

**Universidade de São Paulo
Instituto de Ciências Matemáticas e de Computação
Departamento de Ciências de Computação**



Simulação Distribuída

**Uma Aplicação de Computação Distribuída
Uma Ferramenta para Avaliação de
Desempenho**

Marcos José Santana

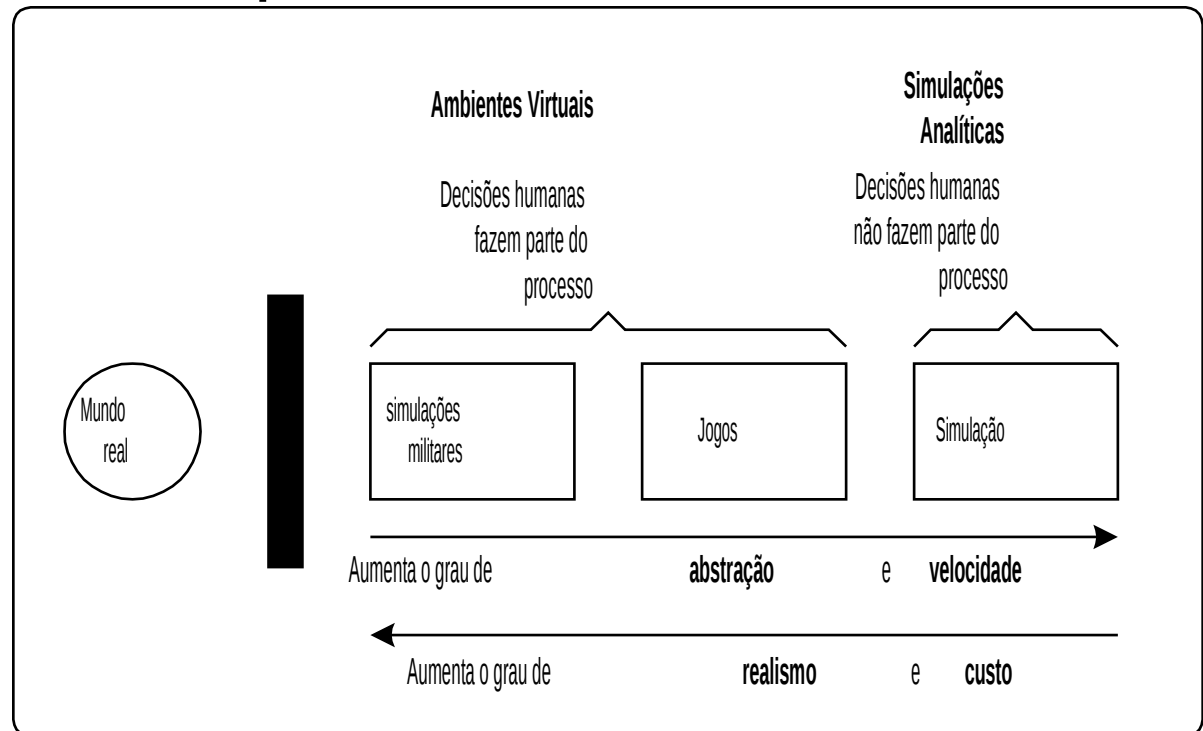


Simulação - Definição

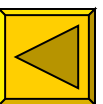
- Desenvolvimento de um modelo matemático ou lógico de um sistema real e então realizar experimentos de modo a prever o comportamento real do sistema
- Vantagens:
 - Versatilidade (aplicada em diferentes situações)
 - Flexibilidade (adaptável a novas situações)
 - Baixo custo (com um mesmo programa pode-se simular diferentes situações do mesmo problema)

Simulação - Utilização

- Criação de ambientes virtuais
- Avaliação de desempenho de sistemas complexos



Exemplo – Simulação Ambientes Virtuais

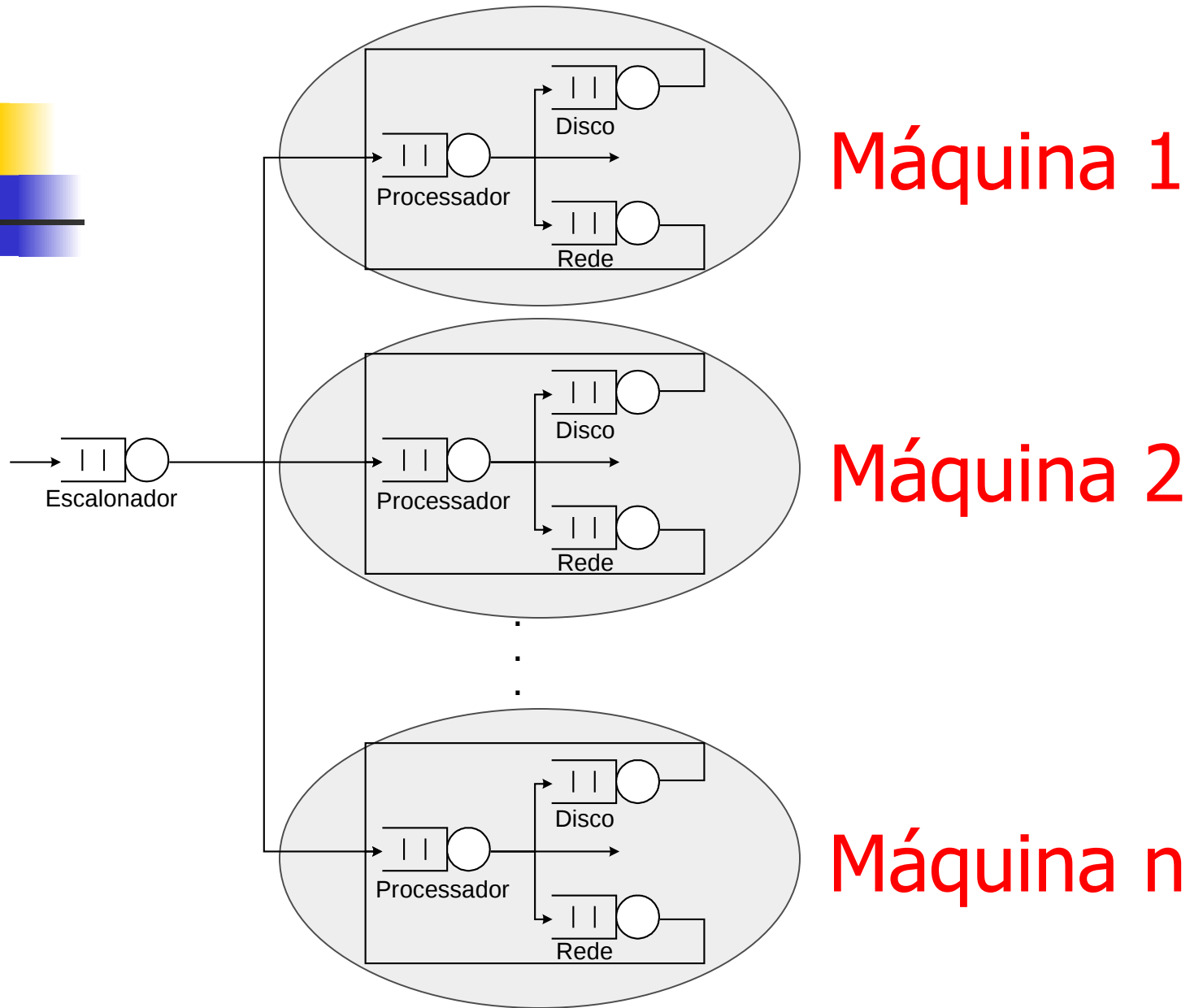
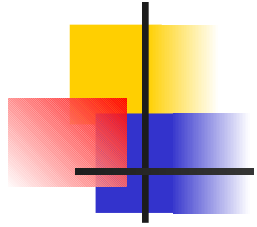




Exemplo – Simulação

Avaliação de Desempenho

- Simulação de um ambiente que faz escalonamento de processos considerando a potência computacional das máquinas
- Pode-se avaliar:
 - Desempenho do ambiente com um determinado índice de carga
 - Desempenho do ambiente considerando diferentes arquiteturas





Simulação Seqüencial

- Orientação por eventos
 - Apenas um evento pode ser simulado por vez
 - Lista de Eventos Futuros
- Orientação por processos
 - Criação de ambiente de programação concorrente
- Orientação por atividades
 - Detalhamento das atividades de um processo



Processos X Eventos X Atividade

PROCESSO



The diagram illustrates the relationship between a process, activities, and events over time. A horizontal timeline at the bottom is labeled 'tempo' with an arrow pointing right. Three vertical arrows point downwards from the timeline to three event labels: 'EVENTO DE CHEGADA', 'EVENTO DE ENTRADA EM SERVIÇO', and 'EVENTO DE FIM DE SERVIÇO'. Above the timeline, a bracket labeled 'PROCESSO' spans the entire duration. A horizontal arrow labeled 'ATIVIDADE' points from the second event to the third event, indicating the activity occurring between them.

