

Câu 1:

### **Phối hợp (Coordination):**

Là quá trình tổ chức và quản lý các thành phần trong hệ thống phân tán để chúng làm việc cùng nhau một cách trật tự nhằm đạt được mục tiêu chung. Nó bao gồm điều phối luồng công việc, chia sẻ tài nguyên, xử lý sự kiện, khóa tài nguyên v.v.

### **Đồng bộ (Synchronization):**

Là một khía cạnh cụ thể trong phối hợp, dùng để đảm bảo rằng các thành phần trong hệ thống có **cùng nhận thức về thời gian hoặc trạng thái hệ thống** để thực hiện hoạt động chính xác, như đồng bộ đồng hồ, thứ tự sự kiện, hoặc truy cập dữ liệu đồng thời.

### **Khác biệt chính:**

- **Phối hợp** là khái niệm rộng hơn, bao gồm cả việc đồng bộ.
- **Đồng bộ** tập trung vào thời gian và trạng thái nhất quán giữa các tiến trình.

Trong hệ thống phân tán, không có đồng hồ chung. Nếu các máy có thời gian khác nhau, thứ tự sự kiện có thể bị hiểu sai → ảnh hưởng đến logic và tính nhất quán.

## **Giải thuật Cristian**

### **◆ Nguyên tắc hoạt động:**

1. Máy khách gửi yêu cầu lấy thời gian đến máy chủ (có đồng hồ chính xác).
2. Máy chủ phản hồi lại thời gian hiện tại của nó.
3. Máy khách ước lượng độ trễ mạng và điều chỉnh đồng hồ dựa trên thời gian nhận được và thời gian gửi yêu cầu.

### **◆ Ước lượng độ trễ mạng và hiệu chỉnh:**

- Gọi:
  - $T_0$ : thời điểm gửi yêu cầu
  - $T_1$ : thời điểm nhận phản hồi
  - $T_s$ : thời gian máy chủ trả lời (thời gian được gửi)
- Độ trễ ước tính:  $(T_1 - T_0) / 2$
- Đồng hồ máy khách =  $T_s + ((T_1 - T_0) / 2)$

➡ Máy khách giả định rằng thời gian đi và về của gói tin là như nhau.

## 4. Giải thuật Berkeley

### Nguyên tắc hoạt động:

1. Một máy chủ (coordinator) định kỳ gửi yêu cầu thời gian đến tất cả các máy khác.
2. Các máy phản hồi lại thời gian của mình.
3. Máy chủ tính trung bình thời gian của tất cả các máy (kể cả chính nó), **trừ các máy có độ lệch lớn bất thường** (outlier).
4. Máy chủ gửi hiệu chỉnh về cho mỗi máy.

### So sánh với Cristian:

Đặc điểm	Cristian	Berkeley
Vai trò máy chủ	Có đồng hồ chuẩn	Máy chủ không cần đồng hồ chuẩn
Cơ chế đồng bộ	Máy khách lấy thời gian	Máy chủ thu thập và tính trung bình
Ưu điểm	Đơn giản, dễ cài đặt	Không cần đồng hồ chuẩn, đồng bộ nhóm
Nhược điểm	Phụ thuộc vào độ chính xác mạng	Tính toán phức tạp, nhạy cảm với lỗi

## 5. Giải thuật trung bình và tham chiếu quảng bá

### Giải thuật đồng bộ trung bình (Average Synchronization):

- Các máy gửi thời gian hiện tại cho nhau.
- Mỗi máy tính trung bình thời gian và tự hiệu chỉnh đồng hồ.

✅ **Ưu điểm:** Không cần máy chủ trung tâm, phân tán đều.

❌ **Nhược điểm:** Dễ bị ảnh hưởng bởi lỗi hoặc máy gửi thời gian sai.

### Giải thuật đồng bộ tham chiếu quảng bá (Reference Broadcast Synchronization - RBS):

- Một máy phát tín hiệu quảng bá không mang thời gian (broadcast).

