Contents

1 Úvod

1.1 Základní principy

Distribuovaný výpočetní výkon Výpočet úlohy se rozdělí mezi více uzlů, je tedy možné paralelní zpracování.

Distribuované uložiště Data se uchovávají na více místech - zajištění redundance, propustnosti a dostupnosti. Je ale složité zajistit konzistenci dat.

1.2 CAP teorém

Eric Brewer přisuzuje distribuovaným systémům tři základní vlastnosti, přitom ale systém může splňovat nejvýše dvě:

1. Konzistence
   - Pro každý požadavek vrácen správný výsledek.
   - Uzly vrací stejná data (distrin. uložiště).
   - Výsledky výpočtu se od jednotlivých uzlů neliší (distrib. výkon).
2. Dostupnost
   - Na každý požadavek přijde odpověď.
   - V praxi musí být systém vždy dostupný.
3. Tolerance výpadku
   - Určuje jak se systém zachová v případě výpadku nějaké jeho části.

2 NoSQL databáze

- Map-Reduce framework. Např. Hadoop.
- Kay-value uložiště. Např. Cassandra.
- Dokumentové uložiště (jako key-value, ale valuje je document). Např. CouchDB, MongoDB

2.1 Casssandra

- Open source distribuovaná databáze.
- Ukládá data mezi mnoha servery
- Poskytuje vysokou dostupnost, nemá single point of failure.
- Vyniká v rychlém zápisu.

2.1.1 Základní funkce

- Replikace - systém se sám replikuje podle zadaných kritérií.
- Škálování - systém škáluje sám bez nutnosti zásahu do aplikace.
- Nastavitelná konzistence - lze na stavit pro jednotlivé dotazy za běhu aplikace.
- Dotazovací jazyk CQL - *Cassandra Query Language*

Gossip protokol Každý uzel posílá informace každou vteřinu, udržuje tím informace o stavu uzlů. Informace jsou opatřeny timestampem.

Koordinátor Uzel, který *Cassandra driver* vybere jako node, kterému posílá požadavky a dotazy.

Partition token Identifikuje, unikátně každý uzel a na něm partition v celém ringu. Je generovavaný hashovací funkcí.

Partitioner Určuje jakým způsobem jsou data a jejich repliky rozděleny na clusteru - (Fce. MurMur3, Random, ByteOrder).

2.1.2 Pojmy

Keyspace Alternativa pojmu *databáze* ze světa relačních databází.

2.1.3 Cassandra data model

Column Sloupec je nejmenší entitou udržující data. Má název, hodnotu a timestamp vložení.

Row Identifikována pomocí jednoznačného row-key, nemá pevný formát.

Column-family Sdružuje řádkym které obsahují sloupce. Obdoba *tabulky* z relačního světa. *Mapa map*.

Keyspace Udržuje pohromadě všechny column-family a replikační faktor. Dá se přirovnat k *databázi* z relačního světa.

- Wide-column stores - data přibývají do šířky.
- Na nejspodnější úrovni se jedná o key-value storage.
- Funguje na podobném principu jako hash table.
- Nepoužívá se join.
- Chceme li použít range scan, musíme mít row key a column key.

2.2 CQL

- Podobné SQL.
- Stále nelze použít *JOIN*.

2.2.1 Create table

```
CREATE TABLE orders (
  id int PRIMARY KEY,
  code text,
  buyer_id text,
  created_at text,
  payed_at text,
  buyer_ip text
 );
```

2.2.2 Insert data

- Neprovádí se kontrola existence řádku, jedná se tedy spíše o *CREATE OR UPDATE*.
- Lze použít *IF NOT EXISTS*, ale zápis je pomalejší.

```
INSERT INTO orders (code, buyer_id, created_at)
VALUES ('AB1221', 1234, '2016-12-04');
```

2.2.3 Update data

- Pomocí *WHERE* klauzule lze vybrat řádek, ale musí se zadat všechny sloupce tvořící PK.

```
UPDATE orders
SET payed_at = '2017-01-01'
WHERE id = 5;
```

2.2.4 Delete data

- Umožňuje odstranit jen vybrané sloupce.

```
DELETE FROM orders
WHERE id = 5;
```

2.2.5 Batch

- Pomocí *BATCH* lze seskupit více příkazů.
- Šetří síťovou komunikace mezi klientem a serverem.
- Může obsahovat pouse *UPDATE*, *INSERT*, *DELETE*.

2.3 Select data

- Přečte jeden nebo více sloupců a řádků.
- Select klauzule
  - Určuje jaké sloupce mají být vráceny (buď seznam nebo znak *).
- Selector
  - Název sloupce nebo funkce jednoho nebo více sloupců.
  - Např *COUNT(*)*.
- V klauzuly *WHERE* umožňuje výběr jen pokud jsou uvedeny všechny PK.
- Chceme-li v rámci *WHERE* klauzule filtrovat na základě více sloupců, je potřeba dodat *ALLOW_FILTERING*.

```
SELECT id, buyer_id
FROM orders
WHERE id IN (5, 7, 15);
```

3 Hadoop, Map-Reduce

MapReduce Model pro distribuované paralelní výpočty. Data se rozdělí na jednotlivé elementy, které se po rozdělení navzájem neovlivní. Mapovací funkce převede vstupní elementy na výstupní elementy. Kombinační funkce následně výstupní elementy do výstupní hodnoty.

3.1 Hadoop

Open source implementace MapReduce frameworku. Hadoop nabízí také svůj FS (HDFS) - to ale není plně distribuvaný FS. HDFS obsahuje master server, který obhospodařuje namespace.

4 Apache Hive, Pig a Solr

4.1 Apache Hive

Infrastruktura pro těžení dat, postaveno nad Hadoopem. Jedná se o vrstvu, ktrá poskytuje abstrakci pomocí pseudo SQL.

- Dotazovací jazyk Hivu je HiveQL.
- Nabízí podmnožinu SQL dotazů rozšířenou o specifické dotazy (obsahuje např. *JOIN*, $FROM < DOTAZ >$).
- Neobsahuje standardní *INSERT* do existující tabulky.
  - Všechny inserty přepisují data.
  - Chybí také *UPDATE* a *DELETE*.

4.2 Solr

- Využívá Apachae Lucene.
  - Knihovna pro získávání dat.
  - Podporuje synonyma, podobnost, ...
- Index nad databází.
- Poskytuje REST API.
- Distribuované vyhledávání.