International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 07 Issue: 06 | June 2020 www.irjet.net p-ISSN: 2395-0072

An Approach to Implementing HADR in Industrial Projects

Lav Ram Gabri[1], BK Srinivas[2]

[1]*U.G. Student, Dept. of Information Science and Engineering, R.V. College of Engineering, Bengaluru, Karnataka, India*
[2]*Assistant Professor, Dept. of Information Science and Engineering, R.V. College of Engineering, Bengaluru, Karnataka, India*

Abstract - *The monolithic database/systems are almost extinct due to the fact that there exist thousands of VMs, databases, machines etc. in one single multi cloud environment. This multi cloud environment form systems with distributed system architecture. These exist for the reason to reduce dependencies on other parts of the systems and make systems more independent. High availability, disaster recovery, in cyber resiliency protect people from eventual events/shutdowns that do not favour productivity, in fact, the focused factors provide a sense of safety and recoverability for the sensitive data that the consumer provides and requires for transactions. This paper aims to provide different techniques in which high availability and disaster recovery can be implemented for various scenarios in industrial projects.*

Key Words: High availability, disaster recovery, service-based clusters, self-managed clusters, Informix DB2, primary region, secondary region, fault tolerance.

1. INTRODUCTION

Data needs to be resilient, which means it must be able to withstand in case any loss occurs. Data needs to be stored at multiple locations so that it easy for accessing the data in case there is an unforeseen circumstance such as an outlet shutdown, natural calamity, etc. This data must be stored in such a way that one unit of data loss should not cause any information loss. The data needs to keep consistent across all the systems. Data integrity is equally important to make the data maintainable so as to keep the data updated and maintain it across all systems.

According to TechTarget, "High availability refers to a system or component that is continuously operational for a desirably long length of time." For storage, this could mean something as simple as configuring several disks as a RAID or it could mean multiple redundant storage systems and servers, configured to deliver consistent and ongoing uptime.

Should a system actually fail, it is often important for an organization to quickly recover from the event—and this is where the concept of disaster recovery comes into play.

TechTarget defines disaster recovery as a plan which "allows an organization to maintain or quickly resume mission-critical functions following a disaster." To be able to recover from a disastrous event, IT organizations require features that enable data backup or automate the restoring of an environment, while incurring minimal downtime. This allows organizations to maintain the necessary levels of productivity.

High availability and disaster recovery need not necessarily be mutually exclusive.

They are both important in delivering constant levels of business productivity. When both concepts are applied in concert, they can help organizations achieve extremely high levels of fault tolerance. E.g. When a system is disrupted, and all of the communication can be instantly redirected to another available node within the solution. When the disruption is resolved, the system can benefit from automatic failover to the primary node in the solution. This results in maximum availability for business-critical solutions, and more importantly, no disruption to end users.

This paper aims to provide overview into incorporating high availability and disaster recovery (HADR) in industrial projects. It is structured to introduce the concept of HADR in section 1. The requirement for the concept is described in Section 2. Section 3 describes different types of HADR environments. Section 4 describes the approach used in this paper to implement HADR. Section 5 highlights some important points to be kept in mind during the process. Section 6 closes the paper with conclusions derived from the purpose of the paper.

2. SCENARIO REQUIREMENT FOR HADR

The requirement for HADR arises from undesired consequences which may be due to the fact that some systems aren't ready to face out-of-course turbulence. These problems can mainly be classified into:

a. *Natural Calamity:* Earthquakes, lightning strikes etc. might disrupt modules of a system resulting in loss of data, outlet shutdowns etc.

b. *Human Error:* Humans make mistakes, which are common and can cause loss of revenue and important information. This data however can be restored if adequate measures in HADR are taken to avoid errors.

c. *Man-Made Disasters:* Data may be lost due to destruction of assets such as datacenters, systems, industrial hardware etc. These may include disasters like terrorist attacks, feuds etc.

d. *Scaling: Industries may scale their production and may transfer or process data which may cause data to transform.*

3. HADR IN DIFFERENT TYPES OF CLOUDS

HADR can be implemented in cloud computing architectures under different types of categories, namely:

3.1 Based on cloud location

Public clouds are centrally regulated by firms or third-party providers that usually works on subscription basis. It follows a pay-as-you-go model that these clouds form a local instance of the cloud on the machine and a global instance where the permanent data of the customer is kept that changes seldom. The local instances of data on the machines are to be used often or that have been prefetched from the central server which update frequently in terms of HADR.