- Các máy nhận cùng lúc gói tin, và **so sánh thời gian nhận để đồng bộ lẫn nhau.**

#### **Ưu điểm:**

- Loại bỏ sai số do độ trễ truyền đi vì không có timestamp trong gói tin.

#### **Nhược điểm:**

- Phụ thuộc vào khả năng nhận cùng lúc → nhạy cảm với nhiễu.
- Chỉ đồng bộ tương đối giữa các máy thu, không với máy phát.

Câu 3 :

### **Tương tranh (Race Condition):**

Là hiện tượng xảy ra khi **hai hoặc nhiều tiến trình cùng truy cập và thao tác trên một tài nguyên chia sẻ**, và **thứ tự thực hiện không được kiểm soát**, dẫn đến hành vi **không xác định hoặc lỗi logic**.

Ví dụ: 2 tiến trình cùng lúc rút tiền từ một tài khoản ngân hàng mà không có khóa → số dư bị tính sai.

#### **◆ Nguyên tắc loại trừ lẫn nhau (Mutual Exclusion):**

Là nguyên tắc đảm bảo rằng **tại một thời điểm chỉ có một tiến trình được phép truy cập tài nguyên dùng chung** (critical section).

## **2. Vì sao cần giải thuật loại trừ tương hỗ trong hệ phân tán?**

Trong hệ thống phân tán:

- Các tiến trình **chạy trên nhiều máy** → không có bộ nhớ chung.
- Không có **đồng hồ hệ thống dùng chung**.

- Cần cơ chế đảm bảo **truy cập tài nguyên chia sẻ (file, cơ sở dữ liệu, dịch vụ...)** **không bị xung đột**.

→ Vì vậy, cần **giải thuật loại trừ tương hỗ** để:

- Tránh tình trạng **tương tranh**.
- Đảm bảo **tính đúng đắn, nhất quán và an toàn hệ thống**.

### 3. Giải thuật loại trừ tương hỗ tập trung (Centralized Algorithm)

#### ◆ Nguyên lý hoạt động:

- Một tiến trình trung tâm (coordinator/server) quản lý quyền truy cập tài nguyên.
- Mỗi tiến trình muốn vào vùng tới hạn sẽ **gửi yêu cầu đến coordinator**.
- Coordinator:
  - Nếu tài nguyên đang rảnh → gửi **OK (grant)** cho tiến trình.
  - Nếu đang bận → **xếp hàng** các yêu cầu.

→ Khi tiến trình hoàn tất, nó thông báo “xong” để coordinator **chuyển quyền cho tiến trình kế tiếp**.

#### ◆ Ưu điểm:

- Dễ cài đặt, đơn giản.
- Chỉ cần **3 thông điệp** cho mỗi lần truy cập: request – grant – release.

#### ◆ Nhược điểm:

- **Điểm nghẽn cổ chai (bottleneck)**: coordinator xử lý nhiều yêu cầu.
- **Dễ lỗi nếu coordinator bị treo**.
- Không mở rộng tốt.

## 4. Giải thuật phi tập trung (Voting Algorithm / Decentralized)

### ◆ Nguyên lý hoạt động:

- Có **N tiến trình**, mỗi tiến trình chỉ được vào vùng tới hạn nếu nhận được **ít nhất  $(N/2 + 1)$  phiếu** đồng ý.
- Tiến trình gửi yêu cầu đến tất cả các node.
- Các node gửi lại **phiếu chấp thuận** nếu họ chưa cấp quyền cho tiến trình nào khác.
- Sau khi dùng xong, tiến trình gửi **giải phóng** để các node cho người khác mượn quyền.

### ◆ Vấn đề khi cạnh tranh mạnh:

- Nếu nhiều tiến trình gửi request đồng thời → **xung đột**, không ai nhận đủ số phiếu → “loạn” (deadlock hoặc starvation).

### ✓ Giải pháp:

- **Sắp xếp thứ tự ưu tiên.**
- **Thêm timeout hoặc vòng lặp lại** khi không nhận đủ phiếu.
- Kết hợp với **quorum quét luân phiên (rotating coordinator)**.

