



UNIVERSIDADE FEDERAL DO ESTADO DO RIO DE JANEIRO
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA

Relatórios Técnicos
do Departamento de Informática Aplicada
da UNIRIO
n° 0013/2010

Testes de desempenho comparando modelo flexível com VPD tradicional

Sergio Puntar
Leonardo Azevedo
Fernanda Baião
Claudia Cappelli

Departamento de Informática Aplicada

UNIVERSIDADE FEDERAL DO ESTADO DO RIO DE JANEIRO
Av. Pasteur, 458, Urca - CEP 22290-240
RIO DE JANEIRO – BRASIL

Projeto de Pesquisa

Grupo de Pesquisa Participante



Patrocínio



PETROBRAS

Testes de desempenho comparando modelo flexível com VPD tradicional *

Sergio Puntar, Leonardo Azevedo, Fernanda Baião, Claudia Cappelli

Núcleo de Pesquisa e Prática em Tecnologia (NP2Tec)
Departamento de Informática Aplicada (DIA) – Universidade Federal do Estado do Rio de
Janeiro (UNIRIO)

sergio.puntar@uniriotec.br, azevedo@uniriotec.br, fernanda.baiao@uniriotec.br,
claudia.cappelli@uniriotec.br

Abstract. Data security is a fundamental concern in many private and public organizations. However, a fine-grained access control fosters many issues, such as policy manager and performance. This work presents performance evaluation of Oracle RBAC mechanism, the Virtual Private Database, against the flexible model for authorization rule proposed in previous works. The experimental tests were executed on model, data and queries of TPC-H benchmark.

Keywords: RBAC (Role-Based Access Control), VPD (Virtual Private Database), Benchmark TPC-H Benchmark, Flexible Model for Information Authorization, Performance analysis.

Resumo. A segurança de dados é uma preocupação fundamental em muitas organizações privadas e governamentais. Contudo a aplicação de um controle de acesso refinado trás outras preocupações como gerenciamento de políticas e impacto no desempenho. Este trabalho apresenta a avaliação prática de desempenho do mecanismo RBAC (Role-Based Access Control) da Oracle, o Virtual Private Database, quando usado na sua forma tradicional em comparação quando usado em conjunto com o modelo flexível para definição das regras de autorização proposto em trabalhos anteriores. Os testes de desempenho foram executados com base no modelo, dados e consultas do *benchmark* TPC-H.

Palavras-chave: RBAC (Role-Based Access Control), VPD (Virtual Private Database), Benchmark TPC-H, Modelo flexível para autorização de informações, Avaliação de desempenho.

* Trabalho patrocinado pela Petrobras.

Sumário

1	Introdução	vii
2	Propostas de testes de desempenho	vii
2.1	Proposta A	vii
2.2	Proposta B	ix
3	Ambiente de teste de desempenho	ix
4	Resultados dos testes experimentais	xiv
5	Análise dos resultados	xviii
6	Conclusões	xxvii
7	Referências Bibliográficas	xxviii
	Apêndice 1 – Otimização de desempenho	xxix
	Apêndice 2 – Índices criados nas tabelas consultadas	xxx

