

# **Introducción a los Sistemas Distribuidos**

**30221 - Sistemas Distribuidos**

**Rafael Tolosana Calasanz**

**Dpto. Informática e Ing. de Sistemas**

## Bibliografía

- G. Colouris, J. Dollimore, T. Kindberg and G. Blair.  
Distributed systems: Concepts and Design. 5th Edition.  
Addison-Wesley. May, 2011. ISBN: 978-0132143011.

# Motivación

# Motivación

## Programación Concurrente

- ¿Qué objetivo se persigue?



1

<sup>1</sup>Title: "Space Shuttle Program": San Diego Air and Space Museum Archive

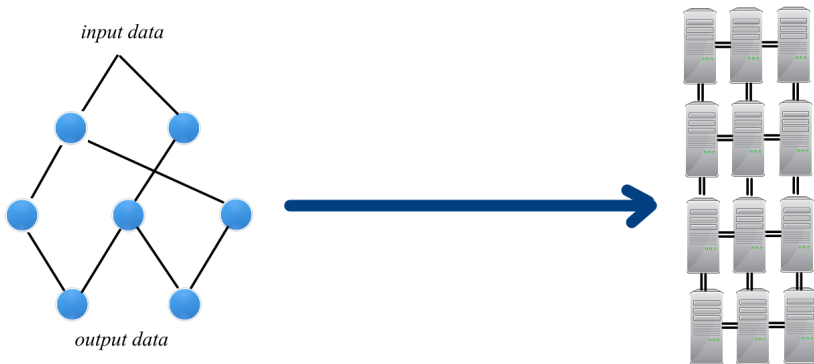
# Motivación

## Programación Concurrente



# Motivación

## Programación Distribuida



# Motivación

## Perspectiva Histórica

- La evolución de la programación concurrente no puede entenderse sin el estudio del desarrollo tecnológico del hardware.
- P. Brinch Hansen: “In the late 1970s, parallel computing was moving from multiprocessors with shared memory towards multicomputers with distributed memory.” <sup>2</sup>

---

<sup>2</sup>The Invention of Concurrent Programming: In The Origin of Concurrent Programming: From Semaphores to Remote Procedure Calls, P. Brinch Hansen, Ed., Springer-Verlag, New York (2002)

# Definición SD



# Definición SD

## Definición: Un Sistema Distribuido

- **Un Sistema Distribuido** es aquel en que sus componentes, localizados en computadores interconectados mediante una red de interconexión, se comunican y coordinan exclusivamente mediante el paso de mensajes.

### *Compartición de Recursos*

- Quizá la motivación principal para construir un SD nace de la motivación para compartir recursos computacionales (reducir costes económicos):
  - CPU
  - almacenamiento
  - red de comunicación

# Características Fundamentales

## Ausencia Memoria Compartida

- **Concurrencia:** de memoria compartida a paso de mensajes
  - ¿Cómo se resuelve el problema de la sección crítica?
  - Patrones Concurrencia en memoria compartida <sup>3</sup>
  - Concurrencia en Go <sup>4</sup>

---

<sup>3</sup>Downey, A. (2008). The little book of semaphores. Green Tea Press.

<sup>4</sup>Cox-Buday, K. (2017). Concurrency in Go: Tools and Techniques for Developers. " O'Reilly Media, Inc." .

# Características Fundamentales

## Ausencia Memoria Compartida

- **Concurrencia:** de memoria compartida a paso de mensajes
  - ¿Cómo se resuelve el problema de la sección crítica?
  - Patrones Concurrencia en memoria compartida <sup>3</sup>
  - Concurrencia en Go <sup>4</sup>
- **Paso de mensajes**
  - Dos operaciones: *send* y *receive*
  - Canal síncrono vs asíncrono (send bloqueante vs no)
  - ¿Cómo se resuelve el problema de la sección crítica con paso de mensajes?

---

<sup>3</sup>Downey, A. (2008). The little book of semaphores. Green Tea Press.

<sup>4</sup>Cox-Buday, K. (2017). Concurrency in Go: Tools and Techniques for Developers. " O'Reilly Media, Inc." .

# Características Fundamentales

## Problema de la Sección Crítica

# Características Fundamentales

## Problema de la Sección Crítica

**Problema:** Resolver el patrón barrera mediante paso de mensajes

- Opción 1: partir de la versión de memoria compartida y realizar las transformaciones oportunas
- Opción 2: alternativa que se fundamenta en la semántica de las operaciones send y receive, sin considerar el diseño de memoria compartida.

