Highly Available Distributed Configuration Stores

Kai Anter, Tim Heibel, Lukas Pietzschmann
lukas.pietzschmann@uni-ulm.de; kai.anter@uni-ulm.de;
tim.heibel@uni-ulm.de

Institute of Distributed Systems
University of Ulm

07/26/2023

Agenda

1. Motivation
2. Consul
   - 2.1 Overview
   - 2.2 Usage
3. etcd
   - 3.1 Overview
   - 3.2 Architecture
   - 3.3 Use cases
4. ZooKeeper
   - 4.1 Overview
   - 4.2 Data model
   - 4.3 Architecture
   - 4.4 Configuration management
5. Summary
6. References

1. Motivation

Why not just plain old configuration files?

Reasons

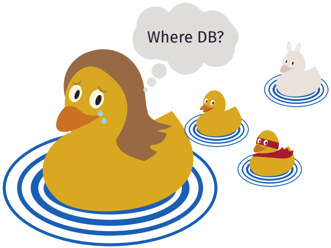
- Scenario: Services start & stop often
  - ▷ Hostnames in our configuration have to be changed frequently & clients have to pull often
- Configuration files have to be replicated
  - ▷ Risk of staleness

Solution: Configuration Store with monitoring $\Rightarrow$ changes can be pushed to clients

Use Cases

For Highly Available Distributed Configuration Stores

- Service Discovery
- Configuration of Applications
- (Node Coordination)



2. Consul

2.1 Overview

General

- Developed by *Hashicorp*
- Released in 2014 as a Service Discovery Platform
- Implemented in *Go*
- Now: Service-To-Service encryption, Health Checks, KV Store, ...
- Leader-based replication with Raft

Communication Types

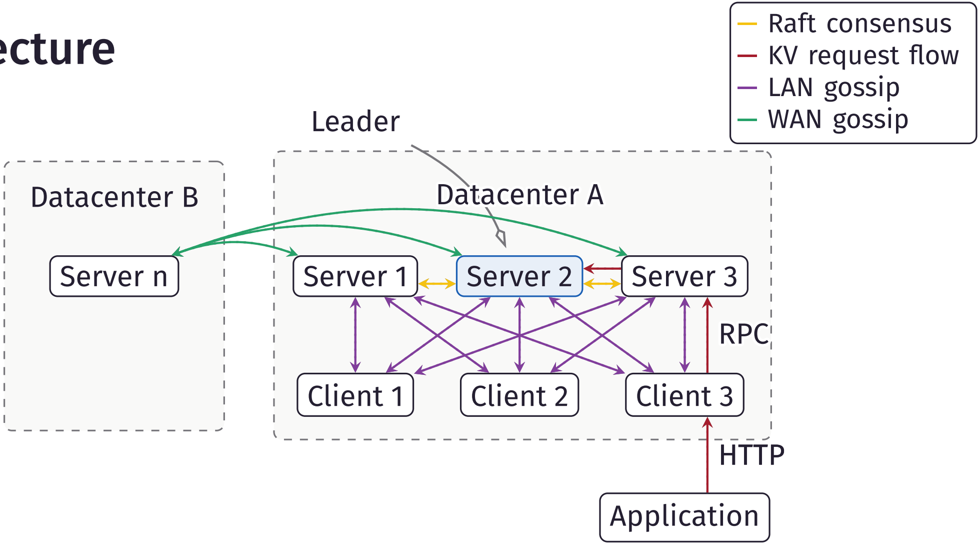
Raft Protocol (Consensus)

- Cluster State Replication
- Example: KV store, service and node IP addresses, configuration
- Crash Tolerance: (N/2) + 1
  - 3 Nodes: 1 Crash
  - 5 Nodes: 2 Crashes
  - 7 Nodes: 3 Crashes

Surf Protocol (Gossip)

- Perform and distribute *service health checks*
- Examples for health checks: via HTTP GET Request, gRPC, TCP, UDP

Architecture



Based on [Has23, Consul Architecture]

Consistency
in Consul

- *Writes*: always sent to leader
- *Reads*: three consistency modes:

default *leader leasing*: leader assumes its role for a specific time window and responds without quorum
(However: risk of 2 concurrent leaders $\Rightarrow$ stale reads)

consistent leader has to verify its role before responding

stale any server agent (leader & follower) can respond

2.2 Usage

CLI agent

for KV store

```
$ consul agent -dev

# other shell:
$ consul kv put my/key 123
$ consul kv get my/key
123
```

