

ĐẠI HỌC QUỐC GIA HÀ NỘI
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ

Nguyễn Đình Nghĩa

**ĐẢM BẢO CHẤT LƯỢNG DỊCH VỤ
MẠNG NGANG HÀNG CÓ CẤU TRÚC**

Chuyên ngành: Mạng máy tính và truyền dữ liệu
Mã số: 9480102.01

TÓM TẮT LUẬN ÁN TIẾN SĨ NGÀNH CÔNG NGHỆ THÔNG TIN

Hà Nội - 2018

Công trình được hoàn thành tại: Trường Đại học Công nghệ, Đại học Quốc gia Hà Nội

Người hướng dẫn khoa học: TS.Nguyễn Hoài Sơn và PGS.TS. Hồ Sỹ Đàm

Phản biện:

.....

Phản biện:

.....

Phản biện:

.....

Luận án sẽ được bảo vệ trước Hội đồng cấp Đại học Quốc gia chấm luận án tiến sĩ họp tại Trường Đại học Công nghệ, Đại học Quốc gia Hà Nội

vào hồi giờ ngày tháng năm 2019

Có thể tìm hiểu luận án tại:

- Thư viện Quốc gia Việt Nam
- Trung tâm Thông tin - Thư viện, Đại học Quốc gia Hà Nội

MỞ ĐẦU

1. Đặt vấn đề

Ngày nay, nhu cầu tìm kiếm, khai thác và chia sẻ thông tin trên mạng Internet rất lớn và có mặt trong các lĩnh vực của xã hội. Các dịch vụ trên mạng internet chuyển dần từ mô hình tập trung sang mô hình phân tán. Mạng ngang hàng ra đời cho phép người dùng trao đổi, chia sẻ các thông tin, hình ảnh, âm thanh, video, chia sẻ phần cứng như: bộ nhớ, băng thông, chu trình CPU...v.v.

Đã có nhiều ứng dụng được phát triển trên nền tảng công nghệ mạng ngang hàng như: các ứng dụng chia sẻ tệp tin, các ứng dụng tính toán lưới, các ứng dụng truyền thông (Skype, WhatsApp, Lync, Google Talk SETI @ home, IPTV, Video streaming)...v.v. Với các ứng dụng chia sẻ nội dung, người dùng có thể chia sẻ, tìm kiếm và thu thập các tệp tin có kích thước lớn như: tệp tin hình ảnh, âm thanh, video, tệp tin đa phương tiện khác...v.v. Các ứng dụng ban đầu được thiết kế theo hướng không có cấu trúc. Do hạn chế của việc mở rộng mạng trong hướng tiếp cận không cấu trúc cho nên các ứng dụng mạng ngang hàng được phát triển chủ yếu theo hướng có cấu trúc, dựa trên cấu trúc dữ liệu bảng băm phân tán.

Mạng ngang hàng được xây dựng bởi các máy tính tham gia mạng internet, hoạt động trong tầng ứng dụng và trên đỉnh của mạng IP. Liên kết giữa hai nút mạng trong mạng ngang hàng là một liên kết logic, dựa trên nền tảng các liên kết vật lý của mạng lớp dưới.

Chất lượng dịch vụ cho các ứng dụng mạng ngang hàng được đảm bảo bởi hai mức: (1) Chất lượng dịch vụ của lớp mạng vật lý bên dưới, được quy định bởi các tham số như độ trễ, băng thông, jitter...v.v. (2) Chất lượng dịch vụ của lớp mạng phủ bên trên, được quy định bởi đặc tính của mạng ngang hàng.

Với sự phát triển của internet, vấn đề đảm bảo chất lượng dịch vụ cho lớp mạng vật lý bên dưới từng bước được giải quyết. Luận án này giới hạn giải quyết vấn đề đảm bảo chất lượng dịch vụ cho các ứng dụng dựa trên các đặc tính của mạng ngang hàng.

Do đặc điểm kiến trúc và tính chất của mạng ngang hàng cho nên có rất nhiều yếu tố ảnh hưởng đến chất lượng dịch vụ của các ứng dụng mạng ngang hàng. Luận án này tập trung giải quyết các vấn đề dưới đây để đảm bảo chất lượng dịch vụ cho các ứng dụng ngang hàng:

- Cân bằng tải truy vấn trong mạng ngang hàng:

Các mạng ngang hàng được xây dựng dựa trên sự cộng tác của các nút ngang hàng nhau. Mỗi nút tham gia mạng có khả năng xử lý các câu truy vấn khác nhau, vùng không gian khóa do các nút quản lý cũng khác nhau, một nút quản lý dữ liệu có tính phổ biến cao và các nút có thể tham gia hoặc rời mạng một cách tự do.

Các nút trong mạng ngang hàng vừa đóng vai trò là nút xử lý truy vấn, vừa đóng vai trò là nút chuyển tiếp truy vấn. Một nút quản lý nhiều khóa dữ liệu hoặc quản lý dữ liệu có tính phổ biến cao phải xử lý nhiều yêu cầu tìm kiếm. Khi yêu cầu xử lý truy vấn vượt quá khả năng của một nút sẽ đưa một nút vào tình trạng quá tải làm cho truy vấn không thành công, đồng thời ảnh hưởng tới việc chuyển tiếp các câu truy vấn khác. Do đó cần phải có giải pháp để cân bằng tải để đảm bảo được chất lượng dịch vụ trong mạng ngang hàng.

- Tắc nghẽn trong mạng ngang hàng:

Trong mạng ngang hàng, truy vấn dữ liệu được phát ra từ nút yêu cầu đến nút đích và đi qua một dãy các nút khác trong mạng. Nếu trên đường đi gặp một nút quá tải thì truy vấn sẽ không thể đến được đích làm

cho tỷ lệ truy vấn thành công. Do đó, nghiên cứu giải pháp điều khiển tắc nghẽn để lựa chọn tuyến đường đi mới cho các truy vấn là vấn đề cần giải quyết để đảm bảo chất lượng dịch vụ cho các ứng dụng ngang hàng.

- Mất mát dữ liệu trong mạng P2P:

Trong mạng ngang hàng các nút ra/vào tự do mà không có một sự thông báo trước nào. Khi một nút rời khỏi hệ thống, toàn bộ dữ liệu do nó quản lý được chuyển cho nút khác trong mạng quản lý làm tăng đáng kể số lượng truy vấn phải xử lý cho nút nhận dữ liệu. Các tệp tin dữ liệu gốc được lưu trữ trên nút rời khỏi mạng cũng không tồn tại trong mạng. Chính vì vậy, nếu một truy vấn đến nút quản lý khóa (truy vấn thành công) mà tệp tin liên quan đến khóa đó nằm trên nút rời khỏi mạng thì không thể lấy được tệp tin theo yêu cầu. Do vậy, cần phải có giải pháp sao lưu dữ liệu để đảm bảo tính sẵn sàng của dữ liệu nhằm nâng cao chất lượng dịch vụ cho ứng dụng ngang hàng.

2. Mục tiêu của luận án

Luận án phân tích các đặc điểm của mạng ngang hàng ảnh hưởng đến cân bằng tải xử lý của các nút, khả năng định tuyến của các nút và tính sẵn sàng của dữ liệu và đề xuất một số giải pháp đảm bảo chất lượng dịch vụ cho các ứng dụng mạng P2P, đặc biệt là các ứng dụng truy vấn dữ liệu P2P, ứng dụng chia sẻ nội dung.

Mục tiêu của luận án là tăng tỷ lệ thành công của các câu truy vấn và nâng cao tính sẵn sàng của dữ liệu để đảm bảo chất lượng dịch vụ cho các ứng dụng. Chúng tôi đề xuất thuật toán cân bằng tải xử lý truy vấn và thuật toán điều khiển tắc nghẽn giúp nâng cao tỷ lệ thành công của các câu truy vấn; đề xuất thuật toán sao lưu dữ liệu để đảm bảo tính sẵn sàng của dữ liệu trong các ứng dụng mạng P2P.

3. Đóng góp của luận án

Luận án đề xuất các giải pháp sau:

- **Giải pháp cân bằng tải:** Đề xuất một thuật toán đảm bảo cân bằng tải trong mạng ngang hàng có cấu trúc. Giải pháp đề xuất dựa trên thuật toán cân bằng tải theo ngưỡng của Ganesan kết hợp với việc bổ sung khái niệm thư mục để lưu trữ thông tin về các nút nặng tải và nhẹ tải, trong đó xem xét đến cả tải xử lý các câu truy vấn tìm kiếm một nút trong quá trình thực hiện cân bằng tải.

- **Giải pháp điều khiển tắc nghẽn:** Đề xuất giải pháp điều khiển tắc nghẽn tại một nút trong quá trình xử lý các truy vấn. Giải pháp đề xuất thay thế một nút bị tắc nghẽn trong bảng định tuyến của nút chuyển tiếp truy vấn bằng một nút không tắc nghẽn tốt nhất được chọn từ danh sách các nút không tắc nghẽn tiếp theo trên đường đi để tạo ra một tuyến đường mới, tránh tắc nghẽn từ nút chuyển tiếp câu truy vấn đến nút quản lý khóa.

- **Giải pháp sao lưu dữ liệu:** Đề xuất giải pháp sao lưu dữ liệu dựa trên việc phân cụm không gian khóa DHT để đảm bảo tính sẵn sàng của dữ liệu và hỗ trợ cân bằng tải cho hệ thống. Dữ liệu sao lưu được đặt tại các nút trong cụm với nút quản lý khóa.

4. Cấu trúc của luận án

Luận án được tổ chức như sau:

Phần mở đầu giới thiệu khái quát về đảm bảo chất lượng dịch vụ trong mạng ngang hàng có cấu trúc và đưa ra các vấn đề cần giải quyết, đồng thời đánh giá khái quát những đóng góp chính trong luận án này.

Chương 1 trình bày về các kiến thức nền tảng có liên quan phục vụ cho việc nghiên cứu của các chương sau. Nội dung của chương này đề cập đến các vấn đề của mạng ngang hàng như: khái niệm mạng ngang hàng, các đặc trưng của mạng ngang hàng, phân loại mạng ngang hàng và các ứng dụng trên mạng ngang hàng, bảng băm phân tán, giao thức mạng ngang hàng có cấu trúc Chord và vấn đề đảm bảo chất lượng dịch vụ trong mạng ngang hàng có cấu trúc.

Chương 2 trình bày về giải pháp cân bằng tải trong mạng ngang hàng có cấu trúc. Chương này giới thiệu tổng quan về cân bằng tải trong mạng ngang hàng có cấu trúc, đánh giá các nghiên cứu liên quan và đề xuất giải pháp cân bằng tải để nâng cao số lượng của các câu truy vấn đến nút đích.

Chương 3 trình bày về giải pháp điều khiển tắc nghẽn trong mạng ngang hàng có cấu trúc. Nội dung của chương giới thiệu về điều khiển tắc nghẽn và một số nghiên cứu liên quan về điều khiển tắc nghẽn và đề xuất giải pháp điều khiển tắc nghẽn trong mạng ngang hàng có cấu trúc dựa trên cơ chế thay đổi băng định tuyến của một nút.

Chương 4 trình bày giải pháp về sao lưu dữ liệu để đảm bảo tính sẵn sàng của dữ liệu. Chương này giới thiệu khái quát về sao lưu dữ liệu trong mạng ngang hàng có cấu trúc, đánh giá các nghiên cứu liên quan về sao lưu dữ liệu từ đó đề xuất giải pháp sao lưu mới để đảm bảo chất lượng dịch vụ cho các ứng dụng mạng ngang hàng có cấu trúc.

Chương 5 là phần kết luận. Chúng tôi đánh giá các kết quả đã đạt được, những hạn chế và hướng nghiên cứu tiếp theo.

Chương 1. KIẾN THỨC NỀN TẢNG

1.1. Mạng ngang hàng

Mạng ngang hàng được định nghĩa là một cấu trúc mạng phân tán, các thành phần tham gia (nút mạng) cùng nhau chia sẻ tài nguyên (như năng lực xử lý, bộ nhớ lưu trữ, tốc độ đường truyền...v.v.). Các tài nguyên chia sẻ tạo nên dịch vụ và nội dung chia sẻ trong mạng ngang hàng. Các nút mạng truy cập và sử dụng trực tiếp tài nguyên từ các nút khác mà không thông qua các nút trung gian. Các nút tham gia mạng vừa đóng vai trò là máy cung cấp tài nguyên, vừa đóng vai trò là máy yêu cầu tài nguyên.

