ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH

**ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**KHOA MẠNG MÁY TÍNH VÀ TRUYỀN THÔNG**

Logo, company name

Description automatically generated

**BÁO CÁO ĐỒ ÁN CHUYÊN NGÀNH**

**Đánh giá hiệu suất các cơ chế đồng thuận trong Blockchain**

Performance Evaluation of Consensus on Blockchain

**Giáo viên hướng dẫn: Th.S Nguyễn Khánh Thuật**

**Sinh viên thực hiện: Trần Đình Huy – 21522167**

**Lớp: NT114.P11.ATCL**

**Thành phố Hồ Chí Minh, tháng 1 năm 202****5**

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH

**ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**KHOA MẠNG MÁY TÍNH VÀ TRUYỀN THÔNG**

Logo, company name

Description automatically generated

**BÁO CÁO ĐỒ ÁN CHUYÊN NGÀNH**

**Đánh giá hiệu suất các cơ chế đồng thuận trong Blockchain**

**Performance Evaluation of Consensus on Blockchain**

**Giáo viên hướng dẫn: Th.S Nguyễn Khánh Thuật**

**Sinh viên thực hiện: Trần Đình Huy – 21522167**

**Lớp: NT114.P11.ATCL**

**Thành phố Hồ Chí Minh, tháng 1 năm 2025**

LỜI CẢM ƠN

Đầu tiên, nhóm chúng em xin trân trọng gửi lời cảm ơn đến Ban giám hiệu Trường Đại học Công nghệ Thông tin – Đại học Gia Thành Phố Hồ Chí Minh đã tạo điều kiện về cơ sở vật chất, môi trường học thuật thuận lợi cho quá trính học tập và nghiên cứu của chúng em.

Hơn thế nữa, nhóm chúng em xin chân thành cảm ơn quý thầy cô trường Đại Học Công Nghệ Thông Tin, đặc biệt là quý thầy cô khoa Mạng Máy Tính và Truyền Thông, và lời cảm ơn sâu sắc đến thầy Nguyễn Khánh Thuật – người hướng dẫn hết sức tận tâm, nhiệt tình, luôn luôn động viên, tạo mọi điều kiện tốt nhất cho chúng em khi thực hiện Đồ án chuyên ngành tại Trường ĐHCNTT – ĐHQGHCM. Những ý kiến đóng góp, lời chỉ dẫn của thầy luôn là nguồn cảm hứng và định hướng cho chúng em ngày càng hoàn thiện, phát triển hướng nghiên cứu ngày một tốt hơn. Đồng thời, suốt khoảng thời gian làm việc cùng với các thầy cô cũng là cơ hội tuyệt vời để chúng em ngày càng hoàn thiện bản thân hơn trong giai đoạn hội nhập và phát triển của đất nước.

Trong quá trình thực hiện Đồ án chuyên ngành và viết bài báo cáo, chúng em nhận ra bản thân còn nhiều hạn chế về kiến thức chuyên ngành và trình độ lý luận, dù đã cố hết sức nhưng có lẽ là vẫn khó tránh khỏi những sai sót trong thời gian nghiên cứu và báo cáo. Vì vậy, chúng em rất mong được quý thầy cô chỉ bảo, góp ý để chúng em có thể hoàn thiện hơn về mặt kiến thức, cũng như tích lũy thêm nhiều kinh nghiệm thực tiễn hữu ích cho tương lai sau này.

Cuối cùng, chúng em xin kính chúc quý thầy cô của trường Đại Học Công Nghệ Thông Tin, quý thầy cô khoa Mạng Máy Tính và Truyền Thông, và đặc biệt là thầy Thuật có được nhiều sức khỏe để có thể dốc lòng trong sự nghiệp “trồng người”.

**Thành phố Hồ Chí Minh, tháng 1 năm 2025**

**Trần Đình Huy – 21522167 – ATCL2021**

**MỤC LỤC**

[LỜI CẢM ƠN 3](#_Toc188614270)

[DANH MỤC HÌNH ẢNH 5](#_Toc188614271)

[DANH MỤC BẢNG 6](#_Toc188614272)

[DANH MỤC VIẾT TẮT VÀ GIẢI NGHĨA 7](#_Toc188614273)

[CHƯƠNG I: TỔNG QUAN ĐỀ TÀI 8](#_Toc188614274)

[1. Ngữ cảnh 8](#_Toc188614275)

[2. Hướng giải quyết đề tài 9](#_Toc188614276)

[CHƯƠNG II: KIẾN THỨC NỀN TẢNG 10](#_Toc188614277)

[1. Hyperledger Fabric 10](#_Toc188614278)

[2. Cơ chế đồng thuận Raft 10](#_Toc188614279)

[a. Cấu trúc của Raft 11](#_Toc188614280)

[b. Các giai đoạn hoạt động của Raft 11](#_Toc188614281)

[CHƯƠNG III: PHƯƠNG PHÁP THỰC HIỆN – TRIỂN KHAI 12](#_Toc188614282)

[1. Xây dựng kịch bản 12](#_Toc188614283)

[2. Triển khai 12](#_Toc188614284)

[a. Cấu trúc 12](#_Toc188614285)

[b. Môi trường 12](#_Toc188614286)

[c. Cấu hình 13](#_Toc188614287)

[CHƯƠNG IV: KẾT QUẢ THỰC NGHIỆM 17](#_Toc188614288)

