Chapter 5
Highly Available Fault-Tolerant Cloud Database Services

Chetan Jaiswal
University of Missouri Kansas City, USA

Vijay Kumar
University of Missouri Kansas City, USA

ABSTRACT

Legacy database systems manage transactions under a concurrency control and a recovery protocol. The underlying operating system creates transaction execution platform and the database executes transactions concurrently. When the database system fails then the recovery manager applies “Undo” and/or “Redo” operations (depending upon the recovery protocol) to achieve the consistent state of the database. The recovery manager performs these set of operations as required by transaction execution platform. The availability of “Virtual” machines on cloud has given us an architecture that makes it possible to eliminate the effect of system or transaction failure by always taking the database to the next consistent state. We present a novel scheme of eliminating the effect of such failure by applying transaction “roll-forward” which resumes its execution from the point of failure. We refer to our system as AAP (Always Ahead Processing). Our work enables cloud providers to offer transactional HA-DBMS as an option that too with multiple data sources not necessarily relational.

We are what our thoughts have made us; so take care about what you think. Words are secondary. Thoughts live; they travel far. - Swami Vivekananda

1. INTRODUCTION

Cloud computing provides a global data management platform that is capable of satisfying data processing needs of all types of customers (individuals, organizations, etc.) It has changed the way data are managed (processing and data storage). The cloud exploits the power of virtualization in managing all its activities including transaction management. It has made the conventional way of dealing with

DOI: 10.4018/978-1-5225-0153-4.ch005
information nearly obsolete by offering a more user-friendly and need-matching computing platform to its users. Although users can have their desired computing platform, many issues still need innovative solutions to make the platform universally acceptable. One such issue is to provide high availability (HA) of a processing entity to provide database Fault Tolerance (FT) management services. These properties still remain complex and expensive to achieve. HA makes sure that irrespective of failures, the system is available whenever a user needs it. If the system is available 90% of the time then it is called as “One Nine” availability, if it is available 95% of the time then it is called as “Two Nines” and if it is available 99.999% of the times then it is called “Five Nines”. FT systems are those systems that continue to work, possibly at a reduced level, rather than entirely failing (High Availability, n.d.) (Fault Tolerance, n.d.).

HA with FT can be achieved through hardware (replicated processors) or through software (advanced database architecture and commodity server and storage technology). Our goals are to make transaction management (processing and recovery) on the cloud cost-effective and to achieve HA of “Five Nines” or more. We achieve this by eliminating the effect of transaction and system failures on transaction processing. A natural inclination to do so is to alter or to extend the conventional data processing approaches for the cloud. This, however, would not work satisfactorily because of the following reasons.

A. It is not likely to utilize cloud resources efficiently and datacenters may become a bottleneck because they are geographically dispersed. A conventional approach assumes that databases are in the vicinity of the processing units.

B. Cloud virtualization can be exploited to achieve HA in a much better way than it is done in (Minhas, et al., 2013).

C. A conventional DBMS calls for a relational system at the core. The current trends is to use an RDF (Resource Description Framework), KV (Key-Value) or some other data format for information processing, but it creates a major challenge for the deployment of TM-DM (Transaction Manager-Data Manager) with ACID semantics on the cloud. We refer to a TM-DM model as a conventional RDBMS solution where components such as data management, concurrency control, and recovery management are bundled together. For further information refer to (Bernstein, Hadzilacos, & Goodman, 1987).

D. FT and HA are very expensive on conventional TM-DM systems because they need two completely replicated physical machines running in a lockstep fashion.

2. OUR IDEA

In conventional database systems, some serialization technique is used to execute transactions and “undo” or “redo” (or a combination of them is used to recover from) transaction or system failures. A checkpointing operation is used to reduce the cost of recovery (Bernstein, Hadzilacos, & Goodman, 1987). Undo and redo operations are expensive and do not help much in achieving our measure of HA and FT. Our idea is to eliminate the effect of failure from database processing. We introduce the idea of “roll-forward” to achieve our objective. A roll-forward of a transaction always takes its execution forward and eventually commits. Note that a roll-forward operation is different from a redo. In this chapter we refer to our system as Always Ahead Processing (AAP).

Undo and redo are executed by the database system in order to reach a consistent state. An undo operation is required in a situation when an uncommitted transaction is allowed to write on the database