

SISTEMAS DISTRIBUIDOS

RESÚMEN UNIDADES 1-4

LIBRO: FRANCISCO LOPEZ

Bibliografía

- **A.Tanenbaum, M.Van Steen.** *Distributed Systems, principles and paradigms.* Ed. Prentice Hall, 2002.
- **G.Coulouris, J.Dillmore, T.Kindberg.** *Distributed Systems - Concept and Design.* Addison-Wesley, Third Edition, 2001.
- **Francisco de Asís López Fuentes.** *Sistemas distribuidos - México : UAM, Unidad Cuajimalpa, c2015.*
- Búsqueda en la Web del estado del arte de los sistemas distribuidos.

Definiciones

- *“Un Sistema Distribuido es una colección de computadores independientes que aparecen ante los usuarios como un único sistema coherente.” (Andrew S. Tanenbaum)*
- *“Un sistema en el cual tanto los componentes de hardware y software de un computador conectados en red se comunican y coordinan mediante paso de mensajes.” (G.Coulouris, J.Dollimore, T.Kindberg)*
- *“Aquel que le impide a uno continuar su trabajo cuando falla un computador del cual uno nunca ha oído hablar.” (L.Lamport)*

Ejemplos de sistemas distribuidos

□ **Intranet Phoenix o Moodle ULS.**

- Espacio cuenta es visible desde todos los computadores. Cliente-Servidor

□ **World Wide Web (WWW).**

- Aparece antes los usuarios como un repositorio gigantesco de documentos. Archivos distribuídos.

□ **Factura electrónica en Chile (SII).**

- Muchos contribuyentes y compradores, Un ente centralizado. Procesamiento distribuído.

Ejemplos de sistemas distribuidos

- **Sistemas de almacenamiento distribuido.**
 - ▣ Hadoop Distributed File System (**HDFS**) y sistemas de almacenamiento en la nube como Amazon S3. alta escalabilidad.
- **Sistemas de procesamiento distribuido.**
 - ▣ **Apache Spark**, que permite realizar cálculos en paralelo en un clúster de máquinas. mejorar el rendimiento y la capacidad de procesamiento.
- **Sistemas de bases de datos distribuidas.**
 - ▣ **Apache Cassandra** base de datos distribuida altamente escalable, y Google Spanner.

Ejemplos de sistemas distribuidos

- **Sistemas de procesamiento de mensajes distribuidos.**
 - ▣ Apache Kafka y **RabbitMQ**, que proporcionan colas de mensajes distribuidas y capacidades de transmisión de datos.
- **Sistemas de computación en la nube.**
 - Cómputo, almacenamiento y red bajo demanda a través de Internet. Amazon Web Services (**AWS**), Microsoft Azure y Google Cloud Platform (GCP).
- **Sistemas de streaming distribuido.**
 - **Apache Kafka** Streams y Apache Flink, que brindan capacidades de procesamiento de streaming distribuido y análisis en tiempo real.

Ejemplos de sistemas distribuidos

- **Sistemas de búsqueda distribuida.**
 - ▣ búsqueda en múltiples nodos para permitir búsquedas rápidas y escalables. Ejemplo **Elasticsearch**.
- **Sistemas de computación distribuida en memoria.**
 - ▣ utilizan la memoria distribuida en múltiples nodos para acelerar el procesamiento y análisis de datos. **Apache Ignite** y **Redis Cluster**.
- **Sistemas de control de versiones distribuidos.**
 - ▣ Git: gestión y colaboración de desarrollo de software. **Github** y **Gitlab**.

Ejemplos de sistemas distribuidos

- **Sistemas de cómputo distribuido para IA.**
 - ▣ Entrenamiento de modelos y procesamiento de datos, en múltiples nodos para acelerar el tiempo de ejecución. **TensorFlow** y **PyTorch** (cómputo distribuido para el aprendizaje profundo y otras tareas de IA).
- **Sistemas de gestión de clústeres.**
 - ▣ Garantizan la disponibilidad, escalabilidad y rendimiento de las aplicaciones distribuidas. **Apache Mesos, Kubernetes** y **Docker Swarm**.
- **Sistemas de recomendación distribuidos.**
 - ▣ Recomendaciones personalizadas. Sistema de recomendación de películas de **Netflix** y el motor de recomendaciones de **Amazon**.

