

## **Phương pháp cân bằng tải có trọng số trên các điểm truy cập dữ liệu**

### **Lĩnh vực kỹ thuật của sáng chế**

Sáng chế đề cập đến phương pháp cân bằng tải có trọng số trên các điểm truy cập dữ liệu, đảm bảo khả mở rộng tuyến tính dung lượng của hệ thống truy cập dữ liệu và cân bằng tải khi thêm, bớt hoặc thay đổi phần cứng của các điểm truy cập dữ liệu.

### **Tình trạng kỹ thuật của sáng chế**

Trong các hệ thống truy cập dữ liệu lớn, người phát triển phần mềm sẽ gặp phải bài toán phân mảnh dữ liệu và cân bằng tải trên nhiều điểm truy cập dữ liệu khác nhau, đảm bảo khả năng truy cập và chịu tải lớn của hệ thống. Ngoài ra, khi thêm, bớt hoặc thay đổi phần cứng của các điểm truy cập dữ liệu, cần cân bằng lại dữ liệu trên các điểm truy cập dữ liệu, đồng thời đảm bảo hệ thống không bị gián đoạn trong quá trình hoạt động.

Hiện nay, có nhiều hệ thống cơ sở dữ liệu đã xây dựng các phương pháp phân mảnh dữ liệu khác nhau, ví dụ như: MySQL, MongoDB, Aerospike ... Tuy nhiên các hệ thống trên gặp phải hạn chế là không phân biệt trọng số của các điểm truy cập dữ liệu. Các điểm truy cập dữ liệu cần phải giống hệt nhau về phần cứng để đảm bảo khả năng cân bằng tải. Ngoài ra trong quá trình hoạt động, nếu thay đổi phần cứng của một điểm truy cập dữ liệu sẽ làm hệ thống bị mất cân bằng tải.

Phương pháp cân bằng tải dưới đây đảm bảo được khả năng mở rộng tuyến tính của hệ thống, khi thêm các điểm truy cập dữ liệu mới khả năng chịu tải của hệ thống cũng được tăng theo tỷ lệ tuyến tính. Đồng thời phương pháp này có phân biệt trọng số của các điểm truy cập dữ liệu, cho phép các điểm truy cập dữ liệu có cấu hình phần cứng tùy biến khác nhau, trong quá trình hoạt động có thể tùy chỉnh trọng số của các điểm truy cập dữ liệu một cách phù hợp.

### **Bản chất kỹ thuật của sáng chế**

Phương pháp cân bằng tải có trọng số trên các điểm truy cập dữ liệu nhằm mục đích giải quyết các vấn đề cốt lõi trong quá trình vận hành hệ thống truy cập dữ liệu có dung lượng lớn là cân bằng tải trong quá trình hoạt động và phân bố lại dữ liệu khi thêm, bớt hoặc thay đổi phần cứng của các điểm truy cập dữ liệu, hạn chế tối đa việc dịch chuyển dữ liệu.

Để đạt được mục đích nêu trên, phương pháp cân bằng tải có trọng số trên các điểm truy cập dữ liệu bao gồm các bước:

Bước 1: thực hiện cập nhật lại bảng định tuyến, dịch chuyển các nút ảo của các nút mạng với nhau, khi thêm, bớt hoặc thay đổi trọng số của nút mạng, thực hiện chuyển nút ảo của các nút có số nút ảo giảm cho các nút có số nút ảo tăng, cụ thể: với hệ thống bao gồm  $N$  nút ảo và  $M$  nút mạng có định danh và trọng số tương ứng lần lượt là  $ID_1, ID_2, ID_3, \dots, ID_m$  và  $W_1, W_2, W_3, \dots, W_m$ .

Khi thêm một nút mạng có định danh  $ID_{m+1}$ , cần đảm bảo tỷ lệ dữ liệu bị dịch chuyển là:  $W_{m+1} \div \sum_{k=1}^{m+1}(W_k)$ .

Khi bớt một nút mạng có định danh  $ID_i$  với  $i \in [1, M]$ , cần đảm bảo tỷ lệ dữ liệu bị dịch chuyển là:  $W_i \div \sum_{k=1}^m(W_k)$ .

Khi thay đổi trọng số của các điểm truy cập dữ liệu với danh sách trọng số mới lần lượt là  $Q_1, Q_2, Q_3, \dots, Q_m$ , cần đảm bảo tỷ lệ dữ liệu bị dịch chuyển là:  $\sum_{k=1}^m |Q_k \div Q - W_k \div W| \div 2$ .

Trong đó  $M$  là tổng số nút trước khi thêm/bớt;  $W_{m+1}$  là trọng số của nút được thêm;  $W_i$  là trọng số của nút bị bớt;  $Q$  là tổng trọng số mới;  $W$  là tổng trọng số cũ;  $Q_k$  là trọng số mới của nút  $ID_k$ ;  $W_k$  là trọng số của nút  $ID_k$ .

Bước 2: lưu bảng định tuyến cũ (trước khi cập nhật) trên mảng  $A_1$  và bảng định tuyến mới (sau khi cập nhật lại ở bước 1) trên mảng  $A_2$ .

Bước 3: thực hiện khóa truy cập tới vùng dữ liệu bị dịch chuyển, cụ thể:

Với các yêu cầu đọc dữ liệu, ứng với khóa  $K$ , thực hiện hàm băm  $F(x)$  tính giá trị  $I = F(K)$ ; thực hiện đọc dữ liệu ứng với khóa  $K$  trên nút có định danh  $A_1[I]$  và trả về kết quả.

Với các yêu cầu ghi dữ liệu, ứng với khóa  $K$ , thực hiện hàm băm  $F(x)$  tính giá trị  $I = F(K)$ ; nếu  $A_1[I] = A_2[I]$  dữ liệu nằm ngoài vùng dịch chuyển, thực hiện ghi dữ liệu ứng với khóa  $K$  vào nút có định danh  $A_1[I]$  và trả về mã thành công; nếu  $A_1[I] \neq A_2[I]$  dữ liệu nằm trong vùng dịch chuyển, từ chối thực hiện yêu cầu ghi dữ liệu và trả về mã lỗi tương ứng.

