HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG



Nguyễn Đoàn Quân

NGHIÊN CỨU MÔ HÌNH CÂN BẰNG TẢI ÁP DỤNG LÝ THUYẾT HÀNG ĐỢI CHO GAME ONLINE NHIỀU NGƯỜI CHƠI

Chuyên ngành : Hệ thống thông tin

Mã số: 8.48.01.04

TÓM TẮT ĐỀ ÁN TỐT NGHIỆP THẠC SĨ

HÀ NỘI - 2025

Đề án tốt nghiệp được hoàn thành tại:
HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

Người hướng dẫn khoa học: PGS.TS Lê Hữu Lập

Phản biện 1: TS. Trịnh Bảo Ngọc

Phản biện 2: TS. Trần Tiến Công

Đề án tốt nghiệp sẽ được bảo vệ trước Hội đồng chấm đề án tốt nghiệp thạc sĩ tại Học viện Công nghệ Bưu chính Viễn thông

Vào lúc: 11 giờ 45 ngày 24 tháng 01 năm 2026

Có thể tìm hiểu đề án tốt nghiệp tại: - Thư viện của Học viện Công nghệ Bưu chính Viễn thông

MỞ ĐẦU

1. Lý do chọn đề tài

Sự bùng nổ của ngành game trực tuyến đã biến các hệ thống game thành những hệ thống phân tán phức tạp, đòi hỏi khả năng duy trì hàng triệu kết nối thời gian thực với độ trễ thấp. Tuy nhiên, các kỹ thuật cân bằng tải truyền thống vốn tối ưu cho Web lại bộc lộ nhiều hạn chế trong môi trường này. Xuất phát từ nhu cầu cấp thiết đó, đề tài nghiên cứu ứng dụng lý thuyết hàng đợi vào cân bằng tải được thực hiện nhằm tối ưu hóa hiệu năng hệ thống và khắc phục các nhược điểm cũ, mang lại giá trị cao về cả khoa học lẫn thực tiễn.

2. Tổng quan về vấn đề nghiên cứu

Mặc dù các giải pháp cân bằng tải kinh điển là tiêu chuẩn phổ biến, chúng lại kém hiệu quả đối với Game Online nhạy cảm về độ trễ, đặc biệt khi hệ thống chịu tải cao hoặc phần cứng không đồng nhất.

Nguyên nhân cốt lõi là các thuật toán này hoạt động theo cơ chế "mù trạng thái", giả định năng lực máy chủ là ngang bằng nhau mà không nhận diện được sự biến thiên thực tế dựa trên mức tiêu thụ tài nguyên (RAM/CPU), dẫn đến không phát hiện kịp thời tình trạng thời gian chờ tăng vọt. Hiện nay, việc áp dụng lý thuyết hàng đợi (như M/M/1) để giải quyết vấn đề này đã có thành tựu trong viễn thông, nhưng việc cụ thể hóa thành thuật toán thực tế thích ứng với máy chủ Game vẫn là một thách thức cần giải quyết.

3. Mục đích nghiên cứu

Mục đích cốt lõi của đề án là nghiên cứu và kiểm chứng một mô hình cân bằng tải mới thông minh hơn nhằm khắc phục các hạn chế của hệ thống cũ, tập trung vào ba nhiệm vụ chính:

- **Mô hình hóa toán học:** Ứng dụng lý thuyết hàng đợi (mô hình M/M/1 biến thiên) để biểu diễn mối quan hệ giữa tải trọng đầu vào, tài nguyên hệ thống và thời gian chờ dự kiến.
- **Phát triển thuật toán QPLB:** Thiết kế thuật toán có khả năng dự báo tắc nghẽn dựa trên dữ liệu đo lường thực tế (RAM, CPU) để tính toán "Chi phí tắc nghẽn" và tự động điều hướng lưu lượng tránh các máy chủ suy giảm hiệu năng.
- **Đánh giá thực nghiệm:** Triển khai trên môi trường giả lập sát thực tế để chứng minh hiệu quả vượt trội về tỷ lệ lỗi, độ ổn định và khả năng tận dụng tài nguyên so với các giải pháp truyền thống

4. Đối tượng và phạm vi nghiên cứu

Đối tượng nghiên cứu:

- Hệ thống kiến trúc máy chủ Game Online theo mô hình Client-Server.
- Các thuật toán cân bằng tải truyền thống.
- Các mô hình toán học trong Lý thuyết hàng đợi.
- Các chỉ số hiệu năng hệ thống: Thời gian phản hồi, thông lượng, tỷ lệ lỗi và mức tiêu thụ tài nguyên.

Phạm vi nghiên cứu:

- Đề tài tập trung nghiên cứu sâu vào tầng phân phối kết nối của hệ thống Game.
- Nghiên cứu được thực hiện trên môi trường giả lập sử dụng công nghệ Container (Docker) và nền tảng Node.js để mô phỏng các cụm máy chủ game.
- Đề tài không đi sâu vào việc tối ưu hóa mã nguồn trò chơi hay đồ họa, mà tập trung giải quyết bài toán định tuyến và quản lý tài nguyên hệ thống hạ tầng.

