

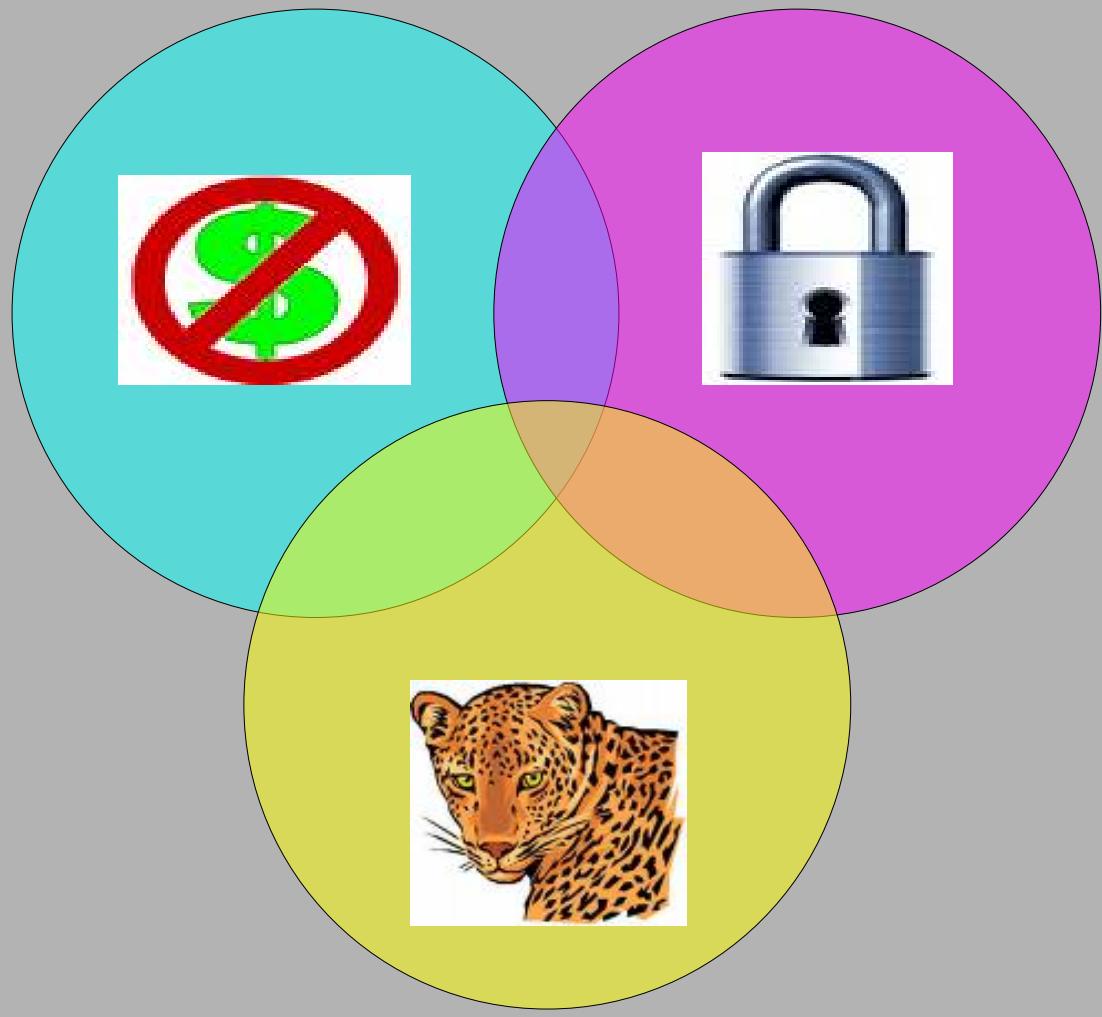


PostgreSQL Tuning: O elefante mais rápido que um leopardo

por Fernando Ike



Equilíbrio Difícil





O banco está lento!

Problemas comuns

- 60% dos problemas são relacionados ao mau uso da linguagem SQL;
- 20% dos problemas são relacionados a má modelagem do banco de dados;
- 10% dos problemas são relacionados a má configuração do SGDB;
- 10% dos problemas são relacionados a má configuração do SO.*



O banco está lento!

Escolhas Erradas

- Concentração de regras de negócio na aplicação para processos em lote;
- Integridade referencial na aplicação
- Mal dimensionamento de I/O (CPU, Plataforma, Disco)
- Ambientes virtualizados (Vmware, XEN, etc..) em AMD64/EMT64
- Uso de configurações padrões do SO e/ou do PostgreSQL



Melhor Hardware

- Servidores dedicados para o PostgreSQL
- Storage com Fiber Channel e iSCSI: Grupos de RAID dedicados
- RAID 5 ou 10: por Hardware
- Mais memória! (Até 4GB em 32 bits)
- Processadores de 64 bits: Performance até 3 vezes do que os 32 bits (AMD64 e EMT64 - Intel)



Melhor SO

- Sistemas Operacionais *nix: Linux (Debian, Gentoo), FreeBSD, Solaris, etc
- Em Linux: use Sistemas de arquivos XFS (noatime), Ext3 (writeback, noatime), Ext2
- Instale a última versão do PostgreSQL (atualmente 8.3) e à partir do código-fonte
- Não usar serviços concorrentes (Apache, MySQL, SAMBA...) em discos, semáforos e shared memory
- Usar, se possível, um kernel (linux) mais recente (e estável)