Bước 4: thực hiện sao chép dữ liệu từ các điểm truy cập cũ sang điểm truy cập mới, cụ thể: duyệt toàn bộ cơ sở dữ liệu, với mỗi khóa  $K$  của từng bản ghi, thực hiện

hàm băm  $F(x)$  tính giá trị  $I = F(K)$ ; nếu  $A_1[I] \neq A_2[I]$  thực hiện sao chép bản ghi ứng với khóa  $K$  từ nút có định danh  $A_1[I]$  sang nút có định danh  $A_2[I]$ ; nếu  $A_1[I] = A_2[I]$  bản ghi ứng với khóa  $K$  không bị dịch chuyển, thực hiện duyệt bản ghi tiếp theo.

Bước 5: sau khi sao chép toàn bộ dữ liệu cần dịch chuyển, các yêu cầu truy cập dữ liệu được xử lý bình thường với việc sử dụng phương pháp phân mảnh dữ liệu và bảng định tuyến mới  $A_2$ , ứng với khóa  $K$ , thực hiện hàm băm  $F(x)$  tính giá trị  $I = F(K)$ ; bản ghi ứng với khóa  $K$  được truy cập trên nút có định danh  $A_2[I]$ .

Bước 6: thực hiện dọn dẹp các bản ghi bị trùng sau khi sao chép dữ liệu, cụ thể: duyệt toàn bộ cơ sở dữ liệu, với mỗi khóa  $K$  của từng bản ghi, thực hiện hàm băm  $F(x)$  tính giá trị  $I = F(K)$ ; nếu  $A_1[I] \neq A_2[I]$  thực hiện xóa bản ghi ứng với khóa  $K$  trên nút có định danh  $A_1[I]$ ; nếu  $A_1[I] = A_2[I]$  bản ghi ứng với khóa  $K$  không bị trùng, thực hiện duyệt bản ghi tiếp theo.

### **Mô tả vắn tắt các hình vẽ**

Hình 1: mô tả phương pháp phân mảnh dữ liệu.

Hình 2: mô tả quá trình cập nhật lại bảng định tuyến khi thêm một nút.

Hình 3: mô tả quá trình cập nhật lại bảng định tuyến khi bớt một nút.

Hình 4: mô tả quá trình cập nhật lại bảng định tuyến khi thay đổi trọng số của các nút.

Hình 5: mô tả quá trình ghi dữ liệu khi đang sao chép dữ liệu sau khi cập nhật lại bảng định tuyến.

Hình 6: mô tả quá trình sao chép dữ liệu sau khi cập nhật lại bảng định tuyến.

Hình 7: mô tả quá trình dọn dẹp dữ liệu sau khi sao chép dữ liệu thành công.

Hình 8: mô tả tổng quan các bước thực hiện phương pháp đề xuất.

### **Mô tả chi tiết sáng chế**

Phương pháp cân bằng tải có trọng số trên các điểm truy cập dữ liệu được cấu thành bởi hai phương pháp chính: phương pháp phân mảnh dữ liệu và phương pháp cân bằng dữ liệu khi thêm, bớt hoặc thay đổi trọng số của các điểm truy cập dữ liệu.

Một số thuật ngữ sử dụng trong phần mô tả chi tiết sau đây được định nghĩa:

Từ khóa	Định nghĩa
Nút mạng (Node):	Một điểm truy cập dữ liệu trong hệ thống truy cập dữ liệu.
Nút ảo (Virtual node):	Một nút mạng tương ứng với nhiều nút ảo. Dữ liệu truy cập đến nút mạng thông qua các nút ảo tương ứng.
Trọng số:	Một số thực không âm ứng với mỗi nút mạng. Nút mạng có trọng số càng lớn thì có càng nhiều nút ảo.
Bảng định tuyến:	Một mảng có kích thước cố định chứa danh sách các nút ảo.
Bản ghi (Record):	Một cấu trúc dữ liệu được lưu trữ trên các nút mạng.
Khóa (Key):	Mỗi bản ghi có một khóa tương ứng. Bản ghi được truy cập thông qua khóa tương ứng.
Hàm băm (Hash function):	Một hàm có đầu vào là giá trị của khóa và đầu ra là một số tự nhiên bất kỳ thuộc đoạn $[0, N]$ với $N$ là một số nguyên dương xác định.

Phương pháp phân mảnh dữ liệu được sử dụng để xác định điểm truy cập dữ liệu của bản ghi ứng với một khóa  $K$  xác định. Trong trường hợp dữ liệu đang được cân bằng lại sau khi thêm, bớt hoặc thay đổi trọng số của các nút, dữ liệu được truy cập theo phương pháp cân bằng dữ liệu được nêu ở sau.

Phương pháp phân mảnh dữ liệu sử dụng bảng định tuyến là một mảng  $A$  gồm  $N$  phần tử chứa định danh (ID) của các nút mạng. Phần tử có giá trị  $X$  được gọi là nút ảo của nút có định danh  $X$ . Số nút ảo của một nút mạng được xác định:

$$C = N \times W_1 \div W_2$$

Trong đó:  $C$  là số nút ảo của nút mạng;  $N$  là tổng số nút ảo;  $W_1$  là trọng số của nút mạng;  $W_2$  là tổng trọng số của các nút mạng.

Khi cần truy cập bản ghi ứng với một khóa  $K$  xác định, áp dụng hàm băm  $F(x)$  tính giá trị  $I = F(K) \in [0, N - 1]$ .

Xác định giá trị của phần tử  $A[I]$  ứng với định danh của nút mạng chứa bản ghi có khóa  $K$ .

Tham chiếu hình 8, phương pháp cân bằng tải có trọng số trên các điểm truy cập dữ liệu theo sáng chế đề xuất được thực hiện tuần tự như sau:

Bước 1: thực hiện cập nhật lại bảng định tuyến (dịch chuyển các nút ảo của các nút mạng với nhau) khi thêm, bớt hoặc thay đổi trọng số của nút mạng.

Với hệ thống bao gồm N nút ảo và M nút mạng có định danh và trọng số tương ứng lần lượt là  $ID_1, ID_2, ID_3, \dots, ID_m$  và  $W_1, W_2, W_3, \dots, W_m$ .

