Les systèmes distribués

- Éléments préliminaires
- La synchronisation et les horloges
- Les mécanismes d’automatisation
- Gestion de la concurrence dans le système
- La tolérance et reprise sur panne

Chapitre troisième

Qu’est ce qu’un système distribués

- Un système distribué est un ensemble de ressources matérielles et logicielles qui coopèrent de façon transparente à la réalisation d’un ensemble de tâches utiles
- D’un point de vue applicatif ils peuvent être perçus comme étant des systèmes qui permettent de décomposer des programmes complexes en un sous ensemble indépendants de tâches et qui peuvent donc s’exécuter en parallèle
- Ils ont tout d’abord été créés pour répondre à des besoins de recherche: mettre en réseau un ensemble d’unités de calculs pour coopérer sur la résolution d’un problème mathématique complexes
- Aujourd’hui ils sont omniprésents, on peut les retrouver naturellement dans des datacenters sur internet ou accessoirement les solliciter comme service cloud

Caractéristiques des systèmes distribués

- Transparence
  - L'utilisateur interagit avec le système via une GUI fluide et intuitive ou via une API par un langage de haut niveau
  - Tous les mécanismes de fiabilité, disponibilité, sûreté, synchronisation et de sécurité sont automatisés
- Mise à l'échelle et élasticité
  - Scale-up/ scale-down horizontale et verticale
  - Intégration automatique ou semi-automatique des nouveaux nœuds
  - Topologie dynamique
- Tolérance aux pannes
  - Détection des pannes
  - Reprise sur pannes
- Autonomie et auto-gestion
  - Indépendance totale ou relative des actions administrateur
  - Déploiement automatique des programmes parallélisables
- Interopérabilité
  - Hétérogénéité des composants de traitement
  - Limite de la ressource bande passante

Usages des systèmes distribués

- Les systèmes d'informations complexes
  - Militaires
  - Bancaires
  - Sanitaires
  - Transport aérien
- Les services web et cloud
- Les réseaux paire-à-paire
- La recherche scientifique

Le modèle de communication

- Il existe différents types de systèmes distribués suivant la nature du modèle de communication et des natures d'unités mémoire ainsi que leurs emplacements

- Deux modèles de communications majeurs peuvent être retenu
  - Par mémoire tampon
  - Par échange de messages

- Ajouté à cela, les unités de traitements coopérant possèdes des unités mémoires qui sont situées physiquement à proximité

- Il peut exister ou pas une mémoire centrale accessible par toutes les unités de traitement

Modélisation et évènements

- Un système réparti est représenté par des lignes temporelles parallèles schématisant chacune un site différent et inter-communiquant exclusivement par le billet d'échanges de messages

- Chaque site se comporte comme un automate opérant des actions faisant évoluer son état, ses actions sont appelées des évènements et peuvent êtres soient
  - Locaux : changement de l'état du site par une action interne
  - Distants
    - Émission d'un message
    - Réception d'un message

Asynchronisme

- En prenant en considération les concepts des modèles de communications et du modèle temporel des systèmes réparti il est facile de dégager certaines propriétés des systèmes répartis

- La communication ne se faisant que par échange de message, il devient compliqué d'émettre des hypothèses sur les durées de transmission des messages et par conséquent sur les durées d'exécutions des processus

- Il est facile pour chaque site isolé d'ordonner les évènements suivant l'ordre d'avènement. Réaliser cette même chronologie sur des sites distants interconnectés est compliqué en raison d'absence d'horloge commune, d'hypothèses sur les délais et d'asynchronisme dans les performances de traitement

Hypothèses

- Des mécanismes doivent être implémentés pour éviter la perte de messages et le maintien de l'intégrité de ces derniers sur le réseau organisant le système réparti

- En l'occurrence dans la plupart des cas nous favoriserons FIFO comme politique de délivrance de messages et ne nous intéresserons plus aux phénomènes intermédiaires, nous nous concentrerons sur les évènement sur sites et qui pour la partie réseau ne porte que sur l'émission et la réception d'un message

