

**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN**



**BÁO CÁO LAB 03
CONSENSUS RAFT ALGORITHM**

THÀNH VIÊN NHÓM: Nguyễn Thiên Phúc

GVHD: TS. Ngô Huy Biên
ThS. Ngô Ngọc Đăng Khoa

NĂM HỌC 2024 – 2025

MỤC LỤC

I.	Thông tin sinh viên.....	1
II.	Cấu trúc dự án.....	1
1.	Yêu cầu 1: Chạy lại source code part 1 và báo cáo kết quả.....	1

I. Thông tin sinh viên

- 20127681 – Nguyễn Thiên Phúc

II. Cấu trúc dự án

1. Yêu cầu 1: Chạy lại source code part 1 và báo cáo kết quả

- Source code của part 1 thực hiện test case TestElectionFollowerComesBack, Test này kiểm tra cơ chế bầu cử và xử lý nút bị ngắt kết nối rồi quay lại trong thuật toán **Raft**.

Tổng quan về log test

- Ba nút khởi động: 0, 1, 2, mỗi nút mở cổng lắng nghe.
- Tất cả các nút bắt đầu bộ đếm bầu cử.
- Nút 1 thắng cử, trở thành Leader.
- Leader gửi AppendEntries để duy trì quyền lãnh đạo.
- Nút 2 bị ngắt kết nối (Disconnect).
- Sau một thời gian, nút 2 quay lại (Reconnect) nhưng nhận ra nhiệm kỳ đã lỗi thời.
- Nút 2 tự đề cử lại và thắng cử.
- Cụm ổn định với nút 2 làm Leader.
- Tất cả các nút được tắt để kết thúc test.
- Test hoàn thành thành công (PASS).

Các bước thực hiện chạy code

- Bước 1: Thực hiện clone repo gốc của tác giả tại link github: <https://github.com/eliben/raft>
- Bước 2: cd part1 và chạy lệnh go test -v -race -run TestElectionFollowerComesBack |& tee /tmp/raftlog để build part 1
- Bước 3: Kiểm tra file log sau khi chạy part 1

Phân tích các bước trong log

1. Khởi tạo cụm với ba nút

```
[0] election timer started (286ms), term=0
[1] election timer started (173ms), term=0
[2] election timer started (199ms), term=0
```

- Mỗi nút **bắt đầu lắng nghe** trên các cổng tương ứng.
 - Mỗi nút khởi động **bộ đếm thời gian bầu cử** với thời gian ngẫu nhiên.
2. Bắt đầu bầu cử

```
[1] becomes Candidate (currentTerm=1); log=[]
[1] election timer started (266ms), term=1
[1] sending RequestVote to 2: {Term:1 CandidateId:1 LastLogIndex:0 LastLogTerm:0}
[1] sending RequestVote to 0: {Term:1 CandidateId:1 LastLogIndex:0 LastLogTerm:0}
```

- Nút 1 không nhận được AppendEntries từ leader (vì chưa có leader), nên **nâng lên Candidate**.
 - Gửi yêu cầu bỏ phiếu (RequestVote) đến nút 0 và 2 với term = 1.
3. Nút 1 thắng cử

```
[1] wins election with 2 votes
[1] becomes Leader; term=1, log=[]
```

- **Nhận được đủ phiếu bầu (2/3)**, nút 1 trở thành **Leader**.
4. Leader gửi heartbeat đến follower

```
[1] sending AppendEntries to 0: ni=0, args={Term:1 LeaderId:1 PrevLogIndex:0 PrevLogTerm:0 Entries:[] LeaderCommit:0}
[1] sending AppendEntries to 2: ni=0, args={Term:1 LeaderId:1 PrevLogIndex:0 PrevLogTerm:0 Entries:[] LeaderCommit:0}
```

- Leader gửi AppendEntries đến tất cả follower để duy trì quyền kiểm soát
5. Mô phỏng sự cố đứt mạng, nút 2 bị ngắt kết nối

```
[1] sending AppendEntries to 2: ni=0, args={Term:1 LeaderId:1 PrevLogIndex:0 PrevLogTerm:0 Entries:[] LeaderCommit:0}
[0] AppendEntries: {Term:1 LeaderId:1 PrevLogIndex:0 PrevLogTerm:0 Entries:[] LeaderCommit:0}
[0] AppendEntries reply: {Term:1 Success:true}
[2] becomes Candidate (currentTerm=2); log=[]
```

- Nút 1 vẫn gửi AppendEntries cho 2, nhưng nút 2 không phản hồi vì đã bị ngắt kết nối
6. Nút 2 bị ngắt kết nối và không nhận được heartbeat từ leader
- Sau khi bị ngắt kết nối, nút 2 không nhận được heartbeat từ leader và tự đề cử mình lên candidate và tự bầu cử, nhưng do không có kết nối và không nhận được phiếu vote từ các node khác đủ số lượng nên nút 2 không trở thành leader được.

Quá trình này lặp đi lặp lại cho đến khi term = 4 thì hệ thống có kết nối lại

```
[2] becomes Candidate (currentTerm=4); log=[]
[2] sending RequestVote to 1: {Term:4 CandidateId:2 LastLogIndex:0 LastLogTerm:0}
[2] sending RequestVote to 0: {Term:4 CandidateId:2 LastLogIndex:0 LastLogTerm:0}
[2] election timer started (244ms), term=4
[1] RequestVote: {Term:4 CandidateId:2 LastLogIndex:0 LastLogTerm:0} [currentTerm=3, votedFor=-1]
[1] ... term out of date in RequestVote
[1] becomes Follower with term=4; log=[]
[1] ... RequestVote reply: {Term:4 VoteGranted:true}
[1] election timer started (182ms), term=4
[2] received RequestVoteReply {Term:4 VoteGranted:true}
[2] wins election with 2 votes
[2] becomes Leader; term=4, log=[]
```

- Nút 2 kết nối lại nhưng phát hiện nhiệm kỳ hiện tại đã lỗi thời.
- Tự nâng lên Candidate và gửi yêu cầu bầu cử (RequestVote) đến nút 0 và 1.
- Nút 2 nhận được đa số phiếu bầu (2/3) và trở thành leader mới

- Sau đó nút 2 gửi heartbeat đến các follower

```
[1] AppendEntries: {Term:4 LeaderId:2 PrevLogIndex:0 PrevLogTerm:0 Entries:[] LeaderCommit:0}
[1] AppendEntries reply: {Term:4 Success:true}
[0] AppendEntries: {Term:4 LeaderId:2 PrevLogIndex:0 PrevLogTerm:0 Entries:[] LeaderCommit:0}
[0] AppendEntries reply: {Term:4 Success:true}
```

- Sau đó tắt các nút đi để kết thúc test

Nhận xét:

Test này kiểm tra hai tình huống quan trọng trong Raft:

1. Quá trình bầu cử ban đầu: Nút 1 thắng cử và trở thành leader.
2. Xử lý follower bị mất kết nối rồi quay lại:
 - Nút 2 bị ngắt kết nối.
 - Khi quay lại, phát hiện nhiệm kỳ lỗi thời và tự đề cử.
 - Nút 2 thắng cử, thay thế leader cũ.