- Tham chiếu hình 2, khi thêm một nút mạng có định danh  $ID_{m+1}$ , cần đảm bảo tỷ lệ dữ liệu bị dịch chuyển là:

$$W_{m+1} \div \sum_{k=1}^{m+1} (W_k)$$

Trong đó: M là tổng số nút trước khi thêm;  $W_{m+1}$  là trọng số của nút được thêm;  $W_k$  là trọng số của nút  $ID_k$ .

Với từng nút mạng có định danh  $ID_i$  với  $i \in [1, M]$ , thực hiện chuyển các nút ảo của nút  $ID_i$  cho nút  $ID_{m+1}$ . Số nút ảo cần chuyển là:

$$N \times \left( W_i \div \sum_{k=1}^m (W_k) - W_i \div \sum_{k=1}^{m+1} (W_k) \right)$$

Trong đó: N là tổng số nút ảo; M là tổng số nút trước khi thêm;  $W_i$  là trọng số của nút được thêm;  $W_k$  là trọng số của nút  $ID_k$ .

- Tham chiếu hình 3, khi bớt một nút mạng có định danh  $ID_i$  với  $i \in [1, M]$ , cần đảm bảo tỷ lệ dữ liệu bị dịch chuyển là:

$$W_i \div \sum_{k=1}^m (W_k)$$

Trong đó:  $W_i$  là trọng số của nút bị bớt;  $W_k$  là trọng số của nút  $ID_k$ .

Với từng nút mạng có định danh  $ID_j$  với  $j \in [1, M]$  và  $j \neq i$ , thực hiện chuyển các nút ảo của nút  $ID_i$  cho nút  $ID_j$ . Số nút ảo cần chuyển là:

$$N \times \left( W_j \div \left( \sum_{k=1}^m (W_k) - W_i \right) - W_j \div \sum_{k=1}^m (W_k) \right)$$

Trong đó: N là tổng số nút ảo; M là tổng số nút trước khi bớt;  $W_i$  là trọng số của nút bị bớt;  $W_j$  là trọng số của nút  $ID_j$ ;  $\bar{W}_k$  là trọng số của nút  $ID_k$ .

- Tham chiếu hình 4, khi thay đổi trọng số của các điểm truy cập dữ liệu với danh sách trọng số mới lần lượt là  $Q_1, Q_2, Q_3, \dots, Q_m$ , cần đảm bảo tỷ lệ dữ liệu bị dịch chuyển là:

$$\sum_{k=1}^m |Q_k \div Q - W_k \div W| \div 2$$

Trong đó:  $Q$  là tổng trọng số mới;  $W$  là tổng trọng số cũ;  $Q_k$  là trọng số mới của nút  $ID_k$ ;  $W_k$  là trọng số cũ của nút  $ID_k$ .

Với từng nút mạng có định danh  $ID_i$  với  $i \in [1, M]$ , số lượng nút ảo bị thay đổi của nút này là:

$$C = N \times (Q_k \div Q - W_k \div W)$$

Trong đó:  $Q$  là tổng trọng số mới;  $W$  là tổng trọng số cũ;  $Q_k$  là trọng số mới của nút  $ID_k$ ;  $W_k$  là trọng số cũ của nút  $ID_k$ ;  $C$  là số nút ảo bị thay đổi.

- $C = 0$ : Số nút ảo giữ nguyên.
- $C > 0$ : Số nút ảo tăng.
- $C < 0$ : Số nút ảo giảm.

Thực hiện chuyển nút ảo của các nút có số nút ảo giảm cho các nút có số nút ảo tăng.

Bước 2: lưu bảng định tuyến cũ (trước khi cập nhật) trên mảng  $A_1$  và bảng định tuyến mới (sau khi cập nhật lại ở bước 1) trên mảng  $A_2$ .

Bước 3: thực hiện khóa truy cập tới vùng dữ liệu bị dịch chuyển:

Tham chiếu hình 5 mô tả quá trình ghi dữ liệu khi đang sao chép dữ liệu sau khi cập nhật lại bảng định tuyến:

Với các yêu cầu đọc dữ liệu:

- Ứng với khóa  $K$ , thực hiện hàm băm  $F(x)$  tính giá trị  $I = F(K)$ .
- Thực hiện đọc dữ liệu ứng với khóa  $K$  trên nút có định danh  $A_1[I]$  và trả về kết quả.

Với các yêu cầu ghi dữ liệu:

- Ứng với khóa  $K$ , thực hiện hàm băm  $F(x)$  tính giá trị  $I = F(K)$ .
- Nếu  $A_1[I] = A_2[I]$  dữ liệu nằm ngoài vùng dịch chuyển, thực hiện ghi dữ liệu ứng với khóa  $K$  vào nút có định danh  $A_1[I]$  và trả về mã thành công.
- Nếu  $A_1[I] \neq A_2[I]$  dữ liệu nằm trong vùng dịch chuyển, từ chối thực hiện yêu cầu ghi dữ liệu và trả về mã lỗi tương ứng.

Bước 4: thực hiện sao chép dữ liệu từ các điểm truy cập cũ sang điểm truy cập mới:

Tham chiếu hình 6 mô tả quá trình sao chép dữ liệu sau khi cập nhật lại bảng định tuyến. Duyệt toàn bộ cơ sở dữ liệu, với mỗi khóa  $K$  của từng bản ghi, thực hiện hàm băm  $F(x)$  tính giá trị  $I = F(K)$ .

Nếu  $A_1[I] \neq A_2[I]$  thực hiện sao chép bản ghi ứng với khóa  $K$  từ nút có định danh  $A_1[I]$  sang nút có định danh  $A_2[I]$ .

Nếu  $A_1[I] = A_2[I]$  bản ghi ứng với khóa  $K$  không bị dịch chuyển, thực hiện duyệt bản ghi tiếp theo.

Bước 5: sau khi sao chép toàn bộ dữ liệu cần dịch chuyển, các yêu cầu truy cập dữ liệu được xử lý bình thường với việc sử dụng phương pháp phân mảnh dữ liệu và bảng định tuyến mới  $A_2$ .

