PostgreSQL Troubleshooting e Monitoramento

Eduardo Ferreira dos Santos

Dataprev
Empresa de Tecnologia e Informações da Previdência Social
eduardosantos@previdencia.gov.br
eduardosan.wordpress.com

11 de Novembro de 2010

Para começar

- Banco de dados não é para amadores.
- O Sistema Operacional pode ser o melhor amigo (ou inimigo) do DBA.
- Especificar corretamente o hardware DEVE ser trabalho do DBA, pois é extremamente importante.
- Os riscos dos erros do DBA são sempre maiores. Aprenda a conviver com o conservadorismo.

Sumário

1. Monitoramento
2. Troubleshooting
3. Administração assistida(?)
4. Referências

1 Monitoramento

2 Troubleshooting

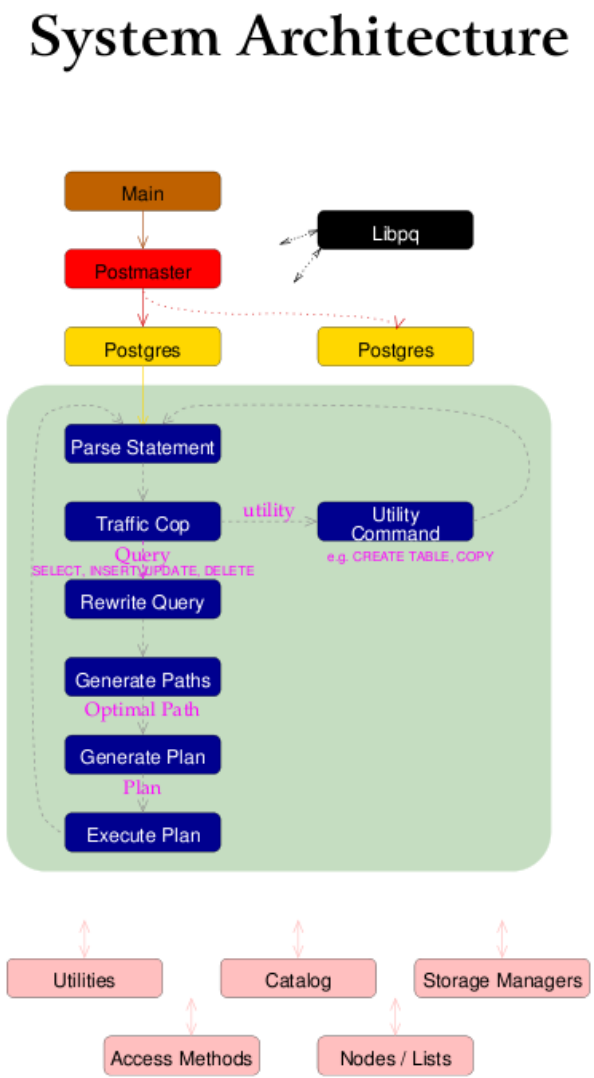
3 Administração assistida(?)

4 Referências

Entendendo o SGBD

- Como o PostgreSQL utiliza o processador?
- Como é a utilização da memória?
- Como é o consumo de disco?
- Como o SO pode ser otimizado?
- Como identificar os componentes externos?

Entendendo o SGBD ([Momjian, 2010])



Entendendo o SGBD

- Hardware:
  - Processador
  - Memória principal
  - Memória secundária
- Sistema operacional e seus subsistemas (principalmente o kernel);
- Sistema de execução de consultas;
- Processamento de transações;
- Armazenamento.

Processador

- O caminho de uma consulta [PostgreSQL, 2010]:
  1. A conexão de uma aplicação ao servidor PostgreSQL deve ser estabelecida. O programa transmite a consulta ao servidor e espera pelos resultados;
  2. O estágio de *parser* verifica se a consulta possui a sintaxe correta e cria uma árvore de consulta;
  3. O sistema de reescrita recebe a àrvore criada criada pelo *parser* e busca quaisquer regras (armazenadas no catálogo do sistema) que possam ser aplicadas à árvore. São então realizadas as transformações fornecidas pelas regras.
  4. Uma das funções do sistema de reescrita é na realização de visões (*views*). Todas as vezes em que uma consulta em uma visão (ou uma tabela virtual) é realizada, o sistema de reescrita altera a consulta para outra que acessa as tabelas base fornecidas em sua definição.

