RESEARCH PAPER | Engineering | Volume : 4 | Issue : 7 | July 2014 | ISSN - 2249-555X

A Novel and Autonomic Model to Enhance Performance and Security of a Relational Database Management System.

KEYWORDS: Log Shipping, SQL Injection, e-Purge, Indexing.

S.Deepa Lakshmi
M.Tech(IT) Student, School of Information Technology & Engineering, VIT University,Vellore,India

S L Aarthy
Assistant Professor, School of Information Technology & Engineering, VIT University, Vellore – 632014.

ABSTRACT *The most important challenge faced from small level to top level business companies is how efficiently the company's information are stored and protected from unauthorized use. This information includes past as well as current transactions, top priority information about its customers, vendors, etc. As the business grows, this information also grows tremendously occupying huge volumes of storage thereby paving the path to space as well as performance overhead. Database servers are used in various software environments in which each application has its own functionality and workload. Now this is the most challenging problem to be resolved by the database administrator who is sole responsible of the server holding the confidential data of a company. Also according to OWASP top 10 web application attacks, SQL Injection has been ranked as first, which is based on altering input queries to the database and eventually getting sensitive data information or even corrupting the database by appending any drop sql commands. Therefore suitable methods needs to be deployed immediately in order to save any information loss or piracy which impacts the growth of the business highly. In this paper, we have developed a model to improve performance as well as security of the database by setting new strategies and by exploiting various features of SQL*

1. INTRODUCTION

A database administrator is responsible for the security , integrity and performance of all databases in a server. To improve performance, a DBA should choose indices that is suitable for a particular workload. This process is termed as Index tuning[1]. System can initiate this process by recommending suitable indices by evaluating the process workload. However choosing the right indices is a challenging optimization task: some indices may be chosen for a particular condition queries whereas this could create a performance overhead on other cases. Due to this, every DBA rely on automated tools for indexing which can choose set of indices after analysing the workload. We have developed a novel model which would perform index tuning along with other performance and security measures of the workload. As it is perform online with other query processing techniques a DBA can request for indices at any point of time. Another important responsibility of a DBA is to upgrade the existing database server to higher versions to exploit new features added in it. Installing and testing latest versions of database servers is a complicated task of a DBA. Our model studies the important features of latest versions of SQL and give it as an advantage to the existing product. SQL Log Shipping is a high availability technology which helps in database recovery during failure. It involves taking a database backup and following transaction log backups from the source or primary server and restoring it into the destination or secondary server. Purging is one of the performance measure that is attracted by DBA's. As it directly reduces the space of the database used, it increases the performance of the database drastically. Database approach is to maintaining the data consistent across different databases. In a business environment data integrity is established between locations where different servers are hosted and also across databases where postings done from one to other viz. sales to general ledger. In such a scenario, purging of data in one database should also be handled in the other database so that there would not be any data mismatch across modules. We have proposed a new purging technique e-Purge which sustains the data integrity across different modules thereby different databases in a traditional business setup. Finally as a security measure we identified the OWASP(Open Web Application Security Project) topmost web application attacks viz. Sql injection and validated the queries which can be attacked using sql injection methodology.

2. RELATED WORK

Database Performance tuning is a field where the scope for research is far above the ground. We were motivated by the below topics, namely semi automatic index tuning, no db philosophy, stochastic database cracking over adaptive indexing and database virtualization.

2.1 Semi automatic index tuning

In this paper, a semi automatic index tuning tool was proposed which aims in providing better index recommendations to DBA at any point of time. It also gives scope to the DBA to provide feedback about the index that is chosen. WFIT(Work Function Algorithm for Index Tuning) [1] index-tuning algorithm[1] was proposed which implements semi-automatic index on end to end basis. This algorithm takes the current workload and the DBA feedback as input and computes a set of indices as output. The workload (Q) of a database is modelled as a stream of queries and updates[1].

2.2 NoDB Philosophy

A new need as a result of data deluge[2] is described as NoDB philosophy in which the traditional row-wise data is transformed into NoDB system. Querying directly from raw data files instead of data loading or restoring is implemented in this paper. A typical row-store dbms arranges the data in tuples one after which are stored in pages in the background. These pages should be loaded first in order to convert from page format to tuple format. PostgreSQL 9.0 is used to prove that the NoDB philosophy provide better access to the data than the traditional dbms. Positional maps and caching technologies are proposed in handling raw data files.

