Middleware Models and Architectures

Claudia Salado Méndez

1 Resumen Conferencia

1.1 Introducción a los Sistemas Distribuidos

¿Qué son? - Son sistemas en los cuales sus componentes están localizados en ordenadores conectados por una red, se comunican y coordinan mediante el paso de mensajes. Las características principales de estos sistemas es que hay consistencia entre los componentes estos pueden cooperar de forma cooperativa si tienen un objetivo común o competitiva, otra característica es que no tienen un reloj global interactúan mediante paso de mensajes y lo último es que los fallos son en componentes independientes, por ejemplo ataques o crasheos en los nodos o retrasos pérdidas en el sistema de comunicaciones.

1.2 Desafíos del Diseño

- **Heterogeneidad:** son por ejemplo tecnologías, protocolos, lenguajes de programación, servidores, clientes, ...
- **Franqueza:** es la falta de restricciones para que los nodos heterogéneos puedan acceder a servicios y facilitar su interpolación.
- **Seguridad:** esta engloba la confidencialidad, que protege cuando usuarios sin autorización acceden, también la integridad que protege de alteraciones y la disponibilidad.
- **Concurrencia**: permite que el acceso a recursos compartidos sea simultáneo.
- **Transparencia:** tiene 8 significados distintos: acceso, ubicación, concurrencia, replicación, falla, movilidad, rendimiento y transparencia de escala.
- **Escabilidad:** que sea eficiente y eficaz incluso aumentando mucho el número de usuarios.
- **Fallos:** los fallos en los SD son fallos parciales, se puede controlar el nivel de servicio para solucionarlos.

1.3 Middleware

¿Qué es? - Es cualquier capa de software que esté entre el sistema operativo y las distintas aplicaciones. Estas a través de las APIs nos dan servicios y abstracciones para el desarrollo en alto nivel, las cuales aumentan la productividad de los programadores para acceder a recursos y para comunicaciones al SOy primitivas de send/receive.Estas abstracciones admiten muchas de las propiedades de la transparencia.

En la actualidad es común utilizar un software de desarrollo llamado integración de aplicaciones, las aplicaciones necesitan interoperar con otros sistemas (legacy systems), COTS, ... El middleware es el que se encarga de unirlas por eso lo llamamos “glue technology”. Ahora veremos algunos ejemplos de modelos de middleware como RPC, MO, TS, ...

1.4 Remote Procedure Call

¿Qué es? - Es una ampliación de las llamadas a procedimientos de abstracción de lenguajes de programación tradicionales.
¿Cómo funciona? - Se compila una especie de prototipo generando unos procedimientos, stubs, hay un stub para el cliente y otro para el servidor, ambos se combinan y unen con el “caller” y el “callee” (llamador y destinatario) respectivamente.Los stubs le ahorran al programador los detalles de “empaquetar” los argumentos, enviar un mensaje, esperar el menaje, ...

Los fallos en los nodos y en los canales pueden causar pérdidas en las llamadas a mensajes y al recibirlos, también crasheos del servidor después de recibir una llamada pero antes de enviar la respuesta.

1.5 Message Oriented Middleware

Está basado en la abstracción de colas de mensajes entre procesos, es una generalización de la abstracción del mailbox de algunos sistemas operativos. La comunicación es indirecta, asíncrona, productor-consumidor, uno a muchos y muchos a muchos.

- **Productor/Consumidor:** los mensajes van tipados, los productores los meten en las colas y los consumidores cogen los que les interesan, se garantiza el envío de mensajes, el programador no tiene preocupaciones de bajo nivel, tiene una interfaz de programación de alto nivel.

- **Pub/Sub:** los consumidores están interesados en un nuevos tipos de mensajes, suscribiendote al MOM, los productores publican el contenido y el MOM notifica que hay contenido disponible. Tanto las comunicaciones pull como push son posibles, el mensaje es empujado(push) al consumidor y este decide cuando tira (pull) el mensaje.

- **Java Message Service (JMS):** es un conjunto de interfaces y semánticas asociadas que definen cómo accede un cliente JMS. Es un estándar para que los programas Java envíen y reciban mensajes por MOM.

1.6 Blockchain

Es un distributed public ledger (DLT) de transacciones digitales o eventos ejecutados y compartidos entre los participantes.

