

Concurrency Control and Self-optimization based on SQL Server Database System

Rongchuan Guo

Jiangxi University of Traditional Chinese Medicine Yao Hesheng Research Center, Jiangxi,
Nanchang 330006, China

Email: rongchuan613@163.com

Keywords: Transaction; SQL Server; concurrent; lock mechanism

Abstract. With the great development of computer science technology and electronic information, all enterprises have established their own information systems. The database as the core and foundation of information system, it has also been more and more widely used. So the performance of the database has been widespread concern in the industry. Traditional performance testing work by hand, it is impossible to simulate high-volume users concurrent access to the real situation of the system, the effect is not very satisfactory. With the continuous improvement of technical means there have been many related software testing tools to simulate a large number of concurrent users of the scene, so the system performance inspection. The concurrency control mechanism of SQL Server is analyzed, and the principle of concurrency service design is summarized on the basis of practice. The typical concurrency problem, custom lock application and so on are discussed. Finally, an application example of concurrency control is given.

Introduction

Database management system in the multi-user concurrent access to data functions, by the concurrent operation of the database consistency and integrity requirements of the challenge [1, 2]. If not necessary for the control of concurrent transactions, even if the program does not have any errors will damage the integrity of the database [3]. In the era of network information, many application systems are faced with the issue of concurrency control, the use of the technology has a direct impact on the development and application of the system. In order to ensure data consistency and integrity, the database management system provides a strong Weak concurrency control functions, a variety of application development tools also provide the realization of the database concurrent control commands [4].

With the deepening of computer application, MS SQL Server, Oracle, Sybase, DB2 and other large multi-user database in the client server distributed computing environment has been widely used in many fields [5, 6]. At the same time, in order to adapt to the development of Internet technology, they have to browser/server (Browser/Server referred to as B/S) mode of calculation provides support. The modern information application system is becoming more and more complex, and the system is based on the same information platform on the mutual cooperation, mutual cross are also increasingly close, for a large amount of data collection, processing, transmission, storage, processing and other data management work put forward new development requirements. It is very important to support the database system to meet the current information application system to achieve efficient information interoperability and processing needs [7]. In addition, due to the continuous development of the database application system, the users of the database are increasing, the throughput of the users is large, the network communication is busy, and the database is growing rapidly, which makes the performance of the database become more and more prominent [8].

At present, various large-scale database application systems, the data volume is more and more large, the data processing more and more complex, the database security and performance requirements are also getting higher and higher. How to make full use of existing hardware and software resources, access to the system's maximum throughput and data processing capabilities has become a database application system is an important research topic. Database system performance is

mainly measured by response time and throughput, the faster the response, the greater the throughput, the better the system performance. In many large-scale data acquisition system, the collection time is large, the user interaction time is short, there is an explosive access to the system, while the existence of the database read and write large amounts of data. Therefore, the optimization and adjustment of database application system performance, so that the entire database application system to meet the above requirements is very important.

1. Introduction of SQL Server

Microsoft SQL Server is a relational database management system. Relational model is the most popular data storage and data retrieval model. And as the back-end of the client server framework SQL Server must have the appropriate mechanism to solve thousands of simultaneous computer access problems, so that each computer system can independently deal with the server from the database of various data. SQL Server database is built on the basis of the Microsoft operating system, which provides a powerful client-server platform, is able to simultaneously support multiple concurrent users of the relational database system. Microsoft SQL Server is established in a mature and powerful relational model, which is currently the most popular model for database storage and retrieval.

The purpose of the database performance evaluation has three options, improvements and design. Specifically, in a number of system options to select the most suitable for the needs of the database system program, that means within a certain range of prices to select the best performance of the system, to achieve better performance than the performance of the existing system defects and bottlenecks To improve and optimize the operating efficiency of the system involved in the future to predict, in terms of performance costs to achieve the best design and configuration. Database performance evaluation is inevitable to be associated with the corresponding computer system. An important aspect of its performance evaluation is the database and system processing power.

2. Testing process and results

LoadRunner with the load test process was shown in Figure 1.

1. Plan: Prepare for testing, such as the selection of typical business processes.
2. Script recording: captures end user activities into a script.
3. Set up the scenario: Use the LoadRunner Controller to set up the load test environment.
4. Run the scenario: Drive, manage and monitor the load test through the LoadRunner Controller.
5. Analysis Results: Use LoadRunner Analysis to create diagrams and reports and evaluate performance.

