BESOINS DES APPLICATIONS REPARTIES

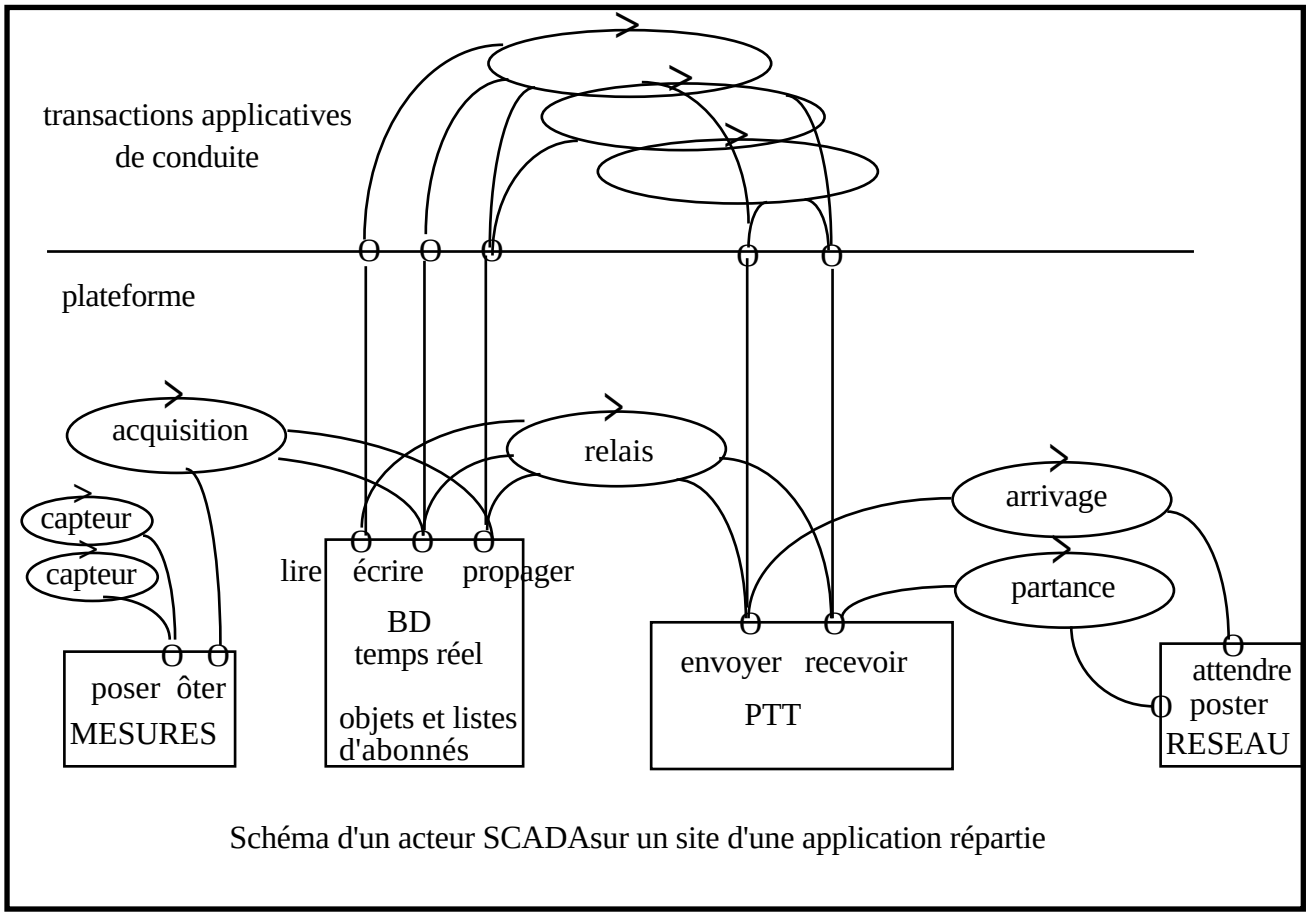


Schéma d'un acteur SCADAsur un site d'une application répartie

-

APPLICATION REPARTIE = ENSEMBLE DE SITES

CHAQUE SITE SCADA COMPREND

UNE PLATE-FORME AVEC:

des modules d'acquisition : captures concurrentes, acquisition synthétique

des bases de données temps réel : lecteurs rédacteurs en mémoire centrale

une messagerie : producteurs consommateurs

un module réseau : communication de messages intersites

des processus de service

UNE COUCHE APPLICATIVE AVEC :

des processus appelés transactions applicatives

BESOINS DES APPLICATIONS REPARTIES

REPARTITION SIMPLE (CLIENT SERVEUR)

accès local ou distant aux BD TR, chaque BD cohérente individuellement
abonnement à BD primaire: copies secondaires pour lecture, sur autres sites
transactions avec accès à une seule BD primaire à la fois
messagerie entre les processus de divers sites

REPARTITION PLUS COMPLEXE (APPLICATION COOPÉRATIVE)

