Sumário

- Recuperação em sistemas distribuídos.
- Salvaguardas e reposição.
- Salvaguarda coordenada.
- Salvaguarda não-coordenada.

Ideia chave

- Salvaguardar o estado de uma computação num meio fiável.

  Memória não volátil (disco).

  Memória estável (disco duplicado, etc).

  Outra máquina (se só falhar uma).

- Em caso de falha (por paragem), retomar a execução a partir da última salvaguarda.

Problemas

- Quando guardar.

  Salvaguarda coordenada e não coordenada.

- O que guardar.

  Qual o estado do processo?

- Relação com outros componentes.

Quando guardar

- É necessário saber:

  Qual a probabilidade de ocorrerem faltas.

  Quanto tempo demora a fazer uma salvaguarda.

- Tenta-se conciliar o tempo de recuperação com a penalização no desempenho imposta pela salvaguarda.

Quem decide

- O programador, coloca explicitamente pontos de salvaguarda no código.
- O compilador realiza essa tarefa automáticamente.
- O sistema operativo/ ambiente de suporte decide quando fazer salvaguardas.

O que guardar

- Salvaguarda transparente.

  Guarda todo o estado do processo.

  Significativamente maior que o estado útil (exemplo: 10M para 33K úteis).

- Salvaguarda assistida.

  É o programador que decide qual o estado a guardar em cada salvaguarda.

