氏　　　　　名	田中（たなか）　剛（つよし）
所　　　　　属	システムデザイン研究科　システムデザイン専攻
学 位 の 種 類	博士（工学）
学 位 記 番 号	シス博　第 115 号
学位授与の日付	平成 31 年 3 月 25 日
課程・論文の別	学位規則第 4 条第 1 項該当
学位論文題名	Improvement of Cost Calculation Accuracy for Database Query Optimization by Introducing Performance Model Based on CPU Architecture (CPU アーキテクチャを考慮した性能モデルの導入によるデータベース・クエリ最適化のためのコスト計算の精度向上)
論文審査委員	主査　教授　　石川　博 委員　准教授　片山　薫 委員　准教授　横山　昌平 委員　教授　　宮崎　純　（東京工業大学）

【論文の内容の要旨】

Non-volatile memory is applied not only to storage subsystems but also to the main memory of computers to improve performance and increase capacity. In the near future, some in-memory database systems will use non-volatile main memory as a durable medium instead of using existing storage devices, such as hard disk drives or solid-state drives. In addition, cloud computing is gaining more attention, and users are increasingly demanding performance improvement. In particular, the Database-as-a-Service (DBaaS) market is rapidly expanding. Attempts to improve database performance have led to the development of in-memory databases using non-volatile memory as a durable database medium rather than existing storage devices.

For such in-memory database systems, the cost of memory access instead of Input/Output (I/O) processing decreases, and the Central Processing Unit (CPU) cost increases relative to the most suitable access path selected for a database query. Therefore, a high-precision cost calculation method for query execution is required. In particular, when the database system cannot select the most appropriate join method,

the query execution time increases. Moreover, in the cloud computing environment the CPU architecture of different physical servers may be of different generations. The cost model is also required to be capable of application to different generation CPUs through minor modification in order not to increase database administrator's extra duties.

To improve the accuracy of the cost calculation, a cost calculation method based on CPU architecture using statistical information measured by a performance monitor embedded within the CPU (hereinafter called measurement-based cost calculation method) is proposed, and the accuracy of estimating the intersection (hereinafter called cross point) of cost calculation formulas for join methods is evaluated.

In this calculation method, we concentrate on the instruction issuing part in the instruction pipeline, inside the CPU architecture. The cost of database search processing is classified into three types, data cache access, instruction cache miss penalty and branch misprediction penalty, and for each a cost calculation formula is constructed. Moreover, each cost calculation formula models the tendency between the statistical information measured by the performance monitor embedded within the CPU and the selectivity of the table while executing join operations. The statistical information measured by the performance monitor is information such as the number of executed instructions and the number of cache hits. In addition, for each element separated into elements repeatedly appearing in the access path of the join, cost calculation formulas are formed into parts, and the cost is calculated combining the parts for an arbitrary number of join tables.

First, to investigate the feasibility of the proposed method, a cost formula for a two-table join was constructed using a large database, 100 GB of the TPC Benchmark™ H database. The accuracy of the cost calculation was evaluated by comparing the measured cross point with the estimated cross point. The results indicated that the difference between the predicted cross point and the measured cross point was less than 0.1% selectivity and was reduced by 71% to 94% compared with the difference between the cross point obtained by the conventional method and the measured cross point. Therefore, the proposed cost calculation method can improve the accuracy of join cost calculation.

Then, to reduce the operating time of the database administration, the cost calculation formula was constructed under the condition that the database for measuring the statistical value was reduced to a small scale (5 GB). The accuracy of cost calculations was also evaluated when joining three or more tables. As a result, the difference between the predicted cross point and the measured cross point was reduced by 74% to 95% compared with the difference between the cross point obtained by the conventional method and the measured cross point. It means the proposed method can improve the accuracy of cost calculation.

Finally, a method is also proposed for updating the cost calculation formula using the measurement-based cost calculation method to support a CPU with architecture from another generation without requiring re-measurement of the statistical information of that CPU. Our approach focuses on reflecting architectural changes, such as cache size and associativity, memory latency, and branch misprediction penalty, in the components of the cost calculation formulas. The updated cost calculation formulas estimated the cost of joining different generation-based CPUs accurately in 66% of the test cases.

In conclusion, the in-memory database system using the proposed cost calculation method can select the best join method and can be applied to a database system with CPUs from different generations.