Introducción a los Sistemas Distribuidos

Motivación

Crecimiento de:

- Nivel de integración e interdependencia entre sistemas,
- Volúmenes de datos a procesar,
- Capacidades de cómputo,
- Paralelismo,
- Virtualización.

Definición

Sistema en el que el fallo de un computador que ni siquiera sabes que existe puede dejar tu propio computador inutilizable.

- Colección de computadoras -> **multiprogramación**
- Independientes -> **autónomos**
- Un sólo sistema -> **distribución transparente al usuario**
- Interconectadas por red -> **sistemas aislados NO SON distribuidos**
- Comunican y coordinan acciones -> **colaborativos**
- Intercambiando mensajes -> **protocolos de comunicación**
- Fallo de un computador -> **problemas no determinísticos**

Parámetros de diseño

- Escalabilidad
- Transparencia
- Tolerancia a Fallos *(Availability, Reliability, Safety, Maintainability)*
- Acceso a Recursos Compartidos
- Sistemas distribuidos abiertos *(Interfaces, Interoperability, Portability)*

Modelos de análisis

- Modelo de **Estados** *(interleaved model)*
- Modelo de **Eventos** *(happened before)*

Características

Centralizados

- **Control.**
- **Homogeneidad.** Estándares p/ software y hardware.
- **Consistencia.** Políticas fuertes.
- **Seguridad.** Menos superficie de ataque.

Distribuidos

- **Disponibilidad.** Tolerancia a Fallos.
- **Escalabilidad.**
- **Reducción de Latencia.** Localidad de recursos.
- **Colaboración.** Interacciones entre sistemas.
- **Movilidad.** No están atados al alcance de una única PC.
- **Costo.** Simplicidad. Delegación.

Descentralizar vs. Distribuir

- *Centralizar* = concentración de autoridad (nivel jerárquico + alto).
- *Descentralizar* = transferir toma de decisiones a eslabones inferiores.
- **Distribuir** = modelo descentralizado de control de computadoras para coordinar actividades con coherencia.

Ley de Conway

"Cualquier organización que diseñe un sistema, inevitablemente producirá un diseño cuya estructura será una **copia de la estructura de comunicación de la organización**."

- Diseñamos s/ lo que conocemos y estamos acostumbrados a hacer.
- Pueden ser soluciones eficientes.

Virtualización

- Necesidad de **independencia real** de recursos.
- Seguridad en los **accesos**.
- Concepto de **Hypervisor**:
  - Manager de VMs.
  - Emula *hardware capabilities*.
  - Administra recursos del `Host OS` -> `Guest OS`.
  - Mecanismos de Seguridad.

Docker

- Soporte de OS:
  - **Namespaces.** Aislamiento de recursos.
  - **Cgroups.** Seguridad.
  - **Union Mount.**
- Engine:
  - **Daemon.** Manejo de recursos.
  - **REST API.** Flexible, simple.
  - **CLI.** REST API a través de socket.