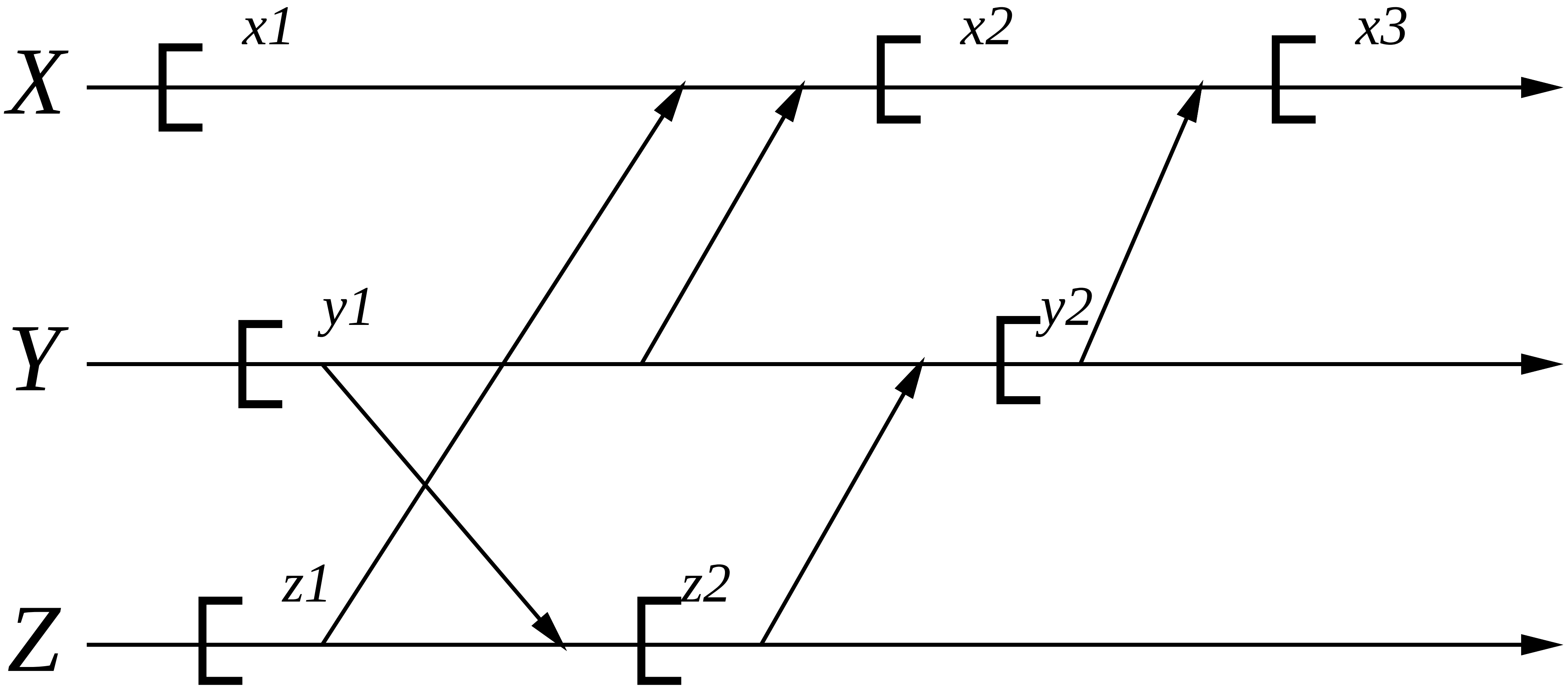
Mensagens orfãs

- Mensagens enviadas por um processo que falhou mas que não foram memorizadas na salvaguarda.

  Ao recuperar o processo pode não voltar a gerar essas mensagens.

  É necessário recuperar os processos que receberam essas mensagens.

Mensagens orfãs: ilustração



Mensagens perdidas

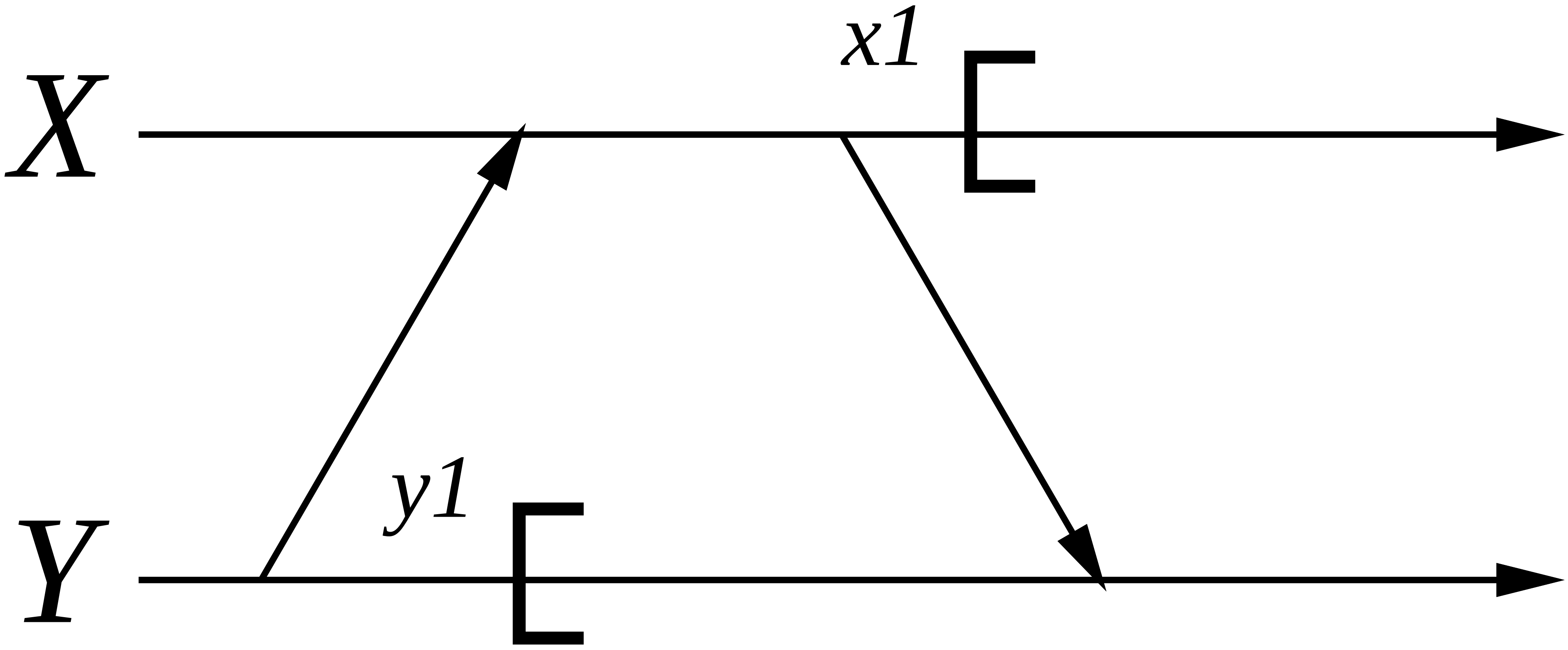
- Mensagem que foi recebida mas não chegou a ser salvaguardada.

