SQL Server Database Performance Optimization Based on Multi-layered Queuing Network Model

Jingbo Shao¹,², Xiaoxiao Liu³, Yingmei Li¹, Jingyu Liu¹

¹ College of Computer Science and Information Engineering, Harbin Normal University, 150025 Harbin, China
² Postdoctoral Research Station of Information and Communication Engineering, Harbin Engineering University, 150001 Harbin, China
³ College of Information, Beijing City University, 100083 Beijing, China
{Jingbo Shao, zro_bo}@163.com

Abstract. With the ever-increasing complexity and variety of database workload, database application system has been imposed on higher and higher performance requirements. This paper proposes multi-layered queuing network model for performance prediction. The performance tuning algorithm for SQL server database based on the multi-layered queuing network is presented in detail. And TPC-C benchmark is adopted for simulation. Experimental results show the proposed method achieves 16.8% performance increase on average, and TPS is improved by 40% compared to previous method.

Keywords: performance optimization; multi-layered queuing network; SQL server database

1 Introduction

With the rapid development of science and technology, information system has become necessary in people’s daily life. Optimization of database system plays an important role, and runs through the entire life cycle of database applications.

However performance for most database systems is only assessed after the completion of the entire system at early stage. Worsely is, performance assessment for some database system is performed after system deployment [1]. The earlier optimization work starts, the less costs. Database system performance tuning should be taken into consideration in design stage [2]. Andrew proposed a novel approach to database performance optimization meeting the requirements of query process [3]. Kim proposed a method to improve the performance of the entire sequence match. This method is 16.17-32.64% times faster than when dealing with the real world stock sequence, and 8.64-14.29% times faster when treating large-scale integrated sequence [4]. Queuing network model is used to describe the dynamic behaviors of the Oracle database system design [5]. However the traditional queuing network model is only applicable to modeling independent tasks, while multi-layered queuing networks...
deals with interdependent tasks, and finding out performance bottleneck between them in order for implementation of performance adjustment and optimization of SQL server database system[6]. The authors suggest that database technology should be based on self-tuning loop feedback control system and are bound to build on mathematical models [7].

In this paper, SQL server database performance model based on multi-layered queuing network is presented. And simulation is performed on TPC-C benchmark. The extent of database performance optimization is predicted.

2 Modeling for SQL Server Database System Performance Optimization

Using queuing network to evaluate database performance, the database designer should use database query optimization techniques to estimate costs of a particular transaction executed on a given database [1].

In this model, the server can be expressed as shared CPU resources; transactions are using these resources. Transactions reach the reception desk (CPU), and CPU serves the transaction immediately when idle, if the reception desk is busy, the transaction needs to wait in queue in front of the reception desk until the previous transaction is served, and leaves the system once service is completed.

A first-in, first-out (FIFO) mechanism is followed. When multiple users use SQL server system simultaneously, each terminal establish a connection to the CPU via network; requests from all the terminals are sent to the controller for processing; and the final processed information is sent to user terminals.

3 Experimental Results

Online transaction processing benchmark TPC-C[8] is adopted for test analysis of performance optimization results for SQL server database. TPC-C can deal with new transactions, transaction ending, and transaction status. For TPC-C, the number of transactions that can be processed per second (TPS) is utilized for system performance measurement after running a certain period of time. The metrics in Benchmark Factory is adopted for autonomous testing of database performance. The hardware configuration is shown as Table 1.

Aiming at four different database configurations, the average queuing time is simulated for MQN and the experimental results are shown in Figure 1.

<table>
<thead>
<tr>
<th>Table 1. Database configuration for SQL server system</th>
</tr>
</thead>
<tbody>
<tr>
<td>Processor</td>
</tr>
<tr>
<td>Server</td>
</tr>
<tr>
<td>Client</td>
</tr>
</tbody>
</table>
It can be seen from Figure 2, for the same number of clients, the greater SQL server database is, the longer the average queuing time. With the number of clients increase, the average queuing time upgrades, which conforms to the theory of queuing network.

![Figure 1. Average queuing time for different transactions](image1)

Using the proposed MQN model for SQL server database performance optimization, the related results are shown in Fig.2. In Fig.2, y-axis represents the time of database running (month) and x-axis the total amount of transactions processed per second by database TPS (million). TPS1 and TPS2 represent TPS for SQL server database before and after using MQN. From Figure 2, it can be seen after using the proposed method MQN, performance for SQL server database is improved by 16.8% and TPS by 40%.

![Figure 2. Performance comparison for SQL server2000,4GB](image2)
4 Conclusion

A novel database performance optimization model for SQL server is proposed and established. The multi-layered queuing network model is described, formulated and the related algorithm is outlined. The final analysis of SQL server database performance is made. Experimental results on the TPC-C benchmark using HQN indicate that SQL server database performance optimization achieves 40% more performance improvement than the classical algorithm on average.

References