the paper was published), conducting experiment on NUMA system with just 4 node a little bit isn't appropriate as the scale of some supercomputer at that time quite bigger than those author's system.

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

The stated problem in this paper is the row-buffer conflict. This happen when a core alternatively accesses different page frames mapped to the same bank or when several cores access the bank simultaneously. This leads to significant performance degradation.

In order to catch up the trend increasing in CPU core and memory DRAM, the author propose that we should rethink about the internal memory allocation policy of operating system's kernel. They presented a counter-intuitive observation that randomization memory access is outperform to sequential memory accesses (or pattern access). They author implemented their solution based on that observation, resulted in M^3 which are a memory allocation algorithm at kernel level for Linux. Also, they conducted their experiment which lead to a positive result that M^3 improved the overall performance for memory intensive applications up to 85% and 40% on average.

The strength of this paper is that introduced memory container notion concept which is representative for a unit of memory formed from a collection of distinct, non-overlapping pages. This notion might facilitate for further optimization solution to memory allocation problem.

The limitation of this paper is that they haven't mentioned about whether their solution might introduce some overhead to non-intensive memory application or not.