## 5. Giải thuật phân tán (Lamport's Algorithm)

### ◆ Nguyên lý hoạt động:

- Dựa trên **nhãn thời gian logic (Lamport Timestamp)** để sắp thứ tự yêu cầu truy cập tài nguyên.
- Mỗi tiến trình gửi yêu cầu đến tất cả tiến trình khác kèm theo thời gian logic.
- Mỗi tiến trình phản hồi lại **ACK (chấp thuận)**.
- Tiến trình chỉ được truy cập tài nguyên nếu:
  - Nhận đủ ACK từ tất cả các tiến trình.
  - Yêu cầu của nó là **sớm nhất trong hàng đợi** (dựa trên timestamp).

♦ **Ưu điểm:**

- Không cần coordinator.
- Không có điểm chết đơn (single point of failure).

♦ **Nhược điểm:**

- **Gửi nhiều thông điệp:** mỗi truy cập cần  **$3(N - 1)$**  thông điệp.
- **Phức tạp** trong việc duy trì hàng đợi theo thời gian.

## 6. So sánh các giải thuật loại trừ tương hỗ

Tiêu chí	Tập trung	Phi tập trung	Phân tán (Lamport)	Thẻ bài (Token-based)
<b>Thông điệp (per access)</b>	3	$\sim 2N$	$3(N-1)$	$N$ (broadcast) hoặc 1
<b>Độ trễ</b>	2 msg	$\geq 2$ msg	2 msg + xử lý queue	1 (nếu có token), $\leq N-1$
<b>Điểm chết (Deadlock)</b>	Có nếu coordinator treo	Có nếu xung đột vote	Có thể nếu mất ACK	Có thể nếu mất token
<b>Starvation</b>	Không nếu có hàng chờ	Có thể	Có thể	Có thể
<b>Ưu điểm</b>	Đơn giản	Phân tán, không điểm chết đơn	Trật tự hợp lý, không server	Truy cập nhanh nếu có token
<b>Nhược điểm</b>	Bottleneck, dễ treo	Nhiều xung đột vote	Nhiều thông điệp	Phải giữ token, mất token là lỗi

Câu 5 :

Trong hệ thống phân tán quy mô lớn (như các hệ thống toàn cầu), việc biết vị trí của các tiến trình là **rất quan trọng** để:

- **Tối ưu hóa hiệu năng:** gửi dữ liệu đến node gần hơn giúp giảm độ trễ.
- **Cân bằng tải theo vị trí địa lý.**

- **Chọn máy chủ gần nhất** trong các hệ thống CDN (Content Delivery Network).
- **Bảo mật và kiểm soát truy cập** dựa theo vùng (geofencing).
- **Hỗ trợ các ứng dụng cần vị trí**: giao thông, logistics, AR/VR, IoT,...

## 2. Kiến trúc và nguyên lý hoạt động của hệ thống GPS

### ◆ Kiến trúc tổng thể:

- Gồm **24 đến 32 vệ tinh** hoạt động đồng thời (tối thiểu 4 phủ mọi nơi).
- Vệ tinh chia thành **6 mặt phẳng quỹ đạo** nghiêng  $55^\circ$ , mỗi mặt chứa 4–6 vệ tinh.
- Mỗi vệ tinh quay quanh Trái Đất  $\sim 2$  vòng/ngày (khoảng 12 giờ/quỹ đạo).

### ◆ Đồng hồ nguyên tử:

- Mỗi vệ tinh được trang bị **đồng hồ nguyên tử cực kỳ chính xác** (sai số  $\sim 10^{-9}$  giây).
- Cung cấp **thời gian tuyệt đối** cho các tín hiệu quảng bá.

### ◆ Thông điệp quảng bá:

Mỗi vệ tinh phát một tín hiệu chứa:

- Thời gian phát tín hiệu ( $T_i$ ).
- Tọa độ vị trí vệ tinh tại thời điểm đó ( $x_i, y_i, z_i$ ).
- Dữ liệu quỹ đạo (ephemeris) và trạng thái hệ thống.

