

# Treinamento PostgreSQL - Aula 07

Eduardo Ferreira dos Santos

SparkGroup  
Treinamento e Capacitação em Tecnologia  
eduardo.edusantos@gmail.com  
eduardosan.com

06 de Junho de 2013

# Cronograma

Semana 1: 27 de Maio a 3 de Junho Administração de Dados

Semana 2: 4-11 de Junho Administração de Banco de Dados

Semana 3: 13-18 de Junho Alta disponibilidade

Semana 4: 19-24 de Junho Performance Tuning

## 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**.
- Estudar é importante, mas para um DBA **é muito mais importante!**

## Sumário

- 1 Monitoramento
  - Introdução
  - Processador
  - Memória Principal
  - Memória secundária
  - Ferramentas de monitoramento
- 2 Troubleshooting
  - Introdução
  - Tipos de problema
- 3 Administração assistida(?)
- 4 Referências

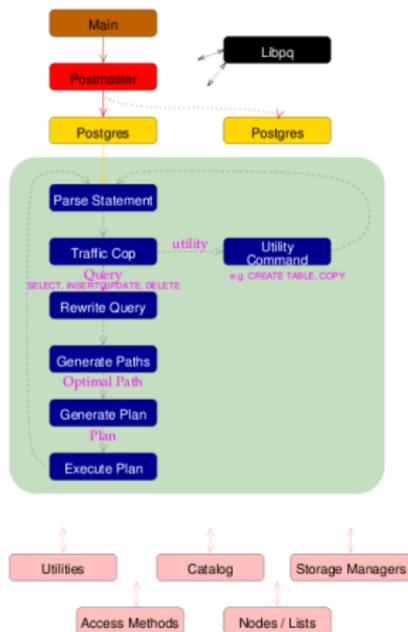
- 1 Monitoramento
  - Introdução
  - Processador
  - Memória Principal
  - Memória secundária
  - Ferramentas de monitoramento
- 2 Troubleshooting
  - Introdução
  - Tipos de problema
- 3 Administração assistida(?)
- 4 Referências

# 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])

## System Architecture



# 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.

## 1 Monitoramento

- Introdução
- **Processador**
- Memória Principal
- Memória secundária
- Ferramentas de monitoramento

## 2 Troubleshooting

- Introdução
- Tipos de problema

## 3 Administração assistida(?)

## 4 Referências

# 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 árvore 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.

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]:
  - ④ O otimizador recebe a árvore de consulta (possivelmente reescrita) e cria um plano de execução que será a entrada do executor.
  - ⑤ O plano é criado através da criação de todos os possíveis caminhos que levam ao resultado. (...) O caminho mais barato (mais rápido) é expandido em um plano completo que o executor pode utilizar.
  - ⑥ 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.

# Processador

- O bom DBA **entende** suas consultas e conhece a melhor forma de otimizá-la.
- EXPLAIN ANALYZE é seu amigo!

```
SELECT distinct lm.landmark_id ,
               lm.name ,
               ts_rank_cd(to_tsvector(aslm.name || ' ' || arlm.name || ' ' || asrlm.name), query) as rank
FROM lm_landmarks lm ,
     ac_state_language_map aslm ,
     ac_regions ar ,
     ac_region_language_map arlm ,
     ac_sub_regions asr ,
     ac_sub_region_language_map asrlm ,
     plainto_tsquery('Barcelona') query
WHERE aslm.state_id = lm.state_id
AND ar.state_id = lm.state_id
AND arlm.region_id = ar.region_id
AND asr.region_id = ar.region_id
AND asr.sub_region_id = asrlm.sub_region_id
AND query @@ to_tsvector(aslm.name || ' ' || arlm.name || ' ' || asrlm.name)
ORDER BY rank DESC;
```

# Processador

- A consulta é lentíssima!
- Como sabemos o que consome de processador?

```
EXPLAIN ANALYZE SELECT distinct lm.landmark_id ,
                        lm.name ,
                        ts_rank_cd(to_tsvector(aslm.name || ' ' || arlm.name || ' ' || asrlm.name), query) as rank
FROM lm_landmarks lm,
     ac_state_language_map aslm ,
     ac_regions ar ,
     ac_region_language_map arlm ,
     ac_sub_regions asr ,
     ac_sub_region_language_map asrlm ,
     plainto_tsquery('Barcelona') query
WHERE aslm.state_id = lm.state_id
AND ar.state_id = lm.state_id
AND arlm.region_id = ar.region_id
AND asr.region_id = ar.region_id
AND asr.sub_region_id = asrlm.sub_region_id
AND query @@ to_tsvector(aslm.name || ' ' || arlm.name || ' ' || asrlm.name)
ORDER BY rank DESC;
```

