Smoothing non-uniform communication latencies for OLTP

Danica Porobic
École Polytechnique Fédérale de Lausanne
danica.porobic@epfl.ch

1. HARDWARE ISLANDS

Transaction processing applications traditionally run on the high-end servers. Up until recently, such servers had uniform core-to-core communication latencies. Now with multisocket multicores, for the first time we have *Islands*, i.e., groups of cores that communicate very fast with cores that belong to the same group and several times slower with cores from other groups. In current mainstream servers, each chip is an Island; as the number of cores on a chip increases, however, soon we will identify Islands within a single chip.

How do transaction processing systems perform on these non-uniform platforms? Aren't multisocket multicores just another form of abundant parallelism? We already have transaction processing systems that scale up on multicores and scale out across machines. Shouldn't one of the existing designs be good enough for hardware Islands?

2. OLTP ON HARDWARE ISLANDS

To answer these questions, we conduct a detailed analysis on the impact of non-uniform hardware topology on the performance of different transaction processing system configurations [4]. We conclude that no single optimal configuration exists: the ideal configuration depends on the hardware topology and the workload. For example, shared-nothing is twice as fast as shared-everything configuration for perfectly partitionable workloads, while situation is completely opposite for non-partitionable workloads and workloads that exhibit heavy skew. Island-sized shared-nothing configurations fall between the two extremes. However, the choice of optimal configuration depends on the combination of workload and hardware topology.

We address this challenge in ATraPos, a shared-everything system that adapts to Islands using automatic partitioning of the system state and dynamically assigning worker threads to specific partitions [3]. In this way, we can remove all intersocket accesses from the critical path of transaction execution for perfectly partitionable workloads. For other workloads, ATraPos relies on finding a good partitioning and placement scheme that balances the load across partitions and minimizes the synchronization overheads across Islands. To ensure robust performance in the presence of shifting workload patterns, we use lightweight monitoring mechanism to detect and quick repartitioning mechanism to adapt to any change.

3. LOOKING AHEAD

In the near future, we can expect the number of cores on a chip to continue increasing without the increase in processor frequencies. Furthermore, power constraints will limit the fraction of cores that can be powered at any point in time leading to core specialization [1]. At the same time network latencies are decreasing and low latency interconnects are becoming mainstream at the level of a rack of machines [5]. In this setting, we will have hundreds of processor cores in a single system organized in a hierarchy of Islands [2, 6].

In order to efficiently utilize such hierarchical systems for transaction processing, we need to fundamentally redesign our software with focus on locality of communication and explicit awareness of the underlying hardware. Transactions typically access a few data items, often creating hotspots, and different transaction types can have very different data access patterns [7]. Our software needs to be agile and continuously adapt its configuration to the workload and underlying hardware topology to serve the workload with maximum efficiency.

4. REFERENCES

[1] N. Hardavellas, M. Ferdman, B. Falsafi, and A. Ailamaki. Toward dark silicon in servers. *IEEE Micro*, 31(4):6–15, 2011.

[2] P. Koka, M. O. McCracken, H. Schwetman, X. Zheng, R. Ho, and A. V. Krishnamoorthy. Silicon-photonic network architectures for scalable, power-efficient multi-chip systems. In *ISCA*, pages 117–128. ACM, 2010.

[3] D. Porobic, E. Liarou, P. Tözün, and A. Ailamaki. ATraPos: Adaptive Transaction Processing on Hardware Islands. In *ICDE*, pages 688–699, 2014.

[4] D. Porobic, I. Pandis, M. Branco, P. Tözün, and A. Ailamaki. OLTP on hardware islands. *PVLDB*, 5(11):1447–1458, 2012.

[5] S. M. Rumble, D. Ongaro, R. Stutsman, M. Rosenblum, and J. K. Ousterhout. It's time for low latency. In *HotOS'13*, pages 11–11, 2011.

[6] R. Sivaramakrishnan and S. Jairath. Next Generation SPARC Processor Cache Hierarchy. In *HotChips*, 2014.

[7] P. Tözün, I. Pandis, C. Kaynak, D. Jevdjic, and A. Ailamaki. From A to E: Analyzing TPC's OLTP Benchmarks – The obsolete, the ubiquitous, the unexplored. In *EDBT*, pages 17–28, 2013.

7th Biennial Conference on Innovative Data Systems Research (CIDR'15) January 4-7, 2015, Asilomar, California, USA.