  Perde-se ao ser feita a recuperação.

  Equivalente a perder-se no meio de comunicação.

- Se é importante a sua recepção, deve existir um outro mecanismo que garanta a sua retransmissão.

Mensagens perdidas: ilustração



Livelock

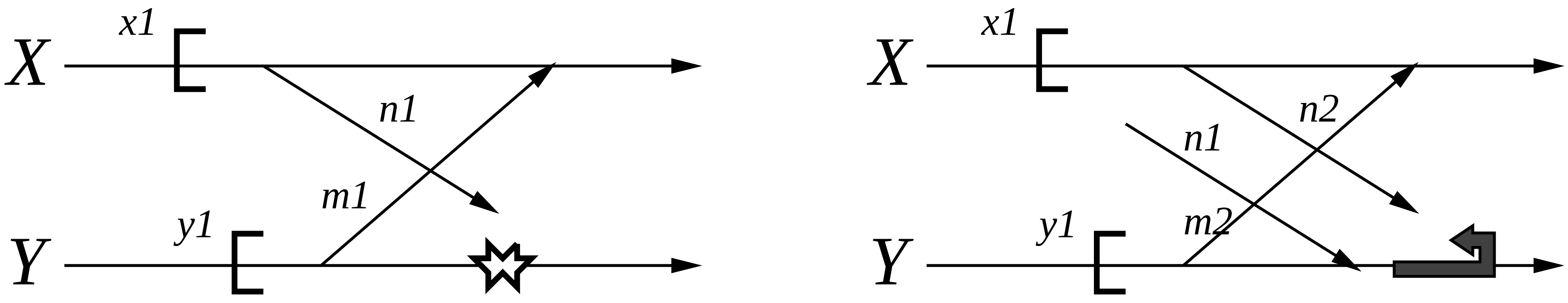
- Se um processo ao recuperar, começar de imediato a processar mensagens de outros processos:

  Pode processar uma mensagem de um processo que posteriormente também foi obrigado a recuperar...

  ... e deve recuperar de novo (e de novo e de novo ...).

- É necessário coordenar a recuperação.

Livelock: ilustração



Conjunto coerente de salvaguardas

- Cada processo têm a sua salvaguarda.
- Um conjunto de salvaguardas diz-se *fortemente coerente* quando:

  Não existem mensagens orfãs.

  Não existem mensagens perdidas.

Salvaguardas coerentes

- É necessário coordenadar os participantes de modo a impedir a troca de informação durante as salvaguaradas.
- Modelo ligeiramente mais fraco: conjunto de salvaguardas *coerente*.

  Permitem-se mensagens perdidas (mas não mensagens orfãs).

Modo simples de obter salvaguardas coerentes

- Fazer uma salvaguarda sempre que se envia uma mensagem.

  De modo a que a mensagens fique sempre na salvaguarda.

- Deste modo não há mensagens orfãs.

- Mais salvaguardas do que o desejável.

Tipos de salvaguardas

- Coordenadas.

  Os processos negoceiam quando devem realizar as salvaguardas, de modo a garantir a sua coerência.

- Independentes.

  Cada processo faz uma salvaguarda de modo independente. Tenta-se encontrar um conjunto coerente na recuperação.

