Universidad

Nacional de

Ingeniería

Área de Conocimiento de Tecnología de la Información y Comunicación

Comunicación y Sincronización.

**Clase Práctica 3 del curso Sistemas Distribuidos.**

|  |  |  |
| --- | --- | --- |
| **Elaborado por:** |  | **Docente:** |
| **Br. Darling Jimenez**  **Carnet: 2018-0087U** | **Br. Axel Solís**  **Carnet: 2018-0369U** | **Ing. Araceli Torres.** |

Agosto del 2025

Managua, Nicaragua

**I. Conteste:**

**1. Establezca la diferencia entre comunicación síncrona y asíncrona para los**

**sistemas distribuidos**

La diferencia principal es el tiempo. La comunicación síncrona es en tiempo real, como una videollamada, donde el emisor y el receptor interactúan al mismo tiempo. La comunicación asíncrona no es en tiempo real, como un correo electrónico, donde el mensaje se envía y la respuesta puede llegar después.

**2. Diga las ventajas del modelo Cliente-Servidor**

Las ventajas del modelo Cliente-Servidor son que es más fácil de administrar, los datos están centralizados en el servidor, lo que mejora la seguridad y el control. Además, es escalable, podemos agregar más clientes sin afectar el rendimiento del servidor.

**3. En cuanto a la sincronización de relojes, ¿Cuál es el principal problema que**

**enfrenta dicha sincronización?**

El principal problema de la sincronización de relojes es que no hay un reloj global perfecto y los relojes de las computadoras tienden a desviarse (drift). Esto hace que sea difícil para los nodos de un sistema distribuido tener una noción precisa y compartida del tiempo.

**4. Retome uno de los protocolos de Mensajería y de Consenso, explique**

**brevemente la utilidad de los mismos.**

Protocolo de Mensajería: MQTT. Es un protocolo muy ligero y eficiente, ideal para dispositivos con recursos limitados, como los del Internet de las Cosas (IoT). Su utilidad es permitir que los dispositivos se comuniquen de forma sencilla y con bajo consumo de energía, por ejemplo, para que un sensor de temperatura envíe datos a una plataforma.

Protocolo de Consenso: Paxos. Su utilidad es lograr que varios nodos en un sistema distribuido se pongan de acuerdo sobre un valor único. Esto es crucial para la consistencia de los datos, por ejemplo, en bases de datos distribuidas, para que todos los nodos "vean" la misma información y no haya conflictos.

**5. Para la comunicación Distribuida ¿Qué significa y que implica la tolerancia a fallos?**

En la comunicación distribuida, la tolerancia a fallos significa que el sistema puede seguir funcionando, aunque una o varias de sus partes fallen. Esto implica que el sistema está diseñado para manejar errores, como la caída de un servidor o la pérdida de una conexión de red, sin que el servicio se caiga por completo.

**II. Ejercicio práctico:**

**Analice su entorno cotidiano estudiantil/laboral y desglose entre el uso**

**común una aplicación (ejemplo: WhatsApp, Google Drive, Netflix); para con**

**ello responder:**

**1. En qué momento/caso podríamos identificar la comunicación síncrona y**

**Asíncrona**

**Aplicación elegida: Google Drive**

Comunicación síncrona: Cuando varios usuarios están editando un mismo documento de Google Docs al mismo tiempo. Los cambios que hace un usuario aparecen casi instantáneamente en la pantalla de los demás. Hay una interacción en tiempo real.

Comunicación asíncrona: Cuando comparto un documento con un compañero y le dejo un comentario. Él lo verá cuando se conecte más tarde y podrá responder en su momento. La comunicación no es inmediata, cada quien actúa en su propio tiempo.

En Google Drive, el modelo Cliente-Servidor Distribuido se aplica de la siguiente manera:

Clientes: Somos nosotros, los usuarios, usando la aplicación de Google Drive en nuestro navegador, computadora o celular. Solicitamos servicios, como subir un archivo, descargar uno o editar un documento.

Servidores Distribuidos: Los archivos no están en un solo lugar. Google tiene muchos centros de datos (servidores) distribuidos por todo el mundo. Cuando subimos un archivo, se almacena en uno de esos servidores. Cuando lo solicitamos, el sistema lo busca y lo envía. Esta distribución hace que el servicio sea más rápido y confiable, ya que si un servidor falla, hay otros que pueden tomar su lugar. Los servidores se comunican entre sí para replicar los datos y mantenerlos sincronizados.

