Segurança de dados

Aula 11 – Alta disponibilidade - parte 2

Gustavo Bianchi Maia

gustavo.maia@faculdadeimpacta.com.br

Sumário

- Revisão
  - Termos da Segurança da Informação
  - Introdução à Alta Disponibilidade
- Técnica de Replicação
- AC4

Revisão Segurança da Informação

- Quais as fases do ciclo de vida da informação ?
- Quais os pilares da segurança da informação que garantem a confiabilidade da informação ?
- Defina:
  - Vulnerabilidade
  - Ameaça
  - Risco
  - Incidente
  - Dano
  - Recuperação
- Explique os componentes de um plano de gestão de continuidade de negócio
  - **PGC**: Plano de gestão de crises
  - **PCO**: Plano de continuidade operacional / Plano de contingência
  - **PRD**: Plano de recuperação de desastres

Revisão Alta Disponibilidade

- Por que precisamos de Alta Disponibilidade ?
- O que significa um sistema ser “tolerante à falhas” ?
- Explique os seguintes termos:
  - SPOF
  - MTTF
  - MTTR
  - MTBF
  - SLA
  - SLO
  - Uptime
  - Downtime
- Por quê Backup / Restore NÃO é considerada uma técnica de alta disponibilidade ?

Mecanismos de Alta disponibilidade

Na disciplina de Segurança de Dados, as técnicas para gerenciamento de contingência (ou redundância) associadas à alta disponibilidade, ou HA do termo em inglês High Availability serão:

- **[LS]** - Log Shipping - Técnica de envio de logs transacionais via BACKUP/RESTORE entre servidores.
- **[Rep]** - Replicação (Transactional Replication without updatable subscribers) - Técnica de envio de informações (artigos) para outros servidores (assinantes) coordenados por uma central de distribuição
- **[PtP]** - Replicação (Peer-to-peer) - Técnica de compartilhamento de informações entre múltiplos servidores, cada um tendo o papel de publicador, distribuidor e assinante dos dados do conjunto.
- **[Mirroring]** - Espelhamento - Técnica de sincronização entre bancos de dados.
- **[Cluster]** - Clustering - Técnica de utilização de múltiplos nós (instâncias) utilizando [preferencialmente] um Shared Disk Array
- **[AG]** - AllWaysOn - Técnica de sincronização entre bancos pela formação de grupos de disponibilidade (AG = Availability Groups)

Por quê Backup / Restore NÃO é considerada uma técnica de alta disponibilidade ?

Mecanismos de Alta disponibilidade

Lista de siglas ou termos que serão utilizadas em nossas aulas, segue uma revisão / apresentação:

DR = Recuperação de desastres, reversão de danos ou prevenção de perda de dados

HA *= High Availability (Alta disponibilidade), redução ou eliminação do tempo de DownTime.*

MTTR *= Tempo de reparo médio, ou o tempo de se colocar em prática uma solução de DR.*

WSFC *= Windows Server Failover Cluster (cluster à nível sistema operacional)*

FCI *= Failover Cluster Instance (cluster das instâncias do MSSQL)*

SDA *= Shared Disk Array (Conjunto de Discos Compartilhados)*

Backup INIT *= Rotina de Inicialização, BACKUP do principal seguido de RESTORE nas réplicas*

DNS *= Registro de aliases na rede –Uso do DNS ao invés de IPs ou Nomes físicos para localização de servidores*

FIREWALL *= Configurações ou necessidade de manutenção de portas de acesso na rede*

RAID *= Solução a nível dos discos, normalmente utilizados para garantir segurança e/ou velocidade.*

Failover *= técnica de promoção um servidor secundário (ou réplica) à primário.*

Principal = Servidor principal utilizado pelas aplicações da empresa (nó ativo, status = RECOVERY)

Réplica = Nome dado os múltiplos servidores secundários que não são o principal, capazes [ou não] de failover.

