El diseño de un sistema práctico para máquinas virtuales tolerantes a fallos

Enfoques para Servidores Tolerantes a Fallos

Enfoque primario/respaldo

- Se mantiene un servidor de respaldo en caso el servidor primario falle.
- El estado del servidor de respaldo se debe mantener casi igual que el del servidor primario. Una manera de hacerlo es comunicando los cambios en el estado del servidor primario al servidor de respaldo continuamente.

Problema:

- El ancho de banda para poder enviar el estado podría ser considerablemente grande.

Enfoque estado-máquina

- La idea es modelar los servidores como máquinas de estado determinístico que se mantienen el sincronía al iniciarlos en el mismo estado y asegurándonos que reciban las mismas consultas en el mismo orden.
- Dado que se pueden presentar operaciones no determinísticas se requiere coordinación extra para mantener la sincronía. La información extra es mucho menor que la cantidad de estado en cambio.

Problema:

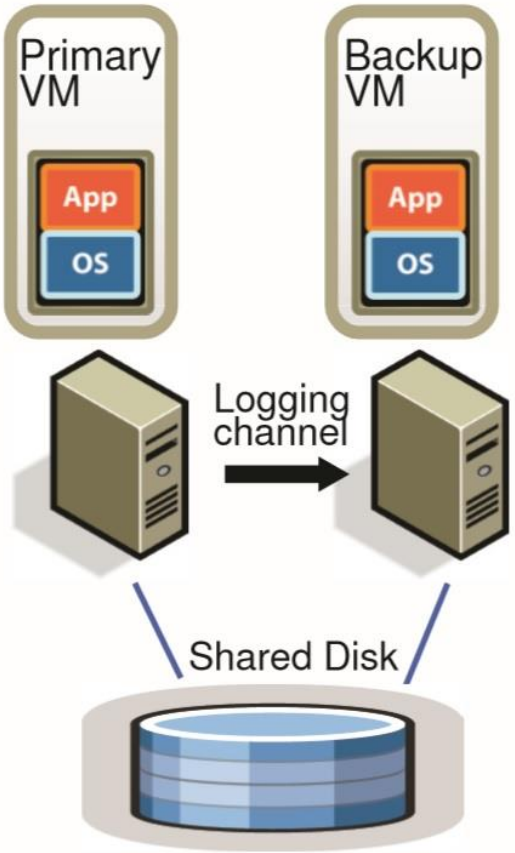
- Implementar la coordinación para asegurar la ejecución determinística de servidores físicos es difícil.

Enfoque Estado-Máquina en VM

- Una máquina virtual (VM) que corre con un hipervisor es una plataforma excelente para implementar el enfoque estado-máquina.
- Como el hipervisor tiene control total de la ejecución de la VM, el hipervisor puede capturar la información necesaria respecto a las operaciones no determinísticas de la VM primaria y replicar estas operaciones en la VM de respaldo.
- Debido a que este enfoque hace poco uso del ancho de banda, es posible una mayor separación física entre la VM primaria y la VM de respaldo.

Diseño Básico para la Tolerancia a Fallos (FT)

- Para una VM dada se ejecuta otra VM de respaldo en un servidor separado.
- Se dice que las dos VMs están en **lockstep virtual**.
- Los discos virtuales de las VMs están en un almacenamiento compartido.
- Solo la VM primaria anuncia su presencia en la red.
- Todas las entradas recibidas por la VM principal son enviadas a la VM de respaldo a través de una conexión de red conocida como **logging channel** (canal de registros).



- Los output de la VM de respaldo son descartados por el hipervisor.

- Para detectar si alguna de las VMs ha fallado se monitorea el tráfico del canal de registros como si fuera un ritmo cardiaco (**heartbeating**).

- Debemos asegurarnos que sólo una de las VMs toma el control de la ejecución, incluso si se pierde la comunicación entre ambas máquinas virtuales.

Implementación de la Repetición Determinística

Retos para replicar la ejecución de una VM:

- Capturar correctamente todas las entradas y no determinismo para asegurar la ejecución determinística de la VM de respaldo.
- Aplicar correctamente las entradas y no determinismo en la VM de respaldo.
- Procurar no afectar el rendimiento durante la replicación.
- Capturar y replicar los efectos secundarios indefinidos que se puedan producir en microprocesadores x86 al realizar operaciones complejas.

