

RAIN: Always On Data Warehousing

Jorge Vieira¹, Marco Vieira², Marco Costa¹, Henrique Madeira²

¹ Critical Software SA
Coimbra, Portugal
{jvieira, mcosta}@criticalsoftware.com

² CISUC, Department of Informatics Engineering, University of Coimbra
Coimbra, Portugal
{mvieira, henrique}@dei.uc.pt

Abstract. The Redundant Arrays of Inexpensive DWS Nodes (RAIN) technique is a node-level data replication approach that introduces failover capabilities to DWS (Data Warehouse Striping) clusters. RAIN is based on the selective replication of fact tables' data across the cluster nodes and endows DWS clusters with the capability of providing query answers even when one or more nodes are unavailable. Two distinct replication modes are supported: simple redundancy (RAIN-0) and striped redundancy (RAIN-S). In this demo we are going to show a DWS cluster using the RAIN technique, focusing on the execution of queries in the presence of nodes failures and on the process of recovering failed nodes.

Keywords: Data warehousing, redundancy, replication, recovery, availability.

1 Introduction

The success of organizations depends more and more on the information they have and, consequently, on the systems used to store and manage that information. In fact, markets globalization, with the consequent increase of the competitiveness, lead organizations to regard information as one of their most valuable resources.

The availability of tailored information to help decision makers during decision support processes is of utmost importance. Data warehouses (DW) are becoming one of the main assets for enterprise information analysis and manipulation [3]. Enterprises continuously create huge amounts of operational data that is typically integrated in a centralized data warehouse.

A data warehouse can store data ranging from hundreds of Gigabytes to the dozens of Terabytes [2]. Obviously, this requires large and highly-available storage devices and the capability of accessing and processing the data in due time. A low response time for the decision support queries issued by the users is of utmost importance.

DWS is a low-cost approach that distributes the data of a data warehouse by a cluster of computers, providing near linear speedup and scale up as new nodes are added to the cluster [1]. However, adding nodes to the cluster also increases the probability of node failure, which in turn leads to data availability problems.

In [5] we propose an approach that provides DWS clusters with high-availability even in the presence of node failures. The RAIN (Redundant Array of Inexpensive Nodes) technique is based on the selective data replication over a cluster of low-cost nodes and includes two types of replication: simple redundancy (RAIN-0) and striped redundancy (RAIN-S). RAIN-0 consists of replicating the facts data from each node in other nodes of the cluster. In RAIN-S facts data from each node is randomly distributed in $N-1$ sub-partitions (where N is the number of nodes in the cluster) and each sub-partition is replicated in at least one of the other nodes. See [5] for more details on RAIN.

The goal of this demo is to show the always on data warehousing capabilities of DWS clusters based on the RAIN approach. During the demo we will exhibit the advanced potential of DWS and RAIN, namely in what concerns to: high-performance using low cost hardware and software; always on data warehousing even in the presence of node failures; very easy and fast node recovery; and non-stop data loading.

2 The Demo

Fig. 1 depicts the setup that will be used for the demo. The basic platform consists of a small cluster using four heterogeneous machines and running PostgreSQL 8.2 database engine over the Debian Linux Etch operating system. The machines are connected using a dedicated Fast Ethernet Network.

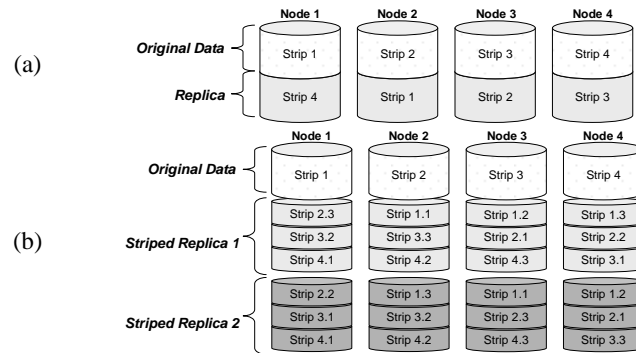


Fig. 1. Cluster setup for the demo. Two configurations will be used: a) a RAIN-0 cluster designed to tolerate the failure of one node (RAIN 0(4, 1)) and b) a RAIN-S cluster configured to tolerate the failure of two nodes (RAIN S(4, 2)).

The newly TPC-DS™ Benchmark (TPC-DS) [4] is used as case study. TPC-DS is a performance benchmark for decision support systems. This benchmark evaluates the essential features of decision support systems, including queries execution and data load. In the demo we use a small size database with 10GB (scale factor 10 in TPC-DS).

