**LOS SISTEMAS DISTRIBUIDOS**

**Autor:**

**Bertha Eugenia Mazón Olivo**,

Ingeniera en Sistemas,

Magíster en Informática Aplicada,

12 años de experiencia en docencia superior.

**1. INTRODUCCIÓN.**

Desde el inicio de la era de las computadoras en la década de los 40 hasta cerca de los 80, surgió la computación centralizada que consistía en algunos terminales, sin capacidad de procesamiento ni almacenamiento, conectados a un supercomputador que se encargaba de todo el trabajo. Al incrementar los usuarios y la necesidad de procesamiento, crece el riesgo de fallo del computador principal y de todo el sistema; cambiar o actualizar el supercomputador implicaba costos elevados. A partir de la mitad de la década de los ochentas aparecen dos avances tecnológicos fundamentales: el desarrollo de microprocesadores poderosos y económicos y el desarrollo de redes de área local (LAN) de alta velocidad, con posibilidad de conectar cientos de máquinas a velocidades de transferencia de millones de bits por segundo (Mbps). Estos avances propiciaron la evolución de sistemas centralizados a procesamiento distribuido y luego a cliente-servidor.

A continuación se describe panorámicamente los aspectos relevantes de los Sistemas Distribuidos.

**2. DESARROLLO.**

Las organizaciones a medida que se expanden en el mercado, crecen también los procesos y su información, haciéndose inminente la innovación continua de su sistema, de sus herramientas, la tecnología y sobre todo del personal capacitado.

Para la gestión de los sistemas de información en las organizaciones que poseen cientos o miles de usuarios requiriendo acceso a dispositivos, programas y datos, es necesario pensar en sistemas distribuidos físicamente, pero lógicamente integrados como si fueran un solo sistema.

**2.1. ¿Qué es un Sistema Distribuido?**

Un sistema cuyos componentes hardware y software se encuentran en computadores conectados en red, se comunican y coordinan sus acciones mediante el paso de mensajes controlados por protocolos prefijados por un esquema cliente servidor.

**2.2. Características.**

Algunas de las características relevantes de los sistemas distribuidos son: la compartición de recursos, la concurrencia, la apertura, la escalabilidad, la carencia de un reloj global, la tolerancia a fallos de su componentes y sobretodo la transparencia. A continuación se describe cada una:

* **Compartición de Recursos**. Los recursos en un sistema distribuido están físicamente encapsulados en computadoras y pueden ser accedidos por otras computadoras mediante la red; para que la compartición de recursos sea efectiva se requiere de un programa gestor de recursos, que ofrezca un interfaz de comunicación permitiendo su acceso, manipulación y actualización de una manera fiable y consistente; los usuarios se comunican con estos programas gestores para acceder a los recursos compartidos del sistema según dos modelos de sistemas distribuidos: el modelo cliente-servidor y el modelo basado en objetos.
* **Concurrencia**. Permite que los recursos disponibles en la red puedan ser utilizados simultáneamente por los usuarios. Esto es posible, si varios procesos en una única máquina se ejecutan concurrentemente. Sin embargo, Si el computador está equipado con un único procesador central, la concurrencia tiene lugar entrelazando la ejecución de los distintos procesos. O, si la computadora tiene varios procesadores, entonces se pueden ejecutar los procesos en paralelo. En los sistemas distribuidos hay muchas computadoras, cada una con uno o más procesadores.
* **La apertura**. Los sistemas distribuidos abiertos se basan en la provisión de un mecanismo uniforme de comunicación entre procesos e interfaces publicados para acceder a recursos compartidos. Los sistemas distribuidos abiertos pueden construirse a partir de hardware y **software heterogéneo**, posiblemente proveniente de vendedores diferentes. Pero la conformidad de cada componente con el estándar publicado debe ser cuidadosamente comprobada y certificada si se quiere evitar tener problemas de integración.
* **La escalabilidad**. Tanto el software de sistema como el de aplicación no deberían cambiar cuando la escala del sistema se incrementa. La demanda de escalabilidad en los sistemas distribuidos ha conducido a una filosofía de diseño en que cualquier recurso simple hardware o software puede extenderse para proporcionar servicio a tantos usuarios como se quiera. Cuando el tamaño y complejidad de las redes de computadores crece, es un objetivo primordial diseñar software de sistema distribuido que seguirá siendo eficiente y útil con esas nuevas configuraciones de la red. Las técnicas necesarias para conseguir estos objetivos incluyen el uso de datos replicados, la técnica asociada de caching, y el uso de múltiples servidores para manejar ciertas tareas, aprovechando la concurrencia para permitir una mayor productividad.
* **Tolerancia a Fallos**. Cuando uno de los componentes de un sistema distribuidos falla, solo se ve afectado el trabajo que estaba realizando el componente averiado. Un usuario podría desplazarse a otra estación de trabajo y el proceso servidor podría ejecutarse en otra máquina sin problema. El diseño de sistemas tolerantes a fallos se basa en dos cuestiones, complementarias entre sí: redundancia hardware (uso de componentes redundantes) y recuperación del software (diseño de programas que sean capaces de recuperarse de los fallos). En los sistemas distribuidos la redundancia puede plantearse en un grano más fino que el hardware, pueden replicarse los servidores individuales que son esenciales para la operación continuada de aplicaciones criticas. La recuperación del software tiene relación con el diseño de software que sea capaz de recuperar (roll-back) el estado de los datos permanentes antes de que se produjera el fallo.
* **La carencia de un reloj global**. Se refiere a que no existe una temporización general para la ejecución de los procesos entre los diferentes componentes, es más bien distribuida.
* **Transparencia**. La transparencia se define como la ocultación al usuario y al programador, la separación de las aplicaciones y de los componentes de un sistema distribuido, de manera que el sistema se percibe como un todo, en vez de una colección de componentes independientes. La transparencia ejerce una gran influencia en el diseño del software de sistema. Existen varios tipos de transparencia, la más importante es la transparencia de red que abarca el acceso y localización de los recursos, simulando ser un sistema centralizado.

