Performance optimization mehtod on smart grid information platform

Lin Qian, Dong-hui Li, Xiao-zhi Wu, Guang-xin Zhu, Jiang-hui Liu
Information system integration branch company, Nari Group Cooperation, Nanjing, China.
E-mail: {qianlin,lidonghui,wuxiaozhi,zhuguangxin,liujianghui}@sgepri.sgcc.com.cn

***Abstract*—In recent years, with the development of economics, Chinese government feels obliged to build a strong smart grid, state grid company starts the construction process of SG-ERP system comprehensively. However, system performance bottleneck gradually has been spotted due to expanding of IT systems. So it has important theoretical value and strong practical significance to do research on smart grid information platform. This paper proposed a framework of performance tuning for large transaction database used in smart grid based on state grid information platform construction, this method will break system performance bottlenecks, improving the service quality of the platform. The results show that the proposed method can greatly increase the quality of system running, and provide a reliable guarantee for the running in strong smart grid.**

Keywords-transaction database; performance optimization framework ; oracle; smart grid;

I. INTRODUCTION

State grid as a company which undertakes power security of our country carries out a serious of business system including capital and financial management, marketing management, production safety, OA management, human resources, materials management, project management, integrated management systems. Recently, Chinese government committed to construct strong smart grid, so it is urgent to enhance the information processing capacity of business system. IT system of state grid company experienced migration from single server to large-scale, high-speed, comprehensive information system. Especially state grid company deployed SG-ERP business system implementation tasks in the period of 12th five-year plan, so that it becomes a problem of tuning and optimizing existing system to reduce the time to complete business processes. This paper will propose a serious of solution through analyzing factors which affect service quality and framework of improving database performance of producing system based on feature of eight application system and practical experiences from database performance tuning work in various provinces.

This paper arranges as below: chapter 2 describe basic theory and related concept of optimize oracle database performance. Chapter 3 analyzes reasons which generated performance bottleneck and solution to deal with it, finally proposed a framework of refining processing. Chapter 4 we proved the proposed framework greatly promoted system running efficiency. Chapter 5 is ending with conclusion.

II. RELATED KNOWLEDGE

A. *Oracle database deployment in electric power field*

The database was born in the late 1960s and early 1970s, is a collection of massive data and information. Modern database technology has not only the basic functions of data storage, query, but also discovers the potential inherent relationship intelligently. It provides a kind of efficient and convenient solution for information construction in large enterprises and institutions.

ORACLE Company developed a series of Oracle database products with high performance, stability, and cross-platform features, handling numbers of concurrent transaction in 1979. So Oracle database has been first choice software for large database systems in various fields of social production. It has three characters below:

- Provide better performance, reliability, security, and flexibility
- Lower the cost and complexity of IT implementation and management
- Deliver greater productivity, agility, and better business intelligence

State Grid Corporation is related to national fate and power security, provides services to electric power clients from 28 provinces. Its business volume enlarges with high speed in the "Eleventh Five-Year" period. Large Oracle database is good at dealing with concurrent transaction, OLTP request and mass storage. It is a common way to use oracle database to construct business system.

B. *Database performance bottleneck*

Database performance will differ in different application background, so various elements may affect OLTP index efficiency[1]. Typical system bottleneck including

- IO overload
- Limited memory[2] cannot fit for oracle data buffer requirements
- Oracle shared pool is not enough for dealing with many concurrent processing.
- parameters of operating system are set inappropriately
- Configuration of oracle system.

C. *Database performance tuning*

Oracle database ordinarily is deployed in large-scale, high concurrence volume and complex index system. But with the development of business transaction in modern society, massive data which arrived in time are obliged to select an efficient method to optimize existing database, so

Published by Atlantis Press, Paris, France.

that it could improve the ability of handling a large number of OLTP requests, decrease memory and CPU consuming rate. It is very significant to achieve the destination below in the work of database tuning.

- Update the throughput rate: The problems of database performance are often shown as low efficient throughput and long time waiting for accessing to business system. So engineers try their bests to reduce hard disc access, instead to read data from high speed memory.
- Response time is a critical index to judge system performance. So it is urgent to find a way of arranging SQL running order properly, reducing index time for users.

III. DATABASE PERFORMANCE TUNING SCHEME FOR SMART GRID INFORMATION SYSTEM

To provide sustained, stable and efficient service for smart grid, we need to execute health inspection and system test to collect details from oracle database which is running on server before tuning its performance.