Các mạng ngang hàng là các mạng ảo, được xây dựng trên đỉnh (top) các mạng vật lý, gồm tập hợp các nút mạng liên kết với nhau. Các nút mạng có sự khác nhau về băng thông đường truyền, tốc độ xử lý, bộ nhớ lưu trữ, dữ liệu chia sẻ...v.v. Các mạng ngang hàng có tính ổn định không cao. Một nút tham gia và rời mạng một cách tự nhiên, không có sự thông báo trước. Khác với các hệ thống khách/chủ, mạng ngang hàng không có thành phần trung tâm để điều khiển, tổ chức, quản trị hoặc duy trì hệ thống trong thời gian mạng hoạt động. Các nút tham gia tạo thành mạng phủ ảo mà không cần quan tâm đến vị trí địa lý.

1.2. Ứng dụng mạng ngang hàng

1.2.1. Phân phối nội dung dựa trên mạng ngang hàng

Mục tiêu chính trong việc thiết kế kiến trúc mạng ngang hàng là hỗ trợ phân phối nội dung giữa cộng đồng người sử dụng và làm giảm tải trên các máy chủ trung tâm. Tuy nhiên mỗi hệ thống lại có vai trò và cách thức chia sẻ nội dung khác nhau. Mỗi nút trong hệ thống là một kho nội dung phân tán để phân phối và chia sẻ nội dung cho các nút khác. Truyền thông đa phương tiện thời gian thực là một lĩnh vực chia sẻ nội dung phổ biến khác dựa trên mạng ngang hàng.

1.2.2. Truyền thông dựa trên mạng ngang hàng

Kiến trúc mạng ngang hàng được sử dụng rộng rãi để xây dựng nhiều ứng dụng truyền thông khác nhau. Các ứng dụng này cung cấp cơ sở hạ tầng cho các nút cộng tác, truyền thông thời gian thực và trực tiếp với nhau, điển hình như các ứng dụng: Skype, AOL, AIM, ICQ, Yahoo, MSN, NetNews và Jabber.

1.2.3. Xử lý và tính toán phân tán dựa trên mạng ngang hàng

Ý tưởng đằng sau các ứng dụng xử lý tính và tính toán phân tán là dựa trên kiến trúc của mạng ngang hàng để phối hợp sức mạnh xử lý có sẵn (khả năng CPU) của mỗi nút. Trong hệ thống đó, nhiệm vụ tính toán ban đầu được chia thành nhiệm vụ nhỏ hơn, các nhiệm vụ nhỏ được gán cho các nút khác nhau xử lý và sau đó kết quả được tổng hợp lại. Trong tính toán, cần có sự kiểm soát tập trung cho việc phối hợp và đồng bộ hóa giữa các nút được tốt hơn. Kiến trúc mạng ngang hàng cho phép các ứng dụng có thể sử dụng máy tính cá nhân để xử lý một bài toán dành cho siêu máy tính với chi phí thấp.

1.2.4. Cộng tác dựa trên mạng ngang hàng

Kiến trúc mạng ngang hàng giúp các nút hoạt động cộng tác với nhau để thực hiện các mục tiêu chung. Groove là ứng dụng dựa trên kiến trúc mạng ngang hàng do Microsoft phát triển và tạo ra môi trường cộng tác cho các nút tham gia. Groove cho phép người dùng cộng tác trực tiếp với nhau mà không cần qua các máy chủ trung tâm. Groove cung cấp cơ chế đồng bộ hóa cho phép các nút cộng tác một cách an toàn, các thông điệp trao đổi với nhau dưới dạng XML.

1.2.5. Hạ tầng công nghiệp/nền tảng dựa trên mạng ngang hàng

Kiến trúc mạng ngang hàng cung cấp cơ sở hạ tầng cho các ứng dụng phân tán. JXTA là hạ tầng mạng ngang hàng khá phổ biến, cung cấp một môi trường lập trình, hạ tầng công nghiệp cho mục đích chung. Hạ tầng JXTA là dịch vụ phân tán độc lập với các giao thức tầng giao vận. JXTA tạo ra một hệ thống mạng ngang hàng với các chức năng cơ bản, cung cấp các thành phần cho việc xây dựng các ứng dụng mạng ngang hàng.

1.2.6. Các hệ thống cơ sở dữ liệu và tìm kiếm dựa trên mạng ngang hàng

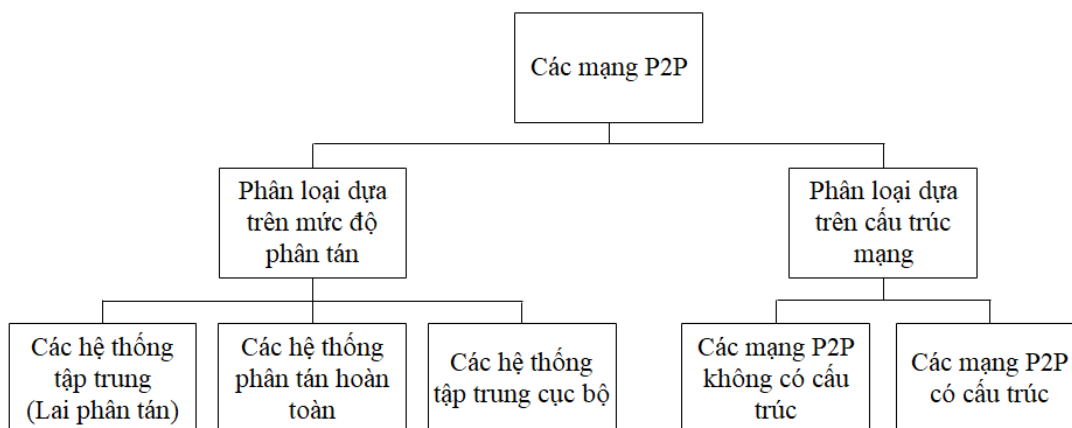
Kiến trúc mạng ngang hàng cũng được sử dụng cho một số hệ thống tìm kiếm và cơ sở dữ liệu phân tán. Open Cola Folders, Pelbio, aKa InfraSearch, WebV2 và Sciencenet là các bộ công cụ tìm kiếm dựa trên P2P. Mô hình quan hệ cục bộ (Local Relational Model - LRM), PIER và các hệ thống Piazza, AmbientDB và Xpeer là các hệ cơ sở dữ liệu quan hệ/phân tán dựa trên kiến trúc P2P.

1.2.7. Các ứng dụng khác

Bên cạnh các lĩnh vực ứng dụng nói trên, kiến trúc mạng ngang hàng còn được sử dụng rộng rãi cho nhiều ứng dụng khác. Lĩnh vực phổ biến bao gồm các ứng dụng trò chơi (2AM, CenterSpan), nghiên cứu y tế (đặc biệt là nghiên cứu ung thư), lĩnh vực giáo dục và nhiều lĩnh vực kinh doanh.

1.3. Phân loại mạng ngang hàng

Phân loại mạng ngang hàng được minh họa trong hình 1.1 và được mô tả trong các mục dưới đây:



Hình 1.1. Phân loại mạng ngang hàng

1.3.1. Phân loại theo mức độ phân tán

Mức độ phân tán là một đặc tính quan trọng của các hệ thống mạng ngang hàng, nó đánh giá sự phụ thuộc của các nút vào một số nút biết trước. Theo mức độ phân tán, mạng ngang hàng được phân thành các loại sau:

1.3.1.1. Các hệ thống mạng ngang hàng tập trung

Các hệ thống mạng ngang hàng tập trung gồm một máy chủ trung tâm lưu giữ thông tin của tất cả các nút tham gia mạng và một thư mục quản lý các tệp tin chia sẻ trong mạng nhằm tạo thuận lợi cho các nút trong quá trình cộng tác với nhau. Các hệ thống mạng ngang hàng tập trung còn được gọi là các hệ thống phân tán lai.

1.3.1.2. Các hệ thống phân tán hoàn toàn

Các hệ thống mạng ngang hàng phân tán hoàn toàn thường được gọi là các hệ thống mạng ngang hàng thuần túy. Trong các hệ thống này không có thành phần trung tâm đóng vai trò tổ chức cho các nút tham gia, tất cả các nút trong mạng tự cộng tác với nhau để thực hiện các công việc.

1.3.1.3. Các hệ thống tập trung cục bộ

Hệ thống mạng ngang hàng phân tán một phần là hình thức cải tiến của hệ thống phân tán thuần túy. Các hệ thống này hoạt động cơ bản giống với các hệ thống phân tán thuần túy. Tuy nhiên trong hệ thống phân tán một phần, một số nút được gán vai trò cụ thể hơn so với các nút khác. Các nút đặc biệt này được gọi là superpeer hoặc supernode, hoạt động như một máy chủ trung tâm để quản lý một số nút nhất định.

1.3.2. Phân loại theo cấu trúc mạng ngang hàng

Theo cấu trúc, mạng ngang hàng được phân thành hai loại sau:

1.3.2.1. Mạng ngang hàng không cấu trúc

Mạng ngang hàng không cấu trúc không có bất kỳ thông tin cụ thể nào về nội dung đa phương tiện của các nút được yêu cầu. Trong các hệ thống này không có thông tin cho việc tổ chức nút và nội dung đa phương tiện lưu trữ trên các nút tương ứng. Việc tìm kiếm một nội dung được thực hiện một cách ngẫu nhiên vì các nút không có thông tin về vị trí của nút chứa nội dung cần tìm kiếm. Nút cần tìm kiếm nội dung đa phương tiện gửi yêu cầu truy vấn đến các nút khác nhau và truy vấn này được lan truyền cho đến khi tìm thấy nội dung được yêu cầu.

1.3.2.2. Mạng ngang hàng có cấu trúc

Các mạng ngang hàng có cấu trúc sử dụng bảng băm phân tán (DHT) để hỗ trợ khả năng mở rộng. Các mạng ngang hàng có cấu trúc tổ chức các nút trong mạng phủ và chỉ mục cho nội dung đa phương tiện được đặt tại các vị trí rất cụ thể. Tổ chức của mạng phủ cung cấp một ánh xạ cho mỗi tệp tin và các truy vấn tìm kiếm được thực hiện với sự giúp đỡ của bảng ánh xạ này. Cấu trúc mạng phủ có mối liên hệ chặt chẽ với các vị trí của nội dung dựa trên bảng băm được kiểm soát đã tạo ra một giải pháp có khả năng mở rộng cho truy vấn chính xác bởi vì vị trí của nội dung đa phương tiện cần tìm được biết trước khi tìm kiếm.

1.4. Mạng ngang hàng có cấu trúc

Mạng ngang hàng có cấu trúc được xây dựng dựa trên cấu trúc bảng băm phân tán.

1.4.1. Bảng băm phân tán

Bảng băm phân tán (DHT) là một hệ thống phân tán cung cấp chức năng tìm kiếm dữ liệu tương tự như bảng băm thông thường. Mỗi cặp khóa và giá trị được lưu trữ phân tán tại các nút trong DHT, bất kỳ nút nào trong hệ thống cũng có thể lấy được giá trị ứng với một khóa xác định.

1.4.1.1. Quản lý dữ liệu phân tán

Bảng băm phân tán quản lý dữ liệu thông qua phân phối nó trên một số nút và cài đặt cơ chế định tuyến để tìm kiếm nút chứa dữ liệu theo yêu cầu. Mỗi nút trong DHT quản lý một vùng dữ liệu cụ thể và lưu trữ thông tin cục bộ về một số nút trong toàn bộ hệ thống phân tán, phục vụ cho việc trao đổi thông tin định tuyến.