# Resultados

```

Unique (cost=72049.69..72065.27 rows=1558 width=86) (actual time
=61248.918..62537.913 rows=12 loops=1)
-> Sort (cost=72049.69..72053.58 rows=1558 width=86) (actual time
=61248.913..61905.781 rows=388496 loops=1)
    Sort Key: (ts_rank_cd(to_tsvector((((aslm.name)::text || ' '::text) ||
(arlm.name)::text) || ' '::text) || (asrlm.name)::text)),
query.query), lm.landmark_id, lm.name
    Sort Method: external merge  Disk: 15544kB
-> Nested Loop (cost=6019.09..71967.07 rows=1558 width=86) (actual
time=443.145..55025.760 rows=388496 loops=1)
    Join Filter: (query.query @@ to_tsvector((((aslm.name)::text ||
' '::text) || (arlm.name)::text) || ' '::text) || (asrlm.na
me)::text)))
-> Function Scan on plainto_tsquery query (cost=0.00..0.01 rows
=1 width=32) (actual time=0.065..0.067 rows=1 loops=1)
-> Merge Join (cost=6019.09..32989.44 rows=1558170 width=54) (
actual time=417.412..30064.655 rows=6258960 loops=1)
    (...)
Total runtime: 62546.280 ms
(35 rows)

```

# Monitoramento

- Pergunta de 1 milhão de dólares:
- Como saber se o meu banco de dados precisa de otimização?

# Monitoramento

- Pergunta de 1 milhão de dólares:
- Como saber se o meu banco de dados precisa de otimização?
- Resposta:

# Monitoramento

- Pergunta de 1 milhão de dólares:
- Como saber se o meu banco de dados precisa de otimização?
- Resposta:
- Depende!

# Monitoramento

- Pergunta de 1 milhão de dólares:
- Como saber se o meu banco de dados precisa de otimização?
- Resposta:
- **Depende!**
- O que é lento pra você?
- Lentidão x Consumo de recursos de máquina
- Lembrete: usuários de Windows comumente culpam seus computadores quando há alguma lentidão em seu SO. E em bancos de dados? E no PostgreSQL?

# 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

```

top - 16:28:19 up 15 days, 6:39, 3 users, load average: 0.38, 0.25, 0.24
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 72 48 8 231 5 0 94 0
0 0 232 147252 1904228 4818296 0 0 24 8 135 377 4 95 0
1 0 232 147348 1904232 4818160 0 0 0 68 10 40 1 99 0
1 0 232 147704 1904240 4817976 0 0 0 48 162 359 26 0 74 0
0 0 232 147952 1904240 4818004 0 0 0 56 219 5 95 0
0 0 232 148048 1904240 4818052 0 0 4 76 94 315 2 97 0
0 0 232 147672 1904240 4818276 0 0 0 36 89 202 1 98 0
0 0 232 147604 1904240 4818192 0 0 12 124 103 374 7 93 0
0 0 232 147148 1904240 4817936 0 0 0 36 173 427 1 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 95 0
0 0 232 146644 1904248 4818200 0 0 0 48 145 10 90 0
0 0 232 146644 1904248 4818200 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

```

- 1 Monitoramento
  - Introdução
  - Processador
  - **Memória Principal**
  - Memória secundária
  - Ferramentas de monitoramento
- 2 Troubleshooting
  - Introdução
  - Tipos de problema
- 3 Administração assistida(?)
- 4 Referências

# Memória principal

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

# Memória Principal

- Por que estou preocupado com a memória?

