Sistemas Distribuidos I

Introducción

- **Definición**.
  - Características.
- Parámetros de diseño.
- **Ventajas** vs. Centralizado.
  - Disponibilidad.
  - Escalabilidad.
  - Reducción de Latencia.
  - Colaboración.
  - Movilidad.
  - Costo.
- Virtualización.
- Propiedades.
  - **Safety.**
  - **Liveness.**

Multi-programación

- Multithreading.
- **Multiprocessing.**
- Mecanismos de Sincronización.
  - **IPCs.**
- Paralelización de tareas.
  - **Camino crítico.**
  - Speedup.
    - Ley de Amdahl.
    - Ley de Gustafson.
  - **Work-Span.**
- MIMD.
  - Multiprocessors.
  - **Multicomputers.**

Middleware

- **Definicion(es).**
- Objetivos.
  - **Transparencia.**
- Clasificación.

Documentación

- Definición de **Arquitectura**.
- Diseño **evolutivo**.
- Modelos.
  - **Vistas 4 + 1.**
    - **Física.** Despliegue, Robustez.
    - Lógica. Clases, Estados.
    - Procesos. Actividades, Secuencia, Colaboración.
    - Desarrollo. Componentes, Paquetes.
    - Escenarios. Casos de uso.
  - C4 Model.

Arquitectura de Capas

- Interfaces.
- **Layers** (lógicas).
- **Tiers** (físicas).

Interfaces

- Exposición selectiva.
- **Contratos.**
  - Inter-Aplicaciones. **API**. Ecosistema de apps.
  - Intra-Aplicaciones.
- Modelado.
  - **Entidades.**
  - Procesos.

REST(ful)

- **Entidades.**
- Protocolos.
  - Comunicación: **HTTP**/S.
  - Serialización: JSON/XML.
- Estado -> mutable mediante CRUD.
- **Versionado.** Semver.

Nombres y direccionamiento

- Nombre. Identidad única.
  - Abstracción.
- **Mapeo.** Reutilizar direcciones cambiantes.

Grupos

- **Abstracción.**
- Colección de procesos.
- Difusión de mensajes.
- **Atomicidad.** Entrega de paquetes.

Tiempo

- Definición.
- Usos.
- **Relojes Físicos.**
  - Drift. Sincronización periódica.
  - **Network Time Protocol.** Stratums.
- **Relojes Lógicos.**
  - Conceptos.
    - Evento.
    - Estado.
    - **Ocurre antes.**
  - Definición.
  - **Algoritmo de Lamport.**
  - Vectores de relojes.

Sincronismo

- Definición.
- Tipos. **Timeout asociado.**
- Propiedades.
  - **Steadiness.**
  - **Tightness.**
- Órden.
  - Envío != delivery.
  - **Hold-back queue.**
- Historia y corte.
  - Corte **consistente**.
  - Algoritmo Chandy & Lamport. **Snapshots.**

Arquitecturas Distribuidas Simples

- Cliente-Servidor.
- Peer-to-Peer. **Colaboración.**
- **RPC.**
  - Transparencia.
  - Interfaces -> Portabilidad.
  - **IDL.**
  - **Implementación** (stub, communication module).
- **Distributed Objects.**
  - Middleware.
  - Estándares:
    - CORBA.
    - RMI.

Comunicación

MOM

- Definición.
  - **Transparencia.**
- Variantes. **Broker(less).**
  - BUS.
  - **Queues.**

Patrones

- Request-Reply.
- Publisher-Subscriber.
- **Pipeline.**
  - **DAG.**

Coordinación de Actividades

- Coordinación.
- Replicación.
- Acceso a Recursos Compartidos.
- Casos de estudio.
  - **OpenMPI.** Mensajes.
  - Apache.
    - **Flink.** Pipelines de datos.
      - Dataflow.

        * **Beam.** Portabilidad (lenguajes/runners).

Map Reduce

- Filosofía.
- **Master-Workers.** Caso ideal.
- **Chunks** de datos.
- Función.
  - **Map.** Intermediate k:v.
    * Agrupa por key.
  - **Reduce.** RPC.