1.4.1.2. Địa chỉ trong bảng băm phân tán

Bảng băm phân tán sử dụng chung một không gian địa chỉ để định danh các nút và dữ liệu trong hệ thống. Không gian địa chỉ là các số nguyên lớn, có phạm vi từ 0 đến $2^{160} - 1$. Mỗi nút tham gia hệ thống chịu trách nhiệm quản lý một phần không gian địa chỉ liên tục nhau. Với một giá trị trong không gian địa chỉ, một nút có thể xác định chính xác nút quản lý một giá trị giá trị này thông qua chức năng tìm kiếm trong DHT.

Các giao diện ứng dụng của bảng băm phân tán trừu tượng hóa từ các chi tiết này và cung cấp các thao tác chung, đơn giản. Hàm $put(k, Data)$ với đầu vào là định danh k của dữ liệu và nội dung dữ liệu ($Data$) được dùng để chuyển dữ liệu đến nút quản lý định danh k . Định danh k và dữ liệu $Data$ được gọi là bộ (*khóa, giá trị*). Hàm $get(k)$ được sử dụng tìm kiếm dữ liệu tương ứng với một định danh k .

1.4.1.3. Định tuyến

Định tuyến là chức năng quan trọng của bảng băm phân tán. Dựa trên thủ tục định tuyến, các thông điệp với định danh ID được chuyển qua các nút DHT đến nút quản lý định danh ID . Các thuật toán định tuyến trong DHT giải quyết vấn đề tìm kiếm.

1.4.1.4. Lưu trữ dữ liệu

Có hai phương thức lưu trữ dữ liệu trong bảng băm phân tán: lưu trữ trực tiếp và lưu trữ gián tiếp. Trong lưu trữ trực tiếp, dữ liệu được lưu trữ tại nút quản lý định danh của dữ liệu. Trong lưu trữ gián tiếp, một tham chiếu đến nút chứa dữ liệu được lưu trữ trong bảng băm phân tán và dữ liệu tự nó được lưu trữ trên nút chèn dữ liệu.

1.4.2. Mạng ngang hàng Chord

Chord là một giao thức mạng ngang hàng có cấu trúc phổ biến nhất được xây dựng dựa trên bảng băm phân tán. Trong Chord, khóa của một tệp tin được quản lý bởi một nút trong mạng. Tùy thuộc vào ứng dụng, dữ liệu liên quan đến khóa có thể được lưu trữ tại nút quản lý khóa. Chord sử dụng phương pháp băm đồng nhất để tạo định danh dữ liệu và định danh của nút.

1.4.2.1. Mô hình của Chord

Các nút trong mạng Chord được tổ chức thành dạng một vòng tròn có kích thước m bit. Định danh của các nút bắt đầu từ 0 đến $2^m - 1$. Khóa k được quản lý bởi nút đầu tiên có định danh bằng hoặc lớn hơn k . Nút đó được gọi là nút successor của khóa k hay $successor(k)$. Successor của một khóa chính là nút gần nhất theo chiều kim đồng hồ tính từ khóa k .

Khi nút n tham gia vào mạng, một số khóa được đặt tại successor của n chuyển sang cho nút n quản lý. Successor của một nút là nút tiếp sau nút đó trên vòng Chord, predecessor là nút liền trước nút đó trên vòng Chord.

1.4.2.2. Tìm kiếm trong mạng Chord

Tìm kiếm đơn giản: Mỗi nút duy trì một liên kết đến nút successor của nó. Truy vấn với khóa k được chuyển từ nút khởi tạo truy vấn đi qua các nút successor cho đến khi gặp nút quản lý khóa k .

Tìm kiếm nâng cao: Chord bổ sung thêm một số thông tin phục vụ việc định tuyến. Với không gian khóa có kích thước m bit, mỗi nút n duy trì một bảng định tuyến gồm m hàng, được gọi là bảng finger.

Với cách tổ chức như vậy, mỗi nút chỉ lưu thông tin về một số giới hạn các nút có trong mạng và chỉ biết một số nút nằm gần với nó.

1.4.2.3. Tham gia và ổn định mạng

Chord sử dụng giao thức *stabilization* để kiểm tra tính hợp lệ và cập nhật con trỏ successor khi một nút tham gia và rời khỏi mạng.

1.4.3. Một số giao thức mạng ngang hàng có cấu trúc khác

1.4.3.1. Giao thức CAN

CAN là giao thức mạng ngang hàng có cấu trúc khác, hoạt động dựa trên cấu trúc bảng băm phân tán. Mỗi nút trong mạng CAN quản lý một vùng (zone) trong không gian định danh và có thông tin về các vùng lân cận trực tiếp. CAN hỗ trợ các thao tác cơ bản như: chèn, tìm kiếm và xóa các thông tin khóa từ bảng băm. CAN sử dụng một không gian tọa độ ảo d -chiều để lưu trữ các giá trị khóa trong bảng băm.

1.4.3.2. Giao thức Pastry

Pastry là một giao thức mạng ngang hàng có cấu trúc được cài đặt dựa trên bảng băm phân tán. Khi tham gia hệ thống, một *nodeID* được gán ngẫu nhiên cho nút. Mỗi nút duy trì một bảng định tuyến đọc theo các nút láng giềng. Khi một nút mới tham gia vào hệ thống nó cập nhật bảng định tuyến và thông báo cho các nút lân cận khác về sự hiện diện của nó. Một nút trong mạng Pastry sử dụng *nodeID* liên quan đến thông điệp để định hướng gói tin đến nút gần nhất.

1.4.4. Vấn đề đảm bảo chất lượng dịch vụ mạng ngang hàng có cấu trúc

1.4.4.1. Chất lượng dịch vụ trong mạng ngang hàng có cấu trúc

Chất lượng dịch vụ trong mạng ngang hàng có cấu trúc được phân loại thành hai nhóm. Một là chất lượng dịch vụ phù hợp với mạng nói chung; Hai là chất lượng dịch vụ dựa trên các đặc điểm của mạng ngang hàng.

Theo nhóm thứ nhất, các tham số được dùng để đánh giá chất lượng dịch vụ cho các ứng dụng khác nhau bao gồm: băng thông, tính chính xác của dữ liệu, việc mất mát dữ liệu, độ trễ truyền tin và sự biến đổi tính thời gian ngẫu nhiên (jitter). Tùy vào từng ứng dụng cụ thể mà các yêu cầu đối với các tham số này cũng khác nhau. Luận án không đề cập đến vấn đề đảm bảo chất lượng dịch vụ theo hướng này.

Theo nhóm thứ hai, chúng ta cần quan tâm đến các đặc tính của mạng ngang hàng để đảm bảo chất lượng dịch vụ cho các ứng dụng. Các yêu cầu đối với các đặc tính của mạng ngang hàng đó là:

1. Khả năng tin cậy để tiếp cận các nút được kết nối mạng.
2. Khả năng khám phá hiệu quả các nguồn tài nguyên mới trong mạng.
3. Khả năng thích nghi với sự thay đổi liên tục của các nút trong mạng.
4. Khả năng mở rộng để đáp ứng các nhu cầu của ứng dụng.

Đây là hướng nghiên cứu chính trong luận án của chúng tôi.

Phần tiếp theo trình bày chất lượng dịch vụ và làm thế nào để đảm bảo chất lượng cho các dịch vụ khác nhau trong mạng ngang hàng có cấu trúc.

1.4.4.2. Đảm bảo chất lượng cho mạng ngang hàng có cấu trúc

Trong mạng ngang hàng có cấu trúc, vấn đề quản lý đã được triển khai để đảm bảo chất lượng dịch vụ cho các ứng dụng. Với phương pháp phân loại lưu lượng tin cậy, nhà cung cấp dịch vụ có thể bảo đảm mức dịch vụ người dùng cuối để thu hút khách hàng.

Các tham số để đánh giá chất lượng dịch vụ tìm kiếm tệp tin trong mạng ngang hàng là số kết quả truy vấn, thời gian phản hồi truy vấn. Đây là một thách thức không nhỏ để có được chất lượng dịch vụ tốt với hiệu năng cao trong thiết kế một hệ thống mạng ngang hàng. Trong mạng chia sẻ tệp tin ngang hàng, nút quản lý khóa của tệp tin và nút chứa tệp tin gốc có thể khác nhau. Yêu cầu tìm kiếm tệp tin được phát ra từ một nút trong mạng, qua giao thức định tuyến yêu cầu đó sẽ đến được nút quản lý khóa của tệp tin. Nút quản lý khóa cung cấp thông tin về nút chứa tệp tin gốc cho nút phát yêu cầu tìm kiếm, sau đó nút tìm kiếm tệp tin liên hệ trực tiếp với nút quản lý tệp tin để lấy tệp tin về.

Luận án đề xuất hai giải pháp nâng cao tỷ lệ thành công của các câu truy vấn. Một là giải pháp cân bằng tải sử dụng phương pháp dịch chuyển định danh của các nút để đảm bảo cân bằng tải. Hai là giải pháp điều khiển tắc nghẽn để hướng các câu truy vấn đi theo con đường tốt hơn đến đích.

Sao lưu tệp tin đến nhiều nút là một giải pháp hiệu quả để nâng cao tính sẵn sàng cho dữ liệu trong môi trường mạng ngang hàng. Với mục đích cung cấp chất lượng dịch vụ tốt cho người dùng với chi phí sao lưu thấp, chúng tôi đề xuất một giải pháp sao lưu dữ liệu để đảm bảo tính sẵn sàng cho dữ liệu đảm bảo kiểm soát được số bản sao và chi phí sao lưu thấp.

1.5. Kết luận

Chương 1 trình bày các kiến thức hết sức cơ bản phục vụ cho việc nghiên cứu đảm bảo chất lượng dịch vụ cho mạng ngang hàng và đưa ra các vấn đề để giải quyết trong các chương tiếp theo.

Trong các chương tiếp theo của luận án mỗi chương giải quyết một vấn đề được chúng tôi đưa ra trong chương 1. Các vấn đề này được giải quyết sẽ là nhân tố rất quan trọng để nâng cao chất lượng dịch vụ cho mạng ngang hàng có cấu trúc nói chung và ứng dụng chia sẻ tệp tin ngang hàng nói riêng.

Chương 2. CÂN BẰNG TẢI TRONG MẠNG NGANG HÀNG CÓ CẤU TRÚC

2.1. Đặt vấn đề

Mạng ngang hàng dựa trên DHT có nhiều ưu điểm trong việc tổ chức lưu trữ và tìm kiếm dữ liệu, song cũng tồn tại không ít những nhược điểm. Một trong những vấn đề cần giải quyết trong các mạng DHT là làm thế nào để cân bằng tải giữa các nút tham gia hệ thống.

Có nhiều yếu tố gây mất cân bằng tải giữa các nút trong mạng ngang hàng có cấu trúc, bao gồm: khả năng xử lý câu truy vấn của các nút không giống nhau; số lượng các khóa do các nút quản lý cũng khác nhau; việc lựa chọn ngẫu nhiên định danh của các nút và của dữ liệu; tính phổ biến của các tệp tin, sự ra/vào hệ thống của các nút (nút churn) ..v.v.

Đã có nhiều nghiên cứu về cân bằng tải được các tác giả đưa ra cho hệ thống mạng ngang hàng dựa trên DHT, các giải pháp này được phân thành hai nhóm chính:

(a) Theo hướng sử dụng khái niệm server ảo: Mỗi nút vật lý quản lý một hoặc nhiều server ảo với các định danh ngẫu nhiên được chọn từ không gian định danh. Các server ảo hoạt động như các nút tham gia mạng DHT. Thông thường mỗi nút vật lý cần duy trì $O(\log n)$ server ảo để làm giảm nhân tố mất cân bằng xuống đến một hằng số. Với một hệ thống DHT gồm nhiều nút có khả năng chịu tải khác nhau, mỗi nút vật lý sẽ chọn một số lượng server ảo tỷ lệ với khả năng của nó.

(b) Theo hướng không sử dụng khái niệm server ảo: Để đạt được sự cân bằng trong hệ thống, các giải pháp đưa ra đều phải dịch chuyển định danh của các nút trong hệ thống khi có một nút tham gia hoặc rời khỏi hệ thống.

2.2. Các nghiên cứu liên quan

2.2.1. Cân bằng tải theo ngưỡng

Thuật toán cân bằng tải theo ngưỡng luôn duy trì tải của mỗi nút dưới một ngưỡng δ nào đó. Một nút trong mạng sẽ cố gắng chuyển tải cho các nút khác khi tải của nó tăng quá ngưỡng δ .