[1. Kết quả 17](#_Toc188614289)

[2. Link mã nguồn 21](#_Toc188614290)

[CHƯƠNG IV: KẾT LUẬN 22](#_Toc188614291)

[TÀI LIỆU THAM KHẢO 23](#_Toc188614292)

DANH MỤC HÌNH ẢNH

[Hình 1: Ý tưởng cơ bản về quy trình làm việc của Hyperledger Fabric 8](#_Toc138549309)

[Hình 2: Tổng quan Raft 10](#_Toc138549310)

[Hình 3: fabric-sample 13](#_Toc138549311)

[Hình 4: configtx.yaml cho 5 node 14](#_Toc138549312)

Hình 5: crypto-config.yaml cho 5 node. 14

Hình 6: Cấu trúc folder triển khai cho các trường hợp 15

Hình 7: Cấu trúc folder triển khai kịch bản cụ thể 16

Hình 8: Kết quả cho 1 raft\_metrics.xlsx. 17

Hình 9: Mức độ sử dụng CPU giữa các mô hình node với nhau. 18

Hình 10: Thời gian bầu cử giữa mô hình node với nhau. 18

Hình 11: Thời gian bầu cử giữa mô hình node với nhau. 19

Hình 12: Network In giữa mô hình node với nhau. 19

Hình 13: Network Out giữa mô hình node với nhau. 20

DANH MỤC BẢNG

Bảng 1. Giá trị thông số ở “raft\_metrics\_3\_node.xlsx” 17

Bảng 2: Giá trị thông số ở “raft\_metrics\_5\_node.xlsx” 17

Bảng 3: Giá trị thông số ở “raft\_metrics\_7\_node.xlsx” 17

Bảng 4: Giá trị thông số ở “raft\_metrics\_9\_node.xlsx” 18

Bảng 5: Giá trị thông số ở “raft\_metrics\_11\_node.xlsx” 18

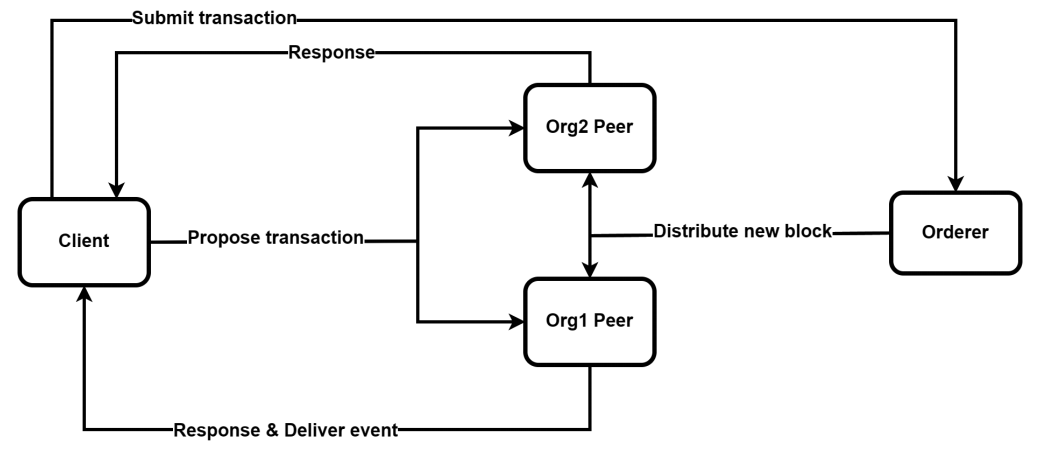
DANH MỤC VIẾT TẮT VÀ GIẢI NGHĨA

|  |  |
| --- | --- |
| **Từ viết tắt** | **Nghĩa** |
| Blockchain | Chuỗi khổi |
| Node | Nút |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

CHƯƠNG I: TỔNG QUAN ĐỀ TÀI

* 1. Ngữ cảnh

- Blockchain là một hệ thống được xây dựng dựa trên kiến trúc phi tập trung, trong đó việc duy trì tính nhất quán và toàn vẹn của dữ liệu giữa các node yêu cầu sử dụng các cơ chế đồng thuận hiệu quả.



Hình 1: Ý tưởng cơ bản về quy trình làm việc của Hyperledger Fabric

- Các dịch vụ Blockchain-based trên nền tảng đám mây, chẳng hạn như Microsoft Azure và Amazon Web Services (AWS), được gọi chung là BaaS (Blockchain-as-a-Service). Mô hình BaaS cho phép chỉ các node được cấp quyền mới có thể tham gia vào hệ thống blockchain, qua đó cải thiện đáng kể hiệu suất hệ thống về thông lượng (throughput) và độ trễ (latency) so với các hệ thống blockchain công khai. Điều này đạt được nhờ vào việc giới hạn chỉ những node đáng tin cậy tham gia vào hệ thống.

- Một trong những nền tảng BaaS phổ biến nhất hiện nay là Hyperledger Fabric, một dự án mã nguồn mở do Linux Foundation phát triển. Hyperledger Fabric hỗ trợ việc lựa chọn và tích hợp nhiều cơ chế đồng thuận khác nhau tùy theo nhu cầu cụ thể của hệ thống. Các thuật toán đồng thuận điển hình bao gồm Kafka, Raft, và Practical Byzantine Fault Tolerance (PBFT), mỗi thuật toán có cách hoạt động riêng để đảm bảo tính toàn vẹn dữ liệu và độ tin cậy của hệ thống Hyperledger Fabric.