Mecanismos de Alta disponibilidade

Após a apresentação de todas as técnicas, você deverá apresentar a seguinte tabela preenchida conforme orientações:

Técnica	Principal Finalidade	Janela de perda	Tipo de Failover	Qtde Réplicas	Reversibilidade	Status Réplica	Nível Sincronização	Setup / Recomendações (Liste os números)	Edição mínima	Versão mínima	Disp. Azure
LS											
Rep											
PtP											
Mirror											
WSFC											
AG											

Mecanismos de Alta disponibilidade

Onde:

- **Principal Finalidade.**
  - Propósito ou 'ponto forte' da técnica, para DR ou HA (mesmo atendendo às duas, uma ainda é a principal)
  - **Opções: DR** (evitar perda de dados, pouca perda), **HA** (downtime zero, aplicação não 'sentir' a queda)
- **Janela de perda / tempo de cópia / tipo de sincronização.**
  - Tempo em que as informações do principal levam para serem transferidas para o secundário. Durante esta janela, tais informações estão vulneráveis, ou seja, numa eventual crise, elas podem ser perdidas.
  - **Opções: Min**utos**, Seg**undos**, Zero** (sem perda, ou sem necessidade de cópia de dados)
- **Tipo de Failover** (recomendação)
  - Capacidade de promoção do servidor secundário à primário. Assume-se que, se há risco de perda de dados, é necessário a intervenção humana para realizar ou autorizar um failover (ou seja, se há perda de dados, não é recomendado deixar o sistema decidir pelo failover, devido aos falsos positivos como: erro de rede, latência)
  - **Opções: Manual** (pois há riscos de perdas de dados)**, Auto**mático (não há riscos de perda de dados)
- **Quantidade de réplicas / servidores secundários.**
  - Número de réplicas que posso configurar para que, em caso de falha, um deles possa assumir o papel de primário em caso de failover. Ou se possível, utilizar para me conectar no SGBD.
  - **Opções: Um** (um principal, limite de uma única réplica)**, N** (um principal, possibilidade de múltiplas réplicas)
- **Reversibilidade**
  - Promoção do secundário à primário (failover) e depois de volta à secundário (outro failover) sem necessidade de re-configuração (Backup INIT) ou risco de existirem dois servidores que se acham principais.
  - **Opções: Total** (Ida e volta, à vontade)**,** Só **Ida** (feito o failover não há ou não é recomendado o retorno)

Mecanismos de Alta disponibilidade

Onde:

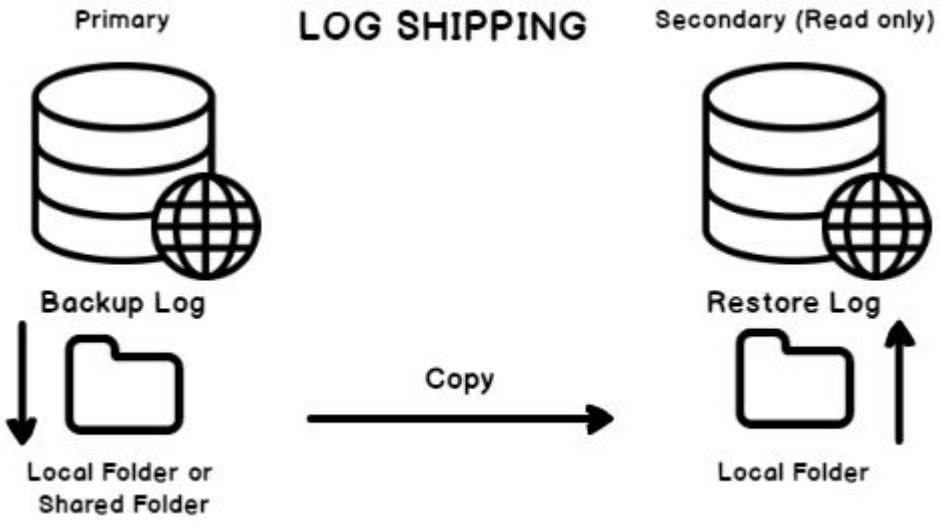
- **Status da réplica / Usos extras para o servidor Secundário**
  - Capacidade de uso das réplicas para algo mais além da função de alta disponibilidade, afetado diretamente pelo tipo de recuperação (ou status) das réplicas.
  - **Opções: Não** (NORECOVERY)**, RO** - Read Only (STANDBY)**, RW** - Read Write (RECOVERY)
- **Nível de sincronização:**
  - Nível em que a técnica é aplicada, limita a quantidade de informações que pode ser sincronizada.
  - **Opções: Inst**ância (Tudo o que está na instalação)**, D**ata**B**ase (Banco de dados)**, Obj**etos (tabelas, visões)
- **SETUP / Dificuldade de implementação / pré-requisitos / Recursos extras.**
  - Configurações, etapas necessárias ou altamente recomendadas para a implantação desta técnica.
  - **Opções múltiplas**(anote os números/ordem, ex:1,3,5): **WSFC**[1], **FCI**[2], **SDA**[3], **Backup INIT**[4], **DNS**[5], **RAID**[6], **Firewall**[7]
- **Edições / Licenciamento mínimo necessário.**
  - Capacidade de execução em uma determinada Edição do SQL Server
  - **Opções: Express, Compact, Web, Standard, Enterprise**
- **Versões / Disponibilidade mínima nas versões:**
  - Versões inicial em que esta técnica se tornou disponível.
  - **Opções: 2000, 2005, 2008, 2008R2, 2012, 2014, 2016, 2017, 2019**
- **Disponibilidade no Azure:**
  - Possibilidade de utilizar esta técnica em um banco de dados localizado no Azure (Azure Cloud Databases).
  - **Opções: Não** (Não disponível)**, Parcial** (uso limitado, ex:só importação)**, Sim** (uso normal, como uma instância local)

