POSE DE POINTS DE REPRISE RÉPARTIS

*** tolérance aux pannes et reprise arrière (ex. longs calculs scientifiques)**

*** validation en transactionnel (ex. cohérence et prévention d'interblocage)**

• DEUX OPÉRATIONS A CONSIDÉRER

sauvegarde des information nécessaires à la reprise (points et messages)

retour arrière à un état global cohérent et ré-exécution de l'application

• TECHNIQUES DE POSE DES POINTS DE REPRISE

sauvegarde (pose) périodique indépendamment sur chaque site

effet domino

sauvegarde cohérente déclenchée périodiquement par un site initiateur
(exemple : voir l'algorithme de Chandy, Lamport)

synchronisation explicite
coûteux en message
initiateur quelconque (tolérance aux pannes)

sauvegarde adaptative de Xu et Netzer (1993)

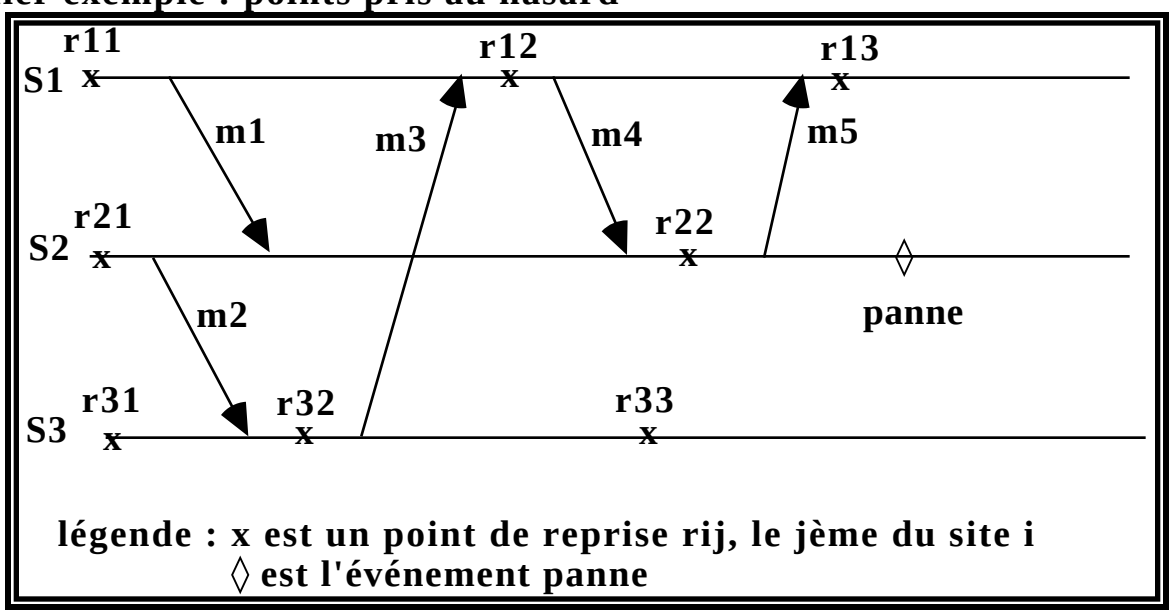
synchronisation implicite entre sites
points de contrôle forcés sur un site récepteur

Référence : [XU 93] J. XU, R. NETZER, Adaptative Independant Checkpointing for Reducing Rollback Propagation, 5th IEEE Symp. on Parallel and Distributed Processing, Dec. 93, Dallas TX, USA, 8 pages, 1993.

SAUVEGARDE INDÉPENDANTE SUR CHAQUE SITE

• effet domino : le retour arrière d'un site entraîne le retour des autres au delà du dernier point de reprise et cela parfois jusqu'au début du traitement

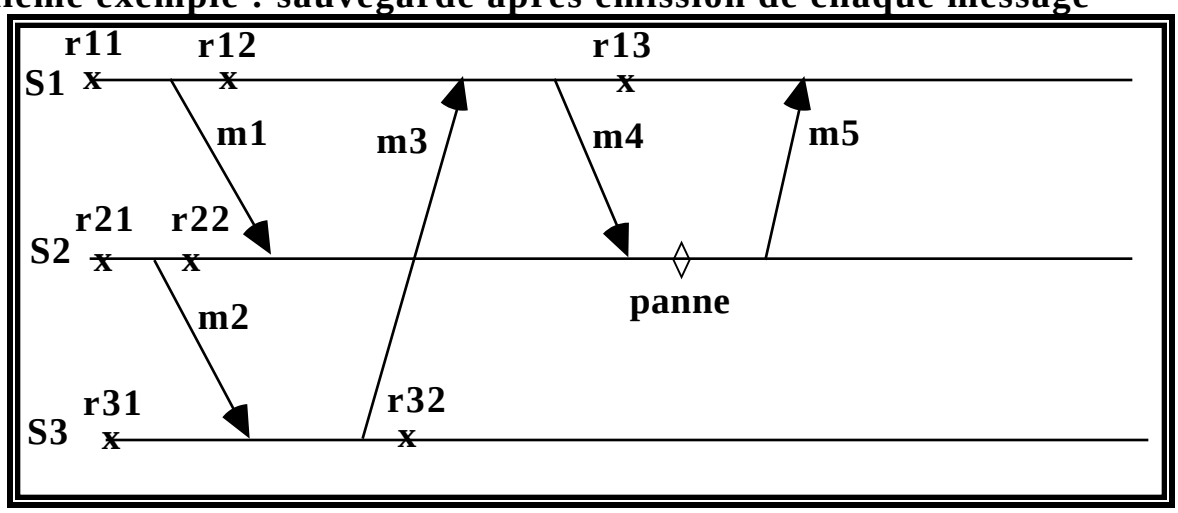
Premier exemple : points pris au hasard



Pour retrouver un point cohérent et repartir après réparation (ou avec un processeur en redondance passive sur S2), S2 remonte à r22, mais comme l'émission de m5 est perdue, il faut faire remonter S1 à r12 ; puis comme l'émission de m4 est perdue, il faut faire remonter S2 à r21 ; etc.