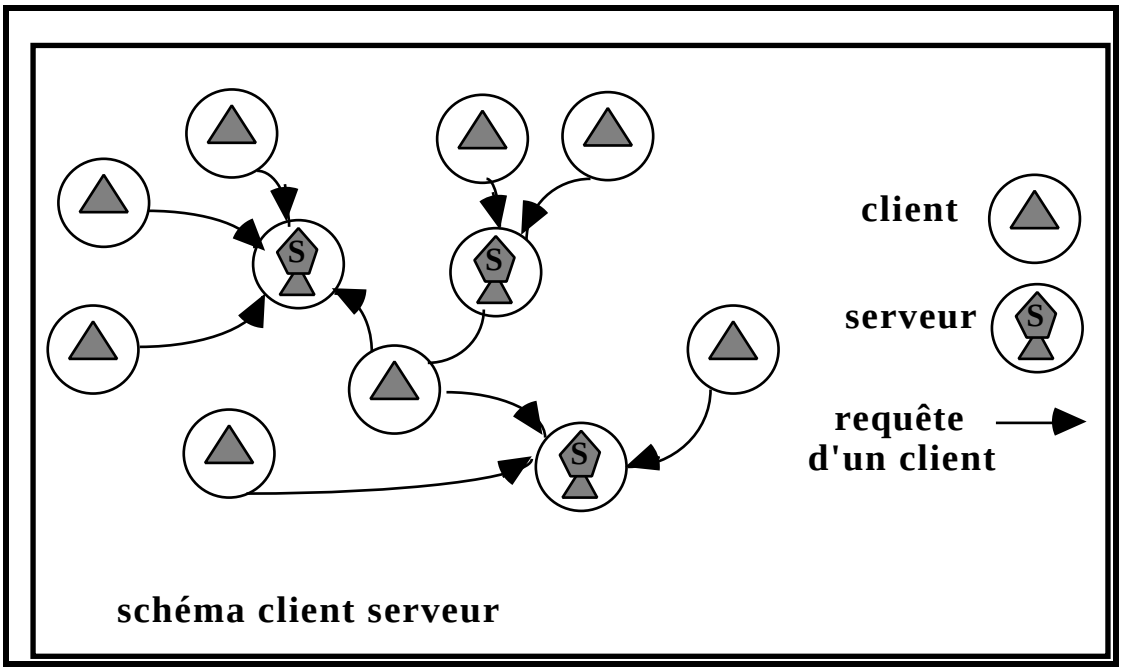
Transactions avec accès emboîtés à plusieurs BD :
problèmes de cohérence globale et interblocage

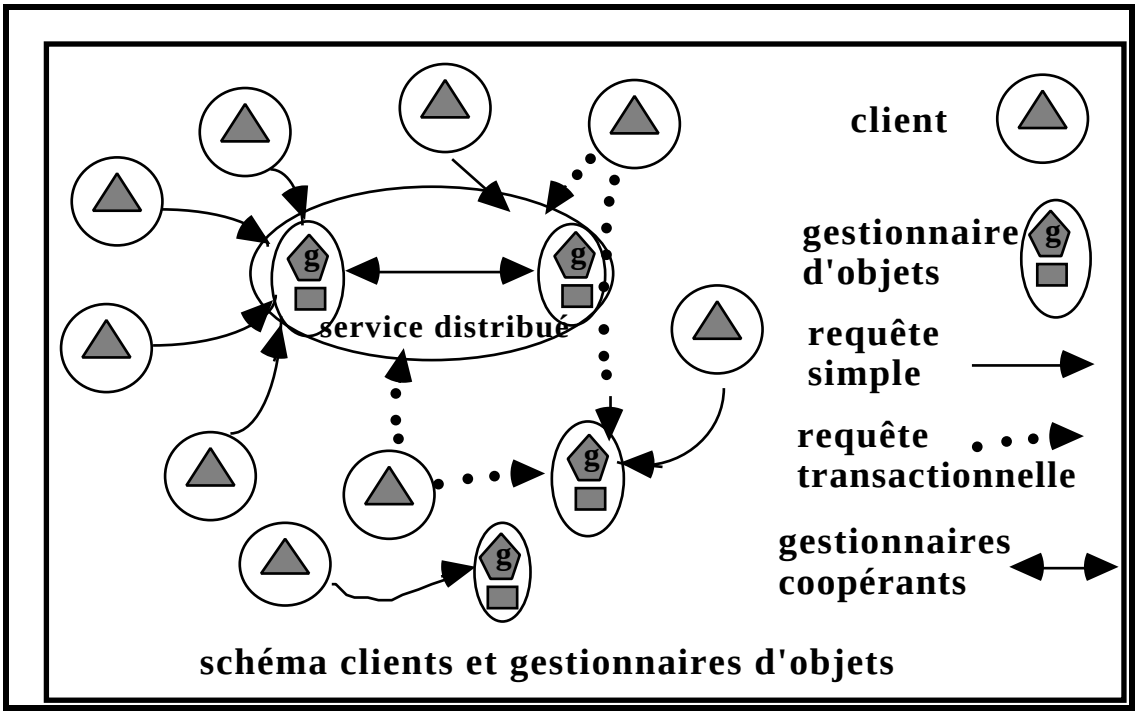
Copies multiples d'une même BD avec écritures sur chaque copie : cohérence faible ou forte (problème des caches multiples)

Ensemble de transactions coopérantes. Problèmes de synchronisation :
démarrage dans un état cohérent
coordination par un site fixe, mais si absent (panne, maintenance) alors élection d'un nouveau coordinateur
terminaison de la coopération
mise au point répartie
points de reprise cohérents
diffusion d'information dans un groupe