# Características Fundamentales

## Ausencia de un Reloj Global

- Cuando los procesos distribuidos tienen que coordinarse, lo hacen mediante el intercambio de mensajes. A menudo la coordinación se fundamenta mediante la ordenación temporal de las acciones. Pero tecnológicamente los relojes de las máquinas no van a estar sincronizados.
- No hay un reloj global, como consecuencia directa de la ausencia de un SD

# Características Fundamentales

## Fallos Parciales Posibles

- En un programa secuencial (o incluso en uno concurrente con memoria compartida), muy habitualmente los fallos / averías que se producen conllevan a que todo el programa falle.
- En un SD, puede haber fallos de una parte no total de los procesos distribuidos.
- También puede haber fallos en la red de comunicación
- La separación geográfica del sistema puede tener muchas ventajas:
  - Un SD puede seguir funcionando aún con fallos parciales
  - La capacidad de incorporar mayor capacidad (recursos) computacionales cuando sea necesario

# Propiedades SSDD



# Propiedades SSDD

## Heterogeneidad

- Internet permite acceder a los usuarios a servicios y ejecutar aplicaciones en computadores distintos (heterogéneos)
- La heterogeneidad se manifiesta:
  - Red de comunicación
  - Hardware
  - Sistemas Operativos
  - Código fuente: Diferentes Lenguajes de Programación, bibliotecas, etc.

# Propiedades SSDD

## Openness

- Esta característica es aquella mediante la cual un sistema distribuido puede **extenderse** y reimplementarse de varias formas:
  - Por ejemplo, añadiendo una determinada funcionalidad
  - Sustituyendo un componente (trozo de código) por otro, etc
- Se consigue mediante:
  - La documentación del sistema y de sus interfaces
  - Cumplir con un estándar
  - Relacionado con el concepto de interoperabilidad

# Propiedades SSDD

## Seguridad

- Muchos de los SSDD tienen un valor importante para sus usuarios
- La seguridad es clave en los SSDD y tiene que garantizar 3 aspectos:
  - Confidencialidad: protección contra la revelación de información
  - Integridad: protección contra la manipulación no autorizada
  - Disponibilidad: protección contra una interferencia no deseada

# Propiedades SSDD

## Escalabilidad

- Es la capacidad que tiene un SD para ser capaz de operar ante un aumento significativo de la carga de trabajo, sin ver mermados los requisitos no funcionales
- En la actualidad esta propiedad se relaciona con la de elasticidad

# Propiedades SSDD

## Tolerancia a Fallos

- Los SSDD están inherentemente expuestos a fallos de diversa índole
- El sistema tiene que estar diseñado para ser capaz de operar en presencia de los fallos proporcionando la funcionalidad deseada
- Fases de Gestión de un fallo:
  - Detección → Identificación → Corrección

# Propiedades SSDD

## Concurrencia

- Los procesos distribuidos son inherentemente concurrentes
- Muchas veces las aplicaciones y los servicios distribuidos utilizan recursos computacionales que están compartidos
- Es necesario poner los medios y mecanismos necesarios para garantizar la corrección

# Propiedades SSDD

## Transparencia

- Es la propiedad mediante la cual un SD se percibe por un usuario como un todo y no como un conjunto de componentes
- Podemos ver la transparencia desde distintos puntos de vista:
  - Acceso transparente a recursos locales y remotos
  - Acceso independiente de:
    - la localización
    - la concurrencia (sincronización)
    - la ocurrencia de fallos
    - la carga de trabajo en el sistema

# Propiedades SSDD

## Calidad del Servicio (QoS)

- Hace referencia a los requisitos no funcionales del sistema
  - Cómo tiene que proporcionar el sistema la funcionalidad
- Se mide habitualmente mediante el tiempo de ejecución, el throughput, el consumo energético, el coste económico, etc.



# Resumen

# Resumen

- **Comunicación** mediante el paso de mensajes
  - Síncrona / Asíncrona
- **Noción** de Sistema Distribuido
- **Características** Sistemas Distribuidos
  - Ausencia Reloj Global
  - Fallos Parciales
  - Concurrencia
- **Propiedades** / Retos Sistemas Distribuidos

# **Introducción a los Sistemas Distribuidos**

**30221 - Sistemas Distribuidos**

**Rafael Tolosana Calasanz**

**Dpto. Informática e Ing. de Sistemas**