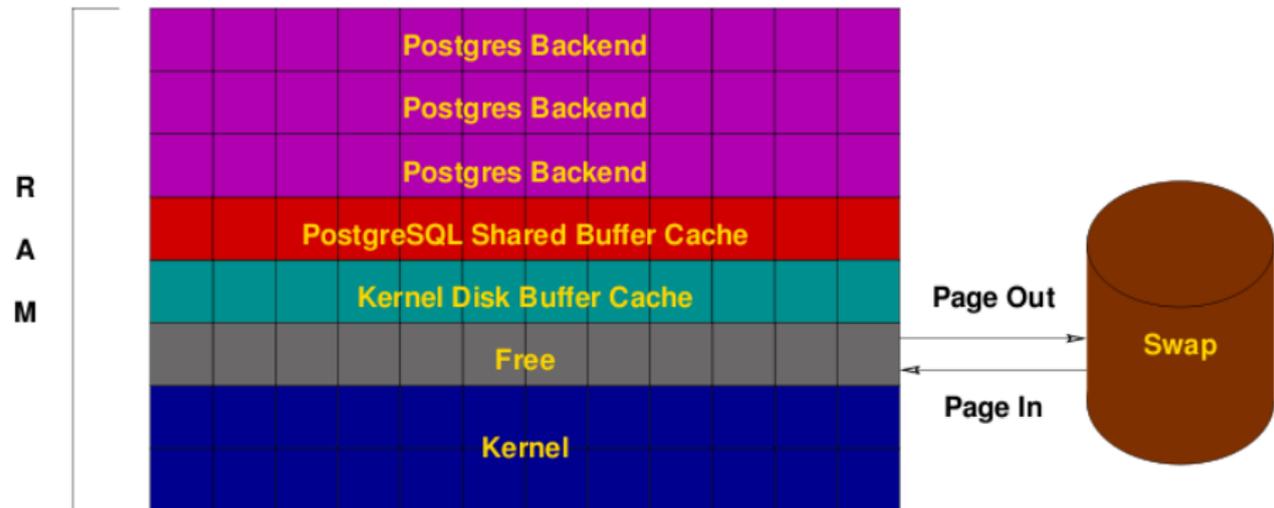
# Memória Principal

- Por que estou preocupado com a memória?
- Resposta: **porque o banco de dados está lento**

# Memória Principal

- Por que estou preocupado com a memória?
- Resposta: **porque o banco de dados está lento**
- No caso de lentidão do banco de dados, quase sempre culpamos a memória.
- Agora que conhecemos o PostgreSQL, somos capazes de indentificar se o problema está na memória?

# Memória Principal



[Momjian, 2010]

# Memória Principal

- É muito difícil descobrir quanto de memória cada consulta utiliza;
- O primeiro passo para otimizar a performance é alterar os `shared_buffers`;
- Uma configuração correta de `shared_buffers` é o que mais afeta o tempo de resposta do banco.
- Qual valor colocar?

# Memória Principal

- É muito difícil descobrir quanto de memória cada consulta utiliza;
- O primeiro passo para otimizar a performance é alterar os `shared_buffers`;
- Uma configuração correta de `shared_buffers` é o que mais afeta o tempo de resposta do banco.
- Qual valor colocar?
- Não existe receita mágica, mas normalmente deve ser algo entre **25%** e **40%** da memória disponível.

# Memória Principal

- É muito difícil descobrir quanto de memória cada consulta utiliza;
- O primeiro passo para otimizar a performance é alterar os `shared_buffers`;
- Uma configuração correta de `shared_buffers` é o que mais afeta o tempo de resposta do banco.
- Qual valor colocar?
- Não existe receita mágica, mas normalmente deve ser algo entre **25%** e **40%** da memória disponível.
- Como monitorar?