- Le principe de causalité dans les évènements locaux est employé et nous tentons de mettre en place des approches permettant de l'élargir aux sites subissant des évènements de nature distante

L’ordre causal selon Lamport

- L’ordre causal pour des évènements se produisant sur le même site correspond à l’ordonnancement temporel des ces derniers, en d’autres termes l’ordre causal des évènements est leurs ordres temporels

- En vue d’étendre la précédence causale à tout le système il faut trouver une manière de définir un ordre à la fois global et cohérent avec les informations locales propres à chaque site

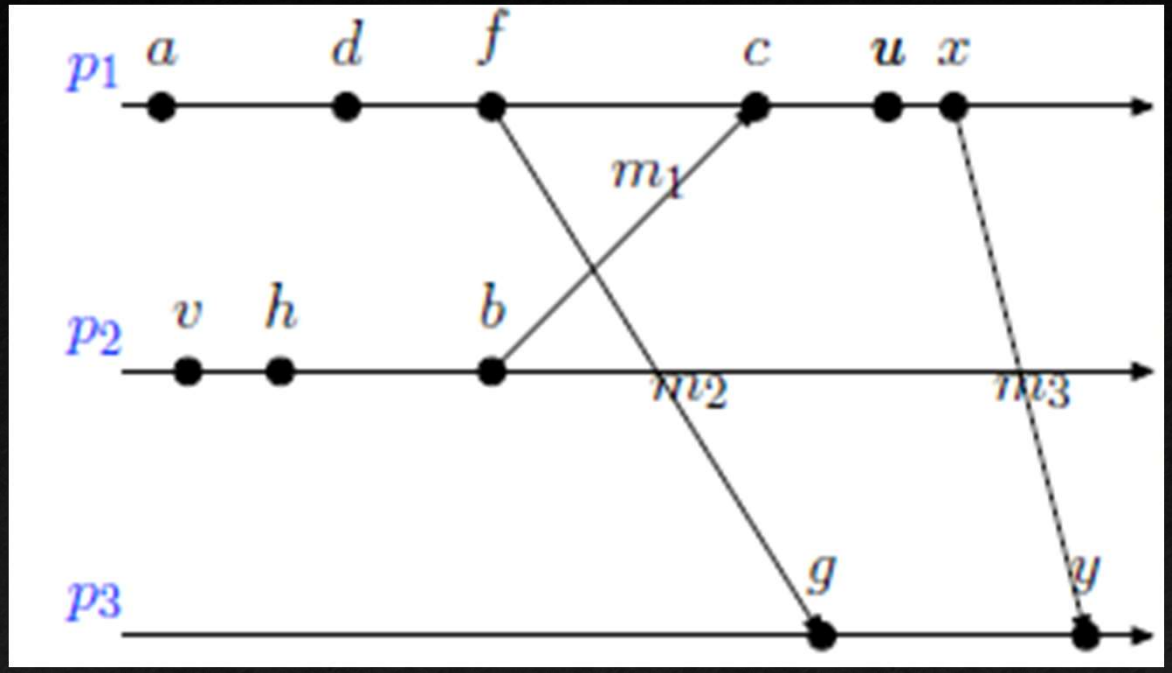
- Lamport propose une définition d’ordre partiel qui stipule que un évènement a précède un évènement b (a->b) ssi
  - a et b sont produits par le même site et a précède chronologiquement b
  - a est un envoi de message et b en est la réception
  - Transitivité: il existe un évènement intermédiaire c tel que
    - a->c et c->b

Le zoo évènementiel d'un évènement (e)

- Les évènement précédents (prédécesseurs ou antérieurs) à un évènement e sont tous les évènements qui peuvent par les trois règles précédentes êtres définis comme causalement déterminants de e

- Les évènement suivants (successeurs ou postérieurs) à un évènement e sont tous les évènements qui peuvent par les trois règles précédentes êtres définis comme causalement déterminés par e

- Les évènement concurrents (parallèles ou indépendants) à un évènement e sont tous les évènements qui ne peuvent en aucun cas par les trois règles précédentes êtres définis comme causalement déterminés par e ou déterminants de e