Tham chiếu hình 1, mô tả quá trình truy cập dữ liệu sử dụng bảng định tuyến mới. Cụ thể được thực hiện như sau:

- Ứng với khóa  $K$ , thực hiện hàm băm  $F(x)$  tính giá trị  $I = F(K)$ .
- Bản ghi ứng với khóa  $K$  được truy cập trên nút có định danh  $A_2[I]$ .

Bước 6: thực hiện dọn dẹp các bản ghi bị trùng sau khi sao chép dữ liệu:

Tham chiếu hình 7, quá trình dọn dẹp các bản ghi bị trùng được thực hiện như sau:

Với mỗi khóa  $K$  của từng bản ghi, thực hiện hàm băm  $F(x)$  tính giá trị  $I = F(K)$ .

Nếu  $A_1[I] \neq A_2[I]$  thực hiện xóa bản ghi ứng với khóa  $K$  trên nút có định danh  $A_1[I]$ .

Nếu  $A_1[I] = A_2[I]$  bản ghi ứng với khóa K không bị trùng, thực hiện duyệt bản ghi tiếp theo.

### **Các lợi ích mà sáng chế đạt được.**

Các lợi ích mà sáng chế đạt được bao gồm:

Phương pháp phân mảnh dữ liệu: đảm bảo khả năng mở rộng tuyến tính của hệ thống truy cập dữ liệu, khả năng chịu tải của hệ thống tăng theo tỷ lệ tuyến tính với số lượng điểm truy cập dữ liệu.

Phương pháp cân bằng tải khi thêm, bớt hoặc thay đổi trọng số của các điểm truy cập dữ liệu: Hạn chế tối đa lượng dữ liệu cần dịch chuyển, giảm thiểu sự gián đoạn dịch vụ và đảm bảo tính toàn vẹn của dữ liệu trong quá dịch chuyển.

Phân biệt trọng số của các điểm truy cập dữ liệu, cho phép sử dụng các điểm truy cập dữ liệu có cấu hình phần cứng khác nhau.



## **Yêu cầu bảo hộ**

1. Phương pháp cân bằng tải có trọng số trên các điểm truy cập dữ liệu bao gồm các bước:

bước 1: thực hiện cập nhật lại bảng định tuyến, dịch chuyển các nút ảo của các nút mạng với nhau, khi thêm, bớt hoặc thay đổi trọng số của nút mạng, thực hiện chuyển nút ảo của các nút có số nút ảo giảm cho các nút có số nút ảo tăng, cụ thể: với hệ thống bao gồm  $N$  nút ảo và  $M$  nút mạng có định danh và trọng số tương ứng lần lượt là  $ID_1, ID_2, ID_3, \dots, ID_m$  và  $W_1, W_2, W_3, \dots, W_m$ :

khi thêm một nút mạng có định danh  $ID_{m+1}$ , cần đảm bảo tỷ lệ dữ liệu bị dịch chuyển là:  $W_{m+1} \div \sum_{k=1}^{m+1}(W_k)$ ;

khi bớt một nút mạng có định danh  $ID_i$  với  $i \in [1, M]$ , cần đảm bảo tỷ lệ dữ liệu bị dịch chuyển là:  $W_i \div \sum_{k=1}^m(W_k)$ ;

khi thay đổi trọng số của các điểm truy cập dữ liệu với danh sách trọng số mới lần lượt là  $Q_1, Q_2, Q_3, \dots, Q_m$ , cần đảm bảo tỷ lệ dữ liệu bị dịch chuyển là:  $\sum_{k=1}^m |Q_k \div Q - W_k \div W| \div 2$ ;

trong đó  $M$  là tổng số nút trước khi thêm;  $W_{m+1}$  là trọng số của nút được thêm;  $W_i$  là trọng số của nút bị bớt;  $Q$  là tổng trọng số mới;  $W$  là tổng trọng số cũ;  $Q_k$  là trọng số mới của nút  $ID_k$ ;  $W_k$  là trọng số của nút  $ID_k$ ;

bước 2: lưu bảng định tuyến cũ trên mảng  $A_1$  và bảng định tuyến mới trên mảng  $A_2$ ;

bước 3: thực hiện khóa truy cập tới vùng dữ liệu bị dịch chuyển, cụ thể:

với các yêu cầu đọc dữ liệu, ứng với khóa  $K$ , thực hiện hàm băm  $F(x)$  tính giá trị  $I = F(K)$ ; thực hiện đọc dữ liệu ứng với khóa  $K$  trên nút có định danh  $A_1[I]$  và trả về kết quả;

với các yêu cầu ghi dữ liệu, ứng với khóa K, thực hiện hàm băm  $F(x)$  tính giá trị  $I = F(K)$ ; nếu  $A_1[I] = A_2[I]$  dữ liệu nằm ngoài vùng dịch chuyển, thực hiện ghi dữ liệu ứng với khóa K vào nút có định danh  $A_1[I]$  và trả về mã thành công; nếu  $A_1[I] \neq A_2[I]$  dữ liệu nằm trong vùng dịch chuyển, từ chối thực hiện yêu cầu ghi dữ liệu và trả về mã lỗi tương ứng;

bước 4: thực hiện sao chép dữ liệu từ các điểm truy cập cũ sang điểm truy cập mới, cụ thể: duyệt toàn bộ cơ sở dữ liệu, với mỗi khóa K của từng bản ghi, thực hiện hàm băm  $F(x)$  tính giá trị  $I = F(K)$ ; nếu  $A_1[I] \neq A_2[I]$  thực hiện sao chép bản ghi ứng với khóa K từ nút có định danh  $A_1[I]$  sang nút có định danh  $A_2[I]$ ; nếu  $A_1[I] = A_2[I]$  bản ghi ứng với khóa K không bị dịch chuyển, thực hiện duyệt bản ghi tiếp theo;

bước 5: sau khi sao chép toàn bộ dữ liệu cần dịch chuyển, các yêu cầu truy cập dữ liệu được xử lý bình thường với việc sử dụng phương pháp phân mảnh dữ liệu và bảng định tuyến mới  $A_2$ , ứng với khóa K, thực hiện hàm băm  $F(x)$  tính giá trị  $I = F(K)$ ; bản ghi ứng với khóa K được truy cập trên nốt có định danh  $A_2[I]$ ;

bước 6: thực hiện dọn dẹp các bản ghi bị trùng sau khi sao chép dữ liệu, cụ thể: duyệt toàn bộ cơ sở dữ liệu, với mỗi khóa K của từng bản ghi, thực hiện hàm băm  $F(x)$  tính giá trị  $I = F(K)$ ; nếu  $A_1[I] \neq A_2[I]$  thực hiện xóa bản ghi ứng với khóa K trên nút có định danh  $A_1[I]$ ; nếu  $A_1[I] = A_2[I]$  bản ghi ứng với khóa K không bị trùng, thực hiện duyệt bản ghi tiếp theo.