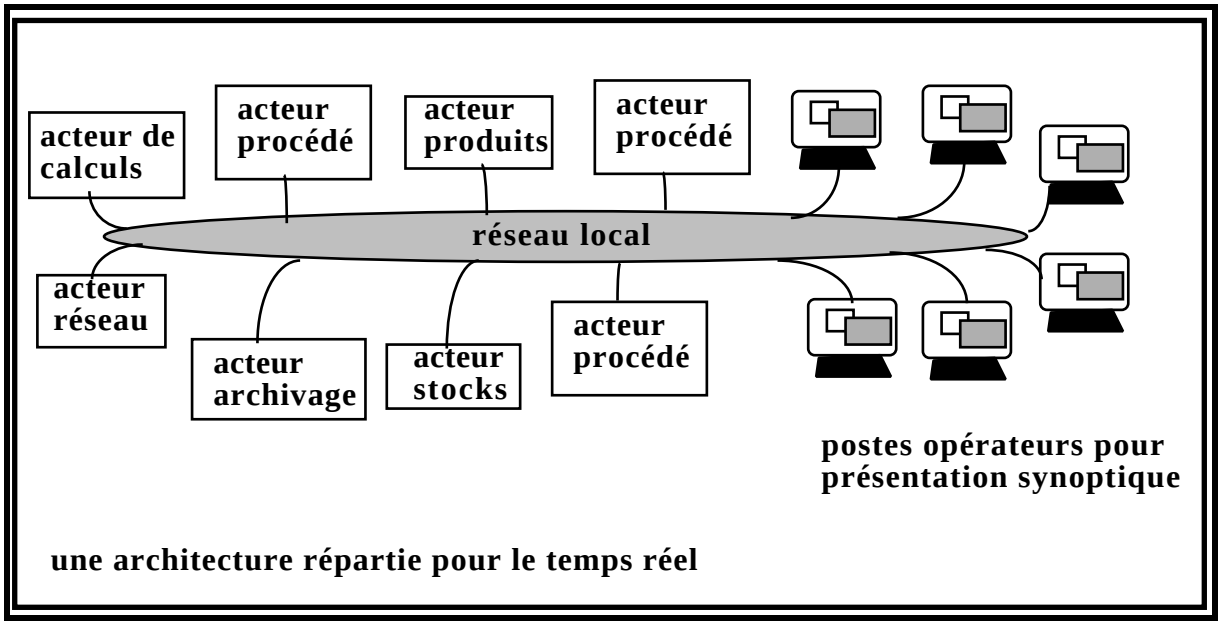
Diffusion fiable et ordonnée à un groupe de processus sur des sites divers
Mobilité des sites

