

对于archive模式的全节点完整的跑一次区块大概需要14天([geth v1.9.0](#))对于重放合约获取数据该时间较长，并且geth客户端不能定点重放合约，因此需要编写geth-query重放合约工具，并且优化该工具的运行效率。

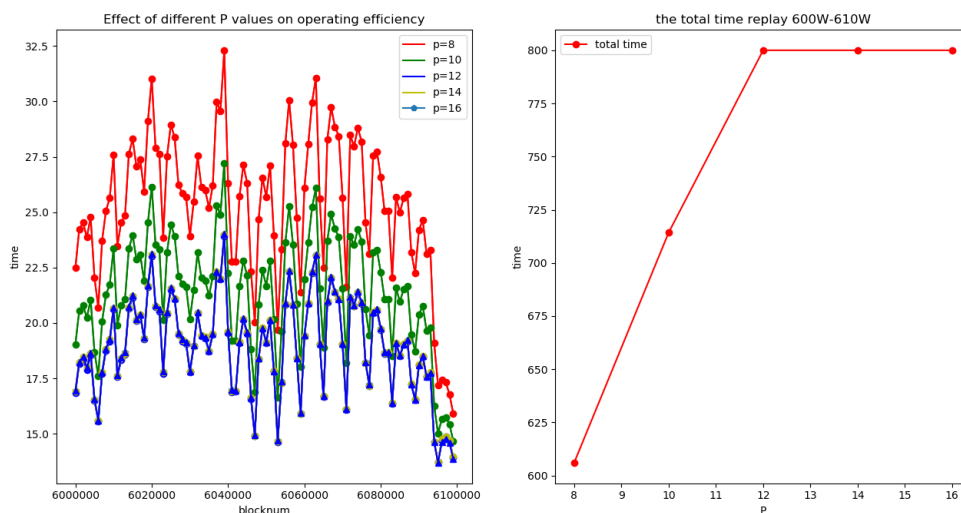
1. 对需要重放的区块数量平均分成cpu逻辑内核的份数，然后每一份启动一个协程开始重放区块数据。
2. 构建一个协程池，协程池内的协程数目为cpu的核数，然后定义一个batchBlockNum，每次向协程池中的空余协程投放batchBlockNum个block。因为每一个协程都在大量读写磁盘，这里使用协程池限制协程数量主要是因为磁盘I/O的限制。这里的运行效率主要和batchBlockNum的大小(CPU计算量的大小)以及使用逻辑核数P的值有关。

对于思路1主要是在区块数据中，前面区块数据运行较快，后面区块数据运行慢。通过平分区块数量不能达到平分计算量的效果，因此会导致运行前面区块的协程早已运行完毕，运行后面区块的协程还在继续运行，并且此时CPU的逻辑核没有用完，导致CPU利用效率不高。

对于思路2，具体分析如下：

```
# 本机环境
cpu: K9900 8核16线程
mem: 32G
SSD: 4T + 4T
```

首先将逻辑核数用满16核，三星阵列的读取速度最高位800M每秒左右。然后尝试使用12核发现三星阵列的读取速度也已经跑满，测试600W-610W的速度如下所示：



上面的图显示当P值12及12以上的时候，geth-query的执行效率提升不高，因此600W-610W的g的最优值为12

对于batchnum大小的衡量如下：

进一步工作

由于前面的block世界状态大小较小，对于磁盘读写压力较小，猜想对于前面的块较高P值能够适当提升geth-query执行前面区块的执行效率。