* 1. Hướng giải quyết đề tài

- Mặc dù nhiều nghiên cứu đã được thực hiện nhằm đánh giá hiệu năng của các cơ chế đồng thuận, vẫn tồn tại các hạn chế như chưa đánh giá đầy đủ về tiêu thụ tài nguyên, thiếu các phân tích toàn diện về tham số cấu hình của BaaS của các thuật toán đồng thuận.

- Đồ án tham khảo nhiều phướng thức đánh giá hiệu suất khác nhau để giải quyết vấn đề này. Qua đó, nhóm chúng em lựa chọn đánh giá cơ chế đồng thuận Raft ở Blockchain, thu thập các số liệu để đánh giá và biểu diễn ở dạng biểu đồ để tăng tính trực quan.

CHƯƠNG II: KIẾN THỨC NỀN TẢNG

1. Hyperledger Fabric

- Hyperledger Fabric là một nền tảng blockchain mã nguồn mở được phát triển bởi Linux Foundation, được thiết kế đặc biệt cho các ứng dụng doanh nghiệp. Đây là một trong những dự án nổi bật trong hệ sinh thái Hyperledger với khả năng tùy chỉnh linh hoạt và hiệu suất cao, cung cấp nhiều tính năng bao gồm:

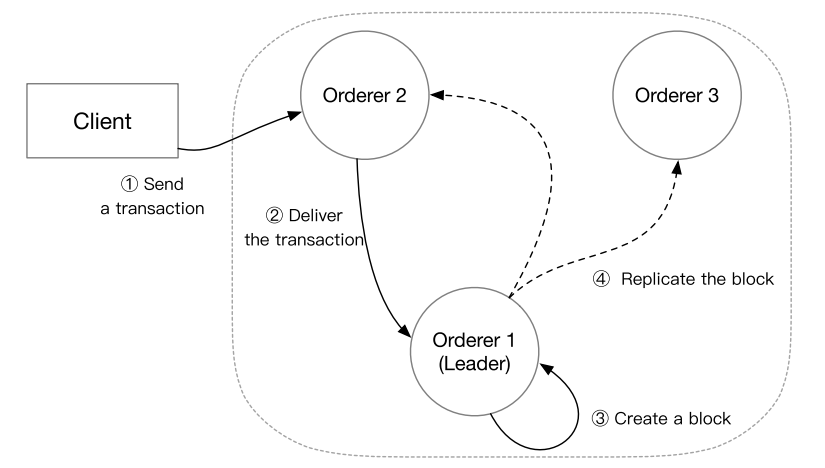
+ Kiến trúc mô-đun: Cho phép tích hợp và lựa chọn các thành phần theo nhu cầu, như cơ chế đồng thuận, quản lý nhận dạng, và hợp đồng thông minh (smart contract).

+ Hệ thống quản lý quyền: Chỉ những nút (nodes) được cấp quyền mới có thể tham gia mạng, giúp tăng cường bảo mật và cải thiện hiệu suất.

+ Hỗ trợ đa dạng cơ chế đồng thuận: Hyperledger Fabric tích hợp nhiều thuật toán đồng thuận như Raft, Kafka, và Practical Byzantine Fault Tolerance (PBFT), giúp đảm bảo tính nhất quán và tin cậy của hệ thống.

1. Cơ chế đồng thuận Raft

- Raft là một thuật toán đồng thuận được thiết kế dễ hiểu nhằm đảm bảo tính nhất quán trong các hệ thống phân tán. Cơ chế Raft đảm bảo rằng các nút (nodes) trong hệ thống phân tán luôn đồng thuận về thứ tự của các giao dịch, kể cả khi một số nút gặp lỗi.



Hình 2: Tổng quan Raft

1. Cấu trúc của Raft

- Raft tổ chức các nút trong hệ thống thành ba vai trò chính:

* Leader: Là nút trung tâm, chịu trách nhiệm nhận giao dịch, tạo log, và sao chép dữ liệu đến các nút khác (followers).
* Follower: Lắng nghe và tuân theo các chỉ thị của leader, đồng thời lưu trữ bản sao dữ liệu để đảm bảo tính đồng nhất.
* Candidate: Khi leader không khả dụng, follower có thể chuyển sang trạng thái candidate để tham gia vào quá trình bầu chọn leader mới.

1. Các giai đoạn hoạt động của Raft

- Các giai đoạn hoạt động chính của Raft bao gồm như dưới.

**Bầu chọn leader:**

* Khởi động: Khi một leader gặp lỗi hoặc không gửi tín hiệu (heartbeat) trong khoảng thời gian nhất định, follower sẽ bắt đầu quy trình bầu chọn
* Bầu chọn: Một follower chuyển sang trạng thái candidate và gửi yêu cầu bầu chọn đến các nút khác. Mỗi nút sẽ bỏ phiếu cho candidate mà nó nhận được yêu cầu đầu tiên. Candidate nhận được hơn một nửa số phiếu sẽ trở thành leader mới.