5. Phương pháp nghiên cứu

Để đạt được các mục tiêu đề ra, đề tài sử dụng kết hợp các phương pháp nghiên cứu sau:

- **Phương pháp nghiên cứu lý thuyết:** Thu thập, tổng hợp và phân tích các tài liệu khoa học, bài báo chuyên ngành về kiến trúc game server, kỹ thuật cân bằng tải và lý thuyết hàng đợi. Đồng thời sử dụng toán học để xây dựng công thức tính toán thời gian chờ kỳ vọng dựa trên mô hình M/M/1.
- **Phương pháp mô hình hóa và giả lập:** Thiết kế kiến trúc hệ thống bao gồm Load Balancer trung tâm, cụm Game Server và Client tạo tải giả lập. Sử dụng Docker và PM2 để quản lý và giới hạn tài nguyên cho từng node nhằm tái hiện các kịch bản quá tải.
- **Phương pháp thực nghiệm và so sánh:** Xây dựng các kịch bản kiểm thử đa dạng bao gồm kịch bản cơ sở, kịch bản chênh lệch cấu hình, và kịch bản quá tải tài nguyên. Thực hiện đo đạc số liệu định lượng đối với từng thuật toán , sau đó so sánh đối chứng kết quả giữa thuật toán đề xuất (QPLB) và các thuật toán truyền thống để rút ra kết luận khoa học về tính hiệu quả và khả năng ứng dụng.

6. Nội dung chính của đề án

Nội dung chính của đề án được cấu trúc thành 3 chương:

- Chương 1: Nghiên cứu tổng quan về game online, các kỹ thuật cân bằng tải, cơ sở lý thuyết hàng đợi và khả năng ứng dụng trong mô hình hệ thống cân bằng tải trong game online.
- Chương 2: Xây dựng cơ sở lý luận và thiết kế chi tiết cho giải pháp cân bằng tải dự báo dựa trên hàng đợi.
- Chương 3: Thử nghiệm và đánh giá kết quả về giải pháp cân bằng tải được đề xuất.

CHƯƠNG 1. NGHIÊN CỨU TỔNG QUAN

1.1. Tổng quan về game

Trong bối cảnh kỷ nguyên số, ngành công nghiệp trò chơi điện tử đã có những bước phát triển vượt bậc, trở thành một trong những lĩnh vực giải trí mang lại doanh thu khổng lồ trên toàn cầu. Dựa trên yêu cầu về hạ tầng kết nối mạng, trò chơi điện tử được phân chia thành hai nhóm chính: Game Offline (vận hành độc lập trên thiết bị người dùng) và Game Online (yêu cầu kết nối mạng thường trực để tương tác thời gian thực).

Đề tài tập trung nghiên cứu Game Online, đặc biệt là nhóm game nhiều người chơi (MMO), do các yêu cầu khắt khe về băng thông và xử lý dữ liệu. Về mặt kiến trúc hệ thống, Game Online được chia thành hai mô hình chính:

- Trò chơi dựa trên phiên đấu: Điển hình như các game MOBA hay FPS (CS:GO, PUBG). Đặc điểm là người chơi được chia vào các phòng nhỏ (Room) với số lượng giới hạn, trạng thái phòng chơi sẽ được giải phóng khi trận đấu kết thúc.
- Trò chơi trực tuyến nhiều người chơi: Điển hình như các game nhập vai MMORPG. Đặc điểm là hàng nghìn đến hàng triệu người chơi cùng tham gia vào một thế giới ảo duy nhất và liên tục. Đây là mô hình phức tạp nhất, đặt ra thách thức lớn cho hệ thống cân bằng tải trong việc xử lý lượng kết nối khổng lồ đồng thời và duy trì tính nhất quán dữ liệu trong thời gian dài.

1.2. Các đặc trưng về game online

Để thiết kế hạ tầng hiệu quả, cần xác định rõ lưu lượng game mang những đặc thù khác biệt hoàn toàn so với các dịch vụ Internet truyền thống trên ba khía cạnh chính:

- Đặc trưng về lưu lượng truy cập:
    - Kích thước gói tin: Hệ thống chủ yếu xử lý các chuỗi gói tin rất nhỏ (hơn 90% dưới 200 bytes). Có sự bất đối xứng lớn về kích thước: chiều từ người chơi đến máy chủ rất nhỏ (lệnh điều khiển), trong khi chiều ngược lại lớn hơn do phải tổng hợp trạng thái.
    - Tính chu kỳ: Máy chủ gửi dữ liệu theo nhịp cố định (ví dụ 50ms), tạo ra mô hình lưu lượng dạng "bùng nổ" thay vì liên tục, gây áp lực xử lý tức thời lớn lên thiết bị mạng.
    - Giao thức truyền tải: Đa số sử dụng giao thức UDP để ưu tiên tốc độ thời gian thực. Hệ thống chấp nhận việc mất gói tin thay vì chờ truyền lại (TCP) để tránh việc dữ liệu bị lỗi thời.
- Đặc trưng về thời gian phản hồi (Response Time & Latency): Đây là yếu tố quyết định trải nghiệm người dùng (QoE). Do game cực kỳ nhạy cảm với độ trễ, hệ thống không thể sử dụng kỹ thuật đệm (buffering) để xử lý tắc nghẽn mà buộc phải ưu tiên khả năng chuyển tiếp nhanh cho các gói tin nhỏ.
- Hành vi tương tác của người chơi: Thời gian phiên chơi tuân theo phân phối Weibull, thường kéo dài dưới 2 giờ. Hệ thống đòi hỏi sự ổn định cao ngay từ đầu để tránh tỷ lệ rời bỏ, đồng thời phải có khả năng điều phối tài nguyên động theo múi giờ và thói quen sinh hoạt của người chơi.

