Etcd a GRPC

Ondřej Smola

12.04.2022 | CTC

Konsenzus

Konsenzus

- Model
  - Asynchronní systém se selháními
  - procesy mohou havarovat (fail-stop, tj. nikoliv byzantsky)
  - zprávy se mohou ztrácet (ale dodržují pořadí → nedokonalý FIFO kanál)
  - musí garantovat bezpečnost a měl by maximalizovat živost (dostupnost)
  - obojí garantovat nelze (FLP teorém)

Consensus

- Symetrický/bez lídra
  - všechny servery mají stejnou roli
  - klienti mohou kontaktovat kterýkoliv server
- Asymetrický/s lídrem
  - v každém okamžiku je jeden server lídrem a ostatní přijímají jeho rozhodnutí
  - klienti komunikují s lídrem
  - Raft/Etcd

Volba vůdce

- Design distribuovaných algoritmů pro řadu problémů distribuovaných výpočtů se zjednoduší, když jsou asymetrické a předpokládají, že jeden z procesů má roli lídra (a s ní spojenou logiku)
  - konsensus, replikace, vyloučení procesů, …
- V situacích, kdy uvažujeme selhání procesů, musíme být schopni lídra dynamicky nahradit

Problém volby

- Ze skupiny procesů vybrat lídra (který bude řešit specifické úkoly) a dát vědět všem procesům ve skupině, kdo je lídrem

- Co se stane, když lídr selže?
  - nějaký proces detekuje pomocí detektoru selhání a spustí nové volby

- Algoritmus pro volbu lídra musí zajistit:
  1. zvolí právě jednoho lídra z bezvadných procesů
  2. všechny bezvadné procesy ve skupině se shodnou na tom, kdo je lídr

Model

- Skupina N procesů s unikátními identifikátory.
  - známe všechny procesy, ale nevíme, které jsou aktivní (bezvadné)
- Procesy mohou havarovat

- FIFO perfektní komunikační kanál mezi každým párem procesů, tj. zprávy se neduplikují, nevznikají, neztrácejí a jsou doručovány v pořadí odeslání

- Asynchronní systém: neznámá, ale konečná latence

Další požadavky

- Každý proces může vyvolat volby
- Jeden proces může vyvolat v jeden okamžik pouze jedny volby
- Více procesů může vyvolat volby současně - pak požadujeme, aby se nakonec shodly na jednom lídrovi
- Výsledek volby lídra by neměl záviset na tom, který proces volby vyvolal
- Po skončení běhu algoritmu volby lídra má každý proces ve své proměnné $ELECTED$ identifikátor lídra s nejvyšší hodnotou volebního kritéria (a nebo nikdo, tj. volby skončily neúspěšně)
- Volební kritérium:
  - typicky nejvyšší identifikátor, tj. IP adresa
  - nejúplnější log v případě RAFTu
  - musí být fixní a známe všem procesům při zahájení volby

Etcd

Etcd

- Silně konzistentní distribuované úložiště typu klíč hodnota
- Naprogramováno v Go (https://github.com/etcd-io/etcd)
- Pro consensus používá protokol Raft

- Odolné vůči výpadkům strojům, síťovým chybám
- Silně konzistentní operace (CAS)

- Data uložena v paměti a na disk
  - Omezení na velikost paměti a celkově na maximálně 8GB
  - Maximální velikost klíče s hodnotou je 1.5 MB

Výkon

- https://etcd.io/docs/v3.5/op-guide/performance/

- 3 stroje
- 8 vCPUs, 16GB Memory, 50GB SSD
- Výkon
  - ~ 45 000 zápisů za sekundu
  - ~ 140 000 čtení za sekundu
  - > 10 000 000 sledujících klientů

Datový model

- Data ukládána do perzistentního úložiště s podporou historických verzí pojmenovaného bbolt
  - https://github.com/etcd-io/bbolt
- Podpora historických verzí umožňuje konzistentní snímek databáze v konkrétním čase
- Historické verze dat jsou odstraněny během kompakce

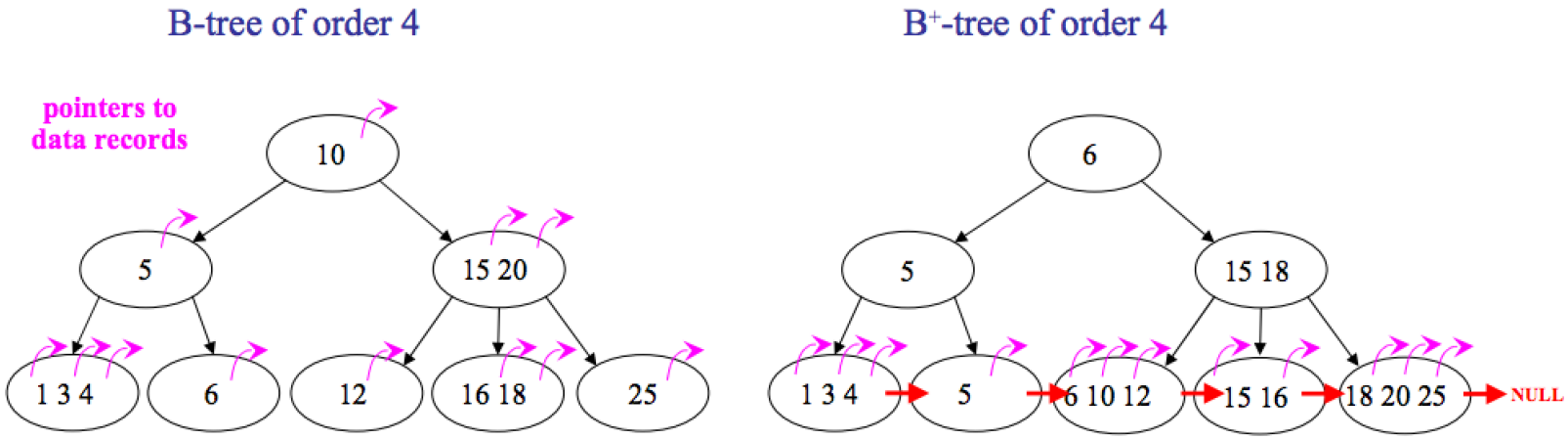
- U každého klíče je dále ukládána generace která začíná vytvoření klíče a končí jeho smazáním
  - Klíč může mít za dobu životnosti databáze více generací