BESOINS DES APPLICATIONS REPARTIES

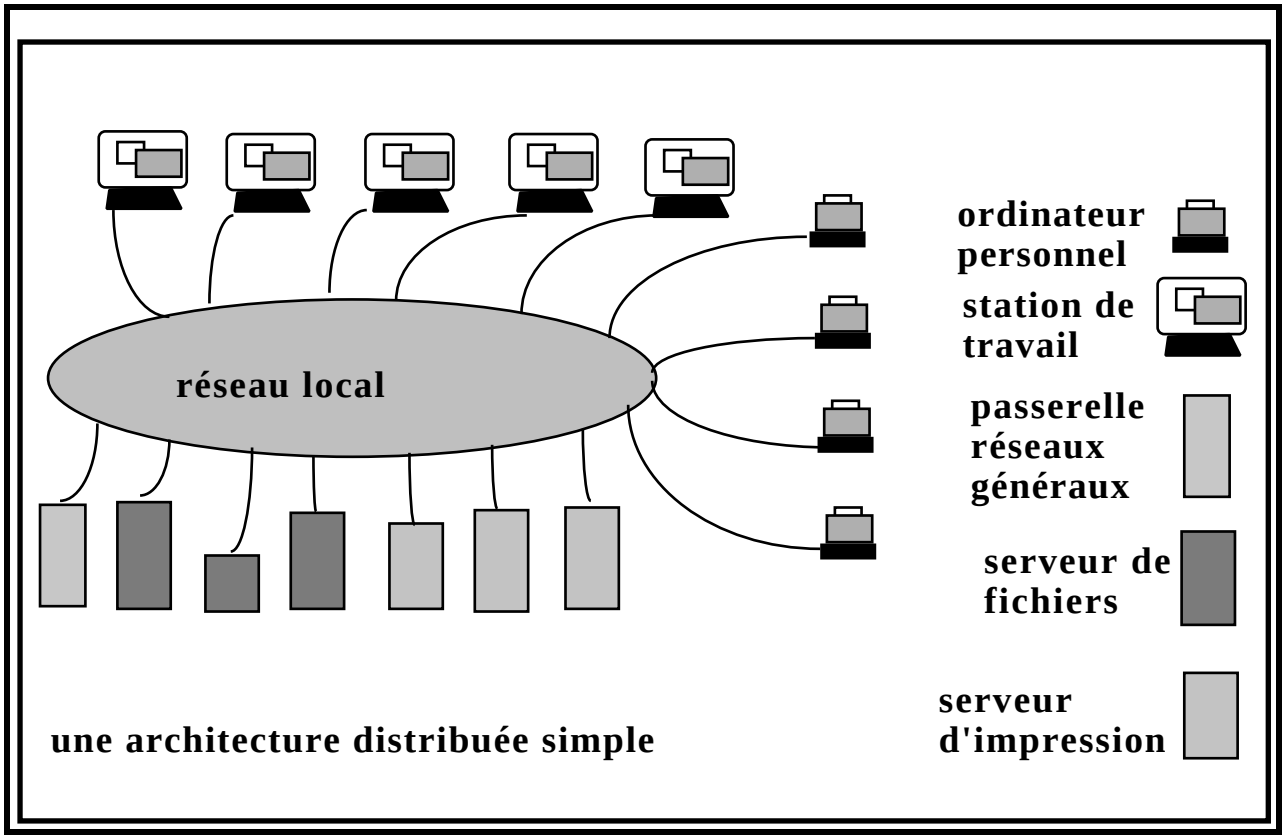




EXEMPLES D'ARCHITECTURE PHYSIQUE
(ARCHITECTURE ORGANIQUE)



une architecture répartie pour le temps réel



une architecture distribuée simple

LE REEL DES COMMUNICATIONS ELEMENTAIRES

POINT A POINT MODE MESSAGE

message d'un émetteur vers un récepteur sur un canal

perte de message, absence de récepteur (panne, maintenance,...)

contrôle d'erreur par acquit et délais de garde, mais duplication possible

numérotation des messages successifs

réception désordonnée d'une suite de messages

contrôle de flux pour asservir les vitesses de l'émetteur et du récepteur (producteur-consommateur)

incertitude sur l'état du canal, de l'émetteur et du récepteur

durées de transfert variable, asynchrone (mais il existe des bus synchrones)

POINT A POINT MODE CLIENT SERVEUR

l'émetteur est le client et le récepteur est le serveur

message requête d'un client vers un serveur sur un canal

message réponse du serveur au client

le client n'envoie pas d'autre requête avant d'avoir reçu la réponse

mêmes problèmes d'erreurs, d'incertitudes et de variabilité des délais

DIFFUSION A UN GROUPE

Un émetteur et N récepteurs sur le canal

exemples : Ethernet, Token ring

perte de message pour R récepteurs avec 0 <= R <= N

d'une émission à l'autre perte sur des récepteurs différents

pas le même ordre sur tous les sites pour une suite de réceptions

pas de perception unique de l'ordre d'émission si émetteurs différents

durées de transfert variable

incertitudes sur l'état du canal, de l'émetteur et des récepteurs

incertitude sur la composition du groupe

LES SYSTÈMES RÉPARTIS

Ensemble fini de sites interconnectés par un réseau de communication
Pas de mémoire commune
Pas d'horloge physique partagée par 2 processus ou plus

CHAQUE SITE

• processeur, mémoire locale, mémoire stable (permanence des données en dépit de défaillances du processeur)

RÉSEAU DE COMMUNICATION

• connexe : tout processus peut communiquer avec tous les autres
• communication et synchronisation entre processus par messages via le réseau de communication

DIFFICULTÉS CARACTÉRISTIQUES

OBSERVATIONS :
deux observations faites sur deux sites distincts peuvent différer
- **par l'ordre de perception des événements**
- **par leur date**

DÉCISIONS COHÉRENTES ENTRE PLUSIEURS SITES
• visions différentes de l'état des ressources du système
• pas d'état global de référence en temps réel au moment où il faut prendre des décisions
• risque accru de défaillance (plus de composants)
=> la défaillance d'un élément n'est pas un événement rare
=> importance des hypothèses sur les défaillances possibles

attention, à ne pas confondre avec les
SYSTÈMES PARALLÈLES CENTRALISÉS
encore appelés des multiprocesseurs à mémoire commune
• existence d'une mémoire commune (physique, réflexive, virtuelle)
• existence d'une horloge ou d'un rythme commun

LES MODELES DE LA COMMUNICATION ELEMENTAIRE

MODÈLE DE COMMUNICATION SYNCHRONE FIABLE

hypothèses les plus fortes

Tout message arrive avant le délai d_{max}, qu'il soit point à point ou diffusé.

Communication fiable : pas de perte de message, pas de panne de site

Réseau connexe : tout site peut communiquer avec tous les autres
parfois réseau isotrope : même délai d_{max} pour tous les canaux

Les horloges des sites sont aussi synchronisées
(leur écart est borné par dh_{max})

Les vitesses des processeurs sont supérieures à un minorant connu

Exemple : réseaux locaux avec heure reçue par radio sur chaque site, réseaux locaux industriels avec synchronisation d'horloge véhiculée par le réseau, réseaux locaux avec redondance des bus et des processeurs

• Mais cette hypothèse, valable avec une certaine probabilité, n'est pas toujours réaliste :
probabilité trop faible pour l'application,
probabilité variable dans le temps (cas des réseaux dont la charge instantanée est totalement imprévisible)

ELECTION EN COMMUNICATION SYNCHRONE FIABLE

Les sites S1, S2,..., Si, ..., SN doivent élire un site coordonnateur
Chaque site Si a un identificateur unique uid(i)
et peut diffuser un message <élection, i, uid(i))>
Tous les sites démarrent l'élection en même temps à P
(à dates fixes, par exemple, élection périodique)

Soit $T = d_{max} + dh_{max}$,
Chaque site Si
(i) s'attend à recevoir un message d'élection à partir de P
(ii) s'il n'a rien reçu au bout de T * uid(i) secondes, mesurées sur son horloge, il diffuse son message d'élection.
Résultat : le premier site qui diffuse son message est l'élu
Cet algorithme synchrone élit le site de plus petit uid
commentaire : simple n'est-il pas? mais l'hypothèse synchrone est restrictive (pas de pannes, horloges communes) la "nature" est asynchrone.