Índice de figuras

Figura 1 – Consulta para retornar o espaço ocupado pelas tabelas	x
Figura 2 - Consulta C1: Relatório do resumo de preços	xi
Figura 3 - Consulta C2: Checagem de prioridade de pedido	xii
Figura 4 - Consulta C3: Volume do fornecedor local	xii
Figura 5 - Predicado da tabela LINEITEM para a regra T1	xiii
Figura 6 - Predicado da tabela ORDERS para a regra T2	xiii
Figura 7 - Predicado VPD da tabela LINEITEM para a regra T2	xiii
Figura 8 - Predicado VPD da tabela ORDERS para a regra T3	xiii
Figura 9 - Predicado VPD da tabela LINEITEM para a regra T3	xiii
Figura 10 - Gráfico comparativo dos tempos de execução dos testes "a frio"	xv
Figura 11 – Gráfico comparativo dos tempos de execução das funções de autorização para cada regra "a frio"	xvi
Figura 12 – Gráfico comparativo dos tempos de execução dos testes "a frio"	xvii
Figura 13 – Gráfico comparativo dos tempos de execução das funções de autorização para cada regra "a quente"	xviii
Figura 14 – Gráfico da execução frio x quente com VPD tradicional	xix
Figura 15 - Gráfico da execução frio x quente com modelo flexível	xix
Figura 16 – Comparação do VPD tradicional x modelo flexível no teste a frio	xx
Figura 17 - Comparação do VPD tradicional x modelo flexível no teste a quente	xxi
Figura 18 - Comparação da utilização x não utilização de índices de predicados com VPD tradicional no teste a frio	xxii
Figura 19 - Comparação da utilização x não utilização de índices de predicados com VPD tradicional no teste a quente	xxiii
Figura 20 - Comparação da utilização x não utilização de índices de predicados com modelo flexível no teste a frio	xxiv
Figura 21 - Comparação da utilização x não utilização de índices de predicados com VPD tradicional no teste a quente	xxv
Figura 22 - Regra T1a	xxv
Figura 23 - Regra T1b	xxvi
Figura 24 – Regra T1c	xxvi
Figura 25 - Comando se sugestão de utilização de índice para a regra T1	xxvi
Figura 26 - Comando se sugestão de utilização de índice para a regra T2	xxvi
Figura 27 - Comando se sugestão de utilização de índice para a regra T3	xxvii

Índice de tabelas

Tabela 1 - Tabelas utilizadas para realização dos testes de desempenho	x
Tabela 2 - Tabelas utilizadas para realização dos testes de desempenho	x
Tabela 3 – Resultado dos testes de desempenho "a frio"	xiv
Tabela 4 - Resultado dos testes de desempenho "a frio" com índices de predicado	xiv
Tabela 5 – Tempo de execução das funções de autorização para cada regra “a frio”	xv
Tabela 6 – Resultado dos testes de desempenho "a quente"	xvi
Tabela 7 - Resultado dos testes de desempenho "a quente" com índices de predicado	xvii
Tabela 8 – Tempo de execução das funções de autorização para cada regra “a quente”	xvii
Tabela 9 - Comparação do VPD tradicional x modelo flexível no teste a frio	xx
Tabela 10 – Comparação do VPD tradicional x modelo flexível no teste a quente	xxi
Tabela 11 - Comparação da utilização x não utilização de índices de predicados com VPD tradicional no teste a frio	xxii
Tabela 12 - Comparação da utilização x não utilização de índices de predicados com VPD tradicional no teste a quente	xxiii
Tabela 13 - Comparação da utilização x não utilização de índices de predicados com modelo flexível no teste a frio	xxiii
Tabela 14 - Comparação da utilização x não utilização de índices de predicados com modelo flexível no teste a quente	xxiv
Tabela 15 -Comparação do tempo de execução das consultas sem índice, com índice e com sugestão de índice de predicado	xxvii
Tabela 16 - Tipos de política VPD	xxix

1 Introdução

Este relatório apresenta testes de desempenho da proposta de modelo flexível [Azevedo *et al.*, 2010a, 2010b] para tratar a execução de regras de autorização, implementado sobre o Virtual Private Database [Jeloka *et al.*, 2008] (funcionalidade da Oracle para autorização de acesso) em relação ao uso do Virtual Private Database tradicional.

A proposta de modelo flexível é apresentada em [Azevedo *et al.*, 2010a, 2010b], o qual apresenta a avaliação do uso da proposta em relação às operações disponibilizadas em banco de dados (*select*, *insert*, *update*, *delete*, *group by*, *having*, *select case*, *connect by* etc) de acordo com as consultas e dados do Benchmark TPC-H [TPC-H 2009].

Os testes experimentais foram realizados de acordo com consultas e dados do Benchmark TPC-H.

