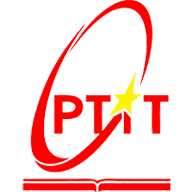
**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO THỰC TẬP**

**CÔNG TY VCCORP CORPORATION**

**ĐỀ TÀI:**

**GIẢI THUẬT ĐỒNG THUẬN TRONG HỆ THỐNG PHÂN TÁN**

**GIẢNG VIÊN HƯỚNG DẪN: PHAN THỊ HÀ**

**CỐ VẤN THỰC TẬP: NGÔ VĂN VĨ**

TÊN SINH VIÊN: TRẦN NGUYÊN THỊNH

MÃ SV: B18DCCN645 **LỜI CẢM ƠN**

Lời đầu tiên em xin gửi lời cảm ơn chân thành nhất tới cô Phan Thị Hà. Trong quá trình thực tập và làm việc tại VCCorp Corporation đã rất thoải mái và quan tâm tới bọn em trong suốt sáu tuần vừa qua. Cuối cùng sau hơn một tháng thực tập em đã hoàn thành xong chương trình thực tập tốt nghiệp trên. Xin cảm ơn cô rất nhiều!

Tiếp theo, em gửi lời cảm ơn sâu sắc tới anh Ngô Văn Vĩ – người trực tiếp cho đề tài và giúp đỡ bọn em trong quá trình vừa qua. Thông qua tiểu luận này, em xin trình bày lại tất cả những điều mà em đã tìm hiểu được trong suốt thời gian vừa qua

1. **BÁO CÁO NỘI DUNG THỰC TẬP**
2. **Tổng quan.**
3. **Vị trí thực tập:** Thực tập sinh Java Backend
4. **Mục tiêu:**

* Tìm hiểu giải thuật đồng thuận bè trong hệ thống phân tán
* Tạo được ứng dụng demo cho giải thuật
* Tìm hiểu thêm một số kiến thức trong lập trình web nói chung và spring boot nói riêng như: Message Broker (Kafka, RabbitMQ)
* Hiểu rõ về định hướng nghề nghiệp của bản thân
* Tích luỹ thêm kinh nghiệm thực tế.

1. **Đề tài được giao.**
2. **Tìm hiểu về giải thuật đồng thuận raft trong hệ thống phân tán.**

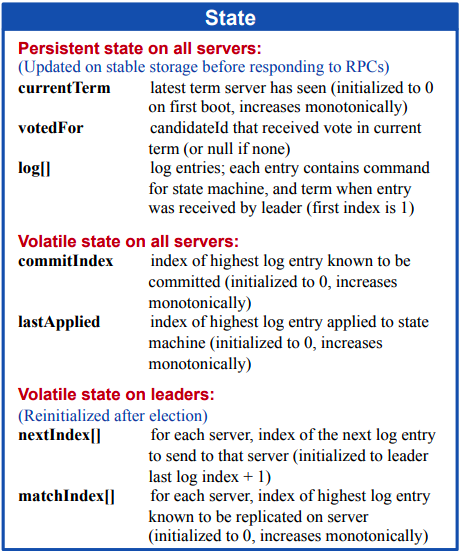
Các thuật toán đồng thuận là một cơ chế cho phép một cụm nhiều máy tính (server) hoạt động mạch lạc, suôn sẻ, dù cho trong đó có một vài server đang bị ngắt kết nối với cụm. Do đó, các thuật toán đồng thuận luôn đóng một vai trò quan trọng trong việc xây dựng các hệ thống tính toán quy mô lớn đáng tin cậy.

Trong đó Raft là một thuật toán đồng thuận hoạt động tương đối hiệu quả, có những ưu điểm đáng kể so với các thuật toán đồng thuận khác, nó có 3 đặc tính quan trọng sau:

* + **Stronger Leader** (vai trò của Leader mạnh hơn): Raft cho phép Leader có vai trò mạnh mẽ hơn các thuật toán đồng thuận khác. Ví dụ, các log entries chỉ được broadcast từ Leader đến các server khác. Điều này giúp đơn giản hóa quản lý việc replicate các log entries giữa các server với nhau và giúp Raft dễ hiểu hơn (so với thuật toán Paxos của Leslie Lamport).
  + **Leader Election** ( việc bầu cử Leader): Raft sử dụng một bộ đếm giờ ngẫu nhiên để bầu leader, đó đơn giản là thêm một thay đổi nhỏ vào heartbeats so với các thuật toán đồng thuận khác, nhưng nó lại giúp giải quyết conflict giữa các server trong cụm một cách đơn giản và nhanh chóng.
  + **Membership changes** (thay đổi vai trò của server trong cụm): nghĩa là thay đổi vai trò của một server nào đó từ follower thành leader và ngược lại, cơ chế mà Raft sử cho việc thay đổi này là sử dụng một phương pháp đồng thuận chung mới (joint consensus), trong quá trình thay đổi này vai trò của một server sẽ là sự overlap giữa leader và follower . Điều này cho phép tổng thể cụm vẫn tiếp tục hoạt động bình thường trong khi các server trong cụm đang trong quá trình thay đổi vai trò.