**2.3. El Modelo Cliente Servidor.**

El modelo cliente-servidor de un sistema distribuido es el modelo más conocido y más ampliamente adoptado en la actualidad. Un cliente es una máquina que realiza una solicitud al servidor; a su vez, el servidor es otra máquina que atiende las peticiones de los clientes y provee servicios como: el acceso a datos, la ejecución de algún programa o el acceso a algún dispositivo hardware. En los sistemas cliente servidor se suman una serie de elementos que hay que considerar: tipos de clientes y servidores, protocolos, software intermedio (middleware), formatos de intercambio de datos (Ejm. HTML, XML, JSON), las tecnologías para desarrollo de aplicaciones(PHP, J2EE, ASP.NET, etc), sistemas operativos, redes de computadores, seguridades, entre otros aspectos.

**2.4. Evolución de las Aplicaciones de sistemas distribuidos**

|  |  |
| --- | --- |
| **1. APLICACIONES CENTRALIZADAS**  **CLIENTE**  **(Presentación)**  **SERVIDOR**  **(Aplicación + Datos)**   * El servidor: procesa y almacena datos * Se distribuye la presentación o interfaz del usuario a clientes. * Ejm: telnet, Xwindows, Windows Terminal, VNC, etc. * El servidor es un supercomputador, costoso y difícil de actualizarlo | **2. GRUPO DE SERVIDORES**  **Clientes (Presentación)**  **Grupo de Servidores**  - Aplicación + datos  - Distribución del procesamiento |
| **3. CLIENTE/SERVIDOR EN 2 CAPAS**  **CLIENTE**  **(Presentación + Aplicación)**  **SERVIDOR**  **(Base de Datos + SP)**   * El Cliente: Presentación + Aplicación * El Servidor: Bases de Datos + Procedimientos almacenados (SP) * Intercambio de sentencias SQL, No orientado a transacciones, muy orientado a 4GL | **4. PROCESADORES DE TRANSACCIONES (OLTP)**  **CLIENTE**  **(Presentación + Aplicación)**  **Procesador de Transacciones**  **SERVIDOR**  **(Datos + SP)**  **Transacción**  **Sentencias SQL** |
| **5. OBJETOS DISTRIBUIDOS**  **CLIENTE**  **SERVIDOR**  **(Objetos) Distribuidos)**  **CORBA/RMI/DCOM**  **PETICIÓN**   * Primeros pasos en RPC’s (llamadas a procedimientos remotos) * DCE (Distributed Computing Environment) * Un cliente en cualquier parte de la red, invoca un método de un objeto remoto ubicado en un servidor. * Adecuados en comunicaciones: Cliente a Servidor y Servidor a Servidor * Tecnologías: * RMI (Invocación Remota de Métodos) de Java * DCOM de Microsoft * CORBA de la OMG: Independiente del Sistema Operativo | **6. WORLD WIDE WEB**  **SERVIDOR WEB**  **SERVIDOR DE DATOS**  **CLIENTE WEB**   * WWW es un sistema de documentos (o páginas web) interconectados por enlaces de hipertexto, disponibles en Internet * Protocolos: HTTP, WAP de TCP/IP * Ejecución en el cliente: HTML, MIME, CSS, DOM, JavaScript, Ajax, XML * Ejecución en el servidor web: CGI, PHP, JSP, Servlets, ASP * Web 2.0 |
| **7. INTEGRACIÓN DE APLICACIONES WEB**  **CON OBJETOS DISTRIBUIDOS** | http://cybertesis.upc.edu.pe/upc/2008/vasquez_dc/xml/ressources/100000000000014F000001A1445E18CC.jpg**8. COMPONENTES DISTRIBUIDOS**   * Evolución de Objetos Distribuidos * Bloques de construcción de aplicaciones llamados COMPONENTES * Conceptos de CONTENEDORES * Servicios disponibles para los Componentes ofrecidos por el Contenedor: * Comunicaciones * Transacciones * Ejecución * Ciclo de Vida * Persistencia * Plataformas de desarrollo: **J2EE y Microsoft.NET** |
| **9. ARQUITECTURA ORIENTADA A SERVICIOS (SOA)**  http://4.bp.blogspot.com/_o4qGtxlF994/S7-E5Za58vI/AAAAAAAAAWo/kn0a4Vi625M/s1600/como+entender+los+servicios.JPG  SOA. Es un concepto de arquitectura de software que define la utilización de servicios para dar soporte a los requisitos del negocio. Operan sobre cualquier arquitectura o tecnología, están geográficamente dispersos y bajo cualquier figura de propiedad.Las funcionalidades de la capa aplicativa son expuestas en forma de servicios (servicios web). Los servicios son desplegados a los usuarios finales. | **10. AGENTES INTELIGENTES**  Un agente inteligente es una entidad software que, basándose en su propio conocimiento, realiza un conjunto de operaciones para satisfacer las necesidades de un usuario o de otro programa, bien por iniciativa propia o porque alguno de éstos lo requiere.[http://t1.gstatic.com/images?q=tbn:ANd9GcQOUS481LSaic5GhqD-fEF2XqqP7ojirKW0d59wkKqQfa9eS6GFhttp://t1.gstatic.com/images?q=tbn:ANd9GcQOUS481LSaic5GhqD-fEF2XqqP7ojirKW0d59wkKqQfa9eS6GF](http://www.google.com/imgres?imgurl=http://iacarolmarroquin.files.wordpress.com/2010/08/g_index_2.jpg&imgrefurl=http://iacarolmarroquin.wordpress.com/2010/08/28/agentes-inteligentes/&usg=__QNqKXtUgL3xUQ2A312E7p6CLt0c=&h=216&w=334&sz=15&hl=es&start=4&zoom=1&tbnid=ImWto6YVFdU9BM:&tbnh=77&tbnw=119&ei=FRPnTdD9JIHegQerrZnNCg&prev=/search?q=agentes+inteligentes&um=1&hl=es&biw=1003&bih=562&tbm=isch&um=1&itbs=1) |