Khi tải của một nút n trong mạng vượt quá một ngưỡng T_j thì đầu tiên nó cố gắng chuyển tải cho một trong hai nút láng giềng có tải nhỏ hơn. Nếu cả hai nút láng giềng đều có tải lớn hơn và không thể nhận được tải nữa thì nó tìm một nút nhẹ tải trong mạng và có tải nhỏ nhất, nhờ nút này nhận tải hộ bằng cách dịch chuyển định danh của nút nhẹ tải vừa tìm được vào giữa nút n và *predecessor* của n .

2.2.2. Cân bằng tải dựa trên server ảo.

Mỗi server ảo có vai trò như một nút trong mạng DHT. Một nút vật lý có thể quản lý một hoặc nhiều server ảo. Server ảo chịu trách nhiệm quản lý một vùng không gian định danh liên tục nhau. Mỗi nút vật lý có thể quản lý các vùng không gian định danh không liên tục nhau.

Cân bằng tải dựa trên server ảo, một nút vật lý căn cứ vào khả năng xử lý truy vấn để tạo ra số lượng server ảo cho phù hợp khi gia vào mạng để tạo ra sự cân bằng giữa các nút (thuật toán $\log(N)$ virtual server) hoặc căn cứ vào tình trạng tải xử lý truy vấn của một nút để điều chỉnh số lượng server ảo cho phù hợp. Khi tải của một nút ở mức thấp, nút đó tạo thêm một số server ảo để nhận thêm tải cho hệ thống. Khi một nút quá tải, nó thực hiện việc xóa bớt một số server ảo để làm giảm tải cho nút đó (thuật toán Proportion). Trong thuật toán Transfer, một nút tạo ra một số lượng server ảo phù hợp khi tham gia hệ thống và dịch chuyển

server ảo giữa các nút để đảm bảo cân bằng tải. Có ba sơ đồ dịch chuyển server ảo được đưa ra đó là: sơ đồ One-to-One, One-to-Many và Many-to-Many.

Sơ đồ "One-to-One": các server ảo từ nút nặng tải được chuyển nút nhẹ tải để làm giảm tải cho nút nặng tải đồng thời không làm quá tải cho nút nhẹ tải.

Sơ đồ "One-to-Many", một thư mục D là một nút trong hệ thống được lựa chọn ngẫu nhiên để lưu thông tin về các nút nhẹ tải. Định kỳ mỗi nút nhẹ tải sẽ gửi thông tin về tải của mình cho thư mục D quản lý. Các nút nặng tải sẽ hỏi thư mục D để biết thông tin về các nút nhẹ tải. Tiếp theo, một số server ảo của nút vật lý nặng tải sẽ được di chuyển tới một hoặc nhiều nút nhẹ tải đã đăng kí trong D .

Sơ đồ Many-to-Many: Sơ đồ này ánh xạ nhiều nút tải nặng sang nhiều nút tải nhẹ. Để nhiều nút tải nặng và nhiều nút nhẹ tương tác được với nhau, một vùng máy chủ ảo toàn cục (pool) được tạo ra - một bước trung gian trong việc di chuyển một máy chủ ảo từ một nút tải nặng đến một nút tải nhẹ. Vùng này là một cấu trúc dữ liệu cục bộ sử dụng để cho việc thực hiện cân bằng tải.

2.3. Cải tiến thuật toán cân bằng tải theo ngưỡng

2.3.1. Một số khái niệm

Xét hệ thống gồm n nút vật lý (các máy tính hoặc các thành phần trong một hệ thống P2P). Nút n_i có khả năng cố định là C_i , tương ứng với số lượng tối đa các câu truy vấn mà nút đó có thể xử lý. Tại một thời điểm cụ thể, nút n_i có tải làm việc (work load) W_i (số câu truy vấn phải xử lý).

Hệ số sử dụng u_i của một nút là tỷ lệ giữa tải làm việc trên khả năng của nó: $u_i = W_i / C_i$. Hệ số sử dụng μ của toàn bộ hệ thống bằng tỷ lệ của tổng tải làm việc của các nút trên tổng khả năng của các nút:

$$\mu = \frac{\sum_{node\ n} W_n}{\sum_{node\ n} C_n}$$

Khi $u_n > 1$, ta nói nút n là quá tải; trong trường hợp ngược lại nút n được gọi là nhẹ tải. Một nút quá tải sẽ không có khả năng xử lý các câu truy vấn.

Trong hệ thống có một số nút đặc biệt gọi là thư mục (directory). Nút thư mục là một nút trong mạng có bổ sung thêm cơ chế quản lý thông tin về các nút nhẹ tải có thể di chuyển được. Một nút nhẹ tải n được gọi là di chuyển được nếu định danh của n dịch chuyển sang một vị trí khác nằm ngoài khoảng không gian định danh từ đến successor của n mà không làm successor của nút n quá tải.

2.3.2. Thuật toán ThresholdPlus

Thuật toán gồm ba bước dưới đây:

Bước 1: Nút nhẹ tải thông báo cho các thư mục

Hệ thống gồm một số nút được chọn ngẫu nhiên làm thư mục để lưu trữ thông tin về các nút nhẹ tải có thể dịch chuyển được.

Định kỳ trong khoảng thời gian, các nút sẽ kiểm tra tải của mình. Nếu một nút có định danh L là nhẹ tải, nó sẽ gửi một thông điệp cho successor của nó để hỏi xem nó có thể di chuyển sang định danh khác được không. Nếu nút L có thể di chuyển được, nó sẽ gửi thông báo chứa thông tin của nó cho nút phụ trách thư mục mà nó đăng kí.

Bước 2: Chia sẻ tải cho các nút láng giềng

Khi một nút n trong hệ thống bị quá tải, nó sẽ thực hiện quá trình cân bằng tải như sau. Trước hết nó hỏi hai nút láng giềng (*predecessor* của n và *successor* của n) xem có thể nhận tải được hộ không. Nếu được nó sẽ chia tải cho một trong hai nút trong láng giềng của n có tải thấp hơn.

Bước 3: Định danh lại một nút

Trong trường hợp không thể chia sẻ tải cho hai nút láng giềng B, C , nút A sẽ hỏi thư mục nó đăng ký và thư mục sẽ gửi cho nút A thông tin về nút nhẹ tải có thể di chuyển được.

Dựa trên thông tin trả lời từ thư mục, nút A sẽ yêu cầu nút nhẹ tải mà nó biết dịch chuyển định danh về vị trí giữa *predecessor*(A) và A để nhận tải hộ nút A . Nếu không tìm được nút nhẹ tải nào thỏa mãn, nút A sẽ hỏi các thư mục còn lại.

Thuật toán được mô tả bằng đoạn giả mã dưới đây.

THRESHOLDPLUS(v, t)

```
1   Các nút nhẹ tải thông báo tình trạng tải cho thư mục
2    $v.level_t \leftarrow$  ngưỡng của nút  $v$  tại thời điểm  $t$ 
2   if  $v.level_t \leq v.level_{t-1}$  then
3       return
4    $v' \leftarrow$  láng giềng với ngưỡng nhỏ nhất
5   if  $v'.level_t < v.level_t$  then
6       chuyển tải từ  $v$  sang  $v'$ 
7   else
8       Hỏi thư mục để tìm nút có tải nhẹ
9        $s \leftarrow$  nút có tải nhỏ nhất
10      Chuyển định danh của  $s$  vào giữa  $Pred(v)$  và nút  $v$ 
```

2.4. Đánh giá giải pháp

2.4.1. Phương pháp đánh giá

Chúng tôi sử dụng chương trình mô phỏng để đánh giá hiệu quả của thuật toán *ThresholdPlus* và so sánh với thuật toán cân bằng tải theo ngưỡng do Ganesan đề xuất.

Chương trình mô phỏng được phát triển từ bộ mô phỏng của Jonathan Ledlie và hoạt động theo các bước thời gian rời rạc. Mỗi bước thời gian gồm ba kịch bản: nút tham gia và rời hệ thống, cập nhật bảng tìm đường, truy vấn dữ liệu và thực hiện cân bằng tải. Trong các thí nghiệm mô phỏng có lựa chọn 10 thư mục và lưu trữ ở 10 nút ngẫu nhiên trong mạng gồm 4096 nút vật lý. Thời gian sống trung bình của các nút trong mạng là 15 phút, 30 phút, 1 giờ, 2 giờ và 3 giờ. Mô phỏng được thực hiện trong khoảng thời gian là 3 giờ. Mỗi nút mới tham gia vào hệ thống được khởi tạo một bảng tìm đường rỗng. Các nút dựa theo cơ chế Chord để điền thông tin trong $\log(n)$ hàng của bảng tìm đường.

Các nút khởi tạo các câu truy vấn một cách ngẫu nhiên từ một tập khóa truy vấn theo phân bố dạng Uniform hoặc phân bố dạng Zipf. Định kỳ trong khoảng thời gian trung bình 30 giây, các nút kiểm tra ngưỡng tải và thực hiện cân bằng tải nếu tải của nút vượt quá ngưỡng cho phép.

2.4.2. Các kết quả mô phỏng

2.4.2.1. Ảnh hưởng thời gian sống tới các thuật toán cân bằng tải.

Trong thí nghiệm này thời gian sống trung bình của các nút là 15 phút, 30 phút, 1 giờ, 2 giờ và 4 giờ. Mỗi lần chạy thí nghiệm, khả năng của một nút trong mạng được giữ cố định, mỗi nút được khởi tạo trung bình 10 câu truy vấn trong một giây. Truy vấn được khởi tạo theo dạng Uniform và dạng Zipf với tham số $\alpha=1.2$.

Kết quả của thí nghiệm cho thấy với số truy vấn trung bình 10 truy vấn/nút, thuật toán của đề xuất có tỷ lệ thành công lớn hơn thuật toán Threshold khoảng 9% đối với truy vấn dạng Uniform và khoảng 14% đối với truy vấn dạng Zipf.

2.4.2.2. Ảnh hưởng số lượng câu truy vấn tới thuật toán cân bằng tải

Trong thí nghiệm khả năng xử lý truy vấn của các nút được giữ cố định, thời gian sống trung bình của các nút trong mạng là 2 giờ, các câu truy vấn đặt vào hệ thống được phân bổ dưới dạng Uniform và dạng Zipf với $\alpha=1.2$. Số lượng trung bình các câu truy vấn đặt vào một nút trong hệ thống tăng 0.01/giây truy vấn đến 100 truy vấn/giây.

Kết quả thí nghiệm cho thấy thuật toán đề xuất có khả năng thích nghi tốt hơn thuật toán *Threshold* khoảng 6% đối với truy vấn dạng Uniform và khoảng 10% đối với truy vấn dạng Zipf.

2.4.2.3. Ảnh hưởng câu truy vấn dạng Zipf tới thuật toán cân bằng tải

Các tham số khả năng của các nút trong mỗi lần chạy thí nghiệm được giữ không đổi, thời gian sống trung bình của một nút trong mạng là 2 giờ, số truy vấn trung bình đặt vào mỗi nút là 10 truy vấn/giây. Truy vấn phân bổ cho các nút trong mạng theo dạng Zipf với tham số α thay đổi ($\alpha=0.8, 1.2, 2.4$ và 4.8).

Kết quả thí nghiệm cho thấy thuật toán *ThresholdPlus* có tỷ lệ thành công lớn hơn thuật toán *Threshold* khoảng 6%.

2.5. Kết luận

Chương 2 của luận án đã trình bày một số liên cứu liên quan về cân bằng tải trong mạng ngang hàng có cấu trúc để nâng cao khả năng cân bằng tải xử lý các câu truy vấn của các nút trong mạng, tăng tỷ lệ thành công của các câu truy vấn và đề xuất một thuật toán cải tiến thuật toán cân bằng tải theo ngưỡng.

Các kết quả mô phỏng và đánh giá cho thấy trong điều kiện mạng gần với thực tế (khả năng của các nút không giống nhau, các câu truy vấn đặt vào các nút không đồng đều, thời gian sống của các nút không giống nhau) thuật toán đề xuất hoạt động tốt hơn so với thuật toán cân bằng tải do Ganesan đề xuất.

