**ALTA DISPONIBILIDAD**

Elmer Santisteban & Alicia Villón.

Octubre 2016.

Universidad Científica del Sur.

Base de Datos Relacionales

**Tabla de Contenidos**

**¿Qué es la Disponibilidad?**

La disponibilidad es una de las características de las arquitecturas empresariales que mide el grado con el que los recursos del sistema están disponibles para su uso por el usuario final a lo largo de un tiempo dado. Ésta no sólo se relaciona con la prevención de caídas del sistema (también llamadas tiempos *fuera de línea*, *downtime* u *offline*), sino incluso con la percepción de "caída" desde el punto de vista del usuario: cualquier circunstancia que nos impida trabajar productivamente con el sistema – desde tiempos de respuesta prolongados, escasa asistencia técnica o falta de estaciones de trabajo disponibles – es considerada como un factor de baja disponibilidad.

**¿Cómo medimos la disponibilidad?**

De primera instancia, todo sistema debe tener establecido un Acuerdo de Nivel de Servicio (Service Level Agreement – SLA) que defina cuánto tiempo y en qué horarios debe estar en línea. En el caso de aplicaciones de baja criticidad, dicho SLA puede ser de 8×5 horas a la semana excluyendo días festivos; para sistemas con mayor criticidad como una red de cajeros automáticos se tienen niveles de servicio que alcanzan las 24 horas al día, los 365 días del año. Así entonces, suponiendo un sistema con un SLA de 24×365 podríamos calcular su disponibilidad de la siguiente manera:

Disponibilidad = ((A – B) /A) x 100 por ciento)

Donde:

**A** = Horas comprometidas de disponibilidad: 24 x 365 = 8,760 Horas/año.

**B** = Número de horas fuera de línea (Horas de "caída del sistema" durante el tiempo de disponibilidad comprometido). Por ejemplo: 15 horas por falla en un disco; 9 horas por mantenimiento preventivo no planeado.

así entonces:

Disponibilidad = ((8,760 – 24) /8,760) x 100 por ciento) = 99.726%

Cuando se realicen negociaciones para definir objetivos de disponibilidad con los usuarios, es necesario hacerlos consientes de las implicaciones técnicas y económicas, como se muestra en la siguiente tabla:

**¿Cómo medimos la disponibilidad?**

De primera instancia, todo sistema debe tener establecido un Acuerdo de Nivel de Servicio (Service Level Agreement – SLA) que defina cuánto tiempo y en qué horarios debe estar en línea. En el caso de aplicaciones de baja criticidad, dicho SLA puede ser de 8×5 horas a la semana excluyendo días festivos; para sistemas con mayor criticidad como una red de cajeros automáticos se tienen niveles de servicio que alcanzan las 24 horas al día, los 365 días del año. Así entonces, suponiendo un sistema con un SLA de 24×365 podríamos calcular su disponibilidad de la siguiente manera:

Disponibilidad = ((A – B)/A) x 100 por ciento)

Donde:

**A** = Horas comprometidas de disponibilidad: 24 x 365 = 8,760 Horas/año.

**B** = Número de horas fuera de línea (Horas de "caída del sistema" durante el tiempo de disponibilidad comprometido). Por ejemplo: 15 horas por falla en un disco; 9 horas por mantenimiento preventivo no planeado.

así entonces:

Disponibilidad = ((8,760 – 24)/8,760) x 100 por ciento) = 99.726%

Cuando se realicen negociaciones para definir objetivos de disponibilidad con los usuarios, es necesario hacerlos concientes de las implicaciones técnicas y económicas, como se muestra en la siguiente tabla:

|  |  |  |  |
| --- | --- | --- | --- |
| **Disponibilidad (%)** | **Tiempo offline/año** | **Tiempo offline/mes** | **Tiempo offline/día** |
| 90% | 36.5 días | 73 hrs | 2.4 hrs |
| 95% | 18.3 días | 36.5 hrs | 1.2 hrs |
| 98% | 7.3 días | 14.6 hrs | 28.8 min |
| 99% | 3.7 días | 7.3 hrs | 14.4 min |
| 99.5% | 1.8 días | 3.66 hrs | 7.22 min |
| 99.9% | 8.8 hrs | 43.8 min | 1.46 min |
| 99.95% | 4.4 hrs | 21.9 min | 43.8 s |
| 99.99% | 52.6 min | 4.4 min | 8.6 s |
| 99.999% | 5.26 min | 26.3 s | 0.86 s |
| 99.9999% | 31.5 s | 2.62 s | 0.08 s |

Disponibilidad para un sistema 24×7 y tiempos de caída permitidos.