Aucune coupure r1a, r2b, r3c n'est cohérente sauf la coupure initiale r11, r21, r31.

Deuxième exemple : sauvegarde après émission de chaque message



On peut repartir avec r13, r22, r32 si on a noté l'émission de m1 et m4

MODÉLISATION

• On note -> la dépendance causale
• On note $r_{p,j}$ le jème point de reprise sur le site S_p;
Chaque Sp pose un point de reprise initial r_{p1} au début de l'exécution.
Une ligne de reprise est un ensemble de points de reprise, un par site.
Une ligne de reprise R est cohérente si elle forme un état global cohérent : tout message enregistré reçu sur un site a été enregistré émis sur un autre.
enregistré<=> fait partie du passé, de la coupure
• L'intervalle de reprise $I_{p,i}$ est la suite d'événements entre $r_{p,i}$ et $r_{p,i+1}$ (il contient $r_{p,i}$ mais pas $r_{p,i+1}$)

CHEMINS EN ZIGZAG

• Il y a un chemin en zigzag de $r_{p,i}$ à $r_{q,j}$ si et seulement si il existe des messages $m_1, m_2, \ldots, m_n$ ($n \geq 1$) tels que

(1) m_1 est envoyé par le site S_p après $r_{p,i}$
(2) si m_k ($1 \leq k < n$) est reçu par le site S_r, alors m_{k+1} est envoyé par S_r dans le même intervalle de reprise ou dans un intervalle postérieur (noter que m_{k+1} peut être envoyé avant ou après l'arrivée de m_k),
(3) m_n est reçu par le site S_q avant $r_{q,j}$

CYCLE EN ZIGZAG

Le point de reprise r appartient à un cycle en zigzag si et seulement si il y a un chemin en zigzag de r à lui-même.

CHEMIN CAUSAL ENTRE POINTS DE REPRISE

si $r_{p,i}$ -> $r_{q,j}$, alors on a un chemin causal

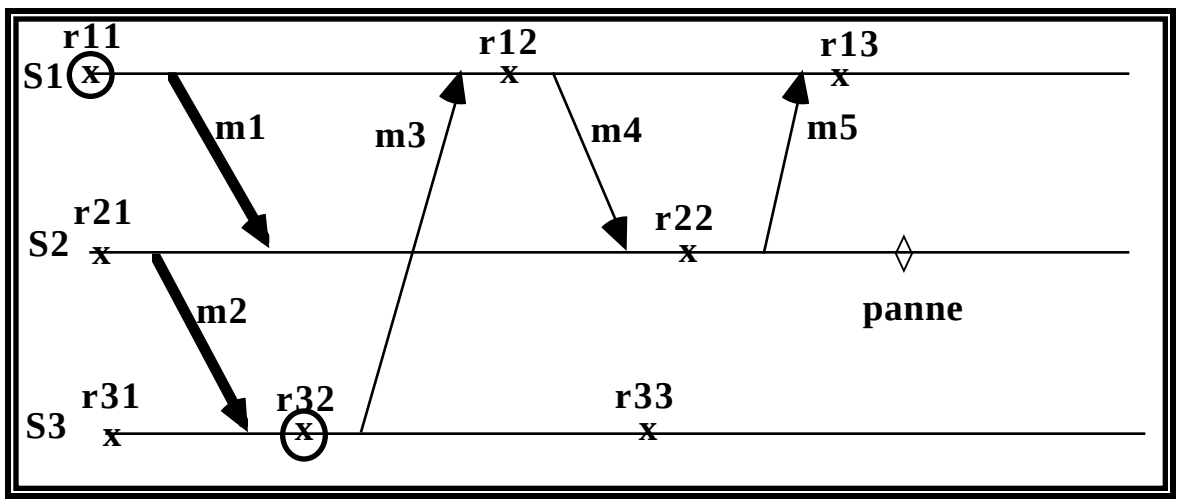
chemin causal => chemin en zigzag
chemin en zigzag ≠> chemin causal