Processador

- O caminho de uma consulta [PostgreSQL, 2010]:
  1. O otimizador recebe a árvore de consulta (possivelmente reescrita) e cria um plano de execução que será a entrada do executor.
  2. O plano é criado através da criação de todos os possíveis caminhos que levam ao resultado. (...) O caminho mais barato (mais rápido) é expendido em um plano completo que o executor pode utilizar.
  3. O Executor caminha recursivamente através da árvore e busca as linhas no formato representado pelo plano de execução. O executor utiliza então o sistema de armazenamento enquanto está verificando relações, fazendo ordenações (*sorts*) e junções (*joins*), avalia as qualificações e finalmente envia as linhas encontradas.

Ferramentas de monitoramento

```
  1  [|||                                                    2.6%]     Tasks: 69 total, 1 running
  2  [|||                                                    3.1%]     Load average: 0.15 0.22 0.24
  3  [||                                                     2.0%]     Uptime: 15 days, 06:38:46
  4  [|||                                                    3.2%]
  Mem[||||||||||||||||||||||||||||||||||||||||||||||||||1283/8010MB]
  Swp[|                                                     0/956MB]

    PID USER      PRI  NI  VIRT   RES   SHR S CPU% MEM%   TIME+  Command
  15739 postgres  20   0 2140M 1964M 1928M S  0.0 24.5 13:55.64 postgres: www-data ct-gcie 192.168.8.35(49381) idle
  15725 postgres  20   0 2163M 1887M 1827M S  0.0 23.6 11:53.99 postgres: www-data ct-gcie 192.168.8.35(49379) idle
  15897 postgres  20   0 2140M 1879M 1842M S  0.0 23.5 14:24.81 postgres: www-data ct-gcie 192.168.8.35(37865) idle
   4968 postgres  20   0 2100M 1866M 1864M S  0.0 23.3 20:36.17 postgres: writer process
  15719 postgres  20   0 2139M 1775M 1739M S  0.0 22.2 13:23.12 postgres: www-data ct-gcie 192.168.8.35(49378) idle
  16976 postgres  20   0 2153M 1769M 1723M S  0.0 22.1 11:05.41 postgres: www-data ct-gcie 192.168.8.35(38966) idle
  15726 postgres  20   0 2134M 1695M 1666M S  5.7 21.2  7:56.78 postgres: www-data ct-gcie 192.168.8.35(49380) idle
  15812 postgres  20   0 2139M 1490M 1455M S  0.0 18.6 10:53.10 postgres: www-data ct-gcie 192.168.8.35(57934) idle
  15718 postgres  20   0 2132M 1469M 1443M S  0.0 18.3  4:39.01 postgres: www-data ct-gcie 192.168.8.35(49377) idle
  20590 postgres  20   0 2128M 1214M 1190M S  3.8 15.2  4:56.05 postgres: www-data ct-gcie 192.168.8.35(33252) idle
  15872 postgres  20   0 2152M 1133M 1083M S  0.0 14.1  6:46.43 postgres: www-data ct-gcie 192.168.8.35(37854) idle
   1136 postgres  20   0 2140M 1034M  996M S  0.0 12.9  2:26.59 postgres: service0 spb 192.168.9.39(58531) idle
```

