

MỘT SỐ THUẬT TOÁN ĐẢM BẢO NHẤT QUÁN DỮ LIỆU TRONG CÁC HỆ PHÂN TÁN

SOME ALGORITHMS ENSURES DATA CONSISTENCY IN DISTRIBUTED SYSTEMS

Lê Văn Sơn

Trường Đại học Sư phạm, Đại học Đà Nẵng
Email: levansupham2004@yahoo.com

Nguyễn Hồng Minh

Trường Đại học An ninh Nhân dân
Email: hongminhnguyen1982@gmail.com

TÓM TẮT

Nhân bản dữ liệu là giải pháp tối ưu, quan trọng được sử dụng trong các ứng dụng phân tán nhằm đáp ứng tính sẵn sàng cao của dữ liệu, giảm thiểu yêu cầu sử dụng băng thông của đường truyền mạng, tăng khả năng chịu lỗi, khả năng mở rộng... Tuy nhiên, kỹ thuật nhân bản cũng đặt ra nhiều thách thức, trong đó, đảm bảo tính nhất quán dữ liệu là khó khăn chủ yếu trong xây dựng phần mềm ứng dụng. Ở nội dung bài báo, chúng tôi nghiên cứu, đánh giá nhiều thuật toán đã được xây dựng, ứng dụng dựa trên một số tiêu chí quan trọng của hệ phân tán như: số lượng nhân bản, số lượng message lưu thông trên hệ thống và thời gian đáp ứng. Hơn nữa, chúng tôi còn đề xuất thuật toán nhằm giải quyết vấn đề đặt ra. Thuật toán được xây dựng dựa trên mô hình nhất quán tuần tự và có những ưu điểm sau: triển khai thực hiện thuận tiện, giảm yêu cầu sử dụng băng thông và thời gian đáp ứng nhanh.

Từ khóa: nhân bản; thuật toán; nhất quán dữ liệu; bản sao; phân tán; mô hình nhất quán dữ liệu.

ABSTRACT

Data replication is an optimal and important solution, which is used in distributed applications to meet the need of high availability of data, minimizing bandwidth consumption, increasing fault-tolerance, scalability and security, etc. However, replication technique also poses many complicated challenges, in which, ensuring consistency of data is the major difficulty in developing application software. In this article, many algorithms, which have been developed and applied, are researched and measured basing on some important criteria of distributed systems including the number of replicas, the number of circulating messages in the system and the response time. Moreover, we also recommend an algorithm to solve this problem. The algorithm is based on the sequential consistency model and has the advantages of easy implementation, low bandwidth consumption and fast response time.

Key words: replication; algorithm; data consistency; replicated; distributed; consistency data model

1. Đặt vấn đề

Hiện nay, nhiều ứng dụng phân tán, làm việc cộng tác đang được phát triển mạnh vì nhiều ưu điểm: chi phí, hiệu năng, khả năng mở rộng, độ tin cậy và tính phân tán cố hữu của ứng dụng [1]. Tuy nhiên giải quyết bài toán về tính sẵn sàng của dữ liệu trong các ứng dụng này là đòi hỏi khó.

Nhân bản dữ liệu là giải pháp hiệu quả đối với yêu cầu trên [2,3]. Bởi vì nó giải quyết được những vấn đề quan trọng như:

* Tăng độ tin cậy của hệ thống

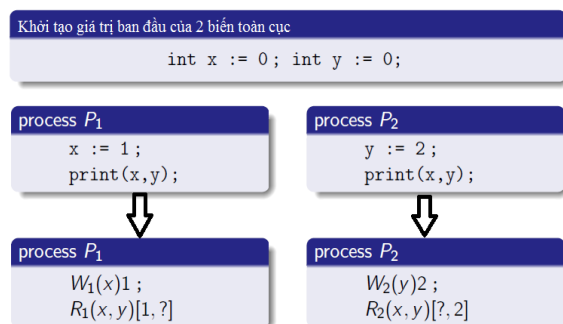
* Việc duy trì nhiều bản sao của đối tượng dữ liệu sẽ giúp hệ thống hoặc người dùng khôi phục lại dữ liệu bằng cách chuyển đến làm việc trên các bản sao

* Tăng hiệu năng và tính sẵn sàng của hệ thống

Nhân bản dữ liệu là giải pháp tăng tính sẵn sàng của dữ liệu, giảm thiểu yêu cầu sử dụng băng thông, tăng khả năng chịu lỗi, khả năng mở rộng [2,3]... từ đó làm tăng hiệu năng của hệ thống phân tán.