Cada transacción que esté en el blockchain está validada por un consenso entre los nodos, la información almacenada no puede eliminarse ni modificarse, cada bloque contiene una o más transacciones guardadas, esto es inmutable y puede verificarse desde el primer bloque.

Algunas de las aplicaciones del blockchain con las arquitecturas IoT: aplicaciones de seguridad, diagnóstico y mantenimiento de máquinas, trazabilidad, seguimiento en cadena de un suministro, etc.

1.7 Tuple Space Model

Está basado en en uno de los primeros middleware, las aplicaciones se comunican a través de la abstracción de un espacio virtual compartido, es un modelo muy simple de sincronización e intercambio de datos de tuplas.

Java Space: las interacciones son asíncronas, el espacio d ecada tupla lo podemos ver como unas asociacion de espacios de memoria, que se accede por tipado no por direcciones, Para leer se requiere un patrón para compararlo con tuplas que ya están en el espacio, encaja muy bien con el paradigma Maestro/Esclavo. Hay implementada tecnología Jini.
En Java Space el espacio de la tupla se convierte en el espacio de un objeto pero en Jini se convierten en servicios y proxy.

1.8 Distributed Objects

Un objeto es una encapsulación de una estructura de datos en las operaciones para acceder a él. En los lenguajes tipados es una instancia de un tipo abstracto de datos (ADT).
En los lenguajes de programación los subprogramas y los objetos son buenos módulos software, pero los subprogramas solo nos otorgan abstracción funcional, la información escondida es el algoritmo que implementa la funcionalidad. Por otro lado los objetos nos dan abstracción de datos, la información escondida son estructuras de datos encapsuladas o algoritmo de operaciones.

Hay que destacar:

- **RMI:** no está basado en ORB, es específico para java, los objetos del servidor están registrados en un catálogo, los objetos de los clientes miran en el catálogo, cogen la referencia e invocan los procedimientos remotos. Se usa URL por ello se necesita un servidor web.

1.9 **Component Based Architectures**

El modelo de componentes es una evolución del modelo de objetos distribuidos, un componente es un módulo software con una interfaz clara y ejecutable, las dependencias con otro componentes tienen que estar explícitas en la interfaz, desplegable independientemente, también se puede unir con otros módulos para que sea más complejo.

- **Contenedores:** entorno de ejecución de componentes, maneja el ciclo de vida de los componentes, su activación, su detención y reactivación, … Ofrece servicios estándar predefinidos, como seguridad, persistencia, etc. Su ventaja es que se desvía mucho esfuerzo del desarrollo y programación al montaje y despliegue.

1.10 **Service Oriented Architectures (SOA)**

Es un enfoque para acoplarse libremente, independientemente del protocolo, computación distribuida basada en estándares. Recurso software disponible en la red y considerado un servicio. Se dice que SOA era la solución de tecnología industrial que proporciona flexibilidad

Principios para el mantenimiento, desarrollo y uso de una SOA;

- Reutilización, granularidad, modularidad, componibilidad, componentización e interoperabilidad.
- Cumplimiento de estándares ,ya sean comunes o específicos.
- Identificación y categorización de servicios, aprovisionamiento y entrega, y seguimiento.

1.11 **Microservice Architectures**

Un microservicio es una aplicación que se puede implementar , escalar y probar de forma independiente, y tiene una única responsabilidad.

El internet de las cosas - es una infraestructura global que habilita servicios avanzados de cosas interconectadas basadas en la existencia y evolución, información interoperable y tecnologías de comunicación.

Arquitectura IoT:

- **Capa de aplicación:** proporciona al cliente servicios y procesamiento funcional de datos y almacenamiento.
- **Capa de la red:** descubre, conecta y convierte los dispositivos a través de la red y los coordina con la capa de aplicación.
- **Capa de percepción:** sensores, actores y dispositivos que interactúan con el medio.