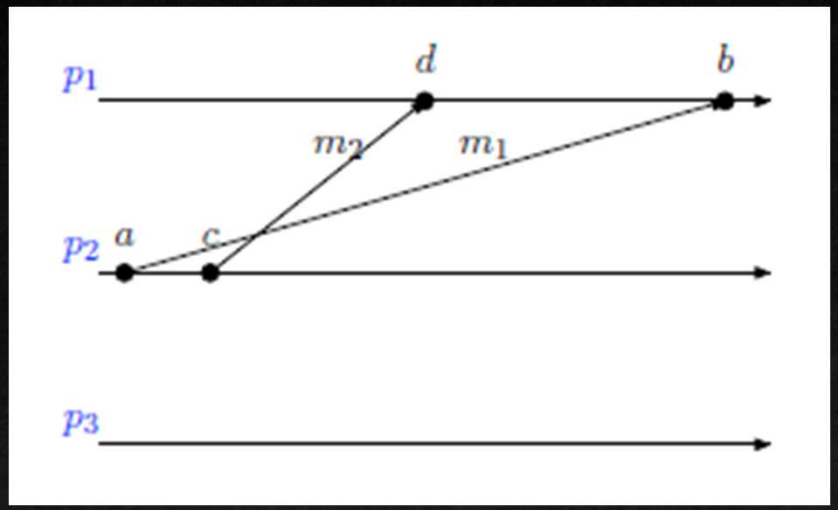
Exemple

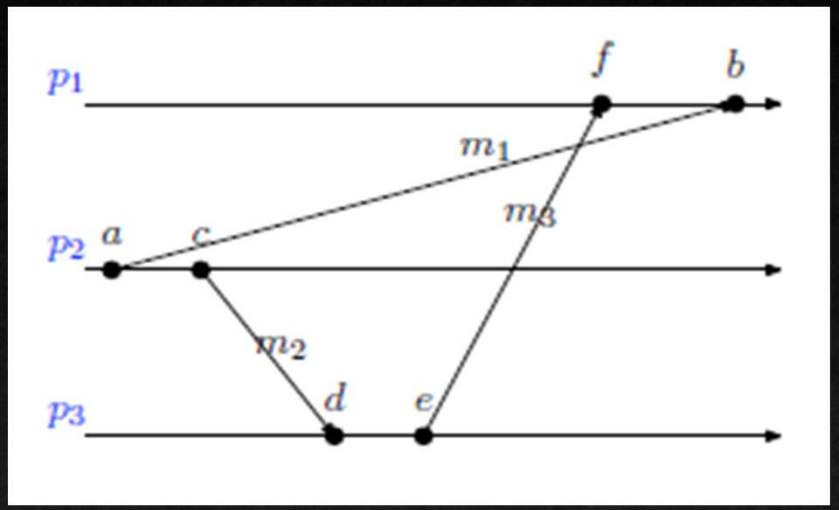
Les modes de délivrances de messages

- Dans le mode FIFO, tous les messages en destination d'un site j émanant d'un site i sont délivrés sur j dans le même ordre d'émission depuis le site i

- Dans le mode de délivrance causale, en assumant que deux messages en destination du site j émanent respectivement des sites i et k, en assumant que l'émission du message de i précède causalement l'émission du message de k alors la délivrance sur le site j se fera en respectant ce même ordre causal

- Pour avoir un ordre total il faudrait avoir une horloge qui permette d'avoir pour chaque paire d'évènements distincts deux estampilles temporelles différentes et l'ordre de délivrance global doit être FIFO et causal



