GUJARAT TECHNOLOGICAL UNIVERSITY
(GTU)
INNOVATION COUNCIL (GIC)
Patent Search & Analysis Report
(PSAR)

Date of Submission : 18/09/2016

Dear Pethani Akshay Dilipbhai,

Studied Patent Number for generation of PSAR : 16BE7_130030116040_2

PART 1: PATENT SEARCH DATABASE USED

1. Patent Search Database used : Google Patents

Web link of database : https://patents.google.com/

2. Keywords Used for Search : Distributed,Database,Data

3. Search String Used : Managing data in distributed database management system

4. Number of Results/Hits getting : 9999

PART 2: BASIC DATA OF PATENTED INVENTION /BIBLIOGRAPHIC DATA

5. Category/ Field of Invention : Computer/IT Engineering

6. Invention is Related to/Class of Invention : DBMS

6 (a) : IPC class of the studied patent : G06F9/466 Transaction processing

7. Title of Invention : Managing access to data in a distributed database environment

8. Patent No. : US6237001B1

9. Application Number : US09095463

9 (a) : Web link of the studied patent : https://patents.google.com/patent/US6237001B1

10. Date of Filing/Application (DD/MM/YYYY) : 10/06/1998

11. Priority Date (DD/MM/YYYY) : 23/04/1997

12. Publication/Journal Number :

13. Publication Date (DD/MM/YYYY) : 22/05/2001

14. First Filled Country : Albania :

15. Also Published as

Sr.No	Country Where Filled	Application No./Patent No.
1		

16. Inventor/s Details.

Sr.No	Name of Inventor	Address/City/Country of Inventor
1	Roger J Bamford	USA (MIT)
2	Boris Klots	USA

17. Applicant/Assignee Details.

Sr.No	Name of Applicant/Assignee	Address/City/Country of Applicant
1	ORACLE CORPORATION	CALIFORNIA

18. Applicant for Patent is : Company

PART 3: TECHNICAL PART OF PATENTED INVENTION

19. Limitation of Prior Technology / Art

In typical database systems, users store, update and retrieve information by submitting commands or statements to a database application. To be correctly processed, the statements must comply with the database language that is supported by the database application. One popular database language is known as Structured Query Language (SQL).

A logical unit of work that is comprised of one or more database language statements is referred to as a transaction. When multiple users share a database system, it is impractical to allow only one transaction to execute at a time. However, when many transactions are allowed to execute at the same time, the issue of consistency arises. Transactional database management systems provide a special statement—“commit”—which a user or program submits to delineate the statements that make up a transaction.

To ensure predictable and reproducible results, techniques have been developed which protect concurrently executing transactions from interacting with one another in uncontrolled ways. Transactions are designed to make database changes in such a way that the database, as a whole, moves from one consistent state to another.

Not only must transactions leave the database in a consistent state upon completion, but transactions must also see the database in a consistent state when they begin. This condition is difficult to satisfy because there may be concurrently executing transactions, each of which may be causing a temporary inconsistency visible only within that single transaction. Consequently, transactions must not be able to see the changes made by concurrently executing transactions until those transactions commit (i.e. terminate and make their changes permanent).

One approach to ensure that transactions do not see changes made by concurrently executing transactions is to prevent transactions from reading a data item that has been updated until the transaction that updated the data item commits. However, this approach reduces the concurrency in the system by causing transactions that wish to read data items (readers) to wait for transactions that have written to data items (writers) to commit.

An alternate approach to ensure that transactions see a consistent view of the database is to provide a mechanism that allows a reader transaction to see a version of the data item that does not include the updates made by any uncommitted transaction. However, in systems that provide a less-than-newest version of a data item, it is not always apparent which versions of a data item can and cannot be supplied to a transaction. Further, among those versions of a data item that can be supplied to a transaction, some versions will consume more overhead to supply than other versions.

Based on the foregoing, it is desirable to provide a mechanism for determining which versions of a data item may be supplied to a transaction. It is further desirable to provide a mechanism that selects the versions of data items to supply to transactions in a way that reduces the overhead required to supply the version of the data item. Specifically, it is desirable to select versions that require less overhead to construct, while still ensuring that the versions seen are recent enough.

20. Specific Problem Solved / Objective of Invention

Access to data in a distributed database environment

21. Brief about Invention

According to one aspect of the invention, a method for is provided for selecting a version of a data item to be provided to a transaction executing on a distributed database system. The method includes the steps of determining version criteria based on which location of a plurality of locations in the distributed database system is associated with the data item and which location of the plurality of locations in the distributed database system is the location at which the transaction is executing. A version of the data item to be provided to the transaction is selected based upon the version criteria.

According to another aspect of the invention, the method includes the step of generating a MUST SEE time and a CANNOT SEE time for each of the plurality of locations. In this situation, the step of selecting, based upon the version criteria, a version of the data item to be provided to the transaction, includes the step of selecting, based upon a particular MUST SEE time and a particular CANNOT SEE time for the location associated with the data item, a version of the data item to be provided to the transaction.

According to yet another aspect of the invention, the step of determining version criteria includes the step of determining a base snapshot time for the location where the transaction is executing and a delta snapshot time for the location associated with the data item. In this situation, the step of selecting, based upon the version criteria, a version of the data item to be provided to the transaction includes the step of selecting, based upon the base snapshot time and the delta snapshot time, a version of the data item to be provided to the transaction.

22. Key learning Points

1. Accessing stored data in a distributed database environment.
2. A method and apparatus for determining which version of a data item to provide to a transaction that requests the data item.

23. Summary of Invention

According to one aspect of the invention, a method for is provided for selecting a version of a data item to be provided to a transaction executing on a distributed database system. The method includes the steps of determining version criteria based on which location of a plurality of locations in the distributed database system is associated with the data item and which location of the plurality of locations in the distributed database system is the location at which the transaction is executing. A version of the data item to be provided to the transaction is selected based upon the version criteria.

According to another aspect of the invention, the method includes the step of generating a MUST SEE time and a CANNOT SEE time for each of the plurality of locations. In this situation, the step of selecting, based upon the version criteria, a version of the data item to be provided to the transaction, includes the step of selecting, based upon a particular MUST SEE time and a particular CANNOT SEE time for the location associated with the data item, a version of the data item to be provided to the transaction.

According to yet another aspect of the invention, the step of determining version criteria includes the step of determining a base snapshot time for the location where the transaction is executing and a delta snapshot time for the location associated with the data item. In this situation, the step of selecting, based upon the version criteria, a version of the data item to be provided to the transaction includes the step of selecting, based upon the base snapshot time and the delta snapshot time, a version of the data item to be provided to the transaction.

24. Number of Claims : 21

25. Patent Status : Granted Patent & In-force Patent

26. How much this invention is related with your IDP/UDP?

< 70 %

27. Do you have any idea to do anything around the said invention to improve it? (Give short note in not more than 500 words)

This is patent is not for particular feature related to our project but we will use methods suggested by inventors to manage our database.