

Open Questions: Chordy

a) What are the pros and cons of a more frequent stabilization procedure?

Ventajas:

- Convergencia más rápida: Reduce el tiempo de integración de nuevos nodos o la reparación del anillo tras fallos.
- Consistencia mejorada: Al hacerse comprobaciones frecuentes, se minimizan las inconsistencias temporales en los punteros sucesor y predecesor.
- Recuperación rápida ante fallos: Detecta las salidas de nodos con mayor rapidez, lo que reduce los errores de *lookup/add*.

Desventajas:

- Aumento del tráfico de red: La estabilización requiere de intercambios de mensajes {request, Peer} y {status, Pred}. Una estabilización más frecuente podría saturar la red, al incrementar el envío de estos.
- Mayor exigencia a la CPU: Los nodos dedican más tiempo a procesar la lógica de estabilización y sus mensajes.
- Rendimiento reducido: La contención de recursos puede retrasar las operaciones como adds y *lookups* de pares de clave y valor.

b) Do we have to inform the new node about our decision? How will it know if we have discarded its friendly proposal?

Cuando un nuevo nodo se propone como predecesor, este envía un mensaje {notify, NewPeer} a un nodo existente. El receptor decide si acepta o descarta esta propuesta basándose en su predecesor actual y el rango de claves.

El nuevo nodo no recibe una respuesta explícita sobre la decisión del receptor. El nuevo nodo deduce el resultado mediante su propio proceso de estabilización:

Si se acepta, el nodo receptor actualiza su predecesor al nuevo nodo y envía un mensaje {status, Pred} a su sucesor durante la estabilización. Esta actualización se propaga y el nuevo nodo confirma su integración al recibir mensajes consistentes de predecesor/sucesor.

En caso de ser descartado, el nodo receptor no actualiza a su predecesor. El nuevo nodo, durante su estabilización, envía una solicitud {request, Peer} a su sucesor y recibe una respuesta {status, Pred}. Si esta respuesta indica un predecesor diferente, el nuevo nodo detecta que su propuesta no fue aceptada y se ajusta en consecuencia (por ejemplo, reintentar unirse a otra parte del anillo).

c) What would happen if we did not schedule the stabilization procedure?

Ocurriría lo siguiente:

- Estructura de anillo estática: El anillo se mantiene fijo y es incapaz de adaptarse a la incorporación de nuevos nodos, ni a la salida de los existentes. Los punteros de predecesor y sucesor no se actualizarían, congelando el sistema en su estado inicial.
- No hay detección ni reparación de fallos: Los fallos de los nodos no serían detectados, interrumpiendo la continuidad del anillo. Por ejemplo, si un sucesor cae, este no puede encontrar un nuevo sucesor, lo que interrumpe acciones como *adds* y *lookups*.
- Operaciones inconsistentes: Los *add* y *lookups* de pares clave-valor fallarían o devolverían resultados incorrectos, ya que los nodos no pueden redirigir las solicitudes a través de nodos nuevos o faltantes.
- Pérdida de tolerancia a fallos: Se pierde dinamismo y capacidad del sistema para autorrepararse y mantener la consistencia.

d) What will happen if a node is falsely detected of being dead (e.g., it has only been temporally unavailable)?

En la implementación de Chord *node3*, los nodos utilizan *erlang:monitor/2* de Erlang para detectar fallos, recibiendo un mensaje de "DOWN" cuando un proceso monitorizado (sucesor o predecesor) parece inactivo.

Se produce una detección falsa si un nodo no está disponible temporalmente (por ejemplo, por problemas de red).

Si el nodo inactivo es un sucesor, su predecesor adopta al siguiente nodo (el sucesor del sucesor) como su nuevo sucesor, eliminando así al nodo inactivo del anillo.

Si es un predecesor, el sucesor establece su predecesor en *nil*, a la espera de una nueva propuesta.

Cuando el nodo temporalmente inactivo se recupera, intenta reincorporarse al anillo contactando con su sucesor conocido u otro nodo. Sin embargo, su posición anterior podría estar ocupada, ya que el anillo se ha ajustado.

Por último, se puede producir una interrupción temporal, ya que hasta que la estabilización resuelva la reincorporación, los *lookups* o *adds* dirigidos al rango de claves afectado podrían fallar o generar resultados inconsistentes.

En resumen, los falsos positivos en la detección de fallos pueden interrumpir temporalmente el anillo y la consistencia de los datos, aunque la estabilización eventualmente corrige la estructura.

e) What node must send the confirmation message to the client when adding a key-value element to the store?

El nodo que debe enviar el mensaje de confirmación al cliente es el nodo responsable de la clave, una vez que haya añadido el par clave-valor a su *store* local y lo haya replicado correctamente en su sucesor. Esto asegura que, en caso de fallo del nodo principal, el sucesor podrá conservar los datos, ya que la confirmación se envía solo cuando nos aseguramos que en el sucesor se ha replicado correctamente.

f) How does the code deal with duplicated requests by the same client to add elements to the store?

Desde el punto de vista del cliente, el sistema **no previene el envío** de solicitudes duplicadas. Pero desde el punto de vista de almacenar los datos el código maneja las peticiones duplicadas de adición de elementos comprobando si la clave ya existe en el *store* local, y si es así, reemplazando el valor anterior. Por tanto, si el cliente envía la misma solicitud dos veces, y el mensaje original sí fue procesado, pero la confirmación se perdió, el nodo simplemente volverá a sobrescribir el mismo par {Key, Value}. Así no se generan duplicados en el *store* ni se rompe la consistencia y, por tanto, el cliente puede repetir la solicitud sin consecuencias negativas (es idempotente).

g) Outline the necessary changes in the code to implement finger tables.

El *add* debe modificarse para que, al agregar una llave, no se reenvíe forzosamente al sucesor inmediato, sino al sucesor más cercano en la *finger table*. Este será el sucesor inmediato si la clave se encuentra entre el ID de este nodo y el ID del sucesor inmediato, o si es el nodo más grande de la tabla de dedos menor que la clave.