KV Store

- For configuration, locks, metadata, ...
- Max value size of 512 KB
- Requests to KV store via CLI or HTTP API

Long Polling

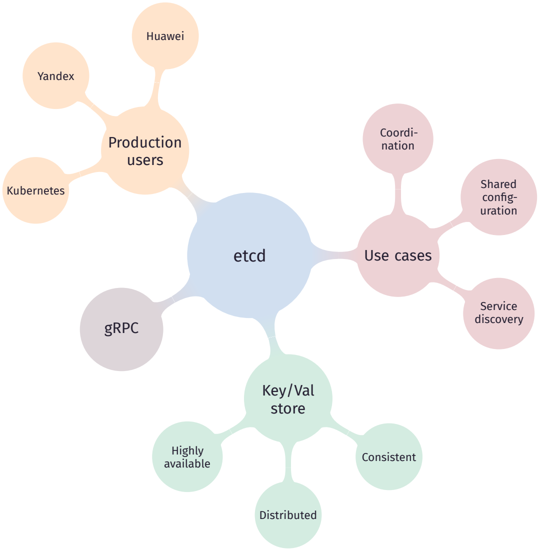
```
$ curl -v http://localhost:8500/v1/kv/my/key
# ...
< X-Consul-Index: 19
# ...

$ curl -v http://localhost:8500/v1/kv/my/key?index=19
# blocks until value is changed or timeout is reached
# (max. 10 minutes)
```

3. etcd

3.1 Overview

Overview



etcd
Production users
Huawei
Yandex
Kubernetes
Use cases
Coordi-
nation
Shared
config-
uration
Service
discovery
gRPC
Key/Val
store
Highly
available
Distributed
Consistent

History

1. First commit 2013 by CoreOS
2. 2014 etcd V0.2 - Kubernetes V0.4
3. 2015 First Stable Release of V2.0
   - includes Raft
4. 2018 CNCF (**C**loud **N**ative **C**omputing **F**oundation) Incubation
5. 2019 etcd V3.4
6. 2021 etcd V3.5

3.2 Architecture

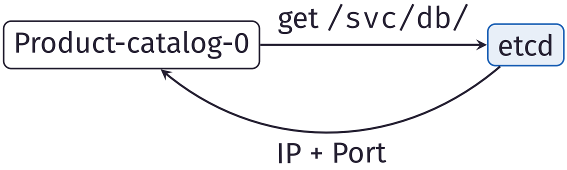
Architecture

- Operates across multiple nodes (cluster)
- Employs Raft algorithm
- Leader election if current leader crashes

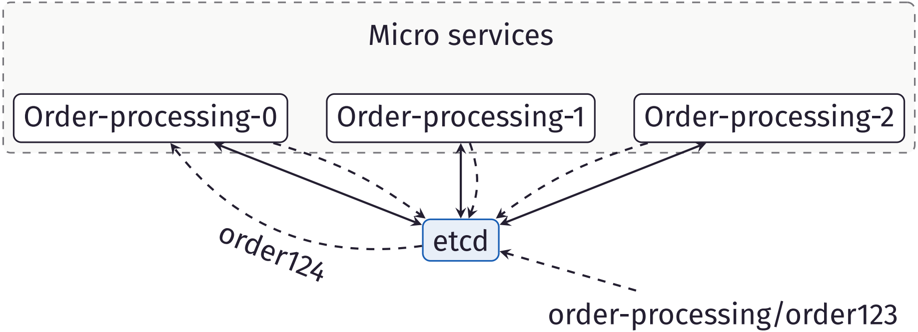
▷ high availability, consistency, distribution