2.1 Queries Execution

Query execution in DWS is enabled through the use of a middleware that allow client applications (e.g., Oracle Discoverer, JPivot, Crystal Reports) to connect to the system without knowing the cluster implementation details. The DWS middleware was adapted to the RAIN technique, which means that it is prepared to transparently allow the system to continue providing exact queries answers when nodes fail.

When a node failure is detected in a RAIN-0 configuration the DWS middleware automatically redirects the corresponding query to one of the nodes that contains the replica of the failed node data.

In a RAIN-S configuration, when a node fails the DWS middleware automatically forwards a set of queries (equivalent to the query that should be sent to the failed node) to all other nodes. If Y nodes fail at the same time in a configuration able to tolerate Y node failures or more the DWS middleware always distributes the queries in such way that the final nodes workload is as uniform as possible.

As mentioned before, both RAIN-0 and RAIN-S cluster operation will be shown in the demo, including operation in the absence of node failures and operation in the presence of one or two node failures.

2.2 Cluster Data Loading

In typical data warehousing systems data loading occurs in two different moments: 1) initial load when the system starts to be used; 2) periodical load of the new data that represents the activity of the business since the last load. The initial load of the DWS system comprises the load of all dimensions in all nodes and the partitioning of the facts data across the several nodes. The periodical load includes updating the dimensions data in all nodes and the distribution of the new facts data through the several cluster nodes.

A relevant problem in a typical DWS cluster (without redundancy) is that when one or more nodes are unavailable, data loading is not possible (because there are nodes that are not available to store their data) and has to be delayed to a moment where all nodes are available. As we will show in the demo, the RAIN technique allows data loading to be still completely performed even when a node is unavailable during the data loading (data of the unavailable node is loaded into that node replicas). When the node becomes available again it is automatically synchronized using the data previously loaded into its replicas.

2.3 Node Recovery

Node recovery is another important aspect that will be demonstrated. In fact, in a typical DWS cluster node recovery is not possible when stored data gets corrupt. The only way to recover a node is to recollect the data from the operational sources and rebuild the entire cluster. The use of RAIN eases the node recovery, as the recovery process can be accomplished using the data existing in the failed node replicas, without need of cluster downtime. After a node failure, two kinds of recovery can be

needed: complete recovery or partial recovery. Complete recovery is needed when the node failure results in lost of node data. Partial recovery is required when the node still contains all its data, with exception to data loaded during the node unavailability.

Both complete recovery and partial recovery will be demonstrated. When partial recovery is performed the missing data is copied to the node from its replicas. When using RAIN-S the impact on other nodes is very small, as only a small amount of data has to be copied from each cluster node. When using RAIN-0, the impact is slightly high (all data is obtained from a single node), but still not very high, as the data to be copied corresponds to only a small period of time.

A complete recovery can be performed using the same approach used for the partial recovery, as all the data needed for a given node is distributed among the other nodes. However this has a higher impact in the performance as more data has to be loaded.

An important aspect we will show is that during node recovery the system continues to process query requests by using the replicas to process them. Nodes being recovered are kept offline until all its data is updated. Note that, node recovery is performed while the cluster is idle (or with minimum load) which minimizes the overhead caused by nodes recovery in the queries execution. In fact, our mechanism is able to pause or slowdown the recovery process when queries are being executed.

3 Conclusion

In this demo we present the Redundant Arrays of Inexpensive DWS Nodes (RAIN) technique, which is a node-level data replication approach that introduces failover capabilities to DWS (Data Warehouse Striping) clusters. This technique endows DWS clusters with the capability of providing exact queries answers even in the presence of node failures. The advanced capabilities of DWS and RAIN are demonstrated using a 4 nodes cluster designed to tolerate failures of one and two nodes. The goal of the demo is to show that high-performance is possible even using low cost hardware and software; always on data warehousing is feasible even in the presence of node failures; node recovery is very easy and fast; and non-stop data loading is achievable. This guarantees high data availability, a key characteristic for future data warehouses.

References

1. Bernardino, J., Madeira, H., "A New Technique to Speedup Queries in Data Warehousing", ABDIS-DASFA, Symp. on Advances in DB and Information Systems, Prague, 2001.
2. IDC, "Survey-Based Segmentation of the Market by Data Warehouse Size and Number of Data Sources", 2004.
3. Kimball, R., Ross, M., "The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling (Second Edition)", Ed. J. Wiley & Sons, Inc, ISBN: 0471200247, 2002.
4. Transaction Processing Performance Council, "TPC BenchmarkTM DS (Decision Support) Standard Specification, Draft Version 32", 2007, available at: <http://www.tpc.org/tpcds/>.
5. Vieira, J., Vieira, M., Costa, M., Madeira, H., "Redundant Array of Inexpensive Nodes for DWS", The 13th International Conference on Database Systems for Advanced Applications (DASFAA 2008), New Delhi, India, 2008.