

Replicação em PostgreSQL

Contribuição de Alexandre Otelo
10 de dezembro de 2007
Última Atualização 12 de dezembro de 2007

Introdução

Muitas pessoas se perguntariam para quê migrar a sua aplicação de uma solução proprietária para uma solução livre, somente pelo custo da licença? E a formação de pessoal especializado e o suporte técnico, como fica, e as ferramentas de administração, replicação, backup, performance e outros.

É fato que as ferramentas proprietárias são melhores como um todo, mas cresce no mundo todo pessoas que desenvolvem soluções e as publicam, tornando cada vez mais robusta, esta por definição é SOFTWARE LIVRE. Você é livre para escolher, o suporte vem pela internet até mesmo em poucas horas, e não leva dias como em soluções proprietárias. Você é livre para desenvolver a sua própria solução para o seu negócio, você é livre para distribuí-la, mantendo os direitos que recebeu. Não podemos dizer que a ferramenta é instável, por que não existe ainda no mundo auditoria tão severa como a de estar com o código fonte aberto para o mundo.

A medida que se estuda mais este sistema, encontramos novos horizontes que existem na sua maioria das ferramentas proprietárias já são realidade. Neste documento procurei despejar a melhor didática para o estudo de caso com Slony.

1. PostgreSQL

PostgreSQL

é um Sistema Gerenciador de Banco de Dados, Objeto-Relacional, com licença BSD, ou seja, gratuito e aberto, multiplataforma, que vem crescendo muito o número de adeptos ao seu uso pela suas vantagens, principalmente a sua robustez.

2- Replicação

2.1 – Conceitos de replicação

Replicação

de banco de dados é a cópia dos dados de uma base de dados original para outra base. Isto permite que possa disponibilizar os dados em um ou mais sites. Isto melhora a disponibilidade dos dados. A cópia destes dados podem ser parcial ou total.

A replicação poderá ser usada para Sistemas distribuídos, Sistema de alta disponibilidade, ou talvez em processo de ETL em Data Warehouse.

Nos conceitos atuais de replicação a primeira classificação das ferramentas é relativo a sua arquitetura é síncrona ou assíncrona.

Replicação síncrona é uma cópias fiel em qualquer momento dos dados e transações. No mecanismo assíncrono, o servidor Master assume toda carga de escrita, atualização, deleção e leitura, com a diferença que o Slave apenas disponibiliza leitura da base conforme a figura 2.

Nos servidores Master podem ocorrer transações de leitura, escrita, atualização e deleção, já no servidor Slave, poderá ocorrer apenas operações de leitura, conforme figura 1.

Entre servidores Multimaster você poderá ter síncrono ou assíncrono, já em servidores Master e Slave, assíncrono. Isto é determinado pela ferramenta utilizada, e o que determina qual arquitetura será utilizada é o objetivo de implementação. Segue abaixo um comparativo de ferramentas para replicação no PostgreSQL:

Slony-I

Mamooth

Postgres-R

PG-Replicator

Tipo:

Assíncrono

Assíncrono

Síncrono

Assíncrono

Uso:

Master

- multislave

Master
- multislave

MultiMaster

Multimaster

Propaga
DDL

Não

Não

Não

?

Sist.
Operacional

Todos

Todos

Todos

Linux

BLOB

Não

?

?

Sim

Nestas
três arquiteturas propostas acima, cada uma tem uma
aplicabilidade específica preferencial.

- Síncrono
Multimaster: Balanceamento de carga ou alta disponibilidade;
- Assíncrono
Multimaster: Alta disponibilidade;
- Assíncrono
Master-Slave: Relatórios diversos níveis.

Fig1
– Servidores Master estão ambos sincronizados transmitindo
escrita, atualizações e deleções em tempo
real.

Fig2
- Servidores Master e Slave estão ambos sincronizados
transmitindo escrita, atualizações e deleções
em tempo real do Master para o Slave.

Para
quem ainda não está familiarizado com as
particularidades do mecanismo síncrono ou assíncrono
talvez se pergunte o que aconteceria se um “Servidor A”
replicasse assincronamente ao “Servidor B” e vice versa, isto não
seria parecido com o mecanismo síncrono? A resposta disso é
imediate, o servidor Slave é atualizado apenas e
exclusivamente pelo mecanismo de sincronização. Isto
provocaria um “lock”, travamento, de entrada de dados em ambos os
servidores, ficando ambos apenas disponível para leitura, o
que é logicamente um erro.

A
maior aplicabilidade dos mecanismos síncronos são para
proporcionar alta disponibilidade, enquanto o mecanismo assíncrono
pode ser usado para melhoria de performance e disponibilidade de
relatórios complexos, e por que não dizer parte

constituente de um Data Warehouse,
no processo de Extração, Transformação e
Carga(ETL).