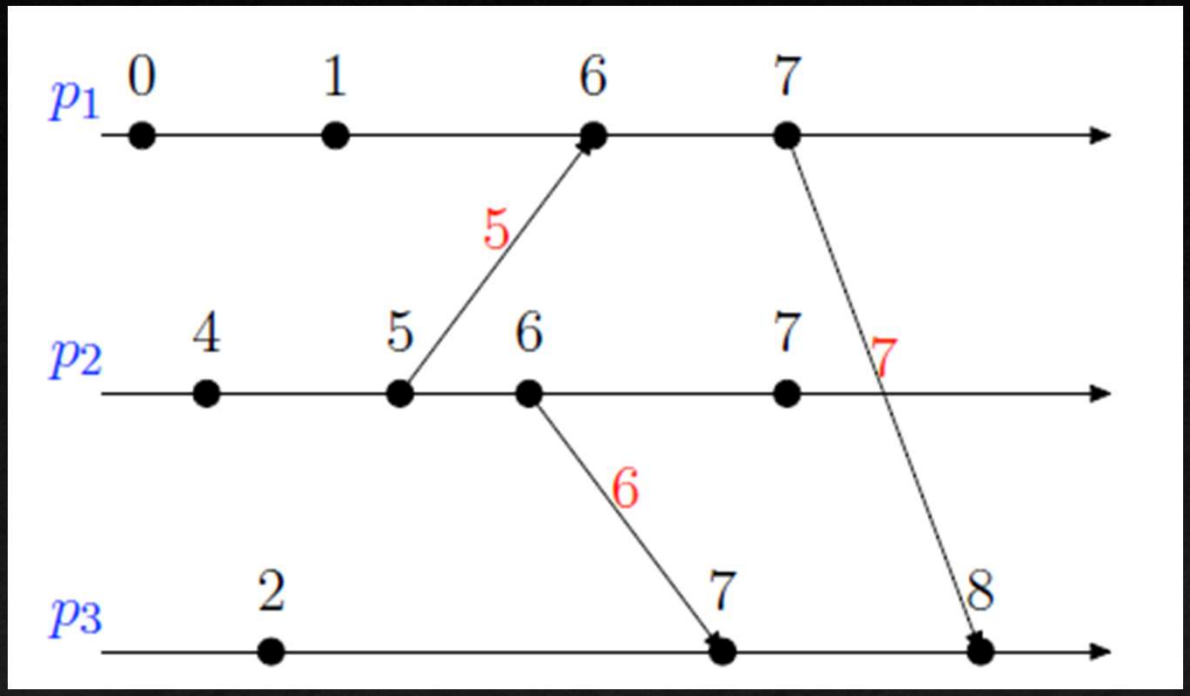


Exemples

Les horloges de Lamport

- Une horloge de lamport est la donnée d'un ensemble de singletons de cardinalité égale aux nombre de sites, chaque valeur est localement augmentée quand un évènement se produit sur le site en question

- Les règles de fonctionnement de cette horloge sont très simple à énoncer
  - Si un évènement est local alors
    - HL(i) = HL(i)+1
  - Si e est l'envoi d'un message depuis d'un site i
    - HL(i) = HL(i)+1
    - EL(M) = HL(i)
  - Si e est la réception d'un message sur un site j alors
    - HL(j) = HL(j)+1