2 Comparación con Información Adicional

Para poder comparar la información dada en la conferencia, he buscados otras y también algunos artículos de revistas, algunas de las cosas más importantes que he sacado:

En cuanto al middleware nos da la misma introducción que en la charla del profesor russo, nos dice que el middleware es un tipo de servicios comunes entre la plataforma y las aplicaciones, y que comparten una interfaz, unos protocolos estándar y que realizan diferentes nodos de hardware basados en el. El artículo explica los antecedentes de investigación sobre tecnología del middleware y arquitectura de red ubicua, esta nos la presenta como:

"La red ubicua se puede dividir en capa de integración de red, capa de convergencia de servicios y capa de convergencia de aplicaciones, que corresponden al lado del dispositivo, al lado de la plataforma y al lado de la aplicación en el sistema de red ubicuo real."[1]

También nos explica cómo son las interfaces de los middlewares, estas son para dispositivos y funciones de administración del middleware, este artículo nos muestra el diseño de las interfaces y de algunas funciones del middleware.

Tambien he buscado algunos artículos sobre el blockchain ya que le dimos bastante importancia en clase y en la charla se habló de ello, en concreto he encontrado un artículo que habla sobre las innovaciones del blockchain y explica el concepto, nos explica que el blockchain como tecnología tiene el potencial de cambiar la forma en la que se realizan las transacciones y redefinir cómo operan las organizaciones y sociedad, También nos cuenta que el blockchain no se limita solo al uso con las criptomonedas, hoy en día se usa para proteger y administrar activos digitales y mantener los registros de ellos en formas inmutables.

"Blockchain estaba destinado a servir como un libro de contabilidad abierto para todas las transacciones dentro de una red de usuarios para resolver el problema del doble gasto combinando tecnología peer-to-peer con criptografía de clave pública" [2]

Por último he buscado información sobre las llamadas a procedimientos remotos (RPC), encontré una charla que nos decía que la RPC es un paradigma popular para la comunicación entre los procesos IPC entre procesos en diferentes ordenadores a través de la red y que que es muy utilizado en los sistemas distribuidos. Es bastante simple conceptualmente y fácil de implementar pero tiene muchos problemas involucrados que resultan en diferentes implementaciones de RPC. En este documento se analizan todas esas implementaciones como Xerox Courier, SUN ONC, Apollo RPC,... .[3]

3 Valoración/Opinión Personal

En mi opinión y tras haberme informado más de estos temas al buscar nueva información, creo que la charla del profesor Russo estuvo bastante bien, aunque no pude enterarme de muchas cosas y algunos ejemplos que nos mostró, y me costó un poco seguirle el hilo, gracias a las diapositivas pude enterarme mas o menos de todo, creo que me sirvió para entender y tener una vista más general de todo lo que vemos en la asignatura, porque toco temas que hemos visto en clase, o que hemos utilizado en las prácticas.

En general creo que fue interesante y una nueva experiencia acudir a esta charla. Lo único que a lo mejor hubiera preferido es que se profundizará más en algún tema porque todo lo que nos explicó, o de lo que yo me pude llegar a enterar, fue por encima y si ahora tengo bastantes más conceptos básicos de las cosas más importante de los sistemas distribuidos pero si la charla se hubiera centrado más en algún tema creo que a mi me hubiera llamado más la atención por ejemplo con el blockchain que lo hemos hablado mucho en clase y para el segundo examen nos tuvimos que informar de su importancia.

En conclusión, la charla fue interesante, a mi parecer fue amena y a la hora de que algunos de mis compañeros hicieran preguntas y el profesor les respondiera, solo estuviéramos nosotros en la llamada tambien me parecio cercana, también- me ha hecho conocer más estos conceptos que aunque los hemos escuchado bastantes veces en clase, ahora tengo una idea más clara.

Referencias

1. J. Zhang y Z. Yan, "Investigación en tecnología de middleware y diseño de interfaces en redes ubicuas", *Séptima Conferencia Internacional sobre Tecnología de Medición y Automatización Mecatrónica de* 2015, págs. 659-662, doi: 10.1109 / ICMTMA.2015.164.
   https://ieeexplore.ieee.org/document/7263658

2. *Swan M. Blockchain: Plan para una nueva economía. O'Reilly Media, Inc., 2015.*
   https://www.journals.elsevier.com/information-and-management/call-for-papers/call-for-papers-blockchain-innovations

3. *Tay, BH y Ananda, AL, Una encuesta de llamadas a procedimientos remotos 1990, {julio de 1990}, Association for Computing Machinery, Nueva York, NY, EE. UU.0163-5980,*
   https://doi.org/10.1145/382244.382832