THÉORÈME

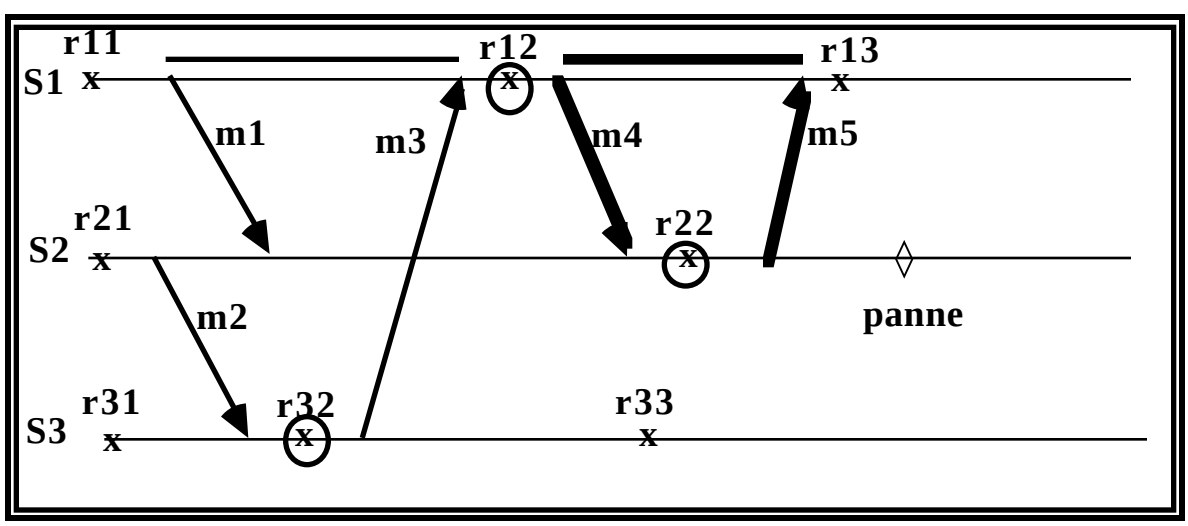
Un ensemble R de points de reprise sur différents sites peut être étendu faire partie d'une ligne de reprise cohérente si et seulement si aucun point de reprise de R n'appartient à un chemin en zigzag vers un autre point de reprise de R (ou vers lui-même, formant un cycle en zigzag).

COROLLAIRE : S'il n'y a pas de chemin en zigzag (ni cycle) dans R on peut toujours construire une ligne de reprise cohérente incluant R

EXEMPLES DE CHEMINS ET CYCLES EN ZIGZAG



chemin en zigzag de r11 à r32, chemin non causal



cycle en zigzag autour de r22 (messages m5 et m4)
cycle en zigzag autour de r32 (messages m3, m1 et m2)
cycle en zigzag autour de r12 (messages m4, m2 et m3)

PRINCIPE DE LA SAUVEGARDE ADAPTATIVE

POINTS DE REPRISE UTILISABLE

• Un point de reprise $r_{p,i}$ est utilisable s'il appartient à une ligne de reprise cohérente, c'est à dire s'il appartient à un état global cohérent.

COROLLAIRE

Un point de reprise $r_{p,i}$ est utilisable s'il n'y a pas de cycle en zigzag qui le contienne.

PRINCIPE DE LA SAUVEGARDE ADAPTATIVE

Empêcher les cycles en zigzag de se former et pour cela :

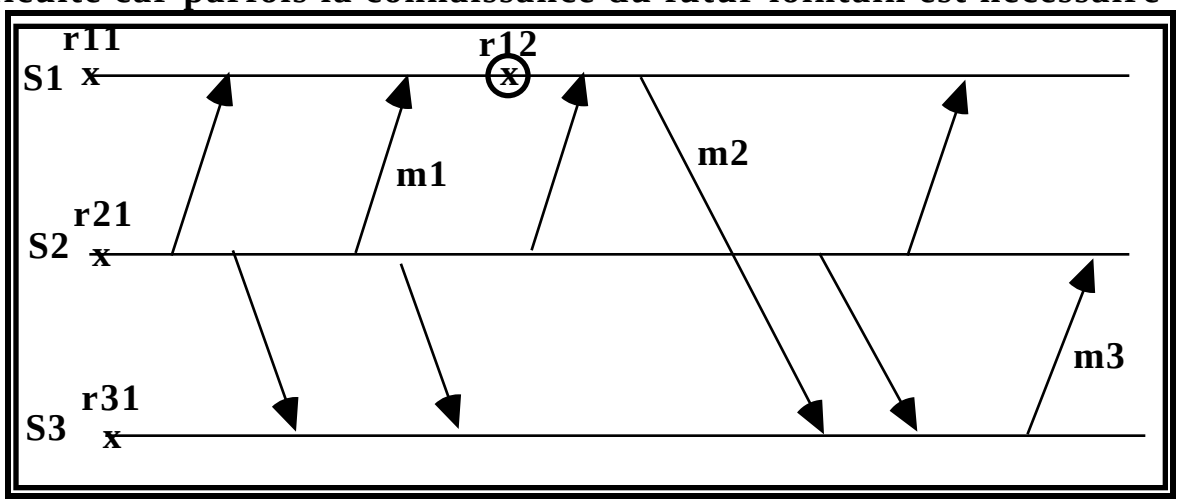
• détecter si un message reçu sur un site achève de construire un cycle,
• si c'est le cas, forcer un point de reprise avant la réception du message.