Tuy nhiên việc sử dụng các bản sao cũng đặt ra nhiều vấn đề phức tạp đối với hệ thống như: An ninh an toàn, lưu trữ các bản sao, đảm bảo tính nhất quán dữ liệu [3]. Do đó, chúng tôi tập trung nghiên cứu thuật toán nhằm đảm bảo tính nhất quán dữ liệu cung cấp cho người dùng. Cụ thể là thuật toán quản lý các thao tác đọc, ghi trên các bản sao [10].

Ví dụ có bài toán đơn giản sau:



Kết quả trả về là một trong 3 khả năng sau:

$$\{x = 0 \wedge y = 0\} \langle P_1 \parallel P_2 \rangle \{R_1(x,y)[1, 0] \wedge R_2(x,y)[1, 2]\}$$

$$\{x = 0 \wedge y = 0\} \langle P_1 \parallel P_2 \rangle \{R_1(x,y)[1, 2] \wedge R_2(x,y)[0, 2]\}$$

$$\{x = 0 \wedge y = 0\} \langle P_1 \parallel P_2 \rangle \{R_1(x,y)[1, 2] \wedge R_2(x,y)[1, 2]\}$$

Như vậy, chúng ta không có được kết quả chính xác do không biết giá trị đọc được của biến có phải là giá trị thay đổi lần cuối cùng của biến đó theo thời gian thực hay không, vì trong hệ thống phân tán không có thời gian chung cho mọi thao tác tham gia trên hệ thống, độ trễ của việc giao tiếp bằng Message giữa các thực thể để đưa ra quyết định... Từ đó đặt ra yêu cầu xây dựng thứ tự thực hiện các thao tác trên cùng một đối tượng theo thời gian logic. Điều này đòi hỏi chi phí rất lớn và tăng độ trễ trên hệ thống.

Nhiều mô hình nhất quán dữ liệu đã được đề xuất như: Mô hình nhất quán chặt, mô hình nhất quán tuyến tính, mô hình nhất quán tuần tự, mô hình nhất quán FIFO... và các thuật toán đảm bảo tính nhất quán dữ liệu đều dựa trên một trong những mô hình đó để thực hiện.

Trong nội dung của bài báo, tác giả thực hiện nghiên cứu so sánh một số thuật toán đảm bảo sự nhất quán dữ liệu trong môi trường làm việc cộng tác, phân tán. Hơn nữa, tác giả cũng đề xuất thuật toán để giải quyết yêu cầu trên.

2. Các thuật toán đảm bảo nhất quán dữ liệu trong hệ phân tán

2.1. Thuật toán Biểu quyết tĩnh (Static Voting) [4,5]

Thuật toán Voting thuộc về lớp giải thuật pessimistic [9] vì sự nhất quán dữ liệu của các bản sao vẫn được đảm bảo trong trường hợp mạng giao tiếp bị phân chia thành các vùng khác nhau do lỗi về đường truyền hay của mỗi máy

trạm. Khi hệ thống bị phân chia thành các phân vùng, mỗi phân vùng có thể có một bản sao khác nhau của cùng một đối tượng. Để giải quyết vấn đề này, giải thuật Voting thực hiện như sau:

