Sistemas distribuidos

Capítulo 1:

Definición de sistema distribuido:

Un sistema distribuido es aquel en el que sus componentes se encuentran en computadoras conectadas a alguna red, que se comunican y coordinan sus acciones a través del paso de mensajes.

Se establece la comunicación mediante un protocolo prefijado por un esquema cliente-servidor.

<http://www.monografias.com/trabajos16/sistemas-distribuidos/sistemas-distribuidos.shtml>

Ejemplos de sistemas distribuidos:

Hadoop.

Dropbox (en general computación en la nube).

Netflix.

BitTorrent.

Whatsapp.

Redis.

MongoDB.

Bitcoin.

Minecraft.

Internet of Things (IoT).

Características de un sistema distribuido:

* Existe concurrencia de componentes:
  + Concurrencia: Es la capacidad del sistema para manejar recursos al mismo tiempo.

<http://coresoft01.blogspot.com/2012/08/que-es-un-sistema-distribuido.html>

* + Dos procesos son concurrentes, paralelos, o se ejecutan concurrentemente cuando ambos pueden ejecutarse al mismo tiempo, es decir, que para ejecutar uno de ellos no es necesario que se haya ejecutado el otro. <https://www.webprogramacion.com/43/sistemas-operativos/concurrencia-de-procesos.aspx>
* Cada computadora se encuentra atendiendo requerimientos diferentes.
* Los usuarios se comportan de forma diferente.
* Existe un cambio de recursos frecuente.
* Los miembros del sistema deben compartir recursos y coordinarse.
* No hay un reloj global:
  + Se refiere a que no hay una hora correcta, ya que los dispositivos parte del sistema, no dependen del tiempo correcto sino del paso de mensajes.

<http://coresoft01.blogspot.com/2012/08/que-es-un-sistema-distribuido.html>

Las coordinaciones para la transferencia de mensajes entre los diferentes componentes para la realización de una tarea, no tienen una temporización general, está más bien distribuida a los componentes.

<http://www.monografias.com/trabajos16/sistemas-distribuidos/sistemas-distribuidos.shtml>

* + Los programas cooperan usando mensajes.
  + Cierta coordinación requiere saber en qué momento ocurrieron las cosas.
* Fallas independientes:
  + Si hay una falla en alguna parte del sistema, este no se vendría abajo completamente, solo la parte afectada.

<http://coresoft01.blogspot.com/2012/08/que-es-un-sistema-distribuido.html>

* + Los fallos no son predecibles.
  + Los fallos pueden ser en computadoras individuales o en toda la red.

¿Por qué construir un sistema distribuido?

* Para compartir recursos entre distintos nodos.
* El rendimiento es mayor que en sistemas no distribuidos.
* Los tiempos de respuesta de los mensajes son menores.
* Existe mayor tolerancia a fallos.
* Existe una alta disponibilidad.
* Existe anonimidad.

Aplicaciones distribuidas:

* Conjunto de procesos distribuidos en una red de computadoras que trabajan juntas para resolver un problema en común.
* Anteriormente el enfoque era cliente-servidor
* Actualmente el enfoque es altamente distribuido, abarcando arquitecturas peer-to-peer y multi-capas.
* Como enfoques peer-to-peer tenemos:
  + BitTorrent
  + Telefonía por Voz sobre IP

<https://es.wikipedia.org/wiki/Voz_sobre_protocolo_de_internet>

* Programación por capas:
  + La programación por capas es una arquitectura cliente servidor cuyo objetivo primordial es la separación de la lógica de negocios de la lógica de diseño, es decir, separar la capa de datos de la capa de presentación al usuario, cada grupo de trabajo está completamente abstraído del resto de niveles.
  + La principal ventaja está en que el desarrollo se puede llevar a cabo en varios niveles, en caso de cambios, solo afecta al nivel requerido, sin tener que revisar todo el código.
  + Como ejemplos tenemos el modelo de interconexión de sistemas abiertos.

<https://es.slideshare.net/maryme/arquitectura-multicapa>

Desafíos:

* Heterogeneidad:
  + Se refiere a la variedad y diferencia que podemos encontrar entre los componentes de un sistema distribuido, dicha heterogeneidad aplica al tipo de redes, hardware de los equipos, sistemas operativos, lenguajes de programación e implementaciones en las que trabajan los diferentes desarrolladores.

<https://prezi.com/dibidjz3oysa/desafios-de-los-sistemas-distribuidos/>

* + El middleware es una capa de software que asiste a una aplicación para comunicarse con otras aplicaciones, redes, hardware y/o sistemas operativos
  + Generalmente se implementa sobre protocolos de internet, que esconden la heterogeneidad de la red, representación de los datos.
  + Simplifica el trabajo de los programadores de generar las conexiones que son necesarias en los sistemas distribuidos, de esta forma se provee una solución que mejora la calidad de servicio, seguridad, envío de mensajes, directorio de servicios, etc.