Ejemplos de sistemas distribuidos

- **Sistemas de procesamiento de transacciones distribuidas.**
 - ▣ Aseguran la consistencia y la integridad de los datos. protocolo de transacción distribuida (**DTP**), que garantiza la atomicidad, consistencia, aislamiento y durabilidad (ACID) en sistemas distribuidos.
- **Sistemas de coordinación y sincronización distribuida.**
 - ▣ Permiten la coordinación y sincronización de tareas y procesos en entornos distribuidos. **ZooKeeper** y **etcd**.
- **Sistemas distribuidos utilizados en IoT.**
 - ▣ AWS IoT Core, Azure IoT Hub, Google Cloud IoT Core, Zigbee y Z-Wave, LORAWAN, OMA Lightweight M2M (LwM2M) y el protocolo MQTT-SN.

Ejemplos de sistemas distribuidos

- **Sistemas de procesamiento de transacciones distribuidas.**
 - ▣ Aseguran la consistencia y la integridad de los datos. protocolo de transacción distribuida (**DTP**), que garantiza la atomicidad, consistencia, aislamiento y durabilidad (ACID) en sistemas distribuidos.
- **Sistemas de coordinación y sincronización distribuida.**
 - ▣ Permiten la coordinación y sincronización de tareas y procesos en entornos distribuidos. **ZooKeeper** y **etcd**.
- **Sistemas distribuidos utilizados en IoT.**
 - ▣ AWS IoT Core, Azure IoT Hub, Google Cloud IoT Core, Zigbee y Z-Wave, LORAWAN, OMA Lightweight M2M (LwM2M) y el protocolo MQTT-SN.

Ventajas de los sistemas distribuidos

- ❑ **Compartir recursos.**
- ❑ **Economía.**
 - ▣ Los nodos o componentes pueden compartir la carga y aprovechar los recursos de manera óptima. Esto puede resultar en un mejor uso de la capacidad de procesamiento, almacenamiento y ancho de banda.
- ❑ **Escalabilidad.**
 - ▣ Pueden crecer en términos de capacidad de procesamiento, almacenamiento y rendimiento al agregar nuevos nodos o recursos a la red.
- ❑ **Fiabilidad y disponibilidad.** Si un nodo o componente falla, otros pueden asumir su función y garantizar que los servicios estén disponibles
- ❑ **Disponibilidad.**
 - ▣ Si un nodo o componente falla, otros pueden asumir su función y garantizar que los servicios estén disponibles para los usuarios.

Desventajas de los sistemas distribuídos

- ❑ **Aumento de la complejidad.**
 - ▣ Costos de administración.
- ❑ **Fallas más frecuentes.**
 - ▣ Red, Latencia, pérdida de mensajes.
- ❑ **Interoperabilidad.**
 - ▣ Falta de estándares únicos.
- ❑ **Seguridad.**
 - ▣ debido a la naturaleza de la distribución de recursos y la comunicación entre componentes, Autenticación y autorización, distribución de claves, encriptación, etc.

Implicancias de LOS S.D.

× **Concurrencia.**

- + Recursos compartidos. Acceso concurrente.
- + Sincronización.

× ***Sin tiempo global.***

- + Cada computador tiene su propio reloj local.
- + Coordinación.

× ***Fallas independientes.***

- + Posibilidad permanente de fallas en cada componente.
- + Detección de fallas y tolerancia a fallas.