Tiếp theo, chúng tôi nghiên cứu để xử lý vấn đề truy vấn trong khoảng và nâng cao hiệu quả hoạt động của thư mục.

Chương 3. ĐIỀU KHIỂN TẮC NGHẼN TRONG P2P

3.1. Đặt vấn đề

Trong những năm gần đây, DHT không chỉ được sử dụng cho các ứng dụng chia sẻ tệp tin mà còn được trong xây dựng hệ thống truy vấn thông tin ngang hàng (P2P-IR). Trong các hệ thống P2P-IR, DHT phải xử lý rất nhiều gói tin trong thời gian ngắn, làm tăng đáng kể tải hệ thống. Tối ưu hóa phương pháp định tuyến trong các mạng DHT là một yêu cầu quan trọng đối với các hệ thống P2P-IR. Hơn nữa, khi một nút nhận được một số truy vấn vượt quá khả năng định tuyến của nó sẽ xảy ra sự cố tắc nghẽn trong mạng. Nếu không có cơ chế kiểm soát tắc nghẽn thích hợp, các truy vấn tiếp theo đến hoặc đi qua nút này sẽ không đến được đích.

Điều khiển tắc nghẽn trong DHT hoạt động ở lớp trên và độc lập với các cơ chế điều khiển tắc nghẽn của TCP. DHT thường sử dụng các thuật toán tham lam để định tuyến các gói tin trong mạng, trong đó một nút luôn lựa chọn nút láng giềng gần nhất để chuyển tiếp gói tin tới nút đích. Các mạng P2P là không đồng nhất, do đó khả năng xử lý gói tin của các nút sẽ không giống nhau, độ ổn định của các nút không cao và các nút phải chia sẻ CPU và tài nguyên mạng cho các tiến trình khác đang chạy trong cùng thời điểm đó,..v.v. Do đó, việc chọn các nút lân cận để chuyển tiếp các gói tin đến đích có thể không đáp ứng với sự thay đổi tải hệ thống trong mạng DHT.

Trong chương này, chúng tôi đề xuất một thuật toán điều khiển tắc nghẽn trong mạng ngang hàng có cấu trúc bằng cách thay đổi bảng định tuyến của mỗi nút dựa trên trạng thái của các nút tiếp theo trong mỗi đường định tuyến. Giải pháp đề xuất được đánh giá thông qua một số thí nghiệm mô phỏng và so sánh với thuật toán Chord.

3.2. Các nghiên cứu liên quan

Các nghiên cứu liên quan đến điều khiển tắc nghẽn trong mạng ngang hàng có cấu trúc chủ yếu tập trung vào hai hướng: kiểm soát tốc độ và điều khiển bảng định tuyến. Trong hướng kiểm soát tốc độ, tốc độ gửi truy vấn bị giới hạn ở nút truy vấn khi một nút trong đường định tuyến bị quá tải. Các phương pháp tiêu biểu cho hướng này gồm: phương pháp CSCC (Credit System Congestion Control), phương pháp BPCC (Back-Pressure Congestion Control) và phương pháp Marking.

Trong phương pháp CSCC, khi một nút đích nhận được một gói tin, nó trả lời nút truy vấn bằng một gói tin *ACK*. Mỗi nút duy trì một hàng đợi để lưu trữ các truy vấn đến. Khi hàng đợi đầy, các truy vấn gửi đến tiếp theo sẽ bị loại bỏ mà không thông báo cho các nút gửi truy vấn.

Trong phương pháp BPCC, mỗi nút duy trì một hàng đợi cho các truy vấn gửi đến. Khi nó nhận được một gói tin, nếu hàng đợi đạt đến giới hạn, nó sẽ gửi trạng thái hàng đợi của nó tới người gửi. Nút gửi sẽ tự động điều chỉnh tốc độ gửi để tránh tắc nghẽn.

Trong phương pháp Marking, mỗi nút tổ chức một hàng đợi cho các truy vấn gửi đến và trả về trạng thái nghẽn bằng cách thêm một cờ *h* vào tiêu đề gói tin. Mục đích là để giữ các nút được chia sẻ tài nguyên với các nút khác đang bị nút cổ chai một cách công bằng mà không làm quá tải nút đang tắc nghẽn. Theo mặc định, cờ *h* bị tắt khi gói tin truy vấn được tạo. Cờ *h* tắt cho biết trạng thái mạng không tắc nghẽn và ngược lại.

Theo một hướng nghiên cứu khác, phương thức bảng định tuyến được dựa trên trạng thái của mỗi nút. Cách tiếp cận này thay đổi đường đi của gói tin đến các nút ít tắc nghẽn hơn. Trong phương pháp Adaptive

Routing, thay vì chọn nút gần nhất với khóa, nó chọn k nút gần khóa và theo dõi hiệu năng của các nút đó để xác định đường đi tốt nhất. Với phương pháp Adaptive Routing là khi muốn tăng số lượng đường định tuyến ta phải tăng kích thước của bảng định tuyến, điều này sẽ tiêu tốn tài nguyên của hệ thống.

3.3. Điều khiển tắc nghẽn bằng thay đổi bảng định tuyến (phương pháp CA-Chord)

Phương pháp CA-Chord hoạt động theo các bước sau:

Bước 1: Phát hiện tắc nghẽn

Mỗi nút trong hệ thống có khả năng định tuyến C là số tối đa các thông báo truy vấn mà nút đó có thể định tuyến trong một đơn vị thời gian.

Nếu tốc độ thông báo đến một nút lớn hơn khả năng định tuyến của nút đó thì nó sẽ không xử lý thêm bất kỳ một gói tin nào nữa và các gói tin đến sẽ bị loại bỏ. Trạng thái này được gọi là trạng thái "*tắc nghẽn cứng*". Trạng thái "*tắc nghẽn cứng*" của một nút cho biết nút đó bị tắc nghẽn hoàn toàn, không còn khả năng xử lý gói tin.

Một tham số T gọi là ngưỡng tắc nghẽn mềm cho biết trạng thái tắc nghẽn của một nút. Ngưỡng tắc nghẽn mềm được tính toán theo công thức (2) dưới đây:

$$T = p * C \quad (2)$$

với p là một tham số hệ thống và $0 < p < 1$.

Khi tốc độ gói tin gửi đến của một nút đạt đến ngưỡng tắc nghẽn mềm, nút đó được xem là đi vào trạng thái bị tắc nghẽn. Trong trường hợp này, nút sẽ kích hoạt thủ tục xử lý tránh tắc nghẽn, tuy nhiên các gói tin đến vẫn được xử lý như bình thường. Trạng thái này được gọi là "*trạng thái tắc nghẽn mềm*".

Mỗi nút duy trì một danh sách các nút láng giềng kế tiếp trong không gian định danh ở trạng thái không tắc nghẽn. Khi một nút thay đổi từ trạng thái không tắc nghẽn sang trạng thái tắc nghẽn hoặc ngược lại, nó sẽ thông báo trạng thái của nó cho các nút trong danh sách nút phía trước nó (như trong giao thức Chord thông thường). Các nút này sẽ cập nhật danh sách các nút láng giềng kế tiếp không bị tắc nghẽn khi chúng nhận được thông báo.

Bước 2: Xử lý tắc nghẽn

Khi một nút trong trạng thái "*tắc nghẽn mềm*" nhận được một gói tin mới, nó sẽ thực hiện các tác vụ sau để tránh định tuyến gói tin đi qua các nút tắc nghẽn:

- Nút tắc nghẽn sẽ gửi thông báo tắc nghẽn tới nút gửi gói tin để thông báo trạng thái tắc nghẽn của nó và thông tin về nút thay thế. Nút đầu tiên trong danh sách các nút láng giềng kế tiếp không bị tắc nghẽn được chọn làm nút thay thế. Bất cứ khi nào một gói tin được gửi đến nút tắc nghẽn, nếu nút gửi không nhận được bất kỳ thông báo nghẽn nào, nút tắc nghẽn sẽ gửi một thông báo nghẽn tới nút gửi và lưu thông tin của nút gửi vào cơ sở dữ liệu của nó để hạn chế việc gửi các gói tin thông báo không cần thiết.

- Khi nút gửi nhận được gói tin thông báo tắc nghẽn, nó thay thế nút tắc nghẽn trong bảng định tuyến bằng nút thay thế được chọn. Do đó, các gói tin mới sẽ được chuyển đi theo đường định tuyến mới.

Bước 3: Xử lý sau khi hết tắc nghẽn

Khi trạng thái của nút thay đổi từ trạng thái tắc nghẽn sang trạng thái thông thường (nghĩa là số lượng gói tin đến trong một đơn vị thời gian nhỏ hơn ngưỡng tắc nghẽn T), các thao tác sau đây được thực hiện:

- Nút vừa ra khỏi trạng thái nghẽn gửi thông báo hết tắc nghẽn tới các nút đã nhận được thông báo nghẽn.

- Nút nhận thông báo hết tắc nghẽn thay đổi tuyến đường đang hoạt động trong bảng định tuyến trở lại tuyến đường ban đầu.

3.4. Đánh giá giải pháp

3.4.1. Phương pháp đánh giá

Chúng tôi thực hiện một số thí nghiệm mô phỏng và so sánh với thuật toán định tuyến trong giao thức Chord. Các thí nghiệm thực hiện trong điều kiện mô phỏng một hệ thống mạng gắn với hệ thống mạng thực tế, gồm 4096 nút vật lý và hoạt động theo thời gian rời rạc, sử dụng các tham số đầu vào của J. Ledlie. Mỗi bước thời gian gồm các kịch bản: nút tham gia và rời hệ thống, cập nhật bảng tìm đường, truy vấn dữ liệu. Trong các thí nghiệm, ngưỡng tắc nghẽn T được đặt bằng 50% khả năng định tuyến của một nút.

3.4.2. Các kết quả mô phỏng

3.4.2.1. Đánh giá tỷ lệ truy vấn thành công

Trong thí nghiệm đầu tiên, các khóa truy vấn đặt vào một nút theo hai dạng Uniform và Zipf với tham số $\alpha = 0,8$. Trong mỗi vòng của thí nghiệm, chúng tôi đặt 20 truy vấn vào một nút. Thời gian sống trung bình của một nút tương ứng là 15 phút, 30 phút, 60 phút, 120 phút và 180 phút.

Kết quả thí nghiệm cho thấy tỷ lệ thành công của thuật toán CA-Chord do chúng tôi đề xuất cao hơn thuật toán Chord 42% đối với các truy vấn dạng Uniform và 37% đối với các truy vấn dạng Zipf.

Thí nghiệm thứ hai chúng tôi đánh giá tỷ lệ thành công của các câu truy vấn dữ liệu khi thay đổi số lượng truy vấn trung bình được đặt vào một nút. Thời gian sống trung bình của một nút được giữ cố định là 1 giờ. Các khóa truy vấn được tạo ra theo phân bố Uniform và Zipf với $\alpha = 0,8$.

Kết quả thí nghiệm cho thấy, tỷ lệ truy vấn thành công của thuật toán đề xuất tốt hơn thuật toán Chord là 32% và 41% tương ứng với truy vấn Zipf và Uniform.

Thí nghiệm thứ ba được tiến hành để đánh giá ảnh hưởng của ngưỡng tắc nghẽn T đến tỷ lệ truy vấn thành công. Số lượng truy vấn trung bình đặt vào một nút là 20 truy vấn/giây, thời gian sống trung bình của một nút là 1 giờ. Ngưỡng tắc nghẽn T được thay đổi để xử lý tắc nghẽn trong một nút. Các khóa truy vấn đặt vào một nút được thực hiện theo hai dạng: phân bố Uniform và phân bố Zipf với $\alpha = 0,8$. Kết quả của thí nghiệm cho thấy CA-Chord đạt được hiệu suất tốt nhất khi giá trị của T là 50% khả năng định tuyến của một nút.

Thí nghiệm thứ tư chúng tôi đánh giá khả năng đáp ứng của các thuật toán đối với các truy vấn dạng Zipf. Số lượng các câu truy vấn đặt vào một nút trung bình là 20 truy vấn/giây, thời gian sống trung bình của một nút là 1 giờ. Các khóa truy vấn được tạo bởi phân bố xác suất Zipf với giá trị α là 0,8, 1,2, 2,4 và 4,8.

