

Chương II Các hệ cơ sở dữ liệu phân tán

Nguyễn Kim Anh
anhnk_fti@mail.hut.edu.vn

Bộ môn Hệ thống Thông tin, SoICT

1

Nội dung

- Tổng quan về các hệ CSDLPT
- Phân đoạn dữ liệu
- Biểu diễn các yêu cầu với các mức trong suốt khác nhau
- Thiết kế CSDLPT
- Xử lý và tối ưu hóa truy vấn phân tán
- Quản trị giao dịch và điều khiển tương tranh

2

Giao dịch

- Giao dịch bao gồm một tập các thao tác mà được xử lý như một đơn vị không chia cắt được
 - Đảm bảo tính nhất quán
 - Trong suốt đối với sự thực hiện đồng thời có cạnh tranh
 - Trong suốt đối với các lỗi
- Giao dịch phân tán là một giao dịch yêu cầu thực hiện trên nhiều trạm của mạng

Các tính chất của giao dịch

- **Nguyên tử (Atomicity)**—một giao dịch là một đơn vị thực hiện nguyên tử → hoặc tất cả các thao tác được thực hiện hoặc không một thao tác nào được thực hiện
- **Nhất quán (Consistency preservation)**—một giao dịch luôn luôn biến đổi CSDL từ một trạng thái nhất quán này đến một trạng thái nhất quán khác
- **Độc lập (Isolation)**—một giao dịch cần được thực hiện một cách độc lập với các giao dịch khác → các kết quả bộ phận của một giao dịch chưa hoàn thành là không thể thấy được đối với các giao dịch khác
- **Bền vững (Durability or permanency)**—các kết quả (các cập nhật) của một giao dịch hoàn thành được ghi nhận một cách bền vững trong CSDL → các cập nhật không bị mất đi dù có các lỗi xảy ra.

Các cấu trúc giao dịch

- Cấu trúc phẳng

```

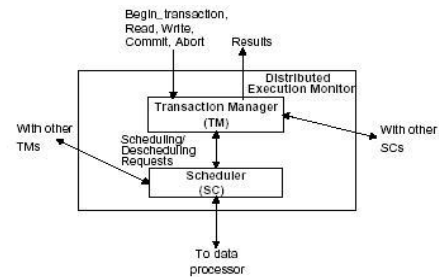
Begin_transaction
  T1();
  T2(); .....
End_transaction
  
```

- Cấu trúc lồng

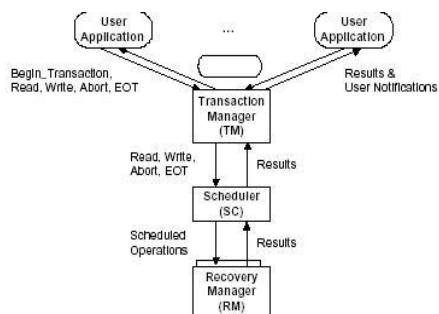
```

Begin_transaction
  Begin_transaction T1
    Begin_transaction T2
      T3(); .....
    End_transaction T2
  End_transaction T1
End_transaction
  
```

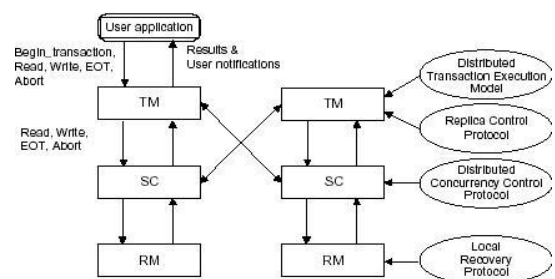
Kiến trúc tham chiếu



Sự thực hiện giao dịch tập trung



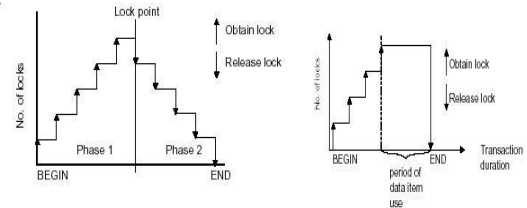
Sự thực hiện giao dịch phân tán



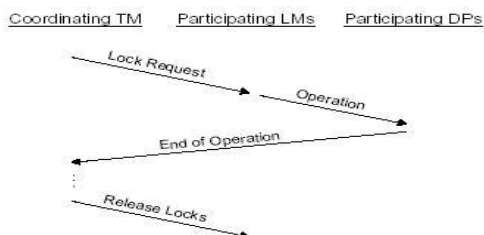
Các thuật toán điều khiển tương tranh

- Bi quan
 - Giao thức khóa 2 pha
 - Giao thức dựa trên nhãn thời gian
 - Lai
- Lạc quan
 - Dựa trên kỹ thuật khóa
 - Dựa trên nhãn thời gian