2. Phương pháp cân bằng tải có trọng số trên các điểm truy cập dữ liệu theo điểm 1 trong đó, số nút ảo cần chuyển trong trường hợp thêm một nút mạng có định danh  $ID_{m+1}$ :  $N \times (W_i \div \sum_{k=1}^m (W_k) - W_i \div \sum_{k=1}^{m+1} (W_k))$  trong đó: N là tổng số nút ảo; M là tổng số nút trước khi thêm;  $W_i$  là trọng số của nút được thêm;  $W_k$  là trọng số của nút  $ID_k$ .

3. Phương pháp cân bằng tải có trọng số trên các điểm truy cập dữ liệu theo điểm 1 trong đó, số nút ảo cần chuyển trong trường hợp bớt  $ID_i$  với  $I \in [1, M]$ , là:  $N \times (W_j \div (\sum_{k=1}^m (W_k) - W_i) - W_j \div \sum_{k=1}^m (W_k))$  trong đó: N là tổng số nút ảo; M là tổng số nút trước khi bớt;  $W_i$  là trọng số của nút bị bớt;  $W_j$  là trọng số của nút  $ID_j$ ;  $W_k$  là trọng số của nút  $ID_k$ .

4. Phương pháp cân bằng tải có trọng số trên các điểm truy cập dữ liệu theo điểm 1 trong đó khi thay đổi trọng số của các điểm truy cập dữ liệu với danh sách trọng số

mới lần lượt là  $Q_1, Q_2, Q_3, \dots, Q_m$ , từng nút mạng có định danh  $ID_i$  với  $i \in [1, M]$ , số lượng nút ảo bị thay đổi của nút này là:

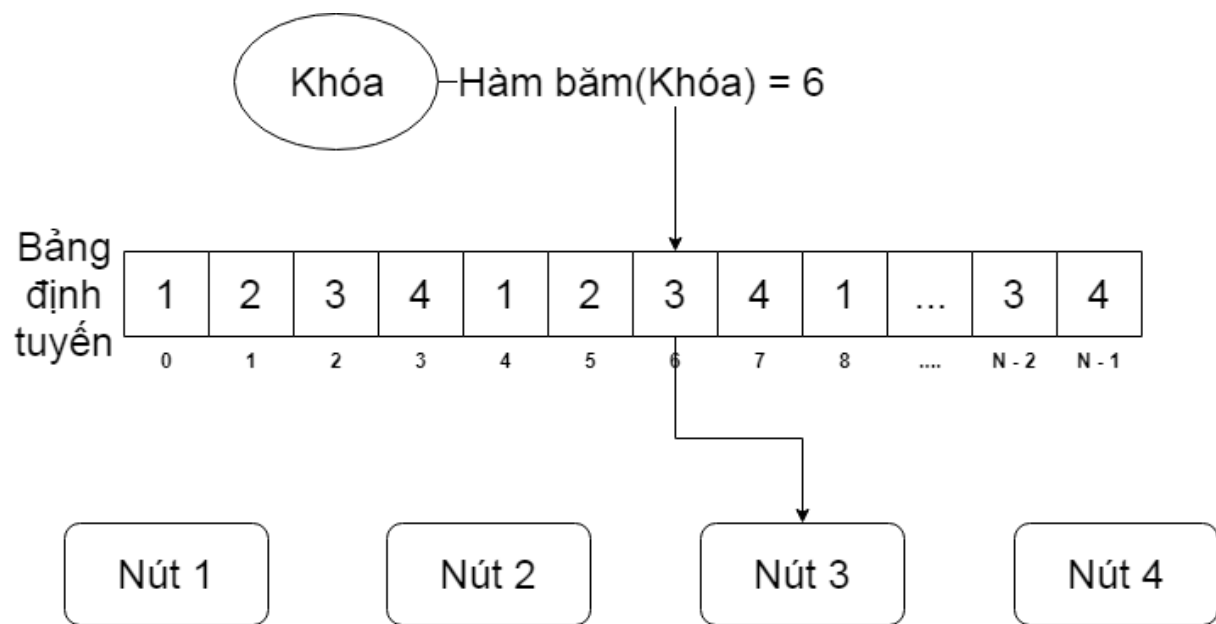
$$C = N \times (Q_k \div Q - W_k \div W)$$

trong đó  $Q$  là tổng trọng số mới;  $W$  là tổng trọng số cũ;  $Q_k$  là trọng số mới của nút  $ID_k$ ;  $W_k$  là trọng số cũ của nút  $ID_k$ ;  $C$  là số nút ảo bị thay đổi.