# Memória Principal

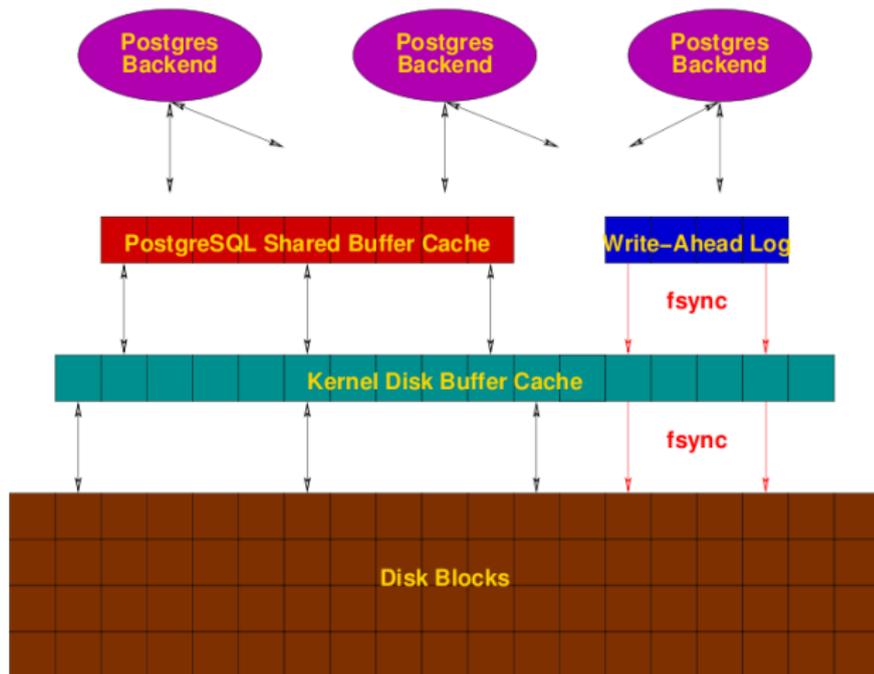
- É muito difícil descobrir quanto de memória cada consulta utiliza;
- O primeiro passo para otimizar a performance é alterar os `shared_buffers`;
- Uma configuração correta de `shared_buffers` é o que mais afeta o tempo de resposta do banco.
- Qual valor colocar?
- Não existe receita mágica, mas normalmente deve ser algo entre **25%** e **40%** da memória disponível.
- Como monitorar?
- Observe se ficou mais rápido;
- Utilize o `pgbench`, que falaremos mais à frente.

- 1 Monitoramento
  - Introdução
  - Processador
  - Memória Principal
  - Memória secundária
  - Ferramentas de monitoramento
- 2 Troubleshooting
  - Introdução
  - Tipos de problema
- 3 Administração assistida(?)
- 4 Referências

# Memória secundária

- Utilização dos sistemas de arquivos para realização das consultas (Veja 3)
- WAL
- Divisão em pequenos arquivos - **Método TOAST**

# Memória secundária



[Momjian, 2010]

# O método TOAST

- Por conceito, uma tupla não pode ficar em mais de uma página no sistema de arquivos
- Cada página tem no máximo 8k<sup>1</sup>
- Para registros muito grandes podem acontecer **problemas de alocação**
- Método TOAST<sup>2</sup>: os registros são então **comprimidos** e **divididos** em várias linhas

---

<sup>1</sup>Pode ser alterado na compilação do banco

<sup>2</sup>*The best thing since slice bread*

# Implementação do método TOAST

- Registros de **tamanho variável**
- Os primeiros 32 bits contêm o tamanho do registro em bytes
- Os dois primeiros bits são reservados e tratados da seguinte forma
  - 00 Valor armazenado sem a aplicação de TOAST
  - 10 O valor armazenado possui cabeçalho (*header*) de um único byte; o restante representa o total do valor armazenado.<sup>3</sup>
  - 01 Dados comprimidos e precisam ser descomprimidos antes de serem utilizados
  - 11 Os valores não possuem nenhum limite particular

---

<sup>3</sup>Se todos os registros restantes forem 0 o valor é um ponteiro para os dados armazenados em outra tabela TOAST

# Memória secundária

Para os casos de:

- Melhores tipos de disco
- Opções de storage

Consulte: <http://savepoint.blog.br/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.
- **MUITO IMPORTANTE:** Consultas também utilizam o disco, e não somente operações de escrita. Leiam novamente o Método TOAST.

- 1 Monitoramento
  - Introdução
  - Processador
  - Memória Principal
  - Memória secundária
  - Ferramentas de monitoramento
- 2 Troubleshooting
  - Introdução
  - Tipos de problema
- 3 Administração assistida(?)
- 4 Referências

# 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]

# pgbench

- Muito similar ao método de tentativa e erro: simula carga no banco para verificar a performance;
- **IMPORTANTE:** Não se deve usar o pgbench para comparar tempo de resposta entre diferentes ambientes. O tempo varia com muitas variáveis e o resultado não é preciso;
- O pgbench deve ser utilizado em várias interações incrementais para monitorar ajustes no **mesmo ambiente**.
- Exemplo:
  - 1 Executo o pgbench;
  - 2 Anoto os resultados;
  - 3 Altero os `shared_buffers`;
  - 4 Comparo os resultados.