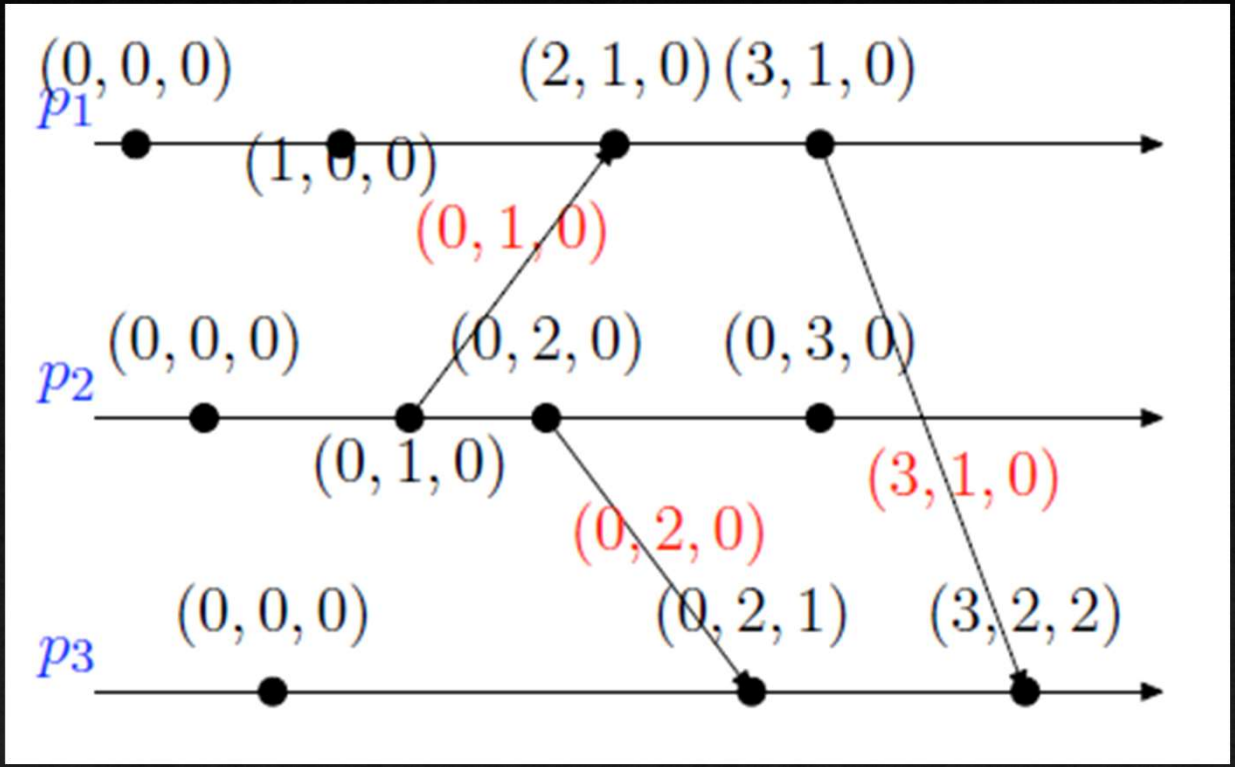
Exemple

Les horloges vectorielles

- Les horloges de Lamport souffrent d'insuffisances telles que le manque d'ordre global et le nom respect des ordres de délivrance FIFO et causale

- Un autre type d'horloges dites vectorielles répondent mieux à la question. Dans ce cas de figure l'horloge est réalisée par un ensemble de vecteurs dont les cardinalités respectives sont égales aux nombres de sites

- Les règles à observer pour maintenir ce type d'horloge sont
  - Si l'évènement est locale à i: HV(i)[i]<- HV(i)[i]
  - Si l'évènement est un envoi de message depuis le site i alors
    - HV(i)[i] <- HV(i)[i] + 1
    - EV(M)[i] <- HV(i)[i]
  - Si l'évènement est la réception d'un message sur le site j alors
    - HV(j)[j] <- HV(j)[j] + 1
    - HV(j)[i] <- MAX(HV(j)[i], EV(m))



