

Sesión docente: Introducción a los sistemas distribuidos

Jesús Peña Rodríguez

Universidad Industrial de Santander, 2024

Objetivo de enseñanza

Esta sesión tiene como objetivo introducir los conceptos básicos de los sistemas distribuidos, los desafíos de su implementación y su clasificación.

Pregunta orientadora

¿Qué son los sistemas distribuidos, cuáles son sus ventajas y desventajas, qué desafíos implica su implementación y cómo se clasifican?

Resultado de aprendizaje

El alumno conoce los conceptos básicos de los sistemas distribuidos, describe los principales desafíos de su implementación e identifica su clasificación.

Metodología

Para el desarrollo de esta sesión formativa se plantea la siguiente metodología: Se inicia impartiendo la clase: Introducción a los Sistemas Distribuidos. En esta se cubrirá la definición, ventajas, desventajas, desafíos y clasificación de los sistemas distribuidos. Luego, se desarrolla la actividad de clase donde el estudiante aplica los conocimientos adquiridos. El estudiante podrá resolver sus dudas durante el desarrollo de la clase y de la actividad con ayuda del docente. Finalmente, se

proporcionarán las referencias bibliográficas y material de apoyo para profundizar sobre el tema visto en clase.

Unidades temáticas

1. Sistemas Distribuidos
 - a. Definición
 - b. Ventajas y desventajas
 - c. Desafíos
 - d. Clasificación

Actividades de aula

El propósito de la actividad es introducir a los estudiantes en los conceptos básicos de los sistemas distribuidos, como la comunicación entre nodos, la coordinación y la tolerancia a fallos. En este caso se usarán **Jupyter Notebooks** y el módulo **socket** de Python.

Para desarrollar exitosamente la actividad se debe tener en cuenta:

1. Participar de la clase donde el docente presenta la base teórica de los sistemas distribuidos, sus características y desafíos.
2. Estudiar los recursos educativos y material de apoyo propuesto para la actividad. En este caso se proporciona un repositorio de Github con el material necesario https://github.com/JesusPenha/Distributed_Systems.git.
3. Desarrollar la actividad de implementación de una comunicación simple cliente-servidor usando **Jupyter Notebooks** y el módulo **socket** de Python.

El estudiante debe entregar la actividad al final de clase. La actividad se evalúa teniendo en cuenta la matriz de evaluación dispuesta en la sesión 1.4.3. Después de la actividad, se llevará a cabo una discusión en clase para reflexionar sobre los conceptos aprendidos y las lecciones extraídas de la experiencia.

Esta actividad ayuda a los estudiantes a reforzar los conceptos teóricos de los sistemas distribuidos y brinda experiencia práctica en el diseño, e implementación de sistemas distribuidos simples.

1. Sistemas distribuidos

Antes de definir los Sistemas Distribuidos, debemos hacer una pequeña reseña histórica de como surgieron. La era de la computación moderna empezó en 1945 con el proyecto ENIAC (Computador e Integrador Numérico Electrónico), entre 1943 y 1945 en la escuela Moore. Los primeros computadores electrónicos eran enormes, costosos y carecían de la capacidad de conectarse entre si. No fue hasta 1980, con la introducción de microprocesadores y la invención de las LANs (Local-area networks, por sus siglas en inglés), que fue posible la implementación de pequeñas redes de comunicación entre computadores en algunos lugares de trabajo. De esta manera surgen las primeras arquitecturas de conexión entre computadores.

Se tienen tres diferentes arquitecturas de conexión entre computadores. Los sistemas centralizados, los descentralizados y los distribuidos como se muestra en la Figura 1.

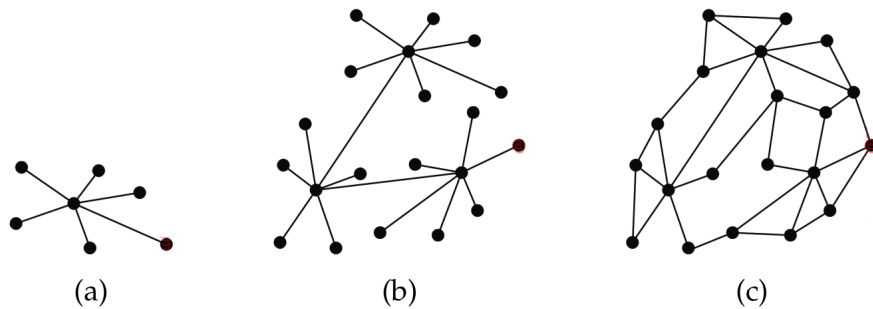


Figura 1: Organización de un sistema a) centralizado, b) descentralizado y c) distribuido.