Estos números (especialmente aquellos que pasan de la marca del 99.5% de disponibilidad) son difíciles de alcanzar, ya que es necesario poder recuperarse ante caídas del sistema de manera transparente. La capacidad e intervalo de tiempo necesarios para recuperarse ante tal eventualidad son directamente dependientes de:

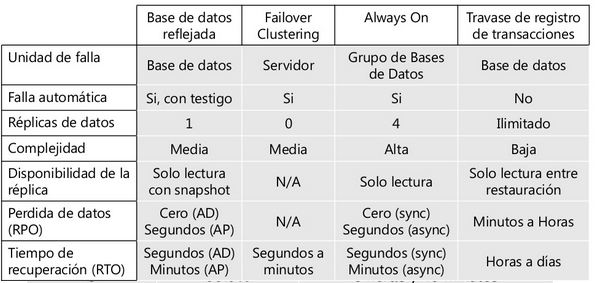
• La complejidad del sistema.   
• La severidad del problema.   
• La disponibilidad del personal de soporte.   
• La madurez en materia de administración del sistema y sus operaciones.   
• Otros factores técnicos o de gestión: falta de refacciones, fallas en la cadena de   
 escalamiento, etc.

**Tecnologías de alta disponibilidad**

SQL Server ofrece varias opciones para crear alta disponibilidad para un servidor o una base de datos. Entre las opciones de alta disponibilidad figuran las siguientes:

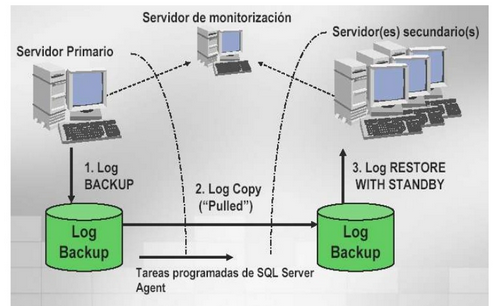
**Trasvase de registros**

* Creación de reflejo de base de datos (Mirror).
* Replicación transaccional.
* Grupos de disponibilidad AlwaysOn
* Instancias de clúster de conmutación por error de AlwaysOn.



**Trasvase de registros**

Datos que puede ser traídos online en caso de falla del sistema. Una base de datos completa es restaurada a un servidor secundario, y luego los archivos de registro de transacción de la base de datos primaria son aplicados a la base de datos secundaria. Este es un costo relativamente bajo a la solución de la disponibilidad de datos.



**Fortalezas y debilidades de trasvase**

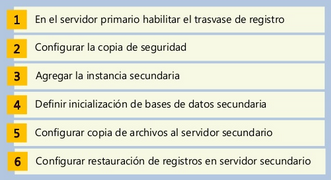
**1.- Fortalezas**

* Maduro y estable.
* Múltiples copias de trabajos de respaldo y restauración.
* Simple de configurar y gestionar.
* No requiere HW especial.
* Las bases de datos secundarias pueden ser consultadas, pero no modificadas.
* Alertas en caso el trasvase no se complete en un umbral de tiempo.
* Alerta en caso el servidor principal deje de funcionar (monitor).

**2.- Debilidades**

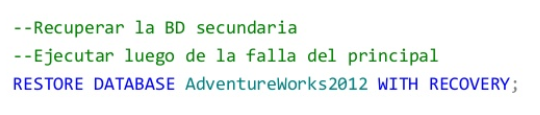
* Fallo de cliente manual.
* Configuración por base de datos.
* No protege bases de datos del sistema.

**Configuración de trasvase de registro**



**Falla manual de trasvase de registro**

En caso de falla en el servidor principal, recuperar manualmente la base de datos en el servidor secundario:



Las aplicaciones deben ser redireccionadas manualmente hacia el servidor secundario.