Mỗi bản sao của tập tin được đánh số gọi là số phiên bản (version number) của tập tin. Số phiên bản ban đầu là không và sẽ tăng lần lượt lên một đơn vị sau mỗi lần bản sao được cập nhật. Như vậy, bản sao nào có số phiên bản lớn nhất chính là bản sao hiện thời (bản được cập nhật sau cùng) của phân vùng đó. Nếu hệ thống muốn thực hiện cập nhật mới đối với tập tin, hệ thống sẽ thực hiện gửi tập tin hiện thời để thực hiện cập nhật cho những phân vùng còn thiếu, sau đó mới thực hiện nhân bản theo nguyên tắc: Thực hiện cập nhật tập tin trên một phân vùng nếu và chỉ nếu phân vùng đó chứa nhiều hơn một nửa số Node mà tập tin được nhân bản. Ví dụ, tập tin được nhân bản cho 1000 Node, như vậy phân vùng phải chứa tối thiểu 500 Node đó. Nếu không có phân vùng nào thỏa mãn điều kiện thì sẽ không thực hiện cập nhật đối với tập tin.

Chính vì vậy, thuật toán Voting có ưu điểm khi sử dụng để đảm bảo nhất quán dữ liệu cho các hệ thống có tần suất phân vùng cao. Giải thuật không tối ưu về số lượng message truyền trên mạng để thực hiện yêu cầu cập nhật.

2.2. Thuật toán vòng được đề xuất bởi Ellis (OEA)[6]

Thuật toán vòng được đề xuất để giải quyết bài toán nhân bản hoàn toàn theo mô hình tuần tự trong cơ sở dữ liệu phân tán. Cấu hình của hệ thống bao gồm các Node được tổ chức thành mạch vòng. Các Node hoạt động độc lập, có bộ nhớ cục bộ và xử lý các yêu cầu của người dùng cục bộ mà nó phục vụ.

Các Node được đánh số thứ tự 1, 2...n và giao tiếp với nhau bằng message theo mạch vòng. Có nghĩa là, Node_i gửi message đến Node_{i+1} và nhận message từ Node_{i-1}. Hơn nữa, mỗi Node cũng được gán độ ưu tiên tương ứng $\Pi(i)$ nhận giá trị từ 1 đến n. Độ ưu tiên cao nhất là 1 và thấp nhất là n.

Mỗi Node được cấu hình gồm: hàng đợi ngoài (E), hàng đợi trong (I) hoạt động theo

nguyên tắc FIFO và có một trong 3 trạng thái *actif*, *passif*, *oisif*. Khi yêu cầu cập nhật tập tin từ người dùng cục bộ được gửi tới Node_i, nó sẽ được đặt vào E_i để chờ xử lý. Khi trạng thái của Node_i là *actif*, nó được quyền gửi yêu cầu cập nhật tập tin cho các Node khác trên mạng. Khi message yêu cầu trở lại vị trí Node_i thì thực hiện cập nhật tập tin và Node_i sẽ gửi thông báo đến tất cả các Node khác.

Thuật toán thiết lập độ ưu tiên của mỗi Node để giải quyết trường hợp khi có nhiều Node cũng đang gửi yêu cầu cập nhật. Khi đó yêu cầu cập nhật của Node có độ ưu tiên thấp hơn sẽ được đặt vào trong hàng đợi I của Node có độ ưu tiên cao hơn để chờ đến lượt được xử lý.

Giải thuật OEA có ưu điểm là mỗi Node chỉ liên lạc với hai Node khác liền kề với nó và sự cập nhật sẽ được thực hiện khi Node dành được khóa. Hơn nữa, giải thuật sử dụng rất ít trạng thái, không sử dụng nhãn thời gian (Timestamps) của các sự kiện và khóa cho các thành phần vì vậy tiết kiệm thời gian đọc. Tuy nhiên, giải thuật trên cũng tồn tại hai bất lợi chủ yếu. Một là, khi cập nhật dữ liệu cần khóa toàn bộ dữ liệu, điều này sẽ không cho phép cập nhật đồng thời trong trường hợp không xảy ra xung đột. Hai là, để thực hiện cập nhật, message phải được truyền hai vòng, dẫn đến độ trễ lớn hơn.