MÉTHODE APPROCHÉE

En fait on ne peut pas détecter si le cycle est en zigzag. Alors en approximation on détecte si le chemin est causal.
(à l'expérience, il y a très peu de chemins en zigzag non causaux)

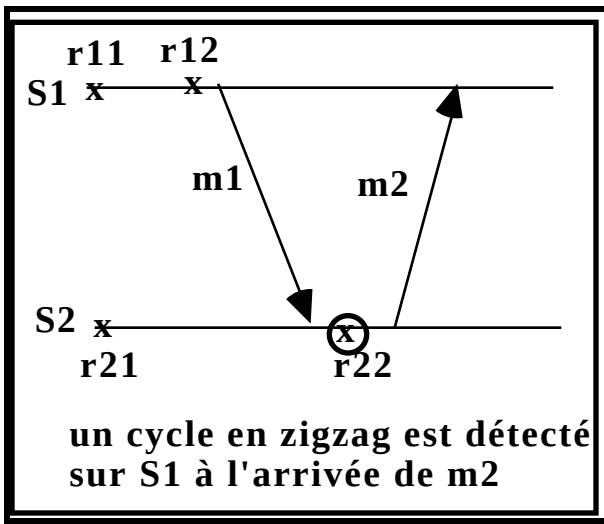
DÉTECTION DES CYCLES EN ZIGZAG

• **difficulté car parfois la connaissance du futur lointain est nécessaire**

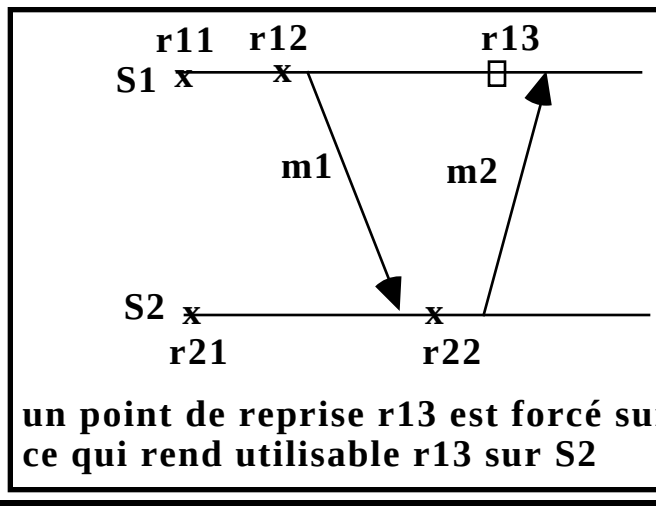


Le cycle en zigzag autour de r12 apparaît bien après r12 quand m3 arrive

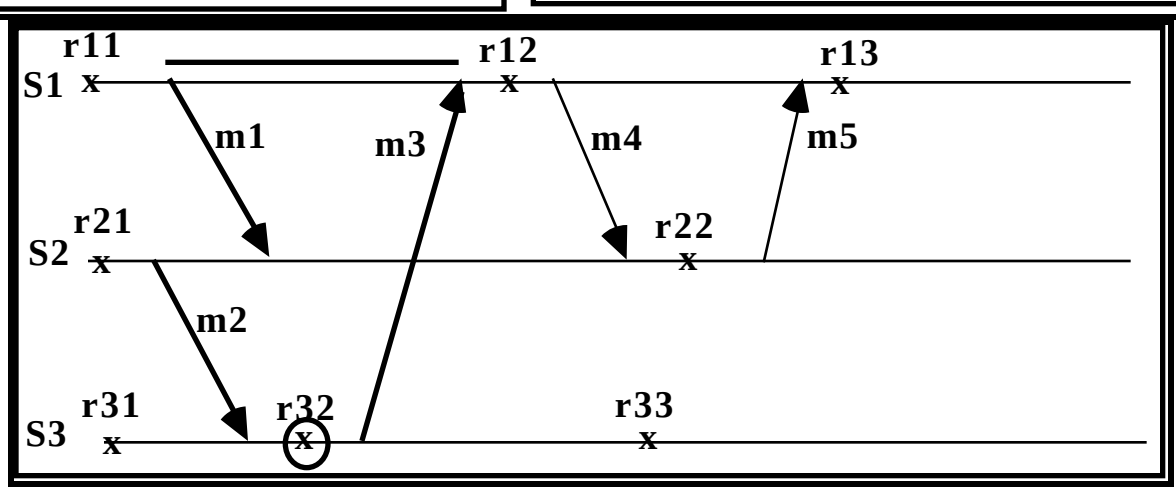
APPROXIMATION : DÉTECTER LES CHEMINS CAUSAUX



un cycle en zigzag est détecté sur S1 à l'arrivée de m2



un point de reprise r13 est forcé sur S1 ce qui rend utilisable r13 sur S2



r32 : cycle en zigzag non détecté (il n'est pas causal)

ALGORITHME DE SAUVEGARDE ADAPTATIVE

• Détecter les dépendances causales :

• On utilise les horloges vectorielles pour dater les points de reprise

On associe une horloge vectorielle V_i à chaque site Si

• Initialement Vi = (0,0,...,0,1,0,...,0) nul partout sauf la ième composante

• A chaque point de reprise nouveau local à Si, on fait $V_i[i] := V_i[i] + 1$

• Chaque message m porte une estampille Vm (Vm = Vi de l'émetteur)

• A la réception de (m, Vm) par un site Si, on enrichit l'historique connu par Si avec l'historique transporté par m :

$$V_i[j] := \max(V_i[j], Vm[j]) \text{ pour tous } j = 1,...n,\ j \neq i$$

• $V_i[j]$ indique le passé de Si situé sur Sj et tel qu'il est connu par Si

indique donc aussi le numéro du dernier point de reprise sur Sj qui est en dépendance causale avec le point de reprise courant sur Si.

• A la création d'un nouveau point de reprise $r_{i,k}$ sur Si, le site Si sauvegarde dans RVi les dépendances causales de $r_{i,k}$. Soit RVi = Vi($r_{i,k}$).

