Corruption detection in MySQL with the help of checker and logs

Priyanka Lohani

Abstract: In the world of storage Databases are used to store data needed for computer programs. It is basically used to store very sensitive data. Software and hardware failures are main issues under database corruption. To prevent data loss and inconsistency of database after failure, recovery is needed. MySQL DBMS, it has been find that in certain cases, corruption can greatly harm the system, leading to untimely crashes, data loss, or even incorrect results. In this paper some other techniques are defined to enhance the performance of the checker for corruption detection technique with the help of logs. The checker will work according to type of the corruption. If there is any empty values in the data file then checker create a bad log file which will have only inconsistent values. Other records will be inserted in the database. If there is any data type mismatch then checker will be automatically terminated.

Index Terms— ARIES, checksum, MySQL corruption, logs, RAID, redo undo operations, recovery management**,**

Introduction:

Databases are used to store data needed for computer programs. It is basically used to store very sensitive data. To prevent data loss and inconsistency of database after failure, recovery is needed. Data corruption mainly refers to errors in the data that occur during writing, reading, storage, transmission, or processing, that will create inconsistency in the original data. When data corruption occurs, the file that contains that data may become inaccessible, and the system or the related application will give an error. Take an example, if a MS word file is corrupted, when you try to open that file with MS Word it will generate an error message, and the file would not be opened. Some programs can give a suggestion to repair the file automatically, and some programs cannot repair it. The main reasons for disk corruption are:

1. One reads the block of data from the disk and receives unexpected contents
2. The second reason is where a disk report that a write has completed but it was never written to the disk.
3. If a drive cache is set to write back mode, a write will be acknowledged when it is put into the disk cache but before it has been written to disk if power is lost before the actual write to the media surface, the write seemingly lost.

Recovery techniques

The recovery techniques used to recover database corruptions. These techniques are given below:

Log based recovery

Log is a file that particularly is used to record modification of the database. A log file has sequence of log records which contains details of each activity of the database which is required for recovery process.

Each log record has two types operation.

- UNDO- Information needed for UNDO
- REDO- Information needed for REDO. [2]

The most important characteristic of failures is the amount of information which is lost because of the failure. There are describing the major form of failures.

- First, Failures without loss of information. In This failure include the error occurred in the system because of this failure we have to abort the transaction. Example of this failure is divided by zero. In this type of failure we have stored information in memory used for the recovery process.
- Second, Failure with loss of volatile storage: in this failure all the records stored in memory is lost but we can recover this failure by using disk storage. Example of this failure is system crashes.

- *Author name Priyanka Lohani is currently pursuing master's degree program in computer science and engineering in Lovely Professional University, India, PH-9888916943-. E-mail: prnklohani@gmail.com*

- And last is Failures with loss of non-volatile storage: in this failure all the information stored the disk is lost. Example of this failure is head crash.

ARIES

ARIES stands for “Algorithm for Recovery and Isolation Exploiting Semantics” used by most of DBMS which guarantees the atomicity and durability properties of transactions. The fact is that in the case of process, transaction, system and media failures ARIES uses a single log sequence number (LSN). Every log record has its associated LSN which has all the information of the disk location or the address of the log record. The LSN contains the LSN of the log record that describes the latest update to the page. In the page where LSN field which is placed in the page itself update when the page is updated and a log record is written. This is important to track the logged updates for the page in the restart and media recovery.

Shadow paging

Shadow paging is again a recovery technique which very useful for crash recovery technique which does not use a log for recovery. Shadow paging technique always considers that the database has fixed size disk block called page. This Database can have any number of pages in which each page must have some identification in the disk.

This Page table is basically used to identify the database pages. Suppose our database consists of n pages then the page table will contain n entries in which each record point to a page on the disk. The main idea behind the technique is keep two page tables during the transaction:

- shadow page table
- current page table

When the transaction starts, these two tables are identical. During transaction execution there will no modification in the shadow page table. When any write operation occurs, it will simply create a new copy of particular page which is going to modify and will accordingly modify current page table to point new page. All these modifications are visible only for the current page. Then there are two copies of database page, a new version which is pointed by current page table entry and old version which is pointed by shadow page table.

Backup

If we talk about log based recovery techniques and shadow paging are recovery techniques used to recover from non damaging failures. Recovery manager of a DBMS is responsible to recovery from catastrophic failures such as disk crashes. Log can be used to recover from media failures if log is not on same disk and never threw away after a checkpoint. Back up is the most useful technique for recover from such failures. Generally full database archive and log periodically copied into cheap medium such as magnetic tapes or optical disk. The archive database copy and log must be stored in remote secure location.