Các kết quả của thí nghiệm cho thấy thuật toán đề xuất hoạt động tốt hơn thuật toán Chord với truy vấn Zipf khoảng 20%.

3.4.2.2. Đánh giá số bước chuyển tiếp truy vấn

Các thí nghiệm tiếp theo chúng tôi đánh giá số bước (tức là các nút) cần thiết để chuyển tiếp truy vấn. Thời gian sống trung bình của một nút được giữ cố định là 1 giờ. Số lượng câu truy vấn trung bình đặt vào một nút thay đổi tương ứng là 0,01; 0,1; 1; 10; 100 truy vấn/giây. Khóa truy vấn được phân bố theo hai dạng Uniform và Zipf với $\alpha=0,8$.

Các kết quả số bước chuyển tiếp câu truy vấn của thuật toán CA-Chord nhỏ hơn so với thuật toán Chord.

Thứ hai, chúng tôi giữ cố định số lượng trung bình các câu truy vấn đặt vào một nút là 20 truy vấn/giây. Thời gian sống trung bình của nút thay đổi tương ứng với các giá trị 15 phút, 30 phút, 60 phút, 120 phút và 180 phút. Khóa truy vấn đặt vào một nút được phân bố theo dạng Uniform và dạng Zipf với $\alpha = 0,8$.

Kết quả của thí nghiệm cho thấy khi thời gian sống trung bình của một nút trong mạng thay đổi số bước để chuyển tiếp truy vấn trong thuật toán CA-Chord đều nhỏ hơn thuật toán Chord.

3.4.2.3. Đánh giá số thông báo tắc nghẽn được gửi

Thí nghiệm tiếp theo được thực hiện để đánh giá số lượng thông báo được gửi bởi một nút trong trường hợp tắc nghẽn. Thời gian sống trung bình của nút là 1 giờ. Số lượng truy vấn trung bình đặt vào một nút thay đổi với các giá trị tương ứng là 0,01; 0,1; 1; 10; 100 truy vấn/giây. Khóa truy vấn được phân bố theo dạng Uniform và Zipf với $\alpha = 0,8$.

Kết quả của thí nghiệm cho thấy khi số lượng truy vấn đặt vào một nút số lượng thông báo tắc nghẽn của truy vấn dạng Uniform cao hơn số lượng thông báo tắc nghẽn của truy vấn dạng Zipf.

3.5. Kết luận

Chúng tôi đã đề xuất một giải pháp mới để điều khiển tắc nghẽn trong các mạng ngang hàng có cấu trúc. Giải pháp được đề xuất mang lại tỷ lệ thành công truy vấn tốt hơn và giới hạn số lượng băng thông truy cập trong hệ thống. Do đó nó thể hiện khả năng giải quyết tắc nghẽn cục bộ và tăng thông lượng trong mạng. Kết quả đánh giá cho thấy rằng giải pháp đề xuất của chúng tôi hoạt động tốt hơn thuật toán Chord thông thường. Tuy nhiên, phương pháp của chúng tôi vẫn giới hạn các nút thay thế cho các nút liền kề của nút tắc nghẽn mà không xem xét khả năng chịu trách nhiệm của nút khi chọn nút thay thế. Ngoài ra, các thay đổi có thể tiếp tục nếu nút đích mới bị tắc nghẽn.

Chương 4. SAO LƯU DỮ LIỆU TRONG MẠNG MẠNG NGANG HÀNG CÓ CẤU TRÚC

4.1. Đặt vấn đề

Với sự phát triển nhanh chóng của Internet, ngày nay nhiều tài nguyên máy tính như khả năng lưu trữ, băng thông mạng, sức mạnh của CPU...v.v. có thể được chia sẻ, sử dụng chung giữa nhiều người dùng thông qua các ứng dụng mạng ngang hàng. Các ứng dụng này cho phép người dùng lưu trữ và chia sẻ các tệp dữ liệu một cách dễ dàng. Các hệ thống chia sẻ tệp tin sử dụng bảng băm phân tán (DHT) cung cấp cơ chế hiệu quả cho việc sắp xếp và tìm kiếm dữ liệu. Khi một nút có yêu cầu sao lưu tệp dữ liệu, nó tạo ra một khóa DHT từ nội dung của tệp dữ liệu và gửi bản sao của tệp dữ liệu tới nút quản lý khóa DHT đó. Khi một nút muốn tìm kiếm một tệp dữ liệu, nó sẽ gửi yêu cầu truy vấn đến nút quản lý khóa DHT của tệp dữ liệu dựa trên thuật toán định tuyến DHT. Nút quản lý khóa sẽ trả lại bản sao của tệp dữ liệu. Tuy nhiên, trong mạng P2P các nút thường xuyên ra/vào mạng làm cho các tệp dữ liệu lưu trữ trên các nút đó bị mất đi, vì vậy cần phải có một cơ chế để duy trì tính sẵn sàng của tệp dữ liệu trong những trường hợp như vậy. Ngoài ra, khả năng lưu trữ của các nút trong mạng P2P rất khác nhau, điều này dẫn đến sự mất cân bằng tải lưu trữ giữa các nút trong mạng.

Chương này đề xuất một số giải pháp sao lưu dữ liệu trong hệ thống chia sẻ tệp tin ngang hàng. Giải pháp đề xuất hoạt động dựa trên phân cụm không gian khóa DHT để đảm bảo tính sẵn sàng của dữ liệu và hỗ trợ cân bằng tải cho hệ thống. Giải pháp sao lưu dữ liệu đề xuất còn đảm bảo các nút gần nhau về mặt vật lý sẽ tham gia vào cùng một cụm. Do đó, thời gian để cập nhật thông tin cụm và chi phí để duy trì dữ liệu trong cụm được giảm.

Kết quả đánh giá cho thấy giải pháp đề xuất hiệu quả hơn rất nhiều so với các giải pháp thông thường. Tỷ lệ truy vấn dữ liệu thành công cao trong khi chi phí lưu trữ dữ liệu và chi phí duy trì các mảnh dữ liệu nhỏ. Các nút trong hệ thống đạt được trạng thái cân bằng tốt hơn, thời gian cập nhật thông tin cụm giảm...v.v.

4.2. Các nghiên cứu liên quan

Có ba cách tiếp cận cơ bản cho việc sao lưu dữ liệu trong mạng ngang hàng có cấu trúc dựa trên DHT. Trong cách tiếp cận sao lưu tại các nút láng giềng, bản sao của tệp tin được lưu trữ tại các nút lân cận gần nhất, tức là danh sách nút successor hoặc leaf-set của nút quản lý tệp tin. Các kết quả thí nghiệm cho thấy, khi tỷ lệ nút ra/vào hệ thống lớn hoặc tải của hệ thống cao thì cách tiếp cận này kém hiệu quả trong việc duy trì tính sẵn sàng của một tệp tin. Hơn nữa, chi phí di chuyển dữ liệu thường cao, mức độ tiêu tốn băng thông của hệ thống tỷ lệ thuận với mức độ ra/vào của các nút. Giải pháp RelaxDHT hạn chế yêu cầu sao lưu dữ liệu ở các nút láng giềng gần nhất bằng cách lựa chọn ngẫu nhiên các nút sao lưu từ tập các nút láng giềng. Cách tiếp cận này có thể làm giảm chi phí di chuyển dữ liệu. Tuy nhiên, khi tải hệ thống cao, cách tiếp cận này vẫn tạo ra vấn đề mất cân bằng lưu trữ giữa các nút trong mạng, dẫn đến tỷ lệ truy vấn thấp.

Sao lưu dữ liệu theo đường đi là một giải pháp sao lưu dữ liệu khác. Trong giải pháp này, các nút sao lưu được chọn dọc theo đường đi của truy vấn từ nút truy vấn đến nút được truy vấn khi nút được truy vấn có tải cao. Cách tiếp cận này làm giảm tải cho các nút được truy vấn. Tuy nhiên, nó không đảm bảo tính sẵn sàng của dữ liệu khi hệ thống thường xuyên có các nút ra/vào.

Cách tiếp cận sao lưu cuối cùng là sao lưu nhiều khóa. Trong cách tiếp cận này, một khóa có liên hệ với một tệp gồm r định danh được lựa chọn trong không gian khóa DHT tương ứng với r nút sao lưu cho một tệp tin. Cách tiếp cận này đảm bảo yêu cầu cân bằng tải truy vấn bằng cách gửi yêu cầu truy vấn đến nút lưu

trữ bản sao một cách ngẫu nhiên. Tuy nhiên, để duy trì tính ổn định vị trí của các sao lưu, chi phí duy trì số bản sao để đảm bảo tính sẵn sàng của dữ liệu cũng như chi phí di chuyển dữ liệu hệ thống khi có các nút ra/vào là tương đối cao.

Một giải pháp sao lưu có cách tiếp cận tương tự với thuật toán đề xuất đó là Plover. Trong Plover, các nút được tổ chức thành một số cụm. Mỗi cụm bao gồm một supernode và một số nút thông thường, gắn với nhau về mặt vật lý. Các supernode tạo thành mạng P2P có cấu trúc và quản lý siêu dữ liệu (metadata), các tệp tin được lưu trữ trong các nút thông thường. Khi một tệp tin có tần suất truy vấn cao, Plover tạo ra bản sao giữa các nút gắn nhau về mặt vật lý dựa trên khả năng lưu trữ sẵn có của các nút.

SWARM phân chia các nút trong mạng thành các cụm dựa trên khoảng cách và sự quan tâm của nút. Vị trí đặt các bản sao được xác định thông qua tỷ lệ truy vấn được tích lũy của một nút để giảm việc dư thừa các truy vấn. Tuy nhiên, SWARM không đảm bảo tính sẵn sàng của các tệp tin khi trong mạng có sự ra/vào của các nút.

Nghiên cứu Friedman et al. sử dụng mã xóa (REC - Replicated Easure Code) để lưu trữ các bản sao dữ liệu trong mạng P2P. Giải pháp này không làm tăng quá nhiều lượng thông tin dư thừa trong việc lưu trữ dữ liệu và làm giảm chi phí cho các hoạt động duy trì. Trong giải pháp đề xuất, chúng tôi cũng sử dụng REC để làm giảm chi phí duy trì các mảnh dữ liệu và nâng cao hiệu quả cho hệ thống.

4.3. Sao lưu dữ liệu dựa trên phân cụm trong mạng P2P

4.3.1 Tổng quan

Sơ đồ sao lưu đề xuất chia mạng DHT thành một số cụm, mỗi cụm chứa một số nút và mỗi nút chịu trách nhiệm quản lý một phần liên tục không gian khóa DHT. Khi một nút có dữ liệu cần sao lưu dữ liệu, nó mã hóa tệp dữ liệu thành n mảnh sử dụng phương pháp mã xóa và các mảnh dữ liệu này được lưu trữ tại một số nút trong cùng một cụm. Với phương pháp mã xóa, nếu thu thập được k mảnh ($k < n$) của tệp dữ liệu có thể tái tạo lại được tệp dữ liệu ban đầu, trong đó k và n là các tham số hệ thống được xác định trước. Để đảm bảo tính sẵn sàng của các tệp dữ liệu khi có các nút ra/vào mạng thì số mảnh của tệp dữ liệu phải được phải duy trì lớn hơn hoặc bằng k , trong đó k số mảnh cần thiết để tái tạo lại tệp dữ liệu.

4.3.2 Quản lý thông tin cụm

Trong giải pháp đề xuất, không gian khóa DHT được chia thành các phân vùng, mỗi phân vùng tương ứng với một cụm. Với không gian khóa d -bit, biên của cụm thứ k được xác định bởi khóa đầu tiên và khóa cuối cùng của cụm theo hướng cùng chiều kim đồng hồ. Các nút có định danh nằm giữa khóa đầu tiên và khóa cuối cùng của một cụm sẽ thuộc cùng một cụm. Nút quản lý khóa đầu tiên của một cụm được gọi là nút đầu tiên của cụm. Nút quản lý khóa cuối cùng của một cụm gọi là nút cuối cùng của cụm. Nút cuối cùng của cụm là nút đầu tiên của cụm kế tiếp.