1.3. Kiến trúc mô hình hệ thống trong game online

Mô hình Client-Server (Khách-Chủ) hiện là tiêu chuẩn công nghiệp cho các game online yêu cầu tính nhất quán cao và chống gian lận.

- Cơ chế vận hành: Máy chủ đóng vai trò trung tâm, nắm giữ "chân lý" về trạng thái toàn cục của thế giới ảo. Nó tiếp nhận luồng dữ liệu đầu vào từ người chơi, thực thi logic game (tính toán vật lý, va chạm) và đồng bộ hóa trạng thái mới nhất đến các máy trạm. Máy trạm (Client) chỉ đóng vai trò thu thập thao tác người dùng (input) và hiển thị hình ảnh, âm thanh dựa trên dữ liệu từ máy chủ.
- Các kỹ thuật xử lý độ trễ: Do giới hạn vật lý về tốc độ truyền tin, hệ thống game tích hợp các kỹ thuật nâng cao để đảm bảo trải nghiệm mượt mà:
    - *Dự đoán phía máy khách (Client-side Prediction):* Client mô phỏng ngay lập tức hành động của nhân vật mà không cần chờ phản hồi từ Server để triệt tiêu cảm giác trễ. Nếu có sai lệch sau khi nhận dữ liệu thực từ Server, Client sẽ thực hiện hòa giải (Reconciliation).
    - *Kỹ thuật bù trễ (Lag Compensation):* Server lưu lại lịch sử trạng thái thế giới game. Khi xử lý một hành động (như bắn súng), Server "tua ngược" thời gian về đúng thời điểm người chơi thực hiện hành động đó để kiểm tra kết quả, đảm bảo công bằng cho người chơi có độ trễ mạng khác nhau.

1.4. Các kỹ thuật cân bằng tải

Cân bằng tải là giải pháp phân phối lưu lượng truy cập đến một nhóm máy chủ nhằm tối ưu hóa hiệu suất và ngăn chặn quá tải. Đối với Game Online, các kỹ thuật truyền thống bộc lộ cả ưu điểm và hạn chế riêng.

- **Các thuật toán cân bằng tải truyền thống:**
    - *Round Robin:* Phân phối yêu cầu lần lượt theo thứ tự. Ưu điểm là đơn giản, chi phí tính toán thấp. Nhược điểm là "mù trạng thái", không nhận biết được máy chủ nào đang quá tải hoặc cấu hình yếu, dễ dẫn đến nghẽn cổ chai cục bộ.
    - *Least Connections:* Điều phối yêu cầu đến máy chủ có ít kết nối nhất. Phù hợp cho các phiên kết nối dài. Tuy nhiên, trong game, số lượng kết nối không phản ánh đúng tải trọng xử lý (một trận chiến phức tạp tốn tài nguyên CPU gấp nhiều lần một kết nối nhàn rỗi), do đó thuật toán này vẫn có thể gây quá tải cho máy chủ đang xử lý tác vụ nặng.
    - *Least Response Time:* Ưu tiên máy chủ phản hồi nhanh nhất. Rất tốt cho trải nghiệm người dùng nhưng dễ gây ra hiện tượng dao động tải (hiệu ứng đám đông), làm mất ổn định hệ thống.
    - *IP Hash:* Phân phối dựa trên địa chỉ IP người dùng để đảm bảo tính duy trì phiên. Nhược điểm là dễ gây phân cực tải (nếu nhiều người chơi dùng chung một IP quán net) và khó mở rộng hệ thống linh hoạt.
- **So sánh Game Online với Ứng dụng Web:** Sự khác biệt căn bản khiến các kỹ thuật cân bằng tải web truyền thống kém hiệu quả với game nằm ở ba điểm:
    1. *Tính chất kết nối:* Web là yêu cầu ngắn, ngắt quãng; Game là luồng liên tục, kết nối bền vững.
    2. *Trạng thái dữ liệu:* Web thường phi trạng thái; Game có trạng thái cao, dữ liệu nhân vật thay đổi liên tục trên RAM server.
    3. *Độ nhạy thời gian thực:* Game không chấp nhận độ trễ >100ms và không thể sử dụng cơ chế đệm dữ liệu như Web.

1.5. Cơ sở lý thuyết hàng đợi và khả năng ứng dụng trong mô hình hệ thống game online

Để khắc phục hạn chế của các thuật toán tĩnh, lý thuyết hàng đợi được đưa vào như một cơ sở toán học để định lượng và dự báo hiệu năng.

- **Định nghĩa và ánh xạ:** Lý thuyết hàng đợi nghiên cứu về các dòng chờ đợi. Trong hệ thống game:
    - *Khách hàng:* Là các gói tin input/action từ người chơi.
    - *Trạm phục vụ:* Là luồng xử lý logic của Game Server (CPU).
    - *Thời gian phục vụ:* Là thời gian CPU cần để xử lý một vòng lặp game (Tick).
    - *Hàng đợi:* Là bộ đệm socket chứa các gói tin chờ xử lý.
- **Ứng dụng trong cân bằng tải:** Việc áp dụng lý thuyết hàng đợi giúp chuyển từ cơ chế "đếm kết nối" sang cơ chế "dự báo tắc nghẽn". Quy trình tích hợp gồm:
    1. Giám sát tốc độ gói tin đến và thời gian xử lý thực tế của từng server.
    2. Sử dụng mô hình toán học (như M/M/1) để tính toán *Thời gian chờ dự kiến (Wq)*. Chỉ số này phản ánh chính xác "sức khỏe" của server hơn là các chỉ số phần cứng thô.
    3. Điều phối người chơi mới đến máy chủ có thời gian chờ dự báo thấp nhất, giúp chủ động ngăn chặn tắc nghẽn trước khi nó xảy ra.

