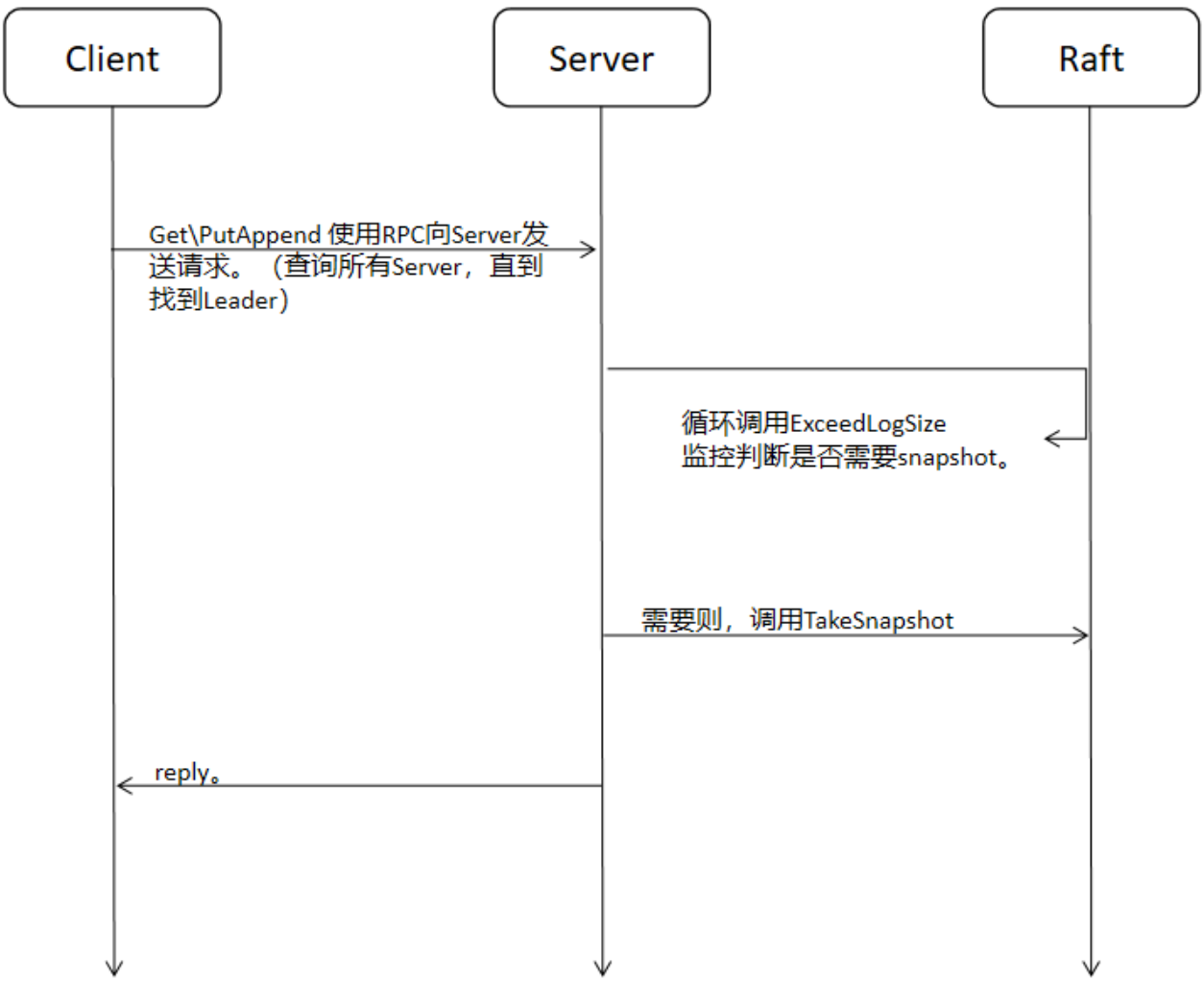
Lab3B

实验3的主要内容就是完成日志压缩。

日志压缩部分，其实要修改最多的是raft的实现，在kv上修改的较少。整体流程应该是这样的。

1、KVServer发现log size大于设定好的阈值，通知对应的Raft server discard log，并把log的snapshot传过去 2、Raft server收到KVServer的通知，截断snapshot之前的log，并通知persister保存KVServer传过来的snapshot 3、leader在发送心跳的时候如果发现有新的snapshot persist了，通知followers InstallSnapshot 4、follower 收到InstallSnapshot，与本地log进行对比，跟新log，并通知persister保存leader传过来的snapshot 5、follower通知对应的KVserver，reset kvStore保持其一致性