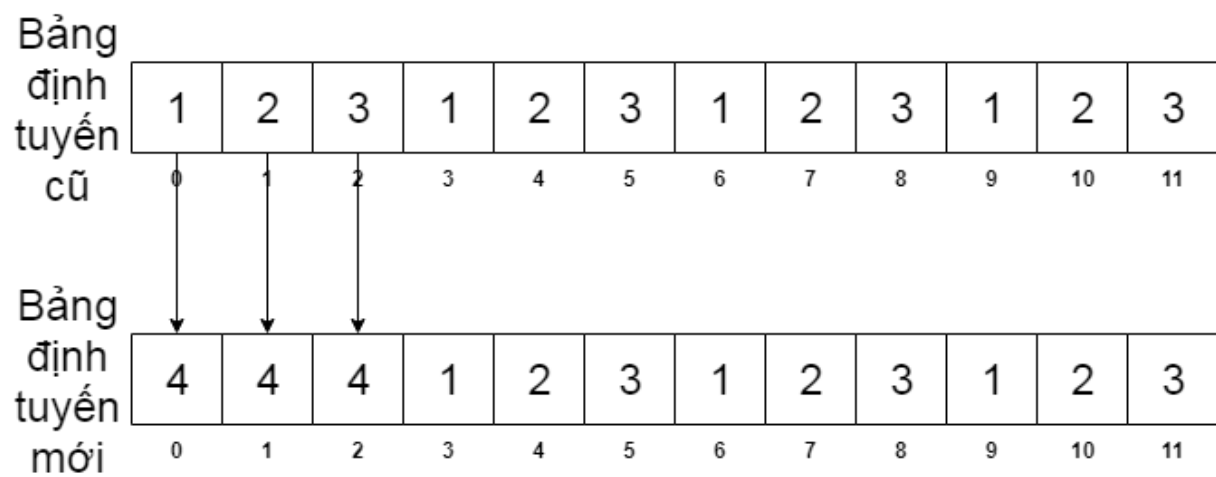
### **Tóm tắt**

Phương pháp cân bằng tải có trọng số trên các điểm truy cập dữ liệu đảm bảo khả năng mở rộng tuyến tính của hệ thống truy cập dữ liệu, khả năng chịu tải của hệ thống tỷ lệ tuyến tính với số lượng điểm truy cập dữ liệu. Phương pháp đề xuất bao gồm các bước: bước 1: thực hiện cập nhật lại bảng định tuyến, dịch chuyển các nút ảo của các nút mạng với nhau, khi thêm, bớt hoặc thay đổi trọng số của nút mạng, thực hiện chuyển nút ảo của các nút có số nút ảo giảm cho các nút có số nút ảo tăng; bước 2: lưu bảng định tuyến cũ trên mảng  $A_1$  và bảng định tuyến mới trên mảng  $A_2$ ; bước 3: thực hiện khóa truy cập tới vùng dữ liệu bị dịch chuyển; bước 4: thực hiện sao chép dữ liệu từ các điểm truy cập cũ sang điểm truy cập mới; bước 5: thực hiện các yêu cầu truy cập dữ liệu được xử lý bình thường với việc sử dụng phương pháp phân mảnh dữ liệu và bảng định tuyến mới  $A_2$ ; bước 6: dọn dẹp các bản ghi bị trùng sau khi sao chép dữ liệu.

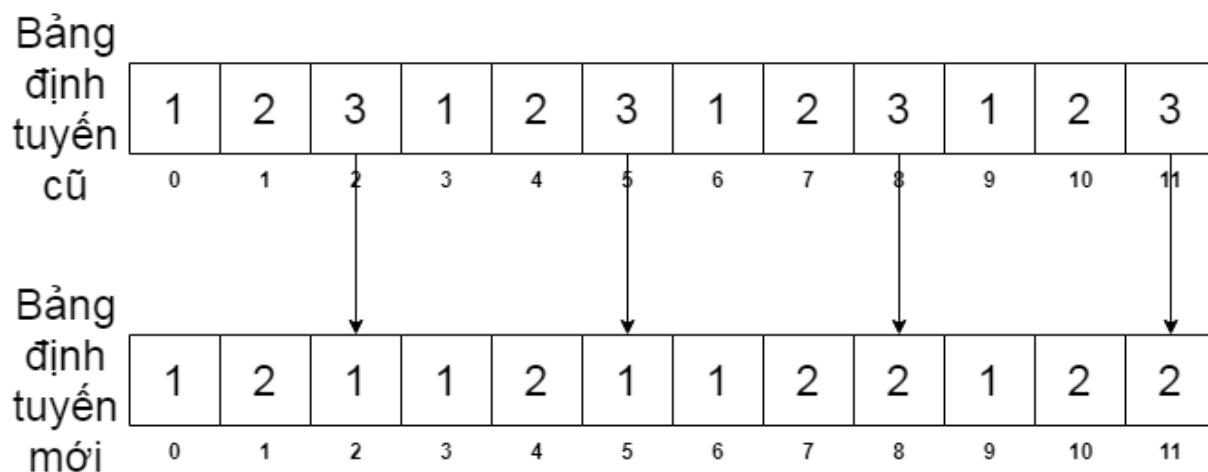
[Hình 8]