ATIVIDADE

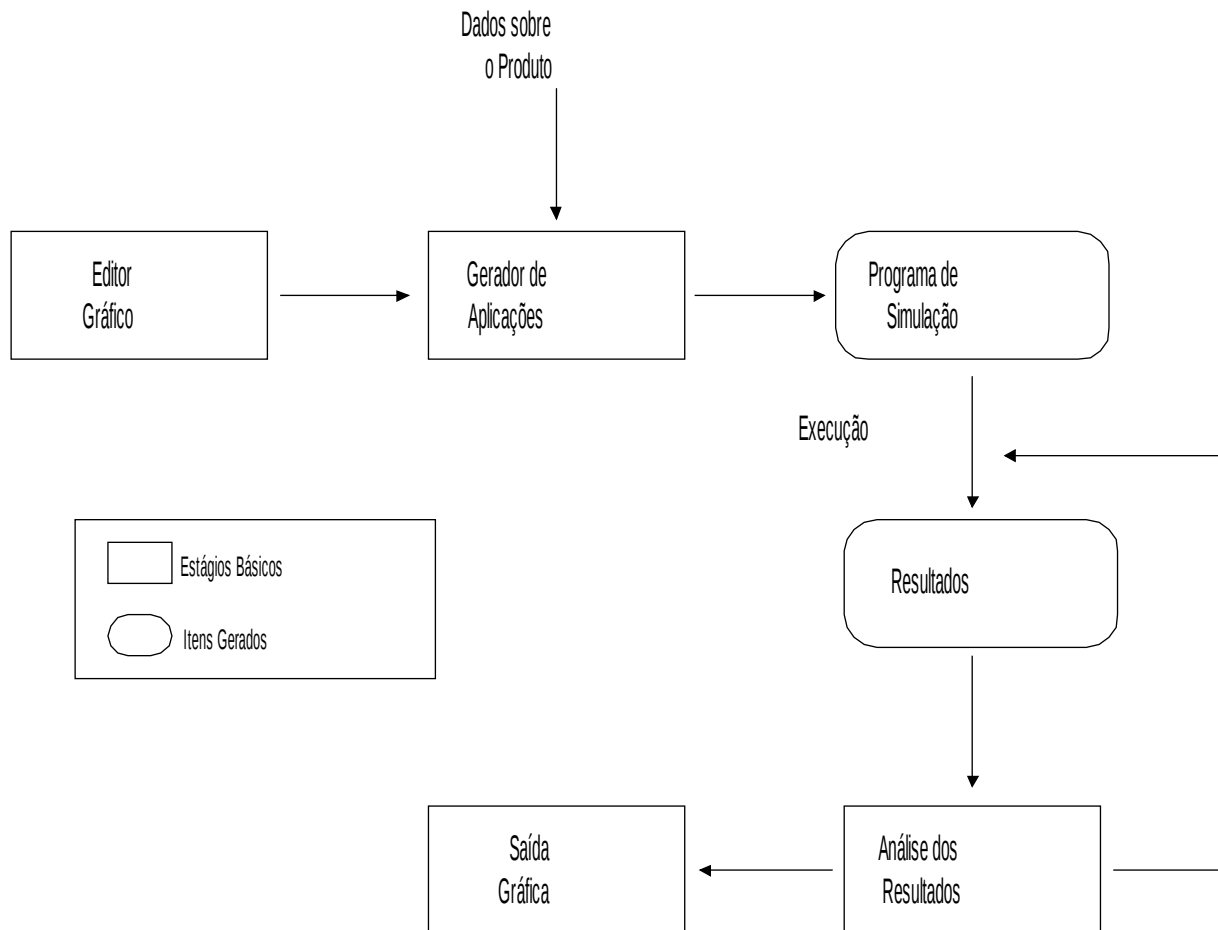
**EVENTO DE
CHEGADA**

**EVENTO DE
ENTRADA EM
SERVIÇO**

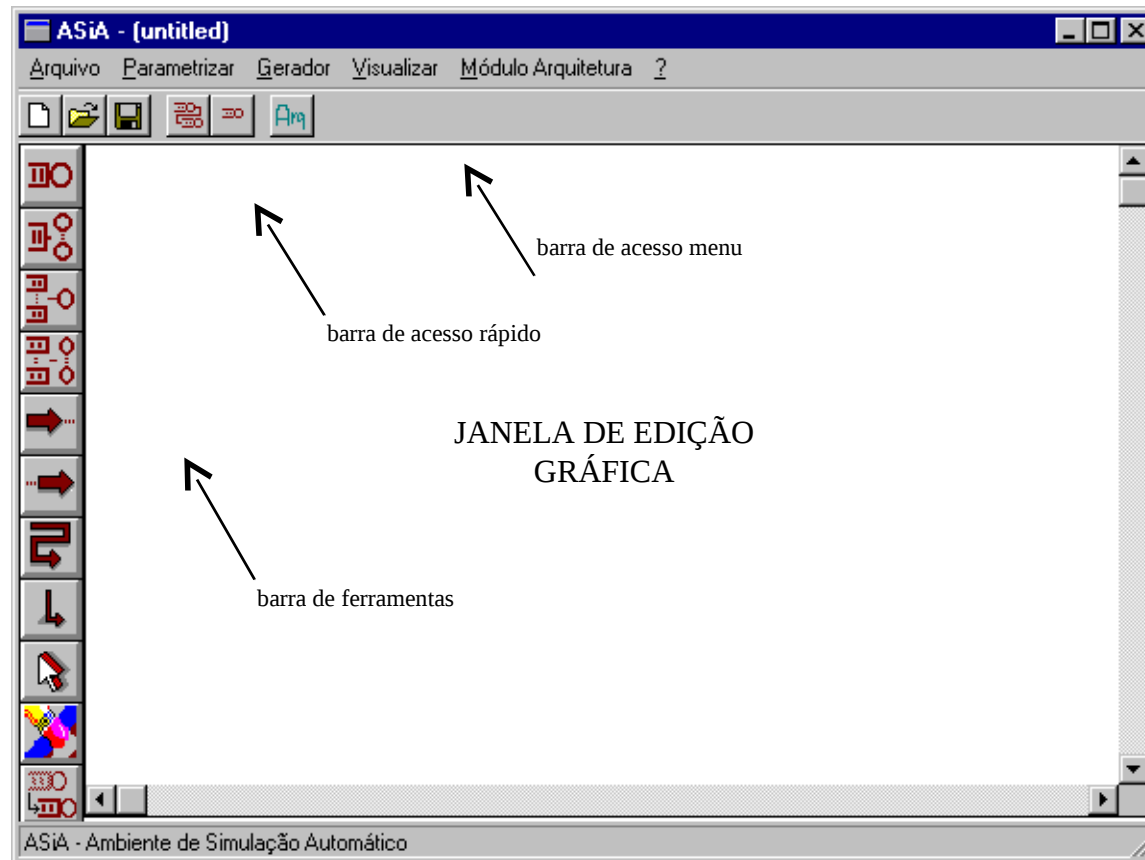
**EVENTO DE
FIM DE SERVIÇO**

tempo

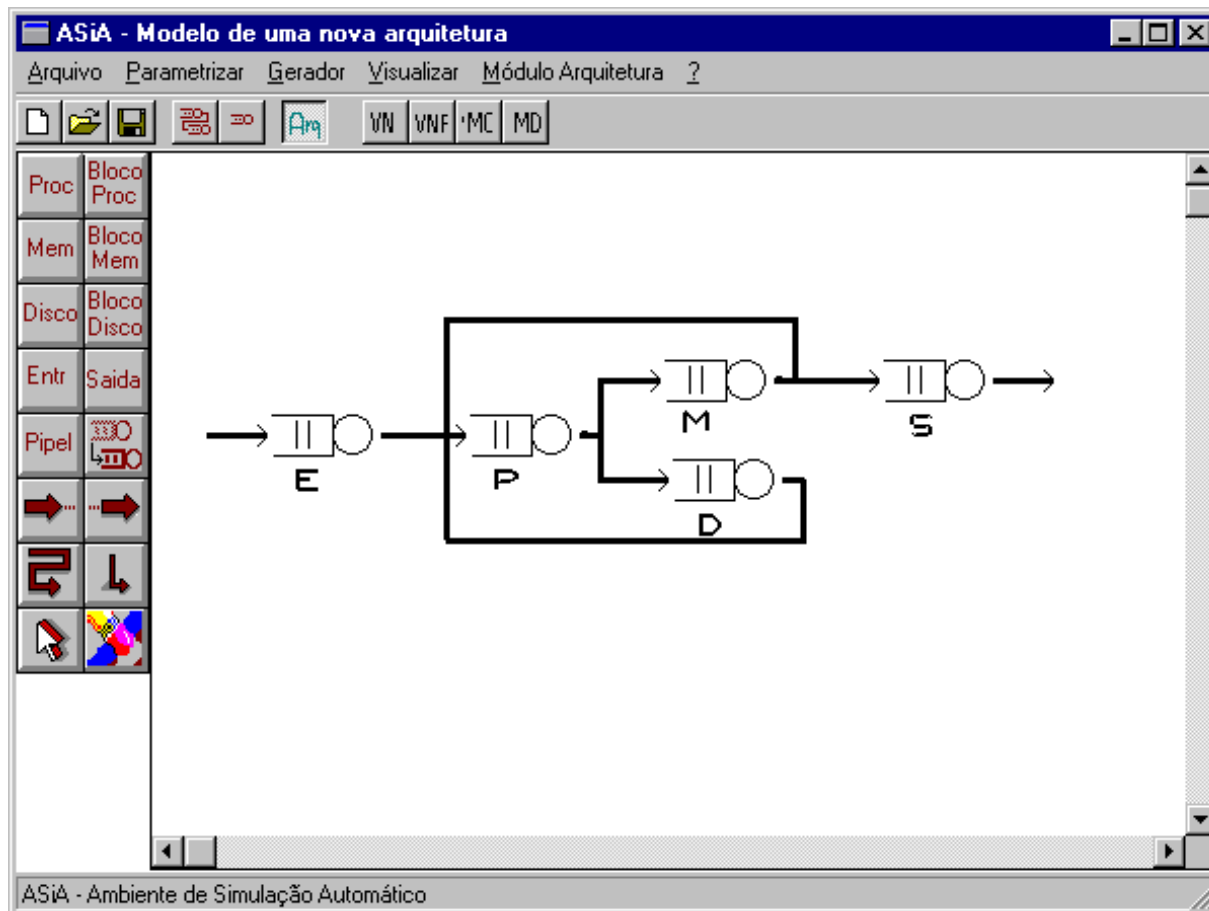
ASIA – Ambiente de Simulação Automático



ASIA – Ambiente de Simulação Automático



ASiA – Ambiente de Simulação Automático



ASIA – Ambiente de Simulação Automático

Von_neum.out - Bloco de Notas

Arquivo Editar Pesquisar Ajuda

smp1 SIMULATION REPORT

MODEL: von Neumann

TIME: 10027.754

INTERVAL: 10027.754

FACILITY	UTIL.	MEAN BUSY PERIOD	MEAN QUEUE LENGTH	OPERATION COUNTS		
				RELEASE	PREEMPT	QUEUE
Entrada1	0.9017	110.269	6.043	82	0	68
Processador1	0.9848	89.773	23.334	110	0	110
Disco1	0.4286	84.272	0.250	51	0	22
Memoria1	0.4479	86.370	0.384	52	0	28
Saida1	0.2903	97.039	0.063	30	0	8