Repetición Determinística VMware

- La repetición determinística registra todas la entradas de una VM y todo el no determinismo posible asociada a su ejecución en un flujo de entradas de registro escrito en un archivo de registro.
- Se registra la información necesaria para replicar operaciones no determinística.
- Para eventos no determinísticos como el timer o interrupciones de IO, se registra la instrucción exacta en la que ocurrió el evento.
- La repetición determinística VMware implementa un registro de eventos y mecanismo de entrega de eventos eficiente.

Protocolo FT

- A fin de asegurarnos el conseguir una tolerancia a fallos, debemos aumentar un protocolo FT estricto en el logging channel.

El requisito principal es el siguiente:

***Requisito de Output**: si la VM de respaldo toma el control luego de una falla de la VM primaria [**failover**], la VM de respaldo se seguirá ejecutando de modo que sea enteramente consistente con todas las salidas emitidas por la VM primaria.*

A pesar que la VM de respaldo se ejecute un poco diferente de la primaria debido a los eventos no determinísticos, mientras se cumpla este requisito no se perderá ningún dato o estado externamente visible durante el failover, y los clientes no notarán interrupción o inconsistencia en el servicio.

- El requisito de output puede ser asegurado si se retrasa cualquier output externo hasta que la VM de respaldo haya recibido toda la información necesaria para repetir la ejecución hasta el punto de dicha operación output.
- Una condición necesaria es que la VM de respaldo haya recibido todas la entradas de registro generadas antes de la operación output.

Inconveniente:

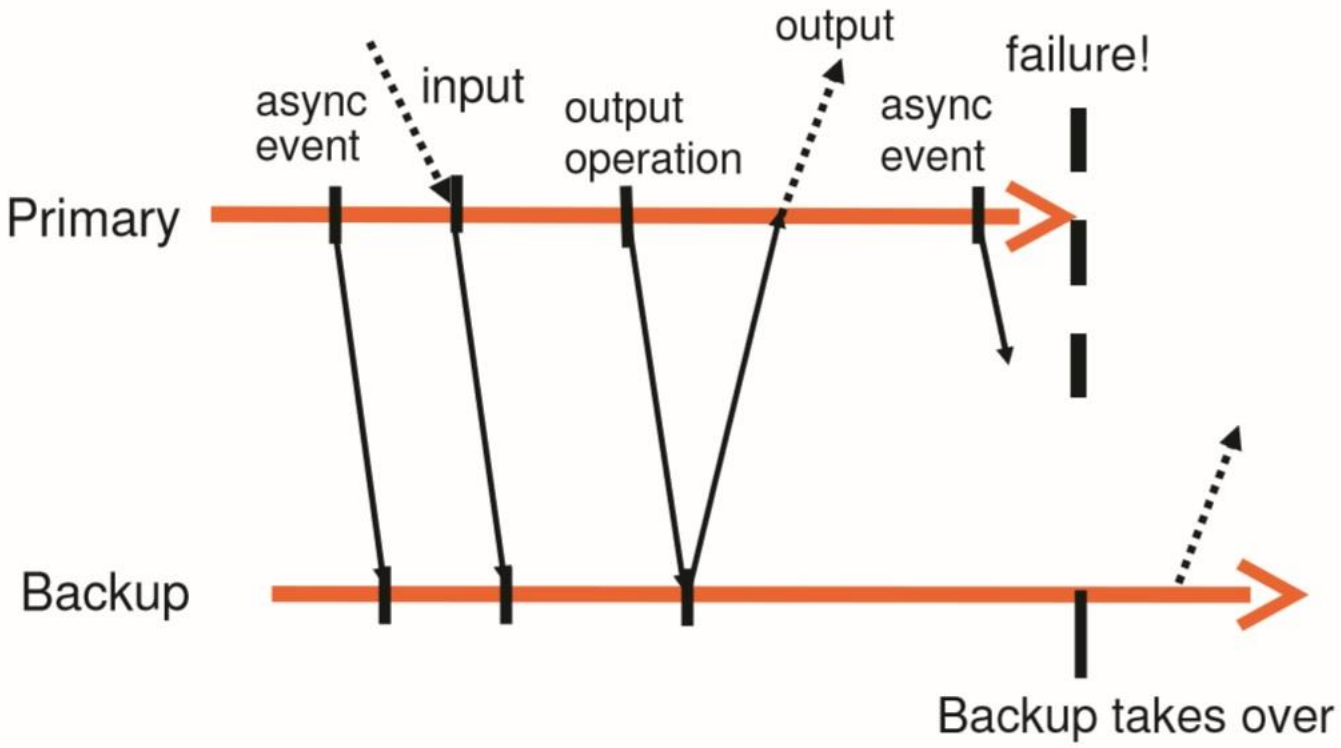
- En caso que la falla se de inmediatamente después que la VM primaria ejecuta la operación output, el respaldo debe saber que sólo debe replicando hasta dicho punto y tomar el control luego de ese punto.