## 3. Xác định vị trí thiết bị từ tín hiệu GPS

### ◆ Giả sử có 4 vệ tinh $S_1, S_2, S_3, S_4$ với:

- Vị trí:  $(x_i, y_i, z_i)$
- Thời gian gửi tín hiệu:  $T_i$
- Thiết bị R nhận tại thời điểm  $T_r$ , nhưng do đồng hồ thiết bị bị lệch  $\Delta r$ , thời gian thực là  $T_r - \Delta r$

◆ **Phương trình khoảng cách (cho từng vệ tinh):**

$$(x_i - x_r)^2 + (y_i - y_r)^2 + (z_i - z_r)^2 = c^2 \cdot (T_{ri} - \Delta r - T_i)^2$$

$$\sqrt{(x_i - x_r)^2 + (y_i - y_r)^2 + (z_i - z_r)^2} = c \cdot (T_{ri} - \Delta r - T_i)$$

- $c$ : vận tốc ánh sáng ( $\sim 3 \times 10^8$  m/s).
- Có 4 phương trình, 4 ẩn:  $x_r, y_r, z_r, \Delta r$ .

➔ Cần ít nhất 4 vệ tinh để giải hệ này và xác định vị trí 3D và hiệu chỉnh đồng hồ.

## 4. Các nguồn sai số chính trong GPS

Nguồn sai số	Ảnh hưởng	Cách khắc phục
Đồng hồ thiết bị lệch	Gây sai số lớn nếu không có hiệu chỉnh	Sử dụng thuật toán hiệu chỉnh (xem phần trên)
Sai số quỹ đạo	Vệ tinh không đúng vị trí tính toán	GPS cập nhật dữ liệu quỹ đạo liên tục
Trễ truyền tín hiệu	Do tầng ion/quyển gây khúc xạ → tín hiệu đi đường vòng	Dùng mô hình ion/quyển để hiệu chỉnh
Đa đường (multipath)	Tín hiệu phản xạ làm lệch thời gian đến	Lọc tín hiệu, chọn vệ tinh tốt
Nhiều từ môi trường	Làm sai số nhỏ, nhưng cộng dồn → vị trí sai vài mét	Sử dụng nhiều vệ tinh, lọc trung bình

**Độ chính xác GPS thương mại** thường trong khoảng **5–10 mét**, GPS vi sai (DGPS) có thể chính xác  $< 1\text{m}$ .

## 5. Phương pháp định vị thay thế GPS trong hệ phân tán

Phương pháp	Nguyên lý	Ưu điểm	Hạn chế
<b>Định vị bằng độ trễ mạng (Network Latency)</b>	Đo thời gian ping giữa các node → suy vị trí tương đối	Không cần phần cứng, ứng dụng nội bộ tốt	Độ trễ không ổn định, chỉ ra tương đối
<b>Hệ tọa độ ảo Vivaldi</b>	Mỗi node duy trì tọa độ trong không gian ảo → cập nhật	Phân tán hoàn toàn,	Vị trí ảo, không phải vị trí địa lý thật



	qua trao đổi với các node khác	không cần GPS	
<b>IP Geolocation</b>	Suy vị trí từ địa chỉ IP (qua cơ sở dữ liệu IP-Location)	Không cần tín hiệu vật lý, dễ tích hợp	Độ chính xác thấp (tầm thành phố, quốc gia), sai lệch lớn
<b>Wi-Fi / Bluetooth RSSI</b>	Dùng cường độ tín hiệu các AP / beacon → tam giác hóa	Trong nhà tốt, độ chính xác cao (1–3m)	Cần nhiều AP/đèn hiệu, khó mở rộng ngoài trời
<b>Cell Tower Triangulation</b>	Dùng độ mạnh tín hiệu từ trạm phát di động	Hữu ích ở nơi không có GPS	