ELECTION EN COMMUNICATION SYNCHRONE NON FIABLE

hypothèses : processeurs et canaux à silence sur défaillance
(pas de panne transitoire ou byzantine, mais silence après i, ii ou iii)
Chaque site Si
(i) à P, diffuse son uid(i) aux autres sites, lui compris
(ii) à P + T, il diffuse le min(i) des uid(j) qu'il reçoit avant P + T,
(iii) à P + 2*T, s'il note un consensus des min(j) reçus (selon les cas, la majorité, ou l'unanimité des valeurs reçues), il élit la valeur de consensus
(si l'élu est tombé en panne après sa phase i, il faut recommencer)

MODÈLE DE COMMUNICATION ASYNCHRONE

CHAQUE SITE

- **processeur, mémoire locale, mémoire stable**
  (permanence des données en dépit de défaillances du processeur)

RÉSEAU DE COMMUNICATION

- **connexe : tout processus peut communiquer avec tous les autres**
- **communication et synchronisation entre processus par messages**
  via le réseau de communication

PROPRIÉTÉS CARACTÉRISTIQUES

- **Pas de mémoire commune**
- **Pas d'horloge physique partagée par 2 processus ou plus**
- **Absence de majorant connu sur le temps de transfert des messages**
- **Absence de minorant connu sur les vitesses des processeurs**

Nota : appelé aussi modèle à délais non bornés
Exemple : cas de Internet et des réseaux longue distance (WAN)

PROBLÈME. Un processus Pi ne peut pas savoir si un autre processus Pj est défaillant ou si la réponse qu'il attend de Pj est en préparation par Pj (processus très lent) ou encore en route (message très lent). En pratique, il est essentiel d'introduire une notion de temps (temps réel et non temps du processus Pi) : combien de temps Pi doit-il attendre avant de suspecter Pj de défaillance

suspecter une défaillance ≠ détecter une défaillance

MODELE DE COMMUNICATION ASYNCHRONE FIABLE

réseau connexe : tout site peut communiquer avec tous les autres
communication fiable : pas de perte de message, pas de panne de site
réaliste sur une courte durée

PROPRIÉTÉ DE CAUSALITE ELEMENTAIRE

Par la nature physique de la communication, l'émission d'un message sur un site précède nécessairement la réception du message sur le site destinataire. Toute réception d'un message est causée par une émission antérieure. (il ne peut y avoir de réception spontanée de message)

Cette relation causale permet d'établir, dans un système réparti, une relation d'ordre partiel entre l'événement d'émission d'un message sur un site et l'événement de réception du message sur un autre site destinataire. cette relation se note (on lit précède) :

$$\forall m, \text{EMISSION}(m) \rightarrow \text{RECEPTION}(m)$$

(notée "happened before" par Lamport)

Plus exactement la relation est établie lorsque le message a été reçu

$$\forall m, \text{RECEPTION}(m) \Rightarrow (\text{EMISSION}(m) \rightarrow \text{RECEPTION}(m))$$

HYPOTHESES PROPRES A UN CANAL C_{ij}

(canal : liaison point à point entre un émetteur et un récepteur)

DEFINITION : m1 double m2 dans le canal C_{ij} si et seulement si

$$EMISSION_i(m2) \rightarrow EMISSION_i(m1)$$

$$\text{et } RECEPTION_j(m1) \rightarrow RECEPTION_j(m2)$$

PROPRIÉTÉS D'ORDRE DES MESSAGES DANS LES CANAUX

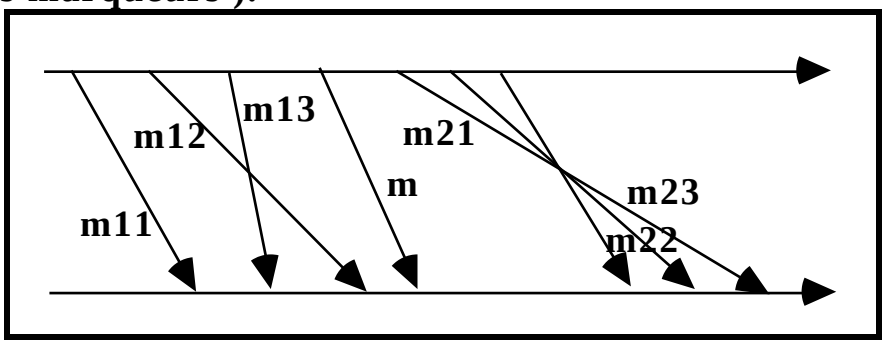
1• un message marqueur m ne peut ni doubler ni être doublé sur C

$\forall m1, \forall m2,$

$EMISSION(m1) \rightarrow EMISSION(m) \Rightarrow RECEPTION(m1) \rightarrow RECEPTION(m)$

$EMISSION(m) \rightarrow EMISSION(m2) \Rightarrow RECEPTION(m) \rightarrow RECEPTION(m2)$

2• un message ordinaire m n'impose pas de condition de réception, mais respecte celles des autres (il ne peut doubler les marqueurs et ne peut être doublé par les marqueurs).