1.6. Kết luận chương 1

Chương 1 đã hệ thống hóa bức tranh toàn cảnh về công nghệ và thách thức của hệ thống Game Online nhiều người chơi. Phân tích cho thấy lưu lượng game mang tính ngẫu nhiên, chu kỳ và cực kỳ nhạy cảm với độ trễ. Các mô hình cân bằng tải truyền thống do thiếu khả năng nhận thức trạng thái xử lý bên trong của ứng dụng game (logic game, vật lý) nên thường hoạt động kém hiệu quả, gây ra tình trạng quá tải cục bộ hoặc lãng phí tài nguyên.

Từ đó, chương này khẳng định sự cần thiết của một cách tiếp cận mới dựa trên định lượng toán học. Việc ứng dụng Lý thuyết hàng đợi để mô hình hóa hành vi dòng dữ liệu và năng lực phục vụ là hướng đi cấp thiết, tạo tiền đề để xây dựng giải pháp cân bằng tải dự báo thông minh sẽ được đề xuất chi tiết trong Chương 2.

CHƯƠNG 2. NGHIÊN CỨU TỔNG QUAN

2.1. Phân tích yêu cầu

Để xây dựng hệ thống cân bằng tải hiệu quả cho Game Online, cần đáp ứng các yêu cầu khắt khe đặc thù so với ứng dụng Web:

- **Yêu cầu hiệu năng:** Hệ thống phải đảm bảo độ trễ thấp (ví dụ <100ms cho game FPS) và độ ổn định cao (giảm thiểu phương sai trễ). Đồng thời, cần tối ưu hóa khả năng xử lý hàng triệu gói tin nhỏ với tần số cao (20-60Hz) và hỗ trợ tốt giao thức UDP.
- **Yêu cầu tính nhất quán:** Do game online có trạng thái (Stateful), dữ liệu phiên chơi được lưu trên RAM server, nên yêu cầu bắt buộc là phải duy trì kết nối bền vững đến đúng máy chủ đã định tuyến ban đầu.
- **Thách thức của kỹ thuật cũ:** Các thuật toán truyền thống gặp vấn đề "mù trạng thái" (số lượng kết nối không phản ánh đúng tải trọng xử lý), gây ra hiện tượng dao động tải và thiếu cơ chế dự báo tắc nghẽn chủ động trước khi sự cố xảy ra.

2.2. Lựa chọn kỹ thuật cân bằng tải phù hợp

Dựa trên các tiêu chí về khả năng nhận thức trạng thái, tính thích ứng với phần cứng không đồng nhất và tối thiểu hóa độ trễ kỳ vọng, đề tài đã phân tích và loại trừ các giải pháp truyền thống:

- **Loại bỏ:** Round Robin (gây quá tải cho máy yếu), Least Connections (không phân biệt được độ nặng nhẹ của tác vụ), IP Hash (dễ gây phân cực tải) và Least Response Time (gây dao động tải).
- **Đề xuất giải pháp:** Áp dụng mô hình **Cân bằng tải dự báo dựa trên hàng đợi (QPLB)**.
- **Cơ sở toán học:** Sử dụng mô hình hàng đợi **M/M/1** để thiết lập mối quan hệ giữa tốc độ đến ($\boldsymbol{\lambda}$) và năng lực phục vụ ($\boldsymbol{\mu}$). Hàm mục tiêu là tối thiểu hóa "Chi phí tắc nghẽn" ($\boldsymbol{C_i}$), được tính toán dựa trên thời gian chờ kỳ vọng, giúp hệ thống tự động tránh xa các máy chủ đang tiệm cận ngưỡng bão hòa ($\boldsymbol{\lambda \to \mu}$).

2.3. Thiết kế mô hình đề xuất

Kiến trúc hệ thống QPLB được thiết kế theo mô hình tập trung, bao gồm 3 thành phần module hóa phối hợp chặt chẽ:

- **Bộ điều phối:** Cổng giao tiếp duy nhất với Client. Thay vì chuyển tiếp dữ liệu, nó chỉ định địa chỉ máy chủ tối ưu để Client kết nối trực tiếp (P2P), giúp giảm tải cho nút trung tâm.
- **Bộ thu thập dữ liệu:** Đóng vai trò giám sát thời gian thực, liên tục cập nhật các chỉ số tải ($\boldsymbol{\lambda}$) và sức khỏe phần cứng từ các Game Server.
- **Bộ máy tính toán:** "Bộ não" của hệ thống, áp dụng công thức M/M/1 để tính toán chi phí tắc nghẽn cho từng máy chủ và ra quyết định điều phối dựa trên dữ liệu thu thập được.