Background & related work

In MySQL DBMS, There are certain cases which the analysis finds that corruption can greatly harm the system; it can be lead to crashes, data loss, or incorrect results. The result says that out of 145 injected faults, 110 lead to serious problems.[9] The RAID is basically designed to tolerate the loss of a certain number of disks or blocks (e.g., RAID-5 tolerates one, and RAID-6 tolerates two), but not to identify corruption. For example, in RAID-5, it is that if a block in a parity set is corrupt, then the parity computation will be incorrect, but it cannot identify corrupted block.[9] Checksum is another technique which is used in various systems to detect data corruption. When a block is read from the disk, its checksum too stored. A checksum is then computed over the data block and compared to the stored checksum; if the two of them do not match, the block is declared as a corrupt block and recovered from a mirror copy. Checksums can improve corruption detection but still it is not a complete solution for three reasons. First, memory is not perfect. For example, a bit-flip in memory before a checksum is computed could lead to a corrupt block being written to disk; the disk system will safely store the corrupted block. Second, software is not perfect; large code bases are typically full of bugs. Lastly, check summing does not protect against complex failures such as torn writes, lost writes, and misdirected writes. [9]
After the detailed observations they pointed out three deficiencies: MYSQL does not have the capability to detect some corruptions due to lack of redundant information, does not isolate corrupted data from valid data, and has inconsistent reactions to similar corruption scenarios.

Online corruptions: from the previous study, the results have shown that MySQL does not detect all kinds of corruption that can arise; the MySQL server is not highly available in the midst of corruptions, and finally MySQL does not have a consistent framework for corruption handling. Some of the details are

1) Incomplete Detection: MySQL ignores many corruptions, which leads to incorrect results being returned, crashes, and

data loss. MySQL ignores detectable corruptions and in some other cases MySQL does not have the ability to detect certain corruptions. In the previous study, B-Tree was taken as an example. In B-Tree pointer corruption, when a pointer is corrupt such that it points to a page not reachable from the parent page, MySQL could detect this by checking the keys. However, since MySQL does not perform such a check, incorrect results are returned. Another problem was a record pointer corruption which could be easily detectable; the MySQL server could compare the key stored in the index with the one stored in the record. But, rather than utilizing this redundant information, MySQL always trusts the keys stored in the records. As a result, incorrect results are returned. Third, in the index format corruption, when the data file length specified in the state header is corrupted to zero, MySQL returns no result to the user without any error message, blindly believing that the data file is empty. [9]
Undetectable corruption: Several corruptions are hard to detect because MySQL does not store enough implicit redundant information in its data structures. In the previous example, B-Tree pointer corruption it is hard to verify that a pointer properly connects two pages in adjacent levels because a page does not store its page level. For example, if a pointer is corrupt such that it points to one of its grand-children MySQL cannot detect this easily.

Offline checker: A DBMS offline checker should be the last tool that catches all corruptions in the database. Checker runs in two modes: check and repair. In this first mode, myisamchk attempts to find all corruptions in the database, while in the second, it tries to rebuild the tables and index files. [9] After the thorough study it has found that it does not catch all corruptions and it does not always repair the database correctly. Observation point to the same issues faced by the running MySQL. Mainly, some detectable corruptions are ignored and some corruptions are not detectable due to the lack of redundant information. As a result, the checker itself can crash and even worse an erroneous repair could happen.

Methodology

After the detailed observations three deficiencies were found in MYSQL:

1. MYSQL does not have the capability to detect some corruptions due to lack of redundant information
2. It does not isolate corrupted data from valid data
3. It has inconsistent reactions to similar corruption scenarios.

To improve the MySQL checker performance I have used the concept of logs. The following approach is used here to improve the performance of MYSQL checker. The steps are as follows:

1. In the very first step analysis is done and checks what kind of corruption can be occurred in the system.
2. The corruption can be

- One reads the block of data from the disk and receives unexpected contents
- The second reason is where a disk report that a write has completed but it was never written to the disk.

3. To improve the data consistency and to improve the checker performance here is creation of a checker in java and with the help of log which can easily check the inconsistency in data.
4. To deal with these corruption we are going to checker which will detect the error in data and will generate a log file (excel file).
5. This log file has different-different information of according to error.

This was the whole information the MySQL checker.

Now next step to describe how it will work

1. The creation of checker using java Net beans. In this the data which will be inserted on MYSQL termed as backup file.
2. Using this backup file the checker will check whether the data is valid data or not.
3. To check whether the insert data is valid or not we will apply some techniques which are as follows:
    - If the inserted data in the data base is having any empty value, then it will generate a log file i.e. excel file which will have all the empty values.
    - To resolve this problem, the excel file will be checked and then some modification will be performed on the data and then finally apply the redo function. Redo function will automatically perform the same operation of insertion.

- An undo function can be performed if we want to delete or drop the last performed action.