Then the problem of oracle performance could be found according to reading AWR report, and begin to run tuning program. Analyzing the reports from system, we should tuning system in two directions, first one is application level, and the other is system level. The paper will discuss tuning method with them.

A. SQL tuning(application level)

SQL language is so smart that different sentence will generate execution time and efficiency. A large number of applications did not consider SQL sentence adjustments to be a significant partition. Application developers always put their eyes on function not running efficiency. So there are numbers of SQL problems in application system. Figure 3.1 shows the proportion of SQL problems happened in electrical power system.

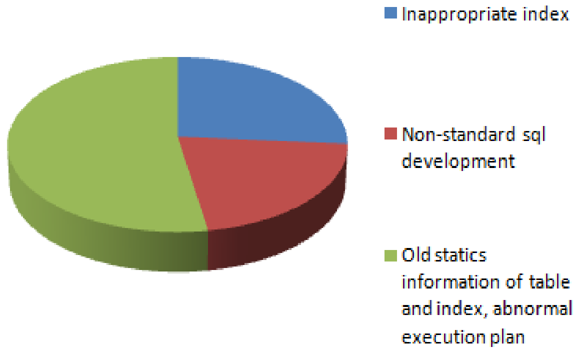


Figure 3.1. SQL problem proportion.

SQL tuning will be done with following steps:

- Create or alternate index on proper position. High quality index will reduce whole table scan and query time, finally speedup executive efficiency.
- Make sentence structure simple through cutoff redundant and overlap statement. We could split a complex SQL into several short and refined parts.
- Arrange SQL sentence execution plan[3] rationally. We could adopt different SQL running order to cater numbers of application Scene.
- Clear up fragments of table space and update statistical information frequently.

B. Opration system tuning

Oracle database performance relies on operation system heavily. If it cannot provide enough ability of dealing with massive data, it will not make influence on oracle running status. So it is very important to tune parameters of database running. Common method including:

- Reducing minperm%, maxclient%, maxperm% value[4]. Parameters given above can adjust File memory which cannot be drop easily, so it will make sense of reducing proportion of this part.
- Enlarge SWAP memory space. Try our best to set SWAP partition into half physic memory, at least one third of it.
- Disc I/O operation is an important bottleneck of database performance. It will cutoff extra I/O consume largely if increasing hit rate. In addition, it is still an effective way to use index, partition and load balance.

C. Parameter of Oracle tuning

- Tables and index adopt large BLOCK SIZE[5] which causing hot block conflict. The method could not help lifting up overall performance of the database. Contrarily it may increase GLOBAL BUFFER BUSY.
- DB CACHE is a key element of enhancing system performance. Some inquire task need to scan the whole table, enlarging DB CACHE will shorten running time, relieve pressure of database.
- Enlarge shared_pool_size to solve computational resource contention during business peak. Adjust shared_pool_reserved_min_alloc to clear up fragment of shared pool, block system down due to ORA-4031.
- Tuning PGA, SGA to perfect status, enlarge log buffer size. SGA (System Global Area) includes shared pool which stores SQL sentence and definition, database buffer cache which stores blocks from index result and log buffer area. On one hand We could generally compute suitable value of database buffer cache according to the formula

$$\text{hitrate=1-}\frac{physicalreads}{dbblockgets+consistentgets} \quad (1)$$

On the other hand, we use formula

$$\frac{\sum(gets-getmisses-usage-fixed)}{\sum(pinslibrarycache)} \quad (2)$$

to know its size. At last, request failure is a criterion to judge log buffer size.

D. *Construction tuning*

The structure of the database objects (mainly referring to tables and indexes) include logical and physical structure plays a crucial role in the performance of the database [6]. In general, after application system is online, adjusting logical structure of the original table become so complex that it is more difficult to complete. System performance should be tuned by adjusting the physical structure of the original table, or the structure of the related index. In addition, it could also add new auxiliary object optimizing performance.

