黑客派

Java LevelDB 初体验

作者：pencilso

原文链接：https://hacpai.com/article/1530071106598

来源网站：黑客派

<h2 id="前言">前言</h2>

<pre><code class="highlight-chroma">LevelDB 是一种Key-Value存储数据库
性能非常强悍 百度百科上介绍 可以支撑十亿级
这段时间在研究区块链的时候发现的这个数据库
</code></pre>
<h2 id="引入SDK">引入 SDK</h2>
<pre><code class="highlight-chroma"><dependency>
    <groupId>org.iq80.leveldb</groupId>
    <artifactId>leveldb-api</artifactId>
    <version>0.10</version>
</dependency>
<dependency>
    <groupId>org.iq80.leveldb</groupId>
    <artifactId>leveldb</artifactId>
    <version>0.10</version>
</dependency>
</code></pre>
<h2 id="初始化DB">初始化 DB</h2>
<pre><code class="highlight-chroma">DBFactory factory = new Iq80DBFactory();
Options options = new Options();
options.createIfMissing(true);
//folder 是db存储目录
DB db = factory.open(new File(folder), options);
</code></pre>
<h2 id="存储Key-Value-值">存储 Key Value 值</h2>
<pre><code class="highlight-chroma">//levelDB 的api存储都是字节数组 所以这里需要转成字数组
db.put(Iq80DBFactory.bytes(key), Iq80DBFactory.bytes(value));
</code></pre>
<h2 id="获取Value">获取 Value</h2>
<pre><code class="highlight-chroma">byte[] bytes = db.get(Iq80DBFactory.bytes(key));
String value = Iq80DBFactory.asString(bytes);
</code></pre>
<h2 id="删除-更改">删除 | 更改</h2>
<pre><code class="highlight-chroma">//删除
db.delete(Iq80DBFactory.bytes(key));

//更改 重新put新的key - value即可
db.put(Iq80DBFactory.bytes(key), Iq80DBFactory.bytes(value));
</code></pre>

<h2 id="遍历所有数据">遍历所有数据</h2>
<pre><code class="highlight-chroma">public LinkedHashMap<String, String> iterator

```
b() {
    DBIterator iterator = db.iterator();
    LinkedHashMap&lt;String, String&gt; linkedHashMap = new LinkedHashMap&lt;&gt;();
    while (iterator.hasNext()) {
        Map.Entry&lt;byte[], byte[]&gt; next = iterator.next();
        String key = Iq80DBFactory.asString(next.getKey());
        String value = Iq80DBFactory.asString(next.getValue());
        linkedHashMap.put(key, value);
    }
    return linkedHashMap;
}
</code></pre>
<script async src="https://pagead2.googlesyndication.com/pagead/js/adsbygoogle.js"></scr
pt>
<!-- 黑客派PC帖子内嵌-展示 -->
<ins class="adsbygoogle" style="display:block" data-ad-client="ca-pub-5357405790190342"
data-ad-slot="8316640078" data-ad-format="auto" data-full-width-responsive="true"></in
>
<script>
     (adsbygoogle = window.adsbygoogle || []).push({});
</script>
<h2 id="测试插入一百万条数据">测试插入一百万条数据</h2>
<blockquote>
 <p><strong>基于 SpringBoot 搭建的控制器</strong></p>
 <p><strong>随机生成 指定数量的 UUID 并且插入到 LevelDB</strong></p>
 <p><strong>从请求 到响应 5.5 秒左右 如果再抛掉生成 UUID 的时间 可能更快 哈哈</strong></
>
</blockquote>
<pre><code class="highlight-chroma">@ResponseBody
@GetMapping("/generate")
public ResponeEntity generate(Long count) {
    DB db = levelDb.getDb();
    //创建批量处理
    WriteBatch batch = db.createWriteBatch();
    for (int i = 0; i &lt; count; i++) {
        String uuid = UUID.randomUUID().toString();
        batch.put(Iq80DBFactory.bytes(uuid), Iq80DBFactory.bytes(uuid));
    }
    //执行写入
    db.write(batch);
    return buildRespone(null);
}
</code></pre>
<p><img src="https://static.hacpai.com/images/img-loading.svg" alt="" data-src="https://i.
mgur.com/JbdgDde.jpg"></p>
<h2 id="测试从一百万数据中取出一条">测试从一百万数据中取出一条</h2>
<blockquote>
 <p>PostMan 请求到响应时间 19 毫秒</p>
</blockquote>
<pre><code class="highlight-chroma">@ResponseBody
@GetMapping("/getLevel")
public ResponeEntity getLevel(String key) {
    byte[] bytes = db.get(Iq80DBFactory.bytes(key));
    String value = Iq80DBFactory.asString(bytes);
```

```
    return buildRespone(value);
}
```
<img src="https://static.hacpai.com/images/img-loading.svg" alt="" data-src="https://i.mgur.com/acovsFQ.jpg">