Compound Treatment of Chained Declustered Replicas Using a Parallel Btree for High Scalability and Availability

Min Luo¹, Akitsugu Watanabe¹, and Haruo Yokota¹,²

¹ Department of Computer Science, Tokyo Institute of Technology
2–12–1 Ookayama, Meguro-ku, Tokyo 152–8552, Japan
² Global Scientific Information and Computing Center, Tokyo Institute of Technology
2–12–1 Ookayama, Meguro-ku, Tokyo 152–8550, Japan

Abstract. Scalability and availability are key features of parallel database systems. To realize scalability, many dynamic load-balancing methods with data placement and parallel index structures on shared-nothing parallel infrastructure have been proposed. Data migration with range-partitioned placement using a parallel Btree is one solution. The combination of range partitioning and chained declustered replicas provides high availability while preserving scalability. However, independent treatment of the primary and backup data in each node results in long failover times. We propose a novel method for compound treatment of chained declustered replicas using a parallel Btree, called the Fat-Btree. In the proposed method, the single Fat-Btree provides access paths to both primary and backup data in all processor elements, which greatly reduces failover time. Moreover, it enables dynamic load balancing without physical data migration, and improves memory space utilization for processing the index. Experiments using PostgreSQL on a 160-node PC cluster demonstrate the effect.

1 Introduction

The explosive growth of digital information, together with the high performance and availability requirements, has driven a continuing interest in the research on database systems in shared-nothing parallel environments in which replication plays an important role in availability and scalability with load balancing among the different processing nodes. However, existing replicated-database systems have a weakness in scaling up under frequent update requests because of the high costs of synchronizing the replicas. Well-known approaches for cloud applications in large data centers, such as PNUTS, Dynamo and BigTable, sacrifice strong consistency for scalability. However, they lose opportunities of using the high system throughput for applications requiring stricter consistency, because the replicas are not consistent most of the time in these approaches. Moreover, the advantage of higher availability gained from the replication may also be lost in the long run because of the inconsistency. Therefore, this trade-off between scalability, availability and consistency has long been an obstacle in efficient replication techniques.
To attack this problem, efficient data access methods guaranteeing consistency between the replicas play an important role. The data access methods must also be capable of handling sophisticated failover and dynamic load-balancing for availability and scalability. Many parallel index structures have been introduced to access the data stored in a shared-nothing environment [16]. For instance, the Fat-Btree [24] and the aB⁺-tree [13] are range-partition-based parallel indexes providing dynamic data allocation for high scalability. However, these parallel indexes are mainly focused on throughput or latency rather than availability.

Chained declustering data placement [9] adopts a low degree of replication to realize availability and scalability. From the availability point of view, only low degrees of replication are required. The range-partition-based index method is well suited to chained declustering. However, as far as we know, no valid indexing methods have been proposed to consider the use of replicas in chained declustering to enhance system availability and scalability.

We propose a database infrastructure for indexing range-partitioned data with a low degree of replication to achieve high scalability and availability without sacrificing data consistency. We first consider a parallel index structure on a range-partitioned chained replication database as a straightforward configuration for the infrastructure. We then propose a novel method of compound replica treatment utilizing the Fat-Btree index. It reduces the management cost of the index structure, balances load without data migration and enables shorter failover times. We also adopt the neighbor write-ahead log protocol (nWAL) [10] adapted to the proposed configuration, to reduce the synchronization cost between the replicas without data loss.

The key innovative points of this work are: a) it is the first proposal for managing both primary and backup within one directory structure; b) it has an efficient automatic load-balance algorithm for dynamic load skew without data migration; and c) it has an efficient failover algorithm for higher system availability without the cost of “promotion” backups. To the best of our knowledge, no previous work provides practical dynamic load-balancing or failover utilities in chained replication database systems, although they have long been claimed [8, 20]. We have evaluated our method on a PC cluster with 160 nodes. The experimental results demonstrate all the above-mentioned effects.

2 Background

We briefly review three technologies in shared-nothing parallel databases: data placement strategies, parallel indexing structures and synchronization methods.

2.1 Data Placement Strategies

Chained declustering [9] offers high availability, scalability and load balancing on shared-nothing parallel database systems [3]. Two physical copies of the relation are declustered by the same partitioning strategy, and the corresponding fragments of these two copies are stored on different Processing Elements (PEs), so data on a failed node will not be completely lost during the failure.