



链滴

leveldb 和 rocksdb 性能对比

作者: [flowaters](#)

原文链接: <https://ld246.com/article/1490535950067>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

<p>
C++版本: <https://github.com/facebook/rocksdb/wiki/Performance-Benchmarks>
</p>
<p>
Java版本: <https://github.com/facebook/rocksdb/wiki/RocksJava-Performance-on-Flash-Sto>
age
</p>
<p>

</p>
<p>
详细的机器配置看上面的链接, 强烈建议读一下。
</p>
<p>
注意: 测试使用的是jemalloc
</p>
<p>

</p>
<p>
测试一: 将1B(即10亿)个key数据随机批量写入空数据库
</p>
<p>

</p>
<p>

</p>
<pre>rocksdb: 103 minutes, 80 MB/sec (total data size 481 GB, 1 billion key-values)
leveldb: many many days (in 20 hours it inserted only 200 million key-values)</pre>
<p>

</p>
<p>
原因:
</p>
<p>
rocksdb其实使用了两步: 先做fillrandom, 再做compact
</p>
<p>
leveldb只是使用了一步: fillrandom
</p>
<p>

</p>
测试二: 将1B个key数据顺序批量 插入空 <spa>
>数据库
<p>

</p>
<p>

</p>
<pre>rocksdb: 36 minutes, 370 MB/sec (total data size 760 GB)

leveldb: 91 minutes, 146 MB/sec (total data size 760 GB)</pre>

<pre></pre>

<pre>Java</pre>

<pre>

<pre>fillseq : 2.48233 micros/op; 311.2 MB/s; 1000000000 ops done; 1 / 1 task(s) finished.</pre>

</pre>

<p>

</p>

<p>

原因:

</p>

<p>

rocksdb使用了多线程compact

</p>

<p>

</p>

测试三: 随机覆盖1B个key数据

<p>

</p>

<p>

</p>

<p>

</p>

<pre>rocksdb: 15 hours 38 min; 56.295 micros/op, 17K ops/sec, 13.8 MB/sec

leveldb: many many days; 600 micros/op, 1.6K ops/sec, 1.3 MB/sec

(in 5 days it overwrote only 662 million out of 1 billion keys, after which I killed the test

</pre>

<p>

</p>

<p>

原因:

</p>

<p>

rocksdb使用了20个compact线程, 使用了6个level, 仅了level 2及以上启用了gzip

</p>

<p>

</p>

<p>

测试四: 随机读取1B个key数据

</p>

<p>


```
</p>
<p>
  <br />
</p>
<pre>rocksdb: 70 hours, 8 micros/op, 126K ops/sec (checksum verification)
leveldb: 102 hours, 12 micros/op, 83K ops/sec (no checksum verification)</pre>
<pre></pre>
<pre>Java:</pre>
<pre>

<pre>readrandom    :   7.67180 micros/op; 101.4 MB/s; 1000000000 / 1000000000 found;
32 / 32 task(s) finished.</pre>

<br />
</pre>
<p>
  <br />
</p>
<p>
  配置: rocksdb额外启用了checksum, leveldb关闭了checksum
</p>
原因: rocksdb没有用<span style="color:#24292E;font-family:-apple-system, BlinkMacSystem
ont, "font-size:16px;background-color:#FFFFFF;">mmapped IOs, 这个在某些linux平台上会慢。r
cksdb把block cache分成了64份, 以降低锁力度。 </span>
<p>
  <br />
</p>
<p>
  <br />
</p>
<p>
  <br />
</p>
<p>
  测试五: 多线程读, 同时单线程写
</p>
<p>
  <br />
</p>
<p>
  <br />
</p>
<pre>rocksdb: 11 hours 30 minutes, 9.640 micros/read, 103734 reads/sec
leveldb: 20 hours 28 minutes</pre>
<p>
  <br />
</p>
<p>
  Java:
</p>
<pre>readwhilewriting :   9.55882 micros/op; 81.4 MB/s; 100000000 / 100000000 found; 3
/ 32 task(s) finished.</pre>
<p>
```


</p>
<p>

</p>
配置: rocksdb release 2.7 and leveldb 1.15
<p>

</p>
<p>

</p>
<p>
PS: RocksDB的配置, 见官网: <https://github.com/facebook/rocksdb/wiki/RocksDB-Tuning-guide>
</p>
<p>

</p>
<p>
中文参考: <http://www.jianshu.com/p/ddf652aa4882>
</p>
<p>

</p>
<p>

</p>