2.3 Stochastic DB Cracking

In this paper, database cracking[4] is proposed to be more efficient to adaptive indexing[3] which is creating and maintaining indices. Cracking aims at automatically adapting appropriate indices without human intervention. Stochastic cracking[3] uses a decision making algorithm to reorganize data from query hints. DDC(Data Driven Center) algorithm was proposed which performs range partitioning of column-store information. It aims in halving the array and reorganize it based on the attribute.DDR(Data Driven Random) algorithm is based on random cracks. These algorithms perform two cracking actions: 1. center or random cracks 2. query bounds. Progressive Stochastic approach was also highlighted to be a

Volume : 4 | Issue : 7 | July 2014 | ISSN - 2249-555X

future work of this research.

2.4 Database Virtualization

Resource virtualization aims at creating a abstract layer between the actual hardware which provides resources namely RAM, power, CPU cycles etc from the logical entities which consumes the resources viz. email, web service etc.[5]. A popular example is the usage of virtual machines (VM) instead of traditional machines using individual processors. This would lead to database systems be run in virtualized machines. Though it has many advantages, it may also lead to many bottlenecks in tuning. This paper studies the different problems that may arise due to database systems running on a virtualized machine and also proposed a cost model based on workload for performance tuning of the database.

3. PROPOSED MECHANISM

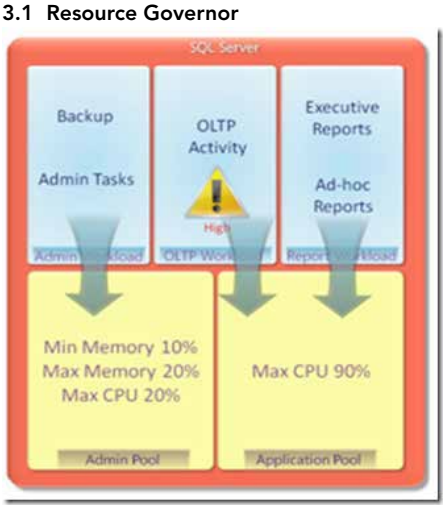
The proposed model should be able to increase the performance of the database operations without involving any major changes in the hardware configuration or the software application thereby being economical. We would like to use an efficient methodology "Log Shipping" available in SQL Server to send transactions from the primary server to any other secondary server automatically. By this overhead of a single server storage space is drastically reduced thereby increasing the overall performance of the primary server. SQL Server latest releases such as SQL Server 2012, 2014 new querying techniques can be extensively replaced instead of old queries in the existing product thereby exploiting the benefits from tuned queries. Our model would focus on conversion of such queries.

Microsoft SQL Server 2008 provides a number of enhancements and new functionality, building on previous versions.

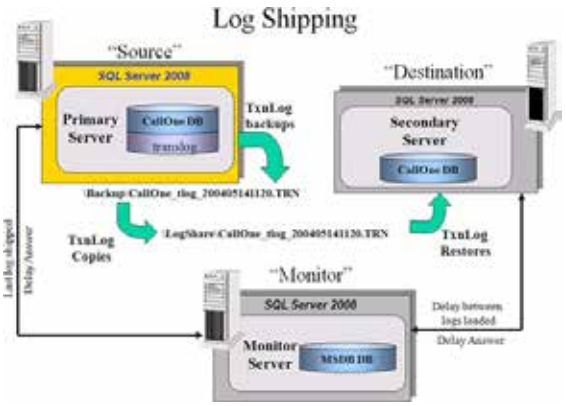
- Activity Monitor
- [SQL Server]Audit
- Backup Compression
- Central Management Servers
- Data Collector and Management Data Warehouse
- Data Compression
- Policy Based Management
- Predictable Performance and Concurrency
- Resource Governor
- Transparent Data Encryption(TDE)
- SQL Log Shipping

SQL injection is a code injection technique, used to attack data driven applications, in which malicious SQL statements are inserted into an entry field for execution. As a security measure, SQL-IA (SQL Injection Attacks) prone queries are identified and the same should be modified.

