

# Sistemas Distribuidos

## Grado en Ingeniería Informática

### Sincronización Sistemas Distribuidos II

Departamento de Ingeniería Informática  
Universidad de Cádiz



Curso 2017 – 2018

# Indice

- 1 Estados Globales
- 2 Depuración distribuida



# Sección 1 | Estados Globales



UCA

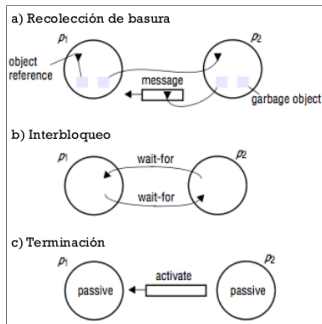
Universidad  
Católica Argentina

# Estados Globales

## Introducción

- Hay tareas para las que necesitamos conocer el estado global del sistema:

- 1 **Recolección de basura:** Detección de objetos distribuidos que ya no se utilizan
- 2 **Detección de Interbloqueos:** Un interbloqueo distribuido ocurre cuando dos procesos esperan mensaje del otro
- 3 **Detección de estados de terminación:** Detectar la terminación de un algoritmo distribuido

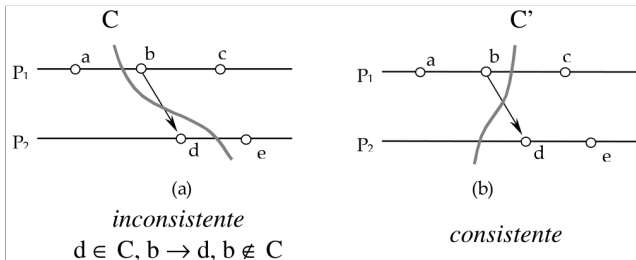


- Es vital tener en cuenta el estado de los procesos y del canal de comunicación

# Estados Globales

## Cortes consistentes

- Un corte **C** es consistente si, para cada suceso que contiene, también contiene todos los sucesos que “**sucedieron antes que**”



- Estado global consistente: **Aqué** que corresponde con un corte consistente

# Estados Globales

## Evaluación de cortes con relojes vectoriales

- Para saber si un corte es consistente, nos podemos basar en los vectores de tiempos:

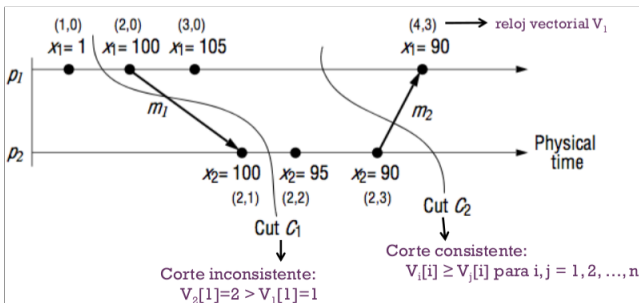
$$\forall i, j : V_i[i](e_i^{ci}) \geq V_j[i](e_j^{cj})$$

- Puesto que cada proceso posee una visión parcial del sistema, para construir un corte consistente (y obtener de paso su estado global asociado) los procesos deben ejecutar un algoritmo distribuido
- Utilidad: detección de interbloqueos, establecimiento de puntos de recuperación de un sistema, finalización distribuida

# Estados Globales

## Evaluación de cortes con relojes vectoriales - EJEMPLO

- Un corte es consistente si, para cada proceso  $P_i$ , su reloj lógico en ese momento es mayor o igual que todos los registros del valor del reloj de  $P_i$  mantenidos por otros procesos



# Estados Globales

Algoritmo de instantánea de Snapshot) de Chandy y Lamport

- **Objetivo** Obtener un conjunto de estados de proceso y del canal de comunicación (instantánea) que sea un estado global consistente
- **Suposiciones**
  - Los canales y procesos no fallan: todos los mensajes se reciben correctamente, y una única vez
  - Los canales son unidireccionales con entrega tipo FIFO
  - Hay canal de comunicación directo entre todos los procesos
  - Cualquier proceso puede tomar una instantánea en cualquier momento
  - Los procesos pueden continuar su ejecución y comunicación mientras se está tomando una instantánea