Solución:

- Para cumplir el requisito de output debemos crear una entrada de registro especial por cada operación output.

***Regla de Output**: la VM primaria no puede enviar un output al mundo exterior hasta que la VM de respaldo haya recibido y reconocido las entradas de registro asociadas con la operación que generó el output.*

- Notar que la regla de output no menciona nada acerca de detener la ejecución de la VM primaria, sólo es necesario retrasar el envío del output.
- Debido a que los sistemas operativos realizan outputs de disco y red no bloqueados con interrupciones asincrónicas para indicar la terminación, la VM puede continuar su ejecución y no necesariamente es afectado de inmediato por el retraso del output.



- primary -> backup: entradas de registro
- backup -> primary: reconocimientos

Detectando y Respondiendo a Fallos

Las VM primaria y de respaldo deben responder rápidamente si la otra VM falla:

- Si la VM de respaldo falla, la VM principal deja de registrar sus entradas y comienza una ejecución normal.
- Si la VM principal falla, la VM de respaldo.

Debido al retraso en su ejecución, la VM de respaldo tendrá entradas de registro que han sido reconocidas pero no han llegado a ser consumidas aún. La VM de respaldo debe continuar su modo de repetición hasta que toda estas entradas hayan sido consumidas y luego recién comenzar a ejecutarse como una VM normal (se vuelve la VM primaria y ya no hay VM de respaldo).

- La nueva VM primaria producirá output al mundo exterior cuando el OS invitado realice operaciones output.

- Durante la transición a modo normal existirán algunas operaciones específicas de dispositivo necesarias para permitir que estos outputs se den correctamente.

- En particular, el VMware FT anuncia automáticamente la dirección MAC de la nueva VM primaria de tal manera que la red conozca en que servidor se encuentra la nueva VM primaria.

Detección de Fallos

- VMware FT utiliza el **heartbeating UDP** entre los servidores que están ejecutando las VMs tolerante a fallos para detectar cuando un servidor se haya malogrado.
- VMware FT monitorea el tráfico de registros que es enviado del VM primario al VM de respaldo, y los reconocimientos notificados por el VM de respaldo al VM primario.

Inconveniente:

- La pérdida de conexión de red entre los servidores también puede producir un error en de heartbeating aún cuando ambos servidores están funcionando ocasionando que se detecte un fallo cuando en verdad no ha ocurrido **(problema split-brain)**.

Solución:

- El uso del almacenamiento compartido para los discos de las VMs permite que se pueda realizar una operación TSL atómica cuando se detecta un error en el heartbeating de tal manera que sólo una de las VMs tomará el control luego de la falla.

Iniciando y Reiniciando VMs FT

- Una vez ocurrido un error y una de las VMs comienza su ejecución normal, se debe iniciar un nuevo respaldo el algún otro host.
- Por tanto, el mecanismo para iniciar una VM de respaldo debe ser utilizable para una VM primaria en cualquier estado, no necesariamente un estado inicial.
- Para VMware FT se puede adaptar la funcionalidad existente VMotion de VMware vSphere, la cual permite la migración de una VM en ejecución de un servidor a otro con disrupción mínima.
- Se puede crear una forma modificada del VMotion (FT VMotion) que crea un copia exacta de una VM de un servidor remoto en ejecución sin destruir el VM en el servidor local.

