論文 / 著書情報
Article / Book Information

題目(和文)	メニーコアプラットフォームのための効率的かつスケーラブルなDBMSに関する研究
Title(English)	A Research on Efficient and Scalable DBMS for Many-core Based Platform
著者(和文)	西方
Author(English)	Fang Xi
出典(和文)	学位:博士（学術）, 学位授与機関:東京工業大学, 報告番号:甲第9905号, 授与年月日:2015年3月26日, 学位の種別:課程博士, 審査員:横田　治夫,佐伯　元司,権藤　克彦,吉瀬　謙二,金子　晴彦
Citation(English)	Degree:, Conferring organization: Tokyo Institute of Technology, Report number:甲第9905号, Conferred date:2015/3/26, Degree Type:Course doctor, Examiner:,,,,
学位種別(和文)	博士論文
Category(English)	Doctoral Thesis
種別(和文)	論文要旨
Type(English)	Summary

（博士課程）
Doctoral Program

論 文 要 旨

THESIS SUMMARY

専攻： Department of	計算工学　　専攻	申請学位（専攻分野）： Academic Degree Requested	博士 Doctor of （　学術　）
学生氏名： Student's Name	西方	指導教員（主）： Academic Advisor(main)	横田　治夫
		指導教員（副）： Academic Advisor(sub)	

要旨（英文 800 語程度）
Thesis Summary（approx.800 English Words ）

Driven by the Moore's Law, computer architecture has entered a new era of multi-core structures. Recently micro-processor manufacturers find that it has become increasingly difficult to make CPUs go faster due to size, complexity, clock skews and heat issues. So they continue the performance curve by another route of developing dual core and multi-core processors. That is, putting multiple CPUs on a single chip and relying on the parallelism ability to get higher performance gain which brings the computing world into so-called multi-core era. The multi-core processor is the mainstream now and there will be many-core processors with more cores on one die in the near future. However, these processors are far from realizing their potential performance when dealing with data-intensive applications such as database management systems (DBMSs). Traditional DBMSs which are dedicated to improving database performance through I/O optimization fail to utilize processor resources efficiently, and the increasing processor-memory gap is slowing down the whole system. Moreover, the increasingly powerful concurrent processing ability of multicore platforms is also stressing the DBMSs. An increasing number of concurrent database processes share resources both at the hardware (caches and memory) levels, and any inefficient resource sharing will create new bottlenecks in DBMSs.

In this thesis, firstly, I analyzed the performance of existing commercial DBMS engine on modern multi-core platform with some emerging new hardware such as SSD, big memory, massive processor cores, and pointed out the I/O sub system is no longer the most important bottleneck for higher system performance. However, with the highly concurrent capability caused by multi-core and many-core processor, the concurrent queries competing in different functions in database engine is causing scalable problems and preventing the system from achieving higher performance. Moreover, the ever increasing memory processor gap makes the memory access is a time costing operation and slowing down the performance of data intensive applications. Therefore, I analyzed the possibility to provide a cache efficient and scalable DBMSs through re-arranging the execution of concurrent database queries on multicore platforms. My proposals are pure middleware, which avoids any modification to existing operating systems (OSs) and DBMSs, thereby making it more practical. This is important because the source code for existing DBMSs is large and complex, making it very expensive to modify.

I proposed CARIC-DA, middleware for achieving higher performance in DBMSs on multicore processors by reducing cache misses with a new cache-conscious dispatcher for concurrent OLTP queries. CARIC-DA logically range-partitions the data set into multiple subsets. This enables different processor cores to access different subsets by ensuring that different database processes are pinned to different cores and by dispatching queries to proper database processes according to the data partitioning information. In this way, CARIC-DA is expected to achieve better performance via a higher cache hit rate for each core's private cache.

I also analyzed the bottlenecks caused by query concurrency in existing database engines on a modern multicore platform using the mixed workload of the TPC-W benchmark and describe strategies for higher scalability and throughput for existing DBMSs on multicore platforms. Firstly, I show how to overcome the limitations of the database engine by introducing a partition-based multi-instance database system on a single multicore platform without any modification of existing DBMSs. Secondly, I analyze the possibility of further improving the performance by optimizing the cache performance for concurrent OLTP and complex join queries.

Moreover, modern multicore platforms are becoming more and more diverse and complex in recent years. Different manufactures are making different efforts to boost their performance by adopting different techniques. It is essential for hardware and software co-design to take advantage of different features of multicore platforms, and to verify the effectiveness of software on different multicore platforms. In this thesis, I analyzed the efficiency of my frameworks on the modern multicore AMD and Intel multicore platforms and provide several insights about how individual cache levels contribute towards improving performance for different multicore platforms.

I implemented prototypes that use unmodified existing Linux and PostgreSQL environments, and evaluated the effectiveness of our proposals. Experiments show that the CARIC-DA framework can provide both higher performance and scalability for both uniform and skewed data sets. On the AMD multicore platform, the L2 cache miss rate can be reduced greatly (by 56%) for select-intensive transactions and the throughput can be improved by 25% for TPC-C transactions by using our CARIC-DA framework. The CARIC-DA also provided stable response times for different data sets, and achieved higher throughput by improving the system scalability on Intel platforms. Experiments using the typical mixed workload of TPC-W benchmark show that the proposal of the multi-instance database system achieved at most 2.5 times higher throughput than the baseline system on the Intel platform. The cache-efficient solution can further improve the throughput by 21% in the proposed multi-instance system.