2.3. Thuật toán Refreshment[7]

Thuật toán làm việc trên môi trường mạng cung cấp cơ chế multicast tin cậy FIFO [8]. Các Node được cấu hình để hoàn toàn tự chủ và có sự phân biệt chức năng của Node: Node Master (M) lưu trữ các bản sao chính, Node Slave (S) lưu trữ các bản sao thứ cấp và Node Master/Slave (MS) lưu trữ cả hai. Khi nút S lưu trữ bản sao thứ cấp của bản sao chính của nút M, thì ta gọi nút S là Slave của nút M và sẽ có cung nối trực tiếp từ M→S. Như vậy, các Node Master chịu trách nhiệm nhận bản dữ liệu đến các Slave của nó (nơi chứa những bản sao thứ cấp). Hệ thống sử dụng các giao dịch update (T) để cập nhật các bản sao chính, các giao dịch refresh (RT) để cập nhật các bản sao.

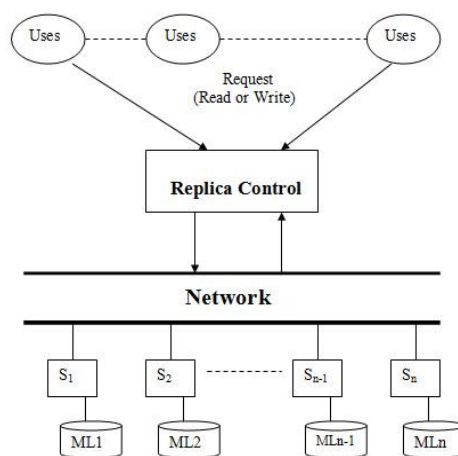
Mỗi Node Slave hoặc Node Master/Slave khi nhận được message yêu cầu cập nhật sẽ đưa

vào hàng đợi thực hiện theo cơ chế FIFO. Như vậy, thành phần của hàng đợi là dãy các thao tác ghi tương ứng với các giao dịch RT và nhãn thời gian (Timestamp) C của giao dịch T (giao dịch cập nhật bản sao chính) tương ứng. Do đó, khi hệ thống xử lý giao dịch RT để thực hiện cập nhật các bản sao thứ cấp phải căn cứ vào nhãn thời gian C, nghĩa là đảm bảo nhất quán dữ liệu dữ liệu giữa bản sao chính và bản sao thứ cấp.

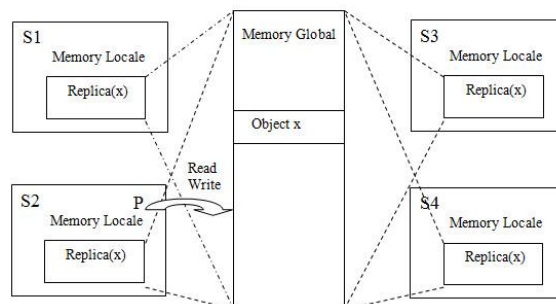
Thuật toán có một số ưu điểm như: triển khai ứng dụng dễ dàng; thực hiện được trên các hệ cơ sở dữ liệu; không phụ thuộc vào thời gian lan truyền cập nhật giữa các Node (nhiều thuật toán yêu cầu thời gian này phải rất nhỏ cỡ vài giây) mà chỉ cần đảm bảo cơ chế multicast FIFO, vì vậy hệ thống không bị lỗi do đường truyền; giảm số lượng message trao đổi trên mạng.

3. Nghiên cứu giải pháp nhất quán dữ liệu

Thuật toán đảm bảo nhất quán dữ liệu mà chúng tôi đề xuất dựa trên mô hình nhất quán tuần tự [11].