FT VMotion

- Clona una VM a un host remoto en lugar de migrarlo.
- Prepara el canal de registros (logging channel) y causa que la VM fuente entre en modo de registros como la VM primaria y que la VM destino entre en modo de repetición como el nuevo respaldo.
- La interrupción en la ejecución de la VM primaria para la operación de copiado es usualmente menos de un segundo.

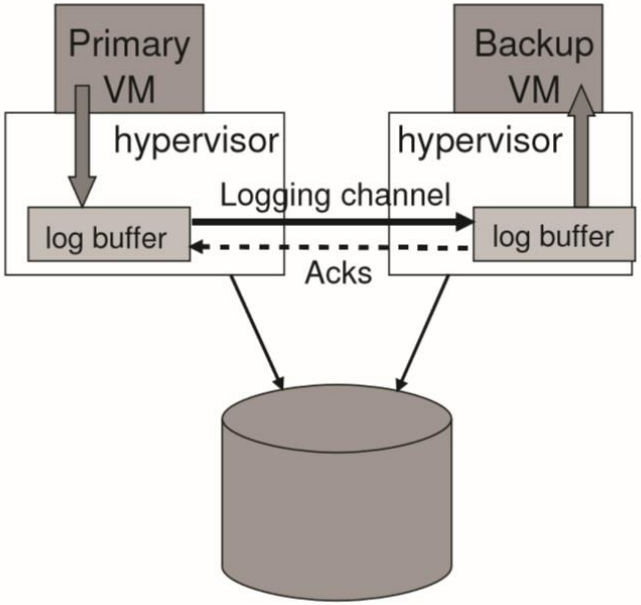
Debido a que las VMs FT se ejecutan en un clúster de servidores que tienen acceso a un almacenamiento compartido, las VMs se pueden ejecutar en cualquier servidor del clúster.

Escogiendo el servidor para el nuevo respaldo

- La flexibilidad mencionada recientemente permite restaurar la redundancia FT incluso cuando fallan más de un servidor.
- VMware vSphere implementar un **servicio de clústering** que mantiene la administración e información de recursos.
- Cuando una VM primaria requiere un nuevo VM de respaldo para restablecer la redundancia, la VM primaria notifica al servicio de clústering que requiere un respaldo.
- El servicio de clústering determina el mejor servidor para ejecutar la VM de respaldo basado en el uso de recursos y las otras restricciones e invoca al VMotion FT para crear el nuevo respaldo (notar que las VMs primaria y de respaldo no se pueden encontrar en el mismo servidor).

Manejando el Canal de Registros

- El hipervisor mantiene un buffer grande de entradas de registro para las VMs primarias y de respaldo: la VM primaria produce las entradas de registro en el buffer y la VM de respaldo las consume.
- El contenido del buffer de registro de la VM primaria es enviada a través del canal de registro lo más pronto posible y el contenido del buffer de registro de la VM de respaldo es leído ni bien llega.



Problemas:

- Si la VM de respaldo encuentra su buffer vacío, detendrá su ejecución hasta recibir una nueva entrada de registro. Aunque esto no genera inconvenientes a los clientes, provoca tiempos muertos.
- Si la VM primaria encuentra su buffer lleno, detendrá su ejecución hasta que una de las entradas de registro se haya retirado del buffer. Esta pausa puede afectar a los clientes de la VM pues no responderá a ninguna consulta hasta poder registrar su entrada y continuar su ejecución. La implementación debe ser diseñada para disminuir la posibilidad de que se llene el buffer de la VM primaria.
- Si el servidor donde se encuentra la VM de respaldo posee varias VMs, la ejecución de la VM de respaldo será lenta y por tanto consumirá las entradas lentamente ocasionando que se sature el buffer de la VM primaria, y además el proceso de recuperación luego de un error será más lento.

Solución:

- Para evitar que la VM primaria se ejecute mucho más rápido que la VM de respaldo, saturando la de respaldo, se envía información adicional junto a los envíos de entradas de registro y las notificaciones de reconocimiento de estas para determinar el retraso de ejecución en tiempo real entre la VM primaria y la de respaldo.
- Si el retraso de ejecución es significativo (más de 1 seg), se disminuye la velocidad de ejecución de la VM primaria notificando al scheduler para que le asigne una menor cantidad de CPU .
- Inversamente, si la VM de respaldo alcanza el ritmo de la VM primaria, se incrementa el límite de CPU de la VM primaria para que la VM de respaldo vuelva a tener un pequeño retraso.