Algoritmo coordenado[koo & Toueg]

- Dois tipos de salvaguardas.

  Provisórias e definitivas.

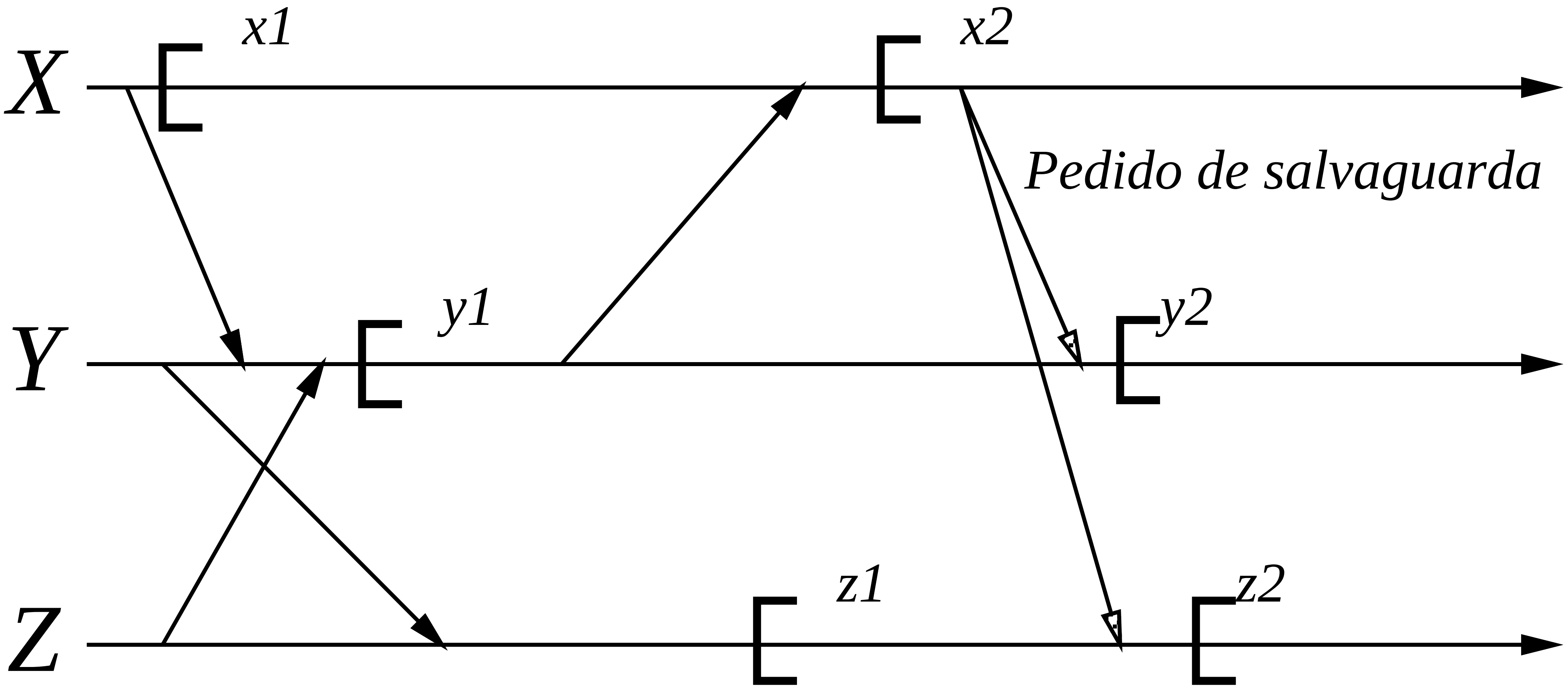
- Baseado num coordenador e canais FIFO.

  O coordenador pede a todos os participantes para fazerem uma salvaguarda provisória.

  Depois de fazer a salvaguarda provisória o participante não envia mais mensagens.