A

alta disponibilidade é desejável em sistemas críticos
que não podem parar. Em caso de crash de um servidor, outro
servidor assume de forma transparente para o usuário final do
sistema.

Para

nosso primeiro exemplo de replicação imaginem uma
empresa matriz no Distrito Federal e com filiais nos estados do Rio
de Janeiro e São Paulo. A estrutura de rede necessariamente
uma máquina deve “enxergar” a outra, seja por IP fixo, VPN
ou outra. A matriz possui a gerência de material e recursos
humanos da empresa toda e as filiais estão on-line podendo
acompanhar atualizações na gestão de material e
dos recursos humanos. Se houver uma queda temporária da rede
a(s) filial(ais) estarão com o último estado
consistente da base de dados, podendo manter seus serviços com
menor risco, como por exemplo de vender algo que já foi
vendido pela matriz ou outra filial, ou de não vender por não
saber que o estoque já foi renovado e está disponível.

Fig3

– Distribuições entre filiais, em residências
ou em escritórios pelos gestores.

Dentre

os mecanismos de replicação iremos no capítulo
seguinte mostrar o Slony, atualmente é o mais conhecido pela
internet e aparentemente é o projeto que ganha mais adeptos.

2.2- Slony no Windows.

Algumas

características do Slony são:

- Replicação

Master Multislave;

- Assíncrono;

- Multiplataforma;

- A

definição da tabela a ser replicada na base “Master”
deve ser idêntica a “Slave”;

- Replicação

de tabelas e seqüências. Os demais objetos devem ser
importados na base Slave de acordo com a necessidade do usuário;

- Em caso de falha o servidor slave não para, e quando retomada a comunicação, o banco slave retorna automaticamente a forma do master em “pouco tempo”;
- Slony deve ser instalado em cada servidor que desejar ser master ou slave;
- Baixo tráfego pela rede;
- As tabelas e seqüências slave ficam permanentemente com lock para escrita, atualização ou deleção. Qualquer tratamento deve ser feito pelo banco através de functions ou triggers;
- Inadequado para clusters multimaster;

2.2.1- Conceitos no Slony

O

Slony é um replicador Master multi-Slave, assíncrono, e é multi-plataforma, ou seja, podemos instalá-lo no Windows, Linux, UNIX, Free BSD, e outros. Para nosso estudo de caso, adotamos a instalação no Windows pela familiaridade da maioria dos usuários, usando ferramenta de administração PgAdmin3 e neste caso o PostgreSQL 8.2, ou seja, apenas ferramentas livres em um sistema operacional proprietário. A seguir será mostrado screenshots de passo a passo da instalação e configuração do servidor de banco de dados e replicação.

2.2.2– Preparação do Sistema Operacional

Para aquisição do PostgreSQL na internet o usuário poderá buscar o site oficial: www.postgresql.org
Com

o download do instalador do PostgreSQL para Windows é relativamente pequeno, aproximadamente 25MB, você estará com todas as ferramentas necessárias para continuar nosso estudo. Lembro que no entanto que para instalar o PostgreSQL no Windows, é necessário usar Sistema de Arquivo NTFS. Para maiores detalhes a documentação está disponível para consulta no site.

A instalação do PostgreSQL é relativamente simples. Deve-se tomar o cuidado de instalar o módulo do Slony neste momento. Caso o usuário esteja aprendendo, recomento um nome fácil para senha do usuário postgres. No caso a senha que usarei no exemplo é 'postgres', mas isto é uma sugestão apenas didática, mas nada segura!

Fig4

– Seleção do Slony no momento de instalação do PostgreSQL.

Após

a instalação do PostgreSQL, falta a instalação do serviço Slony. É uma boa prática a utilização de ferramentas gráficas para administração, mas a instalação do serviço Slony é inevitavelmente na linha de comando. No entanto não se assuste, pois é bem fácil. O Slony é o serviço que rodará paralelo ao serviço de Servidor, que fará a ligação e replicação entre as bases de dados “Master” e “Slave”. Sem o serviço não há comunicação entre as bases de dados.

Uma

vez instalado e configurado o serviço, “nunca mais” será necessário ficar mexendo neste serviço, mesmo que a rede sofra interferência e volte a funcionar, mesmo que o computador servidor reinicie os serviços.

Próximo

passo é começar a configurar engines, do Slony. Para isto recomento a criação de uma pasta(folder) na raiz: C:\slony\

Os

engines precisam construídos e dessa forma fica melhor de administrar caso precise aumentar a criação de bases replicadas. Estes engines são arquivos de texto, que sugiro o nome: [base de dados].conf. Exemplo: Master.conf.