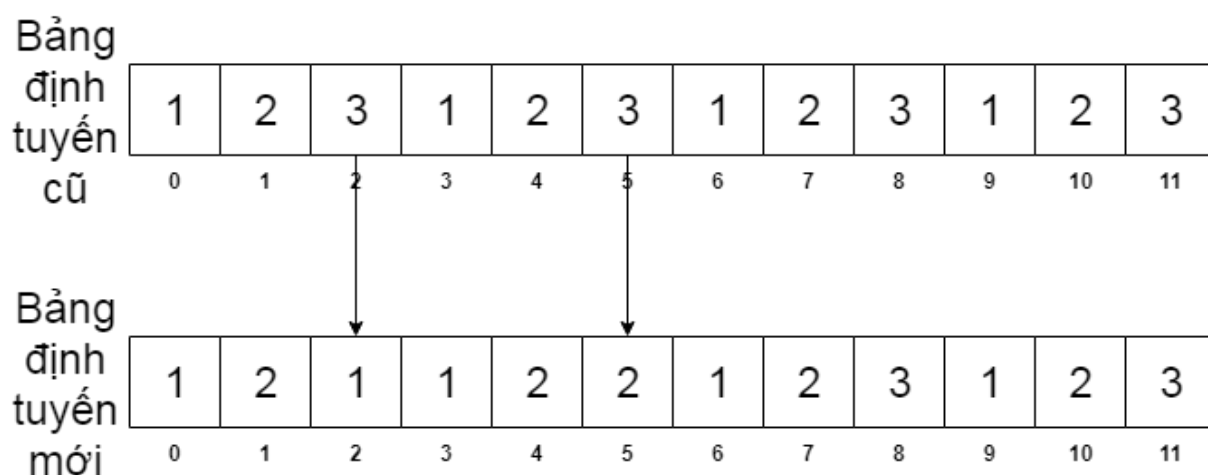
Hình 1



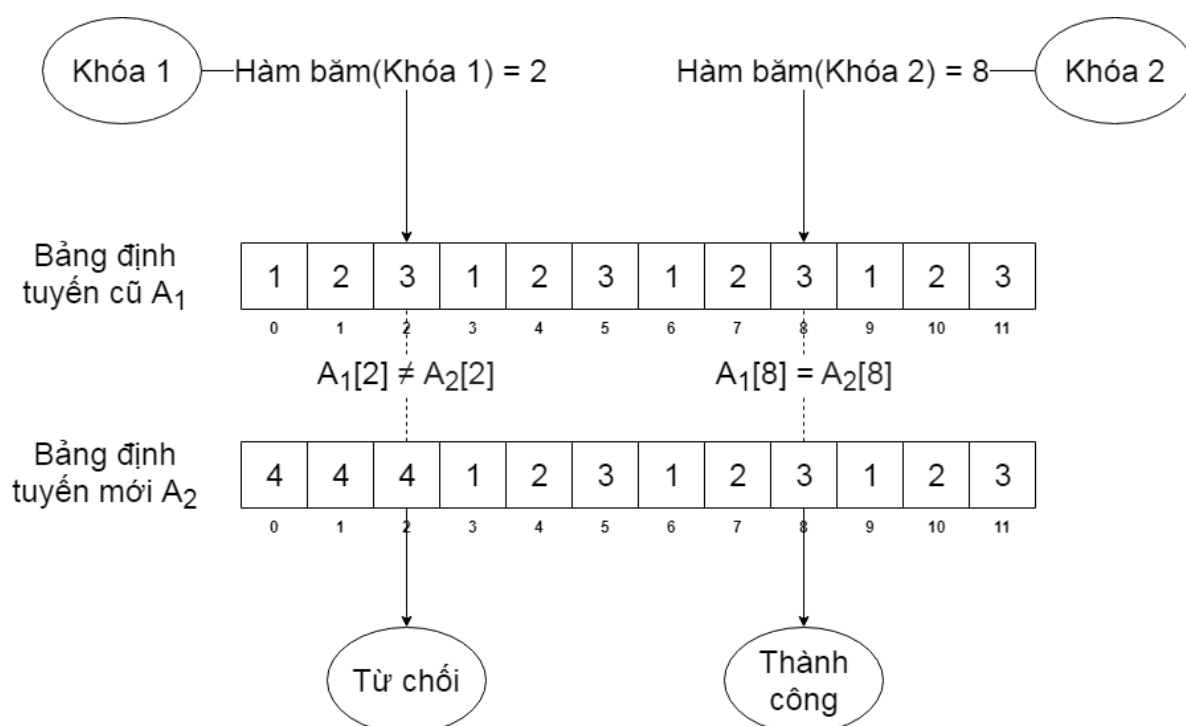
Hình 2



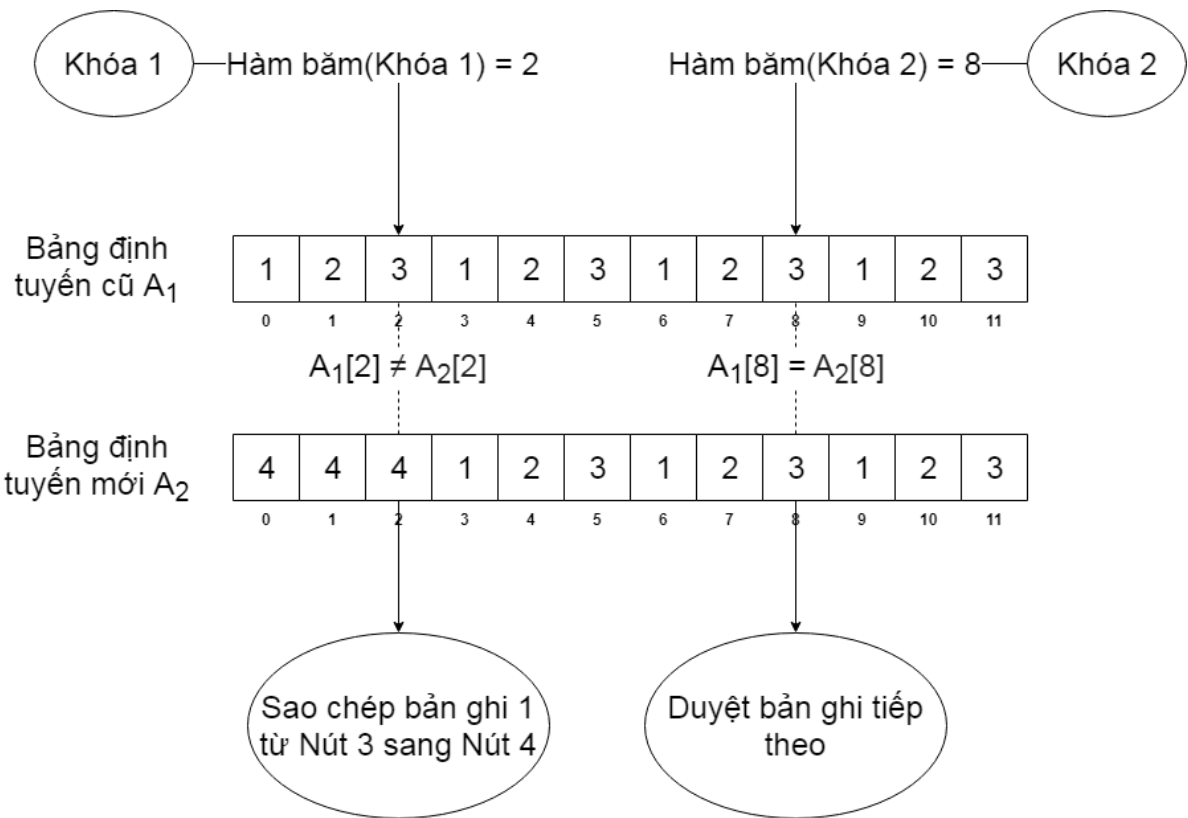