**2. ¿Cómo esta aplicado el modelo Cliente-Servidor Distribuido en la aplicación**

**mencionada?**

**III. Cuadro comparativo**

**Complete la información de la tabla según modelo o concepto dado**

|  |  |  |
| --- | --- | --- |
| **Modelo / Concepto** | **Características** | **Ejemplo** |
| **RCP** | * Transpariencia de ubicación * Sincronismo Orientado a operaciones * Manejo de errores | Un programa en la computadora necesita calcular el valor de una función matemática compleja que solo está disponible en un servidor. Usando RPC, tu programa podría hacer una llamada a calcular\_funcion\_compleja(x), como si calcular\_funcion\_compleja fuera una función local. El sistema RPC se encarga de enviar la solicitud al servidor, esperar la respuesta y devolver el resultado a tu programa. |
| **RMI** | * Orientacion a objetos * Paso de objetos como parametros * Transpariencia de objetos | En un sistema bancario distribuido, un cliente podría querer ver su saldo. Usando RMI, el programa del cliente puede invocar el método obtenerSaldo() en un objeto CuentaBancaria que reside en el servidor del banco. RMI maneja toda la comunicación de red, permitiendo que el cliente interactúe con el objeto CuentaBancaria del servidor como si fuera un objeto local. |
| **Protocolo de Mensajeria** | * Asincronismo * Desacoplamiento * Enfoque en el intercambio de datos * Mayor flexibilidad | Un sistema de comercio electrónico recibe miles de pedidos por minuto. Usando un protocolo de mensajería, cada pedido es un mensaje que se envía a una cola. El servicio de procesamiento de pedidos lee los mensajes de la cola a su propio ritmo. Si el servicio de procesamiento de pedidos se cae, los mensajes no se pierden; simplemente permanecen en la cola hasta que el servicio vuelva a estar en línea |
| **Protocolo de Concenso** | * Acuerdo en sistemas distribuidos * Tolerancia a fallos * Atomismo * Equidad | En una base de datos distribuida, si dos nodos intentan actualizar la misma pieza de datos al mismo tiempo, el protocolo de consenso como Paxos o Raft se utiliza para asegurarse de que todos los nodos se pongan de acuerdo sobre qué actualización debe ser la correcta. El protocolo de consenso garantiza que el estado de la base de datos sea consistente en todos los nodos. |

**IV. Actividad de Análisis**

**Se establece como “Ideal de comunicación” la aplicabilidad de las**

**características de latencia, seguridad, tolerancia a fallos, ausencia de**

**perdida de mensajes; en base a ello:**

* **Elige un sistema distribuido que uses (por ejemplo, banca en línea, almacenamiento en la nube, Streaming de video, Streaming de música, Juegos en línea, sistemas de compras en línea); para con su elección explicar en que forma usted ve aplicada cada una de las características planteadas como “Ideal de comunicación” explicando la respuesta que tiene el sistema escogido a cada característica planteada.**

Consideramos que el streaming de música, en este caso, Spotify, es un excelente ejemplo de un sistema distribuido que usa las características de comunicación ideal.

Latencia (la velocidad): Es el tiempo que tarda una canción en empezar a sonar cuando le das a "Play". Spotify usa servidores que están cerca de ti para que la música empiece casi al instante, evitando que tengas que esperar.

Seguridad (la protección): Es cómo se protegen tus datos. Spotify cifra todo (tus contraseñas, los datos de tu tarjeta) para que nadie pueda verlos. Es como si enviaras tu información en una caja fuerte digital.

Tolerancia a fallos (la resistencia): Es la capacidad del servicio de no caerse. Si un servidor de Spotify falla, el sistema te conecta automáticamente a otro. Así, puedes seguir escuchando música sin interrupciones, como si nada hubiera pasado.

Ausencia de pérdida de mensajes (la fiabilidad): Esto asegura que todas tus acciones (como agregar una canción a una lista o pagar la suscripción) se registren correctamente. Si la red se desconecta por un segundo, el sistema se asegura de que la información no se pierda y se procese una vez que la conexión se restablezca.

**V. Escenario Sincronización**

**Supone que tiene conectados 3 nodos en un sistema distribuido de Comercio**

**Electrónico, cada nodo correspondiente con los servicios de: Pagos, Inventario, Clientes/Usuarios**

1. **Explique la necesidad de sincronización en dicho escenario.**

En un sistema de comercio electrónico, la sincronización es clave. Si el sistema no está sincronizado, un cliente podría comprar un producto que ya no existe en el inventario, o su pago podría no registrarse. La sincronización asegura que todos los datos (pagos, inventario, historial de clientes) sean siempre correctos y estén actualizados en todos los puntos del sistema, evitando errores y problemas de stock.

1. **Hipotéticamente, ¿Qué pasaría si uno de los nodos, se desincroniza en tiempo referente a los otros nodos?**

Si el nodo de inventario se desincroniza, el sistema podría vender un producto dos veces porque no tiene la hora correcta. Esto causa inconsistencias en el stock y problemas para los clientes, afectando el negocio.

1. **Para los protocolos de consenso estudiados: Paxos y Raft, ¿Cuál considera usted que sería mejor aplicar y por qué?**

Raft es la mejor opción. Es más simple de entender e implementar que Paxos porque usa un Líder que coordina todos los cambios. Esta simplicidad hace que sea más fácil mantener la sincronización y reduce errores, lo que es crucial en un sistema de comercio electrónico.