2.4. Quy trình xử lý yêu cầu trong mô hình

Quy trình vận hành khép kín từ khi người dùng khởi tạo yêu cầu đến khi thiết lập kết nối bao gồm:

1. **Tham vấn:** Khi có yêu cầu mới, Dispatcher gửi tín hiệu đến Bộ máy tính toán.
2. **Tính toán:** Hệ thống tính toán chi phí $\boldsymbol{C_i}$ cho tất cả máy chủ dựa trên tải hiện tại và năng lực xử lý.
3. **Ra quyết định:** Chọn máy chủ có chi phí thấp nhất ($\boldsymbol{S_{opt}}$). Nếu máy chủ ưu tiên đã đầy, kích hoạt cơ chế tràn tải sang máy dự phòng.
4. **Kết nối trực tiếp:** Trả về IP/Port cho Client để thiết lập kết nối WebSocket/UDP trực tiếp tới Game Server, bỏ qua Load Balancer cho luồng dữ liệu sau đó.

2.5. Phân tích ưu và nhược điểm của mô hình

- **Ưu điểm:**
  - **Nhận thức năng lực:** Phân biệt được máy chủ mạnh/yếu thông qua tham số $\boldsymbol{\mu}$, giúp tận dụng tối đa hạ tầng không đồng nhất.
  - **Dự báo chủ động:** Sử dụng công thức hàng đợi để dự báo sự gia tăng độ trễ theo hàm mũ, từ đó chủ động điều tiết lưu lượng trước khi hàng đợi vật lý bị tràn.
  - **Ổn định:** Ngăn chặn hiệu ứng sập dây chuyền trong các sự kiện tải đột biến.
- **Hạn chế:** Chi phí tính toán tại tầng điều phối cao hơn O(n) so với thuật toán tĩnh. Độ chính xác phụ thuộc vào việc ước lượng tham số $\boldsymbol{\mu}$ thực tế và độ trễ cập nhật trạng thái của hệ thống giám sát.

2.6. Kết luận chương 2

Chương 2 đã thiết lập cơ sở lý luận và kiến trúc chi tiết cho giải pháp QPLB. Bằng cách chuyển dịch từ cơ chế phân phối tĩnh sang mô hình định lượng dựa trên dữ liệu và lý thuyết hàng đợi M/M/1, giải pháp đề xuất hứa hẹn giải quyết triệt để bài toán tối ưu hóa thời gian chờ và bảo vệ hệ thống trước nguy cơ quá tải, tạo tiền đề cho việc kiểm chứng thực nghiệm trong Chương 3.

CHƯƠNG 3. THỬ NGHIỆM VÀ ĐÁNH GIÁ

3.1. Mục tiêu thử nghiệm

Mục tiêu cốt lõi của chương này là kiểm chứng tính đúng đắn và hiệu quả của mô hình cân bằng tải dự báo dựa trên hàng đợi (QPLB) đã đề xuất. Quá trình thực nghiệm tập trung vào ba nhiệm vụ chính: chứng minh các giới hạn của thuật toán truyền thống (Round Robin, Least Connections) trong môi trường không đồng nhất; kiểm chứng khả năng tự động phát hiện và ngăn chặn quá tải của QPLB; và cuối cùng là so sánh hiệu năng giữa các giải pháp dựa trên các chỉ số định lượng như tỷ lệ lỗi và độ ổn định.

3.2. Môi trường thử nghiệm

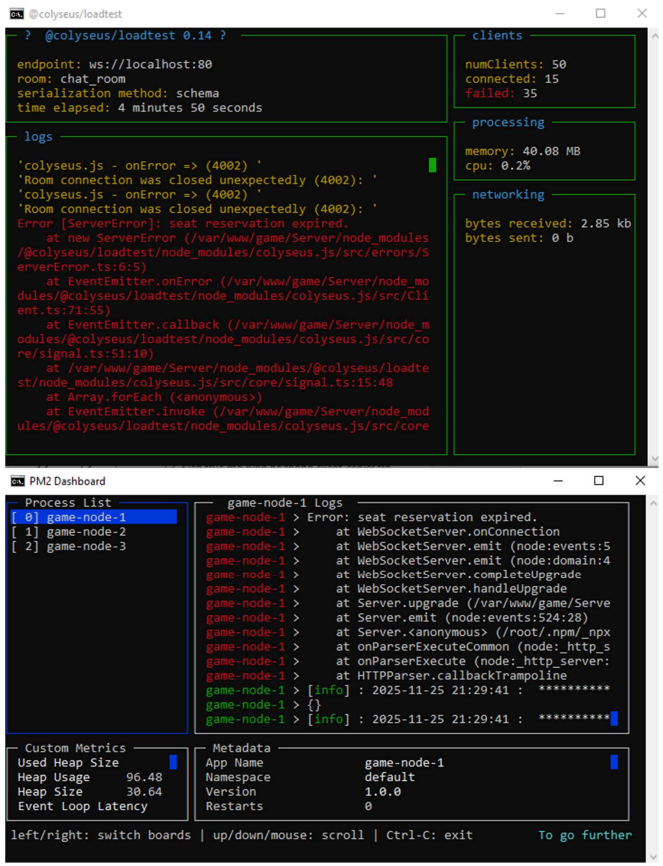
Hệ thống thử nghiệm được xây dựng trên nền tảng ảo hóa Docker để mô phỏng một cụm máy chủ game thu nhỏ sát với thực tế. Kiến trúc bao gồm:

- Bộ cân bằng tải: Sử dụng Nginx đóng vai trò điều phối trung tâm.
- Cụm máy chủ xử lý: Gồm 3 Node chạy Node.js (Framework Colyseus). Trong đó thiết lập cấu hình không đồng nhất: Node 1 là máy cấu hình yếu (độ trễ xử lý cao), Node 2 và Node 3 là máy cấu hình mạnh.
- Cơ sở dữ liệu: Sử dụng Redis và MongoDB để quản lý trạng thái phòng và lưu trữ dữ liệu.