Hình 3



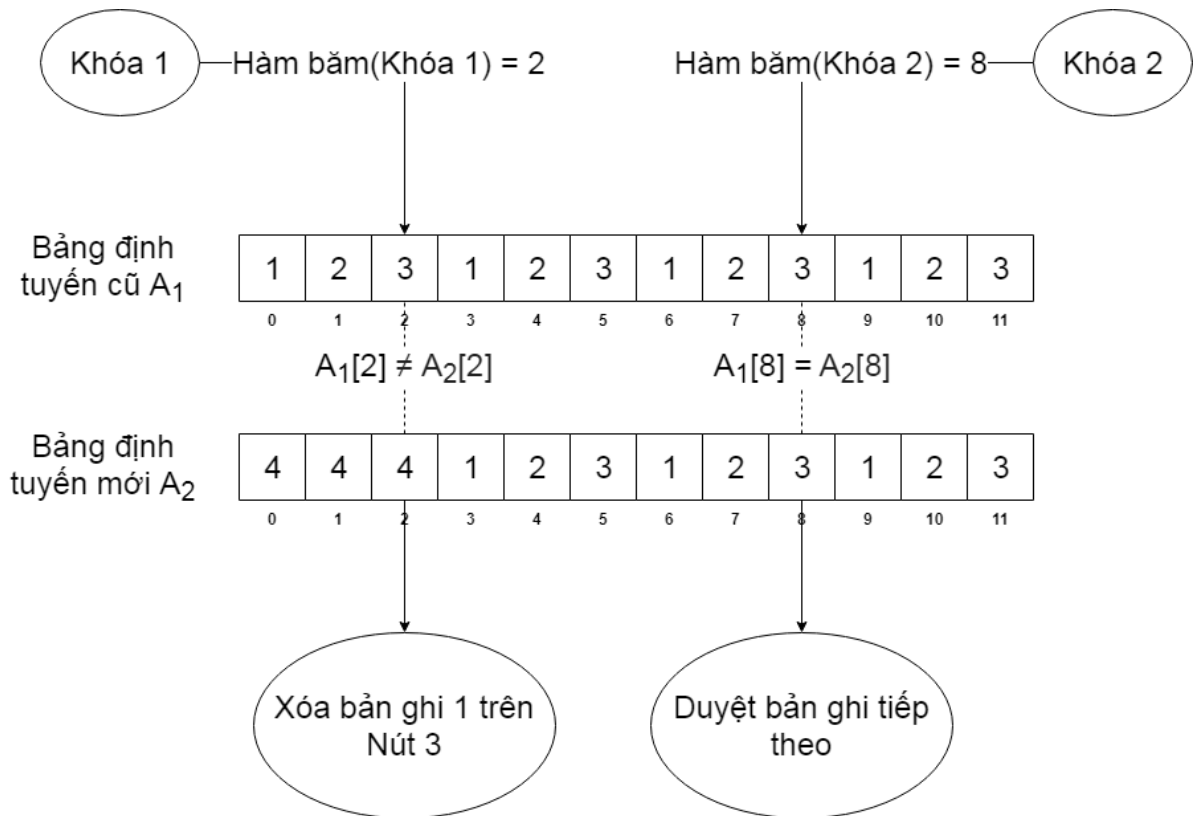
Hình 4



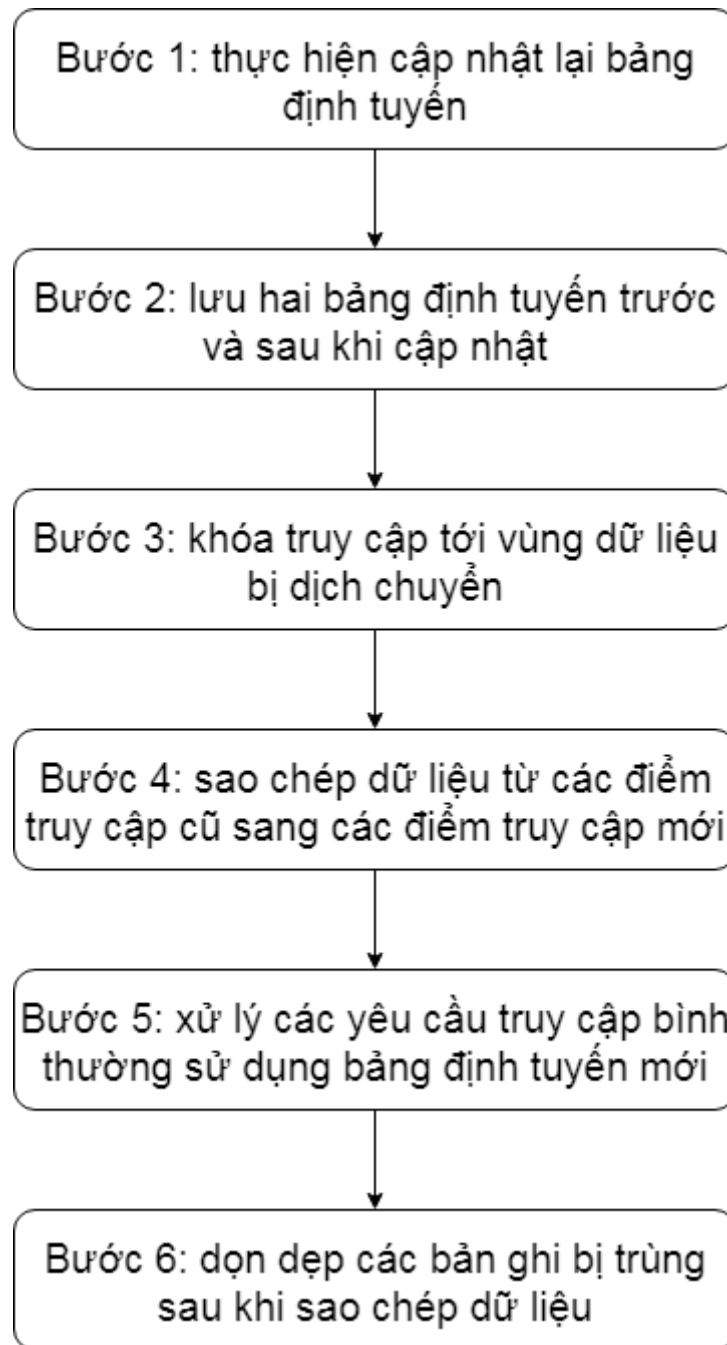
13  
Hình 5



Hình 6



Hình 7



Hình 8