• A l'envoi d'un message m de Si vers Sj, Si surcharge le message m avec Zid = RVi[j], c'est à dire avec le numéro du dernier point de reprise sur Sj qui est sur un chemin causal avec $r_{i,k}$

• A la réception d'un message (m, Vm, Zid) par Si et avant de traiter le message, l'algorithme regarde s'il existe un cycle causal entre le point de reprise courant sur Si, daté $RV_i[i]$ et le point de reprise sur Sj précédant l'émission. Il y a un cycle causal point de reprise sur Sj précédant l'émission si $RV_i[i] = Zid$.

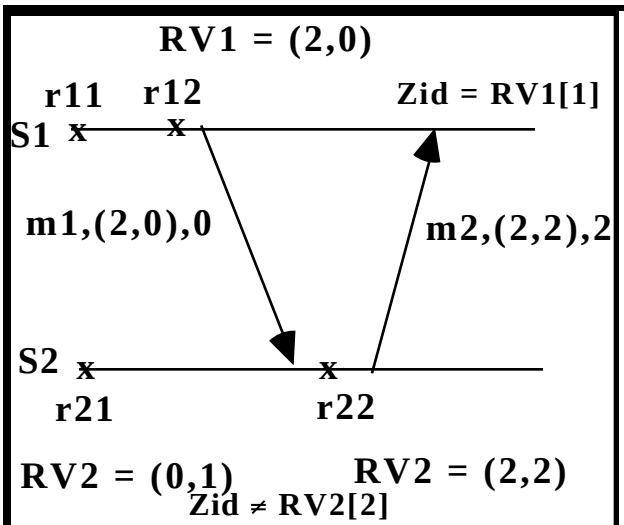
Puis Vi est mis à jour.

ALGORITHME :

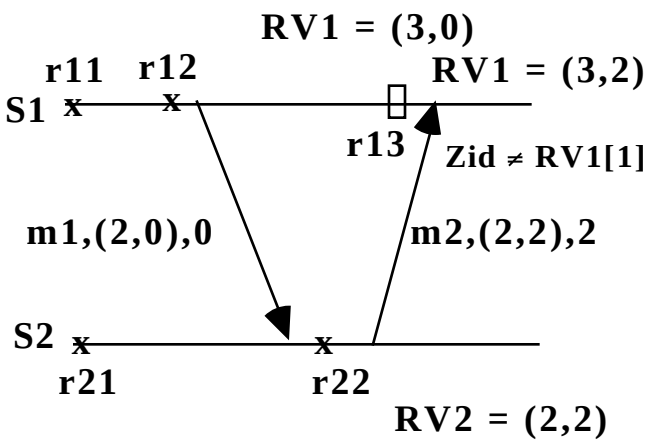
1 : si $Zid = RV_i[i]$ alors forcer un nouveau point de reprise;
$V_i[i] := V_i[i] + 1$;
$RV_i = V_i$
finsi;

2 : $V_i[j] := \max(V_i[j], Vm[j])$ pour tous $j = 1,...n,\ j \neq i$; -- historique