Logická reprezentace

```
type KeyValue struct {
    // key is the key in bytes. An empty key is not allowed.
    Key []byte `protobuf:"bytes,1,opt,name=key,proto3" json:"key,omitempty"
    // create_revision is the revision of last creation on this key.
    CreateRevision int64 `protobuf:"varint,2,opt,name=create_revision,json=
    // mod_revision is the revision of last modification on this key.
    ModRevision int64 `protobuf:"varint,3,opt,name=mod_revision,json=modRev
    // version is the version of the key. A deletion resets
    // the version to zero and any modification of the key
    // increases its version.
    Version int64 `protobuf:"varint,4,opt,name=version,proto3" json:"versio
    // value is the value held by the key, in bytes.
    Value []byte `protobuf:"bytes,5,opt,name=value,proto3" json:"value,omit
    // lease is the ID of the lease that attached to key.
    // When the attached lease expires, the key will be deleted.
    // If lease is 0, then no lease is attached to the key.
    Lease                int64    `protobuf:"varint,6,opt,name=lease,proto3
    XXX_NoUnkeyedLiteral struct{} `json:"-"`
    XXX_unrecognized     []byte   `json:"-"`
    XXX_sizecache        int32    `json:"-"`
}
```

Fyzická reprezentace

- B+strom 3-tuple (revize, sub, type)
  - sub – slouží k odlišení různých klíču
  - type – značka/tombstone
  - hodnotou je rozdíl oproti přechozí revizi
- Bstrom pro mapování mezi klíči a 3-tuple během dotazování na interval



Cluster

- N členů
- N/2 + 1 (většina) členů musí být aktivních pro plnou funkcionalitu
- 1 leader a N - 1 následovníků
  - zvolen pomocí protokolu RAFT (dále)

- Detekce členů
  - Statická – pevně daný seznam
  - DNS _etcd-server-ssl._tcp.example.com
  - S pomocí jiného etcd clusteru

Výpadek následovníka – většina aktivní

- Výpadek klientských spojení s chybovými členy
- Cluster pokračuje dále ve funkcionalitě
- Po obnovení chybových členů dojde k jejich opětovnému zapojení do cluster

- Správa členů clusteru pomocí etcdlctl
  - Výpis členů
  - Přidání nového člena
  - Trvalé odebrání člena

Výpadek vůdce – většina aktivní

- Při výpadku dojde k detekci chybějícího vůdce dle nastaveného timeoutu pro volbu
- V mezičase jsou zápisy ukládány do bufferu
- Po zvolení nového leadera jsou zápisy odeslány ze zásobníku a cluster pokračuje v normální funkci
- Všechny lease jsou automaticky obnovena aby nedošlo k expiraci lease které byly obnoveny předchozím vůdcem těsně před výpadkem
- Během celé oprace nedojde ke ztrátě žádného potvrzeného zápisu, některé zápisy v zásobníku mohou ale timeoutovat

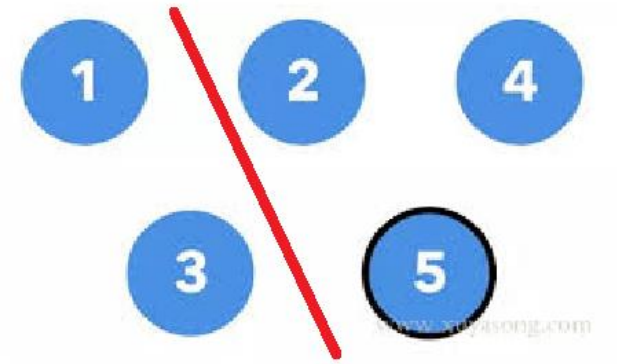
Výpadek většiny

- Dojde k výpadku celého clusteru a zápisy přestanou být akceptovány
- Čeká se na znovuobnovení vetšiny
- Poté dojde k zvolení nového vůdce

- Pokud není možná obnova (např. selhání disků) je nutné zahájit proces “obnovy po katastrofě”
  - Vychází se z obrazu dat přeživšího člena, případně historické zálohy
  - https://etcd.io/docs/v3.5/op-guide/recovery/

Síťové rozdělení

- Řešeno stejně jako
  - Výpadek následovníka – většina aktivní
  - Výpadek vůdce – většina aktivní
- První možnost zvolena pokud po síťvém rozdělení zůstane většina na straně vůdce a druhá možnost pokud vůdce zůstane na straně menšiny



Základní operace

- Práce v IDE

GRPC

Popis

- Moderní aplikační rámec pro RPC komunikaci
- Definice zpráv v binárním jazyce Protobuf

- Možnost propojení různých jazyků a platforem
- Podpora obousměrné binární komunikace
- Podpora autentizace

Grpc

- Práce v IDE