Giao thức khóa 2 pha (Two-Phase locking-2PL)

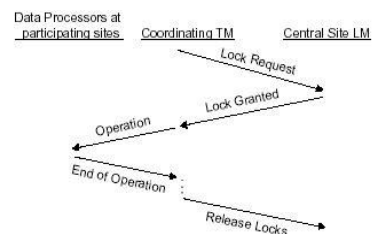


2PL phân tán



- LM (Lock Manager)
- DP (Data Processor)

2PL tập trung



Giao thức dựa trên nhãn thời gian (Timestamp Ordering-TO)

- 2PL là đơn giản và đảm bảo tính khả tuần tự, tuy nhiên, kỹ thuật khóa có thể giảm hiệu năng hệ thống và có thể gây ra khóa chết
- Giao thức dựa trên nhãn thời gian không cố gắng đảm bảo tính khả tuần tự dựa trên kiểm soát độc đoán với thao tác khóa → nó không gây ra khóa chết
- Luật TO: Với 2 thao tác mâu thuẫn O_{ij} and O_{ki} thuộc giao dịch T_i và T_k (T_k trẻ hơn với nhãn lớn hơn), O_{ij} được thực hiện trước O_{ki} nếu và chỉ nếu $ts(T_i) < ts(T_k)$

Thuật toán TO cơ bản

- Giao dịch (T_i) được gán một nhãn thời gian duy nhất tổng thể $ts(T_i)$
- TM gán nhãn cho tất cả các thao tác được phát ra bởi T_i
- Mỗi mục dữ liệu được gán một nhãn đọc và một nhãn ghi:
 - $rts(x)$, $wts(x)$
- for $R_i(x)$
 - If $ts(T_i) < wts(x)$
 - then reject $R_i(x)$
 - else accept $R_i(x)$
 - $rts(x) <- ts(T_i)$
- for $W_i(x)$
 - if $ts(T_i) < rts(x)$ and $ts(T_i) < wts(x)$
 - then reject $W_i(x)$
 - else accept $W_i(x)$
 - $wts(x) <- ts(T_i)$

Thuật toán TO cơ bản

- Thuật toán TO cơ bản là đơn giản và không có khóa chết. Nhược điểm của TO là việc khởi động lại tiềm tàng của giao dịch nhiều lần.
- Việc gán một nhãn thời gian duy nhất tổng thể đối với mỗi giao dịch là một vấn đề không đơn giản.

Thuật toán TO thận trọng

- Thuật toán TO thận trọng cố gắng giảm thiểu các thao tác khởi động lại
- Thuật toán TO thận trọng trì hoãn lại mỗi thao tác cho tới khi có một sự đảm bảo rằng giao dịch chứa nó sẽ không phải khởi động lại

Quản lý khóa chết

- Một số cách giải quyết vấn đề khóa chết: ngăn chặn, tránh, dò tìm và giải quyết.
- Ngăn chặn khóa chết không dễ thực hiện vì nó cần phải có đồ thị tuần tự đầy đủ.
- Một cách tiếp cận tránh khóa chết nổi tiếng trong các Hệ điều hành là Wait-Die & Wound-Die.

Wait-Die & Wound-Wait

WAIT-DIE Rule: If T_i requests a lock on a data item which is already locked by T_j , then T_i is permitted to wait iff $ts(T_i) < ts(T_j)$. If $ts(T_i) > ts(T_j)$, then T_i is aborted and restarted with the same timestamp.

- if $ts(T_i) < ts(T_j)$ then T_i waits else T_i dies
- non-preemptive: T_i never preempts T_j
- prefers younger transactions

WOUND-WAIT Rule: If T_i requests a lock on a data item which is already locked by T_j , then T_i is permitted to wait iff $ts(T_i) > ts(T_j)$. If $ts(T_i) < ts(T_j)$, then T_j is aborted and the lock is granted to T_i .

