How to Stop Under-Utilization and Love Multicores

SPEAKERS

Anastasia Ailamaki, EPFL, natassa@epfl.ch

Erietta Liarou, EPFL, erietta.liarou@epfl.ch

Pınar Tözün, EPFL, pinar.tozun@epfl.ch

Danica Porobic, EPFL, danica.porobic@epfl.ch

Iraklis Psaroudakis, EPFL, iraklis.psaroudakis@epfl.ch

INTENDED LENGTH: 3 hours.

TUTORIAL DESCRIPTION

Designing scalable transaction processing systems on modern hardware has been a challenge for almost a decade. Hardware trends oblige software to overcome three major challenges against systems scalability:

1. Exploiting the abundant thread-level parallelism provided by multicores,
2. Achieving predictively efficient execution despite the variability in communication latencies among cores on multisocket multicores, and
3. Taking advantage of the aggressive micro-architectural features.

In this tutorial, we shed light on the above three challenges and survey recent proposals to alleviate them. First, we present a systematic way of eliminating scalability bottlenecks based on minimizing unbounded communication and show several techniques that apply the presented methodology to minimize bottlenecks in major components of transaction processing systems. Then, we analyze the problems that arise from the non-uniform nature of communication latencies on modern multisockets and ways to address them for systems that already scale well on multicores. Finally, we examine the sources of under-utilization within a modern processor and present insights and techniques to better exploit the micro-architectural resources of a processor by improving cache locality at the right level.

Scaling-up on Multicores: In step with Moore's Law, hardware gives us more and more opportunities for parallelism rather than faster processors since 2005. Exploiting parallelism is crucial for utilizing the available architectural resources and enabling faster software. However, designing scalable systems that can take advantage of the underlying parallelism remains a challenge. The inherent communication in traditional high performance transaction processing systems leads to scalability bottlenecks on today's multicore and multisocket hardware. Even systems that scale very well on one generation of multicores might fail to scale-up on the next generation.

We initially teach a methodology for scaling-up transaction processing systems on multicore hardware. More specifically, we identify three types of communication in a typical transaction

processing system: *unbounded*, *fixed*, and *cooperative*. We demonstrate that the key to achieve scalability on modern hardware, especially for transaction processing systems but also for any system that has similar communication patterns, depends on avoiding the *unbounded* communication points or downgrading them into *fixed* or *cooperative* ones. We show how effective our methodology is in practice by surveying related proposals from recent work.

Scaling-up on Multisockets: Data management applications traditionally run on the highest performing servers of the day. Up until recently, such servers had uniform core-to-core communication latencies – multisocket uniprocessors communicate slowly with each other and cores on a multicore communicate fast. Now with multisocket multicores, for the first time we have ***Islands***, i.e., groups of cores that communicate fast among themselves and slower with other groups. Currently, an Island is represented by a processor socket but soon, with dozens of cores on the same socket, we expect that Islands will form within a chip. In this setting, memory access times vary greatly depending on several factors including latency to access remote memory and contention for the memory hierarchy such as the shared last level caches, the memory controllers, and the interconnect bandwidth.

In the context of transaction processing, it can be appealing to regard multisocket as a distributed system and deploy multiple nodes in a shared-nothing configuration. While this approach works great for perfectly partitionable workloads, it is very sensitive to distributed transactions and the workload skew. At the same time, hardware-oblivious shared-everything systems suffer from non-uniform latencies that amplify bottlenecks in the critical path. First, we present a set of best practices for choosing a good configuration based on different properties of workload and hardware topology. Then, we present a system that achieves scalability on multisockets by utilizing hardware topology-aware data structures and dynamically adapting to workload and hardware changes.

On the other hand, analytical workloads consist of ad-hoc, long running, and scan-heavy queries over relatively static data. In order to optimize performance, the execution engine needs to become NUMA-aware by tackling two main challenges: (a) employing a scheduling strategy for assigning multiple concurrent threads to cores in order to minimize remote memory accesses while avoiding contention on the memory hierarchy, and (b) dynamically deciding on the data placement in order to minimize the total access time of the workload. The two problems are not orthogonal, as data placement can affect scheduling decisions, while scheduling strategies need to take into account data placement. We review the requirements and recent techniques for a NUMA-aware scheduler that optimizes performance for a high number of concurrent analytical queries by taking into consideration data locality, parallelism, resource allocation, and sharing opportunities among concurrent queries.

Improving Micro-architectural Utilization: Recent studies analyzing the micro-architectural behavior of OLTP workloads on modern hardware emphasize that OLTP exploits modern micro-architectural resources very poorly. More than half of the execution time goes to memory stalls; as a result, on processors that have the ability to execute four instructions in a cycle, which is the most common on modern commodity hardware, OLTP achieves around one instruction per cycle (IPC). Such under-utilization of micro-architectural features is a great waste of hardware resources.

Several proposals have been made to reduce memory stalls through improving instruction and data locality to increase cache hit rates. These range from cache-conscious data structures and algorithms to sophisticated data partitioning and thread scheduling for data, and from compilation optimizations, advanced prefetching, to computation spreading and transaction batching for

instructions. We illustrate the strengths and weaknesses of each technique with examples from recent work as well as present the key insights behind each of them.

