REPLICATION OF DATA IN DATABASE SYSTEMS FOR BACKUP ANDFAILOVER

P.Swathi
Lecturer in Computer Science, TSWRDC(W), Nalgonda,India.
Email: punnaswathi89@gmail.com
V.Srujanamma
Lecturer in Computer Science, TSWRDC(W), Nalgonda,India.
Email: vsrujana19@gmail.com

Received: Oct. 2018 Accepted: Nov. 2018 Published: Dec. 2018

Abstract: The production database are transactions intensive transactions can be insert, update on the tables, failover is the replica of the production server, if there is any change we have to implement on the production and it will be automatically implemented on failover or standby database. Now a days the data on the production server is increasing and we need extra storage space on production server to keep data and this is same required on the failover. To generate reports from that data will increase load on the production server and can affect the performance of the server. There are also some threats which can cause loss of data from which we have to protect our database like Hardware failure, loss of machine. Replication is one of the methods for Backup of the running database and its immediate failover during failure. This paper tries to identify some parts for the replication techniques, failover, and transaction states.

Keywords: Asynchronous, Active, Commit, RAID, Rollback, Synchronous.

Introduction: Demand for high performance combined with plummeting hardware prices have led to the widespread emergence of large computing clusters . Database systems are a key component of the computer infrastructure of most organizations. It is thus crucial to ensure that database systems work correctly and continuously even in the presence of a variety of unexpected events. The key to ensuring high availability of database systems is to use replication . In this paper, we propose an approach for Replication of data for Backup and failover. Traditional database replication techniques as provided by both commercial and open source database management system (such as Oracle, IBM DB2, Microsoft SQL Server, Postgre SQL, or MySQL) can be applied .Improved performance came from doing the database processing at one site only, and applying efficient primary-keyed write operations at other sites, with a certification to prevent conflicts, that was done consistently at all sites. Using sites that provide SI as concurrency control, and giving rather than serializability, allows a great improvement in performance .

Data Replication:

A. Synchronous Replication: When synchronous replication is applied, a change madeto a data at the primary site is synchronously replicated to a data volume at a secondary site. This ensures that thesecondary site has an identical copy of the data at all times. Write I/O operation acknowledgement is sent to the application only after the write I/O operation is acknowledged by the storage subsystem at both the primary and the secondary site. Before responding to the application, the storage subsystem must wait for the secondary

subsystem I/O process to complete, resulting in an increased response time to the application. Thus, performance with synchronous replication is highly impacted by factors such as link latency and link bandwidth. Deployment is only practical when thesecondary site is located close to the primary site. When evaluating the use of synchronous replication, an important consideration is the behavior of the storage subsystem when the connection between the primary and secondary subsystem is temporarily disrupted. Synchronous replication does not provide protection against data corruption and loss of data due to human errors . Snapshot technology must be used with synchronous replication to provide full protection against both losses of access to data and loss of data due to data corruption .

B. ASynchronous Replication: In an asynchronous mode of operation, I/O operations are written to the primary storage system and then sent to one or more remote storage systems at some later point in time. Due to the time lag, data on the remote systems is not always an exact mirror of the data at the primary site. This mode is ideal for disk-to-disk backup or taking a snapshot of data for offline processes, such as testing or business planning. The time lag enables data to be replicated over lower-bandwidth networks, but it does not provide the same level of protections synchronous replication. Asynchronous replication is less sensitive to distance and link transmission speeds .

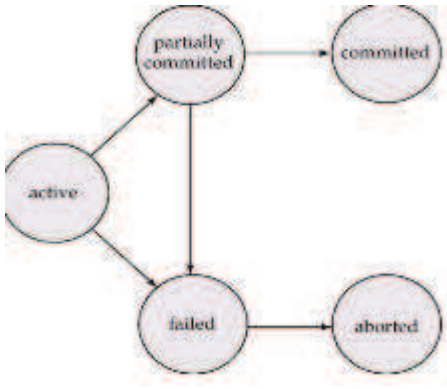
Advantages:

- SAN network-based replication is storage agnostic and server-architecture agnostic.
- SAN network-based replication can be easily integrated into and managed in an application/database environment.
- Better performance, by gathering and propagating multiple updates together, and localizing the execution at a single, possibly nearby, server.
- Better support for fault tolerance, by simplifying server recovery.
- Lower deadlock rate, by eliminating distributed deadlocks.

Disadvantages:

- Replication is limited to a single pair of servers or, at best, several servers running the same version of operating system.
- SAN network-based replication adds latency to the primary data path.
- The SAN infrastructure does not act as a transparent layer between the server and the primary storage, which can make diagnosing faults more difficult.
- Synchronous replication is that a write I/O operation acknowledgement is sent to the application only after the write I/O operation is acknowledged by the storage subsystem at both the primary and the secondary site.
- The main drawback of the deferred update technique is that the lack of synchronization during transaction execution may lead to large transaction.

Transaction States: During its processing, a transaction passes through some well-defined states. The transaction starts in the executing state, when it's read and writes operations are locally executed at the database site where it was initiated. When the client that initiates the transaction requests the commit, the transaction passes to the committing state and is sent to the other database sites. A transaction received by a database site is also in the committing state, and it remains in the committing state until its fate is known by the database (i.e.. commit or abort). Different states are shown as following Fig.



A .Active State: Active State is divided into two phases.

1. Initial Phase: A database transaction is in this phase while its statements start to be executed.
2. Committed Phase: A database transaction enters this phase when its final statement has been executed.

B .Failed State: A database transaction enters the failed state when its normal execution can no longer proceed due to hardware or program errors.