Un sistema centralizado es una red de computadoras en la cual todos los recursos y procesos se ubican en una única computadora. Un sistema descentralizado es una red de computadoras interconectadas en la cual procesos y los recursos son necesariamente dispersos en varias computadoras [Tanenbaum and Van Steen, 2023]. Un sistema distribuido puede ser definido de diferentes maneras:

Leslie Lamport establece que “un sistema distribuido es aquel en el que el fallo de un computador que ni siquiera sabías que existía puede inutilizar tu propio computador” [Ghosh and Ghosh, 2023].

Ghosh propone que “un sistema distribuido consiste de varios elementos de cómputo independientes, geográficamente dispersos y conectados como compu-

tadores, teléfonos inteligentes, sensores, actuadores y dispositivos electrónicos embebidos. Esos dispositivos se comunican a través de mensajes para coordinar y cooperar en el desarrollo de un objetivo computacional común, a pesar de los fallos ocasionales de enlaces o dispositivos”[Ghosh and Ghosh, 2023].

Según Tanenbaum & Steen “un sistema distribuido es una colección de computadoras independientes que aparecen ante los usuarios del sistema como una única computadora”[Tanenbaum and van Steen, 2006].

Coulouris define que “los sistemas distribuidos son aquellos en los cuales los componentes de hardware y software están ubicados en computadoras de una red y se comunican y coordinan sus acciones solamente por medio de mensajes”[Coulouris et al., 2000].

De las anteriores definiciones establecidas por diversos autores podemos concluir: un sistema distribuido es un conjunto disperso de dispositivos independientes, con diferentes prestaciones de hardware/software que se conectan mediante una red de comunicación.

1.1. Ventajas de los sistemas distribuidos

Los sistemas distribuidos tienen las siguientes ventajas:

Rendimiento: Un sistema distribuido puede mejorar el rendimiento de un proceso involucrando múltiples componentes para realizar subtarefas, por ejemplo, operaciones en paralelo.

Especialización y autonomía: Cada módulo del sistema distribuido puede ser diseñado para ejecutar una tarea específica independientemente del tipo de tareas ejecutadas por los otros módulos. Los módulos pueden ser implementados en dispositivos con diferentes prestaciones de hardware y usando diferentes herramientas de software [Ghosh and Ghosh, 2023].

Distribución geográfica y transparencia: Es posible tener dispositivos conectados desde diferentes ubicaciones geográficas. La localización de las máquinas es transparente a las aplicaciones, por ejemplo, las replicas de un sistema de almacenamiento de datos puede ser distribuido en diferentes localizaciones geográficas para evitar la pérdida accidental de información.

Vinculación y optimización dinámica: Un sistema distribuido puede tener un conjunto de recursos computacionales similares que pueden ser asignados dinámicamente a diferentes tareas en diferentes tiempos. El sistema puede optimizar la asignación de recursos para mejorar el desempeño general del sistema.

Tolerancia a fallas: En un sistema distribuido si un componente falla, la tarea puede ser asignada a otro componente.

Apertura, escalabilidad y reconfiguración dinámica: Un sistema distribuido puede ser diseñado como un sistema abierto, donde los componentes individuales interactúan entre sí mediante protocolos de comunicación estándar. Podemos reemplazar componentes obsoletos o adicionar nuevos sin perturbar la operación normal del sistema [Tanenbaum and van Steen, 2006].

1.2. Desventajas de los sistemas distribuidos

Aunque los sistemas distribuidos ofrecen grandes ventajas como vimos anteriormente, estos también tienen desventajas como:

Complejidad: La construcción de sistemas distribuidos puede ser considerablemente más compleja que la de sistemas centralizados. La necesidad de coordinar múltiples componentes distribuidos y garantizar la coherencia de los datos aumenta la complejidad del diseño y la implementación [Tanenbaum and Van Steen, 2023].