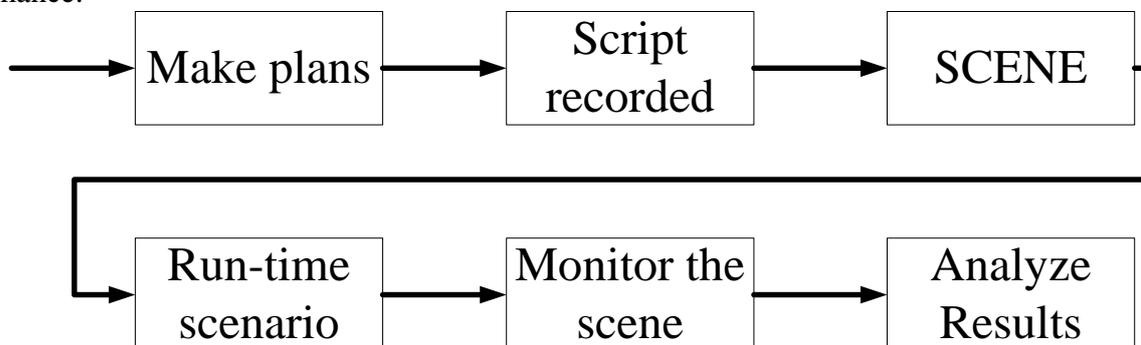
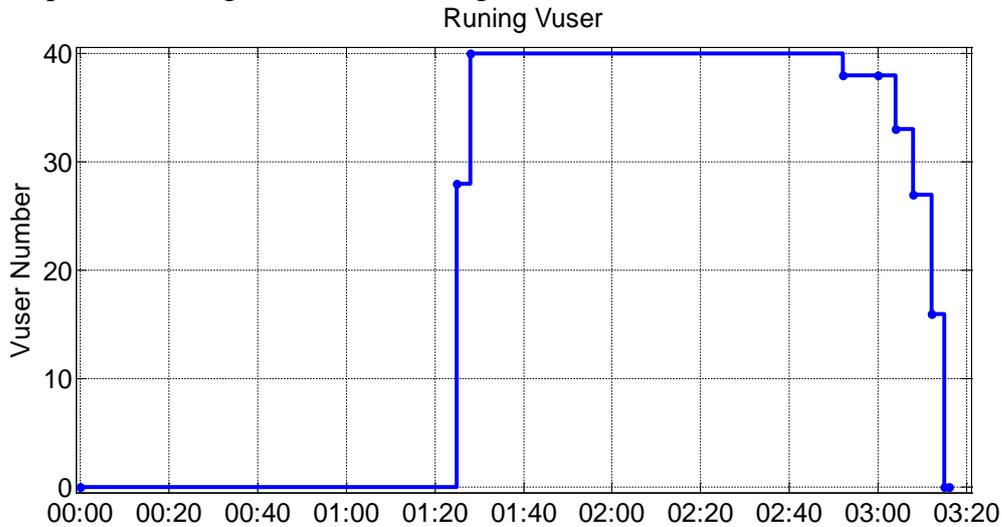


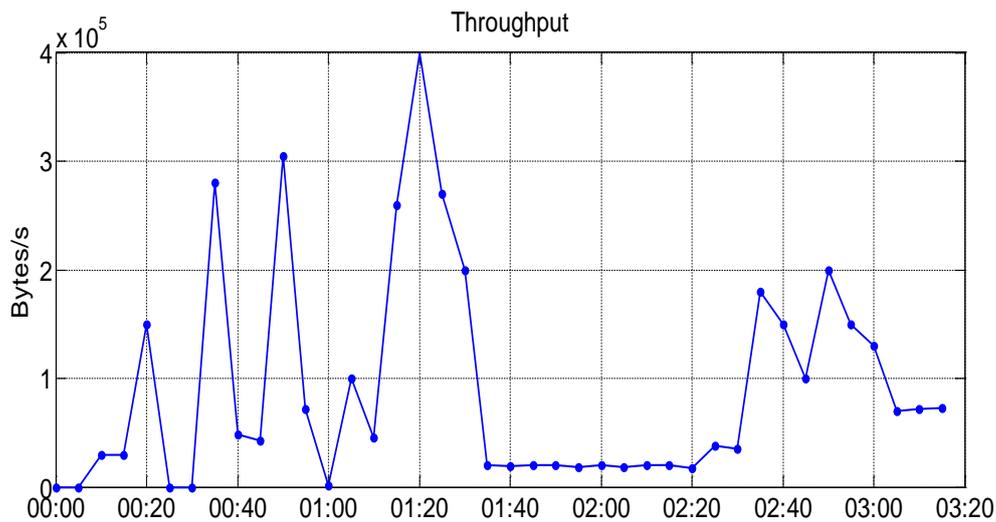
Figure 1. Test flow chart

Multiple users simultaneously load concurrent, concurrent process is only reflected in the beginning of the implementation of the moment, with the server response time of the request is inconsistent or system environment conditions, The number of simultaneous users to load is not a true sense of the concurrent. System pressure is the biggest situation: all users are focused on the system bottleneck to a point on the operation, from the script point of view, this point is the execution of a

script or a statement, in order to simulate the worst case, View the system in the worst case. LoadRunner's integrated monitor suite measures the performance of each single tier, server, and system components during a load test, and Figure 2 shows a breakdown of each scenario.