3.3. Kịch bản Thử nghiệm

3.3.1. Thử nghiệm kịch bản 1: Đánh giá giới hạn của kỹ thuật Round Robin

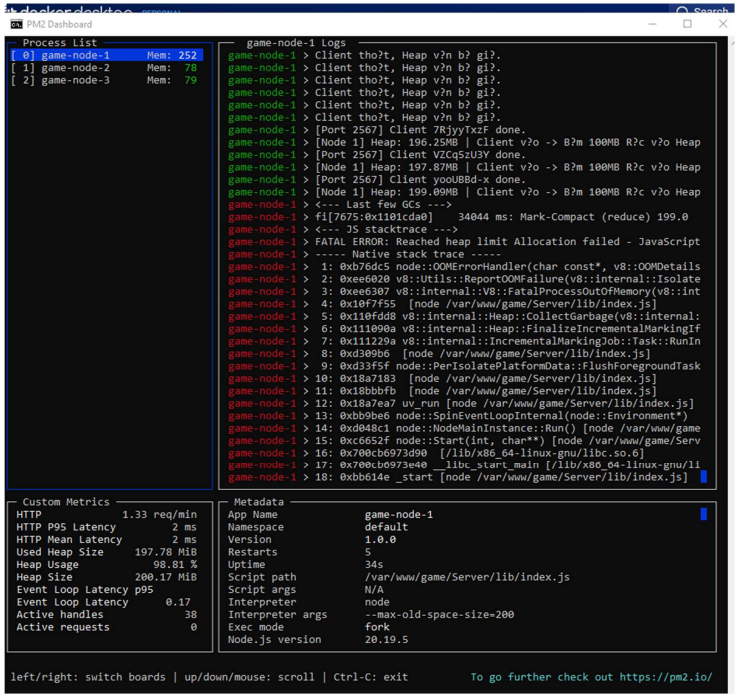
Kịch bản này sử dụng thuật toán phân phối vòng tròn mặc định. Kết quả cho thấy tỷ lệ lỗi kết nối lên tới 70% với mã lỗi chủ đạo liên quan đến việc không tìm thấy thông tin đặt chỗ. Nguyên nhân là do Round Robin hoạt động theo cơ chế phi trạng thái, phân phối yêu cầu tuần tự mà không quan tâm đến dữ liệu phiên người dùng đang nằm ở máy chủ nào, dẫn đến sự bất tương thích với kiến trúc có trạng thái (Stateful) của game online.



Hình Kết quả thử nghiệm lỗi kết nối

3.3.2. Thử nghiệm kịch bản 2: Đánh giá thuật toán Least Connections

Mục tiêu là kiểm chứng giả thuyết rằng việc cân bằng số lượng kết nối không đồng nghĩa với cân bằng tải trọng. Khi triển khai 300 người chơi ảo, thuật toán đã chia đều 100 kết nối cho mỗi máy chủ. Kết quả là Node 1 (cấu hình yếu) bị quá tải RAM và sập nguồn liên tục (khởi động lại 5 lần), trong khi Node 2 và 3 vẫn hoạt động bình thường.



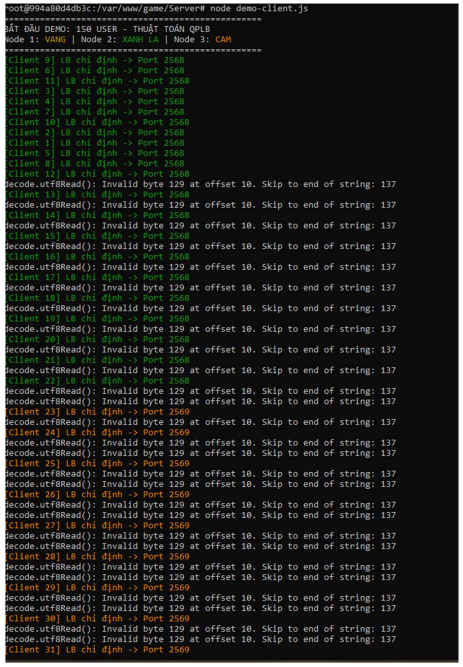
Node 1 đã bị restart 5 lần trong khoảng thời gian 34s khi thực hiện thực nghiệm

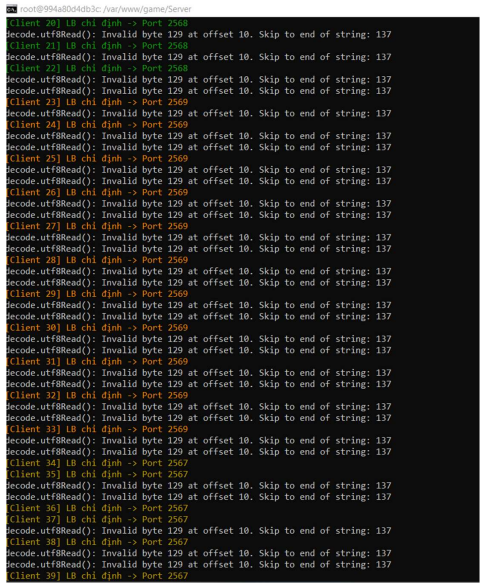
Điều này khẳng định thuật toán Least Connections thất bại trong việc đánh giá tải trọng thực tế, gây ra rủi ro quá tải cục bộ và hiệu ứng sập dây chuyền trong môi trường phần cứng không đồng nhất.