  Depois de todos confirmarem, o coordenador manda tornar as salvaguardas definitivas.

Opitimização: $z2$ é desnecessário



Algoritmo de salvaguarda

Defina-se recebi_de$_p$
    conjunto de processos do qual p recebeu mensagens após a última salvaguarda
Coordenador c
    para cada membro j de recebi_de$_c$
        enviar FazerSalvaguardaProvisória (última_mensagem$_c[j]$);
    esperar que todos membros de recebi_de$_c$ respondam
    para cada membro j de recebi_de$_c$
        enviar TornarSalvaguardaDefinitiva;
Participante p
    ao receber FazerSalvaguardaProvisória (última_mensagem$_q[p]$) vindo de q
        se última_mensagem$_q[p]$ não está na última salvaguarda
            fazer salvaguarda provisória
            para cada membro k de recebi_de$_p$
                enviar FazerSalvaguardaProvisória (última_mensagem$_p[k]$);
                esperar que todos membros de recebi_de$_p$ respondam
        responder a p
    ao receber TornarSalvaguardaDefinitiva
        para cada membro k de recebi_de$_p$
            enviar TornarSalvaguardaDefintiva;
        tornar a salvaguarda definitiva;

Reposição do estado após falha[Koo&Toueg]

- Baseado em coordenador e canais FIFO.

  Só um coordenador executa de cada vez.

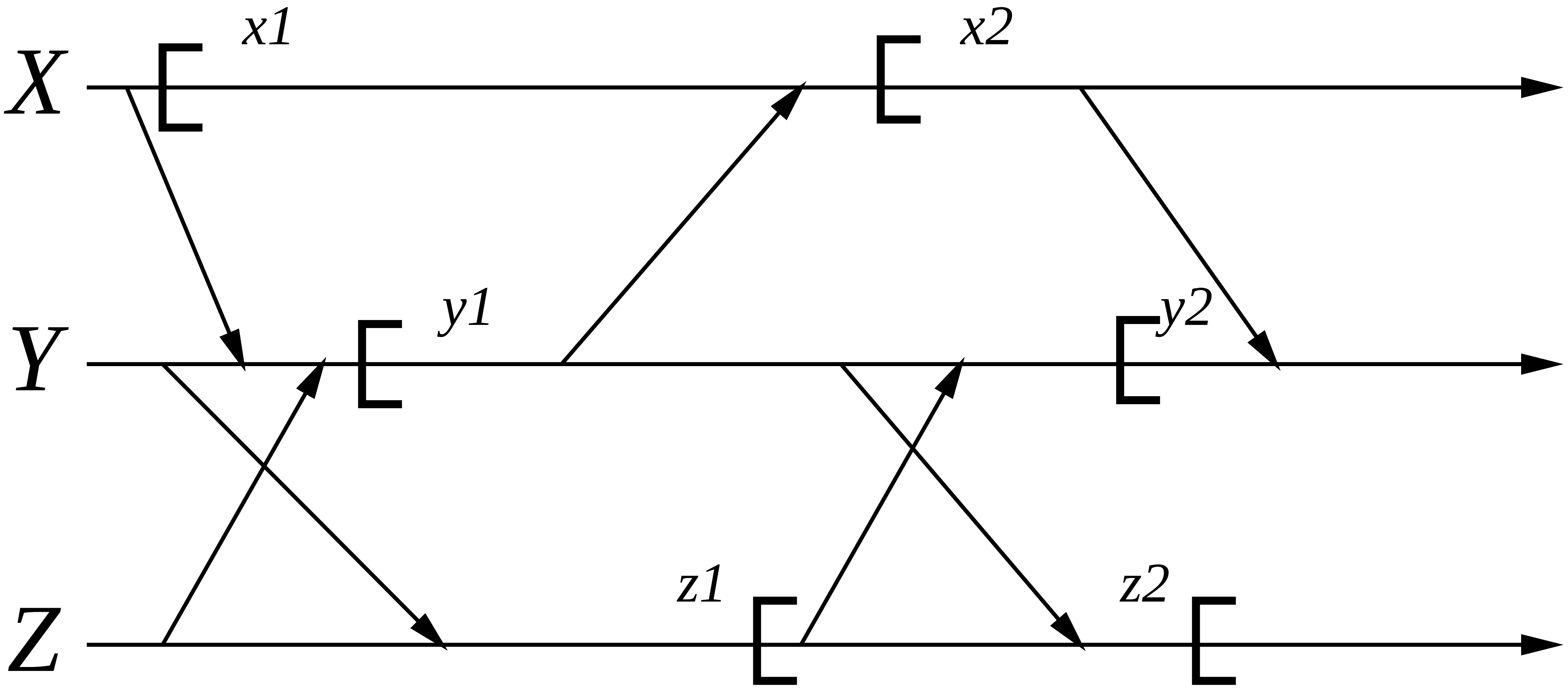
  Não há processamento durante a execução do algoritmo.