Simulação Distribuída

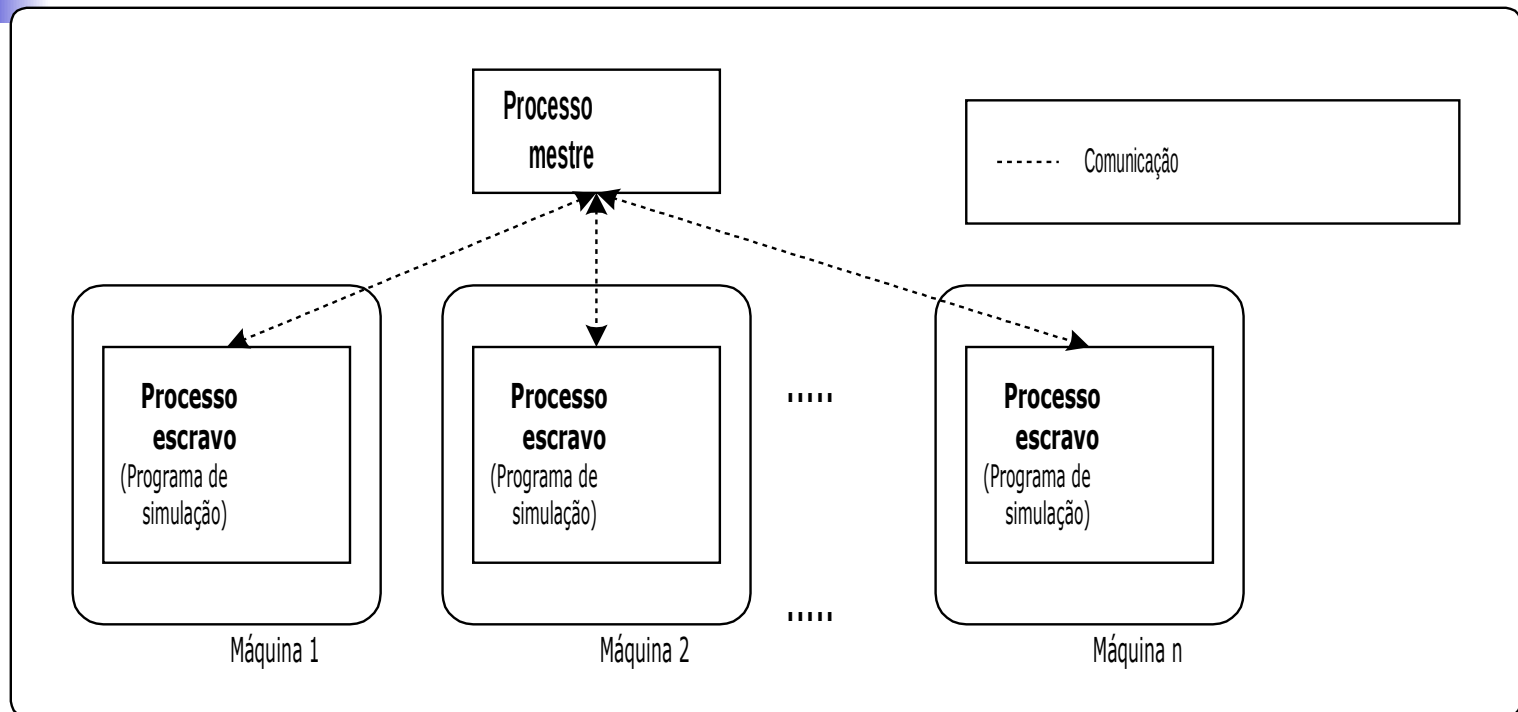
- Simulação Distribuída
 - União de duas áreas: simulação + computação paralela distribuída
 - Utilização de diversos processadores / computadores interligados com o objetivo principal de diminuir o tempo de execução da simulação (para o caso de simulação analítica)
 - Duas técnicas:
 - MRIP
 - SRIP



MRIP

- *Multiple Replication in Parallel*
 - Independentes replicações de um programa seqüencial de simulação são executados simultaneamente em diferentes processadores
 - Cada instância é independente uma da outra e enviam as observações para uma central que calcula uma única estimação do valor médio de cada parâmetro observado
 - Quando a precisão é atingida, a simulação é encerrada

MRIP





SRIP

- *Single Replication in Parallel*
 - Processos lógicos que se comunicam através da troca de mensagens
 - Problema: sincronização dos tempos de simulação entre os processos lógicos -> erros de causa e efeito
 - Solução: protocolos de sincronização: conservativos e otimistas



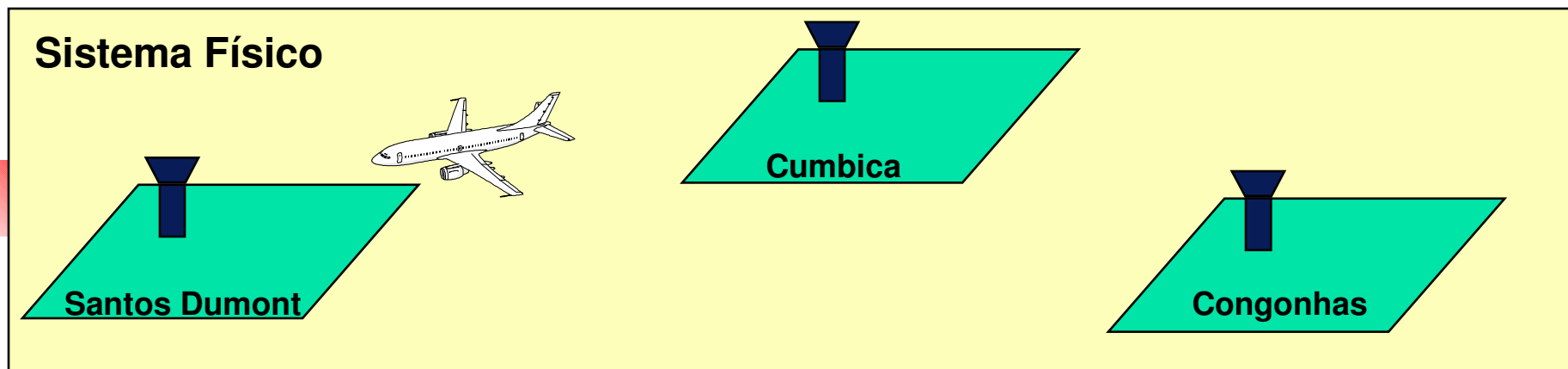
SRIP - Exemplo

- Uma rede de aeroportos
- Cada aeroporto pode ser considerado como um **processo lógico**
- Os processos lógicos podem escalonar eventos (enviando mensagens) para outros processos lógicos



SRIP - Exemplo

- Sistema Físico
 - Coleção de processos físicos que interagem entre si (aeroportos)
- Simulação
 - Coleção de processos lógicos (LPs - *Logical Process*)
 - Cada LP modela um processo físico
 - As interações entre os processos físicos são modeladas através do escalonamento de eventos entre os LPs