Tout marqueur m sépare les messages du canal en deux sous-ensembles

$<_m = \{m1 \mid EMISSION(m1) \rightarrow EMISSION(m)\}$

$\quad$ et m est un marqueur $\Rightarrow RECEPTION(m1) \rightarrow RECEPTION(m)$

$>_m = \{m2 \mid EMISSION(m) \rightarrow EMISSION(m2)\}$

$\quad$ et m est un marqueur $\Rightarrow RECEPTION(m) \rightarrow RECEPTION(m2)$

TYPES DE COMPORTEMENT D'UN CANAL

1• le moins contraint : tous les messages sont ordinaires

2• le plus contraint : tous les messages sont des marqueurs (canal FIFO)

RELATION DE PRÉCEDENCE ENTRE DES ÉVÉNEMENTS RÉPARTIS

événement : instruction exécutée par un processeur

(précédence : "happened before", L.Lamport, CACM 21,7, 1978)

a) A "précède" A' si A et A' sont des événements qui ont été générés dans cet ordre sur le même site S (ordre d'exécution local) : $A \rightarrow A'$

b) A "précède" A' si A est l'événement d'émission d'un message M par le site P et que A' est l'événement de réception du message M sur Q : $A \rightarrow A'$ (ordre causal pour chaque message)

La relation de précédence dans un système réparti est la fermeture transitive des deux relations précédentes.

$$\text{si } A \rightarrow B \text{ et } B \rightarrow C \text{ alors } A \rightarrow C$$

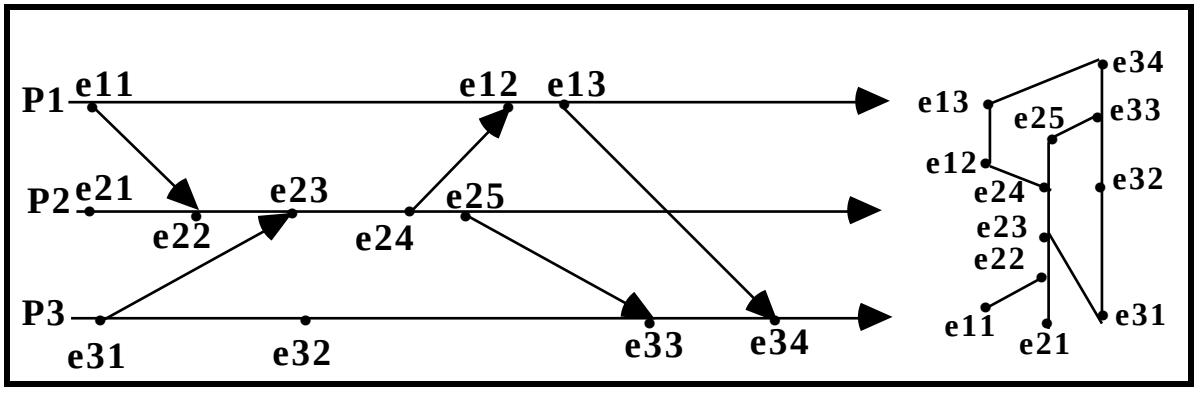
La précédence est un ordre partiel entre les événements du système réparti

La causalité entre événements implique la précédence entre eux :

$$A \text{ cause } B \Rightarrow A \rightarrow B$$

Une précédence entre événements exprime une causalité potentielle :

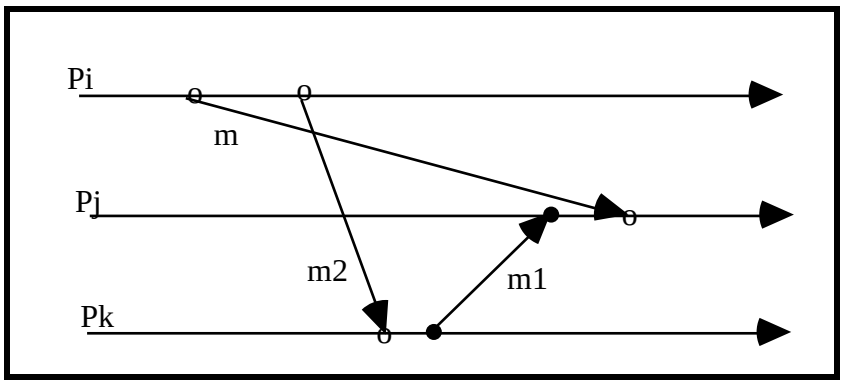
(si $A \rightarrow B$, A peut avoir influencé B).



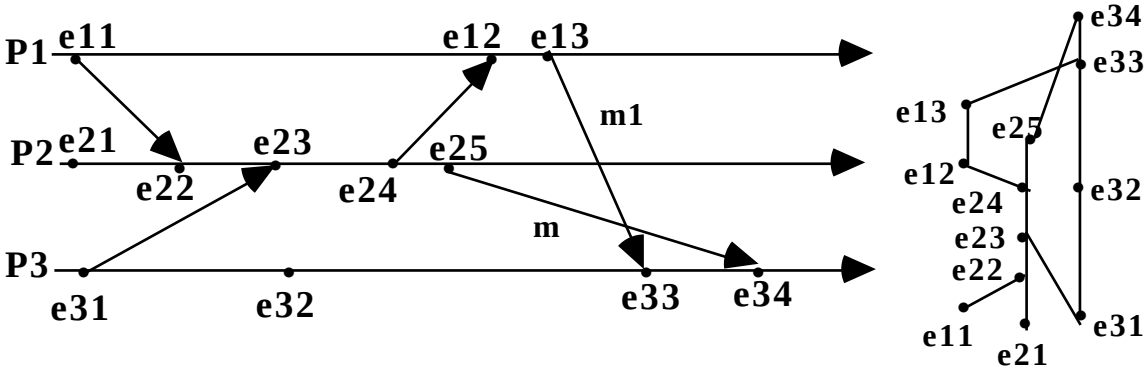
DÉPENDANCE CAUSALE ENTRE MESSAGES

Les messages respectent la dépendance causale si et seulement si :