4. To check whether the inserted data is according to the given length or not.

   To do so again will follow some steps

   - When insertion of data takes place, some problem arises when the inserted data is not of given length. The will show that the data write is committed but in actual it was not.
   - So to resolve this problem the checker will add these errors to the same log file which was created earlier.
   - And then accordingly perform redo and undo operation.

5. The last problem which checker can detect is whether the inserted data is having same data type or not.

   - To resolve this problem checker will automatically terminate the process and will show the error message.
   - To remove or to re perform the above task again redo and undo operations can be used.

Observations

In summary, our results show that the in the previous work checker was far from robustness. It did not catch all corruptions and even it did not always repair the database correctly. Our newly created checker has higher performance than the previous one. It can detect inconsistency in data faced by the running MySQL. Mainly, it works upon the concept of log. Every time a file inserted in the database it will generate a bad log record. Data of the bad log file can be modified can be further inserted in to the database. Many important checks that last MySQL checker was unable to detect can be detected by this checker. Thus, this improvement in building of checker makes it more robust MySQL checker.

Conclusion

The importance of logging technique is to recover the data from the any failure. The most important characteristic of failures is the amount of information which is lost because of the failure. There are describing the major form of failures. First is failure without loss of information. This failure include the error occurred in the system because of this failure, transaction have to aborted. Example of this failure is divided by zero. In this type of failure we have stored information in memory used for the recovery process. Second, Failure with loss of volatile storage: in this failure all the records stored in memory is lost but we can recover this failure by using disk storage. Example of this failure is system crashes. And last is Failures with loss of non-volatile storage: in this failure all the information stored the disk is lost. Example of this failure is head crash. [2]

When bulk of the inserted in the data base, there may be number of inconsistency occurs in the data file. Without knowing these inconsistencies the data is directly inserted into database. These inconsistencies in the database will generate error and failure of the database. Here the enhancement of the checker is represented which is if any invalid data enter in the MYSQL the will checker the subsequent action. If there any null value occurs in the database, checker automatically generates a log file in which all the empty values will be written. If any data type mismatch occurs it will automatically terminate the process. So like this checker can perform some operation to prevent the database from the corruption. By using this we can enhance the performance of the checker as well as MYSQL by removing inconsistencies.

Acknowledgement

It is a pleasure of mine to find myself penning down these lines to express my sincere thanks to all my coordinators to give me this opportunity of preparing this paper, to enhance my professional as well as my technical practice. My deepest gratitude to all my teachers for always boosting my moral and providing me encouraging environment. In the last, I would like to thank my parents, without whom nothing was possible.

References

1. G. Somasundaram Alok Shrivastava EMC Education Services Information Storage and Management, Storing, Managing, and Protecting Digital Information Edited

2. Stefano Ceri and Giuseppe Pelagatti, Distributed databases principles and systems

3. Ding Yuan, Soyeon Park, and Yuanyuan Zhou," Characterizing Logging Practices in Open-Source ", 2012 IEEE, ICSE 2012, Zurich, Switzerland.

4. Istemi Ekin Akku, Ashvin Goel ," Data Recovery for Web Applications" 20l0 IEEE/IFIP International Conference on Dependable Systems & Networks (DSN)

5. Mi-Seon Choi, Hye-Sook Yoon, Eun-Mi Song, Young-Keol Kim, Young-Kuk Kim, Seong-il Jin, Mi-kyong Han, Wan Choi "Two-step Backup Mechanism for Real-Time Main Memory Database Recovery" 2000 IEEE

6. Nedyalko Borisov, Shivnath Babu, Nagapramod Mandagere, Sandeep Uttamchandani, "Dealing Proactively with Data Corruption" ICDE Workshops 2011

7. Oracle maximum availability architecture white paper [May 2012]

8. Philip Bohannon, Rajeev Rastogi, S. Seshadri, Avi Silberschatz, "Detection and Recovery Techniques for Database Corruption", IEEE transactions on knowledge and data engineering, vol. 15, no. 5, September/October 2003

9. Sriram Subramanian, Yupu zhang, Rajiv Vaidyanathan, Haryadi s. gunawi, Andrea c. arpaci-dusseau, Remzi h. arpaci-dusseau, Jeffrey f. naughton , "Impact of Disk Corruption on Open-Source DBMS "ICDE Conference 2010

10. Tang Yanjun , "A Model of Crash Recovery in Main Memory Database" International Conference On Computer Design And Applications 2010

11. Xiu- ju LIU ,A Brief Analysis of the Disaster Recovery Backup Technology in Oracle Database Data Guard, 2nd International Conference on Industrial and Information Systems 2010

12. http://en.wikipedia.org/wiki/Data_corruption

13. http://www.quickanswer.com/corrupt.html

14. http://www.thexlab.com/faqs/datacorruption.html