Các nút trong cùng một cụm trao đổi thông điệp với nhau để cập nhật thông tin của cụm. Thông điệp cập nhật chứa thông tin sau:

- Phạm vi khóa của cụm
- Địa chỉ của nút đầu tiên của cụm
- Một danh sách các nút có dung lượng lưu trữ chưa sử dụng tốt nhất.
- Một danh sách các nút rời mạng trong một chu trình cập nhật.

- Hop count: Số nút mà thông điệp cập nhật đi qua bắt đầu từ nút đầu tiên.

Thông báo cập nhật thông tin cụm được gửi định kỳ. Tuy nhiên, khi số nút rời mạng do lỗi trong một khoảng thời gian cao hơn giá trị ngưỡng, thì đầu tiên nút của cụm sẽ ngay lập tức gửi thông báo cập nhật mà không cần phải chờ đến khi kết thúc chu trình cập nhật cũ.

4.3.3. Sao lưu và truy vấn dữ liệu

Tiến trình sao lưu dữ liệu của một tệp dữ liệu mới được thực hiện theo các bước dưới đây.

- Bước 1. Khi một nút có tệp dữ liệu mới để sao lưu (ví dụ: nút nguồn), đầu tiên nó tạo ra khóa của tệp tin và gửi một thông báo yêu cầu sao lưu đến nút quản lý khóa dựa trên thuật toán định tuyến DHT.

- Bước 2. Nút có quản lý khóa gửi danh sách các nút có dung lượng lưu trữ chưa dùng tốt nhất trong cụm đến nút nguồn.

- Bước 3. Sau khi nhận được danh sách các nút có dung lượng lưu trữ chưa dùng tốt nhất, nút nguồn ngẫu nhiên chọn một số nút sao lưu từ danh sách. Sau đó, nút nguồn tạo các mảnh dữ liệu ra từ tệp tin và gửi mảnh dữ liệu cùng với khóa của tệp tin tới một nút sao lưu dữ liệu để lưu trữ. Nếu nút sao lưu không có đủ khoảng trống để lưu trữ mảnh dữ liệu, nó thông báo cho nút nguồn và nút nguồn sẽ chọn một nút dự phòng khác.

- Bước 4. Khi quá trình sao lưu kết thúc, nút nguồn cung cấp cho nút quản lý khóa của tệp tin danh sách các nút trong cụm lưu trữ các mảnh dữ liệu. Nút quản lý khóa sẽ lưu trữ thông tin khóa của tệp tin và danh sách các nút lưu trữ các mảnh dữ liệu vào cơ sở dữ liệu của nó.

Khi một nút muốn truy vấn một tệp tin, nó gửi một thông báo truy vấn đến nút quản lý khóa của tệp tin. Nút quản lý khóa tìm kiếm trong cơ sở dữ liệu và gửi lại danh sách các nút lưu trữ mảnh dữ liệu của tệp tin. Nút truy vấn chọn ngẫu nhiên các nút lưu trữ mảnh dữ liệu từ danh sách nhận được và gửi thông báo truy vấn đến các nút này cho đến khi nó lấy được k mảnh dữ liệu cần thiết để tái tạo lại tệp tin.

4.3.4. Khôi phục tệp tin

Khi các nút rời mạng, các mảnh dữ liệu của các tệp tin lưu trữ tại nó có thể bị mất. Tuy nhiên, hệ thống cần phải duy trì ít nhất k mảnh dữ liệu của tệp tin trong mạng để đảm bảo tính sẵn sàng của tệp tin.

Để giải quyết vấn đề trên, định kỳ mỗi nút đếm lại số lượng các mảnh dữ liệu của mỗi tệp tin sao lưu tại nút rời mạng. Nếu tổng số mảnh dữ liệu của một tệp tin nhỏ hơn giá trị ngưỡng m thì nút quản lý thực hiện khôi phục lại các mảnh đã mất, trong đó m phải lớn hơn hoặc bằng k (là số mảnh dữ liệu cần thiết để tái tạo tệp tin).

Để khôi phục lại một mảnh dữ liệu bị mất, nút quản lý khóa của tệp tin sẽ chọn một nút sao lưu từ danh sách các nút có dung lượng lưu trữ chưa dùng tốt nhất và gửi một thông báo yêu cầu khôi phục đến nút đó. Nút sao lưu mới nhất sẽ lấy các mảnh dữ liệu từ các nút sao lưu khác, tái tạo lại tệp tin và tạo ra các mảnh dữ liệu bị mất sau đó gửi đến các nút trong cụm để lưu trữ.

4.3.5. Xây dựng cụm

Chúng tôi đề xuất hai phương pháp xây dựng cụm trong giải pháp chúng tôi đưa ra, đó là xây dựng cụm tĩnh và xây dựng cụm động. Hai phương pháp đó được mô tả cụ thể như sau:

Xây dựng cụm tĩnh

Trong phương pháp xây dựng cụm tĩnh, không gian khóa DHT được chia thành m phần tương ứng với m cụm. Với không gian khóa gồm d -bit, biên của cụm thứ k được định nghĩa bởi khóa đầu tiên và khóa cuối cùng theo hướng cùng chiều với kim đồng hồ, được xác định như sau:

$$\left[(k-1) \frac{2^d}{m}, k \frac{2^d}{m} - 1 \right] \quad k = 1, \dots, m$$

Theo cách phân chia này số cụm trong không gian khóa là cố định (m cụm), biên của một cụm cũng cố định theo số cụm. Các nút tham gia mạng sau khi có định danh sẽ thuộc một trong các cụm được phân chia.

Xây dựng cụm động

Trong phương pháp phân cụm động, một cụm tự động được tách thành hai cụm nhỏ hơn khi số lượng các nút trong một cụm lớn hơn một ngưỡng cho trước. Nếu tổng số các nút trong hai cụm liền kề nhỏ hơn một giá trị ngưỡng cho trước thì hai cụm này sẽ được nhập thành một cụm.

Không gian khóa của các cụm mới được chia theo phương pháp dưới đây. Nếu không gian khóa của cụm gốc trước khi chia là $[K_f^d, K_l^d]$, thì không gian khóa của hai cụm mới sau khi chia là một nửa của cụm gốc theo công thức (1).

$$\begin{aligned} K_f^{d_1} &= K_f^d \\ K_l^{d_1} &= K_f^{d_2} = K_f^d + \frac{1}{2}(K_l^d - K_f^d) \\ K_l^{d_2} &= K_l^d \end{aligned}$$

trong đó $[K_f^{d_1}, K_l^{d_1}]$ là không gian khóa của cụm mới thứ nhất và $[K_f^{d_2}, K_l^{d_2}]$ là không gian khóa của cụm mới thứ 2.

4.3.6. Đảm bảo tính cục bộ và cân bằng tải

Để giảm lượng thông tin dư thừa trong mạng và độ trễ truyền tin, các nút trong cùng một cụm nên ở các vị trí gần nhau về vật lý.

Một nút mới tham gia vào mạng có thể lựa chọn một vị trí từ tập các vị trí được chọn ngẫu nhiên trong không gian khóa. Hàm chi phí C được nút tham gia hệ thống sử dụng để lựa chọn vị trí tốt nhất trong số các vị trí dự tuyển.

$$C^{(i)} = (d_{self,succ}^{(i)} + d_{self,pred}^{(i)}) + \alpha C_{bestNodeList}^{(i)} \quad (2)$$

trong đó $d_{self,succ}^{(i)}$ là độ trễ truyền tin giữa nút tham gia hệ thống và nút successor của định danh thứ i ; $d_{self,pred}^{(i)}$ là độ trễ giữa nút tham gia hệ thống với nút predecessor của định danh thứ i , $C_{bestNodeList}^{(i)}$ là tổng khả năng của các nút trong danh sách các nút có khả năng cao của cụm chứa định danh thứ i và α là một tham số của hệ thống.

Giá trị nhỏ nhất trong các giá trị của hàm giá trị theo công thức (2) sẽ được chọn là vị trí của nút mới tham gia trong hệ thống.

4.4. Đánh giá giải pháp

4.4.1 Phương pháp đánh giá

Chúng tôi xây dựng chương trình mô phỏng để đánh giá thuật toán của chúng tôi và so sánh với các thuật toán sao lưu dưới đây:

- Phương pháp sao lưu theo danh sách các nút successor.
- Phương pháp RelaxDHT
- Phương pháp phân cụm cố định.
- Phương pháp phân cụm động.

Chương trình mô phỏng của chúng tôi được mở rộng từ bộ mô phỏng của Jonathan Ledlie và hoạt động theo các bước thời gian rời rạc. Mỗi bước thời gian bao gồm các hoạt động sau: nút vào và nút ra, cập nhật thông tin, sao lưu dữ liệu, duy trì tính nhất quán của tệp tin và truy vấn tệp tin.

Tại mỗi bước, thời điểm các nút ra/vào mạng được tạo dựa trên phân bố xác suất Pareto với thời gian sống trung bình của các nút trong mạng là 15 phút, 30 phút, 1 giờ, 2 giờ và 3 giờ. Mô phỏng được thực hiện trong khoảng thời gian là 3 giờ. Kết quả mô phỏng ghi lại các thống kê trong nửa cuối của thời gian mô phỏng để tránh các kết quả không ổn định. Khả năng của một nút là một số nguyên được sinh ngẫu nhiên có giá trị từ 5 đến 235 (đơn vị dữ liệu). Khả năng trung bình của một nút là 120 đơn vị dữ liệu. Các tệp tin trong hệ thống đều có cùng kích thước và được đặt là 3 đơn vị dữ liệu.

Để mô phỏng độ trễ giữa các nút, chúng tôi sử dụng bộ sinh GT-ITM Generator tạo ra một đồ thị dựa trên mô hình transit-stub gồm 18750 nút. Trong đồ thị có 50 nút transit liên kết trực tiếp với nhau. Mỗi nút transit trung bình kết nối với 15 stub domain và mỗi stub domain trung bình chứa 25 nút. Độ trễ giữa các nút transit có giá trị từ 100 mili giây đến 200 mili giây. Độ trễ giữa các nút transit với stub domain có giá trị từ 20 mili giây đến 50 mili giây. Độ trễ từ các nút đến các stub domain có giá trị từ 1 mili giây đến 10 mili giây.

4.4.2. Các kết quả mô phỏng

4.4.2.1. Đánh giá tỷ lệ truy vấn

Chúng tôi thực hiện mô phỏng để đánh giá tỷ lệ truy vấn, là tỷ số giữa số truy vấn thành công trên tổng số truy vấn. Khóa truy vấn được chọn ngẫu nhiên từ khóa của các tệp tin trong mạng. Trong các mô phỏng của 3 phương pháp tỷ lệ sao lưu được thiết lập là 2, có nghĩa là tổng kích thước của các mảnh dữ liệu sao lưu lớn gấp hai lần kích thước của tệp dữ liệu. Trong đánh giá của phương pháp chúng tôi đề xuất, số lượng các mảnh dữ liệu được tạo từ tệp tin và lưu trữ trong hệ thống (tham số n) là 6 và số lượng các mảnh dữ liệu cần thiết để tái tạo lại tệp tin (tham số k) là 3. Số mảnh dữ liệu để khởi động tiến trình khôi phục (tức là tham số m) là 4. Kích thước của một mảnh dữ liệu là 1 đơn vị dữ liệu. Trong mô phỏng của phương pháp sao lưu theo danh sách các nút successor và phương pháp relaxDHT, hai bản sao của tệp tin được lưu trữ tương ứng tại hai nút successor của nút quản lý khóa tệp tin.

Trong mô phỏng đầu tiên, chúng tôi đánh giá tỷ lệ truy vấn với sự thay đổi tải của hệ thống. Số lượng các tệp tin phân phối vào mạng được thay đổi sao cho tổng kích thước của các tệp tin sao lưu trong mạng thay đổi từ 10% đến 100% tổng khả năng lưu trữ của mạng.