Processo Físico

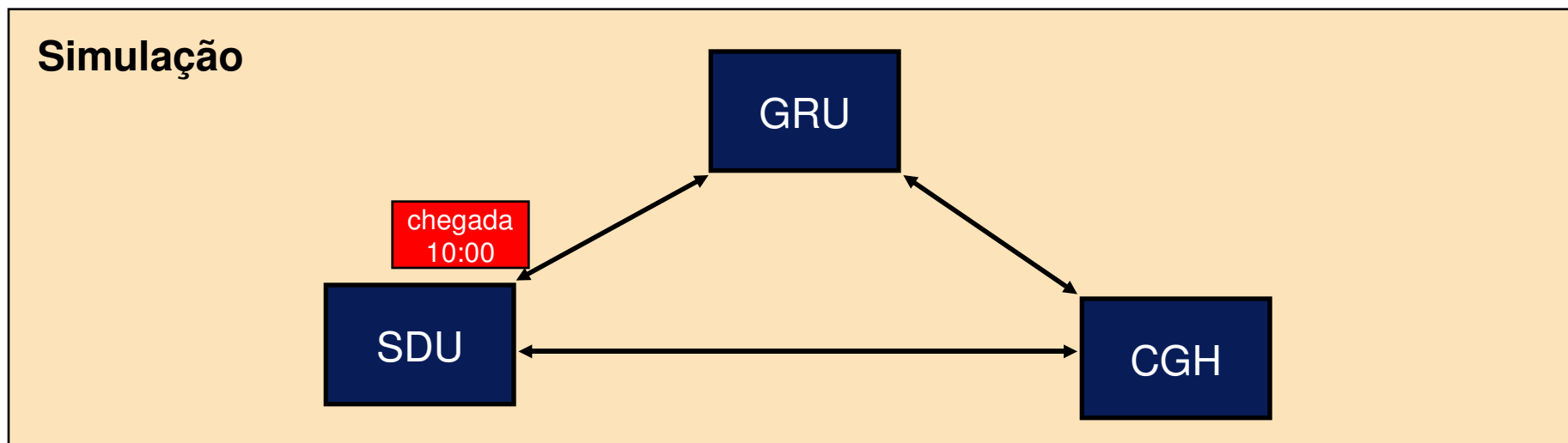


Processo Lógico

Interações entre os processos físicos

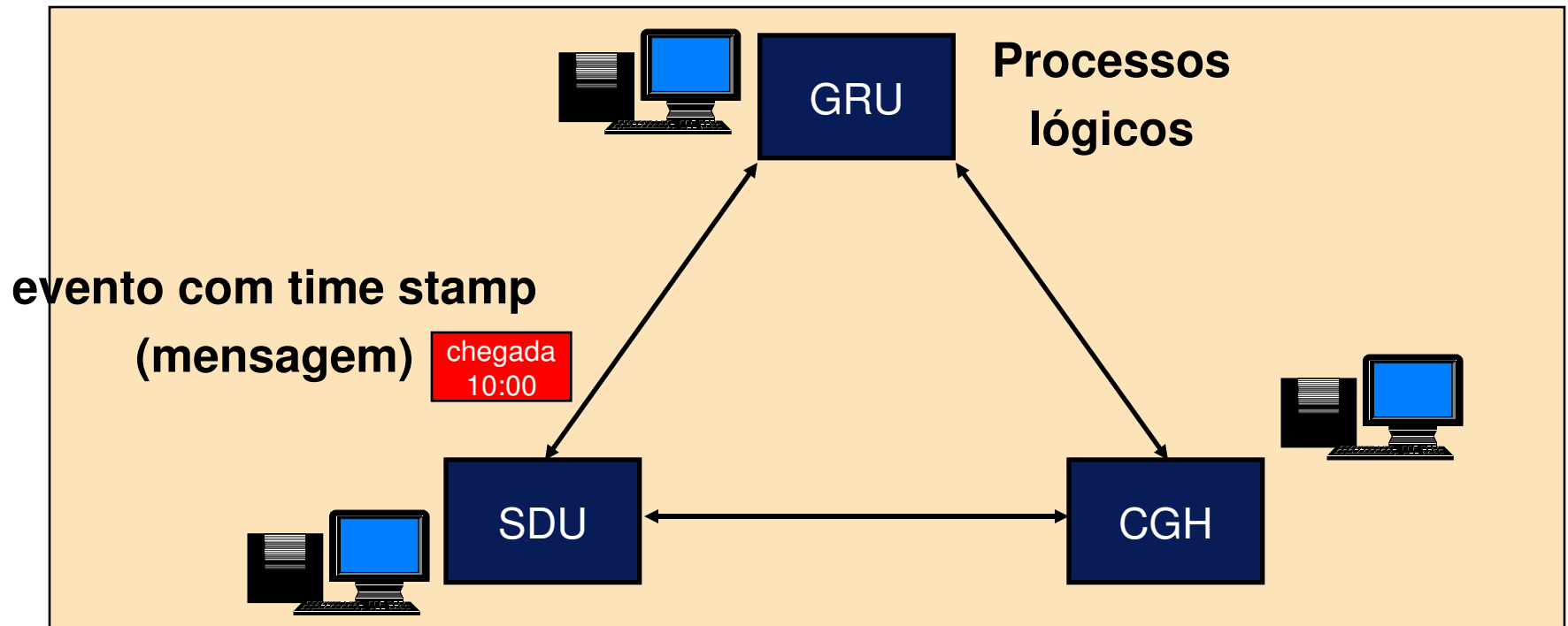


Evento com time stamp (mensagem)



Toda a interação entre os LPs deve ser por troca de mensagens

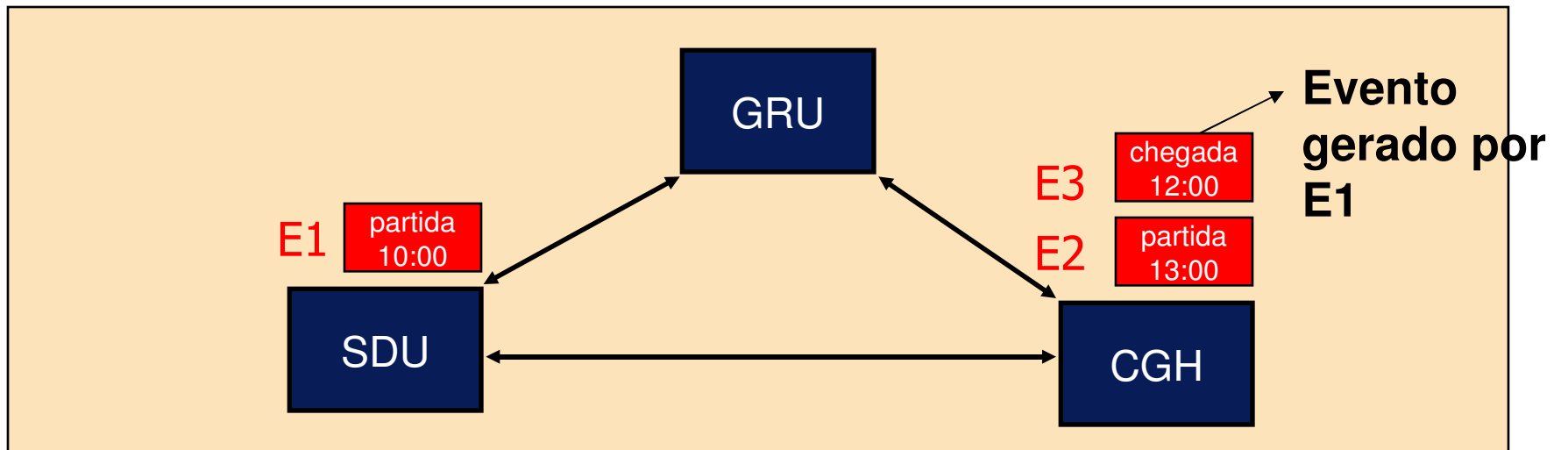
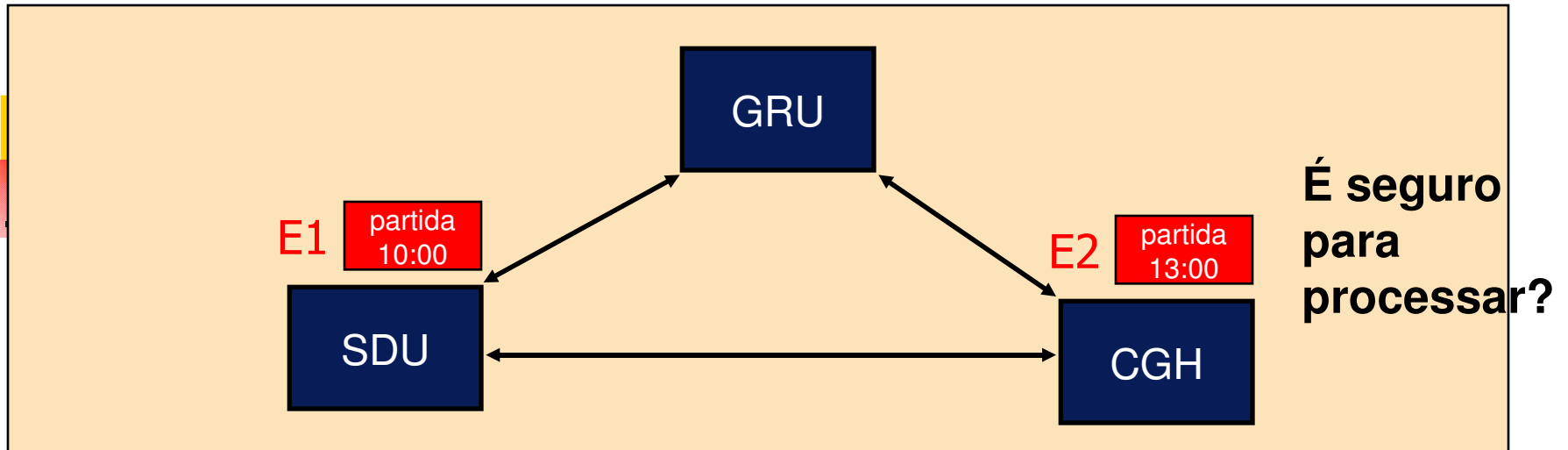
- Diversos processos lógicos mapeados em diversos processadores
- Comunicação é feita através de passagem de mensagens
 - Toda interação é feita através das mensagens
 - Não existe variáveis compartilhadas
- Mapeamento dos LPs em diferentes processadores
 - Pode haver múltiplos elementos por processador



