

Programación Distribuida y Tiempo Real

Características



Libro Van Steen-Tanenbaum

- Evolución
 - Redes ==> Sist. Operativos ==> Sist. Distribuidos
- Sitio web, materiales (ej: figuras)
<https://www.distributed-systems.net/index.php/books/ds3/>
- Pueden pedir una versión electrónica personal
 - De la 3ra ed., incluso corregida
 - De ediciones anteriores
- Diapositivas de clases de Van Steen



Características

- ¿Cómo describir?
 - Objetivos
 - Desafíos
 - “Lo que hay” o “Tendremos que tener en cuenta”
 - Características
- Algunas más importantes que otras
 - Varios aspectos, en particular para desarrollo



Características

- Sin estado global
 - Hardware (memoria, reloj, énfasis en la bibliografía)
 - Software: sistema operativo



Características

- Sin estado global
 - Hardware (memoria, reloj, énfasis en la bibliografía)
 - Software: sistema operativo
- Concurrencia y recursos compartidos (Desafío p/ Coulouris)



Características

- Sin estado global
 - Hardware (memoria, reloj, énfasis en la bibliografía)
 - Software: sistema operativo
- Concurrencia y recursos compartidos (Desafío p/ Coulouris)
- Escalabilidad (Objetivo para Tanenbaum, cap. 1)



Características

- Sin estado global
 - Hardware (memoria, reloj, énfasis en la bibliografía)
 - Software: sistema operativo
- Concurrencia y recursos compartidos (Desafío p/ Coulouris)
- Escalabilidad (Objetivo para Tanenbaum, cap. 1)
- ¿? Seguridad (Desafío para Coulouris)



Características

- Sin estado global
 - Hardware (memoria, reloj, énfasis en la bibliografía)
 - Software: sistema operativo
- Concurrencia y recursos compartidos (Desafío p/ Coulouris)
- Escalabilidad (Objetivo para Tanenbaum, cap. 1)
- ¿? Seguridad (Desafío para Coulouris)
- Apertura-Extensibilidad (Objetivo para Tanenbaum, Desafío para Coulouris) relación con otra característica:
 - Heterogeneidad (Desafío para Coulouris)



Características

- Sin estado global
 - Hardware (memoria, reloj, énfasis en la bibliografía)
 - Software: sistema operativo
- Concurrencia y recursos compartidos (Desafío p/ Coulouris)
- Escalabilidad (Objetivo para Tanenbaum, cap. 1)
- ¿? Seguridad (Desafío para Coulouris)
- Apertura-Extensibilidad (Objetivo para Tanenbaum, Desafío para Coulouris) relación con otra característica:
 - Heterogeneidad (Desafío para Coulouris)
- Transparencia
 - Qué se puede ocultar y qué se debe ver
 - En detalle a continuación



“Transparencies”

Instructor's Guide for Coulouris, Dollimore and Kindberg Distributed Systems: Concepts and Design Edn. 4
© Pearson Education 2005

Access transparency: enables local and remote resources to be accessed using identical operations.

Location transparency: enables resources to be accessed without knowledge of their physical or network location (for example, which building or IP address).

Concurrency transparency: enables several processes to operate concurrently using shared resources without interference between them.

Replication transparency: enables multiple instances of resources to be used to increase reliability and performance without knowledge of the replicas by users or application programmers.

Failure transparency: enables the concealment of faults, allowing users and application programs to complete their tasks despite the failure of hardware or software components.

Mobility transparency: allows the movement of resources and clients within a system without affecting the operation of users or programs.

Performance transparency: allows the system to be reconfigured to improve performance as loads vary.

Scaling transparency: allows the system and applications to expand in scale without change to the system structure or the application algorithms.



“Transparencies”

Instructor's Guide for Coulouris, Dollimore and Kindberg Distributed Systems: Concepts and Design Edn. 4
© Pearson Education 2005

Access transparency: enables local and remote resources to be accessed using identical operations.

Location transparency: enables resources to be accessed without knowledge of their physical or network location (for example, which building or IP address).

Concurrency transparency: enables several processes to operate concurrently using shared resources without interference between them.

Replication transparency: enables multiple instances of resources to be used to increase reliability and performance without knowledge of the replicas by users or application programmers.

Failure transparency: enables the concealment of faults, allowing users and application programs to complete their tasks despite the failure of hardware or software components.

Mobility transparency: allows the movement of resources and clients within a system without affecting the operation of users or programs.

Performance transparency: allows the system to be reconfigured to improve performance as loads vary.

Scaling transparency: allows the system and applications to expand in scale without change to the system structure or the application algorithms.



“Transparencies”

Instructor's Guide for Coulouris, Dollimore and Kindberg Distributed Systems: Concepts and Design Edn. 4
© Pearson Education 2005

Access transparency: enables local and remote resources to be accessed using identical operations.

Location transparency: enables resources to be accessed without knowledge of their physical or network location (for example, which building or IP address).

Concurrency transparency: enables several processes to operate concurrently using shared resources without interference between them.

Replication transparency: enables multiple instances of resources to be used to increase reliability and performance without knowledge of the replicas by users or application programmers.

Failure transparency: enables the concealment of faults, allowing users and application programs to complete their tasks despite the failure of hardware or software components.

Mobility transparency: allows the movement of resources and clients within a system without affecting the operation of users or programs.

Performance transparency: allows the system to be reconfigured to improve performance as loads vary.

Scaling transparency: allows the system and applications to expand in scale without change to the system structure or the application algorithms.



“Transparencies”

Instructor's Guide for Coulouris, Dollimore and Kindberg Distributed Systems: Concepts and Design Edn. 4
© Pearson Education 2005

Access transparency: enables local and remote resources to be accessed using identical operations.

Location transparency: enables resources to be accessed without knowledge of their physical or network location (for example, which building or IP address).

Concurrency transparency: enables several processes to operate concurrently using shared resources without interference between them.

Replication transparency: enables multiple instances of resources to be used to increase reliability and performance without knowledge of the replicas by users or application programmers.

Failure transparency: enables the concealment of faults, allowing users and application programs to complete their tasks despite the failure of hardware or software components.

Mobility transparency: allows the movement of resources and clients within a system without affecting the operation of users or programs.

Performance transparency: allows the system to be reconfigured to improve performance as loads vary.

Scaling transparency: allows the system and applications to expand in scale without change to the system structure or the application algorithms.



Dudas/Consultas

- Plataforma Ideas