- Em duas fases.

  Preparar para recuperar.

  Recuperar para o último conjunto de salvaguardas.

Optimização: $z2$ não necessita recuperar



Algoritmo de reposição

Defina-se comunica_com$_p$
conjunto de processos para os quais p envia mensagens
Coordenador c
para cada membro j de comunica_com$_c$
enviar PrepararReposição (última_salvaguardada$_c[j]$);
esperar que todos membros de comunica_com$_c$ respondam
para cada membro j de comunica_com$_c$
enviar ExecutarReposição;
Participante p
ao receber PrepararReposição (última_salvaguardada$_q[p]$) vindo de q
se última_recebida$_p[q]$ $>$ última_salvaguardada$_q[p]$
para cada membro k de comunica_com$_p$
enviar PrepararReposição (última_salvaguardada$_p[k]$);
esperar que todos membros de comunica_com$_p$ respondam
responder a p
ao receber ExecutarReposição;
para cada membro k de rcomunica_com$_p$
enviar ExecutarReposição;
executar a reposição;

Reposição: impedir mensagens em trânsito

- Todos os participantes devem comunicar explicitamente com os potenciais recipientes de mensagens.
- O facto dos canais serem FIFO, faz com que esta comunicação "limpe" o canal.

  Isto é, garante que não há mensagens em trânsito.

  Impedem-se os cenários de "livelock".

