

UNIVERSITY OF TROMSØ

INF-3200 – Distributed Systems

Fundamentals

Fall 2015

Project 1: Distributed key-Value Store

Tuesday September 1th 2015

Alexander Nordvik

1.0 Introduction

This report describes the design and implementation of a simple Distributed key-Value Store using a cluster to distribute our data, we will look at how requests are sent and received between both the client and between the nodes.

2.0 Technical Background

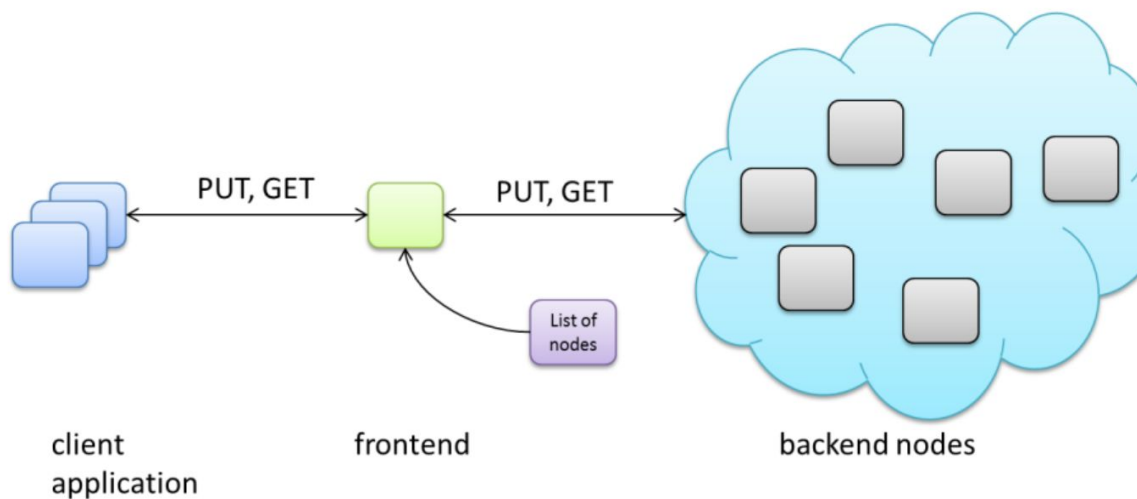


Figure 1:Key-value store architecture

2.1 Distributed Data Stores

A distributed Data store is a network of computers which provide a service for storing and retrieving data. Compared to a single machine system distributing the data across several servers can improve performance and increase scalability, but creating a system like this comes with the cost of a higher complexity and development effort, as well as needing server capacity.

Every process in the network executes the exact same thing, parsing messages in between each other and the client. The most common architecture is a simple distributed hash table.

3.0 Design

3.1 Frontend

The frontend is on a high-level architecture of the system, a client issues PUT and GET requests to the backend, the frontend is responsible for forwarding requests to the backend nodes that hold all the data. For this assignment frontend sends requests to a random backend node.

3.2 Backend

When receiving a request from the frontend the backend is responsible for completing the requested operation. For each Put command sent to the backend it hashes the key and breaks it down into three possible outcomes, 0, 1 or 2, as for this implementation three nodes were used. it then sends a new put command to the respectable node and the data is saved. For Get commands everything is opposite, a random node is called upon by the frontend, the same technique is then used for figuring out which node the data is in, once found everything is returned to the frontend and the client.

4.0 Implementation

This simple Distributed key-Value Store was implemented using Python and compiled in the Terminal.

For the implementation i had a few problems when receiving to the nodes, such as the host name received being “compute-1-1.local” leading to the implementation of a second “Node Network list” in each node of the backend system.

There was also a problem with the GREP files not stopping probably and having several instances of them running when testing.

5.0 Discussion

The implementation of the Distributed key-Value store uses a simple Hash-table and a simple key location for saving the data in the nodes, using only values ranging from 0-2 can potentially cause the nodes to be weighted at different workloads, but the more nodes we have the more spread out and better the implementation will be. But, a good hashing function can also spread it evenly. The GREP log files show that the data is spread almost evenly along the nodes, staying at around 30-30-40 percent for each node. There was an issue where the random chosen node would get significantly more traffic than all the other nodes due to it being the one talkative to the others either that or the hash function needs some work, this was especially hard to figure out as the grep log files were not responding well.

6.0 Conclusion

In this report the author has shown how the simple Distributed key-Value store is implemented, it is fully working on both lab computers and at home.

7.0 References

1.0

<https://github.com/uit-inf-3200/mandatory/blob/master/assignment1/doc/info/assignment.pdf>