**Creación de reflejo de base de datos (mirror).**

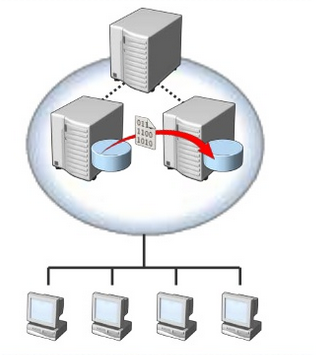
Solución de software para incrementar la disponibilidad de la base de dato.

Mejora el nivel de disponibilidad que tenían las versiones anteriores de SQL Server y provee una alternativa fácil a clúster. Provee una solución tolerante a fallas casi instantánea para las bases de datos. Permite mantener una copia actualizada de una base de datos en un servidor aparte para conmutación en caso de falla del servidor principal.

¿Cómo funciona reflejo de base de datos?

Mantiene un servidor secundario en standby, el cual tiene una copia de la base de datos principal. Si el servidor principal falla, las aplicaciones redireccionan al servidor en secundario.

La conmutación es instantánea en solo unos segundos.

Los servidores que tienen estas bases de datos principal y reflejo se los llama partner servers. En el caso de falla del servidor principal, la sesión del cliente será perdida. El cliente intentara reconectarse al servidor principal. Pero si falla, automáticamente redireccionará la conexión al servidor reflejo

**Fortalezas y debilidades de reflejo**

**Fortalezas**

* Fallo de base de datos y cliente rápido y automático.
* No muy complicado de configurar y administrar.
* No se requiere HW especial

**Debilidades.**

* Descontinuado.
* Requiere tres servidores en el modo de alta disponibilidad.
* No se puede consultar la base de datos reflejada.
* Configuración por base de datos.
* No protege bases de datos del sistema

**Replicación transaccional**.

Los cambios son aplicados a los suscriptores poco después de que ocurren

El proceso implica:

* Tomar una instantánea inicial de los datos
* Cambios en los datos son enviados al suscriptor conforme ocurren

Los cambios son aplicados en el suscriptor en el mismo orden en el que ocurren

**¿Cómo funciona replicación transaccional punto a punto?**

Opción configurable como parte de replicación transaccional. Típicamente usado con servidores autónomos.

Los conflictos son detectados, pero necesitan ser evitados.

Qué es replicación transaccional?




Los cambios son aplicados a los
suscriptores poco después de
que ocurren
El proces...

Cómo funciona replicación transaccional punto a
punto?



Opción configurable como parte de replicación
transaccional
Tí...

**Fortaleza y debilidades de replicación**

**Fortalezas**

* Maduro y estable Opciones flexibles de configuración.
* No requiere de HW especial.
* Puede descentralizar los centros de datos.
* Las bases de datos secundarias pueden ser consultadas y modificadas.

**Debilidades**

* Fallo de cliente de forma manual.
* Protege solo datos, no protege esquemas ni tablas.
* Puede ser difícil reparar.
* Configuración por base de datos.
* Conflictos de datos.

Cómo configurar replicación punto a punto
Configurar la distribución en los servidores
Crear una publicación en el primer ...**Configuración de Replicación punto a punto**

**Consideraciones**

Replicación transaccional punto a punto solo disponible en edición Enterprise.

Idealmente los servidores que conforman cada nodo deben ser idénticos.

Se recomienda que cada nodo tenga su base de datos de publicación, para eliminar un único punto de falla.

Un objeto puede estar incluido en una sola publicación transaccional punto a punto.

No es recomendable usar columnas de identidad. Manualmente se deben reasignar rangos de semilla.

**Clúster para tolerancia a fallos**

Reserva en caliente (Hot Standby).

Soporte de servidor completo y alta disponibilidad en caso de falla de hardware o para por mantenimiento. En caso de falla, el sistema operativo y SQL Server trabajan juntos para proveer un sistema automatizado para fallas. Combinación de uno o más nodos (servidores) con almacenamiento de disco compartidos (SAN)

Servidor virtual: Combinación de recursos, junto con sus nombres de red y dirección IP. Los clientes se conectan a un servidor virtual alojado en uno de los nodos del clúster.

El servidor virtual SQL aparece en una red como si fuese una sola computadora, pero provee tolerancia a fallas a nivel de instancia de un nodo hacia otro, cuando alguno de ellos deja de estar disponible.