- KEEP pool[7]: put frequent reading tables and indexes into KEEP pool, this method may promise these objects will not be swapped out of buffer pool, and thereby decreases physical reading times. Considering objects size and KEEP pool size, system should select hot table and index into KEEP pool carefully, and improve efficiency.
- Index: it is very important to create the appropriate index. In one side, it must comply with some certain rules during constructing SQL sentences, such as: table fields cannot be with the function. In the other side, there are a large number of fragments living in index, it can enhance performance of system through rebuilding index online, decrease system consuming.
- Table fragments arrangement and table storage parameter tuning[8]. Some tables often insert and delete data, so they generate many fragments which occupying large database space. "Alter table <table name> move" and rebuild table after RENAME are two main methods. If the tables need lots of concurrent modifications, we could increase INI_TRANS value to forbid deletion and modification after data has been inserted. Query table is given smaller PCT_FREE, serious hot block conflict table gets greater PCT_FREE value.
- SEQUENCE[9] adjustment: SEQUENCE contention is serious, the loss rate is high. Its performance can be optimized by increasing the portion of SEQUENCE CACHE.
- ORACLE partition[10] divides large tables and indexes into manageable chunks, thereby avoids managing each table as a separate object, and provides scalable performance for large amounts of data. The partition technology reduces management time, because the operations are assigned to smaller storage units and the parallel processing framework improves performance. In addition it can also shield the partition of the fault database, increasing the usability of database system.

IV. ANALYZATION OF TUNING PERFORMANCE

This paper will shown the results in producing environment from electrical power company of a certain province after tuning the database system. It has been proved that the method, focusing on smart grid information can enhance availability, reliability and efficiency of existing database system. We will compare original database and tuning performance database and analyze performance uplifting degree.

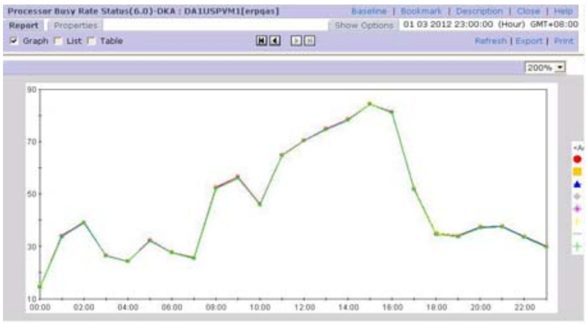
A. *Current situation of system*

We could notice that three key application systems have taken on a lot of pressure on table I. Some important servers run for a long time, so they are urgent to update.

TABLE I. CPU LOAD OF KEY APPLICATION SYSTEM

System	*Configuration*	*CPU status description*	*Remark*
Marketing Production Center system	Two IBM P6-570 32 cores 4.2GHz CPU /256GBmemory	CPU>50 is 85% CPU>75 is 22% CPU>85 is 9%	The pressure of system is serious
		CPU>50 is 57% CPU>75 is 10% CPU>85 is 5%	
Marketing Management Center system	Two IBM P5-595 16 cores 1.6GHz CPU/64GB memory	CPU>50 is 60% CPU>75 is 27% CPU>85 is 16%	6 years serving, it is urgent to update
		CPU>50 is 41% CPU>75 is 16% CPU>85 is 8%	
Energy collection system	Two HP superdome 16cores 1.6GHz Itanium CPU /64GB memory	CPU>50 is 30% CPU>75 is 4% CPU>85 is 1%	Five years serving, it is urgent to update,2.5 million data need to handle in 2012
		CPU>50 is 20% CPU>75 is 2% CPU>85is 0.5%	

The figure 4.1 given below shows the bottleneck of cluster storage HDS USP-VM. When the number of application system is increasing, the amount of data is increasing exponentially. From 2011, USP-VM1 suffers from apparent performance problem.


Processor Busy Rate Status(6.0)-DKA : DA1USPVM1[erpqas]

01 03 2012 23:00:00 (Hour) GMT+08:00

☑ Graph □ List □ Table

Figure 4.1. usage of storage system.

B. *SQL tuning performance*

According to SQL sentence tuning, the application system has been improved largely. The sentences execution time has been reduced significantly. So tuning system greatly enhances the experience of index operation.

We sample the database before and after tuning operation separately. At the same time we compute multiples function (1)

$$\text{Improving multiples}=\frac{\text{index running time before tuning}}{\text{index running time after tuning}} \qquad (1)$$

Table II shows the part result of oracle database tuning, because of confidentiality requirements.

TABLE II. SQL TUNING RESULT

Number	SQL code	Improving Multiples
1	Ak****af	1800
2	Cm****q0	120
3	Af*****d5	100
4	2gh*****px	20
5	81*****ma	25
6	Dm*****mm	42

C. *Storage tuning result*

Defragmentation and database reduction have saved 2.3TB database space, reduced the consumption of storage space. We could watch it from Figure 4.2.