Private clouds are owned by an individual or a single organization in which control over cloud environment is much higher. The security is much higher than in public clouds. The cost of HADR is added up with cost of maintenance that has to be borne by the organization. In terms of HADR, infrastructure is limited and has to be set up by the owner itself allowing higher security and maintainability, given higher costs.

Hybrid clouds provide the best of both private and public clouds, based on purpose served with lower costs and more flexibility. In terms of HADR, infrastructure and services determine the quality of HADR incorporated. Due to constrained environment of both clouds, cloud bursting may result in information loss.

3.2 Based on cluster management in cloud environment

Cloud environment allows services to take control over management of clusters of systems, they can be categorized into –

i. Managed service-based clusters
ii. Standalone self-managed cluster

Managed service-based clusters are professional upgrade suites and maintenance service clusters provided to customers. The ownership lies with an organization. They have certain restrictions and costs are developed over them. It provides additional software, machines/VM etc. to the existent solution called nodes. and is scalable in terms of HADR. Thus, it is very robust with respect to HADR, where the only drawback lies with the dependency of the clusters to the service because of the contractual plans.

A standalone self-managed cluster is a network-connected set of virtual or physical machines into microservices are deployed and managed. A part of the cluster called the cluster node is actually a machine/VM. Adding newer nodes makes the standalone self-managed cluster rebalances the service partition replicas and instances across all the nodes. This results in improved performance but lesser contention to access memory. It is not very powerful compared to the earlier, but it serves as an independent, low cost alternative that provides HADR to industries working on small scale.

4. PROCEDURE TO INCORPORATE HADR

Let us have a primary region on which the actual database and other infrastructure (X) exist and a secondary region (standby region) to which it needs to be backed up in case of any damage or loss to the primary. Let us call the replica of X as X'. X' is the standby replica of the database and other infrastructure. HADR can be idealized as higher levels of data segregation and replication. For the demonstration we have used Informix DB2 by IBM. DB2 is one of the most efficient and scalable databases that provides tons of operations on it.

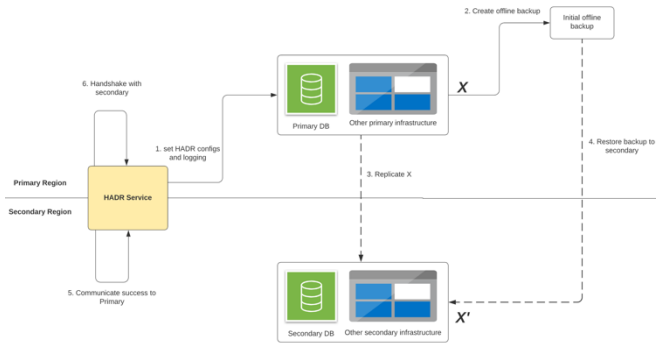

Fig -1: Architecture Diagram of a simple HADR environment for this approach

4.1 IN PRIMARY REGION

a) The service responsible for the course of execution must be enabled with logging and log archiving.
b) The configurations of HADR on primary database must be set up. E.g. service name, port, IP address etc.
c) An offline backup must be taken of the primary database. An online backup can also be taken but it is inefficient as it can take longer to bring the HADR pair (primary and secondary) into peer state.

4.2 IN SECONDARY REGION

a) Using FTP, copy the backup image from primary region to secondary.
b) Restore the second primary backup on secondary.
c) Set up HADR configs as per machine on secondary region. Prefer similar machines for minimal complexity.
d) Start the HADR for database & the other infrastructure after reporting success to primary.

4.3 IN PRIMARY REGION

a) Start HADR on primary server and handshake with secondary.
b) The HADR pair is established and in peer state now and synchronized.