Log Shipping - Revisão

Esta é a técnica de realizar backups em um servidor e automaticamente restaurá-los em outro(s) servidor(es) em uma janela muito pequena (minutos), criando assim um servidor “morno”.

São necessários bancos diferentes para a realização de log shipping, mas por segurança é recomendado que o restore seja realizado em um servidor físico, geograficamente isolado do principal.

Atua a nível de banco de dados, que devem estar em modo de recuperação FULL.
O(s) servidor(es) secundário(s) pode estar em modo somente leitura.
Posso ter um servidor com o papel de monitoria, porém, ainda se recomenda failover manuais.



Fonte: SQL Server Transaction Log and High Availability Solutions (sqlshack.com)

Replicação

A Replicação é uma técnica que pode ser comparada à uma editora.

Suponha que existam milhares de escritores enviando artigos para serem publicados para uma revista, e que dentre todos, o editor deve selecionar quais ele deseja publicar na edição deste mês, fechando então o que chamamos de ‘publicação’.

O conteúdo é então encaminhado à distribuidora, que é responsável por gerenciar transporte da informação até seu ponto final.

Do outro lado temos os assinantes, pessoas que desejam receber tal conteúdo através dos diversos meios de distribuição.

Replicação

No SQL Server esta técnica não é muito diferente.

Tabelas, bancos, arquivos, relatórios etc são os artigos que podem ser selecionados para publicação. (publishers)

Uma central de distribuição precisa ser criada para gerenciar o que será publicado para quem, este serviço pode estar localizado em um servidor à parte, ou dependendo do tipo de modelo escolhido, nos servidores de publicação ou assinatura. (distributor)

Os assinantes são outros servidores, bancos, tabelas etc, onde a informação será enviada pela central de distribuição. (subscriber)

Um detalhe adicional, é que na “editora” do SQL Server, o assinante pode corrigir o texto e devolver à editora, e dependendo do tipo de replicação escolhida, ainda distribuir tal conteúdo para todos os outros assinantes.