Exemple

Les horloges matricielles

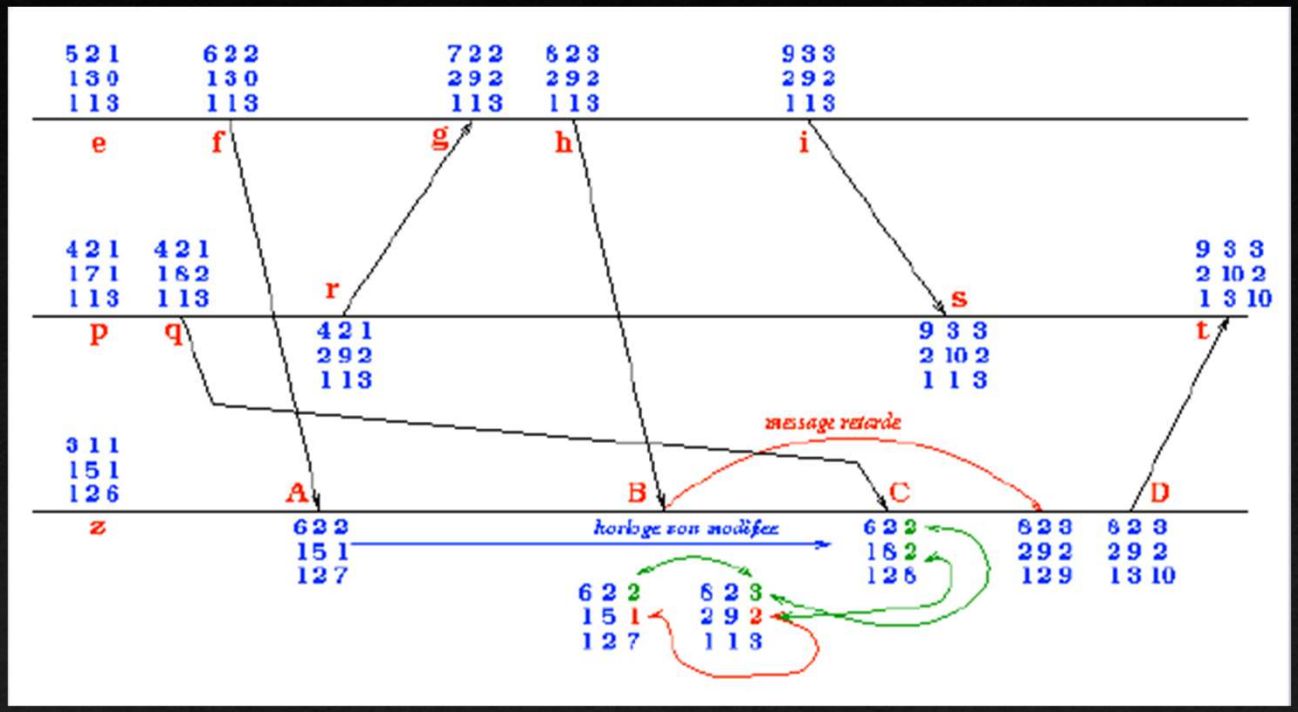
- Les horloges matricielles sont efficaces à la fois pour un ordre global strict et pour le respect des deux modes de délivrance de messages fifo et causale

- La réalisation d'une horloge matricielle passe par l'utilisation d'un ensemble de matrices carrés dont les cardinalités respectives sont le nombre de sites et le carré du nombre de sites

- Il faut observer les règles suivantes pour implémenter une horloge matricielle
  - Évènement local à i: HM[i,i] <- HM[i,i] + 1
  - Emission d'un message vers un site i
    - HM[i,i] <- HM[i,i] + 1
    - HM[i,j] <- HM[i,j] + 1
    - EM(m) <- HM

Les horloges matricielles

- La délivrance de message est un peu plus compliqué, quand un site j reçoit un message depuis un site i il doit
  - Vérifier la réception de tous les messages antérieurs
    - Ordre fifo: EM[j,i] = HM[j,i]
    - Ordre causal: EM[k,i] = HM[k,i], quelque soit k site différent de j
  - Mise à jour de HM au niveau de i
    - HM[i,i] = HM[i,i] + 1
    - HM[i,j] = HM[i,j] + 1
    - Et pour tout k et l différents de i
      - HM[k,l]= max(HM[k,L], EM[k,l]


5 2 1
1 3 0
1 1 3
e
6 2 2
1 3 0
1 1 3
f
7 2 2
2 9 2
1 1 3
g
6 2 3
2 9 2
1 1 3
h
9 3 3
2 9 2
1 1 3
i
4 2 1
1 7 1
1 1 3
p
4 2 1
1 8 2
1 1 3
q
r
4 2 1
2 9 2
1 1 3
9 3 3
2 10 2
1 1 3
s
9 3 3
2 10 2
1 3 10
t
message retardé
3 1 1
1 5 1
1 2 6
z
A
6 2 3
1 5 1
1 2 7
B
horloge non modifiée
C
6 2 2
1 8 2
1 2 6
6 2 3
2 9 2
1 2 9
D
6 2 3
2 9 2
1 3 10
6 2 2
1 5 1
1 2 7
6 2 3
2 9 2
1 1 3

Exemple

Election de leader

- La sélection d'un leader est le fait de désigner un nœud dans le système distribués afin d'effectuer des tâches spéciales relatives à la gestion des autres nœuds coopérant à la réalisation de tâches utiles