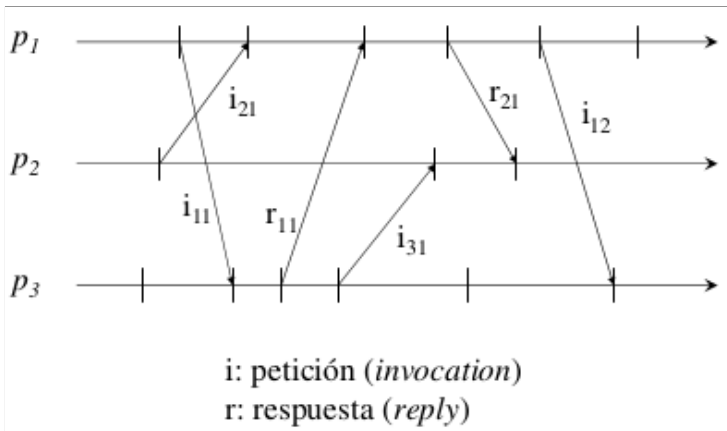
# Estados Globales

## Evaluación de cortes con relojes vectoriales - EJEMPLO

- Regla de recepción de instantánea (mark) en  $P_i$  por el canal  $c$ 
  - si ( $P_i$  no ha registrado su estado todavía)
    - registra su estado de proceso
    - registra el estado de  $c$  como vacío
    - activa el registro de mensajes que lleguen por otros canales
  - si no
    - $P_i$  registra el estado de  $c$  como el conjunto de mensajes recibidos en  $c$  desde que guardó su estado (mensajes posteriores a la instantánea)
- Regla de envío de instantánea por  $P_i$ 
  - Tras registrar su estado, para cada canal de salida  $c$   $P_i$  envía un mensaje de instantánea por el canal  $c$