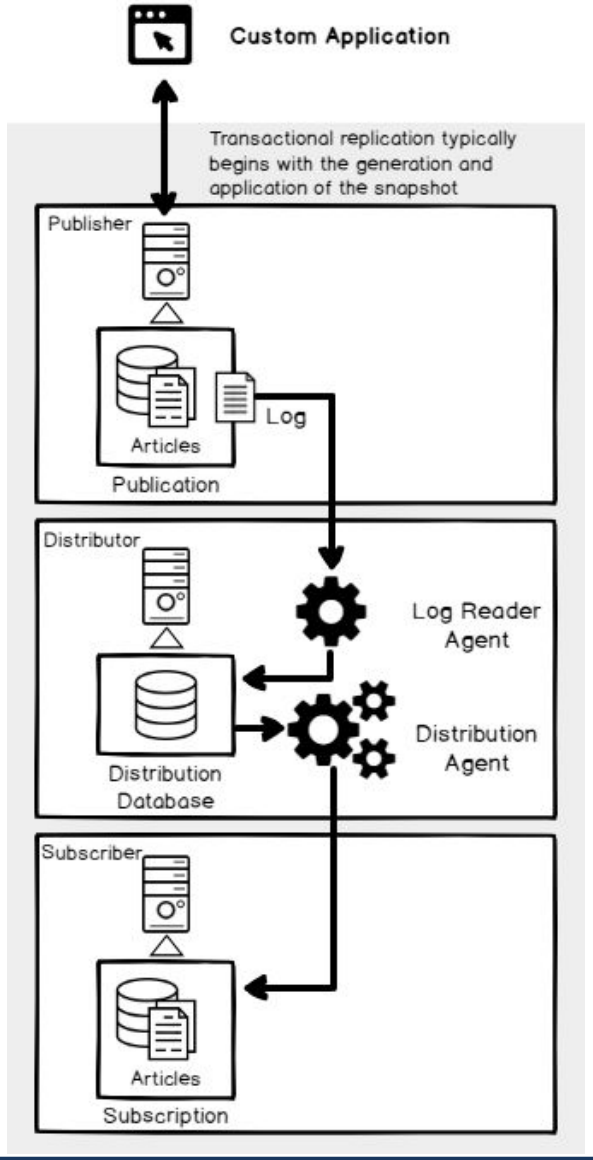
Replicação

É controlado por 3 agentes:

O “SQL Server Snapshot Agent” - Prepara a fila com os objetos / artigos que serão replicados (é o equivalente à realizar os backups dos objetos selecionados).

O “Log Reader Agent” é responsável por monitorar o log transacional de publicação e copiar estas transações do log transacional para o banco de dados da distribuição que serão copiadas para os assinantes pelo agente de distribuição.

O “Distribution Agent” é responsável por copiar os snapshots e os logs cumulativos para os assinantes.



Fonte: SQL Server Transaction Log and High Availability Solutions (sqlshack.com)

Replicação - Tipos de Replicação

Snapshot replication

Esta técnica tira uma 'foto' dos artigos (tabelas/bancos/etc) e os envia aos assinantes, uma vez que o conteúdo foi consumido, a 'foto' original é descartada, por não se tratar do dado real.

É muito recomendada para cenários que não precisam de atualização contínua, ou que contenha dados estáticos.

Transactional replication

Esta técnica não aborda os dados, mas sim as transações (transactional log) sobre os artigos (tabelas), enviando-os em tempo real para os assinantes tentando mantê-los atualizados o mais rápido possível.

Um primeiro snapshot é realizado para a sincronização inicial, depois disso, todo e qualquer comando que gera log será enviado aos assinantes para manter a sincronia da base.

A sincronização rápida é justamente a grande vantagem desta técnica.

Replicação - Tipos de Replicação

Merge replication

Esta técnica é comumente utilizada quando não há comunicação constante entre assinante e distribuidor, permitindo trabalho independente até que os dados sejam sincronizados.

Ela também precisa de um snapshot inicial para configuração, a partir dai, as mudanças são rastreadas com a utilização de triggers.

Peer-to-peer replication

É um subtipo da replicação transacional, onde todos os ‘nós’ são editores e assinantes ao mesmo tempo, a informação nem precisa estar toda completa em cada uma das bases pois todas elas ‘conversam’ entre si para sincronizar e distribuir a informação.