Kết quả cho thấy phương pháp chúng tôi đề xuất có tỷ lệ truy vấn thành công tốt hơn các phương pháp sao lưu thông thường. Khi số lượng dữ liệu sao lưu bằng 50% khả năng lưu trữ của một nút, phương pháp phân cụm động của chúng tôi có thể đạt tỷ lệ truy vấn thành công là 98,88%, trong khi đó tỷ lệ truy vấn thành công của phương pháp relaxDHT và phương pháp sao lưu theo danh sách các nút successor chỉ đạt tỷ lệ tương ứng là 80,88% và 54,40%.

Trong thí nghiệm tiếp theo, chúng tôi đánh giá ảnh hưởng thời gian sống của một nút đến hiệu năng của các phương pháp sao lưu. Các tham số mô phỏng được sử dụng tương tự với các tham số trong mô phỏng trước, ngoại trừ thời gian sống trung bình của một nút được thay đổi từ 15 phút đến 180 phút và tổng kích thước của dữ liệu sao lưu được thiết lập bằng 50% tổng dung lượng lưu trữ của mạng. Kết quả mô phỏng thấy rằng phương pháp phân cụm động có tỷ lệ truy vấn thành công cao hơn khi các nút trong mạng ra vào thường xuyên (thời gian sống trung bình của một nút là nhỏ) nhờ cơ chế khôi phục tập tin do chúng tôi đề xuất.

Chúng tôi cũng nghiên cứu mối quan hệ giữa tỷ lệ truy vấn với số nút trong mạng. Thời gian sống trung bình của một nút được đặt là 15 phút, số cụm trong phương pháp phân cụm tĩnh được đặt là 10 cụm. Kết quả thí nghiệm cho thấy phương pháp phân cụm động có hiệu năng đạt được ổn định với tỷ lệ truy vấn đạt tương đối cao (trên 97,99%).

Để chỉ ra tính hiệu quả về cân bằng tải và duy trì tính cục bộ trong giải thuật đề xuất, chúng tôi đánh giá sự ảnh hưởng của các vị trí lựa chọn khi một nút tham gia hệ thống tới tỷ lệ các câu truy vấn. Khi số vị trí thử tăng lên, độ trễ của các nút trong cùng một cụm giảm và cân bằng tải giữa các cụm được cải thiện. Tỷ lệ truy vấn thành công được tăng từ 98,93% lên 99,9% đối với phương pháp phân cụm động và 84,8% lên 96,22% đối với phương pháp phân cụm tĩnh khi số vị trí thử tăng từ 1 đến 16.

4.4.2.2 Đánh giá chi phí duy trì tập tin

Chi phí duy trì tập tin được tính bằng tổng kích thước các mảnh dữ liệu di chuyển giữa các nút để đảm bảo tính sẵn sàng của các tập tin khi hệ thống có nút rời hệ thống. Chi phí duy trì liên quan trực tiếp đến sự ra/vào của các nút cho nên chúng tôi đánh giá chi phí duy trì với việc thay đổi thời gian sống của các nút.

Kết quả thí nghiệm cho thấy khi thời gian sống trung bình của một nút là 15 phút, phương pháp phân cụm động cần chi phí duy trì dữ liệu cao hơn phương pháp sao lưu theo danh sách successor và phương pháp relaxDHT tương ứng là 25,61% và 22,77%. Tuy nhiên, khi thời gian sống trung bình của một nút lớn hơn 90 phút, chi phí duy trì tập tin trong phương pháp của chúng tôi là thấp hơn phương pháp sao lưu theo danh sách các nút successor đồng thời tỷ lệ truy vấn thành công trong phương pháp của chúng tôi lại cao hơn phương pháp sao lưu theo danh sách các nút successor.

Chúng tôi cũng đánh giá chi phí duy trì với sự thay đổi tổng số nút ra/vào hệ thống mạng. Các kết quả mô phỏng cũng cho thấy rằng các chi phí duy trì trong các phương pháp của chúng tôi là thấp hơn phương pháp sao lưu theo danh sách các nút successor và phương pháp relaxDHT khi số nút trong mạng tăng lên.

4.4.2.3 Ảnh hưởng của các tham số sao lưu tập tin

Chúng tôi đánh giá hiệu quả của các tham số sao lưu tập tin, bao gồm: số mảnh dữ liệu được tạo ra từ tập tin (số n), số mảnh dữ liệu cần thiết để tái tạo lại tập tin (số k) và số mảnh dữ liệu để bắt đầu quá trình duy trì tập tin (số m) với các bộ tham số sau ($n = 6, k = 3, m = 3$), ($n = 6, k = 3, m = 4$), ($n = 6, k = 3, m = 5$) và ($n = 9, k = 3, m = 6$) với k được giữ cố định là 3. Tổng số nút ra/vào hệ thống là 8000 nút.

Trong mô phỏng đầu tiên, chúng tôi đánh giá tác động của các tham số sao lưu tệp tin đến tỷ lệ truy vấn thành công và chi phí duy trì tệp tin với thời gian sống trung bình các nút thay đổi. Tổng số tệp tin sao lưu được thiết lập bằng 20% khả năng chứa của các nút trong mạng. Kết quả mô phỏng cho thấy chi phí duy trì tệp tin của hệ thống với $(n = 6, k = 3, m = 5)$ cao hơn 84.69% so với $(n = 6, k = 3, m = 4)$ khi thời gian sống trung bình của các nút là 15 phút.

Bộ tham số $(n = 9, k = 3, m = 6)$ cho tỷ lệ truy vấn truy vấn cao nhất. Tuy nhiên, dung lượng lưu trữ cao gấp 1,5 lần so với hệ thống với $n = 6$. Bằng cách thay đổi các tham số này cho một số tệp tin có độ ưu tiên cao, chúng tôi có thể đảm bảo tính sẵn sàng của các tệp tin trong mạng P2P.

4.5. Kết luận

Trong chương này, chúng tôi đề xuất một sơ đồ sao lưu dữ liệu dựa trên phân cụm cho các hệ thống sao lưu tệp tin dựa trên DHT và sử dụng phương pháp xoá mã trong việc đảm bảo tính sẵn sàng của tệp tin. Các kết quả mô phỏng chứng minh tính hiệu quả trong giải pháp chúng tôi đề xuất về phương diện tỷ lệ thành công của các câu truy vấn và chi phí duy trì tính sẵn sàng của tệp tin so với các giải pháp thông thường.

Trong nghiên cứu tiếp theo, chúng tôi sẽ giải quyết vấn đề dư thừa truy vấn khi sao lưu các mảnh dữ liệu trong các cụm và xoá các tệp tin không cần thiết khi một nút bị quá tải.

Chương 5. KẾT LUẬN

Chương này, chúng tôi tổng kết lại các kết quả đạt được của luận án, các đóng góp chính của luận án, đánh giá các hạn chế và đưa ra các hướng nghiên cứu tiếp theo.

5.1. Các kết quả đã đạt được

Luận án đã trình bày các kiến thức nền tảng về mạng ngang hàng nói chung như khái niệm, đặc điểm, phân loại, các lĩnh vực ứng dụng giúp người đọc khái quát lại được các vấn đề có liên quan đến nội dung nghiên cứu trong luận án. Luận án cũng trình bày chi tiết về mạng ngang hàng có cấu trúc, bảng băm phân tán và giao thức mạng ngang hàng có cấu trúc Chord. Đây là kiến trúc và giao thức được sử dụng trong các nghiên cứu của luận án.

Đề xuất thuật toán cải tiến của thuật toán cân bằng tải theo ngưỡng trong mạng ngang hàng có cấu trúc để nâng cao tỷ lệ thành công của câu truy vấn. Kết quả đánh giá cho thấy thuật toán đề xuất hoạt động tốt hơn, tỷ lệ các câu truy vấn thành công đạt được cao hơn thuật toán cân bằng tải theo ngưỡng khoảng từ 6% đến 15% tùy vào từng thí nghiệm và điều kiện đánh giá.

Đề xuất phương pháp điều khiển tắc nghẽn trong mạng ngang hàng có cấu trúc bằng cách thay thế bảng định tuyến của một nút trong quá trình truy vấn tệp tin gặp nút bị tắc nghẽn, các nút thay thế trên đường đi được chọn là các nút tiếp theo sau nút tắc nghẽn và còn có khả năng xử lý tốt các câu truy vấn. Kết quả thí nghiệm cho thấy giải pháp đề xuất tốt hơn giao thức Chord khoảng từ 15% đến 40% tùy vào từng điều kiện đánh giá.

Đề xuất phương pháp sao lưu dữ liệu trong mạng ngang hàng có cấu trúc để đảm bảo tính sẵn sàng của dữ liệu qua đó nâng cao chất lượng dịch vụ trong mạng ngang hàng có cấu trúc. Trong đề xuất này chúng tôi đưa ra phương pháp phân cụm tĩnh và phân cụm động cho việc sao lưu dữ liệu. Các nút trong cụm sẽ lưu trữ các mảnh của tệp tin cần sao lưu. Giải pháp này vừa đảm bảo được tính sẵn sàng cho dữ liệu, đảm bảo được cân bằng tải cho các nút trong mạng đồng thời hạn chế tối đa việc sử dụng băng thông mạng trong việc sao lưu dữ liệu. Kết quả cho thấy giải pháp của chúng tôi đưa ra tốt hơn các giải pháp thông thường, tỷ lệ truy vấn và tìm kiếm thành công tệp tin tăng hơn so với giải pháp thông thường từ 20% đến 50% tùy vào từng thí nghiệm và tham số đánh giá.

5.2. Những hạn chế và hướng nghiên cứu tiếp theo

Trên cơ sở các vấn đề chưa được giải quyết và các hướng nghiên cứu mở rộng với bài toán đảm bảo chất lượng dịch vụ trong mạng ngang hàng có cấu trúc, chúng tôi sẽ tiếp tục tìm hiểu sâu hơn theo các hướng sau:

- Giải pháp cân bằng tải đối với các truy vấn trong khoảng và nâng cao hiệu quả hoạt động của thư mục;
- Kết hợp giải pháp đề xuất với một phương pháp điều khiển tắc nghẽn dựa trên điều khiển lưu lượng nhằm giải quyết triệt để vấn đề tắc nghẽn khi mạng chịu tải cao, tránh bị sụp đổ do tắc nghẽn mà vẫn đảm bảo được khả năng phục vụ của toàn mạng.
- Giải quyết vấn đề dư thừa truy vấn khi sao lưu các mảnh dữ liệu trong các cụm;

DANH MỤC CÔNG TRÌNH KHOA HỌC CỦA TÁC GIẢ LIÊN QUAN ĐẾN LUẬN ÁN

1. Hoai Son Nguyen, **Dinh Nghia Nguyen**, Shinji SUGAWARA Senior Member, "A Dynamic-Clustering Backup Scheme for High-Availability Distributed File Sharing Systems". *IEICE Transactions on Communications* Vol.E102-B, No.3, pp.-, Mar. 2019. (SCI, impact factor: 1.09).

2. **Nguyen Dinh Nghia**, Nguyen Hoai Son, "Congestion control algorithm for message routing in Structured Peer-to-peer Networks". *Journal of Computer Science and Cybernetics*. Vol 34, No 2, 2018, pp.145-159.

3. **Dinh Nghia Nguyen**, Xuan Hoang Tran, Hoai Son Nguyen, "A Cluster-based File Replication Scheme for DHT-based File Backup System". *Proceedings of The 2016 international conference on advanced technologies for communications*. pp 204-214.

4. **Nguyễn Đình Nghĩa**, Nguyễn Hoài Sơn, "Tối ưu hóa sao lưu dữ liệu trong mạng ngang hàng có cấu trúc". *Kỷ yếu Hội nghị khoa học công nghệ quốc gia lần thứ VI (Fair 2013)*, Nxb Khoa học tự nhiên và Công nghệ, Hà Nội 2013. pp563-571.

5. **Nguyễn Đình Nghĩa**, Đỗ Cao Minh, Nguyễn Hoài Sơn, "Nâng cao hiệu quả của thuật toán cân bằng tải theo ngưỡng trong mạng ngang hàng có cấu trúc". *Kỷ yếu hội thảo quốc gia về CNTT năm 2009*, Nxb Khoa học và kỹ thuật. pp 206-280.

Danh mục này bao gồm 05 công trình.