O conteúdo do engine é bem simples, pois é somente para o Slony saber qual base ele deverá gerenciar. O conteúdo do engine é simplesmente a indicação do: cluster_name, e do conn_info. Veja o exemplo abaixo:

Fig5

– Engine Master.conf dentro de C:\slony\

Agora

repita este procedimento para o Slave.conf.

Ao

final deste passo você deverá ter os dois arquivos engines do serviço Slony. Master.conf e Slave.conf.

Próximo passo é instalar o serviço Slony no Sistema Operacional Windows neste caso, mas que poderia ser outro sistema operacional. Para isto, vamos abrir o Prompt, e abrir o diretório: "C:\Program Files\Postgresql\8.2\bin"

Fig6
– Tela inicial para instalação do serviço Slony no Windows.

Esta é a única parte que é necessário mexer nas linhas de código, muitas pessoas não gostam, respeito, mas garanto que é bem fácil esta parte.

O comando SLON é chefe disso tudo e seus parâmetros são bem simples:

- regservice
[nome do serviço]: Registra serviço Slony;
- unregservice
[nome do serviço]: Destrói serviço Slony;
- addengine
[nome do serviço] [path para engine]; Acrescenta engine ao serviço Slony;
- delengine
[nome do serviço] [path para engine]; Exclui logicamente engine ao serviço;
- listengines
[nome do serviço]; Lista todos engines do serviço Slony.

Sabendo disso siga os passos para continuar a instalação do serviço:

- slon
- regservice Slony-l


```
- slon
-addengine Slony-I C:\slony\Master.conf

- slon
-addengine Slony-I C:\slony\Slave.conf

- slon
-listengines Slony-I (opcional)
```

Seu
serviço está instalado e disponível conferir em
Painel de Controle> Ferramentas Administrativas> Serviços.

Fig7
– Lista de serviços do Sistema Operacional no caso do
Windows.

É
uma boa prática trocar a inicialização de
Automático para Manual, em caso de desktop pessoal, mas em
servidor de teste ou produção sim, este deverá
ser automático. No momento o serviço não precisa
ser iniciado, mas quando precisar iniciar, parar ou reinicializar,
poderá ser feito nesta tela.

Este
passo terminou, como resultado final do sucesso é o serviço
Slony-I.

2.2.2- Implementando Slony

Chega
de brincar com linhas de comando, a partir de agora toda as ações
de administração do Slony será com PgAdmin3,
para alívio de alguns e desespero de outros mais
conservadores. Acontece que quando a quantidade de replicação
começam a crescer, começa ficar complicado a
administração por linha de código.

Para começar a interação entre PgAdmin3 e o Slony, é necessário a indicação de um caminho nas opções do PgAdmin3.

Fig8

– Configuração PgAdmin3 com Slony.

Escrevendo

o caminho: C:\Program Files\PostgreSQL\8.2\share de acordo com a figura 8, o PgAdmin3 está preparado para começar o cluster.

O

próximo passo, iremos criar uma base de dados Master e uma Slave, para fins didático.

Fig9

– Base de dados criadas.

Crie

um cluster “cl_teste” conforme os moldes passados na figura 10.

Fig10

– Cluster na base Master.

Fig11

– Cluster na base Slave.

Crie

uma tabela de teste em ambas as bases:

```
create
table t (name text not null primary key);
```

Próximo

passo é mostrar para cada um dos nós, Master node e Slave node no nosso exemplo, os caminhos entre si, estes caminhos são chamados de path. Abrindo a árvore das bases de dados, Master, replicação, cl_teste, encontraremos os nodes.

Na

base Master, o caminho do Master node deverá ser mostrado para

enxergar o Slave node com a string "host=localhost dbname=Slave user=postgres password=postgres". A demonstração de como se faz isto está na figura 12.

Fig12

– Digitação do path na base Master, no Master node

Uma base Master

somente precisa o path no Master node. O Slave node não precisa ser preenchido por que não trata-se de cascadeamento, ou seja, "Master" replica para "Slave1" que por sua vez replica para "Slave1a".

Este

processo deve ser repetido na base Slave no Slave node para enxergar o Master node. O node de administração serve como um concentrador de todos os paths daquele cluster.

Você após ter digitado o path no Master node na base Master e Slave node na base Slave esta etapa está completa e seguiremos para o próximo passo. O Listens aparecerão automaticamente, não há nada a ser mexido nele, ele simplesmente surgirá com a criação do path nos padrões corretos.