Replicação - Modelos de Replicação

Single publisher, one or more subscribers

É a topologia mais simples, com um único servidor publicando as informações e pelo menos um assinante, muito usado quando se deseja manter um servidor 'quente' com alta disponibilidade ou espalhar os dados de uma única central (matriz).

Multiple publishers, Single Subscriber

Utilizada em aplicações de ponto único (do inglês POS – Point of Service), onde múltiplas estações enviam dados para uma central onde todas as transações individuais podem ser consolidadas.

Multiple publishers, Also Subscriber

Em aplicações distribuídas, onde cada estação deve ter a liberdade de atualizar seus dados, porém tais atualizações são então publicadas aos demais pontos, desta forma, todos os nós tanto publicam quanto assinam as informações. Mas ainda temos apenas um único centro de distribuição das informações.

Replicação - Modelos de Replicação

Updating Subscriber

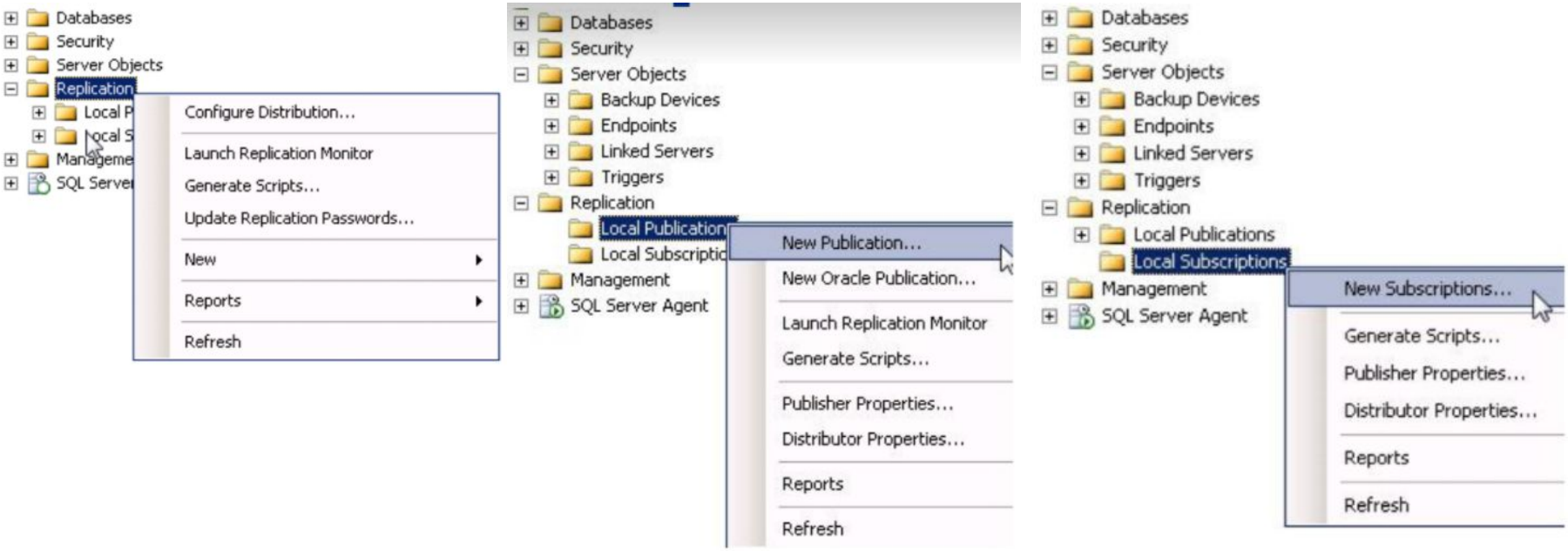
Uma alternativa à solução para aplicações distribuídas, onde cada estação publica e assina os dados apenas relacionando-se com uma central de dados. É a topologia mais utilizada quando se deseja uma matriz centralizadora das informações, mas se permite a liberdade das filiais realizarem alterações locais.