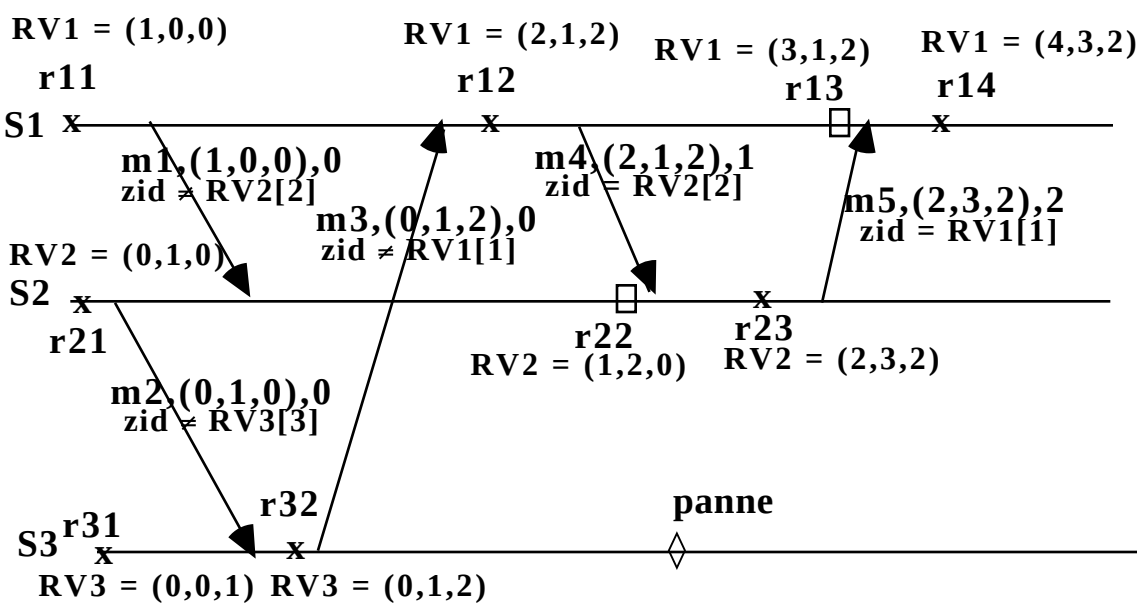
EXEMPLES DE SAUVEGARDE ADAPTATIVE



un cycle en zigzag pour r22 est détecté sur S1 à l'arrivée de m2



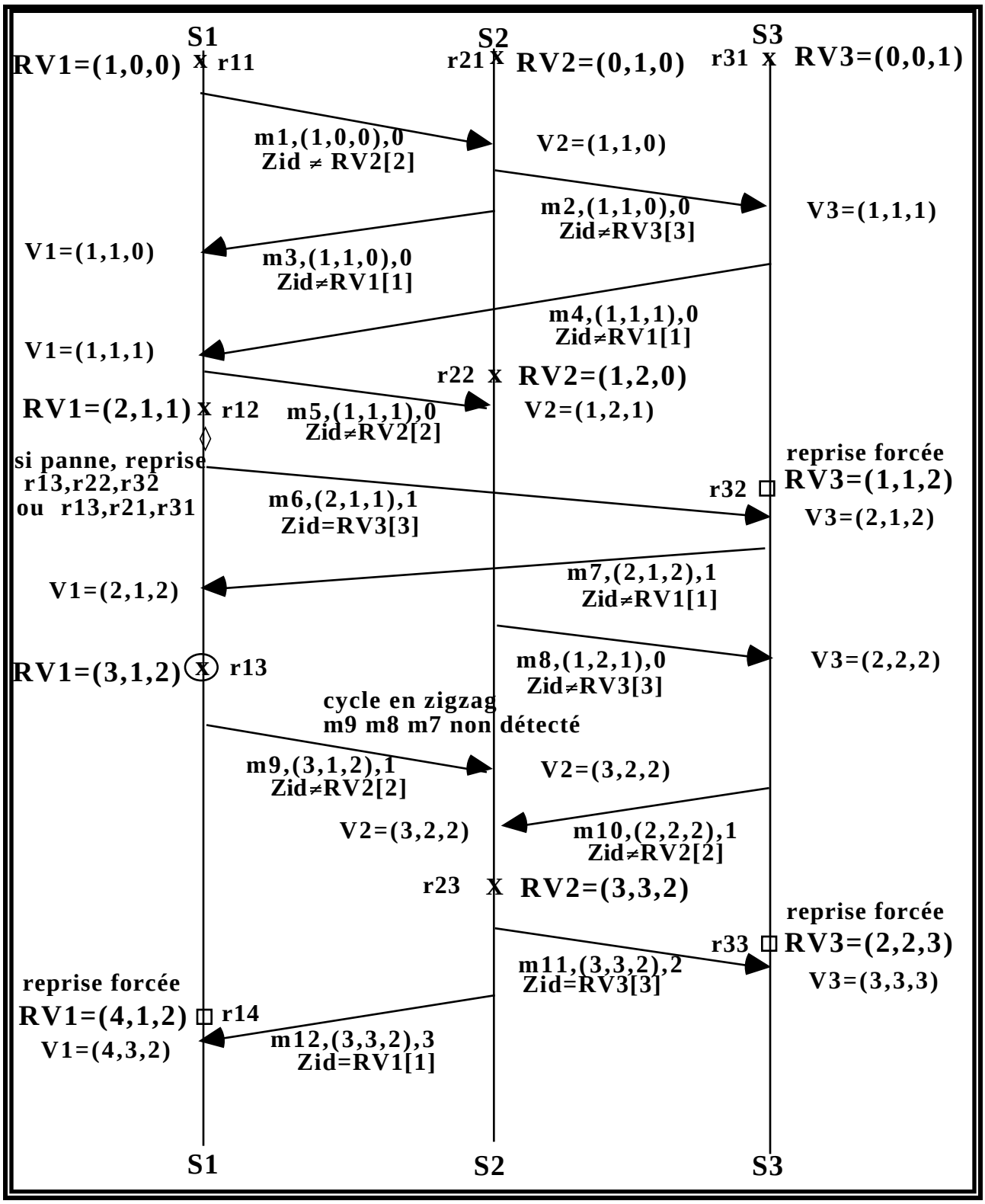
un point de reprise r13 est forcé sur S1 ce qui rend utilisable r22 sur S2



le cycle en zigzag autour de r32 n'est pas détecté, il n'est pas causal, la reprise arrière remontera à r31

un cycle autour de r12 est détecté sur S2, à l'arrivée de m4,1 on force la prise de r22

EXEMPLE DE SAUVEGARDE ADAPTATIVE



trois reprises sont forcées

sept reprises spontanées

CONCLUSION SUR LA SAUVEGARDE ADAPTATIVE

RÉSULTATS EXPÉRIMENTAUX :

• Conditions d'expérimentation

6 programmes de tests (de 13 000 à 370 000 messages échangés au total)

Hypercube Intel iPSC/860 avec 16 noeuds utilisés pour les tests

• Réduction de la propagation arrière pour atteindre une ligne cohérente

Avec une sauvegarde périodique déclenchée indépendamment sur chaque site, il faut en moyenne faire un retour arrière de 3 à 4 points de reprise par site pour obtenir une ligne de reprise cohérente.

En ajoutant la sauvegarde adaptative, on réduit ce retour arrière à un peu moins d'un point de reprise par site.