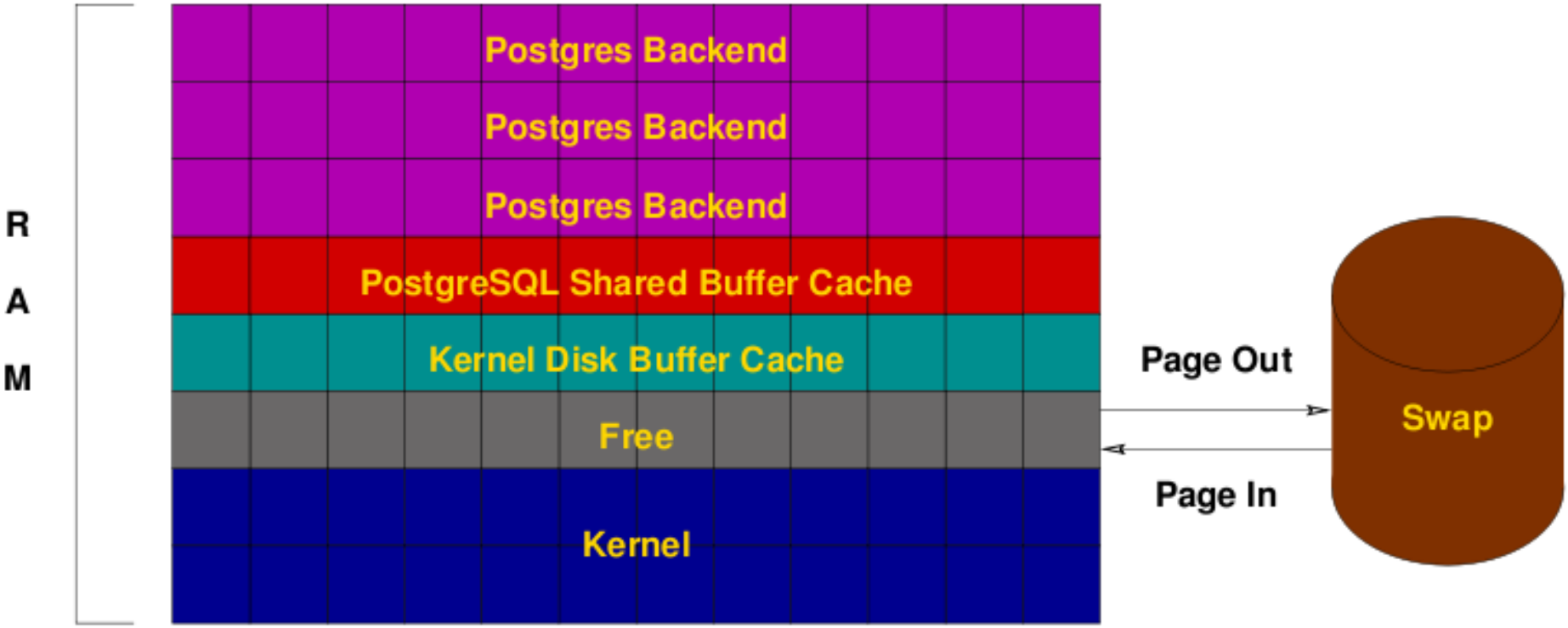
```
top - 16:28:19 up 15 days,  6:39,  3 users,  load average: 0.38, 0.25, 0.24
Tasks: 121 total,   2 running, 119 sleeping,   0 stopped,   0 zombie
Cpu(s): 26.6%us,  0.2%sy,  0.0%ni, 73.1%id,  0.0%wa,  0.0%hi,  0.1%si,  0.0%
Mem:   8202644k total,  8058224k used,   144420k free,  1903964k buffers
Swap:   979924k total,      232k used,   979692k free,  4817504k cached

  PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM    TIME+  COMMAND
  967 postgres  20   0 2123m 910m 891m R  100 11.4   2:32.61 postgres
  968 postgres  20   0 2123m 896m 876m S    6 11.2   2:31.60 postgres
15897 postgres  20   0 2140m 1.8g 1.8g S    1 23.5  14:25.52 postgres
 4969 postgres  20   0 54852 2916  548 S    1  0.0 313:49.34 postgres
```

```
node406:~# vmstat 2
procs -----------memory---------- ---swap-- -----io---- -system-- ----cpu----
 r  b   swpd   free   buff  cache   si   so    bi    bo   in   cs us sy id wa
 0  0    232 147692 1904220 4818048    0    0    72    47    5    1  3  0 96  1
 1  0    232 147404 1904228 4818084    0    0     0    48   87  231  5  0 94  0
 0  0    232 147252 1904228 4818296    0    0    24     8  135  377  4  0 95  0
 1  0    232 147348 1904232 4818160    0    0     0    68   10   40  1  0 99  0
 1  0    232 147704 1904240 4817976    0    0     0    48  162  359 26  0 74  0
 0  0    232 147952 1904240 4818084    0    0     0     0   56  219  5  0 95  0
 0  0    232 148048 1904240 4818052    0    0     4    76   94  315  2  0 97  0
 0  0    232 147672 1904240 4818276    0    0     0    36   89  202  1  1 98  0
 0  0    232 147604 1904240 4818192    0    0    12   124  103  374  7  0 93  0
 0  0    232 147148 1904240 4817936    0    0     0    36  173  427  1  0 99  0
 0  0    232 147620 1904248 4818192    0    0     0    12   67  200  0  0 100  0
 0  0    232 146828 1904248 4818016    0    0     4    64  112  285  5  0 95  0
 0  0    232 146644 1904248 4818200    0    0     0     0   48  145 10  0 90  0
 0  0    232 146644 1904248 4818200    0    0     0     0   14   45  0  0 100  0
 0  0    232 147016 1904248 4818200    0    0     0    84   33  120  0  0 99  0
```

Memória principal

- O que são os danados dos shared_buffers?
- max_connections
- Maldito swap do inferno!!!
- Como monitorar a memória?