**Heartbeat:** Leader gửi các thông báo định kỳ (heartbeat) đến các follower để đảm bảo tính nhất quán trong mạng.

**Sao chép log:** Các lệnh ghi được gửi từ leader đến các follower và đồng bộ hóa chúng.

CHƯƠNG III: PHƯƠNG PHÁP THỰC HIỆN – TRIỂN KHAI

* 1. Xây dựng kịch bản

- Kịch bản: bầu chọn leader được xây dựng dựa trên giao thức Raft, trong đó các nút trong mạng (nodes) thực hiện bầu chọn để chọn ra một leader khi không có leader rõ ràng hoặc khi leader cũ ngừng hoạt động. Cụ thể như sau:

+ Timeout: thời gian ngẫu nghiên (từ 0.15 - 0.3 giây)

+ Bắt đầu bầu chọn: Khi hết Timeout, node Follower sẽ chuyển trạng thái sang Candidate và bắt đầu Vote đến các node còn lại. Node nhận được yêu cầu sẽ quyết định bầu chọn.

+ Nhận được Vote: Các node sẽ bầu cho Candidate nếu chúng đang ở trạng thái Follower. Nếu đủ trên số vote cần thiết (1 vote cho chính nó) thì nó sẽ trở thành Leader.

+ Leader thực hiện nhiệm vụ: gửi heartbeat và log entries đến các Follower, đảm bảo sự đồng bộ của mạng

- Thực hiện với sơ đồ 3 node, 5 node, 7 node, 9 node, 11 node.

- Tiêu chí đánh giá Leader:

+ Số Vote (trên 50% số node)

+ Trạng thái các node: node sẽ chuyển từ Follower sang Candidate khi không nhận được heartbeat từ Leader trong thời gian Timeout. Nếu nhận đủ Vote nó sẽ thành Leader.

- Ghi lại các thông số thời gian tái bầu cử, CPU sử dụng vào file các raft\_metrics.xlsx.

* 1. Triển khai

1. Cấu trúc

- Sử dụng Hyperledger Fabric để tạo ra môi trường, trong đó một node là leader và các node còn lại là follower.

**-** Cấu hình các tham số cần thiết để triển khai thuật toán Raft, bao gồm thời gian bầu chọn, thời gian timeout, và các tham số khác của thuật toán Raft.

1. Môi trường

- Xử dụng VMWare Workstation 15

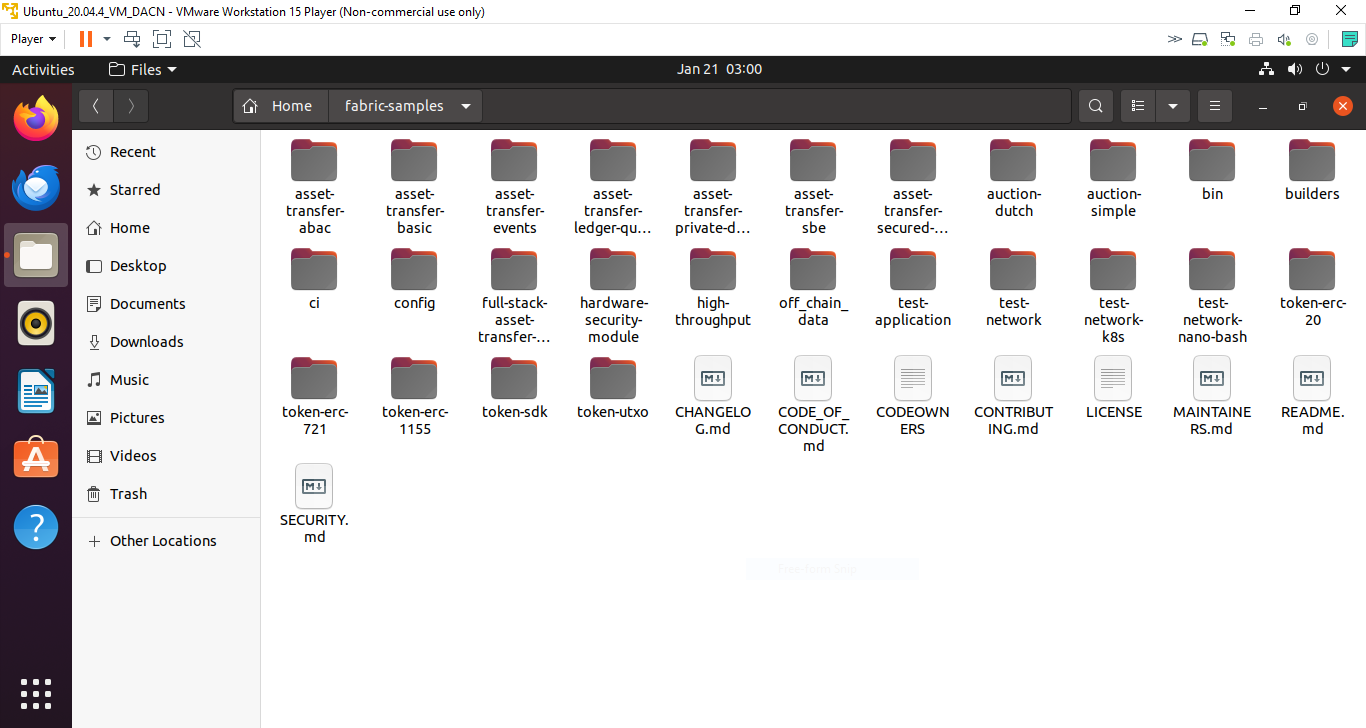
- Ubuntu-deskstop: Ubuntu 20.04 (CPU: 4, RAM 8GB, Ổ cứng: 70GB)