<https://prezi.com/dibidjz3oysa/desafios-de-los-sistemas-distribuidos/>

* + El middleware también proporciona abstracción de programación y esconde heterogeneidad de redes, hardware, sistemas operativos, lenguajes de programación.
  + CORBA (Common OBject Request Broker Architecture) es un middleware y estándar que permite que diversos componentes de software escritos en múltiples lenguajes de programación y que corren en diversos computadores, puedan trabajar juntos.

<https://es.wikipedia.org/wiki/CORBA>

* + Java RMI (Java Remote Method Invocation) es un mecanismo ofrecido por java para invocar un método de forma remota, proporciona un mecanismo simple para la comunicación de servidores en aplicaciones distribuidas basadas exclusivamente en java.

<https://es.wikipedia.org/wiki/Java_Remote_Method_Invocation>

* Glue software: Es el código fuente que sirve únicamente para adaptar diferentes partes del código que de otro modo serían incompatibles, pero no aporta ninguna funcionalidad para cumplir con los requisitos del programa.

<https://translate.google.com.ec/translate?hl=es&sl=en&u=https://en.wikipedia.org/wiki/Glue_code&prev=search>

Enmascaramiento de fallos:

Proceso de asegurar que solo los valores correctos pasarán al límite del sistema a pesar de que un componente fallado.

Transparencia de distribución o transparencia de red:

Cuando los usuarios no tienen que especificar donde se encuentran los datos almacenados. También es cuando no hay diferencia entre aplicaciones de base de datos que se ejecutan en DBMS centralizados o distribuidos.

<http://www.w3ii.com/es/distributed_dbms/distributed_dbms_distribution_transparency.html>

Replicación:

Consiste en replicar los datos para aumentar el la disponibilidad o el rendimiento. Está muy relacionado con la escalabilidad.

<http://samadistribuidos.blogspot.com/2012/03/consistencia-y-replicacion.html>

Agente móvil:

Son programas que tienen la capacidad de moverse por diferentes nodos de una red una o más veces, son autónomos con la función de ejecutar tareas, se envían como objetos a través de plataformas conservando su código, datos y estado de ejecución.

<http://www.maestrosdelweb.com/agentes-moviles-y-sus-principales-caracteristicas/>

Código móvil:

Código que se traslada de un servidor al cliente, y se ejecuta en el cliente.

<https://oposicionestic.blogspot.com/2011/06/sistemas-distribuidos.html>

Cliente ligero:

Es una computadora cliente o software de un cliente en una arquitectura de red cliente-servidor que depende primariamente del servidor central para las tareas de procesamiento, principalmente se enfoca en transportar la entrada y salida entre el usuario y el servidor remoto.

<https://www.ecured.cu/Clientes_Ligeros>

Extensibilidad:

Similar al concepto de compatibilidad, que el sistema puede ofrecer a nuevos componentes. También justifica el uso de estándares, cuanto menos las normas del sistema deben ser conocidas por los componentes. También suele hablarse de sistemas abiertos cuando se refiere a sistemas extensibles.

<https://riunet.upv.es/bitstream/handle/10251/6408/Comunicaciones%20en%20los%20sistemas%20distribuidos.pdf;jsessionid=7275F779F561C152C5CFD7C3D2AD565F?sequence=1>

Seguridad:

Los 3 aspectos de seguridad más importantes son: disponibilidad, confidencialidad e integridad. La disponibilidad se refiere a que los recursos del sistema están disponibles para los componentes que los necesiten, en el instante que sea necesario. Confidencialidad es la protección que el sistema provee para no ser accesible por componentes no autorizados, esto involucra evitar la intrusión y también la suplantación por parte de un componente a otro de un sistema. Por integridad se refiere a la protección que el sistema suministra contra la alteración, accidental o provocada, de los datos que circulan por el sistema.

<https://riunet.upv.es/bitstream/handle/10251/6408/Comunicaciones%20en%20los%20sistemas%20distribuidos.pdf;jsessionid=7275F779F561C152C5CFD7C3D2AD565F?sequence=1>

Escalabilidad:

Un componente es escalable, si la efectividad del mismo no disminuye aun cuando sus componentes aumentan.

<https://riunet.upv.es/bitstream/handle/10251/6408/Comunicaciones%20en%20los%20sistemas%20distribuidos.pdf;jsessionid=7275F779F561C152C5CFD7C3D2AD565F?sequence=1>

Soporte a fallos:

Algunas de las características que un sistema distribuido debe tener con respecto a los fallos son la detección, el enmascaramiento, la tolerancia y la recuperación. La detección de fallos debe realizarse lo antes posible para evitar que errores en el procesamiento tengan consecuencias colaterales en el resto de los componentes. Una vez que el fallo es detectado, este debe enmascararse o atenuarse, los componentes deben conocer cuando se produce un fallo y reaccionar de forma conocida y controlada. La recuperación es la fase en la que el sistema, una vez solucionado el fallo, vuelve a funcionar según los requerimientos.

<https://riunet.upv.es/bitstream/handle/10251/6408/Comunicaciones%20en%20los%20sistemas%20distribuidos.pdf;jsessionid=7275F779F561C152C5CFD7C3D2AD565F?sequence=1>