An Accurate and Detailed Prefetching Simulation Framework for Gem5

Martí Torrents[1], Raúl Martínez[1], Carlos Molina[2]
[1]UPC-Barcelona Tech, Barcelona, Spain. [2]Universitat Rovira i Virgili, Tarragona, Spain.

Hardware data prefetching is an important technique employed by almost all current commercial high-performance processors. Unfortunately, it is not easy to accurately simulate the effect of a prefetcher in a system, especially in multi-core systems. One of the main challenges is the proper emulation of the network-on-chip (NoC) that communicates the different units (cores, banks of cache memory, memory controllers…) of the processor.

Gem5 is able to simulate a simple (Classic) or a detailed (Ruby) memory hierarchy. The classic memory system from gem5 models a simplistic bus between cores and the shared cache, and between the cache and main memory. This bus is also able to show statistics about contention. However, when the number of cores increases, it is not realistic to use a bus to connect them. For this reason, we need to make use of the Ruby memory system, which is able to model a more realistic switched NoC. Unfortunately, this detailed memory system does not support any kind of prefetching. Therefore, we modified gem5 in order to obtain a simulation framework in which the cores, NoC, and simulator are emulated to a high level of detail, including an accurate simulation of hardware data prefetching.

In order to convert gem5 to a prefetching-aware simulator, some new modules needed to be added to the simulator, as well as making some changes to the current files. Figure 1 gives a schematic representation of the modified (blue) and new (green) modules added to Ruby. As the prefetching module has been included as a protocol-independent component, a new SimObject was added as a wrapper between the cache controller and the prefetcher. It is important to note that the prefetching module can be used in conjunction with any protocol and that it is able to interact with the memory hierarchy at L1, L2, or both of them at the same time. Also, we focused our attention on modifying the MOESI CMP directory. Therefore, protocols other than this one will need to be modified accordingly to work with the prefetching module.

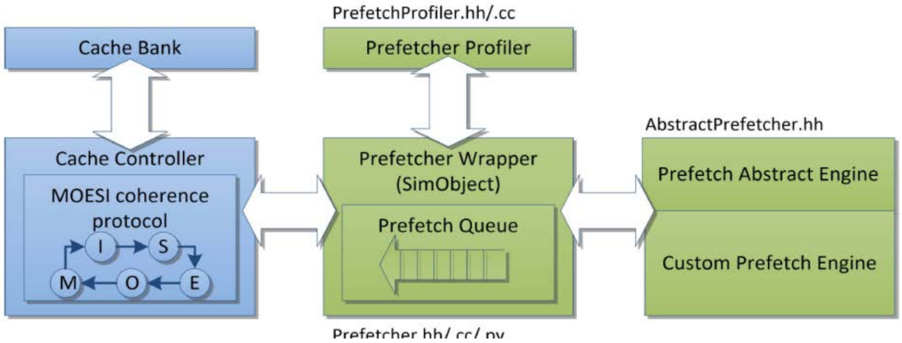

Figure 1: Modified and new elements added to the simulator for prefetching support.

The resulting modified simulator enables the accurate simulation of the combined effect of prefetching and the NoC. The module has been designed to allow prefetching engines to be attached to and switched in the simulator in the easiest way possible. Thus, researchers can test their own prefetching engines in the simulator without needing to have in-depth knowledge of the whole prefetching module. We believe that the resulting simulation infrastructure is useful not only for the evaluation of new prefetching-related techniques in CMP environments, but also for any microarchitecture-related study in a multi-core environment, as it can significantly increase the accuracy of these.

In [1], we introduced and employed this simulation infrastructure to demonstrate that the interaction of prefetching mechanisms with the NoC can be an important factor in multi-core systems. The results obtained showed that the requests generated by a prefetcher have a considerable impact on the global performance of the system, as the requests generated by the prefetcher are traversing the network together with all the other memory requests. For this reason, we believe that it is essential to take into account the prefetching effect on the network when simulating a CMP system. This will enable better-informed decisions to be taken for real systems. Moreover, we also identify, analyze, and quantify in [2], the challenges when trying to prefetch in a distributed and shared memory system, thus paving the way to future research on how to implement prefetching mechanisms.

[1] M. Torrents, R. Martínez, C. Molina. “Network Aware Performance Evaluation of Prefetching Techniques in CMPs”. Simulation Modeling Practice and Theory (SIMPAT), 2014.

[2] M.Torrents, R.Martínez, C. Molina, Prefetching Challenges in Distributed Memories for CMPs, In Proceedings of the International Conference on Computational Science (ICCS'15), Reykjavík, (Iceland), June 2015