Operaciones en VMs FT

La VM de respaldo debe saber cómo responder a las operaciones de control que puedan ser aplicadas a la VM primaria. Por ejemplo:

- Si la VM primaria es explícitamente apagada, la VM de respaldo debe detenerse sin actuar como si hubiera habido un error.
- Cualquier cambio en la administración de recursos de la VM primaria tiene que ser aplicado a la VM de respaldo.

Para este tipo de operaciones se debe enviar entradas especiales de control a través del canal de registros para efectuar las operaciones apropiadas en el respaldo.

VMotion en VMs FT

- La única operación que se puede realizar de manera independiente en la VMs primaria y de respaldo es el VMotion.
- Un VMotion en la VM primaria implica que la VM de respaldo tenga que desconectarse de la VM primaria fuente y reconectarse a la VM primaria destino.
- Para una VMotion normal se requiere que todas las IOs pendientes del disco sean completadas justo al momento en que ocurre el último switchover del VMotion. Esto se puede hacer fácilmente en una VM primaria esperando a que las IOs físicas se completen y entregando estas terminaciones a la VM.

Problema:

- Debido a que la VM de respaldo debe repetir la ejecuciones de la VM primaria y completar las IOs en el mismo punto de ejecución, no hay una manera fácil de causar que todos los IOs sean completados en cualquier punto requerido.
- La VM primaria podría estar ejecutando un trabajo en el que siempre haya vuelos de IOs de disco durante la ejecución normal.

Solución:

- Cuando la VM de respaldo se encuentra en el último switchover de un VMotion, solicita a la VM principal a través del canal de registros que complete temporalmente todas sus IOs de tal manera que la VM de respaldo repita esta operación en un solo punto de ejecución.

Problemas de Implementación para IOs de disco

Problema:

- Pueden ocurrir operaciones de disco simultáneas que acceden a la misma localización de disco y conllevar a no determinismo.

Solución:

- Detectar estas carreras de IO en disco y ejecutarlas secuencialmente de la misma manera en la VM primaria y de respaldo.

Problema:

- Una operación de disco puede competir con el acceso de memoria por parte de una aplicación u OS debido a que las operaciones de disco acceden directamente a la memoria de la VM vía DMA.

Solución:

- Preparar un protección de página temporal en las páginas objetivo de operaciones de disco. Se genera un trap (excepción o fallo) cuando la VM quiere acceder a dicha página. Sin embargo, cambiar las protecciones de la MMU en páginas es una operación cara.
- **Bounce buffers:** son buffers temporales del mismo tamaño que la memoria que quiere acceder la operación de disco. Una operación de lectura lee la información del buffer y una de escritura escribe sobre el buffer y posteriormente se copia el buffer al disco. Esto ralentiza las operaciones de disco, pero no hay una perdida de rendimiento notable.

Problema:

- Luego de que la VM de respaldo pasa a ser la nueva VM primaria, no hay modo en el que la nueva VM primaria se asegure si las IOs de disco fueron completadas satisfactoriamente o no. Debido a que las IOs de disco no son emitidos externamente en la VM de respaldo, no hay una terminación IO explícita para ellos.

Solución:

- Volver a emitir las IOs pendientes durante el proceso en el que la VM de respaldo pasa a ser la primaria. Como todas las IOs especifican qué bloque de memoria o disco será accedido, estas operaciones son idempotentes y pueden ser realizadas incluso si ya se han completado satisfactoriamente.

Problemas de Implementación para IOs de red

Problema:

- El hipervisor actualiza el estado del dispositivo de red de la VM asincrónicamente (e.g. los buffers de recepción pueden se actualizados directamente por el hipervisor cuando la VM está en ejecución), lo cual agrega no determinismo.
- A no ser que podamos asegurarnos que todas las actualizaciones ocurran en el mismo punto en el flujo de instrucciones de la VM primaria a la VM de respaldo, sus ejecuciones podrían divergir.