Esse relatório está organizado em 5 capítulos, sendo o capítulo 1 a presente introdução. No capítulo 2, são apresentadas as propostas de testes de desempenho. No capítulo 3, o ambiente de teste é apresentado. No capítulo 4, são descritos os resultados dos testes experimentais. Nos capítulos 5 e 6 são apresentadas análises dos resultados dos testes e as referências bibliográficas, respectivamente.

2 Propostas de testes de desempenho

Esta seção apresenta duas propostas para testes de desempenho. A primeira proposta (aqui denominada proposta A) foi elaborada de acordo com discussões entre os membros da equipe, enquanto que a proposta B corresponde a uma proposta de testes baseada no TPC-H. A proposta escolhida foi a proposta A, visto que os objetivos dos testes são específicos para o cenário do projeto em que a solução do VPD será aplicada, e não objetivos gerais de avaliação de desempenho como os que são endereçados pela especificação original do TPC-H.

2.1 Proposta A

Proposta levantada em discussão entre os membros da equipe.

- Volume de dados:
 - Registrar o volume de dados existentes nas tabelas que tenham políticas e que sejam acessadas por cada consulta
 - Observação: O *overhead* gerado certamente vai ter relação com o volume de dados da tabela (e conseqüentemente com o tempo total da consulta)
- Unidade de comparação
 - A unidade de comparação utilizada será o tempo.
- Cenários de execução
 - Execução "sem VPD";
 - Execução "com VPD tradicional"; e
 - Execução "com VPD usando a nossa abordagem"

- Observações:
 - Consulta sem VPD = consulta sem restrição
 - Consulta com VPD tradicional e com VPD usando nossa proposta serão praticamente iguais. A única diferença está no acesso às tabelas do modelo flexível.
- Tipo de política
 - O tipo de política permite especificar precisamente como e com qual frequência o predicado da política deve ser alterado. Isto influencia diretamente o desempenho de processamento. No Apêndice 1, são apresentados os cinco tipos de políticas que podem ser criados (estática, estática compartilhada, sensível ao contexto, sensível ao contexto compartilhada, dinâmica). Este documento apresenta os resultados de desempenho utilizando o tipo de política default do Oracle, ou seja, dinâmica.
- Índices
 - Criar índices para as tabelas do TPC-H de acordo com os predicados das consultas que serão realizadas e de acordo com os predicados retornados pelo VPD.
 - Cria índices para as tabelas do modelo genérico, de acordo com a forma que as tabelas deste modelo são acessadas.
- Carga do modelo de regras
 - Incluir várias regras no modelo, entre 1000 a 2000 regras.
- Plano de execução
 - Levantar o plano de execução utilizado para cada uma das consultas a fim de que estes sejam os mesmos para cada execução.
 - Observações:
 - Provavelmente o plano de execução será diferente quando usando o VPD e quando não usando o VPD, pois sem usar o VPD implica não ter predicados. Logo, determinados caminhos de execução necessários quando utilizando o VPD não precisarão ser percorridos quando a consulta não tiver os predicados gerados pelo VPD.
 - Por outro lado, o plano de execução da consulta usando VPD tradicional e o VPD com nossa proposta deverão ser os mesmos.
- Métricas
 - Coletar tempos de execução de cada consulta
 - "a frio" (com cache vazio - sem ter executado nenhuma outra consulta antes)
 - "a quente" (com cache cheio - tempo médio de 10 execuções da mesma consulta, desprezando a 1a execução, e, em seguida, a consulta com menor tempo e com maior tempo de execução)

2.2 Proposta B

Proposta de teste de desempenho levantada na especificação original do TPC-H.

No TPC-H, o *benchmark* é definido como a execução de um teste de carga seguido por um teste de desempenho.

O teste de carga começa com a criação das tabelas do banco de dados e inclui todas as atividades necessárias para levar o sistema sendo testado (SUT – System Under Test) para a configuração imediatamente anterior ao início do teste de desempenho.

O teste de desempenho consiste de duas rodadas. A primeira rodada é executada logo após o teste de carga. A segunda rodada é a rodada imediatamente após a rodada 1. Cada rodada consiste de uma execução do teste de força (*Power Test*), seguido de uma execução do teste de *throughput* (*Throughput Test*).