Parâmetros do SO

Modificando o *nix

- `echo "2" > /proc/sys/vm/overcommit_memory`
- `echo "25%" > /proc/sys/kernel/shmmax`
- `echo "25%/64" > /proc/sys/kernel/shmall`
- `echo "deadline" > /sys/block/sda/queue/scheduler`
- `echo "250 32000 100 128" > /proc/sys/kernel/sem`
- `echo "65536" > /proc/sys/fs/file-max`



Parâmetros do SO

Modificando o *nix

- `ethtool -s eth0 speed 1000 duplex full autoneg off`
- `echo "16777216" > /proc/sys/net/core/rmem_default`
- `echo "16777216" > /proc/sys/net/core/wmem_default`
- `echo "16777216" > /proc/sys/net/core/wmem_max`
- `echo "16777216" > /proc/sys/net/core/rmem_max`



Parâmetros do SO

Modificando o *nix

```
pmanson:~# su - postgres
```

```
postgres@pmanson:~$ ulimit 65535
```

/etc/security/limits.conf

postgres	soft	nofile	4096
postgres	soft	nproc	4096
postgres	hard	nofile	63536
postgres	hard	nproc	63536



Como organizar os Discos

O Melhor I/O

- Discos ou partições distintos para:
 - **Logs de transações (WAL)**
 - Índices: Ext2
 - Tabelas (particionar tabelas grandes)
 - Tablespace temporário (em ambiente BI)*
 - Archives
 - SO + PostgreSQL
 - Log de Sistema

* Novo no PostgreSQL 8.3



postgresql.conf

Memória

- **max_connections**: O menor número possível
- **shared buffers**: 33% do total -> Para operações em execução
- **temp_buffers**: Acesso às tabelas temporárias
- **work_mem**: Para agregação, ordenação, consultas complexas
- **maintenance_work_mem**: 75% da maior tabela ou índice
- **max_fsm_pages**: Máximo de páginas necessárias p/ mapear espaço livre. Importante para operações de UPDATE/DELETE.



postgresql.conf

Disco e Wal

- **wal_sync_method**: open_sync, fdatasync, open_datasync
- **wal_buffers**: tamanho do cache para gravação do WAL
- **commit_delay**: Permite efetivar várias transações na mesma chamada de fsync
- **checkpoint_segments**: tamanho do cache em disco para operações de escrita
- **checkpoint_timeout**: intervalo entre os checkpoints
- **wal_buffers**: 8192kB -> 16GB
- **bgwriter**: ??????
- **join_collapse_limit** = > 8



Tuning de SQL

- Analyze:

```
test_base=# EXPLAIN ANALYZE SELECT foo FROM bar;
```

- Ferramentas:

- **Pgfouine;**
- **Pgadmin3;**
- **PhpPgAdmin;**

Manutenção

- Autovacuum X Vacuum: Depende do uso (Aplicações Web, OLTI, BI)
 - Vacuum:
 - **vacuum_cost_delay**: tempo de atraso para vacuum executar automaticamente nas tabelas grandes
 - Autovacuum (ativado por padrão a partir da versão 8.3):
 - **autovacuum_naptime**: tempo de espera para execução do autovacuum.

Ferramentas de stress

- **Pgbench**: no diretório do contrib do PostgreSQL, padrão de transações do tipo TPC-B.
- **DBT-2**: Ferramenta da OSDL, padrão de transações do tipo TPC-C.
- **BenchmarkSQL**: Ferramenta Java para benchmark em SQL para vários banco de dados (JDBC), padrão de transações do tipo TPC-C.
- **Jmeter**: Ferramenta Java genérica para testes de stress, usado para aplicações (Web, ...) e também pode ser direto para um banco de dados.



Quando o tuning não resolve

- Escalabilidade vertical:
 - Mais e melhores discos;
 - Mais memória;
 - Melhor processador (quad core, 64bits)
- Escalabilidade horizontal:
 - **Pgpool I** (distribuição de carga de leitura e pool de conexões)
 - **PgPool II** (PgPool I + paralelização de grandes consultas)
 - **Slony I** (Replicação Multi-Master Assíncrona)
 - **Warm Stand By/Log Shipping**
 - **Pgbouncer + PL/Proxy + Slony**



Links

- **Documentação Oficial:**
<http://www.postgresql.org/docs/>
- **Power PostgreSQL:**
<http://www.powerpostgresql.com/>
- **PostgreSQL Brasil:**
<http://www.postgresql.org.br>
- **Teste de stress com soluções livres e proprietárias:**
<http://www.vivaolinux.com.br/artigos/verArtigo.php?codigo=7053&pagina=5>
- **Benchmark Brou-Ha-Ha**
<http://blogs.ittoolbox.com/database/soup/archives/benchmark-brouhaha-17939>
- **BenchmarkSQL, DBT-2 - SourceForge:**
<http://www.sf.net>
- **Jmeter**
<http://jakarta.apache.org/jmeter/>



Dúvidas

- Listas de discussão:
 - <http://archives.postgresql.org/pgsql-performance/>
 - <https://listas.postgresql.org.br/cgi-bin/mailman/listinfo/pgbr-geral>
- IRC irc.freenodes.net:
 - **#POSTGRESQL**
 - **#POSTGRESQL-BR**



Contato

- Fernando Ike
 - fernando.ike@b2br.com.br
 - fernando.ike@gmail.com
 - <http://www.midstorm.org/~fike/weblog>