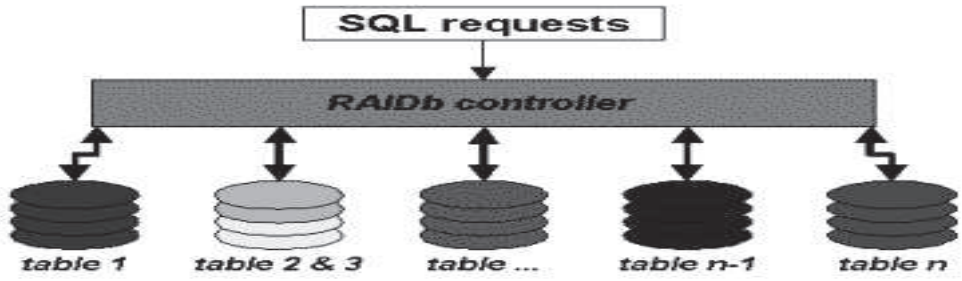
C. Aborted State: A database transaction, if determined by the DBMS to have failed, enters the aborted state. An aborted transaction must have no effect on the database, and thus any changes it made to the database have to be undone, or in technical terms, rolled back.

D. Committed State: A database transaction enters the committed state when enough information has been written to disk after completing its execution with success .

Failover: Failover is one basic fault-tolerance function of mission critical systems that are providing a constantly accessible service . Its purpose is to redirect requests from the failing system to a backup that mimics the operations of the first one .The whole process is supposed to happen automatically and transparently to the end user. Failover or switchover solutions are widely used whenever and wherever high availability is needed. Adopting such an approach is neither new or original , the real challenges of this work are its wide scope, the number of teams involved, and the geographically distributed nature of both the Grid and the related operational tools. The failover concern played a great role in the way this architecture has been designed. The clear separation between the different modules implies indeed an easier replication work, as well as many possibilities of failover scenarios. These scenarios include partial switches, each of these modules being able to work with any of the replicas of the other modules.

Redundant Array of Inexpensive Databases: One of the goals of RAIDb is to hide the distribution complexity and provide the database clients with the view of a single database like in a centralized architecture .RAIDb controllers can offer caching to hold the replies to SQL queries. The controller is responsible for the granularity and the coherence of the cache. There is no restriction to the set of services implemented in the RAIDb controller. Monitoring, debugging, logging or security management services can prove to be useful for certain users. Three basic RAID levels varying the degree of partitioning and replication among the

databases. RAIDb-0(database partitioning) and RAIDb-1 (database mirroring) are similar to RAID-0 (disk striping) and RAID-1 (disk mirroring), respectively.



Conclusion: Data can be replicated automatically and precisely to many locations. However replication works as a defense if we use logical replication over distinct database systems. The terms backup and replication are often used interchangeably. Many businesses rely on Microsoft SQL Server and if the database is not available productivity and revenue are impacted or worse, a company's hard-earned reputation and brand. Virtual Replication leverages continuous, block-level replication that has been purpose-built to maintain the data integrity of rapidly changing databases while minimizing data loss. Now a days the data on the production server is increasing and we need extra storage space on production server to keep data and this is same required on the failover.

References:

1. L. Camargos, F. Pedone, and R. Schmidt, "A primary-backup protocolfor in-memory database replication," in *Proc. Fifth IEEE InternationalSymposium on Network Computing and Applications*, pp. 204-211,July 24-26, 2006.
2. R. Garcia, R. Rodrigues, and N. Preguiça, "Efficient middleware forbyzantine fault tolerant database replication," in *EuroSys '11 Proc. Thesixth Conference on Computer systems,* pp. 107-122, ACM New York,NY, USA, 2011
3. J. Osrael, L. Froihofer, M. Weghofer, and K. M. Goeschka,"Axis2-based replication middleware for web services," in *Proc. IEEEInternational Conference* on *Web Services, ICWS,* pp. 591-598, July9-13, 2007.
4. H. Jungy, H. Han, A. Fekete, and U. Rohm, "Serializable snapshotisolation for replicated databases in high-update scenarios," presentedat the VLDB Endowment, Seattle, Washington, August, 2011.
5. P. Brouwer, "The art of data replication," An Oracle Technical WhitePaper, Oracle Corporation World Headquarters 500 Oracle ParkwayRedwood Shores, CA 94065 U.S.A, pp. 1-28, September 2011.
6. F. Pedone, R. Guerraoui, and A. Schiper, "The database state machineapproach," in *Manufactured in The Netherlands Distributed andParallel Databases*, Kluwer Academic Publishers, pp. 71–98, 2003.
7. Transaction states. [Online]. Available:http://www.jkinfoline.com/transaction-states.html
8. Database transaction. [Online]. Available:http://it.toolbox.com/wiki/index.php/Database_Transaction
9. A. Cavalli, A. Pagano, O. Aidel, C. L'orphelin, G. Mathieu, and R.Lichwala, "Geographical failover for the EGEE-WLCG gridcollaboration tools," in *Proc. International Conference on Computingin High Energy and Nuclear Physics*, vol. 119, 2008.

10. J. P. McDermott, "Replication does survive information warfareattacks," in *Proc. the IFIP TC11 WG11.3 Eleventh InternationalConference on Database Security XI 1998*, pp. 219-228, London, UK, UK, 1998.
11. E. Cecchet, J. Marguerite, and W. Zwaenepoel, "RAIDb: Redundantarray of inexpensive databases," Institut National De RechercheEnInformatique Et EnAutomatique, September, 2003.
12. RAIDb basics. [Online]. Available: http://cjdbc.ow2.org/current/doc/userGuide/html/ar01s10.html

***