Componentes de uma rodada:

- Consulta: consulta do TPC-H escolhida para ser utilizada nos testes.
- Conjunto de consultas: execução seqüencial de cada consulta escolhida, e de todas elas.
- *Stream* de consulta: é definido como a execução de um único conjunto de consultas.
- *Stream* de função (*refresh stream*): é definido como a execução seqüencial de um número integral de pares de funções de atualização (*refresh functions*) submetidas através de um programa em *batch*. Uma função de atualização realiza inserções e remoções de dados. O *benchmark* TPC-H define duas função de atualização:
 - RF1: realiza inserções nas tabelas ORDERS e LINEITEM
 - RF2: remove todas as vendas antigas do banco de dados.
- Par de funções de atualização (*refresh function*): cada uma das duas funções apresentadas RF1 e RF2.
- Sessão: é definida como o contexto de processo capaz de suportar a execução de cada *stream* de consulta ou *stream* de função.

O *Power Test* tem o objetivo de medir a execução a frio quando conectado com um único usuário. Um único par de funções de atualização é executada exclusivamente por um *stream* de função e agendado antes e depois da execução das consultas.

O *Throughput Test* tem o objetivo de medir a capacidade do sistema de processar a maioria das consultas no menor período de tempo. Neste teste, vários pares de funções de atualização são executadas exclusivamente pelo *stream* de atualização e agendado em período de tempo definido pelo avaliador.

3 Ambiente de teste de desempenho

Os testes de desempenho foram executados em um desktop DELL com processador Intel Core 2 Duo E7400, operando a uma frequência de 2,8 GHz, com 4 GB de memória principal PC2-6400, operando a uma frequência de 400 MHz (velocidade de trans-

ferência de 6400 MB/s), e com um disco rígido de 160 GB com barramento Serial ATA 2.0 (velocidade de transferência de 300 MB/s).

A versão do Oracle utilizada foi a 10g Enterprise Edition Release 10.2.0.3.0, instalada sobre o Sistema Operacional Microsoft Windows Vista Home Basic. As consultas foram realizadas através do Oracle SQL Plus, rodando localmente na mesma máquina onde foi instalado o Oracle.

Como mencionado anteriormente, os testes foram realizados de acordo com consultas e dados do Benchmark TPC-H [TPC-H 2009]. A Tabela 1 apresenta a descrição das tabelas do benchmark e o volume de dados cadastrados em cada uma (usando a consulta apresentada na Figura 1) para a realização dos testes.

Tabela 1 - Tabelas utilizadas para realização dos testes de desempenho

Nome da tabela	Número de registros	Espaço ocupado em Kbytes
CUSTOMER	150.160	27.208
LINEITEM	6.001.204	866.112
NATION	25	40
ORDERS	1.500.000	190.200
PART	199.840	30.296
PARTSUPP	800.000	130.272
REGION	5	40
SUPPLIER	10.000	1.704

```
SELECT t.table_name AS "Table Name",
       t.num_rows AS "Rows",
       t.avg_row_len AS "Avg Row Len",
       Trunc((t.blocks * p.value)/1024) AS "Size KB",
       t.last_analyzed AS "Last Analyzed"
FROM   dba_tables t,
       v$parameter p
WHERE  t.owner = Decode(Upper('&1'), 'ALL', t.owner, Upper('&1'))
AND    p.name = 'db_block_size'
ORDER BY t.table_name;
```

Figura 1 – Consulta para retornar o espaço ocupado pelas tabelas

Para avaliar o impacto da utilização do modelo flexível em um cenário mais próximo ao real, as tabelas do modelo foram populadas com um número razoável de regras. A Tabela 2 apresenta a descrição das tabelas do modelo flexível e o volume de dados cadastrados em cada uma (também usando a consulta apresentada na Figura 1) para a realização dos testes.