Memória Principal

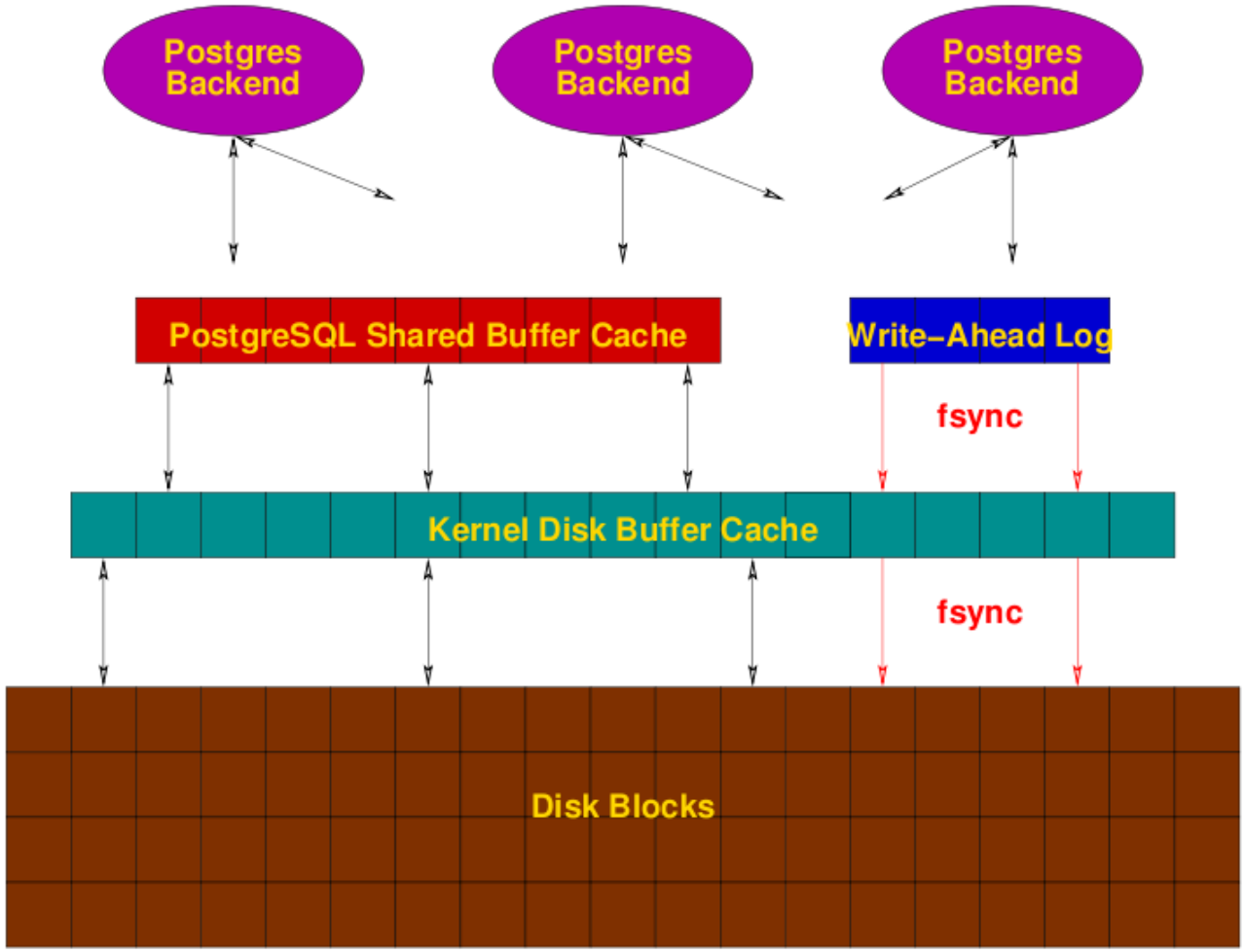


[Momjian, 2010]