$\forall Pi, \forall Pj, \forall Pk, \forall m$ émis sur Cij, $\forall m1$ émis sur Ckj,
$EMISSION_i(m) \rightarrow EMISSION_k(m1) \Rightarrow RECEPTION_j(m) \rightarrow RECEPTION_j(m1)$

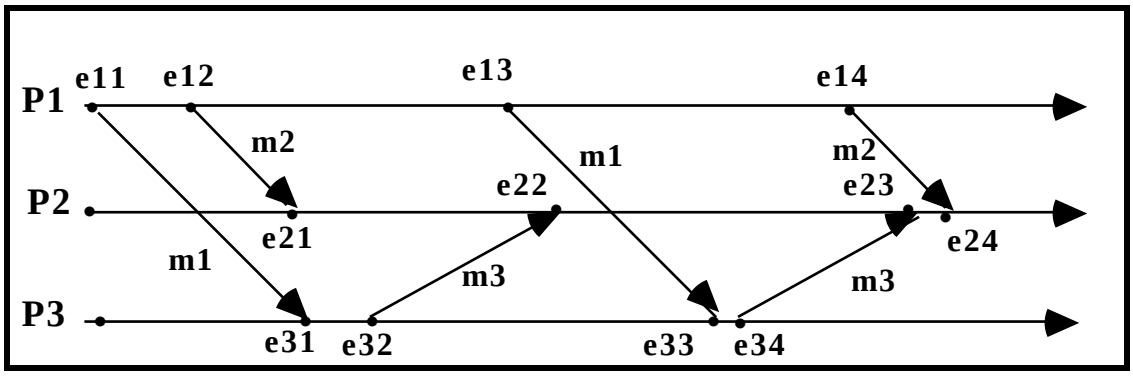


Premier exemple : la réception sur Pj ne respecte pas la dépendance causale



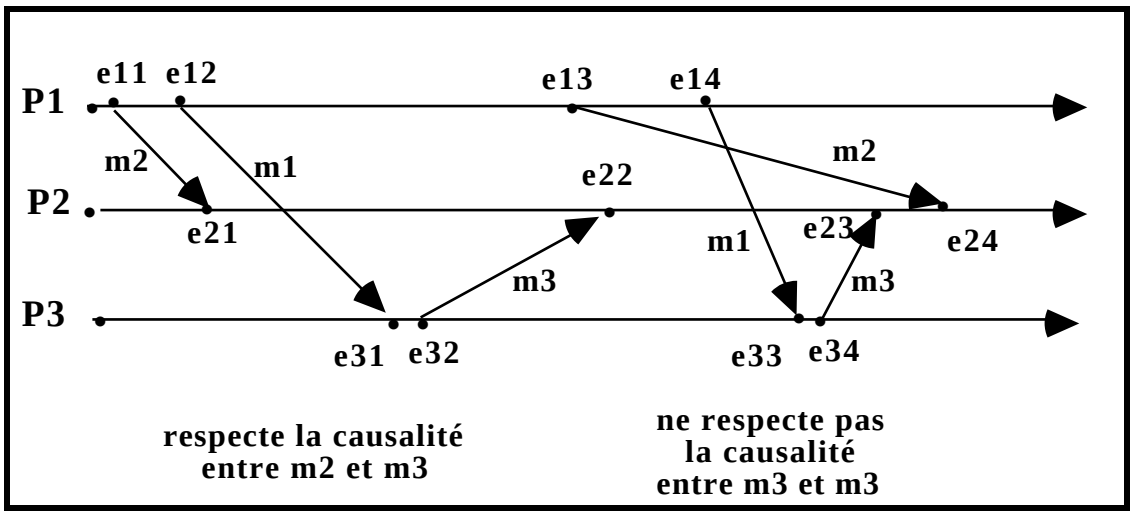
Deuxième exemple : la réception respecte la dépendance causale car les émissions e13 et e25 ne sont pas en relation de précédence

DÉPENDANCE CAUSALE ENTRE MESSAGES
Un problème de logique



pas de dépendance causale entre EMISSION$_1$(m2)→EMISSION$_3$(m3)
Donc il n'y a pas de dépendance causale entre m2 et m3

m1 : "Je <u>vais demander</u> à P2 de te rencontrer. Appelle le de ma part"



EMISSION$_1$(m2)→EMISSION$_3$(m3)⇒RECEPTION$_2$(m2)→ RECEPTION$_2$(m3)
Donc il y a dépendance causale entre m2 et m3

m1 : "<u>J'ai demandé</u> à P2 de te rencontrer. Fixe lui une date et un lieu"

MODELES DE DIFFUSION FIABLE ET COMMUNICATION DE GROUPE

Un message émis doit être reçu par n destinataires.