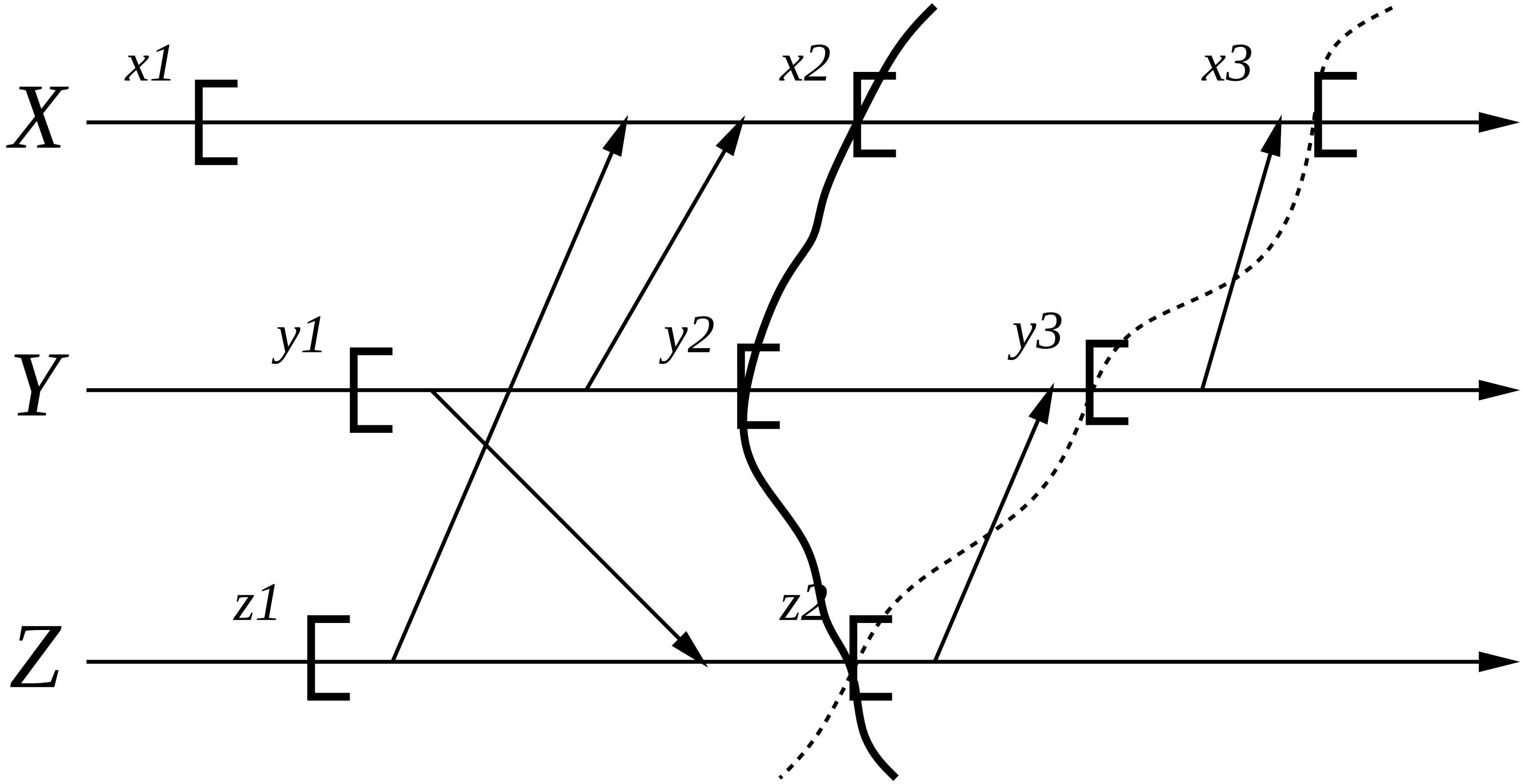
Desvantagens da salvaguarda coordenada

- Troca adicional de mensagens.
- Períodos sem actividade só para realizar as salvaguardas.
- Elevada sobrecarga na ausência da faltas.

Salvaguarda não coordenada

- Cada participante decide fazer uma salvaguarda de modo independente.
- Durante a recuperação é necessário procurar um conjunto de salvaguardas coerente.

Salvaguarda não coordenada



Reposição com salvaguarda não coordenada

1. Por convergência.
2. Todos os participantes, indicam a todos quais as mensagens que fazem parte da última salvaguarda.
3. Se um participante detectar que a sua salvaguarda tem uma mensagem orfã, decide retroceder para uma salvaguarda anterior.
4. Retorna-se ao ponto 2.

Para maior eficiência na reposição

- Caso as computações sejam deterministas.
- Cada participante mantém todas as mensagens que processa entre duas salvaguardas.
- Deste modo pode re-construir um estado intermédio.
- Maior probabilidade de encontrar um estado coerente quando se consideram os estados intermédios.

  Evita-se o efeito dominó.

Salvaguarda não coordenada pessimista

- Se a computação for determinista pode fazer-se uma salvaguarda das mensagens recebidas.

  Intercalada por salvaguardas de estado.

- E se um processo não processar nenhuma mensagem antes desta estar salvaguardada.

- O processo consegue sempre recuperar isoladamente.

  Na prática, faz-se uma salvaguarda (incremental) a cada evento.

Resumo

- Mensagens orfãs.
- Efeito dominó.
- Conjunto de salvaguardas coerentes.
- Salvaguarda coordenada.
- Recuperação não-coordenada.
- Salvaguarda não-coordenada.