Chapter 14
Fault Tolerant Data Management for Cloud Services

Wenbing Zhao
Cleveland State University, USA

ABSTRACT

Database systems are the cornerstones of today's information systems. The availability of database systems largely determines the quality of service provided by the information systems. In this chapter, the authors provide a brief overview of the state-of-the-art database replication and clustering techniques. For many, a low-cost shared-nothing database cluster that uses conventional hardware might be a good starting point towards high availability. The authors envisage that future generation of database management systems will be intrusion tolerant (i.e., they are capable of continuous operation against not only hardware and process crash fault but a variety of security threats as well).

Keywords: Database Cluster, Database Recovery, Database Replication, High Availability, NoSQL database, Transaction, Two-Phase Commit Protocol

INTRODUCTION

The pervasiveness of cloud services has significantly increased the dependability requirement of cloud systems. Most cloud services are implemented according to a three-tier architecture where the presentation, application logic execution, and data management are separately handled by each tier (Zhao, Moser, and Melliar-Smith, 2005). The middle-tier servers implement the application logic and they are designed to be either stateless or to only maintain session state. Hence, this separation of concerns has greatly increased the scalability of such systems because the middle-tier servers can be easily scaled out. On the other hand, this design makes the data management tier ever more important because the

DOI: 10.4018/978-1-5225-7598-6.ch014

availability and integrity of the services hinges on the data management tier. The data must be made highly available and protected against various hardware faults and malicious attacks.

While it is relatively straightforward to ensure high availability for Web servers and application servers by simply running multiple copies according to the three-tier architecture, it is not so for a database management system, which has abundant state. The subject of highly available database systems has been studied for more than two decades and there exist many alternative solutions (Agrawal et al., 1997; Cecchet, Candea, & Ailamaki 2008; Drake et al., 2005; Garcia, Rodrigues, & Preguiça, 2011; Kemme, & Alonso, 2000; Patino-Martinez et al., 2005). In this article, we provide an overview of two most popular database high availability strategies, namely database replication and database clustering. The emphasis is given to those that have been adopted and implemented by major database management systems (Banker 2011; Davies & Fisk, 2006; Vallath 2004).

BACKGROUND

A database management system consists of a set of data and a number of processes that manage the data. These processes are often collectively referred to as database servers. The core programming model used in database management systems is called transaction processing. In this programming model, a group of read and write operations on the some data set are demarcated within a transaction. A transaction has the following ACID properties (Gray & Reuter, 1993):

- **Atomicity:** All operations on the data set agree on the same outcome. Either all the operations succeed (the transaction commits), or none of them are (the transaction aborts).
- **Consistency:** If the database is consistent at the beginning of a transaction, then the database remains consistent after the transaction commits.
- **Isolation:** A transaction does not read or overwrite a data item that has been accessed by another concurrent transaction.
- **Durability:** The update to the data set becomes permanent once the transaction is committed.

An example of an atomic transaction is shown in Figure 1. This transaction involves a debit operation on a savings account and a credit operation on a checking account. If both operations are successful

Figure 1. An example of atomic transactions