Client：

Client不需要进行修改。

Server：

server比较复杂，还有一些关于channel的操作，以及需要阻塞的地方，对于Go语言并不太熟悉，所以会有些挣扎。

```
type KVServer struct {
    mu      sync.Mutex
    pmu      sync.Mutex

    me      int
    rf      *raft.Raft
    applyCh chan raft.ApplyMsg
    dead    int32 // set by Kill()

    maxraftstate int // snapshot if log grows this big

    // Your definitions here.
    sequenceMapper map[int64]int64
    requestMapper  map[int]chan Op
    kvStore        map[string]string

    lastAppliedIndex int
}
```

server新添加一个lastAppliedIndex。这个是用来让raft判断Server下发的snapshot请求是否有效。如果raft判断本地中已经包含了下发的snapshot，就可以忽略这一次请求。

监控部分：

```
func (kv *KVServer) snapshotMonitor() {
    for {
        if kv.killed() || kv.maxraftstate == -1 {
            return
        }
        if kv.rf.ExceedLogSize(kv.maxraftstate) {
            //DPrintf("Need snapshot  maxraftstate:%d",kv.maxraftstate)
            //save state
            kv.mu.Lock()
            snapshot := kv.getSnapshot()
            kv.mu.Unlock()

            //tells Raft that it can discard old log entries
            if snapshot != nil {
                //DPrintf("start snapshot")
                kv.rf.TakeSnapshot(snapshot,kv.lastAppliedIndex)
            }
        }
        time.Sleep(1 * time.Millisecond)
    }
}
```

添加一个新的监控线程，专门用来查看是否需要进行snapshot。

处理部分：

处理部分就只有获得snapshot，除开日志压缩的snapshot只需要两个部分，映射表sequenceMapper和实际存储内容kvStore。

```
func (kv *KVServer) getSnapshot() []byte {
    w := new(bytes.Buffer)
    e := labgob.NewEncoder(w)
    e.Encode(kv.kvStore)
    e.Encode(kv.sequenceMapper)

    return w.Bytes()
}
```

还有就是需要注意的一点，在命令监控上也需要进行判断。

因为可能存在这种情况。那就是 Server认为需要进行snapshot了，已经进行过snapshot。但是由于网络问题，出现了新leader，同时，需要覆盖先前未提交的日志，但是由于这部分日志可能已经被压缩了，所以，这个时候就需要leader将他那份snapshot发过来进行覆盖。因此会存在snapshot更新的情况，所以要判断raft上传的命令是否为snapshot、更新。

```
if msg.Issnapshot {
DPrintf("Find snapshot!\n")
kv.updateMappersFromSnapshot(msg.Snapshot)
continue
}

func (kv *KVServer) updateMappersFromSnapshot(snapshot []byte) {
    kv.mu.Lock()
    defer kv.mu.Unlock()
    r := bytes.NewBuffer(snapshot)
    d := labgob.NewDecoder(r)
    var kvStore map[string]string
    var sequenceMapper map[int64]int64
    if d.Decode(&kvStore) == nil && d.Decode(&sequenceMapper) == nil {
        kv.kvStore, kv.sequenceMapper = kvStore, sequenceMapper
    }
}
```

因为要对raft进行修改，所以做了一次大改，并且已经将raft的主要函数和流程做了详细的pdf解释。

```
lyj@ubuntu:~/Desktop/6.824/src/kvraft$ go test -run 3A
Test: one client (3A) ...
  ... Passed --  15.3  5  1680  319
Test: many clients (3A) ...
  ... Passed --  15.8  5  3684 1499
Test: unreliable net, many clients (3A) ...
  ... Passed --  18.6  5  3280  710
Test: concurrent append to same key, unreliable (3A) ...
  ... Passed --   1.4  3   160   52
Test: progress in majority (3A) ...
  ... Passed --   0.4  5    65    2
Test: no progress in minority (3A) ...
  ... Passed --   1.1  5   194    3
Test: completion after heal (3A) ...
  ... Passed --   1.1  5    96    3
Test: partitions, one client (3A) ...
  ... Passed --  22.9  5  2608  257
Test: partitions, many clients (3A) ...
  ... Passed --  23.8  5  4448 1413
Test: restarts, one client (3A) ...
  ... Passed --  19.8  5  2017  312
Test: restarts, many clients (3A) ...
  ... Passed --  20.4  5  4585 1487
Test: unreliable net, restarts, many clients (3A) ...
  ... Passed --  21.6  5  3435  763
Test: restarts, partitions, many clients (3A) ...
  ... Passed --  27.1  5  4552 1332
Test: unreliable net, restarts, partitions, many clients (3A) ...
  ... Passed --  27.0  5  3759  533
Test: unreliable net, restarts, partitions, many clients, linearizability checks (3A) ...
  ... Passed --  25.5  7  8396 1281
PASS
ok      _/home/lyj/Desktop/6.824/src/kvraft     242.294s
```