- Phần mềm quyết định: Docker và Docker - Compose

- Các tệp thực thi Fabric: Fabric and Fabric-samples

* + 1. Cấu hình

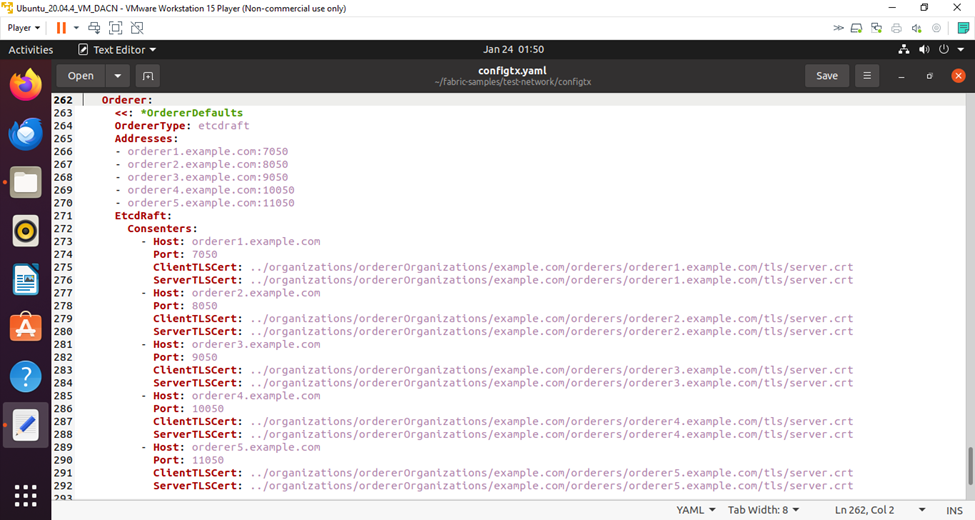
- Cài đặt Docker, Docker Compose và thiết lập môi trường



Hình 3: fabric-sample

- Cấu hình các file configtx/configtx.yaml, crypto-config.yaml tại folder test-network:

+ Cấu hình configtx.yaml với số lượng node triễn khai



Hình 4: configtx.yaml cho 5 node

+ Cấu hình crypto-config.yaml với số lượng node triễn khai



Hình 5: crypto-config.yaml cho 5 node

- Ở test-network, nhóm khởi tạo docker-compose-order để thực hiện việc triển khai và kịch bản đánh giá

- Cấu trúc folder chạy kịch bản đánh giá:

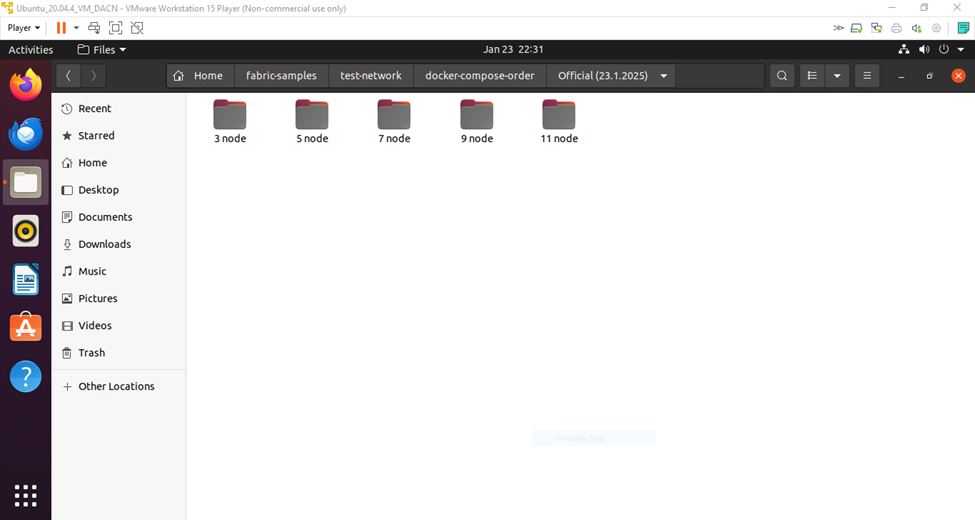
+ docker-compose.yml: tệp cấu hình Docker Compose, định nghĩa các dịch vụ và các container Docker cần chạy

+ Dockerfile: xác định cách xây dựng một image Docker tùy chỉnh; chứa các lệnh để thiết lập môi trường cho các node trong mạng Raft