Memória secundária

- Utilização dos sistemas de arquivos para realização das consultas (Veja 6)
- WAL
- Divisão em pequenos arquivos

Memória secundária



[Momjian, 2010]

Memória secundária

Para os casos de:

- Melhores tipos de disco
- Opções de storage

Consulte:
http://www.midstorm.org/ telles/2008/07/25/postgresql-discos-cia/

Monitorando o disco

- Monitorar o disco pode ser um grande desafio para o DBA.
- A maior parte das ferramentas não dá um número de carga do disco.
- Métrica mais eficiente: load average e blocks in - blocks out.

```
Linux 2.6.26-1-amd64 (nodo406.labcluster)       10-11-2010      _x86_64_

avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           3,11    0,01    0,25    0,51    0,00   96,11

Device:            tps   Blk_read/s   Blk_wrtn/s   Blk_read   Blk_wrtn
sda              24,66       579,87       376,06  767062533  497455345
sda1              0,00         0,01         0,01      14243      11816
sda2              0,60         1,50         6,36    1983346    8407064
sda3             24,06       578,36       369,69  765064592  489036465
```

Monitorando o disco

- Load Average acima de 3 e baixo processamento indica sinais de problema com o disco.
- Relação bi - bo acima de 100: o banco vai travar!!!
- O comando *iostat* mostra o TPS (transações por segundo no disco). Se o número de transações estiver alto e bi - bo estiver crescendo, há problemas no banco.

Ferramentas de monitoramento

- O log é seu melhor amigo
- O SO precisa mostrar informações relevantes
- É possível monitorar a execução diretamente no banco

A bíblia do monitoramento direto no banco

pg_stat_all_indexes	view	postgres
pg_stat_all_tables	view	postgres
pg_stat_database	view	postgres
pg_stat_sys_indexes	view	postgres
pg_stat_sys_tables	view	postgres
pg_stat_user_indexes	view	postgres
pg_stat_user_tables	view	postgres
pg_statio_all_indexes	view	postgres
pg_statio_all_sequences	view	postgres
pg_statio_all_tables	view	postgres
pg_statio_sys_indexes	view	postgres
pg_statio_sys_sequences	view	postgres
pg_statio_sys_tables	view	postgres
pg_statio_user_indexes	view	postgres
pg_statio_user_sequences	view	postgres
pg_statio_user_tables	view	postgres

[Momjian, 2010]

1 Monitoramento

2 Troubleshooting

3 Administração assistida(?)

4 Referências

O que é um problema?

- Lentidão?
- Indisponibilidade momentânea?
- Indisponibilidade prolongada?
- Perda de dados? (PERIGO!!!)

Lentidão

- O que está lento?
  - Consulta demorando muito?
  - Demorando para conseguir uma nova conexão?
  - Conheça o banco e saiba identificar pontos de lentidão.

```
spb=# \d pg\_stat\_activity
          Visao "pg\_catalog.pg\_stat\_activity"
      Coluna      |           Tipo           | Modificadores
------------------+--------------------------+---------------
 datid            | oid                      |
 datname          | name                     |
 procpid          | integer                  |
 usesysid         | oid                      |
 usename          | name                     |
 current\_query   | text                     |
 waiting          | boolean                  |
 query\_start     | timestamp with time zone |
 backend\_start   | timestamp with time zone |
 client\_addr     | inet                     |
 client\_port     | integer                  |
Definicao da visao:
 SELECT d.oid AS datid, d.datname, pg\_stat\_get\_backend\_pid(s.backendid) AS procp
   FROM pg\_database d, ( SELECT pg\_stat\_get\_backend\_idset() AS backendid) s, pg
  WHERE pg\_stat\_get\_backend\_dbid(s.backendid) = d.oid AND pg\_stat\_get\_backend
```

Indisponibilidade momentânea

- Houve algum erro quando o sistema foi reiniciado?
- Alguma das partições não subiu?
- Problema de fencing em Cluster?
- O log é seu amigo!!!

Indisponibilidade prolongada

- Definição da variável TEMPO!
- Validação de consistência.
- Levantando a cópia de segurança ou o backup. De acordo com Telles [Telles, 2010], pg_dump não é backup!!!

Perda de dados

- Como estar seguro sobre a perda dos dados?
- A importância do WAL
- Como tornar o SGBD menos suscetível a problemas assim?
- Uma vez perdido, só um restore salva.