Tabela 2 - Tabelas utilizadas para realização dos testes de desempenho

Nome da tabela	Número de registros	Espaço ocupado em Kbytes
USUARIO	9994	280
PERFIL_USUARIO	9996	160
PERFIL	214	40
PERFIL_REGRA	111	40
REGRA	111	40
REGRA_TABELA	61	40
TABELA	58	40
ESQUEMA	5	40

ATRIBUTO	77	40
OPERADOR	8	40
PARAMETRO	151	40
VALOR_PARAMETRO	455	40
APLICACAO_POLITICA	5	40
APLICACAO_POLITICA_ATRIBUTO	0	0
FUNCAO	1	40

Além das tabelas, algumas consultas do Benchmark TPC-H foram selecionadas para a realização dos testes. As consultas selecionadas são as mesmas utilizadas na avaliação do funcionamento do VPD, apresentadas em [Azevedo *et al.*, 2010a, 2010b].

C1. Consulta de relatório do resumo de preços: Esta consulta lista a quantidade que foi vendida, remetida e retornada (Figura 2). Para os testes consideramos um período de 90 dias.

```
select
  l_returnflag,
  l_linestatus,
  sum(l_quantity) as sum_qty,
  sum(l_extendedprice) as sum_base_price,
  sum(l_extendedprice*(1-l_discount)) as sum_disc_price,
  sum(l_extendedprice*(1-l_discount)*(1+l_tax)) as sum_charge,
  avg(l_quantity) as avg_qty,
  avg(l_extendedprice) as avg_price,
  avg(l_discount) as avg_disc,
  count(*) as count_order
from
  lineitem
where
  l_shipdate <= date '1998-12-01' - interval '90' day (3)
group by
  l_returnflag,
  l_linestatus
order by
  l_returnflag,
  l_linestatus;
```

Figura 2 - Consulta C1: Relatório do resumo de preços

C2. Consulta de checagem de prioridade de pedido: Esta consulta determina o quão bem o sistema de priorização de pedidos está funcionando e apresenta uma avaliação da satisfação dos clientes (Figura 3). Para os testes, consideramos um período de 3 meses.

```
select
  o_orderpriority,
  count(*) as order_count
from
  orders
where
  o_orderdate >= date '1993-07-01'
  and o_orderdate < date '1993-07-01' + interval '3' month
  and exists (
    select
      *
    from
      lineitem
    where
      l_orderkey = o_orderkey
```

```

        and l_commitdate < l_receiptdate
    )
group by
    o_orderpriority
order by
    o_orderpriority;

```

Figura 3 - Consulta C2: Checagem de prioridade de pedido

C3. Consulta de volume do fornecedor local: Esta consulta lista o volume de receita de fornecedores locais, ou seja, o fornecedor e o cliente estão na mesma nação (Figura 4). Para os testes consideramos a região “América” e período de 1 ano.

```

select
    n_name,
    sum(l_extendedprice * (1 - l_discount)) as revenue
from
    customer,
    orders,
    lineitem,
    supplier,
    nation,
    region
where
    c_custkey = o_custkey
and l_orderkey = o_orderkey
and l_suppkey = s_suppkey
and c_nationkey = s_nationkey
and s_nationkey = n_nationkey
and n_regionkey = r_regionkey
and r_name = 'AMERICA'
and o_orderdate >= date '1994-01-01'
and o_orderdate < date '1994-01-01' + interval '1' year
group by
    n_name
order by
    revenue desc;

```

Figura 4 - Consulta C3: Volume do fornecedor local

Em seguida, foram escolhidas e adaptadas de [Azevedo *et al.*, 2010b] as regras para as quais as políticas seriam cadastradas. Para estas regras foram definidos perfis específicos. As Figura 5, Figura 6 e Figura 7 apresentam os predicados correspondentes a cada perfil.

T1. Regra: Restringir o acesso pelo Gerente de vendas do hemisfério Norte das regiões Ásia e América aos itens de pedidos provenientes de fornecedores das nações desta área que possuem balanço na conta superior a R\$9.000,00.

Perfil: Gerente de vendas de baixo risco Ásia e América, com acesso às informações relativas aos fornecedores com balanço na conta superior a R\$9.000,00 do hemisfério Norte das regiões Ásia e América.