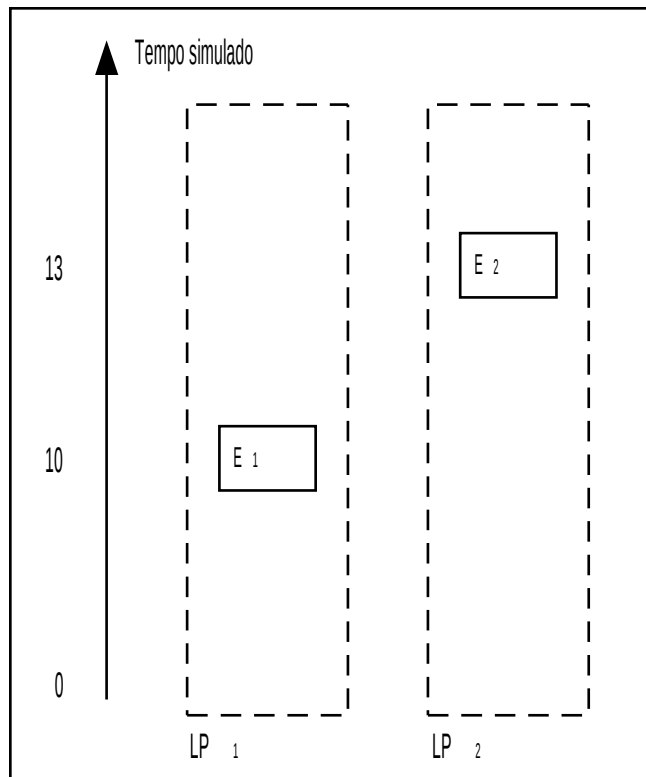


SRIP – Erro de Causa e Efeito

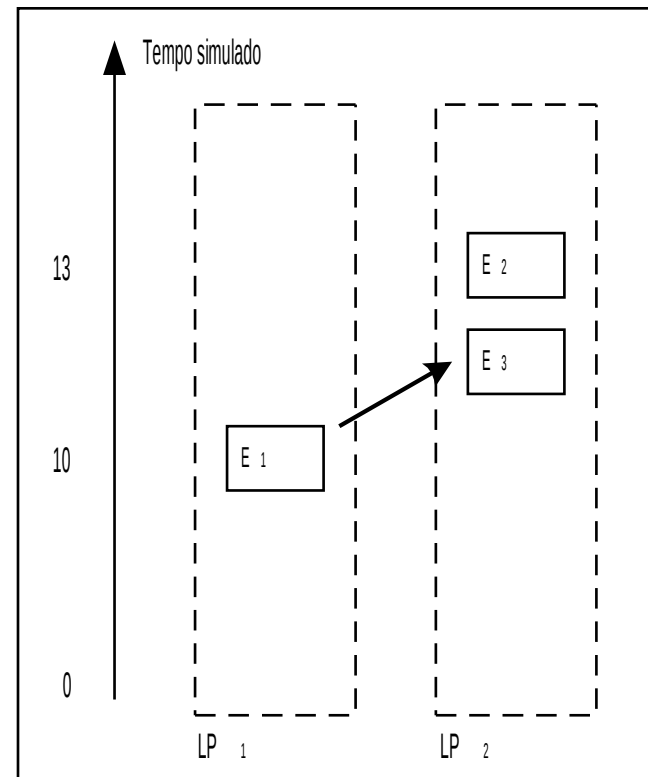
- Regra para todo processo: “Deve-se processar as mensagens que chegam em ordem de time stamp” (**local causality constraint**)



Erro de Causa e Efeito



(a)



(b)



SRIP

- Problema de sincronização:
 - é necessário um algoritmo para garantir que cada LP processe os eventos em ordem de time stamp



SRIP

- **Sincronização conservativa:** evita a violação do erro de causa e efeito (espera até que seja seguro)
 - impedimento do deadlock utilizando mensagens nulas (Chandy/Misra/Bryant)
 - detecção e recuperação do deadlock
- **Sincronização otimista:** permite violações de causalidade local mas detecta-as em tempo de execução e recupera utilizando um mecanismo de rollback
 - Time Warp (Jefferson)
 - muitas outras alternativas



Protocolos Conservativos

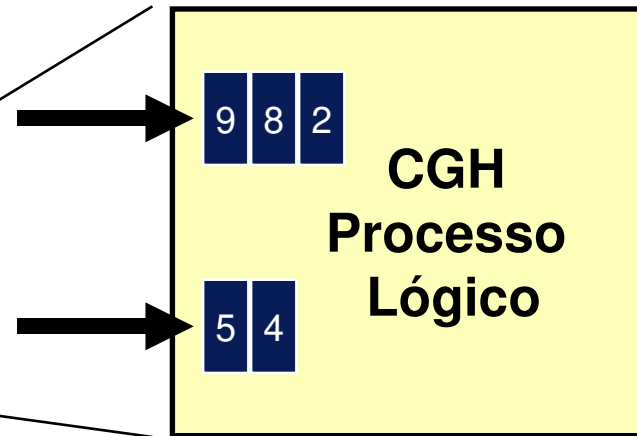
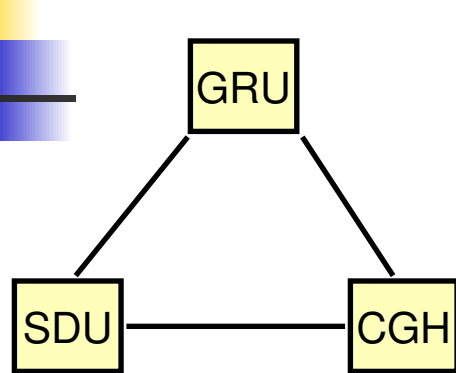
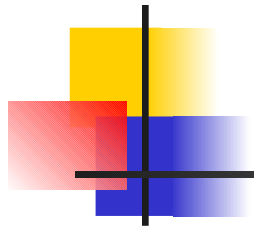
- Impedem a ocorrência de erros de causa e efeito determinando quando é seguro executar um evento
- Problemas:
 - *deadlock*
 - o paralelismo pode não ser bem explorado
- A diferença entre os protocolos conservativos é a maneira com que eles tratam o *deadlock*



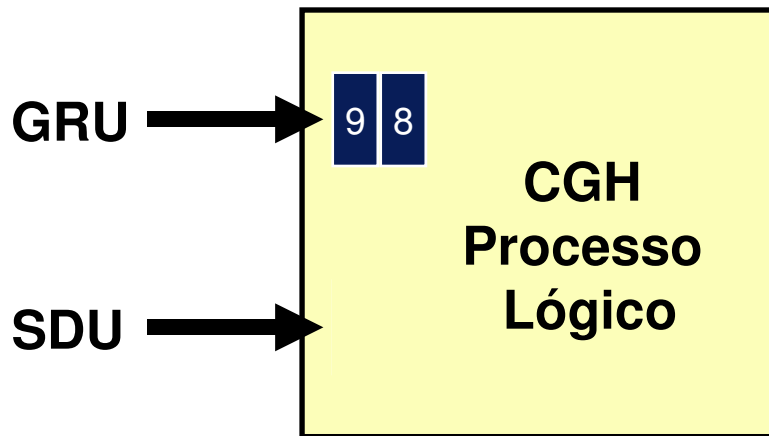
Protocolos Conservativos

- Processos lógicos trocam eventos com time stamp (mensagens)
- Topologia de conexão entre os processos lógicos é estática
- As mensagens enviadas em cada conexão são enviadas em ordem de time stamp
- A rede fornece entrega confiável dos pacotes, preservando a ordem das mensagens

Time stamp da última mensagem recebida em uma conexão é o menor valor de time stamp das subsequentes mensagens que serão recebidas naquela conexão.



uma fila FIFO
por conexão
que chega



- processa evento com time stamp 2
- processa evento com time stamp 4
- processa evento com time stamp 5
- espera até que chegue uma mensagem de SDU



Protocolos Conservativos

Algoritmo A (executado por cada LP):

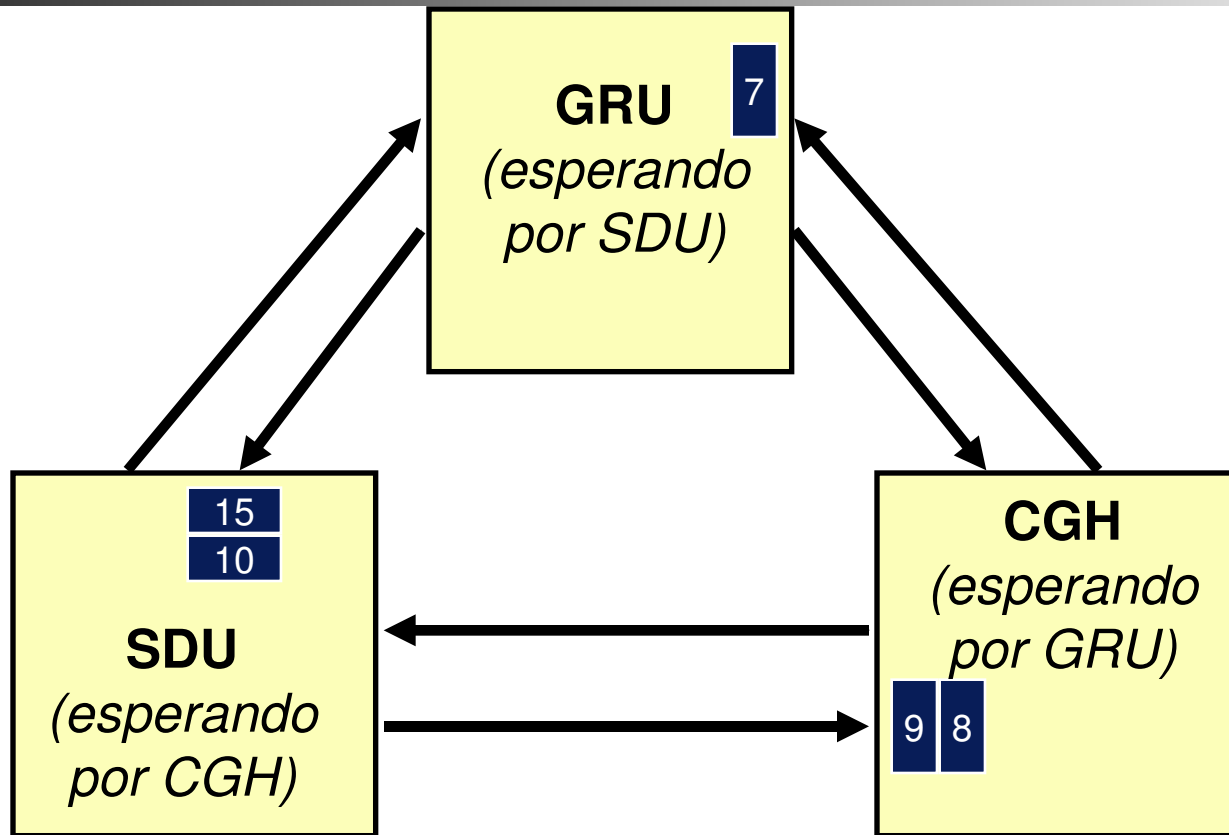
Objetivo: Garantir que os eventos são processados em ordem de time stamp:

WHILE (simulação não acabou)

 espere até que cada FIFO contenha pelo menos uma mensagem
 remova o evento com menor time stamp da FIFO
 processe aquele evento

END-WHILE

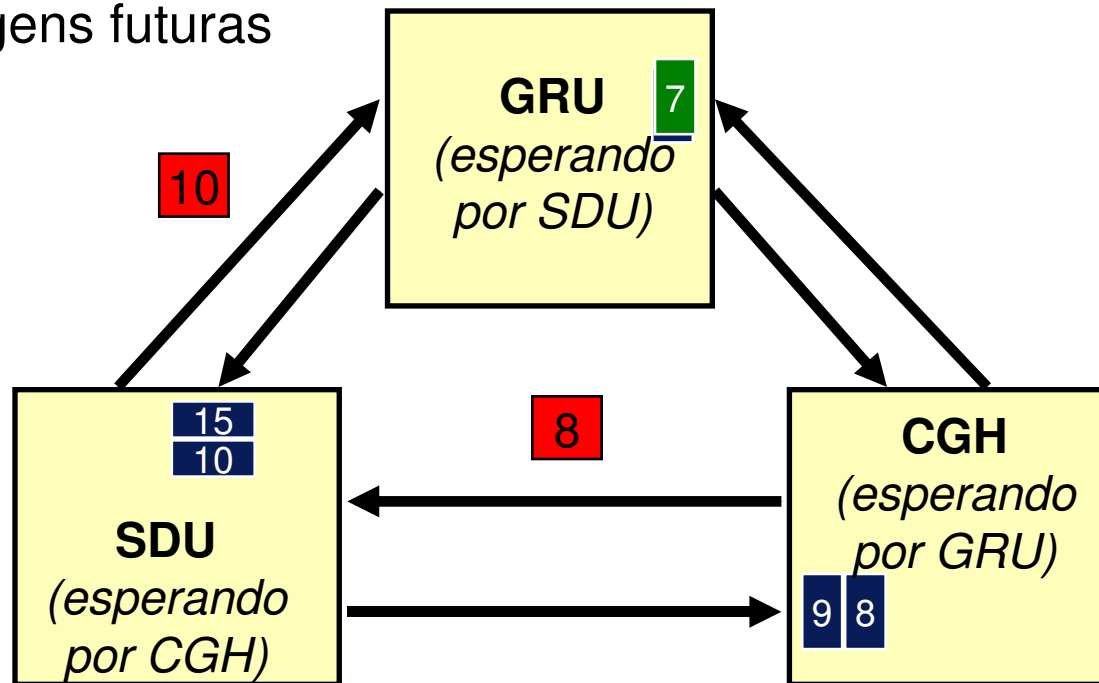
Protocolos Conservativos



DEADLOCK

Protocolos Conservativos

Quebra do deadlock: cada LP envia uma mensagem nula (“null message”) indicando o seu valor mínimo de time stamp para mensagens futuras



- CON envia uma mensagem nula para SDU com time stamp 8
- SDU envia uma mensagem nula para CUM com time stamp 10
- CUM pode processar a mensagem com time stamp 7



Protocolos Conservativos

Algoritmo com Mensagem Nula (executado por cada LP):

Objetivo: Garantir que os eventos são processados em ordem de time stamp e evitar o deadlock

WHILE (simulação não acabou)

espere até que cada FIFO contenha pelo menos uma mensagem
remova o evento com menor time stamp da FIFO
processe aquele evento

*envie mensagens nulas para os LPs vizinhos com o time stamp
indicando o valor mínimo das mensagens futuras (clock corrente mais
lookahead)*

END-WHILE

O algoritmo com mensagem nula conta com a capacidade de “olhar a frente” (lookahead)



Protocolos Otimistas

- Não impedem os erros de causa e efeito
- Assim que um erro é detectado, um mecanismo de *rollback* é acionado para voltar a um estado seguro
- Permite explorar totalmente o paralelismo



Protocolos Otimistas

- Processos lógicos trocam eventos com time stamp (mensagens)
- Topologia de conexão entre os processos lógicos pode ser criada dinamicamente
- As mensagens enviadas em cada conexão não precisam ser enviadas em ordem de time stamp
- A rede fornece entrega confiável dos pacotes, preservando a ordem das mensagens

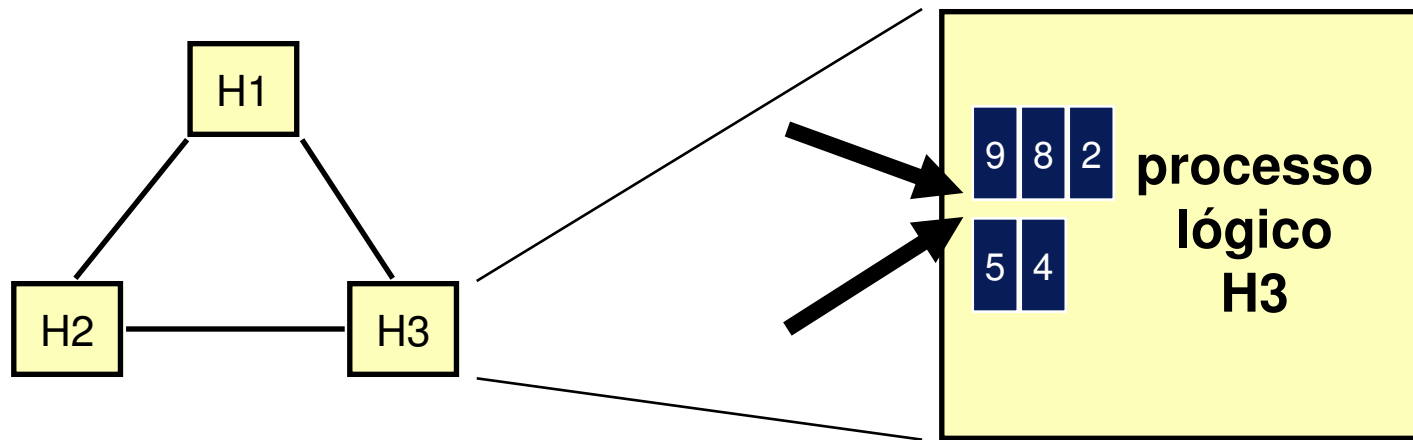


Protocolos Otimistas

Idéia:

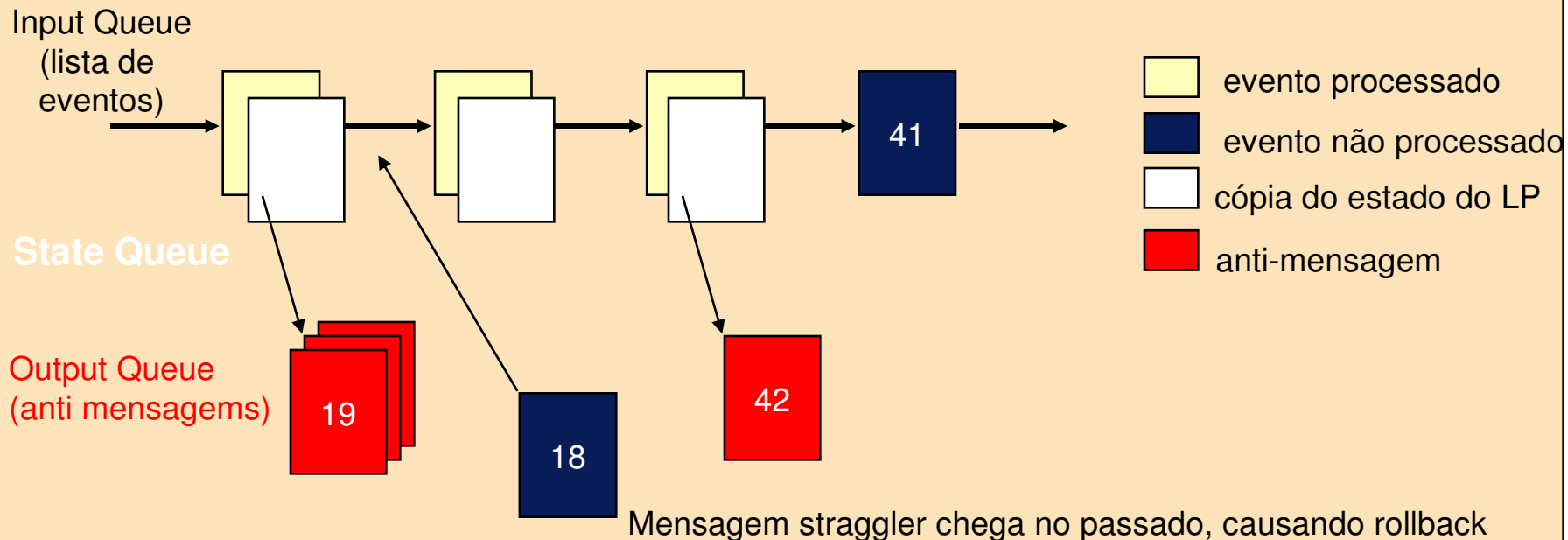
- processamento dos eventos sem se preocupar que mensagens chegarão mais tarde
- se for detectada uma execução fora de ordem, utiliza-se um mecanismo de rollback