3.1 Resource Governor



3.1 SQL Log Shipping



3.2 Query Tuning

3.2.1 Conversion Guidelines with Example

3.2.1.1 Left Outer Join (*=)

Example

Original Code

```
SELECT *
FROM
fs_analysis_master mst,
common..fs_analysis_master_shd shd
WHERE
mst.company_code= 'XXXXX'
AND mst.locn_code='XXXXX'
AND mst.company_code*= shd.fs_company_code
AND mst.locn_code*= shd.fs_locn_code
AND mst.fs_analysis*=shd.fs_analysis
AND mst.sub_analysis*=shd.fs_sub_analysis
AND mst.posting_level='D'
AND mst.del_status='0'
AND shd.fs_lang_id =1
```

Converted Code

```
SELECT *
FROM
fs_analysis_master mst LEFT OUTER JOIN
common..fs_analysis_master_shd shd
ON
(shd.fs_lang_id          = 1
AND mst.company_code     = shd.fs_company_code
AND mst.locn_code        = shd.fs_locn_code
AND mst.fs_analysis      = shd.fs_analysis
AND mst.sub_analysis     = shd.fs_sub_analysis )
WHERE mst.company_code   = 'XXXXX'
AND mst.locn_code        = 'XXXXX'
AND mst.posting_level    = 'D'
AND mst.del_status       = '0'
```

3.2.1.2 Right Outer Join (=*)

Example:

Original Code

```
SELECT *
FROM common..pur_company_vendor_details mst,
apdb..fs_bank_detail bm,
common..pur_company_vendor_details_shd shd
WHERE mst.company_code = 'XXXXX'
AND mst.vendor_code = 'YYYYY'
AND shd.company_code=* mst.company_code
AND shd.vendor_code =* mst.vendor_code
AND shd.payment_serial_no=* mst.payment_serial_no
AND bm.bank_code = ISNULL(mst.vendor_bank_id,'')
AND   shd.lang_id   = 1
AND   bm.company_code = 'XXXXX'
```

Converted Code

```
SELECT *
FROM
apdb..fs_bank_detail bm,
common..pur_company_vendor_details_shd shd RIGHT
OUTER JOIN
```

Volume : 4 | Issue : 7 | July 2014 | ISSN - 2249-555X

```
common..pur_company_vendor_details mst
ON
(shd.lang_id          = 1
AND shd.company_code = mst.company_code
AND shd.vendor_code = mst.vendor_code
ANDshd.payment_serial_no= mst.payment_serial_no )
WHERE  mst.company_code  = 'XXXXX'
AND     mst.vendor_code = 'YYYYY'
AND bm.bank_code= ISNULL(mst.vendor_bank_id, '') AND
    bm.company_code = 'XXXXX'
```

3.2.1.3 Column name or number of supplied values does not match the table definition.
In an insert statement the insert column list should match with the select column list.

Example:
Original Code

```
SELECT fs_analysis,
fs_description,
NULL,
NULL,
1
FROM fs_analysis_master
WHERE company_code          = 'XXXXX'
AND locn_code       = 'XXXXX'
AND del_status      = '0'
AND level_no        = 0
```

Converted Code

```
SELECT fs_analysis,
fs_description,
NULL,
NULL,
1,
NULL,
NULL
FROM fs_analysis_master
WHERE company_code          = 'XXXXX'
AND locn_code       = 'XXXXX'
AND del_status      = '0'
AND level_no        = 0
```

3.2.1.4.Ambiguous column name
When a particular column name is selected more than once in a select statement and when you have an Order by clause for this select on this column name.