Hình 1. Cấu trúc liên kết các Node



Hình 2. Mô hình chia sẻ bộ nhớ

3.1. Mô tả cấu trúc hệ thống

Cấu trúc mạng kết nối các Node trong hệ thống (Hình 1) gồm có:

- Tập hợp các Node $S_1, S_2, \dots, S_n$. Mỗi Node được cấu hình để làm việc độc lập và có bộ nhớ cục bộ để lưu trữ các bản sao của bộ nhớ toàn cục nhằm đáp ứng các yêu cầu (yêu cầu đọc ghi dữ liệu) của người dùng mà nó phục vụ.
- Hệ quản lý, điều khiển nhân bản dữ liệu
- Tập người dùng cục bộ được quản lý bởi các Node

3.2. Thuật toán

Mỗi Node quản lý bản sao của bộ nhớ toàn cục (Hình 2). Mọi thao tác đọc của người dùng luôn luôn thực hiện trên bộ nhớ cục bộ của Node quản lý người dùng đó. Tuy nhiên, mọi thao tác ghi (cập nhật) cần sắp xếp theo trật tự toàn cục để đảm bảo tính nhất quán dữ liệu nhân bản.

Các thao tác được xử lý bởi các Node như sau:

Thao tác đọc bản sao dữ liệu trên Node_i

```
Value Read(ref x){
    Return Replicai(x)}
```

Thao tác đọc được thực hiện đơn giản trên Node cục bộ những vẫn đảm bảo giá trị trả về là bản sao đối tượng hiện thời trên bộ nhớ toàn cục.

Thao tác ghi (cập nhật dữ liệu) trên Node_i

```
Write(ref x, value v){
    abroadcast(x,v,i); /* gửi yêu cầu
    ghi dữ liệu lên hệ thống bằng giao thức
    Atomic broadcast*/
    writei(x).wait();}
```

Khi thực hiện yêu cầu ghi dữ liệu trên đối tượng x , Node cục bộ sẽ gửi yêu cầu cập nhật dữ liệu lên hệ thống bằng giao thức Abcast, để đảm bảo trật tự toàn cục, sau đó chờ hệ thống xử lý để thực hiện cập nhật. Có nghĩa là, khi có nhiều tiến trình đang thực hiện trên các Node khác cũng yêu cầu cập nhật bản sao, thì tất cả các tiến trình đều nhận biết được sự tồn tại của nhau và thực hiện cập nhật dữ liệu trên một trật tự toàn cục theo mô hình nhất quán tuần tự.

Xử lý yêu cầu ghi dữ liệu của Node_i tại Node_j

```
Receive(ref x, value v, site i){
    Replicaj[x]=v;
    If (j == i) writei[x].notify()}
```

Khi có yêu cầu cập nhật giá trị của đối tượng x tại Node_j từ Node_i, Node_j sẽ thực hiện cập nhật lại giá trị của bản sao x trong bộ nhớ cục bộ của mình.

3.3. Điều kiện cần để thuật toán đảm bảo nhất quán dữ liệu

Giả sử có r thao tác đọc bản sao x , trong đó chỉ có w thao tác ghi để thay đổi giá trị của bản sao, như vậy điều kiện cần đối với thuật toán là:

- $r + w > n$
- $w > n/2$
- Ngoài ra cần quản lý trật tự toàn cục đối với tất cả các thao tác ghi

3.4. Đánh giá

Người dùng thực hiện đọc bộ nhớ cục bộ để lấy dữ liệu khi có yêu cầu, do đó thời gian đáp ứng nhanh. Để đảm bảo nhất quán dữ liệu khi thực hiện yêu cầu ghi của một Node, thuật toán sử dụng giao thức Abcast để gửi thông báo tới tất cả Node khác trên hệ thống. Đây là giao thức không sử dụng nhiều message vì vậy giảm lưu lượng lưu thông trên mạng, từ đó giảm yêu cầu sử dụng băng thông đường truyền. Tuy nhiên, thuật toán cũng có những hạn chế nhất định như: mạng phải an toàn để thực hiện gửi và nhận message trong giao thức Abcast và mỗi node thì cần có một định danh duy nhất.