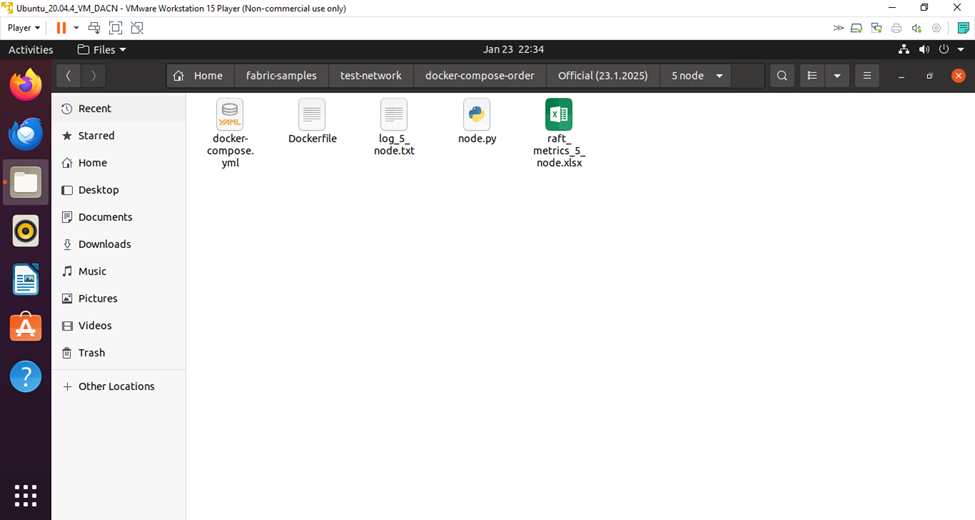
+ node.py: thực hiện mô phỏng thuật toán đồng thuận Raft

+ log.txt: ghi lại log thông báo từ quá trình chạy

+ raft\_metrics.xlsx: ghi lại số liệu và kết quả về hiệu suất của quá trình bầu cử Leader



Hình 6: Cấu trúc folder triển khai cho các trường hợp



Hình 7: Cấu trúc folder triển khai kịch bản cụ thể

CHƯƠNG IV: KẾT QUẢ THỰC NGHIỆM

1. Kết quả



Hình 8: Kết quả cho 1 raft\_metrics.xlsx

- Giá trị thông số ở raft\_metrics.xlsx:

|  |  |  |  |
| --- | --- | --- | --- |
|  | Trung bình | Cao nhất | Thấp nhất |
| Thời gian bầu cử (s) | 1.001543369 | 1.002588511 | 1.00010848 |
| CPU sử dụng (%) | 0.705154639 | 1.5 | 0 |
| RAM sử dụng (GB) | 1.023116279 | 1.023448944 | 1.022735596 |

Bảng 1. Giá trị thông số ở “raft\_metrics\_3\_node.xlsx”

|  |  |  |  |
| --- | --- | --- | --- |
|  | Trung bình | Cao nhất | Thấp nhất |
| Thời gian bầu cử (s) | 1.001883145 | 1.005224466 | 1.000430107 |
| CPU sử dụng (%) | 1.243298969 | 2.5 | 0 |
| RAM sử dụng (GB) | 1.006443063 | 1.007972717 | 1.006065369 |

Bảng 2. Giá trị thông số ở “raft\_metrics\_5\_node.xlsx”

|  |  |  |  |
| --- | --- | --- | --- |
|  | Trung bình | Cao nhất | Thấp nhất |
| Thời gian bầu cử (s) | 1.00205713 | 1.006422997 | 1.000276327 |
| CPU sử dụng (%) | 1.404123711 | 2.6 | 0 |
| RAM sử dụng (GB) | 1.026300686 | 1.030513763 | 1.025867462 |

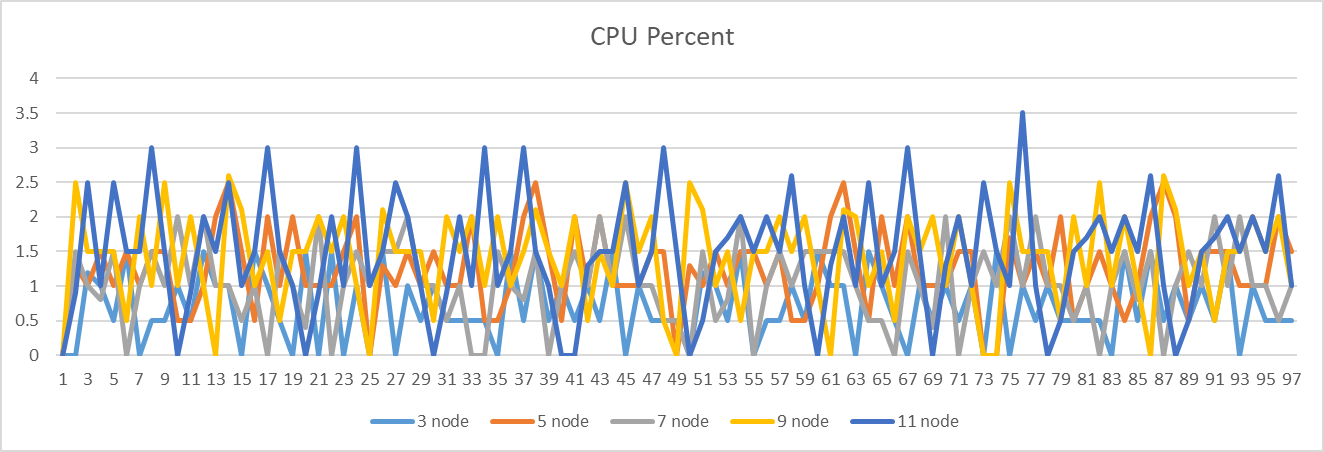
Bảng 3. Giá trị thông số ở “raft\_metrics\_7\_node.xlsx”

|  |  |  |  |
| --- | --- | --- | --- |
|  | Trung bình | Cao nhất | Thấp nhất |
| Thời gian bầu cử (s) | 1.001856325 | 1.005013704 | 1.000424147 |
| CPU sử dụng (%) | 1.404123711 | 2.6 | 0 |
| RAM sử dụng (GB) | 1.026300686 | 1.030513763 | 1.025867462 |