Desafíos de los Sistemas Distribuidos

- ❑ Compartir Recursos
- ❑ Transparencia
- ❑ Usabilidad
- ❑ Escalabilidad
- ❑ **Hardware**
- ❑ **Software**

Hardware

□ Se subdividen en dos categorías:

+ Homogéneos

- La arquitectura y memoria es igual o similar en todos los nodos
- Generalmente conectados a través de una única, usualmente de alto rendimiento, interfaz de red.
- Ejemplo: **CLUSTER**

+ Heterogéneos

- Múltiples arquitecturas
- Diferentes comunicaciones
- Diferentes formas de comunicación
- Ejemplo: **GRID.**

Hardware

- ❑ **Sistemas Operativos Monolíticos.**

- ❑ Múltiples procesos compartiendo recursos sobre una máquina.

- ❑ **Sistemas Operativos Multiprocesador.**

- ❑ Múltiples procesadores compartiendo datos a través de memoria compartida.
- ❑ Primitivas de sincronización: semáforos, locks, monitores.

- ❑ **Sistemas Operativos Multicomputador.**

- ❑ Múltiples computadores conectados por red.
- ❑ Sincronización basada en semánticas particulares de paso de mensajes

Paralelismo v/s Concurrency

- El **paralelismo** busca la ejecución real simultánea de tareas en diferentes recursos de procesamiento, mientras que la **concurrency** se refiere a la capacidad de manejar múltiples tareas de forma intercalada, ya sea ejecutándose al mismo tiempo o no.



Software

- **Se encuentran tres variantes de plataformas de software para sistemas distribuidos.**
 - ▣ Sistemas operativos de Red.
 - ▣ Sistemas operativos distribuidos.
 - ▣ Middleware.

Sistemas Operativos de Red

- Los sistemas operativos de red proporcionan un entorno unificado y coordinado para administrar recursos y servicios distribuidos en esa red. A diferencia de un sistema operativo centralizado, donde un solo sistema operativo controla todos los recursos de una máquina, en un sistema operativo distribuido de red, múltiples sistemas operativos trabajan juntos para coordinar y gestionar los recursos de la red de manera eficiente.

- Ejemplos:
 - Novell Netware, indows NT Server, UNIX, Ubuntu Server, CentOS, Debian, FREBSD, Cisco IOS, Junos OS, macOS Server, etc.

- Servicios distribuídos:
 - DNS, DHCP, Proxy, Active directory, Kerberos, FTP, RADIUS, HTTPS, etc.

Middleware

Software que actúa como intermediario entre diferentes aplicaciones, sistemas o componentes de software, facilitando la comunicación y la interoperabilidad entre ellos. Es decir es una API.

- ▣ Se ejecuta sobre el sistema operativo local de cada computador.
- ▣ Abstrae la complejidad y heterogeneidad de los computadores del sistema.

Ejemplos:

- ▣ **Mensajería:** Comunicación asíncrona y fiable entre aplicaciones distribuidas. Apache Kafka, RabbitMQ y ActiveMQ.
- ▣ **Integración empresarial (EAI):** Apache Camel, Mule ESB y IBM Integration Bus.
- ▣ **Servicios web:** Apache Axis, Microsoft WCF y Spring Web Services.
- ▣ **Transacciones:** Java Transaction API (JTA) y Microsoft Distributed Transaction Coordinator (DTC).
- ▣ **Caché:** Mejora el rendimiento. Memcached, Redis y Apache Ignite.
- ▣ **Seguridad:** Autenticación, autorización y cifrado. Apache Shiro, Spring Security y Microsoft Active Directory.
- ▣ **Gestión de identidad:** Keycloak, Okta y Azure Active Directory.
- ▣ **Otros:** ODBC, Tomcat, APIgee, WhatsApp, ZOOM, etc.

Tabla Comparativa

Desafios de los S.D.	S.O. Distribuido		S.O. de Red	MiddleWare
	Multi-proc.	Multi-comp.		
Transparencia	Muy Alto	Alto	Bajo	Alto
Mismo S.O.	Sí	Sí	No	No
Copias del S.O.	1	N	N	N
Comunicación	Memoria compartida	Mensajes	Archivos	Específico al modelo
Manejo de recursos	Global, central	Global, distribuido	Por nodo	Por nodo
Escalabilidad	No	Moderado	Si	Variable
Usabilidad	Cerrada	Cerrada	Abierta	Abierta