Neste próximo passo é a

hora de iniciar ou reiniciar o serviço Slony no sistema, então vamos lá. Vá ao Painél de controle do Windows, Ferramentas Administrativas, Serviços, procure o serviço Slony-I e inicie ou reinicie neste momento.

Fig13

– Momento de reiniciar o serviço Slony-I.

Para

fins de monitoria do que está acontecendo com o serviço Slony-I, é uma boa prática abrir o Visualizador de Eventos do Windows para ver alguma possível falha até o momento. Este visualizador fica em Painel de Controle, Ferramentas Administrativas e finalmente, Visualizador de Eventos. Vá em Aplicações e lá mostrará o serviço Slony-I e seus logs no sistema.

Fig14

– Visualizador de eventos para saber eventual problema no processo.

Próximo

passo é a criação do Set.

Nele define tabelas e seqüências que serão replicadas. Isto ocorrerá na base Master, não sendo necessário na base Slave, isto por que o Slony entende que seja na igualmente correspondente naquela base.

O

Slony replica apenas tabelas e seqüências, isto por que as functions, views

, triggers, e outros

fazem neste contexto parte da DDL do banco, mas que a partir da replicação das dependências de cada um destes objetos, seus resultados serão iguais às da base Master.

Fig15

– Criação de um SET.

Agora

falta pouco para vermos a nossa base Master replicada. Próximo passo é a inclusão das tabelas e seqüências no set.

A(s)

tabela(s) necessariamente deverão possuir um campo chave unique que será

“a chave” para replicação. Em outras palavras, se houver atualização, deleção ou inserção de registro, o Slony vai busca-lo por esta chave. Por este motivo o set pedirá estas informações no caso da tabela.

Nas seqüências, o mecanismo é parecido, mas sem precisar se preocupar com o campo chave. Você poderá fazer uso de nenhuma ou várias tabelas em um único set, e isto também vale para as seqüências.

Fig

16 – Demonstração de uma tabela inserida no SET.

Por

fim, a última etapa para vermos o nosso mecanismo funcionar, falta apenas a criação do Subscription. Isto ocorrerá na base Master, indicando origem, destino.

Clicando

com o botão direito em cima do Master set, procure a criação de um novo objeto. A sua opção desejada estará lá, New Subscription.

Fig17

– Criação do Subscription.

Nosso

produto final é esta fotografia:

Fig18

– Fotografia final do exercício.

Agora falta o teste. Vamos fazer dois testes para começarmos entender.

Primeiro

teste: Teste da carga, vá a base Master e de carga com vários nomes a sua escolha, pode escrever quantos quiser.

Ao

termino, o teste consiste em estes mesmos nomes estarem em seu correspondente na base Slave. Repita esta operação atualizando os campos e se quiser, apagando os campos também na base Master. Lembre-se que as tabelas envolvidas na replicação na base Slave agora possuem lock de qualquer tipo de operação de atualização, escrita ou deleção, serve apenas para leitura, mas as demais não envolvidas permanecem unlock.

Segundo

teste: Vá aos serviços do Windows e PARE o Slony-I. Com isto simulamos uma “queda de rede” entre base Master e Slave.

Atualize

os registros na Master e observe o que acontece no Slave. Claro que a Slave ficará com a última “fotografia” consistente, mas isto não impede que a base Master saia de produção ou dependa de qualquer favor da base Slave.

Por

fim, reinicie o serviço Slony-I e observe o que acontecerá com a base Slave. Se tudo estiver certo, ele tomará imediatamente o aspecto do Master.

Agora

que você teve um primeiro contato com Slony-I na prática, está livre para inventar desafios, dificuldades e transpor-las. Isto é um excelente exercício para conhecimento principalmente dos conceitos de síncrono e assíncrono, com o tempo o que pode parecer parecido, pode apresentar vantagens e desvantagens impactantes para o sua administração do banco.

Bibliografia

- <http://slony.info/> - Site oficial do projeto Slony; Acesso em out/2007.
- <http://www.postgresql.org/> - Site oficial do PostgreSQL; Acesso em out/2007.
- <http://www.pgadmin.org/> - Site oficial do PgAdmin3; Acesso em out/2007.
- <http://pgreplicator.sourceforge.net/> - Site oficial Pg Replicator; Acesso em out/2007.
- <http://www.commandprompt.com/products/postgresqlcore> – Site do Mamooth PostgreSQL; Acesso em out/2007.
- <http://pgpool.projects.postgresql.org/> - Site oficial PGPOOL; Acesso em out/2007;
- <http://www.postgresql.org.br/> - Site PostgreSQL no Brasil; Acesso em out/2007;
- Elmasri e Navathe “Sistemas de Banco de Dados – Fundamentos e Aplicações - 3a edição”