- La sélection d'un leader permet entre autre de faciliter l'administration et la supervision du réseau de nœuds coopérant
  - Les administrateurs n'ont plus qu'à superviser les leaders
  - Les leaders se chargent des différentes opérations de distributions des tâches et répartition de la charge sur les autres nœuds

- Il convient tout de même de prendre des dispositions car s'appuyer sur des leaders revient à
  - Créer des points réalisant le paradigme du one-point-failure
  - La non intégrité ou conformité du leader entraine un mauvais fonctionnement d'une grande faction du système

- Suivant le contexte et le cas d'utilisation il existe de nombreuses techniques permettant de réaliser l'élection d'un leader

Les sections critiques

- L'accès concurrentiel à une ressource sur le réseau peut s'effectuer suivant un modèle d'exclusion mutuelle

- Une exclusion mutuelle peut être gérée par un échange de messages entre les sites concurrents afin de se synchroniser et de se relayer l'accès

- N'importe quel algorithme pour la gestion des accès à la section critique doit pouvoir assurer les tâches suivantes
  - Le site désirant accéder à la section transmet une requête estampillée
  - Tous les sites ne désirant pas l'accès ou requérant des accès à dates postérieures à son estampille lui répondent par leurs accords
  - Ceux déjà en concurrence ou en exploitation attendront la libération avant de donner leurs OK
  - Dès qu'il a reçu l'accord de tous les concurrents il accède à la ressource et les informe à la libération

Site 1		Site 2		Site 3	
(Accès,14)					
	(D,0ok)	R(Accès,14,1)		R(Accès,14,1)	
		← (*ok*, 1)		← (*ok*, 1)	
	(D,2 ok)				
Ressource		(Accès,21)	(D,0ok)		
R.+ R(Accès,21,2)	(D,21,2)			(Accès,20)	(D,0ok)
R.+ R(Accès,20,3)	(D,21,2),(D,20,3)	R(Accès,20,3)	(D,0oK)		(D,0ok),
Ressource	(D,21,2),(D,20,3)	(*ok*, 3) →	(D,0oK)	R(Accès,21,2)	(D,0ok),(D,21,2)
liberation	(D,21,2),(D,20,3)		(D,0ok)		(D,1ok),(D,21,2)
(*ok*, 2), (*ok*, 3) →			(D,0ok)		(D,1ok),(D,21,2)
(*ok*, 2), (*ok*, 3) →			(D,0ok)		(D,2ok),(D,21,2)
			(D,1ok)	Ressource	(D,21,2)
			(D,1ok)	libération	(D,21,2)
			(D,1ok)	← (*ok*, 2)	
			(D,2ok)		
			ressource		

Exemple

La reprise sur pannes

- La tolérance aux pannes est la capacité d'un système de reprendre suite à une panne impromptue

- Pour pouvoir assurer la continuité d'un système après une panne il faut s'appuyer sur des méthodes préventives, celles ci permettent de réaliser des sauvegardes de points de reprises à des états cohérents successifs du système, on parle de coupure

- Une coupure est dite cohérente si elle vérifie
  - a->b et b dans C alors a est dans C également
  - a est une réception d'un message appartenant à C alors c son émission lui appartient aussi