Data Intensive Applications

- **Sistemas de gran escala.**
  - OLTP vs. OLAP.
  - Almacenamiento.
- **Replicación.**
  - Leader.
  - Multi-leader.
  - Leaderless.
- **Particionamiento.**
  - Horizontal vs. Vertical.
  - Función de partición.
  - **Enrutamiento.**

Distributed Shared Memory

- **Memoria compartida centralizada** (ilusión).
- Implementación. **Memory pages.**
  1. Server.
  2. Migración. *Leased.*
  3. Replicación RO.
  4. Replicación RW.

Distributed File Systems

- **Información persistente centralizada.**
- Factores de diseño.
- Casos de estudio.
  - **NFS.** Independencia de plataformas.
    * VFS, RPC.
    * POSIX.
  - **HDFS.** Hardware bajo costo.
    * GFS.
    * *"Moving computation is cheaper than moving data."*
    * Master-Salve. Namenode, Datanodes.
  - **Big Table.** Clave-Datos disperso.
    * Jerarquía tres niveles.
    * Tablets.
    * Auto-splitting.

Escalabilidad

- Definición. **Crecimiento.**

- Patrones de carga.
- Limitantes.
- **Técnicas.**

Elasticidad

- Definición. **Dinamismo.**
- Componentes.
  - App Load Balancer.
  - Autoscaler.
  - Monitoring Automático.

Arquitecturas orientadas a Servicio

- Monolíticas.
  - Escalables.
- **Service Oriented (SOA).**
  - BPM.
  - Tecnologías.
    - **Enterprise Service BUS.**
    - Service Repository & Discovery.
  - Servicios / Procesos.
    - Interfaces.
- Microservicios.
- Serverless.

Cloud

- Abstracción.
  - **IaaS, PaaS, SaaS.**
- Beneficios.
- Resistencia al cambio.

PaaS

- Definición.
- **Implicancias.**
- Google App Engine. Buenas prácticas.
  - **Servicios.**
  - **Instancias.** AppServers.
    - Dinámicas vs. Residentes.
    - Stateless.
  - Comunicación. **Push queues.**

Tolerancia a Fallos

- Definición.
- **Fallo -> Error -> Falla/Avería.**
- Clasificación.
  - Frecuencia.
  - Tipo.
- Condiciones.
- **Detección de errores.**

Conceptos

- Resiliencia.
- Degradación.
- Enmascarar (redundancia).
- Recuperación.
- Replicación.
  - **Tipos de Replicación.**

Consenso

- Definición. Acuerdo ante decisión.
- **Propiedades.**
  - Agreement.
  - Integrity.
  - Termination.
- **Exclusión mutua distribuida.**
  - Servidor central.
  - Token Ring.
  - Ricart & Agrawala.
- **Elección de líder.**
  - Ring.
  - Bully.
- **Generales Bizantinos.** `N >= 3f + 1`.
- **Paxos.**
  - Quorum: `N >= 2f + 1`.
  - Actores.
    - **Proposer.**
    - **Acceptor.**
    - **Learner.**

Tiempo Real

- Definición. **Requerimientos temporales.**
- Correctitud = valor y tiempo.
- Previsible.
- Tipos. **Hard-Soft.**
- Comunicación: **fiable y sincrónica**.
  - Deadlines.
  - *Profibus, Profinet.*

Sistemas de Control

- Definición.
- Nociones.
  - Control.
  - Proceso.
  - **Variable controlada.** *Output.*
  - **Variable manipulada.**
  - Perturbación.
  - Planta.
  - Controlador.
  - Actuador.
- Ciclos. **Lazo abierto vs. cerrado.**

Cheatsheet: NALSD

- Procedimiento.
  1. **Endpoints.**
  2. **Análisis de volumen.**
  3. **Diseño.**
- Entender:
  - Endpoints.
  - Boundaries (scope).
  - Datos.
- Assumptions.