Dynamo论文读书笔记

0、论文知识点

- Dynamo, a highly available key-value storage system that some of Amazon’ s core services use to provide an “always-on” experience.
- Dynamo是一个高可用的易扩容的键值存储系统。

- To achieve this level of availability, Dynamo sacrifices consistency under certain failure scenarios.
- Dynamo通过牺牲某些失败场景下的一致性来换取可用性。

- using tens of thousands of servers located in many data centers around the world.
- 其中面对跨洲的多机房多活问题。

- Dynamo uses a synthesis of well known techniques to achieve scalability and availability: Data is partitioned and replicated using consistent hashing [10], and consistency is facilitated by object versioning [12].
- 数据使用一致性哈希进行分片和复制；通过对象版本来保证一致性。

- The consistency among replicas during updates is maintained by a quorum-like technique and a decentralized replica synchronization protocol.
- 副本之间的一致性是通过一个类似多数派的去中心化的复制协议实现的。

- Dynamo employs a gossip based distributed failure detection and membership protocol.
- Dynamo采用基于gossip的分布式故障检测和成员协议。

- It demonstrates that an eventually-consistent storage system can be used in production with demanding applications.
- 最终一致性的系统也能满足生产环境的业务需求。

- Some of these services are stateless (i.e., services which aggregate responses from other services) and some are stateful (i.e., a service that generates its response by executing business logic on its state stored in persistent store).
- 定义的无状态服务和有状态服务很好。

- Traditionally production systems store their state in relational databases. For many of the more common usage patterns of state persistence, however, a relational database is a solution that is far from ideal. Most of these services only store and retrieve data by primary key and do not require the complex querying and management functionality offered by an RDBMS. This excess functionality requires expensive hardware and highly skilled personnel for its operation, making it a very inefficient solution. In addition, the available replication technologies are limited and typically choose consistency over availability. Although many advances have been made in the recent years, it is still not easy to scale-out databases or use smart partitioning schemes for load balancing.
- 虽然现在业界的关系数据库做的很好，但是大多是强调一致性而牺牲部分可用性。同时关系数据库提供的很多复杂查询的特性在很多场景是使用不到的，只是根据主键进行一个点查。

- Dynamo has a simple key/value interface, is highly available with a clearly defined consistency window, is efficient in its resource usage, and has a simple scale out scheme to address growth in data set size or request rates.
- 在一个清晰定义的一致性窗口内是高可用的。

- In addition to the actual data persistence component, the system needs to have scalable and robust solutions for load balancing, membership and failure detection, failure recovery, replica synchronization, overload handling, state transfer, concurrency and job scheduling, request marshalling, request routing, system monitoring and alarming, and configuration management.
- 负载均衡、成员、失败检测、失败恢复、副本同步、过载处理、状态转移、并发、任务调度、请求组织、请求路由、系统监控、系统告警和配置管理。

- this paper focuses on the core distributed systems techniques used in Dynamo: partitioning, replication, versioning, membership, failure handling and scaling.
- Dynamo主要解决分片、复制、版本、成员、失败处理和扩容。
  - 分片：使用一致性哈希。
  - 写的高可用：带协调的向量锁
  - 临时故障处理：Sloppy Quorum and hinted handoff
  - 永久故障恢复：使用Merkle trees的反熵
  - 成员和故障检测：基于Gossip的成员协议和故障检测。

- 分片
  - 使用了带虚拟节点的一致性哈希，数据怎么做到均衡的。

- 复制
  - R is the minimum number of nodes that must participate in a successful read operation. W is the minimum number of nodes that must participate in a successful write operation. Setting R and W such that R + W > N yields a quorum-like system.
  - R +。W > N 的多数派协议，但是这里的N是所有节点中的TopN。

- 临时故障处理
  - If Dynamo used a traditional quorum approach it would be unavailable during server failures and network partitions, and would have reduced durability even under the simplest of failure conditions. To remedy this it does not enforce strict quorum membership and instead it uses a “sloppy quorum”; all read and write operations are performed on the first N *healthy* nodes from the preference list, which may not always be the first N nodes encountered while walking the consistent hashing ring.
  - 不是传统的多数派协议，使用了一种"Sloppy quorum"协议。核心点是这里的选取的N个节点是健康度比较高的N个节点。

  - Data center failures happen due to power outages, cooling failures, network failures, and natural disasters.
  - 电源故障、冷却故障、网络故障、自然灾害都会导致数据中心的故障。

- 永久故障解决：
  - Dynamo uses Merkle trees for anti-entropy as follows: Each node maintains a separate Merkle tree for each key range (the set of keys covered by a virtual node) it hosts. This allows nodes to compare whether the keys within a key range are up-to-date. In this scheme, two nodes exchange the root of the Merkle tree corresponding to the key ranges that they host in common. Subsequently, using the tree traversal scheme described above the nodes determine if they have any differences and perform the appropriate synchronization action. The disadvantage with this scheme is that many key ranges change when a node joins or leaves the system thereby requiring the tree(s) to be recalculated.
  - 使用Merkle trees来快速确定哪些副本的数据不一致然后进行复制。

- 成员变更和故障检测
  - 成员变更
    - A gossip-based protocol propagates membership changes and maintains an eventually consistent view of membership.
    - 通过Gossip协议来扩展集群新增节点或者删除节点。

- 内部发现
  - To prevent logical partitions, some Dynamo nodes play the role of seeds. Seeds are nodes that are discovered via an external mechanism and are known to all nodes.
  - 为了避免逻辑分区，最初的种子是外部获取的。

- 失败节点检测
  - Decentralized failure detection protocols use a simple gossip-style protocol that enable each node in the system to learn about the arrival (or departure) of other nodes.
  - 节点之间的状态检测也是基于gossip版本的协议进行。

- In Dynamo, each storage node has three main software components: request coordination, membership and failure detection, and a local persistence engine.
- 那个节点会有几个重要组件：请求协调器、成员信息表、失败检测和一个持久化引擎。

1、总结

- Dynamo是一个牺牲一定一致性来换取高可用，并同时达到线性扩容的目的简直存储系统。
- Dynamo是一个去中心化的系统，数据复制主要是多数派协议NRW。
- Dynamo的成员变更是通过Gossip协议进行扩散。
- Dynamo的成员状态检测也是通过Gossip协议进行的。
- Dynamo使用向量始终锁来保证多版本的写入，并解决写入冲突。
- Dynamo在写入时候也要求写入W和节点成功才算成功。
- Dynamo读取的时候会从R个节点读取数据并进行返回。
- Dynamo使用一致性哈希来解决扩容和负载均衡问题。
- Dynamo作为一个去中心化的系统选取了Merkle Trees来快速不一致性的数据进行同步。