Protocolos Otimistas



processa todos os eventos (2, 4, 5, 8, 9) em ordem de time stamp

Cada LP: processa os eventos em ordem de time stamp, como em uma simulação sequencial, exceto: (1) NÃO descarta os eventos processados e (2) adiciona um mecanismo de rollback



Adicionando rollback:

- uma mensagem que chega no passado do LP inicializa o rollback
- para fazer um rollback em um evento computado deve-se:

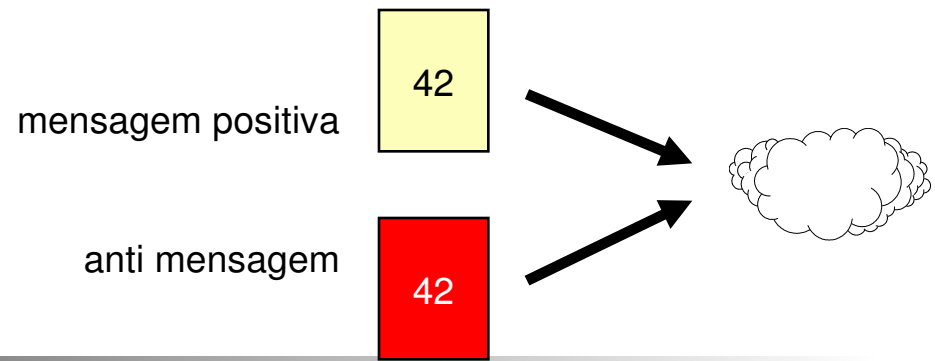
- modificar o estado das variáveis modificadas pelo evento

solução: checkpoint state ou incremental state saving (state queue)

- envio de mensagens

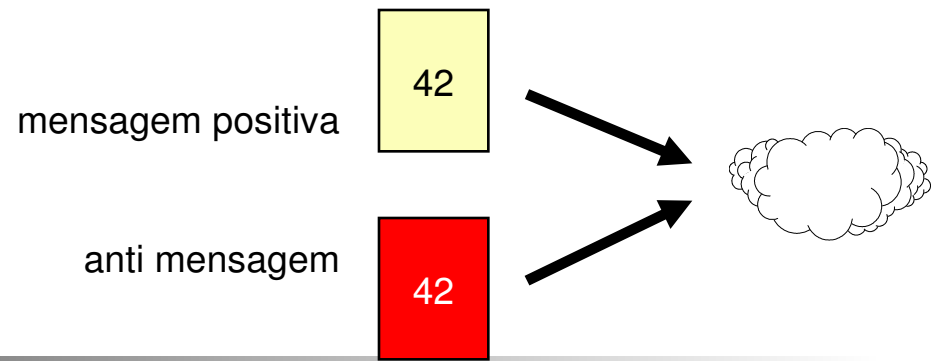
solução: anti-mensagens (output queue)

Anti mensagens

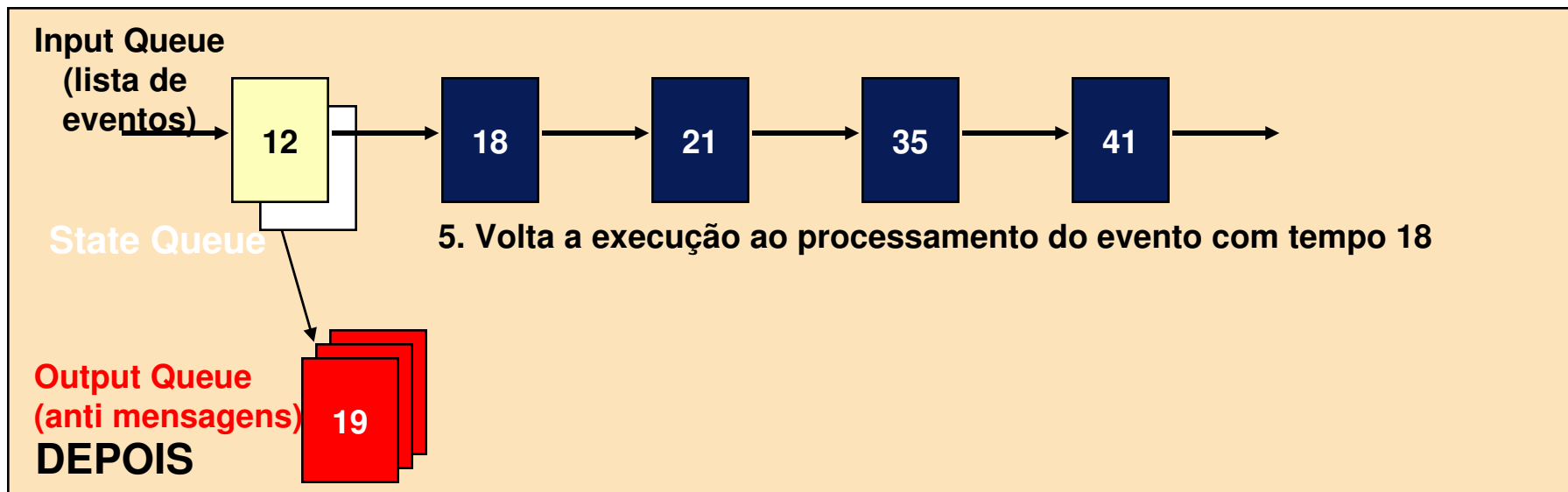
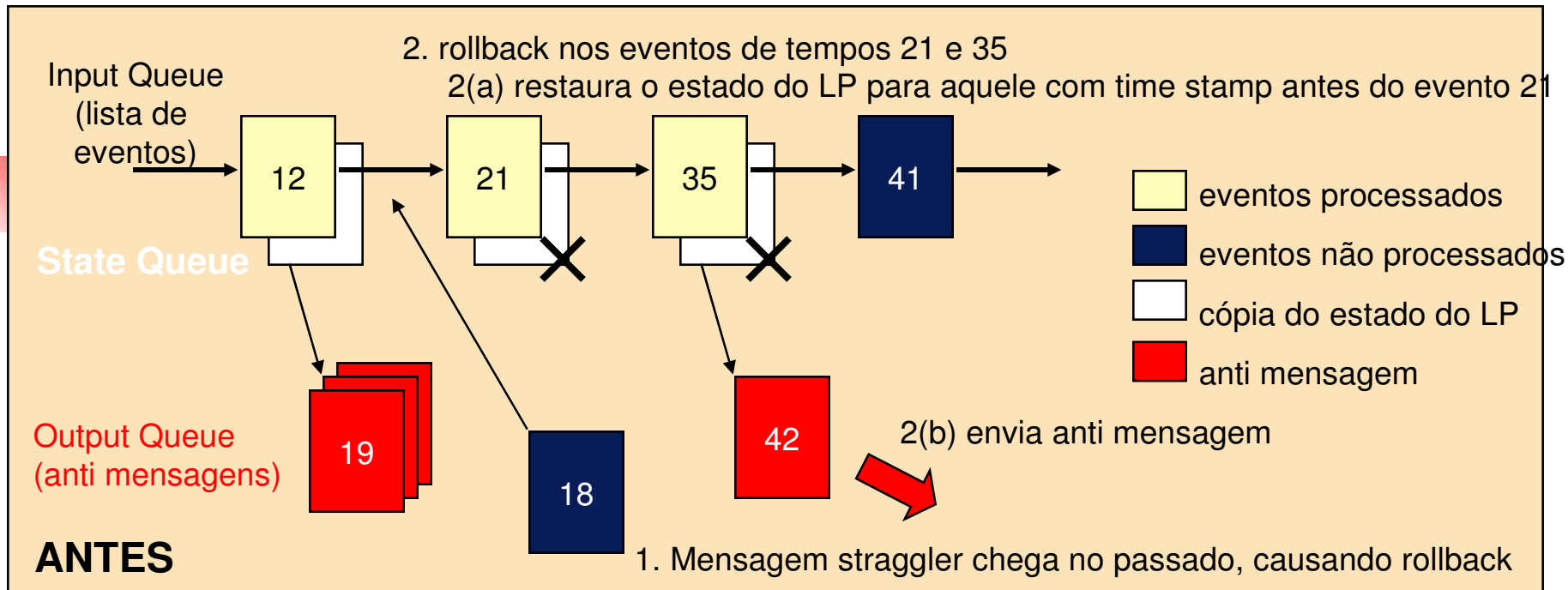


- Utilizadas para cancelar uma mensagem previamente enviada
- Cada mensagem positiva enviada por um LP tem uma anti mensagem correspondente
- Uma anti mensagem é idêntica a uma mensagem positiva, exceto por um bit de sinalização

Anti mensagens



- Quando uma anti mensagem e sua mensagem positiva se encontram na mesma fila, então as duas são eliminadas
- Para desfazer os efeitos de uma mensagem positiva previamente enviada, o LP precisa somente enviar a anti mensagem correspondente
- Envio da mensagem: além de enviar uma mensagem, deve-se deixar uma cópia da anti mensagem em uma estrutura denominada *Output queue*