3.3 Use cases

Service discovery



Distributed coordination



Configuration management

4. ZooKeeper

4.1 Overview

History

December 2006 First commit

November 2007 Version 0.0.1 on Sourceforge

June 2008 Moved to Apache

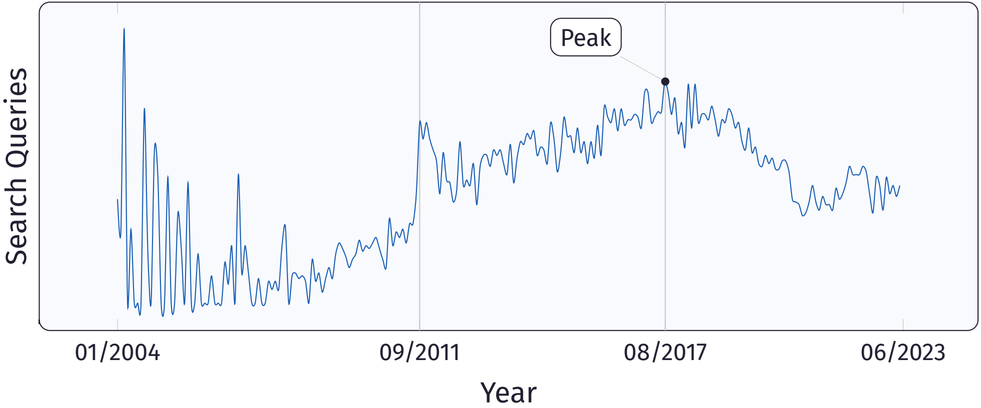
June 2010 First Paper [Hun+10]

November 2010 ZooKeeper becomes a top level project

Januar 2023 Version 3.8.1

Popularity

According to Google



What is ZooKeeper?

ZooKeeper is ...

... a (1) highly available, (2) scalable, (3) distributed, (4) configuration, (5) consensus, (6) group membership, (7) leader election, (8) naming, and (9) coordination service

What is ZooKeeper again?

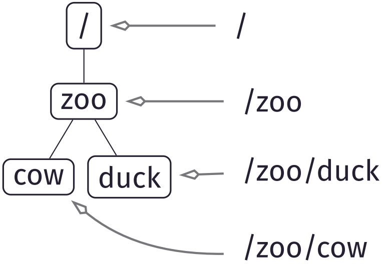
Use cases

Solve various coordination problems in large distributed system

- Leader election
- Barrier
- Queue
- Lock
- Service discovery
- Group services
- Configuration Stores

4.2 Data model

Data model



- Namespaces
- Three types of Znodes:
  - Persistent
  - Ephemeral
  - Sequential
- Not designed to store arbitrary blobs

Operations

Znodes

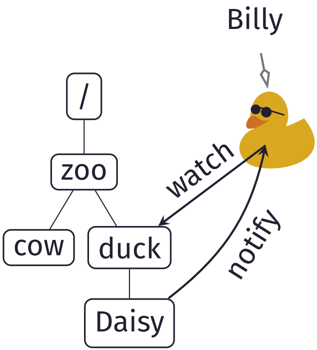
- create
- delete
- exists
- getChildren