• Surcoût pour la sauvegarde adaptative

• faible en gestion de l'horloge vectorielle et de tests sur les messages

• nombre de points de reprise additionnels : pose 4% de points de reprise de plus que la sauvegarde périodique indépendante

BILAN

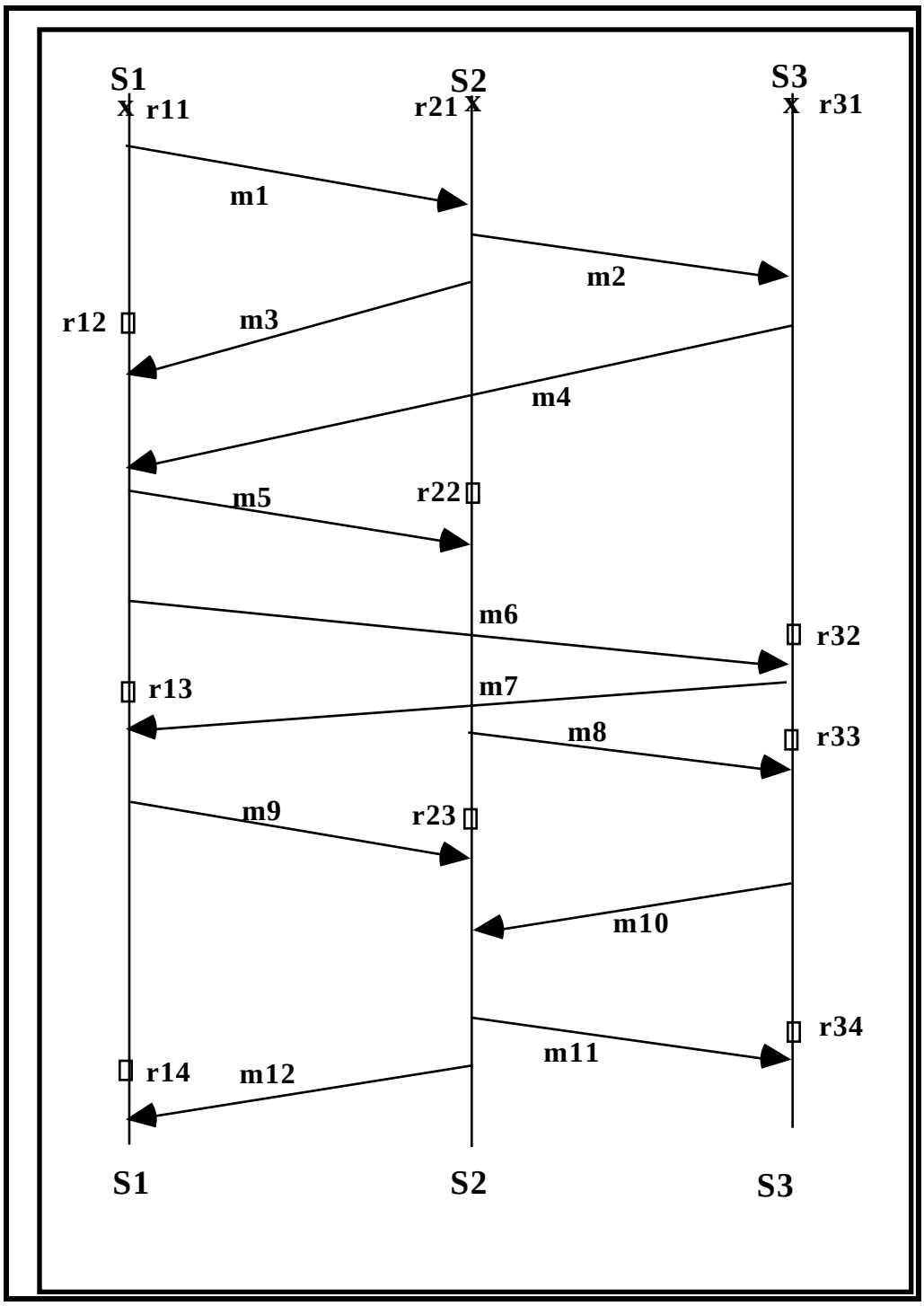
• Condition nécessaire et suffisante pour qu'un ensemble de points de reprise soit cohérent : absence de chemins en zigzag entre ces points

• Méthode heuristique : on recherche les chemins causaux (leur absence est une condition nécessaire) au lieu de rechercher les chemins en zigzag

• Conclusion : gain obtenu par la réduction de l'effet domino

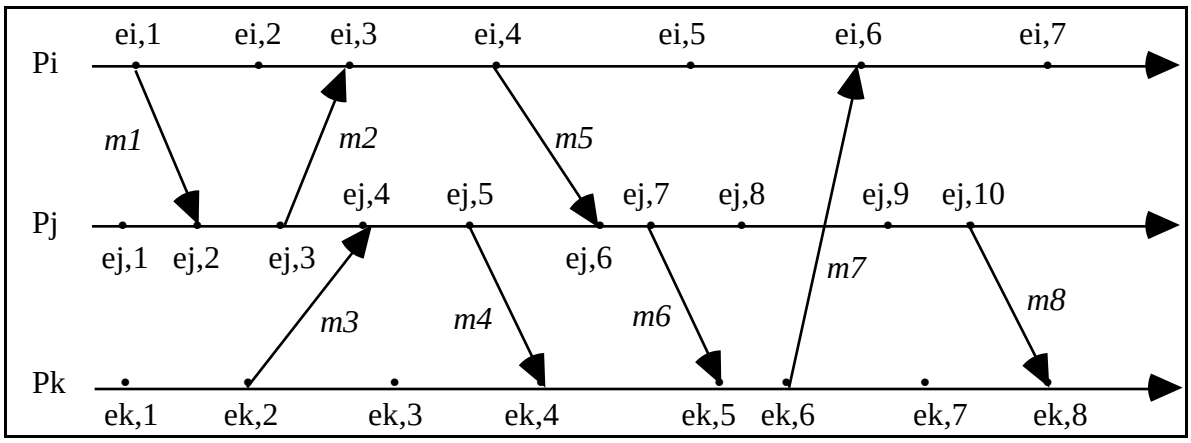
• ne pas oublier qu'il faut en plus faire le retour arrière, avoir sauvegardé les messages qui franchissent la coupure et réexécuter l'application.