Problemas de concurrencia y coherencia: La sincronización de operaciones concurrentes y la garantía de la consistencia de los datos pueden requerir técnicas avanzadas y añadir complejidad al sistema.

Latencia: La comunicación entre los componentes en un sistema distribuido suele implicar la transmisión de datos a través de una red, lo que puede introducir latencia. Esta latencia puede afectar el rendimiento del sistema, especialmente en aplicaciones sensibles al tiempo de respuesta.

Seguridad: Los sistemas distribuidos pueden ser más susceptibles a ciertos tipos de ataques debido a la exposición a redes públicas. Es un desafío garantizar la seguridad y la privacidad de los datos en un entorno distribuido.

Depuración y mantenimiento: La detección y corrección de errores en un sistema distribuido es compleja, ya que los problemas pueden ser difíciles de reproducir y pueden estar relacionados con la comunicación entre nodos.

Costos: Los sistemas distribuidos pueden requerir una infraestructura más compleja y costosa en términos de hardware, redes y software de soporte. Esto puede aumentar los costos de desarrollo, implementación y mantenimiento del sistema.

Memoria: En un sistema distribuido no existe una memoria global ya que los recursos están distribuidos entre los componentes.

1.3. Desafíos

La implementación de un sistema distribuido conlleva ciertos desafíos que deben tenerse en cuenta. Estos desafíos son:

1.3.1. Accesibilidad

Un sistema distribuido debe garantizar la accesibilidad remota de los recursos y compartirlos de una manera controlada y segura. Sin embargo, a medida que incrementa la conectividad y los recursos compartidos, también lo hace la importancia de implementar un sistema seguro de transmisión y almacenamiento de la información.

1.3.2. Transparencia

Un sistema distribuido oculta el hecho que los procesos y recursos están dispersos físicamente en múltiples computadores. Este se presenta al usuario como si fuera un solo computador. La transparencia de un sistema distribuido se da en diferentes flancos. La transparencia de *acceso* oculta como se accede a los recursos y como se representa los datos dependiendo de la arquitectura o sistemas operativos de las máquinas. La transparencia de *localización, migración y relocalización* oculta donde se ubica físicamente el recurso, si el recurso ha migrado a otra localización o si el recurso a sido reubicado mientras este está siendo usado.

Un sistema distribuido puede replicar un recurso en diferentes máquinas con el fin de aumentar la disponibilidad de dicho recurso o para mejorar el rendimiento. La transparencia de *replicación* oculta la existencia de las copias de un recurso. La transparencia de *conurrencia* oculta al usuario que otros usuarios están usando el mismo recurso. Un sistema distribuido debe ser transparente a *fallas* de tal manera que si un recurso falla el usuario no lo notará [Tanenbaum and van Steen, 2006].

1.3.3. Escalabilidad

Un sistema distribuido puede ser escalable respecto a su *tamaño* adicionando más usuarios y recursos, escalable *geográficamente* ya que puede existir grandes distancias entre la ubicación de los usuarios y los recursos y escalable *administrativamente* ya que puede ser administrado por varias entidades administrativas independientes. Sin embargo, la escalabilidad repercute en el desempeño del sistema distribuido.

Dependiendo del tipo de escalabilidad, el sistema distribuido tiene ciertos retos. Cuando un sistema distribuido opera en una LAN, la comunicación entre máquinas ocurre rápidamente en unos cuantos microsegundos, sin embargo, cuando el sistema se distribuye en un gran área, los procesos de intercomunicación pueden tomar tiempos del orden de milisegundos aumentando la latencia.

Un sistema distribuido escalado administrativamente puede tener problemas respecto a las políticas del uso de recursos, gestión y seguridad. Finalmente, un sistema escalado en tamaño acarrea retos en la distribución de los servicios y recursos.

1.3.4. Seguridad

Los sistemas distribuidos están expuestos a una variedad de amenazas de seguridad, como ataques de denegación de servicio, intrusión y acceso no autorizado. Implementar mecanismos de autenticación, autorización y cifrado es fundamental para proteger la integridad y la confidencialidad de los datos en un entorno distribuido.

