# Apache Cassandra

je bezplatný a open-source distribuovaný systém pro správu databází NoSQL. Pro komunikaci používá CQL(Cassandra Query Language), která se podobá SQL. Používá licenci **Apache License 2.0**

Měřeni výkonu datové struktury zaleží na 3 hlavních složek zesílení zápisu, zesílení čtení a zesílení prostoru

**zesílení zápisu**

Zesílení zápisu je poměr mezi množstvím dat zapsaných na úložné zařízení a množstvím dat zapsaných do databáze.

Pokud například zapisujete 10 MB do databáze a dodržujete rychlost zápisu na disk 30 MB, vaše zesílení zápisu je 3.

**zesílení čtení**

Amplifikace čtení je počet čtení disku na dotaz.

Pokud k zodpovězení dotazu potřebujete přečíst 5 stránek, zesílení čtení je 5.

**zesílení prostoru**

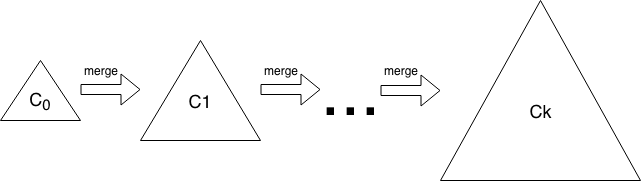
Zesílení prostoru je poměr mezi množstvím dat na úložném zařízení a množstvím dat v databázi.

Pokud do databáze vložíte 10 MB a tato databáze používá 100 MB na disku, pak je zesílení místa 10.

Obecně řečeno, datová struktura se může optimalizovat maximálně pro dva ze čtení, zápisu a zesílení prostoru. To znamená, že je nepravděpodobné, že by jedna datová struktura byla lepší než jiná vůbec všechny tři. Například B-strom má menší amplifikaci čtení než LSM-strom, zatímco LSM-strom má menší amplifikaci zápisu než B-strom

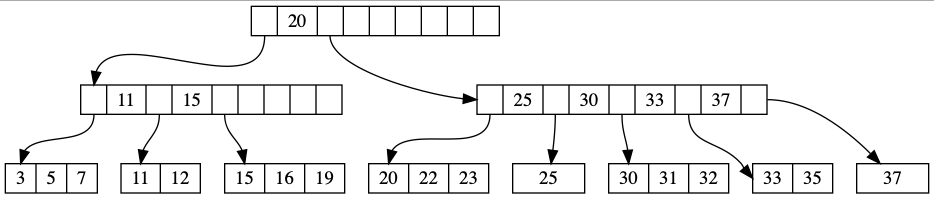
Hlavním důvodem extrémně rychlého zápisu Cassandry je její způsob ukládaní. Cassandra používá LSM-stromy, zatímco tradiční RDBMS používá B+ stromy jako datovou strukturu

**LSM-strom** je datová struktura s výkonnostními charakteristikami, díky nimž je atraktivní pro poskytování indexovaného přístupu k souborům s velkým objemem vkládání, jako jsou data transakčního protokolu. Stromy LSM, stejně jako jiné vyhledávací stromy, udržují páry klíč-hodnota



Strom LSM obsahuje k komponent. Data začínají v C0, pak se sloučí do C1. Nakonec se C1 sloučí do C2 atd.

**B-strom** ukládá klíče v jeho vnitřních svazkách, ale nemusí ukládat klíče v zápisech v jeho listech. B+ strom je jedním z nejznámějších variací B-stromu. Myšlenka za B+ stromem je to, že vnitřní svazky obsahují jen klíče dodatečná vrstva, která obsahuje hodnoty je podél propojených listů.



Kořenový uzel je zobrazen v horní části stromu a v tomto případě náhodou obsahuje jeden pivot (20), což znamená, že záznamy s klíčem k, kde k ≤ 20 jsou uloženy v prvním potomku, a záznamy s klíčem k, kde k > 20 je uloženo u druhého dítěte. První dítě obsahuje dva pivotní klíče (11 a 15), což znamená, že záznamy s klíčem k, kde k ≤ 11 je uloženo v prvním dítěti, záznamy s 11 < k ≤ 15 jsou uloženy v druhém dítěti a záznamy s k > 15 jsou uloženy ve třetím dítěti. Listový uzel nejvíce vlevo obsahuje tři hodnoty (3, 5 a 7).

Úložiště Cassandry má schopnost neukládat null hodnoty. Znamená to, že hodnota null je prostě hodnota, která by tam neměla být, avšak aby byla intuitivní CQL prázdnou hodnotu překládá vypíše jako null.

To poskytuje velkou flexibilitu, protože nulová hodnota, která není explicitně (nebo implicitně, viz později) vložena, nezabírá úložný prostor, ani nevyužívá výpočetní výkon a šířku pásma IO.

Zaplnovat řádky null hodnotami je poměrně jednoduché:

*INSERT INTO mytable (c1) VALUES (null);*

Tímto způsobem explicitně říkáte Cassandře, aby uložila nulovou hodnotu do c1. Podobný efekt však můžete získat s:

*INSERT INTO mytable (c1) VALUES (0x1234);*

Pokud však používáte tuto metodu, tak bude lepší z důvodu, jakým způsobem Cassandra “null” hodnoty ukládá. Cassandra při vkládání “null” uloží “náhrobek”(V Cassandře, vymazaná data nejsou ihned vyčištěno z disku. Místo toho Cassandra zapíše speciální hodnotu zvanou “náhrobek”, která značí, že data byly odstraněny.)

Druhý příklad je lepší z důvodu, jak Cassandra vyhledává.

Druhý příklad také ukládá “náhrobek”. To se stane, když používáte funkci TTL zabudovanou v Cassandře.

Aby se zabránilo náhrobkům, které zpomalují čtení, pro operace ukládání, má Cassandra koncept hodnoty vstupu nenastavený.

*Cassandra 2.2 přidává možnost rozeznání mezi null a nenastavenými vstupy v nativním protokolu v4. S minulými verzemi protokolu, když využíváme připravené dotazy musely se spojit všechny sloupce v dotazu spojit se vstupy, nebo by nastal error. Při používání protokolu v4+, ovladač nyní implicitně nastaví nespojené hodnoty na nenastavené. Toto funguje do té doby, dokud nespojená hodnota není součástí klíče. Aplikace můžou explicitně nastavit nenastavenou hodnotu pomocí cassandra.query.UNSET\_VALUE – název nenastavené hodnoty záleží na používaném ovladače.*

| **Datová struktura** | **Zesílení zápisu** | **Zesílení čtení** |
| --- | --- | --- |
| B+ strom | Θ(B) | O(logBN/B) |
| LSM strom | Θ(klogkN/B) | Θ((log2N/B)/logk) |

N je velikost databáze

K je růstový faktor

B je velikost jednotlivého souboru

Porovnáním různých druhů zesílení mezi stromem B+ a stromem LSM založeným na úrovni můžeme dojít k závěru, že strom LSM má lepší výkon zápisu než strom B+, zatímco jeho výkon při čtení není tak dobrý jako strom B+. Hlavním účelem použít LSM-strom místo B-stromu jako základního úložného jádra je to, že použití technologie mezipaměti(př. Redis) k podpoře výkonu čtení je mnohem jednodušší než zvýšení výkonu zápisu.

Praktické příklady:

<https://www.guru99.com/cassandra-data-model-rules.html>