3.3.3. Thử nghiệm kịch bản 3: Áp dụng thuật toán QPLB (Đề xuất)

Kịch bản này áp dụng mô hình QPLB với khả năng dự báo và chiến lược tràn tải.

- Chiến lược phân phối: Hệ thống tự động nhận diện Node 1 là điểm nghẽn tiềm năng. Kết quả cho thấy sự phân phối bất đối xứng: Node 1 chỉ nhận khoảng 2-3% lượng tải, trong khi Node 2 và 3 gánh vác 97-98% còn lại.
- Khả năng thích ứng: Khi tích hợp mô hình đo lường tài nguyên thời gian thực, thuật toán tự động điều chỉnh giá trị chi phí tắc nghẽn dựa trên mức tiêu thụ RAM. Khi Node 1 chạm ngưỡng báo động, hệ thống chủ động ngắt luồng kết nối vào Node này để bảo vệ an toàn.



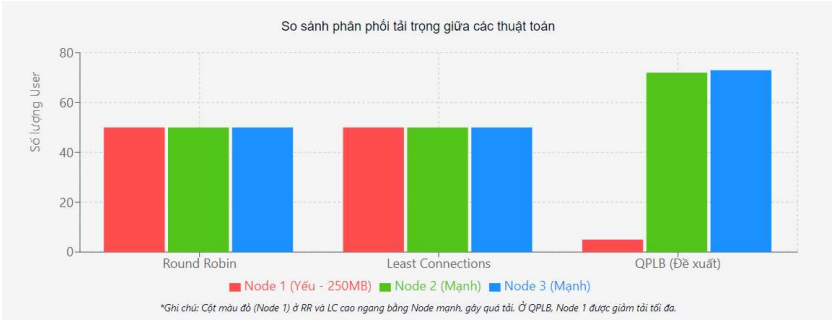


Hình Kết quả điều phối các yêu cầu từ client vào các node trong máy chủ

Kết quả là toàn bộ hệ thống duy trì hoạt động ổn định, không có máy chủ nào bị sập nguồn dù chịu tải trọng lớn.

3.4. Kết quả và Đánh giá tổng hợp

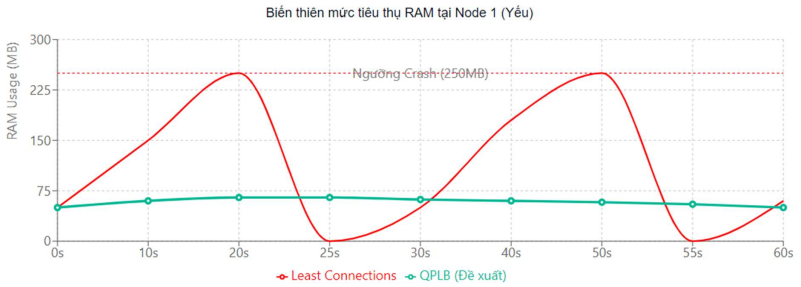
Các chỉ số đánh giá hiệu năng (KPIs) như tỷ lệ hoàn thành yêu cầu, độ ổn định máy chủ và hiệu suất sử dụng tài nguyên đã được đo lường và so sánh.



Hình Biểu đồ so sánh phân phối tải trọng giữa 3 thuật toán

Số liệu tổng hợp cho thấy:

- **Round Robin:** Thất bại do lỗi duy trì phiên (Tỷ lệ đáp ứng thấp).
- **Least Connections:** Thất bại do không bảo vệ được phần cứng yếu (Gây sập nguồn).
- **QPLB:** Đạt tỷ lệ đáp ứng dịch vụ 100%, số lần khởi động lại bằng 0. Biểu đồ biến thiên RAM cho thấy QPLB giữ cho máy chủ yếu luôn hoạt động dưới ngưỡng an toàn, trong khi tận dụng tối đa năng lực của các máy chủ mạnh.



Hình Biểu đồ biến thiên mức tiêu thụ RAM tại Node 1 so sánh giữa các thuật toán

3.5. Kết luận chương 3

Kết quả thực nghiệm đã khẳng định các thuật toán truyền thống không phù hợp cho hệ thống Game Online vận hành trên hạ tầng hỗn hợp do thiếu khả năng nhận thức tài nguyên và trạng thái. Ngược lại, mô hình QPLB đã chứng minh tính ưu việt vượt trội: đảm bảo độ ổn định tuyệt đối, ngăn chặn sập hệ thống và tối ưu hóa thời gian chờ thông qua cơ chế phân phối tải thông minh dựa trên lý thuyết hàng đợi.

