KVSystem性能测试文档

一、性能测试环境：

希捷 SATA 硬盘，局域网环境。

二、顺序写性能：

系统顺序写能达到 200 条/秒，单条记录大小固定 100K。

三、随机读性能：

随机读是系统的瓶颈。以下数据是在 KVS 系统已经注入 110G 数据（防止局部性）的基础上测试的，同时，QPS 高，并不一定代表单条记录响应时间少。以下测试单条记录大小固定 100K。

1.KVS Lib：71 条/秒，延迟 14ms

2.**Server--单独 Client: 23 条/秒，延迟 43ms**（该测试瓶颈在 TCP Send Buffer 为最大 2500K 的限制）

3.Server--5*Client: Server:73 条/秒，Client:15 条/秒，Client 延迟 66ms

四、测试方式：

测试代码在 test 目录下：

```
~/kvs/test# make
```

则生成 lib_put, lib_get, client_put, client_get.直接执行命令可出现使用信息。

KVS Lib 测试：

(1)向 KVS 实例写入 10000 条记录（记录大小 100K)

```
~/kvs/test$ ./lib_put /tmp/kvs.disk index_write 10000
```

(2)生成随机读请求　(实际测试中，生成 100W 记录才可测试随机读性能)

```
~/kvs/test$ while read i ; do echo "$i $RANDOM" ; done < index_write | sort -k2n | cut -d" " -f1 > index_read
```

(3)随机读 5000 条

```
~/kvs/test$ ./lib_get /tmp/kvs.disk index_read 5000
```

Server--Client 测试：

(1)启动 Server:

```
~/kvs$ ./server　8888　　/tmp/cs　C
```

(2)向 Server 写入 10000 条记录

```
~/kvs/test$ ./client_put 10000 127.0.0.1 8888 index_put
```

(3)生成随机读请求文件

```
~/kvs/test$ while read i ; do echo "$i $RANDOM" ; done < index_put | sort -k2n | cut -d" " -f1 > index_get.1
```

(4)单进程随机读 1000 条

```
~/kvs/test$ ./client_get 1000 127.0.0.1 8888 index_get.1
```

(5)多进程随机读，测试 QPS

首先，建立随机请求文件：index_get.1, index_get.2, index_get.3, index_get_4,...，然后分别同时运行./client_get 命令。

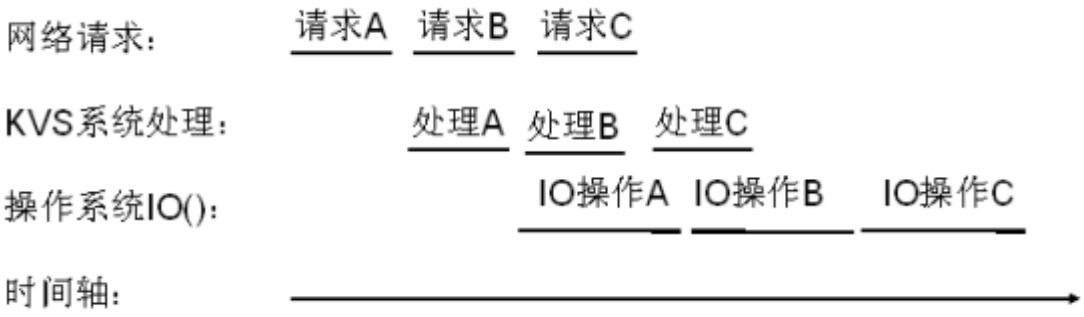
五、改进：

1.手动提高 TCP Send buffer，使 Server--单 Client 准确。(暂时没设置成功)
2.在 Client-Server 的随机读性能中，Dstat 可知，CPU 30%以上在 wait io，并且磁盘吞吐量远远没有达到最大值，可见，性能瓶颈在磁盘寻道中。猜测通过将同步改为异步能降低延迟。

KVS 系统同步处理过程，当 KVS 向操作系统调用 IO 请求时，KVS 等待 IO 返回之后，才继续执行：

因为瓶颈在磁盘寻道中，有两种改进策略：

1.尝试改为异步模式，KVS 系统向 IO 请求后，不等待其返回，直接处理新的 KVS 任务：



2.多线程：