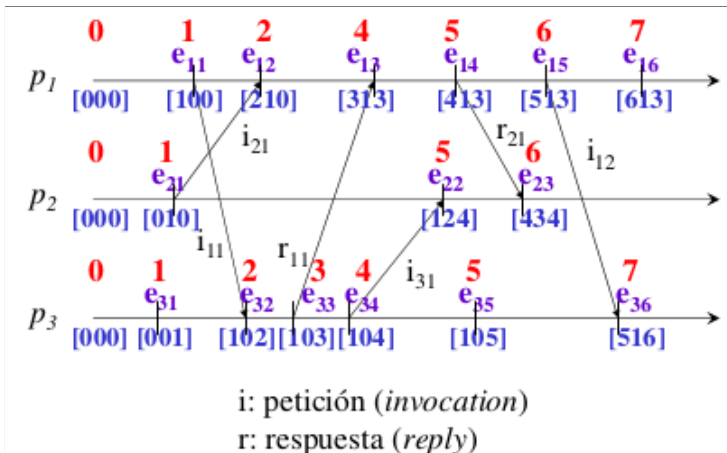
# Estados Globales

## Ejemplo de Snapshot



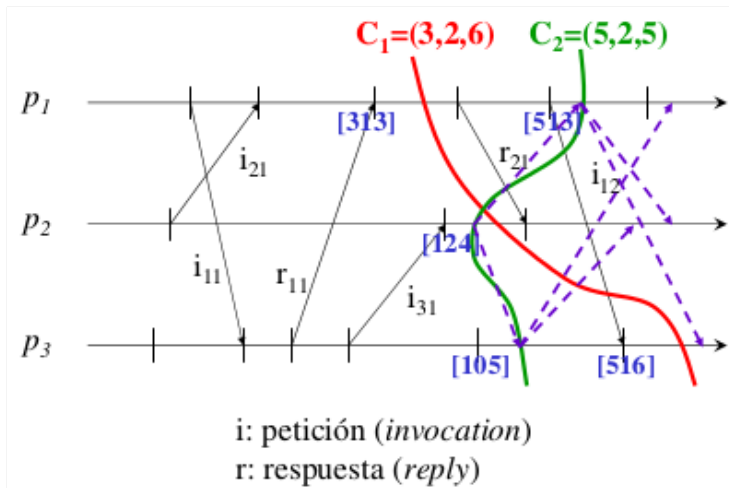
# Estados Globales

## Ejemplo de Snapshot



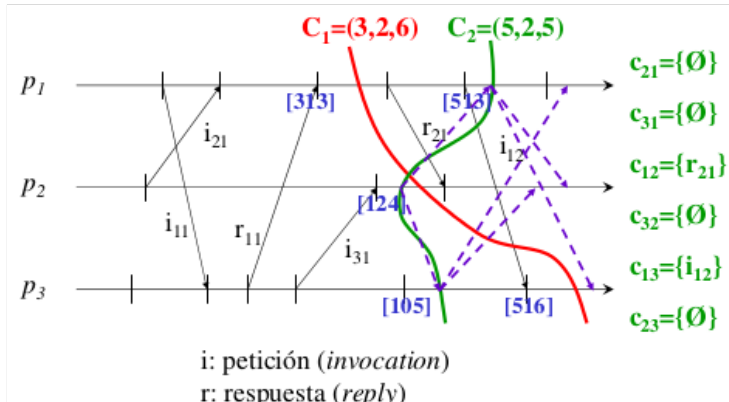
# Estados Globales

## Ejemplo de Snapshot



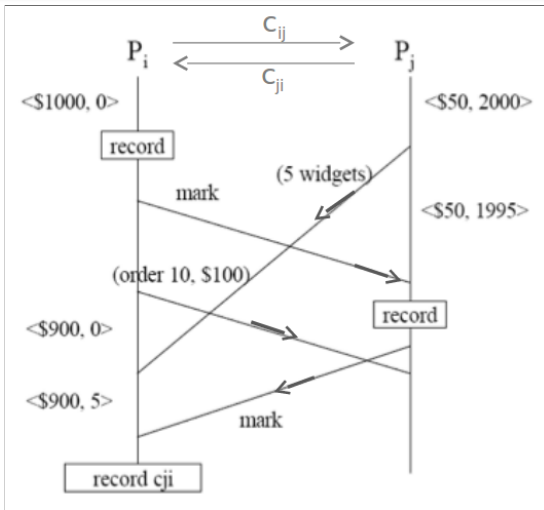
# Estados Globales

## Ejemplo de Snapshot



# Estados Globales

## Otro ejemplo de Snapshot



estado registrado:  $P_i = \langle \$1000, 0 \rangle$ ,  $P_j = \langle \$50, 1995 \rangle$ ,  $c_{ij} = \langle \rangle$ ,  $c_{ji} = \langle 5 \rangle$

## Sección 2 | Depuración distribuida



UCA

Universidad  
Católica Argentina

# Depuración distribuida

## Predicados

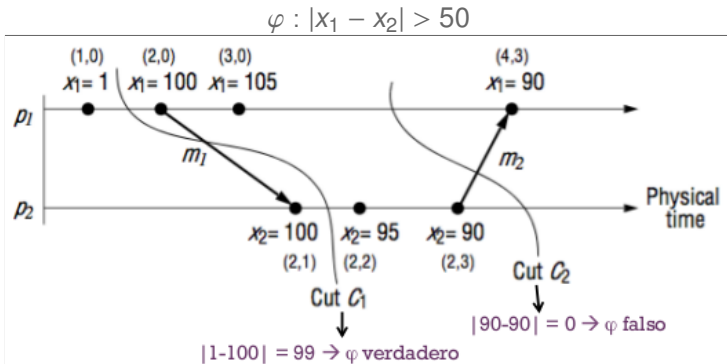
- La ejecución de un SD se puede caracterizar (y depurar) por las transiciones entre estados globales consistentes  
 $S_0 \rightarrow S_1 \rightarrow S_2 \rightarrow S_n$
- Un **predicado** de estado global es una función
  - Determinar una condición del SD equivale a evaluar su predicado
- Características posibles de un predicado
  - **Estabilidad:** el valor del predicado no varía con los nuevos sucesos (por ejemplo, en el caso de interbloqueo o terminación)
  - **Seguridad:** el predicado tiene valor falso para cualquier estado alcanzable desde  $S_0$  (deseable para errores)
  - **Veracidad:** el predicado tiene valor verdadero para algún estado alcanzable desde  $S_0$  (deseables para situaciones necesarias)