Example:
Original Code

```
SELECT company_code,location_code,
tran_type,tran_grp_tmp, tran_no_tmp,
act_inc_date, curr_code,
vch_serial_no, vch_serial_no
FROM gl_bank_postings_temp
WHERE host_id       = '1'
ORDER BY vch_serial_no
```

Converted Code

```
SELECT company_code,location_code,
tran_type,tran_grp_tmp, tran_no_tmp,
act_inc_date, curr_code,
vch_serial_no, vch_serial_no
FROM gl_bank_postings_temp
WHERE host_id       = '1'
ORDER BY 8
```

3.2.1.5 Table Alias should be used in order by clause when a statement has a UNION,INTERSECT or EXCEPT

3.2.1.6 Column name appears more than once in the result column list

3.2.1.7 Aggregate may not appear in the set list of an UPDATE statement

3.2.1.8 HOLDLOCK
The SQL statements with Table Hints like HOLDLOCK should be replaced as WITH (HOLDLOCK) or (HOLDLOCK)

3.2.1.9 Runtime Errors
These problem are encountered while the procedures are executed Ambiguous Column Name Error in #Temp table

3.2.1.10 System Table References
Many system tables that were undocumented in prior releases have changed or no longer exist; therefore, using these tables may cause errors after upgrading to SQL Server 2008.

3.2.1.11 Timestamp
In an insert statement, when trying to insert a timestamp variable to a timestamp column in a table error is encountered.

3.2.1.12 SQL2008 GRANT Permission on various objects
The ALL permission is deprecated and maintained only for compatibility. It DOES NOT imply ALL permissions defined on the entity.

3.3 Targeting SQL Injection
OWASP is an open-source web application security project. It aims at providing a document every year which would list the top 10 vulnerabilities to web applications. In 2013, Injection flaws such as SQL, OS and LDAP injection have been rated first among the vulnerabilities.[6] Sql injection is a method used to target data driven applications in which malicious SQL statements are inserted into an entry point for execution which leads to dropping of objects or retrieving sensitive information. In the proposed model, we have an additional feature to enhance the product from the topmost OWASP attacks. Every stored procedure is scanned for any dynamic execution statement in it which is vulnerable to become an entry point for sql injection.

4. EVALUATION
We compared the working of our proposed model with recent techniques proposed under database performance tuning.

4.1 NODB Approach
There are several tradeoffs in this method. Querying from raw data files will pose an additional overhead to the execution memory. Raw data costs and I/O costs rise up a tuning tool is required to balance the cost. Further raw data files are unreliable as they can be corrupted which is unknown prior to querying which would result in false or corrupt data mismatches and delay in query processing time.

4.2 ASCII Based and AMNESIA
These methods were proposed for detecting SQL Injection attacks wherein ASCII method, the corresponding ASCII code is stored for user login and password to avoid any illegal entry points for hackers. But this method uses only 1 byte per character and also limit the language to be English only. In the latter method (AMNESIA) cannot be applied to web applications that employ new query development paradigms. Moreover it is unable to detect attacks in stored procedures.

The drawbacks that were discussed in the above techniques are rectified to greater extent in the proposed model.

5. CONCLUSION
In this paper, we have implemented a model to tune a given database in SQL by performing query tuning as per the latest service pack release of SQL. Additionally, to increase the performance of the server, Log Shipping feature of SQL is been exploited. Security is provided by detecting SQLIA in the stored procedures and the database administrator is alerted for the same.

Volume : 4 | Issue : 7 | July 2014 | ISSN - 2249-555X

In future we would like to improve the model with more query tuning concepts and also to enhance security by detecting other developing vulnerabilities such as cross side scripting.

REFERENCE [1] Karl Schnaitter, Neoklis Polyzotis, "Semi-Automatic Index Tuning: Keeping DBAs in the Loop", IEEE 2011. | [2] Ioannis Alagiannis Renata Borovica, "NoDB: Efficient Query Execution on Raw Data Files", SIGMOD 2012. | [3] Felix Halim ,Stratos Idreos, "Stochastic Database Cracking: Towards Robust Adaptive Indexing in Main-Memory Column-Stores", Proceedings of the VLDB Endowment,2012 | [4] S. Idreos. Database cracking: Towards auto-tuning database kernels. CWI, PhD Thesis, 2010. | [5] https://we.riseup.net/debian/resource- | virtualization. | [6] https://www.owasp.org/index.php/Top_10_2013-Top_10. | [7] David Botzer and OpherEtzion, "Tuning of the Relationships among Rules' Components in Active Databases Systems", IEEE 2004. | [8] Gaozheng Zhang A Model for Application-Oriented Database Performance Tuning IEEE 2010. |