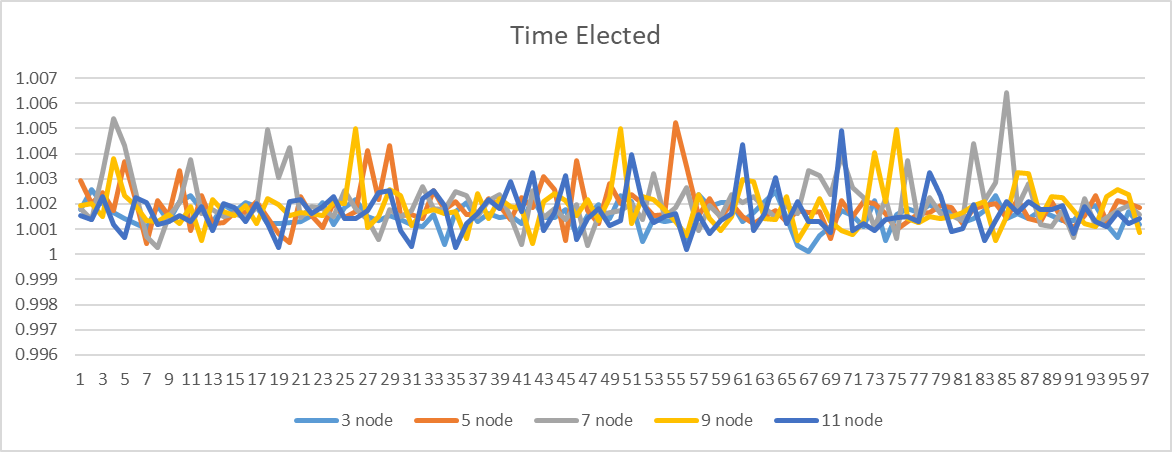
Bảng 4. Giá trị thông số ở “raft\_metrics\_9\_node.xlsx”

|  |  |  |  |
| --- | --- | --- | --- |
|  | Trung bình | Cao nhất | Thấp nhất |
| Thời gian bầu cử (s) | 1.001660996 | 1.00490737 | 1.000169039 |
| CPU sử dụng (%) | 1.478350515 | 3.5 | 0 |
| RAM sử dụng (GB) | 1.050109116 | 1.050121307 | 1.050102234 |

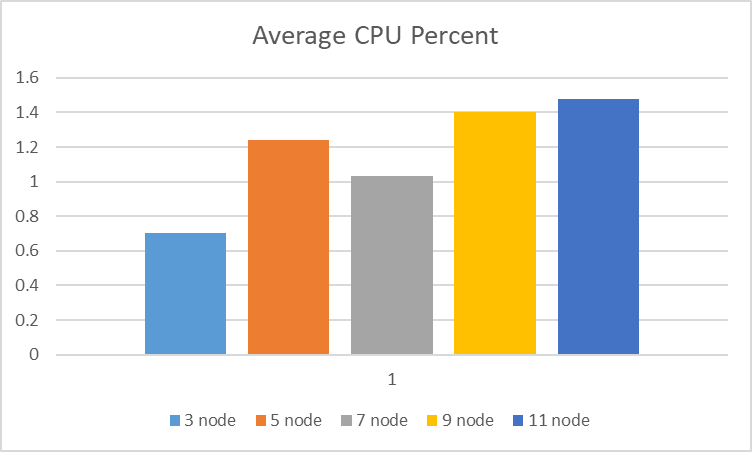
Bảng 5. Giá trị thông số ở “raft\_metrics\_11\_node.xlsx”



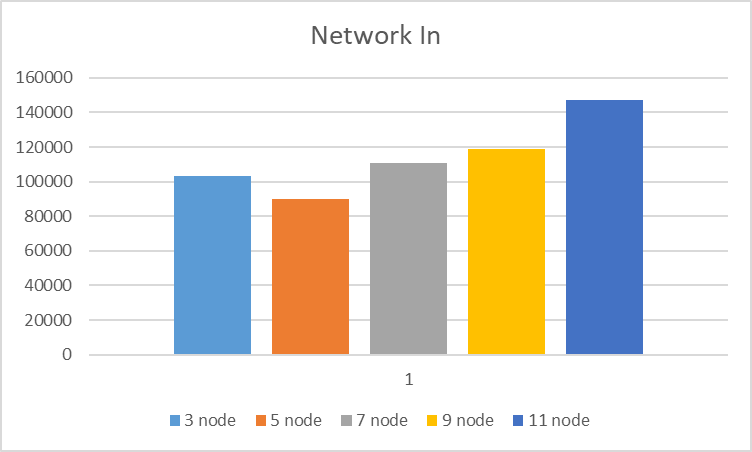
Hình 9: Mức độ sử dụng CPU giữa các mô hình node với nhau