4. Kết luận và hướng nghiên cứu

Nhân bản dữ liệu trong các hệ phân tán là giải pháp chủ yếu, then chốt được sử dụng để đáp ứng yêu cầu sẵn sàng dữ liệu của hệ thống, từ đó tăng hiệu năng, hiệu quả của ứng dụng. Tuy nhiên, việc đảm bảo nhất quán dữ liệu là thách thức lớn trong kỹ thuật nhân bản nhằm đảm bảo tính sẵn sàng cao, giảm băng thông sử dụng, thời gian đáp ứng nhanh trong các ứng dụng làm việc cộng tác, phân tán.

Chúng tôi đã thực hiện nghiên cứu, phân tích, đánh giá một cách đầy đủ những thuật toán tiêu biểu đối với lớp bài toán trên. Hơn nữa, chúng tôi cũng đã thực hiện nghiên cứu, đề xuất

thuật toán dựa trên mô hình nhất quán dữ liệu tuần tự. Đây là mô hình nhất quán yếu, có những ưu việt như: giảm message lưu thông trên mạng từ đó giảm sử dụng băng thông, thời gian đáp ứng nhanh. Ngoài ra, cấu hình của mỗi Node cũng tương đối đơn giản, do đó thuật toán cũng đơn giản trong triển khai thực hiện. Chúng tôi

cũng đã chỉ ra những mặt hạn chế của thuật toán như: yêu cầu sự ổn định của mạng cao, mỗi Node tham gia hệ thống cần có định danh riêng.

Trong thời gian tới, chúng tôi sẽ thực hiện mô phỏng thuật toán trong các cấu trúc mạng khác nhau để thực hiện so sánh, đánh giá toàn diện hơn đối với thuật toán.

TÀI LIỆU THAM KHẢO

- [1] Ranganathan, Kavitha, and Ian Foster. "Identifying dynamic replication strategies for a high-performance data grid." *Grid Computing—GRID 2001*. Springer Berlin Heidelberg, 2001. 75-86.
- [2] Djebbara, Mohamed Redha, and Hafida Belbachir. "Gestion de Répliques dans les Grilles de Données." *Ecole Supérieur d'Informatique—ESI (ex INI) Oued Semar—Alger, Lab LSSD Université des Sciences et de Technologies Oran—USTO*.
- [3] Goel, Sushant, and Rajkumar Buyya. "Data replication strategies in wide area distributed systems." *Enterprise Service Computing: From Concept to Deployment 17* (2006).
- [4] GIFFORD, D. K. Weighted voting for replicated data. In *Proceedings of the 7th Symposium on Operating System Principles* (1979), pp. 150-162.
- [5] SEGUIN, J., SERGEANT, G., AND WILMS, P. A majority consensus algorithm for the consistency of duplicated and distributed information. In *Proceedings of the IEEE International Conference on Distributed Computing Systems* (1979). IEEE, New York, 1979, pp. 617-624.
- [6] C. A. Ellis, Consistency and Correctness of Duplicate Database Systems, *Gth Symposium on Operating System Principles* (1977) 67-84.
- [7] Pacitti, Esther, Pascale Minet, and Eric Simon. "Replica consistency in lazy master replicated databases. Proceeding VLDB '99 Proceedings of the 25th International Conference on Very Large Data Bases, Pages 126-137.
- [8] V. Hadzilacos and S. Toueg, A Modular Approach to fault-Tolerant Broadcasts and Related Problems, Technical Report TR 94-1425, Dept, of Computer Science, Cornell University, Ithaca, NY 14853, 1994.
- [9] Jajodia, Sushil and Mutchler, David "A Pessimistic Consistency Control Algorithm for Replicated Files which Achieves High Availability", *IEEE Transactions on Software Engineering*, Vol. 15, No 1, January 1989, pp 39-46.
- [10] Kotsakis, E. G., and B. H. Pardoe. "Dynamic Quorum Adjustment: A Consistency Scheme for Replicated Objects." *Proceedings of the Third Communication Networks Symposium*. 1996.
- [11] Raynal, Michel, and André Schiper. "From causal consistency to sequential consistency in shared memory systems." *Foundations of Software Technology and Theoretical Computer Science*. Springer Berlin Heidelberg, 1995.

(BBT nhận bài: 09/08/2013, phản biện xong: 19/08/2013)