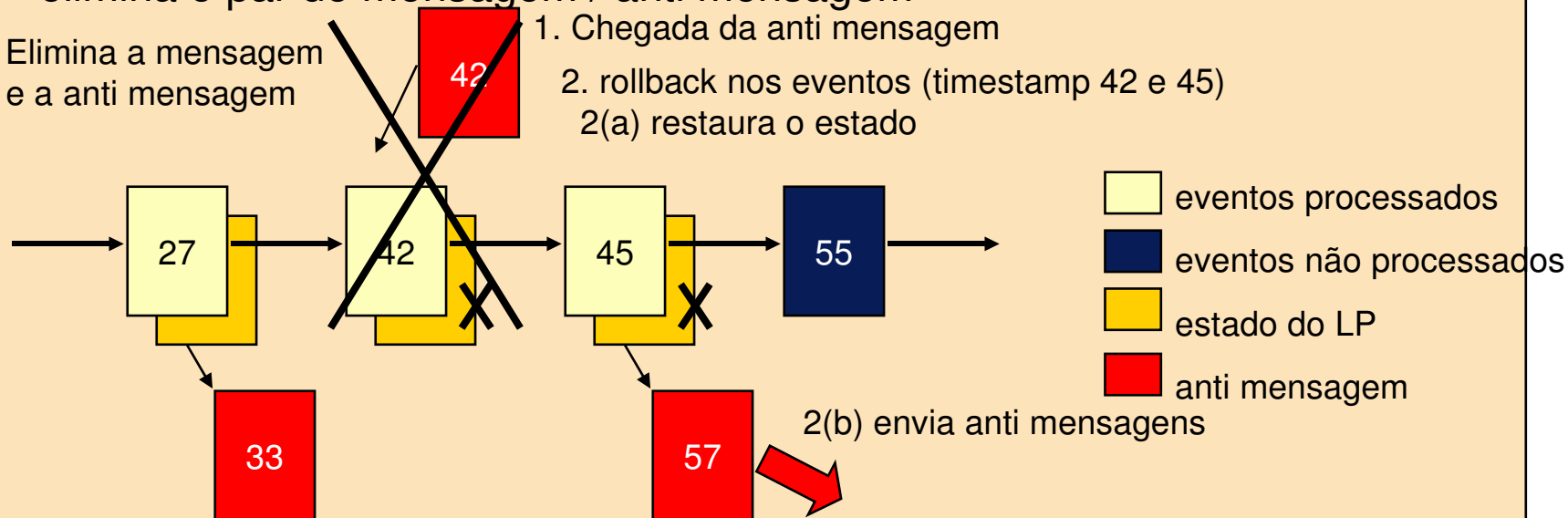


Caso I: Mensagem correspondente ainda não foi processada
elimina o par de mensagem / anti mensagem

Caso II: a mensagem correspondente já foi processada

- rollback para um tempo anterior ao processamento da mensagem
- elimina o par de mensagem / anti mensagem

3. Elimina a mensagem
e a anti mensagem



Pode causa uma “cascata” de rollbacks; recursivamente aplicada elimina todos os efeitos da computação errada

Caso III: mensagem correspondente ainda não foi recebida

- coloca a anti mensagem na fila
- elimina o par de mensagem / anti mensagem quando a mensagem chegar



GVT

- Um mecanismo é necessário para:
 - liberar recursos da memória (por exemplo, estados e eventos mais velhos)
 - Realizar operações de E/S
- Global Virtual Time (GVT)
 - time stamp mínimo de qualquer mensagem não processada (ou parcialmente processada) ou anti mensagem no sistema
 - GVT fornece uma marca no tempo para futuros rollbacks.
- Armazenamento dos eventos e vetores com tempos menores do que o GVT podem ser descartados
- OBSERVAÇÃO: tudo o que foi realizado até o GVT não pode sofrer rollback, o que garante um progresso contínuo.



Cálculo do GVT

- Cálculo do GVT Síncrono vs. Assíncrono
 - **Algoritmos para cálculo Síncrono** : Os LPs param o processamento dos eventos toda vez que o processamento para cálculo do GVT foi detectado
 - **Algoritmos para cálculo Assíncrono**: OS LPs podem continuar o processamento dos eventos e escalonar o novos eventos enquanto o processamento do cálculo do GVT é executado em *background*



ASDA

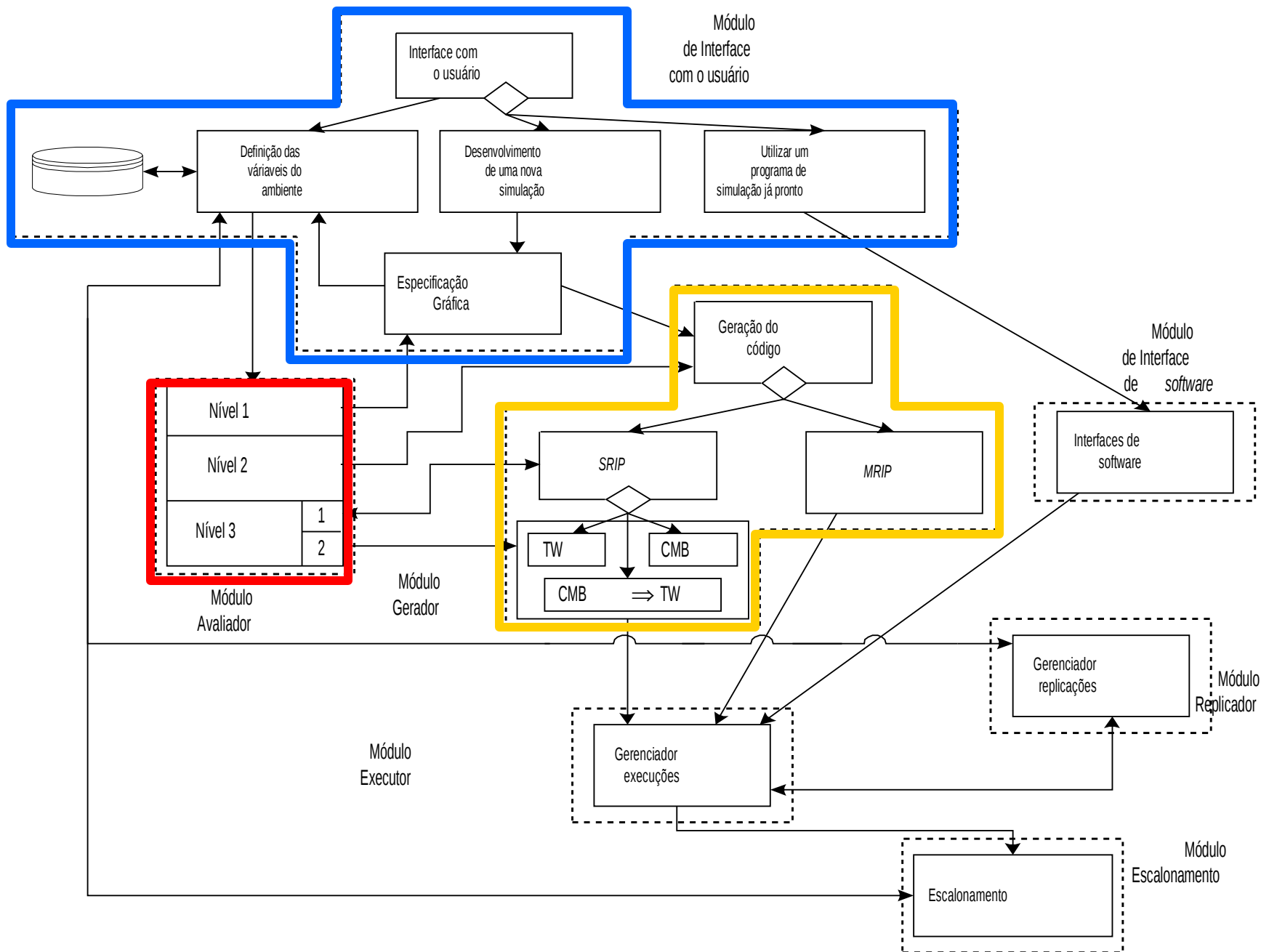
- Ambiente de simulação voltado à avaliação de desempenho de sistemas computacionais
- Oferece ao usuário um ambiente automático para o desenvolvimento de simulação

The logo graphic consists of three overlapping squares: a yellow one at the top left, a red one at the bottom left, and a blue one at the bottom right. A black crosshair is superimposed on these squares, with the vertical line passing through the center of the yellow and red squares, and the horizontal line passing through the center of the red and blue squares.

ASDA

- Ambiente de fácil aprendizagem e utilização
- Interface amigável
- Geração e compilação automática do código da simulação
- Execução e análise dos resultados

ASDA - Ambiente de Simulação Distribuída Automático





Mecanismo de Troca de Protocolos

❖ **Objetivo:** Promover, após a detecção do baixo desempenho, a troca de protocolos CMB ↔ TW

Como ?

▶ Cada protocolo utiliza um conjunto de estruturas de dados e variáveis com características diferentes

Quando ?

▶ Qual o momento certo para realizar a troca



Mecanismo de Troca de Protocolos

**Avaliação de Desempenho em
tempo de Execução do Protocolo de
Simulação Distribuída CMB**

Célia et al

Wperformance – SBC 2006



Mecanismo de Troca de Protocolos

Atividades a serem desenvolvidas:

2. Ler o artigo
3. Considerando todas as desvantagens da simulação distribuída em relação a seqüencial, dê a sua opinião sobre o porque ainda se pesquisa e se utiliza simulação distribuída
4. Compare, em linhas gerais, simulação distribuída otimista e conservadora
5. Na página 14 do artigo, chega-se a três conclusões. É possível relacionar essas conclusões com os dados apresentados no artigo? Discuta.