RETOUR SUR L'EXEMPLE AVEC UNE AUTRE MÉTHODE ((RECEPTION)*(ÉMISSON)*SAUVEGARDE) (heuristique de Russell 1980)



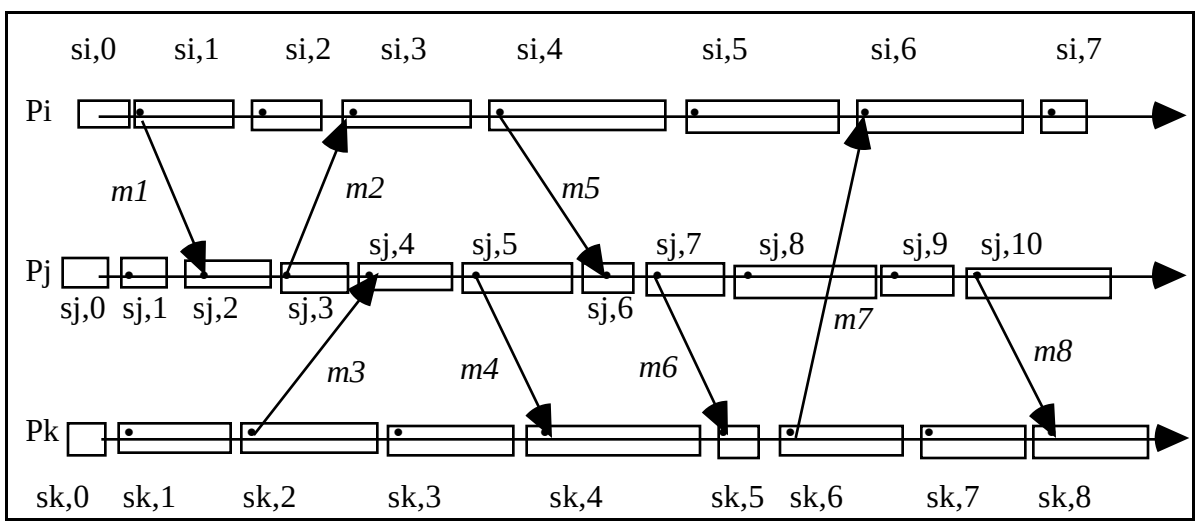
toutes les onze sauvegardes sont forcées

AUTRES MODÉLISATIONS D'UNE ÉXÉCUTION RÉPARTIE

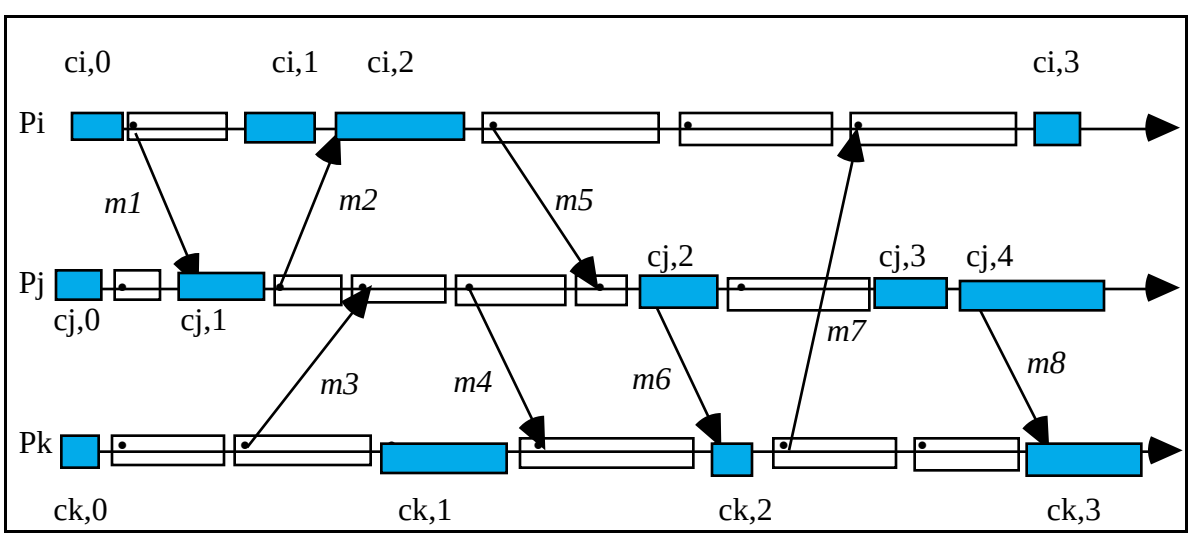


1. Exemple d'exécution répartie modélisée par des événements

événement = exécution d'instruction



2. Même exécution répartie modélisée par des états locaux



3. Même exécution répartie modélisée par des points de contrôle