MODELES DE COMMUNICATION

Communication inclusive ou non (l'émetteur reçoit le même message - le sien enrichi par le réseau- que les récepteurs)

Communication interne ou externe (l'émetteur, client du groupe, n'appartient pas au groupe)

CLASSIFICATION SELON LES EMETTEURS ET LES RECEPTEURS (classification OSI)

MODE CENTRALISE ("multicast")

Un seul émetteur (toujours le même) et n récepteurs.

MODE CENTRALISE A CENTRE MOBILE

L'émetteur est unique par périodes.

MODE MULTI-CENTRE

N processus émetteurs peuvent à tout instant effectuer une diffusion vers P récepteurs.

MODE DECENTRALISE OU MODE CONVERSATION

Un ensemble de N sites peuvent être à tout instant émetteurs et sont tous destinataires des messages.

PROPRIETES D'ORDRE DANS LES GROUPES

DIFFUSION RESPECTANT L'ORDRE LOCAL

Pour deux diffusions successives du même processus, les messages sont délivrés dans le même ordre sur chaque site distant

DIFFUSION RESPECTANT L'ORDRE CAUSAL

diffusion + causalité (Birman et Joseph 1987)

Relation de dépendance causale entre les messages, généralisée à la diffusion

Toute suite de diffusions de messages en relation de causalité implique la délivrance des messages sur tous les sites destinataires dans la même relation de causalité

$\forall Pi, \forall Pk, \forall m$

DIFFUSION$_i$(m) précède causalement DIFFUSION$_k$(m1)

$\Rightarrow$ RECEPTION$_j$(m) précède RECEPTION$_j$(m1) pour tout P_j.

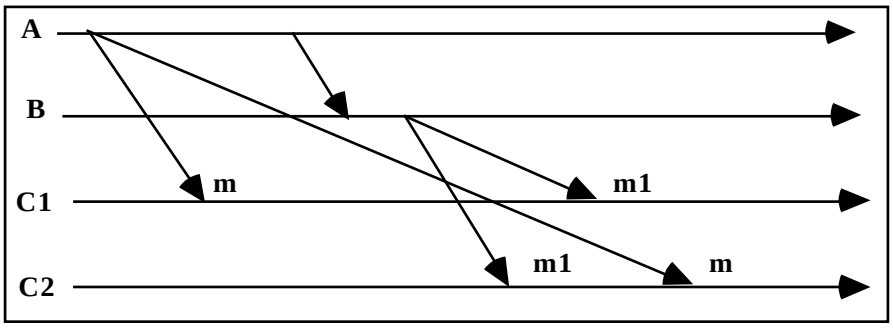
EXEMPLES D'UTILISATION DE LA DIFFUSION CAUSALE

Exemple 1:

A diffuse un courrier électronique m à C1 et C2 qui contient: "je demande à B de vous diffuser du travail par courrier électronique ".

Pour respecter l'ordre causal, C1 et C2 ne doivent pas recevoir le courrier m1 de B avant celui m de A.

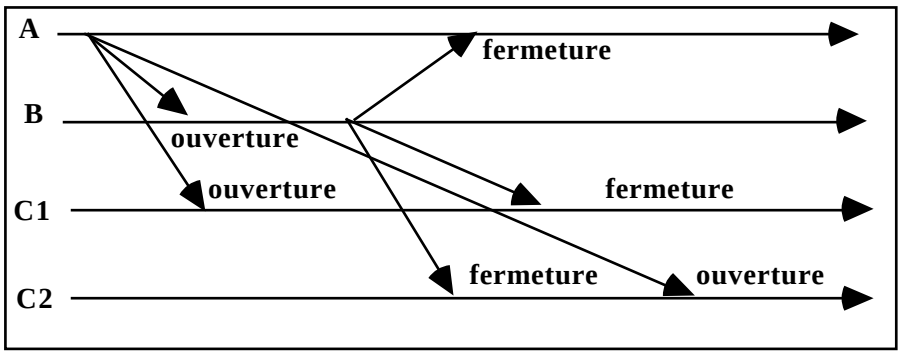
$$\text{émission}_A(m) \rightarrow \text{émission}_B(m1) \Rightarrow \text{réception}_C(m) \rightarrow \text{réception}_C(m1)$$



Exemple 2:

A envoie à C1 et C2 une commande d'ouverture de vanne, en en rendant compte à B. Plus tard B envoie à C1 et C2 l'ordre de fermeture de la vanne, en en rendant compte à A.

Respect de la causalité : le message de A (ouverture) doit être enregistré avant celui de B (fermeture).



DIFFUSION RESPECTANT UN ORDRE TOTAL SIMPLE

Si plusieurs diffusions ont lieu concurremment de différents processus vers le même groupe de diffusion, alors tous les messages sont délivrés aux applications réceptrices dans le même ordre sur tous les récepteurs.

Exemple d'utilisation : copies multiples.

Le fait que toutes les opérations de modifications d'un ensemble de données en copies multiples soient effectuées dans le même ordre sur toutes les copies suffit à assurer le maintien de la cohérence (faible) des copies.

DIFFUSION RESPECTANT L'ORDRE TOTAL CAUSAL

L'ordre total respecte aussi la relation d'ordre causal entre messages