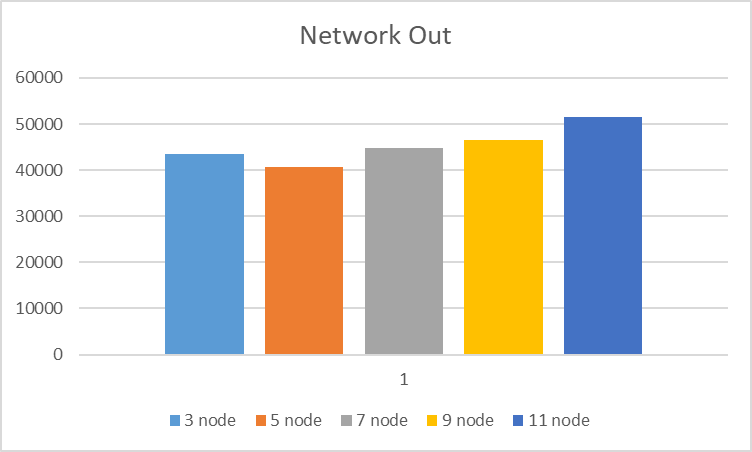
*Hình 10: Thời gian bầu cử giữa mô hình node với nhau*



Hình 11: CPU trung bình giữa các mô hình node



*Hình 12: Network In trung bình giữa các mô hình node*



*Hình 13: Network Out trung bình giữa các mô hình node*

- Lưu lượng Network In/Out ở mỗi mô hình node ổn định, không có sự thay đổi. 🡺 Cơ chế Raft không yêu cầu sử dụng tài nguyên mạng quá nhiều, việc mở rộng số lượng nút không ảnh hưởng đến hiệu suất mạng.

- Thời gian tái bầu chọn (time\_elected) dao động trong khoảng từ 1.0 đến 1.006 giây cho tất cả các nút ở tất cả các mô hình. Điều này cho thấy quá trình bầu chọn diễn ra khá nhanh và không có sự chênh lệch lớn. Thời gian tái bầu chọn tương đối ổn định trong suốt quá trình bầu chọn, cho thấy rằng các nút đã hoàn thành nhiệm vụ bầu chọn của mình mà không gặp phải vấn đề lớn về thời gian.

🡺 Tính nhất quán của cơ chế Raft trong việc chọn bầu chọn khi có sự cố.

- Mức sử dụng CPU của các nút thấp, dao động từ 0% đến 3.5% ở mô hình node và trung bình từ trong khoảng 0.7% đến 1.5%. Điều này cho thấy việc bầu cử trong hệ thống Raft không yêu cầu tài nguyên CPU cao, đồng thời không có sự gia tăng quá lớn khi tăng số lượng node lên (từ 3 node đến 11 node CPU sử dụng trung bình tăng 2%).

- Lượng RAM được sử dụng ổn định ở mức khoảng 1 GB với tất cả mô hình node.

🡺 Khả năng mở rộng, không gây áp lực đáng kể lên tài nguyên.

1. Link mã nguồn

- Link Github:

<https://github.com/Tdhuyvk/NT114.P11.ATCL-Specialized_Project>

- Link OneDrive:

<https://drive.google.com/drive/folders/18T7w0PcCt7vY8MpQqMryFf5WrKuK76A8?usp=sharing>

CHƯƠNG IV: KẾT LUẬN

Với sự phát triển nhanh chóng của các công nghệ blockchain và yêu cầu ngày càng cao về tính toàn vẹn và khả năng chịu lỗi trong các hệ thống phân tán, việc áp dụng các giao thức đồng thuận hiệu quả là một yếu tố quan trọng. Đồ án "Đánh Giá Hiệu Suất Cơ Chế Đồng Thuận Raft Trong Blockchain" đã đạt được những kết quả tích cực, chứng minh được tiềm năng của cơ chế Raft trong việc duy trì tính toàn vẹn và hiệu quả trong các hệ thống blockchain.

Hệ thống được thiết kế với quy trình chặt chẽ, từ việc xây dựng kịch bản bầu chọn người lãnh đạo trong mô hình Raft, đến việc thu thập và phân tích dữ liệu hiệu suất. Các thử nghiệm thực tế đã cho thấy cơ chế Raft hoạt động ổn định trong môi trường với số lượng nút đa dạng, đảm bảo khả năng phục hồi khi có nút bị lỗi, và duy trì khả năng đồng thuận qua quá trình bầu chọn lại người lãnh đạo khi cần thiết. Các chỉ số hiệu suất như thời gian bầu chọn, CPU, RAM, và lưu lượng mạng đã được thu thập và phân tích một cách chi tiết, góp phần làm rõ ưu nhược điểm của Raft trong các tình huống khác nhau.

Tuy nhiên, đồ án vẫn còn một số hạn chế cần cải thiện, chẳng hạn như việc mở rộng hệ thống với số lượng nút lớn hơn, tối ưu hóa các thuật toán xử lý và nâng cao khả năng thích ứng của Raft với các tình huống mạng phức tạp. Đây sẽ là những hướng nghiên cứu tiếp theo để hoàn thiện và nâng cao ứng dụng này.

Cuối cùng, nhóm em hy vọng rằng đồ án này sẽ không chỉ là một bước tiến quan trọng trong hành trình học tập và nghiên cứu của nhóm, mà còn đóng góp một phần nhỏ vào việc phát triển các hệ thống blockchain hiệu quả và bền vững, góp phần xây dựng một nền tảng công nghệ mạnh mẽ cho các ứng dụng trong tương lai.

TÀI LIỆU THAM KHẢO

1. G. Yang, K. Lee, K. Lee, Y. Yoo, H. Lee, and C. Yoo, "Resource Analysis of Blockchain Consensus Algorithms in Hyperledger Fabric," IEEE Transactions on Blockchain, 2022.