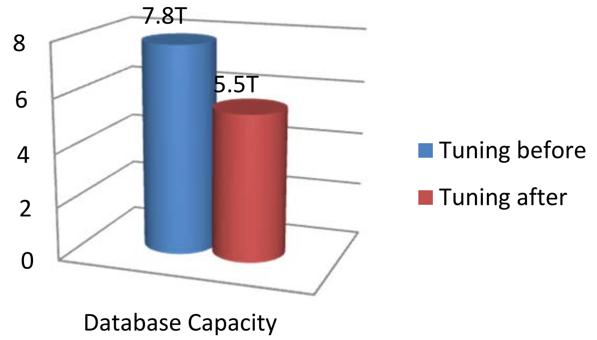


Figure 4.2. Storage tuning result.

D. *CPU and database result*

The indicators of judging performance have been improved apparently through oracle database system tuning in Figure 4.3. At the same time, we have also noticed that CPU overloading has disappeared, CPU peaking has gone to normal level smoothly.

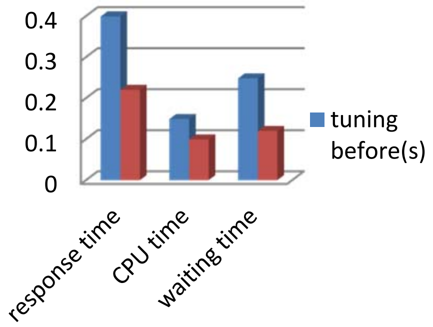


Figure 4.3. Parameters of performance comparison.

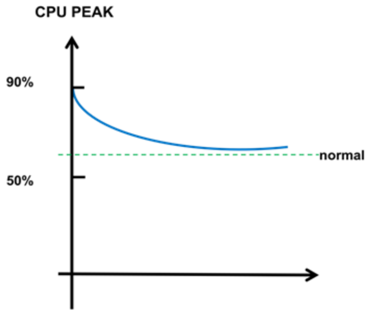


Figure 4.4. CPU load status.

V. CONCLUSION

It is a constantly developing and summary processing for oracle database tuning, while it is a systematic project which involving many aspects. The method proposed analyzes the various factors that affect performance of oracle, it is obliged to choose reasonable optimization measures for different specific situations. It is also need to understand the new technologies deeply, and finally it can be provide better tuning efficiency using advanced features of Oracle.

ACKNOWLEDGMENT

This work was supported by Nari group company(State Grid Electric Power Research Institute) Information systems integration branch company. Thanks for my parents and my professor Liang-xi Qin.

REFERENCES

[1] Oracle Corporation. Oracle Data Mining Concepts,10g Release 2. http://otn.oracle.com/, 2005.

[2] B. Dageville, M. Zaït, SQL Memory Management in Oracle9i. VLDB 2002: 962-973, Hong Kong, China.

[3] N. Bruno and S. Chaudhuri. Automatic Physical Database Tuning: A Relaxation-based Approach. In Proceeding of SIGMOD, pages 227–238, 2005.

[4] B. Dageville and K. Dias. Oracle's Self-Tuning Architecture and Solutions. IEEE Data Engineering Bulletin, 29(3):24–31, 2006.

[5] B. Dageville, D. Das, K. Dias, K. Yagoub, M. Zait, and M. Ziauddin. Automatic SQL Tuning in Oracle 10g. In Proceedings of VLDB, pages 1098–1109, 2004.

[6] Choudhuri and G. Weikum, "Rethinking Database System Architecture: Towards a Self Tuning RISC style Database System", in VLDB, 2000, pp 1-10.

[7] DageVille and K. Dias, "Oracle's self tuning architecture and solutions", IEEE Data Engg. Bulletin, Vol 29, 2006.

[8] K. Brown, M. Carey, and M. Livny, Managing Memory to Meet Multiclass Workload Response Time Goals, VLDB 1993, Dublin, Ireland, 328-341.

[9] Chaudhuri, S.; Weikum G, "Foundations of Automated Database Tuning", Data Engineering, April 2006.

[10] Gerhar Weikum, Axel Moenkerngerg et. al., Self-tuning Database Technology and Information Services : From wishful thing to viable Engineering", Parallel and Distributed Information System 1993.