Solución:

- Se activa un trap al hipervisor, donde se pueden registrar las actualizaciones y luego aplicarlas a la VM, cuando llega un paquete al buffer y cuando se va a transmitir un paquete.

Alternativas de Diseño

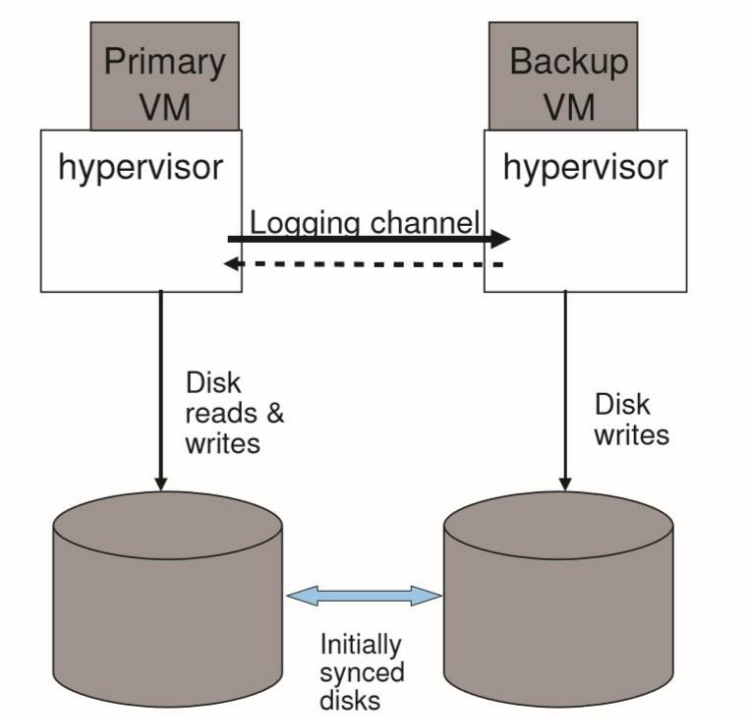
Disco No Compartido

Como el disco compartido es externo a las VMs primaria y de respaldo, las operaciones de escritura se consideran como comunicación al mundo externo por lo que sólo la VM primaria realiza estas operaciones. En este otro diseño:

- La VM de respaldo realiza todas las operaciones de escritura sobre su propio disco virtual, y por tanto su disco virtual se mantiene el sincronía con el de la VM primaria.
- Las escrituras de disco de la VM primaria no tienen que ser retrasadas de acuerdo a la regla de output pues los discos virtuales son considerados parte del estado interno de cada VM.

Desventajas:

- Los discos tienen que ser sincronizados explícitamente cuando se habilita por primera vez la tolerancia a fallos.
- Los discos se pueden perder la sincronización luego de una falla por lo que deben ser re sincronizados explícitamente luego de habilitar una nueva VM de respaldo.
- Se pierde el almacenamiento compartido que se usaba en el problema split-brain.



Ejecutar Lecturas de Disco en la VM de Respaldo

Como la lectura de disco es considerado un input, la VM de respaldo no nunca lee de su disco virtual y los resultados de la lectura le son enviados por el canal de registro. Este otro diseño puede disminuir el tráfico del canal de registro.

Sutilezas:

- La VM de respaldo se ejecutaría más lento.
- Se requiere trabajo adicional si ocurre alguna falla en la operación de lectura en uno de las VMs pero no en la otra.
- Si se aplica esto al diseño de disco compartido, la VM primaria tendrá que retrasar sus operaciones de escritura hasta que la VM de respaldo haya completado las operaciones de lectura previamente hechas.

Conclusiones

- El diseño se basa en replicar la ejecución de una VM primaria a través de una VM de respaldo que pueda tomar el control en caso la VM primaria experimente alguna falla.
- Debido al bajo uso del ancho de banda para los registros, se puede implementar configuraciones en las que la VM primaria y de respaldo se encuentran separadas por largas distancias (1-100 Km).
- VMware FT ofrece protección incluso cuando todo un lugar experimenta una falla.
- El flujo de registros se puede comprimir con técnicas simples de compresión para disminuir el ancho de banda requerido.