Peer-to-Peer

Outra alternativa à solução de transações distribuídas, podem agora cada estação age como um nós independente, responsável pela por publicar, distribuir e assinar os dados de outros servidores. Facilita a inclusão/remoção de nós e permite maior liberdade ao conjunto.

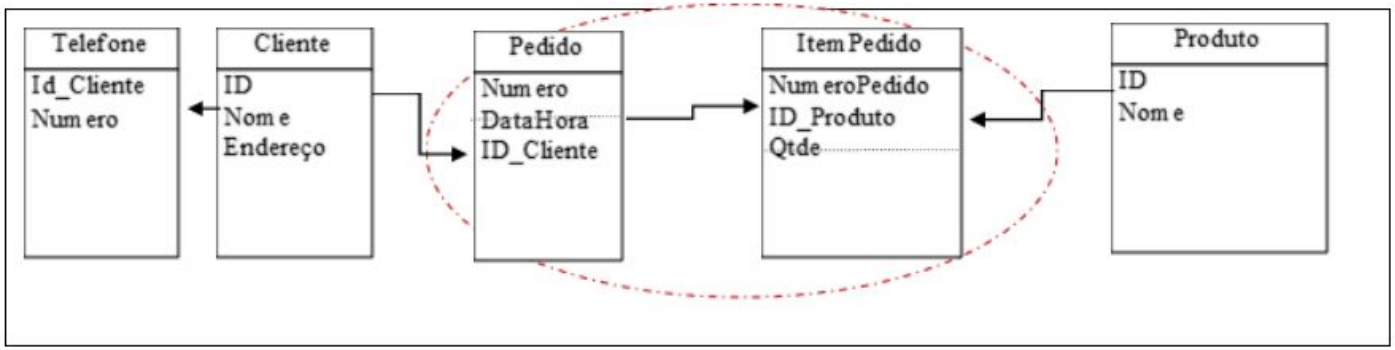
Replicação - Demonstração

Acompanhe as Demonstração do Professor



AC4

AC4 - Implementação de Replicação



Critérios de sucesso / Pontuação:

2 pts – Você configurou os dois ambientes corretamente (criação do banco e tabelas).
+2pts – Você populou a base principal com dados (carga inicial do servidor principal).
+4pts – Você implementou replicação (transactional) de todos os objetos (*Publisher e Subscriber*), crie 1 artigo com todas as tabelas.
+2pts – Você criou uma visão cujo resultado é similar ao seguinte resultset: Nome do cliente, Nome do produto comprado, qtde total comprada. Filtre apenas pedidos no mês atual (use a função getdate ()), este será seu *Article*. Utilizando Snapshot Replication, você finalmente replicou o conteúdo desta visão para uma tabela de destino (*Subscriber*).

Demonstre a replicação ao professor:

Apresente duas consultas, que façam selects na mesma* tabela em ambos os bancos / servidores (utilizar um linked server)

EX:
```
SELECT * FROM [Principal].impacta.dbo.cliente
SELECT * FROM [Secundario].impacta.dbo.cliente
```

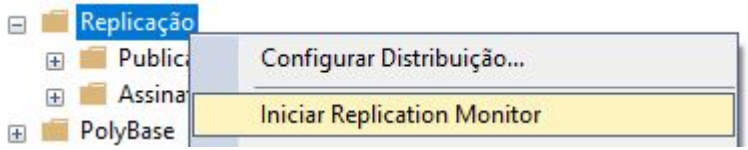
Insira uma linha nova no banco principal

```
INSERT INTO [Principal].impacta.dbo.cliente(...)VALUES(...)
```

Demonstre que os dois servidores ficaram atualizados e idênticos novamente

```
SELECT * FROM [Principal].impacta.dbo.cliente
SELECT * FROM [Secundario].impacta.dbo.cliente
```

Depois, abra o Replication Monitor
(botão direito sobre replicação → iniciar Replication Monitor)
E demonstre que sua aplicação está rodando sem erros.



Obrigado

Segurança de dados

Aula 11 – Alta Disponibilidade - parte 2

Gustavo Bianchi Maia
gustavo.maia@faculdadeimpacta.com.br