5. PRE-REQUISITE KNOWLEDGE FOR PROJECTS

The following points need to be known for implementing HADR further to live projects:

- Mark the projects under a division that need to be made available for HADR.
- Categorize the data involved in the projects as per priority in which preservation should be done.
- List and identify possible scenarios that may cause a requirement for HADR.
- Provide appropriate infrastructure to accommodate for the data to be stored that needs to be given HADR.
- Transform/process the data and using appropriate techniques, such as replication, higher level of authentication, caching, data segregation etc.
- Whenever an event occurs that requires data to be HADR, use information retrieval to extract the data that has been stored.
- Prioritize the data if it needs to be fetched instantly or can be fetched later, make HADR high priority for data that is volatile.

6. CONCLUSIONS

Organizations use HADR because they deal with huge amounts of client and customer data which is valued in terms of trust. Hence, it is utmost important to keep this data preserved and safe under any circumstance, and also provide ease of access whenever necessary for transactions.

Thus, this paper provides a robust but simple, generalized approach to implement HADR on any cloud architecture, benefiting the small scale to large scale industries.

This can however be modified further using multiple cloud architectures, multiple standby regions, caching data etc. On a larger scale multiple services can be added which will further help synchronize data among the regions providing resiliency, privacy, consistency and data integrity.

The paper thus provides an approach to imbibe the aforementioned properties in industrial projects and implement High Availability and Disaster Recovery.

ACKNOWLEDGEMENT

A lot of analysis and reading was done prior to the curation of the content of the given paper. This would definitely not be possible without the guidance of my mentor, Prof. B.K. Srinivas, who constantly guided me and gave me a direction in the due course of the given paper. None of this would have been possible if our Head of Department, Dr. B.M. Sagar, had not provided us with this opportunity.

I would like to also thank Mr. Vinay Rapurohit, Mr. Vatsal Garg, Mr. Rahul Toshniwal, Mr. Rahul Singh and Mr. Abhishek Todmal for enlightening me about HADR and providing me with an opportunity to work on a project in the same field.

REFERENCES

[1] Ashish L. Gawali, Subash RAJAA. Method, apparatus and system for improving failover within a high availability disaster recovery environment. US, 2008.

[2] Qin Zhang, Hong Xu. A High Availability and Disaster Recovery System. 2008 IEEE Conference on Robotics, Automation and Mechatronics, 2008.

[3] T. Adheshiyan, C. R. Attanasio, E. M. Farr, R. E. Harper, D. Pelleg, C. Schulz, L. F. Spainhower, P. Ta-Shma, L. A. Tomek. Using virtualization for high availability and disaster recovery. IBM Journal of Research and Development (Volume: 53, Issue: 4, July 2009).

[4] Ramanna Sathyanarayana, Jaikishan DABHOLE, Taghi Fatemi, Tony Imbierski. Flexible high availability disaster recovery with a set of database servers. US, 2014.

[5] Aditi Tripathi, Mayank Deep Khare, Pradeep Kumar Singh. Efficient and secured approach for faster data availability and restoration in disaster cloud data management. 2015 International Conference on Green Computing and Internet of Things (ICGCIoT), 2015.

[6] Chad M. Lawler, Stephen A. Szygenda, Mitchell A. Thornton. Techniques for Disaster Tolerant Information Technology Systems. 2007 1st Annual IEEE Systems Conference.

[7] Emmanouil Serrelis, Nikolaos Alexandris. Fault Tolerant Production Infrastructures in Practice. 2007 IEEE 18th International Symposium on Personal, Indoor and Mobile Radio Communications, 2007.

[8] D. Clitherow, M. Brookbanks, N. Clayton, G. Spear. Combining high availability and disaster recovery solutions for critical IT environments. IBM Systems Journal, 2008.

[9] Zhe Zhang, Yu Gu, Fan Ye, Hao Yang, Minkyong Kim, Hui Lei, Zhen Liu. A Hybrid Approach to High Availability in Stream Processing Systems. 2010 IEEE 30th International Conference on Distributed Computing Systems, 2010.

[10] Th. Lumpp, J. Schneider, J. Holtz, M. Mueller, N. Lenz, A. Biazetti, D. Petersen. From High Availability and Disaster Recovery to business continuity solutions. IBM Systems Journal, 2008.