**Thuật toán đồng thuận Raft**

* Raft là một thuật toán để quản lý một replicated log
* Raft thực hiện sự đồng thuận bằng cách trước tiên bầu một leader, sau đó giao cho leader hoàn toàn chịu trách nhiệm quản lý replicated log.
* Leader sẽ nhận các log entry từ client, replicate chúng cho các server khác và thông báo cho các server đó khi nào an toàn để apply chúng vào state machine của mỗi server. Việc có một leader sẽ đơn giản hóa quản lý replicated log.
* Leader có thể quyết định nơi đặt các entries mới trong log mà không cần tham khảo các server khác và luồng dữ liệu cũng được gửi theo cách đơn giản từ leader đến các server khác. Một leader có thể fail hoặc bị ngắt kết nối, trong trường hợp đó, một leader mới sẽ được bầu.
* Theo cách tiếp cận bằng khía cạnh leader, Raft phân tách vấn đề đồng thuận thành ba bài toán con tương đối độc lập:
* Leader election (bầu chọn leader): một leader mới phải được bầu khi leader hiện tại bị fail
* Log replication (sao chép log): leader phải nhận các log entries từ client và replicate chúng trên toàn cụm, buộc các server khác phải chấp nhận các entries đó vào log riêng của chúng.
* Safety: tính an toàn của Raft chính là tính an toàn của state machine, nếu bất kỳ server nào đã apply một log entry nào đó cho state machine của nó, thì sẽ không có server nào khác có thể apply một lệnh khác cho cùng log entry đó.
* Các khái niệm trong raft:



Giải thích:

Nhìn vào hình thì ta có thể tháy có 2 loại state trong Raft đó là :

* **Persistent state:** State này sẽ được update trên stable storage trước khi server đó phản hồi cho RPCs
* **Volatile state:** Trong đó, có những thuộc tính chung cho tất cả máy tính, có những thuộc tính riêng mà chỉ leader mới có.

Tóm tắt lại thì server không phải leader và server là leader sẽ có những thuộc tính sau:

|  |  |
| --- | --- |
| non-leader | leader |
| - currentTerm  - votedFor  - log[]  - commitIndex  - lastApplied | - currentTerm  - votedFor  - log[]  - commitIndex  - lastApplied  - nextIndex[]  - matchIndex[] |

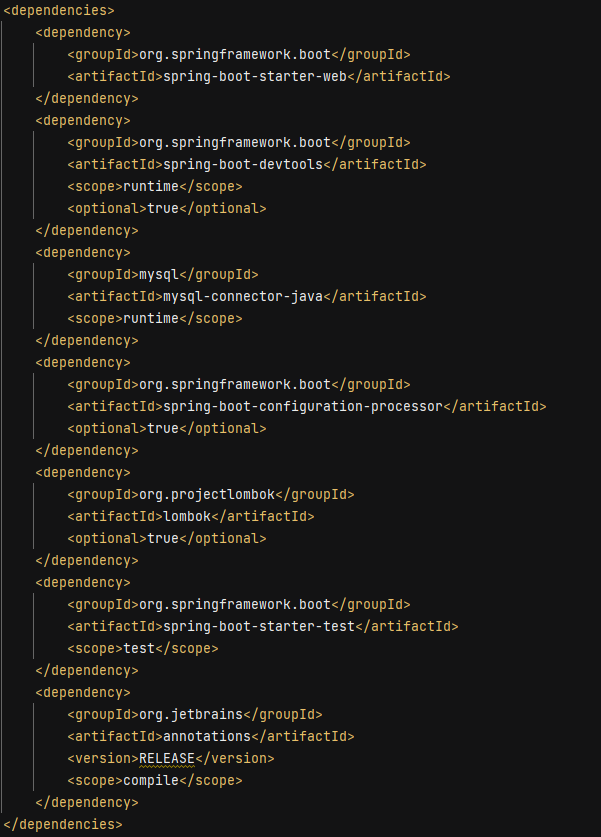
Trong đó:

* + **currentTerm:** nhiệm kỳ mới nhất, khi bắt đầu hệ thống Raft sẽ được khởi tạo là 0, bước tăng là 1
  + **votedFor:** Id của server mà đã được server này vote cho ở nhiệm kỳ hiện tại (currentTerm)
  + **log[]:** log entries; mỗi entry chứa lệnh cho state machine và term mà leader nhận được entry đó (đầu tiên là 1). Vậy log[] là một mảng các item, mỗi item sẽ có 2 thuộc tính là entry và term mà leader nhận được entry đó.
  + **commitIndex:** index của log entry cao nhất đã được committed (khởi tạo từ 0, bước tăng 1)
  + **lastApplied:** index của log entry cao nhất đã được apply vào state machine (khởi tạo từ 0, bước tăng 1)
  + **nextIndex[]**: (thuộc tính riêng của leader) đối với mỗi server, index của log entry tiếp theo sẽ được leader gửi đến server đó (khởi tạo bằng log index của leader + 1 )
  + **matchIndex[]:** đối với mỗi server, index của log entry cao nhất đã được replicated trên server đó (được khởi là 0, bước tăng 1)
* Hiện thì trong vòng 6 tuần vừa qua em đã tìm hiểu được những kiến thức trên với giải thuật đồng thuận raft

1. **Tạo ứng dụng demo cho giải thuật đồng thuận trong hệ thống phân tán.**

Mặc dù được đánh giá là dễ hiểu và triển khai hơn thuật toán Paxos của tác giả Leslie nhưng thuật toán đồng thuận raft vẫn tương đối trừu tượng nên trong demo lần này em sẽ sử dụng thuật toán Paxos để triển khai giải thuật đồng thuận.

Thư viện sử dụng:



Link demo:

[thinh147/consensus-distributed (github.com)](https://github.com/thinh147/consensus-distributed)

Tham khảo:

<https://marcoserafini.github.io/papers/zab.pdf>

<http://www.tcs.hut.fi/Studies/T-79.5001/reports/2012-deSouzaMedeiros.pdf>

<https://raft.github.io/raft.pdf>

<https://web.stanford.edu/~ouster/cgi-bin/papers/raft-atc14>

<https://thesecretlivesofdata.com/rafts>

Các mốc thời gian:

04/07/2022:  Đến công ty nhận bàn giao công việc thực tập.

05/07/2022: Được giao đề tài.

06/07/2022 - 23/07/2022: Tự tìm hiểu lý thuyết

24/07/2022: Báo cáo lý thuyết, bắt đầu triển khai project

10/08/2022: Báo cáo kết quả demo, nhận xét, chỉnh sửa code

14/08/2022: Hoàn thiện code, tài liệu.

1. **ĐÁNH GIÁ TỔNG QUAN CÁ NHÂN**
2. **Tự đánh giá.**

Trong suốt sáu tuần vừa qua em đã được học thêm về phần giải thuật của các hệ thống phân tán, qua đó hiểu thêm được rằng những thứ được dạy trên trường đều được áp dụng rất nhiều ở trong thực tế.

1. **Thuận lợi và khó khăn**

Thuận lợi:

* Anh leader Ngô Văn Vĩ vui vẻ nhiệt tình, support tận tình.
* Hình thức thực tập remote – linh động cho sinh viên học lại – học cải thiện.
* Tìm hiểu kỹ hơn về những thuật toán trong hệ thống phân tán
* Biết được thêm những kiến thức với lập trình backend nói chung và framework Spring-boot nói chung

Khó khăn:

* Với các giải thuật của hệ thống phân tán không phải công ty nào cũng có nên bước đầu đọc còn khá khó hiểu và cần thời gian để tìm hiểu thêm
* Tài liệu chủ yếu bằng tiếng anh nên việc đọc tài liệu còn khá mất thời gian

1. **KẾT LUẬN**

Qua quá trình thực tập vừa qua em đã được làm quen với những người mới, được làm thử các công nghệ chưa từng sử dụng. Từ đó có thể tìm thấy nguồn cảm hứng của bản thân để sau ra trường có thể định hướng được công việc tốt hơn.

Và cũng qua kỳ thực tập này, em nhận ra mình còn nhiều thiếu sót điển hình như việc đọc – nói tiếng anh chưa tốt lắm 🡪 mất thời gian khi được giao đọc tài liệu tiếng anh. Em nghĩ rằng thời gian thực tập vừa qua tuy ngắn nhưng đủ để em nhận ra được nhiều thứ trước khi ra trường.