CS 550 Final Project

Weilun Zhao;A20329942

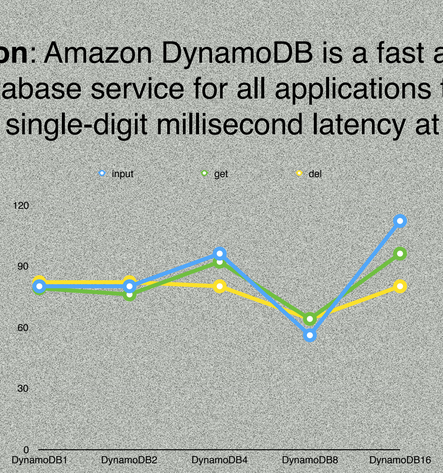
PosterID: 114

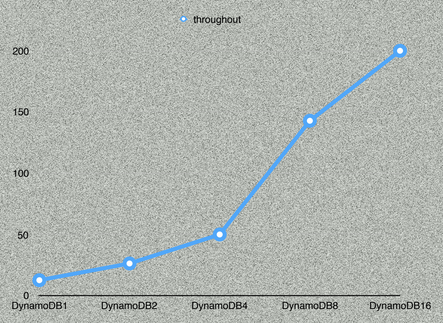
Abstraction:

Latency and throughput are critical element to evaluate a distribute system storing date into database. In this project, I evaluated the performance of EC2 remote system based on DynamoDB, MongoDB, Redis database, and DHT which stores key-value pair by using hash map. In addition, I also add an algorithm for loading balance which can reduce the latency, when clients send requests with key-value pair.

DynamoDB evaluation:

introduction: Amazon DynamoDB is a fast and flexible NoSQL database service for all applications that need consistent, single-digit millisecond latency at any scale.



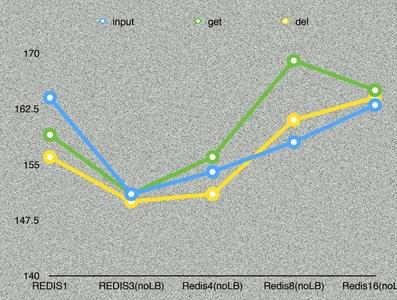


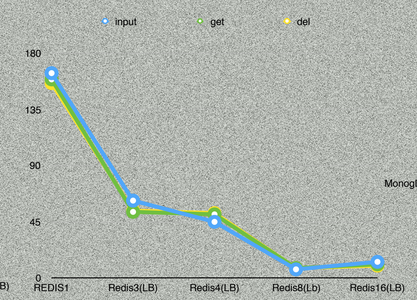
TIPS: Each EC2 node has the client part program and send requests to the dynamoDB at the same time. (by using PSSH)

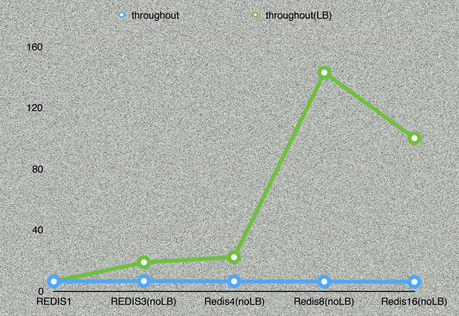
Redis evaluation:

introduction: Redis is an open source (BSD licensed), in-memory data structure store, used as database, cache and message broker.

In the Redis evaluation, I implement the load balance(LB) algorithm and get two sets of evaluation data. The LB method is like a hash map table function to connect destination server chosen by the value of key in the key-value pair, when clients send requests.





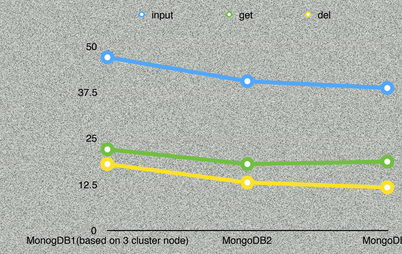


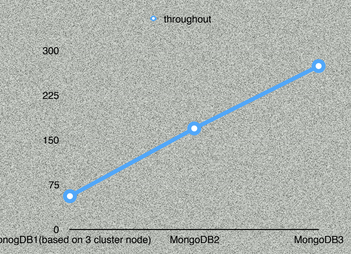
TIPS: the evaluation is based on 1, 2, 4, 8, and 16 nodes cluster, the server part software is deployed at each node. Only one client to send requests to the redis cluster.

MongoDB evaluation:

introduction: MongoDB is a cross-platform document-oriented database.

Evaluation: First, I deploy a cluster with three MongoDB nodes range. Each node have the config server function and run at all times. And client parts of the software should increase concurrency slowly from 1 client to 3 clients





TIPS: There is an evaluation deficiency which the cluster is only include 3 nodes rather than 16. And I don't want to provide fake data or something I don't know.

DHT evaluation:

This part evaluation used load balance method and deploy 1, 2, 4, 8, 16 nodes with server part software. Then one client, which have IP and port for each node, sends requests to servers