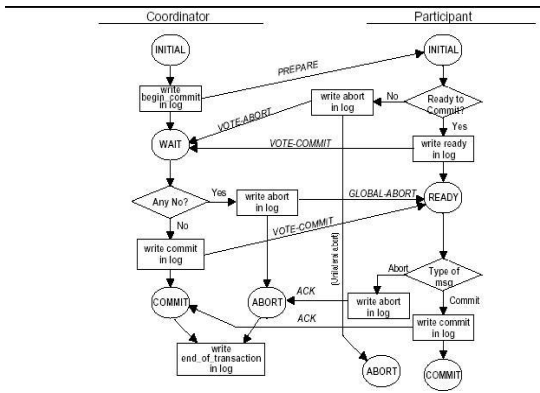
- if $ts(T_i) < ts(T_j)$ then T_j is wounded else T_i waits
- preemptive: T_i preempts T_j if it is younger
- prefers older transactions

Tiếp cận dò tìm khóa chết

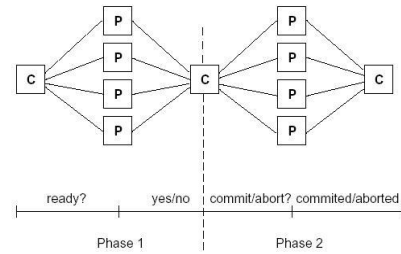
- Bài toán dò tìm khóa chết là NP-đầy đủ để tìm ra cạnh có chi phí tối thiểu với việc phá vỡ chu trình khóa chết.
- Cần xây dựng đồ thị đợi địa phương (Local wait-for graph) và đồ thị đợi tổng thể (Global wait-for graph) để dò tìm các khóa chết phân tán liên quan đến nhiều trạm.
- Các cấu trúc của thuật toán dò tìm khóa chết:
 - Tập trung
 - Phân tán
 - Phân cấp

Giao thức chuyển giao 2 pha (Two-Phase Commit Protocol-2PC)

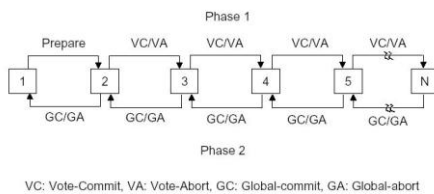
- Các giao thức chuyển giao:
 - Điều khiển sự thực hiện chuyển giao đối với các giao dịch phân tán
 - Đảm bảo tính nguyên tử và tính bền vững
- Luật chuyển giao tổng thể: hoặc tất cả hoặc không.
- Pha 1: Xác định một quyết định chung cho tất cả các thành viên tại các trạm thực hiện giao dịch phân tán
- Pha 2: Cài đặt quyết định chung tại các trạm thực hiện giao dịch phân tán
 - Bộ điều phối (Coordinator)
 - Các thành viên (Participants)



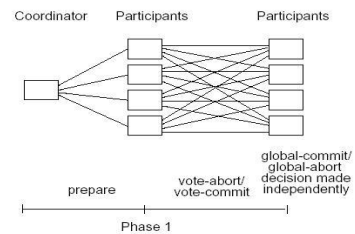
2PC tập trung



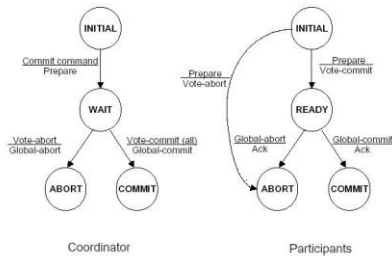
2PC tuyến tính



2PC phân tán



Các trạng thái của giao dịch trong 2PC



Các vấn đề đối với 2PC

- Sự tắc nghẽn
 - Ready kéo theo các thành viên phải đợi bộ điều phối
 - Nếu bộ điều phối có lỗi, các trạm thành viên bị tắc nghẽn cho tới khi phục hồi.
 - Sự tắc nghẽn làm giảm tính sẵn sàng của hệ
- Việc phục hồi độc lập tại các trạm thành viên là không thể thực hiện được
- Giao thức 3PC được đề xuất để giải quyết sự tắc nghẽn.

Giao thức chuyển giao 3 pha (Three-Phase Commit Protocol-3PC)

- Một chứng minh đã chỉ ra rằng điều kiện cần và đủ đối với việc thiết kế các giao thức nguyên tử không tắc nghẽn là:
 - Không có một trạng thái nào kề với cả hai trạng thái commit và abort (2PC vi phạm)
 - Không có một trạng thái không thể chuyển giao (non-committable state) lại kề với trạng thái chuyển giao (commit state) (Abort không thể kế vCommit)

Giao thức chuyển giao 3 pha