**2.5. Ventajas de los Sistemas Distribuidos**

* La economía, es mucho más barato, añadir servidores y clientes cuando se requiere aumentar la potencia de procesamiento.
* El trabajo en conjunto. Por ejemplo: en una fábrica de ensamblado, los robots tienen sus CPUs diferentes y realizan acciones en conjunto, dirigidos por un sistema distribuido.
* La confiabilidad. Al estar distribuida la carga de trabajo en muchas máquinas, la falla de una de ellas no afecta a las demás, el sistema sobrevive como un todo.
* Capacidad de crecimiento. Se puede añadir procesadores al sistema incrementando su potencia en forma gradual según sus necesidades.
* La compartición de recurso. Se pueden compartir recursos, como programas, periféricos y datos
* La Disponibilidad de los Dispositivos. Gracias a la redundancia los usuarios podrán acceder a los servicios a pesar de algún fallo.
* Se logra una mejor comunicación entre las personas. Ejemplo: el correo electrónico.
* Tienen mayor flexibilidad, la carga de trabajo se puede distribuir entre diferentes ordenadores.

**3. CONCLUSIONES**

* Los sistemas distribuidos integran elementos heterogéneos como: hardware y software de cliente y servidor, sistemas operativos, redes, protocolos, seguridades, distintas tecnologías de de aplicaciones distribuidas, entre otros, por lo cual su desarrollo e implementación deben ser cuidadosamente planificados.
* Muchas tecnologías están en constante desarrollo y maduración, lo cual implica un minucioso estudio de muchos factores antes de apostar por alguna tecnología en especial.

**4. REFERENCIAS**

* **Coulouris, G., Dollimore, J. and Kindberg, T. 2001.** *Sistemas distribuidos. Conceptos y diseño.* s.l. : Addison-Wesley, 2001.
* **Hurtado, Omar .** Monografías.com. [Online] http://www.monografias.com/trabajos16/sistemas-distribuidos/sistemas-distribuidos.shtml.
* **Unidad Docente de Sistemas, Facultad Informática, Universidad Politécnica de Madrid.** Sistemas Distribuidos. [Online] http://www-lt.ls.fi.upm.es/sistemas\_dist/index.php?option=com\_content&view=article&id=73&Itemid=94.
* **Universidad EAFIT; Departamento de Informática y Sistemas.** Evolución de los sistemas distribuidos. [Online] http://dis.eafit.edu.co/cursos/st725/material/evol-sd.pdf.