# Exemplos de pg\_pench

```
Postgresql.conf:  
  
max_connections = 250  
shared_buffers = 3000MB  
#temp_buffers = 8MB  
#max_prepared_transactions = 5  
work_mem = 256MB  
maintenance_work_mem = 512MB  
#max_stack_depth = 2MB  
  
pgbench -c50 -t100  
  
maquina2 = 117 tps
```

- 1 Monitoramento
  - Introdução
  - Processador
  - Memória Principal
  - Memória secundária
  - Ferramentas de monitoramento
- 2 Troubleshooting
  - **Introdução**
  - Tipos de problema
- 3 Administração assistida(?)
- 4 Referências

# O que é um problema?

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

# O que é um problema?

- Faz parte do trabalho do DBA identificar com o responsável pelo negócio o que é um problema;
- A identificação de um problema deve vir acompanhada de um planejamento em caso de catástrofe;
- Tenham em mente a máxima da computação: *Shit Happens!!!*  
Nenhum sistema é imune a falhas;
- Tristemente, só em casos de catástrofe nosso trabalho é valorizado.

- 1 Monitoramento
  - Introdução
  - Processador
  - Memória Principal
  - Memória secundária
  - Ferramentas de monitoramento
- 2 Troubleshooting
  - Introdução
  - Tipos de problema
- 3 Administração assistida(?)
- 4 Referências

# 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
```

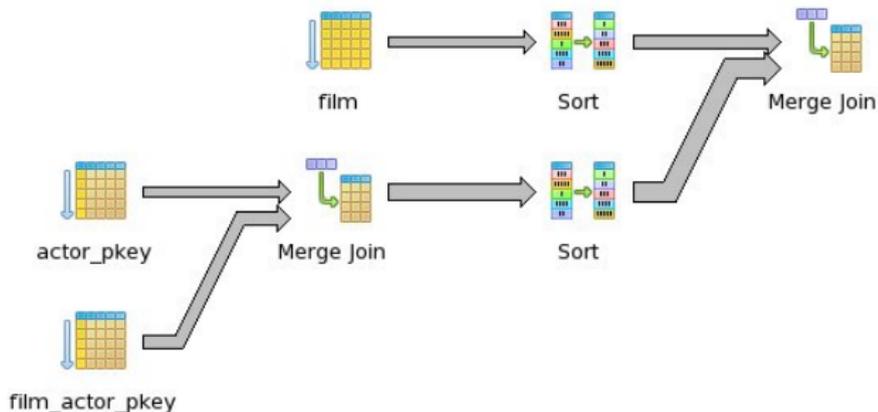
Coluna	Tipo	Modificadores
datid	oid	
datname	name	
procpid	integer	
usesysid	oid	
username	name	
current\_query	text	
waiting	boolean	
query\_start	timestamp with time zone	
backend\_start	timestamp with time zone	
client\_addr	inet	
client\_port	integer	

```
Definicao da visao:
```

```
SELECT d.oid AS datid, d.datname, pg\_stat\_get\_backend\_pid(s.backendid) AS
procpid, pg\_stat\_get\_backend\_userid(s.backendid) AS usesysid, u.rolname
AS username, pg\_stat\_get\_backend\_activity(s.backendid) AS current\_
_query, pg\_stat\_get\_backend\_waiting(s.backendid) AS waiting, pg\_stat\_
_get\_backend\_activity\_start(s.backendid) AS query\_start, pg\_stat\_get\_
_backend\_start(s.backendid) AS backend\_start, pg\_stat\_get\_backend\_
_client\_addr(s.backendid) AS client\_addr, pg\_stat\_get\_backend\_client\_port(s.backendid) AS client\_port
```

# Consultas

- Uma consulta bem escrita é aquela que gera o melhor plano de execução;
- Mais de **80%** dos problemas de performance em bancos de dados estão relacionados a **consultas SQL mal escritas**;
- EXPLAIN ANALYZE



# Consultas

- Não seja preguiçoso e não jogue seu tempo fora! Tentativa e erro demora muito e nem sempre traz os resultados esperados;
- Conheça seu banco de dados e o caminho percorrido pela consulta.

Onde está o problema na consulta abaixo?