# Depuración distribuida

## Predicados:Ejemplo

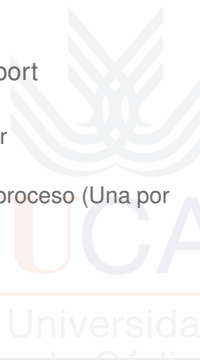
- Imaginemos un sistema de 2 procesos donde queremos controlar el predicado



# Depuración distribuida

## Monitorización

- Depurar un SD requiere registrar su estado global, para poder hacer evaluaciones de predicados en dichos estados
  - Generalmente, la evaluación trata de determinar si el predicado  $\varphi$  cumple con la condición “posiblemente” o “sin duda alguna”.
- Monitorización del estado global:
  - Distribuido: algoritmo de instantánea de Chandy y Lamport
  - Centralizado: algoritmo de Marzullo y Neiger
    - Los procesos envían su estado inicial al proceso monitor
    - Periódicamente, le vuelven a enviar su estado
    - El monitor registra los mensajes de estado en colas de proceso (Una por proceso)



# Depuración distribuida

## Evaluación de predicados

### ■ Objetivo de la monitorización

- Determinar si un predicado  $\varphi$  es “posiblemente” o “sin duda alguna” verdadero en un determinado punto de la ejecución.
- El proceso monitor sólo registra los estados globales consistentes
  - Los únicos en que podemos evaluar el predicado con certeza

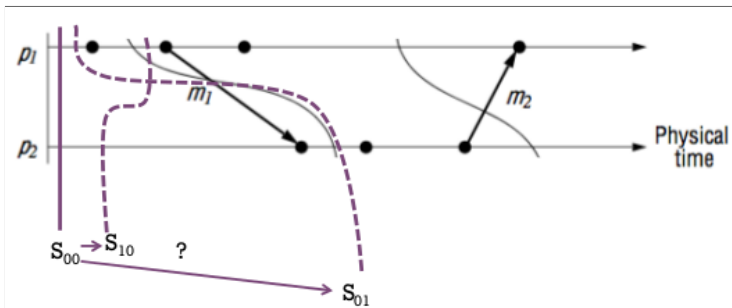
### ■ Monitorización del estado global:

- Teniendo en cuenta el predicado a evaluar, podemos reducir el tráfico de mensajes de estado
  - Tamaño: el predicado puede depender sólo de ciertas partes del estado de un proceso → no es necesario mandar el estado completo
  - Número: el cambio de valor del predicado sólo ocurre en algunos casos → sólo hay que recoger los estados en cambios relevantes

# Depuración distribuida

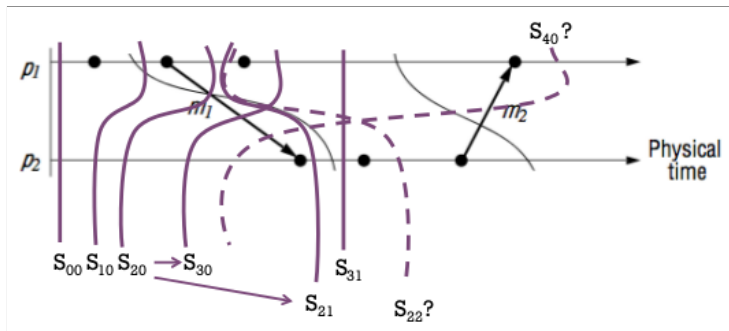
## Red de estados globales

- Mediante la monitorización podemos construir una red de estados globales consistentes
  - $S_{ij}$  = estado global tras  $i$  eventos en el proceso 1 y  $j$  eventos en el proceso 2