KẾT LUẬN

Đề án "Nghiên cứu mô hình cân bằng tải áp dụng lý thuyết hàng đợi cho Game Online nhiều người chơi" xuất phát từ nhu cầu cấp thiết trong việc giải quyết bài toán hiệu năng cho các hệ thống phân tán thời gian thực. Qua quá trình nghiên cứu từ lý thuyết đến thực nghiệm, đề án đã hoàn thành các mục tiêu đề ra và đạt được những kết quả cụ thể sau:

Đề án đã hệ thống hóa được các đặc thù kỹ thuật của lưu lượng Game Online (tính kết nối bền vững, yêu cầu độ trễ thấp, tính chu kỳ). Nghiên cứu đã chỉ ra rằng các thuật toán cân bằng tải truyền thống như Round Robin hay Least Connections vốn được thiết kế cho môi trường Web phi trạng thái, đã bộc lộ những hạn chế mang tính hệ thống khi áp dụng cho Game Online. Cụ thể là sự thiếu hụt khả năng "duy trì phiên" và sự "mù" về năng lực xử lý thực tế của máy chủ.

Đề án đã xây dựng thành công mô hình cân bằng tải dự báo QPLB. Đóng góp cốt lõi của giải pháp là việc ứng dụng lý thuyết hàng đợi để chuyển hóa các dữ liệu thô về tải trọng và năng lực phần cứng thành một chỉ số định lượng duy nhất là chi phí tắc nghẽn. Cơ chế này cho phép hệ thống chuyển từ "phản ứng thụ động" sang "dự báo chủ động", giúp nhận diện sớm các điểm nghẽn cổ chai trước khi chúng gây ra sự cố.

Quá trình kiểm thử trên môi trường giả lập phần cứng không đồng nhất đã cung cấp những bằng chứng định lượng thuyết phục, khẳng định tính đúng đắn của việc áp dụng các mô hình toán học xác suất vào kỹ thuật hệ thống. Khi so sánh, nếu như Round Robin và Least Connections bộc lộ hạn chế nghiêm trọng dẫn đến nguy cơ quá tải cục bộ và sập dây chuyền, thì thuật toán QPLB thể hiện ưu điểm vượt trội về khả năng nhận thức trạng thái. Việc tích hợp lý thuyết hàng đợi giúp QPLB duy trì tỷ lệ đáp ứng dịch vụ tuyệt đối và tối ưu hóa thời gian chờ cho người dùng thông qua việc bảo vệ chủ động các điểm nghẽn tiềm năng.

Giải pháp này có ý nghĩa thực tiễn cao, đặc biệt đối với các nhà phát hành game muốn tận dụng hạ tầng phần cứng hỗn hợp để tối ưu chi phí vận hành mà không làm giảm chất lượng dịch vụ.

Mặc dù giải pháp QPLB đã chứng minh được tính ưu việt trong môi trường giả lập, quá trình nghiên cứu và thực nghiệm cũng đã nhận diện được những điểm yếu và thách thức kỹ thuật cần được khắc phục khi triển khai thực tế. Điểm yếu lớn nhất nằm ở độ trễ cập nhật trạng thái của hệ thống đo lường từ xa. Do cơ chế thu thập dữ liệu hoạt động theo chu kỳ, luôn tồn tại một khoảng trễ nhất định giữa trạng thái thực tế của máy chủ và dữ liệu mà bộ điều phối tiếp nhận. Trong các tình huống tải biến động cực nhanh, các quyết

định điều phối có thể dựa trên dữ liệu chưa kịp cập nhật, làm giảm độ chính xác của thuật toán. Ngoài ra, việc ước lượng tham số năng lực phục vụ hiện tại mới chỉ dựa trên mức tiêu thụ tài nguyên phần cứng mà chưa phản ánh hết độ phức tạp của logic game tại thời điểm thực, ví dụ như sự khác biệt về tải xử lý giữa một nhân vật đứng yên và một nhân vật đang tham gia giao tranh. Hơn nữa, chi phí tính toán tại tầng điều phối của QPLB cao hơn so với các thuật toán tĩnh, có thể tạo ra áp lực xử lý khi quy mô hệ thống mở rộng lên hàng nghìn máy chủ.

Dựa trên những hạn chế kỹ thuật đã nhận diện, hướng phát triển tiếp theo của đề tài sẽ tập trung vào ba giải pháp trọng tâm nhằm hoàn thiện mô hình QPLB cho các hệ thống quy mô lớn.

- Đầu tiên, để khắc phục độ trễ do cơ chế thu thập dữ liệu theo chu kỳ, nghiên cứu sẽ chuyển dịch sang cơ chế đồng bộ hướng sự kiện, cho phép các máy chủ chủ động gửi báo cáo trạng thái ngay lập tức khi phát hiện biến động tài nguyên hoặc tải trọng đột ngột, kết hợp với các giao thức truyền tin tốc độ cao để đảm bảo dữ liệu tại bộ điều phối luôn cập nhật theo thời gian thực.
- Tiếp theo, nhằm nâng cao độ chính xác trong việc ước lượng năng lực phục vụ, đề tài sẽ tích hợp các mô hình trí tuệ nhân tạo và học máy để phân tích lịch sử vận hành, từ đó định lượng chính xác mối tương quan giữa ngữ cảnh logic game (như trạng thái giao tranh hay đứng yên) với tải trọng xử lý thực tế thay vì chỉ phụ thuộc vào chỉ số phần cứng.
- Cuối cùng, để giải quyết áp lực tính toán khi mở rộng quy mô, kiến trúc hệ thống sẽ được nâng cấp theo hướng phi tập trung và tự động hóa trên các nền tảng điện toán đám mây, cho phép tự động mở rộng năng lực xử lý song song và triển khai thuật toán tại các nút biên mạng nhằm loại bỏ điểm nghẽn cổ chai tại trung tâm điều phối.