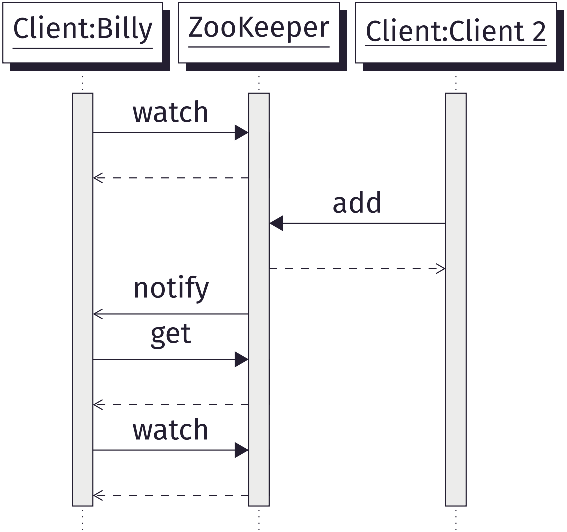
Data

- getData
- setData

- getAcl
- setAcl
- sync

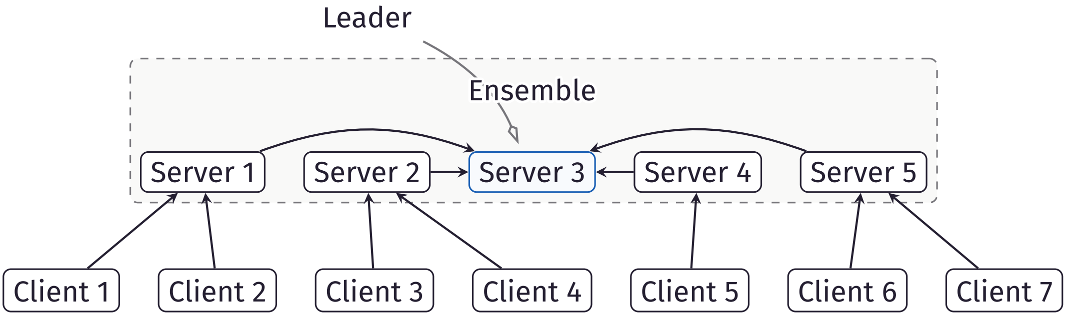
Watches


Billy
/
zoo
cow
duck
Daisy
watch
notify


Client:Billy
ZooKeeper
Client:Client 2
watch
add
notify
get
watch

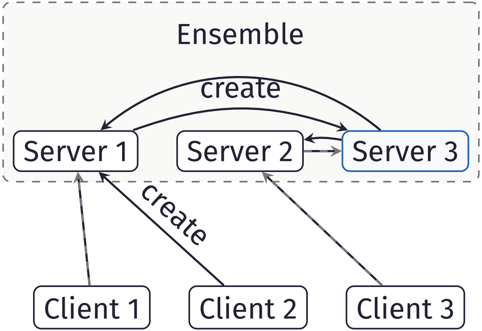
4.3 Architecture

Architecture



Architecture

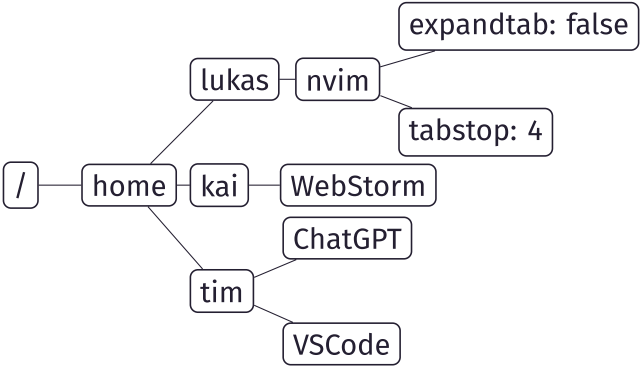
An example



4.4 Configuration management

Workflow

Absolutely simplified




```
set noexpandtab
set tabstop=4
```

5. Summary

Take-home message

- If you need *all inclusive* service discovery framework: Consul
- If you need a fast distributed key-value store: etcd
- If you like legacy systems: ZooKeeper

6. References

References

Consul

[OO14] Diego Ongaro and John Ousterhout. *In Search of an Understandable Consensus Algorithm (Extended Version).* Tech. rep. Stanford University, 2014. URL: `https://raft.github.io/raft.pdf`.

[Has23] Hashicorp. *Consul Documentation.* 2023. URL: `https://developer.hashicorp.com/consul/docs`.

References

etcd

[Clo21] Alibaba Cloud. “Etcd: History, Evolution, and Proper Usage | Medium”. In: *Medium* (Dec. 2021). URL: https://alibaba-cloud.medium.com/getting-started-with-kubernetes-etcd-a26cba0b4258.

[etc23] etcd-io. *etcd*. [Online; accessed 21. Jun. 2023]. June 2023. URL: https://github.com/etcd-io/etcd/blob/main/ADOPTERS.md.

[23a] *v3.5 docs*. [Online; accessed 21. Jun. 2023]. June 2023. URL: https://etcd.io/docs/v3.5.

[23b] *What is etcd? | IBM*. [Online; accessed 21. Jun. 2023]. June 2023. URL: https://www.ibm.com/topics/etcd.

References

ZooKeeper

[Hun+10] Patrick Hunt et al. “ZooKeeper: Wait-Free Coordination for Internet-Scale Systems”. In: *Proceedings of the 2010 USENIX Conference on USENIX Annual Technical Conference*. USENIXATC’10. Boston, MA: USENIX Association, 2010, p. 11.

[Hal15] Saurav Haloi. *Apache zookeeper essentials*. Packt Publishing Ltd, 2015.

References

General

[Com23] Hashicorp Company. "Service Discovery - Consul vs ZooKeeper vs etcd - Bizety: Research & Consulting". In: (2023). URL: https://www.bizety.com/2019/01/17/service-discovery-consul-vs-zookeeper-vs-etcd/.

[Los23] Martin Loschwitz. "Was Etcd, Consul, Zookeeper & Co. in der Praxis taugen". In: (2023). URL: https://www.linux-magazin.de/ausgaben/2015/08/konfig-management/6/.

Kai Anter, Tim Heibel, Lukas Pietzschmann

Ulm, 07/26/2023 lukas.pietzschmann@uni-ulm.de; kai.anter@uni-ulm.de; tim.heibel@uni-ulm.de