# Depuración distribuida

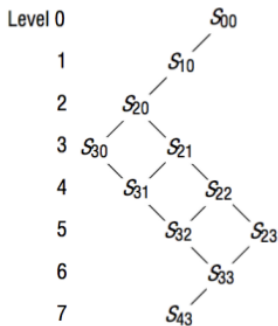
Red de estados globales: ejemplo



# Depuración distribuida

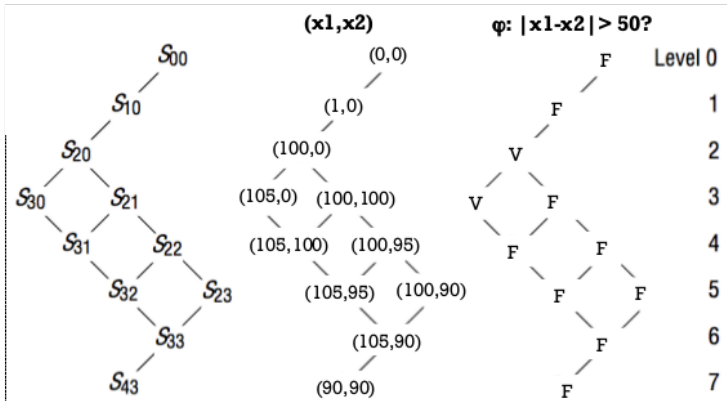
## Red de estados globales

- **Linealización:** ruta entre estados
- **Posiblemente  $\varphi$ :** existe un estado consistente  $S$  a través del que pasa una linealización tal que  $\varphi(S) = \text{Verdadero}$
- **Sin duda alguna  $\varphi$ :** existe un conjunto de estados consistentes  $S^*$  a través del que pasan todas las linealizaciones, tal que, para todo  $S$  en  $S^*$ ,  $\varphi(S) = \text{Verdadero}$



# Depuración distribuida

## Evaluación Instantánea de predicados



# Depuración distribuida

## Evaluación de predicados posiblemente

- Recorremos los estados alcanzables de cada estado inicial
  - Hasta que en algún momento alguno de los estados cumpla que  $\varphi(S_i)$  = Verdadero, o terminamos de recorrer la red.

```
Evaluar posiblemente  $\varphi$  para la red H de N procesos
L=0;    //Nivel de la red de estados
Estados={( $s^0_1, s^0_2 \dots s^0_N$ )}; //Estados del nivel L
mientras (  $\varphi(s_i)$  = Falso para todos los  $s_i$  en Estados)
    L=L+1;
    Alcanzable = {  $S'$  tal que  $S'$  es alcanzable en H desde
        algún S en Estados y nivel( $S'$ ) = L };
    Estados = Alcanzable;
fin mientras
si L <= {nivel máximo de H} salida "posiblemente  $\varphi$ ";
```



# Depuración distribuida

Evaluación de predicados sin duda alguna

- Recorremos los estados alcanzables de cada estado inicial
  - Hasta que en algún momento **todos** los estados cumplan con el predicado  $\varphi(S_i) = \text{Verdadero}$ , o terminamos de recorrer la red.

```
Evaluar sin duda alguna  $\varphi$  para la red H de N procesos
L=0;
si ( $\varphi(s^0_1, s^0_2 \dots s^0_N)$ )  Estados={};
si no                        Estados={{ $s^0_1, s^0_2 \dots s^0_N$ }};
mientras (Estados != {})
    L=L+1;
    Alcanzable = { S' tal que S' es alcanzable en H desde
    algún S en Estados y nivel(S') = L };
    Estados = {S en Alcanzable con  $\varphi(S)=\text{Falso}$ };
fin mientras
salida "sin duda alguna  $\varphi$ ";
```

# Depuración distribuida

## Resumen

Consiste en

- Determinar el predicado que queremos evaluar
- Especificar un método para construir una red o historia de estados globales consistentes
  - Teniendo en cuenta el predicado para optimizar tráfico
- Evaluar si nuestro predicado se cumple en algún momento
  - **Si es posible**, se cumplirá para algunas linealizaciones
  - **Si es sin duda**, se cumplirá para todas las linealizaciones