Cuando se configura clúster, solo se puede tener una instancia de SQL Server en el servidor virtual

**Componentes del clúster**

**Servidor Virtual**

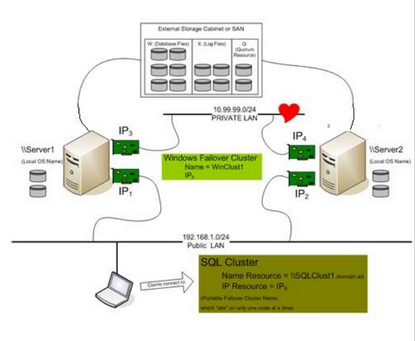
Desde la perspectiva del cliente/aplicación: nombre de servidor o direcciones IP que se usan para acceder

**Componentes de hardware:**

* Nodos del clúster.
* Señal interna (heartbeat).
* Red Interna.
* Red externa.
* Arreglos de discos compartidos del clúster.

**Cómo funciona clúster para tolerancia a fallos?**

* Múltiples nodos (hasta 8).
* Requisitos especiales de almacenamiento.
* Nodos pueden extenderse a subredes.
* SQL Server instalado en el clúster de Windows.



**Fortalezas y debilidades de clúster**

**Fortalezas**

* Estable y maduro.
* Protege la instancia completa y BDs del sistema.
* Más fácil de instalar con la herramienta Windows Cluster Validation.

**Debilidades**

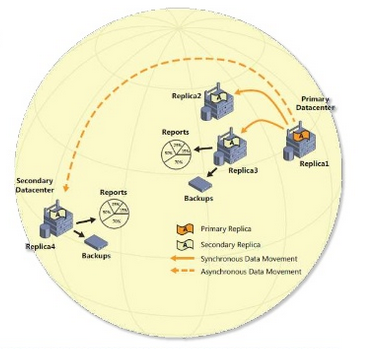
* Requerimientos especiales de HW.
* Costoso.
* Requiere mayor conocimiento y experiencia.
* No duplica datos de bases de datos; clúster debe ser combinado con otra tecnología de AD que duplique los datos.

**Alta disponibilidad con alwayson**

Grupos de disponibilidad AlwaysOn es una solución de alta disponibilidad y recuperación ante desastres de nivel empresarial presentada en SQL Server 2012 que permite maximizar la disponibilidad para una o varias bases de datos de usuario. Grupos de disponibilidad AlwaysOn necesita que las instancias de SQL Server se encuentren en nodos de Clústeres de conmutación por error de Windows Server (WSFC).

**Cómo funciona AlwaysOn**

* Conjunto de réplicas primarias de lectura y escritura.
* Conjunto de hasta 4 réplicas secundarias de solo lectura.
* Basado en Windows Server Failover Cluster (WSFC).
* Modos síncrono y asíncrono.



**Fortalezas y debilidades de Always On**

**Fortalezas**

* Fallo automático de base de datos y cliente de forma automática
* Configuración flexible con múltiples réplicas para fallo
* Acceso de solo lectura a las réplicas
* Se puede respaldar réplicas para descargar trabajo
* No se requiere HW especial

**Debilidades**

* Complejo de implementar.

**Configuración**

**Cómo configurar grupos de disponibilidad
Habilitar QlwaysOn en cada instancia
Iniciar el asistente para nuevo grupo de alt...**

**Consideraciones**

Cada servidor que contiene un grupo de disponibilidad debe ser un nodo de conmutación por error de Windows (ediciones Enterprise y Datacenter).

Todas las instancias deben tener la misma intercalación.

**Conmutación de AlwaysOn**

Tres modos de conmutación

* Automática (sin pérdida de datos)
* Planeada manual (sin pérdida de datos)
* Forzada manual (con posible pérdida de datos)

Esto dependerá de la conexión de la réplica

* Replica de actualización síncrona
* Replica de actualización asíncrona

En la conmutación, la réplica de destino toma el rol de réplica primaria, recupera la base de datos y la pone en línea.

La réplica primaria inicial, si se recupera, cambia a rol secundario

BIBLIOGRAFÍA

* Microsoft Virtual Academy

• <https://www.microsoftvirtualacademy.com>

* Blog dbLearner

• <http://dblearner.com/>

* SQL Server Central

• <http://www.sqlservercentral.com/>

* Libros en línea

• <http://msdn.microsoft.com/es-pe/library/ms130214.aspx>

* Página principal de SQL Server

• http://www.microsoft.com/sqlserver