```

Unique (cost=72049.69..72065.27 rows=1558 width=86) (actual time
=61248.918..62537.913 rows=12 loops=1)
-> Sort (cost=72049.69..72053.58 rows=1558 width=86) (actual time
=61248.913..61905.781 rows=388496 loops=1)
    Sort Key: (ts_rank_cd(to_tsvector((((aslm.name)::text || ' '::text) ||
(arlm.name)::text) || ' '::text) || (asrlm.name)::text)),
query.query), lm.landmark_id, lm.name
    Sort Method: external merge  Disk: 15544kB
-> Nested Loop (cost=6019.09..71967.07 rows=1558 width=86) (actual
time=443.145..55025.760 rows=388496 loops=1)
    Join Filter: (query.query @@ to_tsvector((((aslm.name)::text ||
' '::text) || (arlm.name)::text) || ' '::text) || (asrlm.na
me)::text)))
-> Function Scan on plainto_tsquery query (cost=0.00..0.01 rows
=1 width=32) (actual time=0.065..0.067 rows=1 loops=1)
-> Merge Join (cost=6019.09..32989.44 rows=1558170 width=54) (
actual time=417.412..30064.655 rows=6258960 loops=1)
    (...)
Total runtime: 62546.280 ms
(35 rows)

```

## 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!!!
- O bom DBA deve conhecer todos os fatores da infra que afetam o banco de dados.

# 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:

## Ações sugeridas por [Momjian, 2010]

- Falha na aplicação do Cliente: Nenhuma ação necessária. Transações sofrem ROLLBACK.
- Falha "elegante" no servidor (manda desligar):

## 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:

## 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:

## 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:

## 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):

## 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:

## 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:

## 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:

## 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:

## 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: Crie o índice novamente
- DROP DATABASE acidental:

## 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: Crie o índice novamente
- DROP DATABASE acidental: Recupere do último backup
- Instalação não inicia:

## 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: Crie o índice novamente
- DROP DATABASE acidental: Recupere do último backup
- Instalação não inicia: Normalmente problema no WAL. Veja recuperação do WAL
- Índices corrompidos;

## 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: Crie 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.

## 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: Crie 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

# Seja corajoso!!!

- Se você espera alguma ferramenta do tipo GUI, esqueça.
- Mesmo que existam ferramentas, o mais importante é conhecer o SGBD e seus componentes.
- DBA não é DBV!!!
- Vale lembrar: é papel do DBA conhecer **profundamente** o Sistema Operacional

## Ainda assim quero telinhas...

<http://www.pgfoundry.org>

Group Name	Description
 <a href="#">Wildlife Monitoring Database</a>	The Wildlife <b>Monitoring</b> 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 <b>monitoring</b> the status of elephants and more.
 <a href="#">pgTop - a top clone for PostgreSQL</a>	pgTop is a console-based (non-gui) tool for <b>monitoring</b> the processes and overall performance of a PostgreSQL server. pgTop is written in python.
 <a href="#">PostgreSQL Database Administration Tools</a>	pgtools is a <b>package</b> with usefull utilities for mantaining, <b>monitoring</b> , and administrate PostgreSQL database clusters.
 <a href="#">pgstat</a>	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 <b>monitoring</b> or performance tuning.
 <a href="#">Nagios monitoring plugins for PostgreSQL</a>	Nagios plugin scripts to monitor: transaction id status, blocked queries, long running queries, connection status and more. Please try this project 1st: <a href="http://bucardo.org/wiki/Check_postgres">http://bucardo.org/wiki/Check_postgres</a>

## Vale lembrar

- Nenhum software vai administrar o banco pra você. Ajudam se forem bem operados;
- Se você não sabe o que é partição, tipo de sistema de arquivos, paginação, estratégias de *buffering*, volte algumas casas.
- Está assustado?

## Vale lembrar

- Nenhum software vai administrar o banco pra você. Ajudam se forem bem operados;
- Se você não sabe o que é partição, tipo de sistema de arquivos, paginação, estratégias de *buffering*, volte algumas casas.
- Está assustado?
- É possível se tornar um grande DBA sem sair de casa. Acesse [www.postgresql.org/doc](http://www.postgresql.org/doc) e seja feliz!
- Infelizmente, não é possível evoluir muito em nossa profissão sem o Inglês. Se quiser ser realmente bom, domine a língua primeiro.
- Finalmente, aprenda fazendo. As empresas valorizam muito o estudo, mas para administrar dados tem que ter experiência.

-  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  
Sparkgroup  
Lightbase Consultoria em Software Público

[eduardo.santos@lightbase.com.br](mailto:eduardo.santos@lightbase.com.br)  
[eduardo.edusantos@gmail.com](mailto:eduardo.edusantos@gmail.com)

[www.postgresql.org.br](http://www.postgresql.org.br)  
[www.eduardosan.com](http://www.eduardosan.com)

+55 61 3347-1949