(a) Running Vuser in the entire scene



(b) Throughput in the entire scene

Figure 2. Scene segmentation chart

Through the transaction response time - the entire scene Figure 2, you can monitor the completion of each transaction to spend time. You can see the customer query business opportunities, browse the product library, view the information industry, the time spent. It can be seen, as the scene running time increases the transaction response time is also increasing.

3. Concurrency control and solution

Concurrency strictly speaking, multiple users perform the same operation at the same time. In software for performance testing, often take the means to check the system's stability and robustness. Concurrent operations are an important operation in performance testing. Concurrency Mainly for the server, the key is to see whether the user's operation on the server had an impact. Therefore, the number of concurrent users refers to the same time with the server to interact with the number of online users. These users are characterized by interaction with the server, this interaction can be either one-way data transfer, it can be two-way data transfer. The running cases of concurrent transactions were shown in Table 1.

Table 1. The running cases of concurrent transactions

Manual lock	Read uncommitted	Read uncommitted	Repeatable read	serializable
(row lock)	lost update	lost update	deadlock	deadlock
(paglock or tablock)	lost update	lost update	deadlock	deadlock
(updlock)	correct	correct	correct	deadlock
(tablockx)	correct	correct	correct	correct
(holdlock)	deadlock	deadlock	deadlock	deadlock
System auto-lock	lost update	lost update	deadlock	
(nolog or readcommitted)	lost update	lost update	lost update	lost update
(readpast)	unavailable	skip lock	deadlock	unavailable
(rowlock xlock)	correct	correct	correct	
(paglock xlock)	correct	correct	correct	deadlock
(serializable xlock)	correct	deadlock	deadlock	deadlock

Database server is the core of the whole database application system. Its performance directly affects the performance of the whole system. Optimized and not optimized system, the test results vary greatly, so optimize the system to give full play to their performance is particularly important. If the system is not optimized, the results will be a user does not make sense or even misleading performance values, is not conducive to competition among manufacturers. However, there is no general optimization rule for various databases and their applications.

Summary

Transaction is an important basic concept of database technology, which plays a fundamental role in protecting the recoverability of database and multi-user and multi-transaction concurrency. Lock is a means to realize multi-user and multi-transaction concurrent processing. Control technology in practice, in-depth understanding and flexible application is even more important. This paper introduces LoadRunner as a testing tool to make up for the original tuning class papers only point of view, the method does not do the full test of the problem. Through the preparation of their own test script, test before and after the adjustment of the database system to assess the number of database server performance improvement. The result is justified.

Acknowledgement

This research was supported by the foundation of Jiangxi University Of Traditional Chinese Medicine(No.2014YHS007) .

References

- [1] Ding Z, Wei Z, Chen H. A software cybernetics approach to self-tuning performance of on-line transaction processing systems[J]. *Journal of Systems and Software*, 2016.
- [2] Mateen A, Raza B, Sher M, et al. Workload management: a technology perspective with respect to self-* characteristics[J]. *Artificial Intelligence Review*, 2014, 41(4): 463-489.
- [3] Gueye S M K, De Palma N, Rutten É, et al. Coordinating self-sizing and self-repair managers for multi-tier systems[J]. *Future Generation Computer Systems*, 2014, 35: 14-26.
- [4] Didona D, Romano P, Peluso S, et al. Transactional auto scaler: elastic scaling of in-memory transactional data grids[C]//*Proceedings of the 9th international conference on Autonomic computing*. ACM, 2012: 125-134.

- [5] Stefanovic N, Radojevic I, Ostojic A, et al. Composite web information system for management of water resources[J]. *Water Resources Management*, 2015, 29(7): 2285-2301.
- [6] H. Herodotou, F. Dong, and S. Babu. No one (cluster) size fits all: automatic cluster sizing for data-intensive analytics. In *Proc. of the ACM Symposium on Cloud Computing (SOCC)*, 2011.
- [7] Transaction Processing Performance Council. TPC Benchmark™ C, Standard Specification, Revision 5.1. Transaction Processing Performance Council, 2002.
- [8] P. Xiong, Y. Chi, S. Zhu, J. Tatemura, C. Pu, and H. Hacigümü ,S. Activesla: a profit-oriented admission control framework for database-as-a-service providers. In *Proc. of the ACM Symposium on Cloud Computing (SOCC)*, 2011.