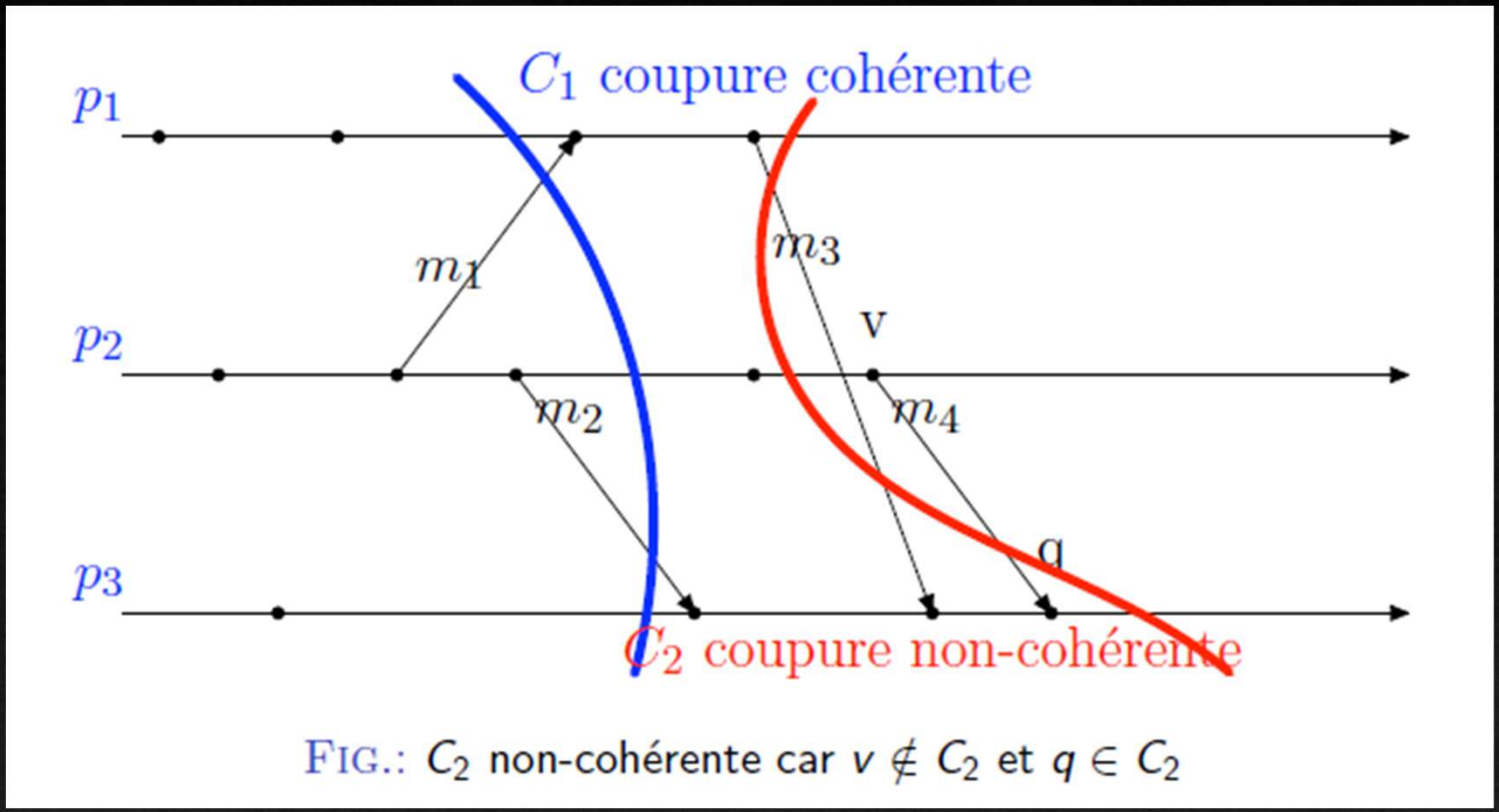


FIG.: C_2 non-cohérente car $v \notin C_2$ et $q \in C_2$

Exemple

En bref

- Un système distribué est un ensemble de nœud coopérant ou bien interopérant à la réalisation de tâches utiles
- Il est transparent, élastique, tolérant aux pannes et autonome
- Les nœuds communiquent par mémoire cache ou par échange de messages, dans le dernier cas il faut mettre en place un moyen d'assurer le délivrance FIFO et causale des messages
- Les horloges de Lamport, vectorielles et surtout matricielles répondent aux besoins d'ordre global strict et d'ordre de délivrance
- L'élection de leader permet soit la désignation par les administrateurs ou par le réseau de façon autonome d'un sous ensemble de nœuds en charge de la coordination du travail et la distribution des tâches
- L'accès concurrentiel aux section critique peut être géré par des approches d'exclusion mutuelle réalisée via l'échange de message
- La reprise sur panne est la capacité du système à reprendre suite à une panne et de maintenir des points de reprises cohérents