1.3.5. Confiabilidad y tolerancia a fallos

Los sistemas distribuidos deben ser capaces de funcionar de manera confiable incluso cuando algunos de sus componentes fallan. Esto implica implementar mecanismos de detección de fallos, recuperación y tolerancia a fallos para garantizar la disponibilidad continua del sistema.

1.4. Clasificación de los sistemas distribuidos

Los sistemas distribuidos se clasifican en tres: sistemas distribuidos computacionales, de información y ubicuos [Tanenbaum and Van Steen, 2023].

1.4.1. Sistemas distribuidos computacionales

Actualmente, una clase de sistema distribuido muy importante son los que desarrollan tareas de computación de alto desempeño. Existen dos subgrupos:

Computación de clúster: Es un conjunto de nodos de cómputo homogéneos, interconectados mediante una red de alta velocidad en una red de área local y cada nodo generalmente usa el mismo sistema operativo. A menudo la computación en clúster es usada para desarrollar tareas de computación paralela. En esta, un nodo de gestión asigna las tareas a los nodos de computación, los cuales están equipados con funcionalidades de comunicación, almacenamiento y tolerancia a fallas.

Computación de malla: Es un conjunto heterogéneo de sistemas descentralizados, que usan distintos dominios administrativos, hardware, software y tecnologías de red. En esta, todos los recursos de diferentes organizaciones operan de

manera colaborativa formando una organización virtual. Los recursos consisten de servidores de cómputo, e.g. supercomputadores, instalaciones de almacenamiento y bases de datos.

En este punto es importante diferenciar entre computación paralela y concurrente. La computación paralela divide la tarea en apartados de menor tamaño que son resueltos en paralelo a partir de la ejecución de múltiples procesos de forma simultánea. La computación concurrente divide la tarea en múltiples partes que acceden a todos los recursos computacionales durante periodos de tiempo separados como se observa en la Figura 2.

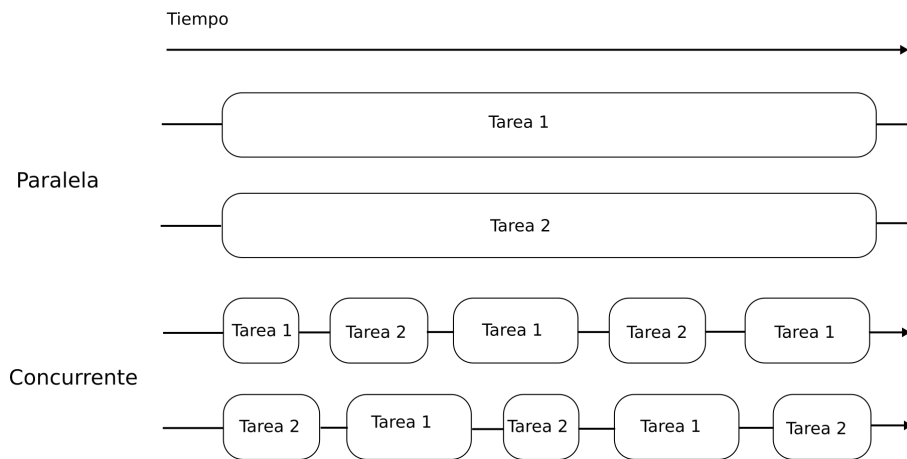


Figura 2: Comparación entre la computación paralela y concurrente.

1.4.2. Sistemas de información distribuidos

Los sistemas de información distribuidos resultan de la necesidad de integrar las múltiples aplicaciones en un sistema de información para mejorar su interoperabilidad. A medida que las aplicaciones se hacían más sofisticadas y se separaban gradualmente en componentes independientes, se hizo evidente que la integración también debía tener lugar dejando que las aplicaciones se comunicaran directamente entre sí [Tanenbaum and van Steen, 2006]. La Figura 3 muestra el uso de un *middleware* como elemento facilitador en la comunicación entre los clientes y los servidores de un sistema distribuido.