TUTORIAL OUTLINE

- INTRODUCTION AND OVERVIEW (15 minutes)
  - Tutorial overview: goal, audience, and schedule
  - Hardware trends
  - Problem statement:
    - three dimensions of scalability
    - challenges traditional data management systems face on modern hardware

- SCALING-UP ON MULTICORES: Eliminating the unbounded communication (45 minutes)
  - Communication types in transaction processing
  - Recent work on scaling-up OLTP on modern hardware
  - Mapping state-of-the-art design principles to the communication types they eliminate

- SCALING-UP ON MULTISOCKETS: NUMA-aware OLTP (30 minutes)
  - Assumptions modern server hardware with NUMA changes for data management systems
  - Quantifying the impact of non-uniform communication on OLTP performance using various design options and workloads
  - Dynamically adjusting to the hardware topology and workload characteristics while designing transaction processing systems that can scale across sockets

- SCALING-UP ON MULTISOCKETS: NUMA-aware OLAP (30 minutes)
  - Memory access bottlenecks in multisocket multicore architectures
  - NUMA-aware analytical algorithms
  - Outline of the requirements of a NUMA-aware scheduler that handles highly concurrent analytical workloads

- MICRO-ARCHITECTURAL UTILIZATION (50 minutes)
  - Results from recent workload characterization studies
    - How well OLTP exploits aggressive micro-architectural resources?
    - What are the main sources of under-utilization?
    - Road-map for performing an effective workload characterization study
  - Techniques to improve data cache locality
  - Techniques to improve instruction cache locality

- CONCLUSIONS AND FUTURE DIRECTIONS (10 minutes)

TARGET AUDIENCE: Researchers and developers in the area of data management systems who are non-experts on modern hardware and the challenges it poses on high-performance transaction and query processing, and PhD students who are interested in learning more about the underlying hardware and seeking a challenging and high-impact research topic on data management systems.

RELATED PREVIOUS TUTORIALS: The first part of this tutorial, scaling-up on multicores, is presented as part of the VLDB 2013 tutorial titled *Toward Scalable Transaction Processing – Evolution of Shore-MT*. This tutorial, however, has broader scope and includes a range of data management systems and hardware platforms. More specifically, it surveys the concept of scalability for data management systems not just on multicores with uniform access latencies but also on multisockets with NUMA and at the micro-architectural level. In addition, it includes examples from a broader range of storage managers, not just from Shore-MT.

SPEAKER BIOGRAPHIES

Anastasia Ailamaki is a Professor of Computer Sciences at École polytechnique fédérale de Lausanne (EPFL) in Switzerland. Her research interests are in database systems and applications, and in particular (a) in strengthening the interaction between the database software and emerging hardware and I/O devices, and (b) in automating database management to support computationally-demanding and demanding data-intensive scientific applications. She has received a Finmeccanica endowed chair from the Computer Science Department at Carnegie Mellon (2007), a European Young Investigator Award from the European Science Foundation (2007), an Alfred P. Sloan Research Fellowship (2005), eight best-paper awards at top conferences (2001-2012), and an NSF CAREER award (2002).

Erietta Liarou is a postdoctoral researcher at the Data-Intensive Applications and Systems (DIAS) lab of EPFL led by Professor Anastasia Ailamaki. Her primary research interests include database architectures, transaction processing on modern hardware, data-stream processing, distributed query processing, and data analytics with emphasis on very large data management. She received her PhD in Computer Science from the University of Amsterdam, The Netherlands, in 2013, and she has also been with the System S group in IBM T.J.Watson Research Center, Hawthorne, NY, USA and the Intelligent Systems Laboratory in Technical University of Crete, Greece.

Pınar Tözün is a fifth year PhD student at École polytechnique fédérale de Lausanne (EPFL) working under supervision of Prof. Anastasia Ailamaki in Data-Intensive Applications and Systems (DIAS) Laboratory. Her research focuses on scalability and efficiency of transaction processing systems on modern hardware. Before starting her PhD, she received her BSc degree in Computer Engineering department of Koç University in 2009 as the top student.

Danica Porobic is a fourth year PhD student at École polytechnique fédérale de Lausanne (EPFL) working under supervision of Prof. Anastasia Ailamaki in Data-Intensive Applications and Systems (DIAS) Laboratory. Her research focuses on designing scalable transaction processing systems for non-uniform hardware. She has graduated top of her class with MSc and BSc in Informatics from University of Novi Sad and has worked at Oracle Labs and Microsoft SQL Server.

Iraklis Psaroudakis is a third year PhD student at École polytechnique fédérale de Lausanne (EPFL) working under supervision of Prof. Anastasia Ailamaki in Data-Intensive Applications and Systems (DIAS) Laboratory. His research focuses on scheduling highly concurrent analytical workloads and he also co-operates with the SAP HANA database team. He has received his diploma from the School of Electrical and Computer Engineering of the National Technical University of Athens.