Tipos de falha segundo [Momjian, 2010]

- Falha na aplicação do Cliente;
- Falha "elegante"no servidor (manda desligar);
- Falha abrupta no servidor;
- Falha no sistema operacional;
- Falha no disco;
- Remoção acidental de dados (DELETE);
- WAL corrompido;

Tipos de falha segundo [Momjian, 2010] (continuação)

- Arquivos removidos;
- DROP TABLE acidental;
- DROP INDEX acidental;
- DROP DATABASE acidental;
- Instalação não inicia;
- Índices corrompidos;
- Tabelas corrompidas.

Ações sugeridas por [Momjian, 2010]

- Falha na aplicação do Cliente: Nenhuma ação necessária. Transações sofrem ROLLBACK.
- Falha "elegante"no servidor (manda desligar): Nenhuma ação necessária. Transações sofrem ROLLBACK.
- Falha abrupta no servidor: Nenhuma ação necessária. Transações sofrem ROLLBACK.
- Falha no sistema operacional: Nenhuma ação necessária. Transações sofrem ROLLBACK. Páginas escritas parcialmente são reparadas
- Falha no disco: Restaure o backup ou use PITR
- Remoção acidental de dados (DELETE): Restaure a tabela do último backup. É possível configurar o banco para visualizar tuplas excluídas.
- WAL corrompido: Veja pg_resetxlog. Reveja as transações e identifique os danos, incluindo transações parcialmente gravadas.

Ações sugeridas por [Momjian, 2010] (continuação)

- Arquivos removidos: pode ser necessário criar um arquivo vazio de mesmo nome do excluído para que o objeto possa ser excluído e restaurado do último backup
- DROP TABLE acidental: Recupere do último backup
- DROP INDEX acidental: Crei o índice novamente
- DROP DATABASE acidental: Recupere do último backup
- Instalação não inicia: Normalmente problema no WAL. Veja recuperação do WAL
- Índices corrompidos; Use REINDEX
- Tabelas corrompidas. Tente reindexar a tabela. Tente identificar o OID da linha corrompida e copie os dados válidos para uma tabela temporária

1 Monitoramento

2 Troubleshooting

3 Administração assistida(?)

4 Referências

Seja corajoso!!!

- Se você espera algo no estilo Oracle, esqueça.
- Mesmo que existam ferramentas, o mais importante é conhecer o SGBD e seus componentes.
- DBA não é DBV!!!

Ainda assim quero telinhas...

http://www.pgfoundry.org

Group Name	Description
Wildlife Monitoring Database	The Wildlife Monitoring Database is the code for the African & Asian Elephant Database (AAED) developed by IUCN, Solertium and a community of volunteers. It is intended to be a PostGIS application for monitoring the status of elephants and more.
pgTop - a top clone for PostgreSQL	pgTop is a console-based (non-gui) tool for monitoring the processes and overall performance of a PostgreSQL server. pgTop is written in python.
PostgreSQL Database Administration Tools	pgtools is a package with usefull utillities for manteining, monitoring, and administrate PostgreSQL database clusters.
pgstat	pgstat is a command line utility to display PostgreSQL information on the command line similar to iostat or vmstat. This data can be used for monitoring or performance tuning.
Nagios monitoring plugins for PostgreSQL	Nagios plugin scripts to monitor: transaction id status, blocked queries, long running queries, connection status and more. Please try this project 1st: http://bucardo.org/wiki/Check_postgres

1 Monitoramento

2 Troubleshooting

3 Administração assistida(?)

4 Referências

Momjian, B. (2010).
Mastering postgresql administration.
http://momjian.us/main/writings/pgsql/administration.pdf Acessado em 10/11/2009.

PostgreSQL, C. (2010).
The path of a query.
http://www.postgresql.org/docs/8.4/interactive/query-path.html Acessado em 10/11/2010.

Telles, F. (2010).
Dump não é backup.
http://www.midstorm.org/ telles/2010/05/06/dump-nao-e-backup/ Acessado em 10/11/2010.

Contato

Eduardo Ferreira dos Santos
Dataprev - D2OP/CPDF/DIT

eduardosantos@previdencia.gov.br
eduardo.edusantos@gmail.com

www.postgresql.org.br
eduardosan.wordpress.com

+55 61 3262-7481