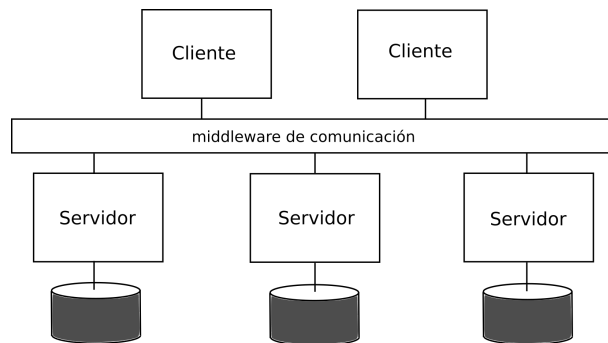


Figura 3: Sistema de información distribuido. El *middleware* media la comunicación cliente-servidor.

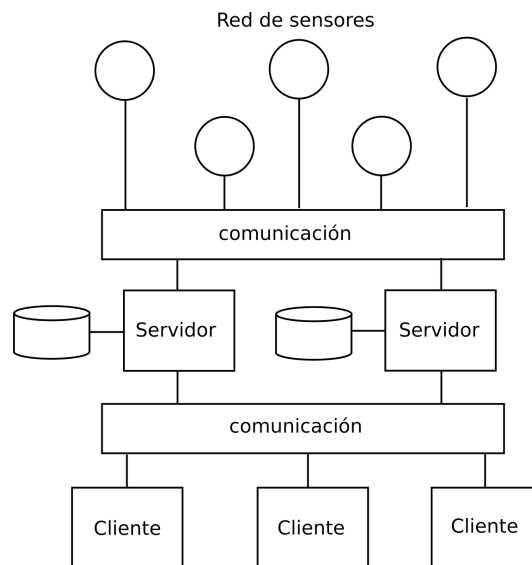


Figura 4: Sistema de distribución ubicuo. Una red de sensores transmite información a un grupo de servidores que almacenan los datos y ofrecen recursos para su procesamiento a pedido de los clientes.

1.4.3. Sistemas distribuidos ubicuos

Los sistemas distribuidos anteriormente mencionados son estables, ya que constan de nodos fijos con una conexión a la red de alta calidad, sin embargo, desde la introducción de sistemas de cómputo móviles y embebidos surgen los sistemas distribuidos ubicuos también llamados pervasivos.

En un sistema ubicuo las aplicaciones no están sujetas a ámbitos administrativos o de red concretos. Los sistemas distribuidos ubicuos se aplican en una gran variedad de campos: el Internet de las cosas (IoT), la computación en la nube, los entornos móviles o las redes de sensores como se muestra en la Figura 4. Generalmente, los dispositivos usan el sistema ubicuo para acceder o proveer información. Esto requiere que el sistema provea canales eficientes de lectura, almacenamiento, gestión e intercambio de información.

Matriz de evaluación

Matriz de evaluación de la actividad					
Evidencia	Criterio	Niveles de desarrollo de la competencia			
		Estratégico (4.5-5.0)	Autónomo (3.8-4.4)	Básico (3.0-3.7)	Inicial (1.0-2.9)
Taller	Aplicación del conocimiento 50 %	Realiza un procedimiento eficiente y hace una revisión sistemática	Resuelve el problema pero no revisa para mejorar	Resuelve el problema de manera incompleta	No soluciona el problema
	Seguimiento de instrucciones 25 %	Cumple totalmente los puntos asignados	Cumple mayormente los puntos asignados	Cumple de manera incompleta los puntos	No realiza la actividad
	Toma de decisiones 25 %	Toma decisiones sencillas, acertadas y coherentes	Toma decisiones acertadas, pero le cuesta justificar	A veces toma decisiones sencillas y acertadas	No toma decisiones

Referencias

- [Coulouris et al., 2000] Coulouris, G. F., Dollimore, J., and Kindberg, T. (2000). *Distributed Systems*. International Computer Science Series. Addison Wesley, Boston, MA, 3 edition.
- [Ghosh and Ghosh, 2023] Ghosh, R. K. and Ghosh, H. (2023). *Distributed systems*. Wiley-Blackwell, Hoboken, NJ.
- [Tanenbaum and van Steen, 2006] Tanenbaum, A. S. and van Steen, M. (2006). *Distributed systems*. Pearson, Upper Saddle River, NJ, 2 edition.
- [Tanenbaum and Van Steen, 2023] Tanenbaum, A. S. and Van Steen, M. (2023). *Distributed Systems*. Maarten Van Steen.