Predicado para tabela LINEITEM (Figura 5):

```

1=2 OR (l_suppkey in (
    select s_suppkey
    from supplier, nation, region
    where s_nationkey = n_nationkey
    and n_regionkey = r_regionkey
    and n_hemisphere IN (1)
    AND r_regionkey IN (1 , 2)

```

```
AND s_acctbal >= 9000))
```

Figura 5 - Predicado da tabela LINEITEM para a regra T1

T2. Regra: Permitir que gerentes de Marketing da nação Romênia possam acessar os itens de pedido de clientes das nações Brasil e Argentina que possuem balanço na conta superior a R\$9.000,00.

Perfil: Marketing Romênia de classe média, localizado na nação Romênia, mas com acesso a produtos das nações Brasil e Argentina de clientes com balanço na conta superior a R\$9.000,00.

Predicado para tabela ORDERS (Figura 6):

```
1=2 OR (o_custkey in (  
  select c_custkey  
  from customer, nation  
  where n_nationkey = c_nationkey  
        and n_name IN ('ARGENTINA' , 'BRAZIL')  
        AND c_acctbal >= 9000))
```

Figura 6 - Predicado da tabela ORDERS para a regra T2

Predicado para tabela LINEITEM (Figura 7):

```
1=2 OR (l_orderkey in (  
  select o_orderkey  
  from orders ))
```

Figura 7 - Predicado VPD da tabela LINEITEM para a regra T2

Para verificar a necessidade da definição de índices para os predicados, foi criada uma nova regra, cujos predicados seriam altamente beneficiados pela utilização de índices. Assim como nos casos anteriores, foi definido um perfil específico para a regra. As Figura 8 e Figura 9 apresentam os predicados correspondentes ao perfil.

T3. Regra: Restringir o acesso pelo Vendedor aos itens de pedido cujo valor não exceda R\$8.000,00 e aos pedidos cujo valor não exceda R\$50.000,00.

Perfil: Vendedor de baixo risco, com acesso somente aos itens de pedido com valor até R\$8.000,00 e somente aos pedidos com valor total até R\$50.000,00.

Predicado VPD para a tabela ORDERS (Figura 8):

```
1=2 OR (o_totalprice <= (8000))
```

Figura 8 - Predicado VPD da tabela ORDERS para a regra T3

Predicado VPD para a tabela LINEITEM (Figura 9):

```
1=2 OR (l_extendedprice <= (50000))
```

Figura 9 - Predicado VPD da tabela LINEITEM para a regra T3

Como o objetivo deste trabalho é avaliar o VPD tradicional em relação ao modelo flexível, as implementações correspondentes a estas duas propostas foram realizadas, para os testes:

- VPD tradicional: foram criadas *functions* para retornar o predicado para cada uma das políticas e estas foram cadastradas para cada tabela envolvida.

- Modelo flexível: os dados necessários para retornar o predicado das políticas foram inseridos no modelo flexível e a *function* genérica foi associada às tabelas envolvidas.

4 Resultados dos testes experimentais

As consultas dos testes de desempenho foram executadas através do Oracle SQL Plus 10.2.0.3.0, que forneceu o tempo de execução de cada uma delas, através do uso do comando SET TIMING aplicado antes de executar as consultas.

A primeira rodada de testes foi feita “a frio”, ou seja, o Oracle Database foi reiniciado antes da execução de cada consulta, garantindo que ela fosse realizada com o *cache* vazio.

A Tabela 3 apresenta os tempos (em segundos) de execução “a frio” das consultas sem aplicação política de autorização (ou seja sem a influência do VPD ou do modelo flexível), das consultas com a influência do VPD tradicional e das consultas com a influência do modelo flexível [Azevedo *et al.*, 2010a, 2010b]. Para os dois últimos casos, o teste foi repetido para cada regra que afetava cada consulta. Além disso, para estes testes, foram criados índices de acordo somente com as consultas C1, C2 e C3. Ou seja, foram criados índices para chaves primárias, chaves estrangeiras e para os campos envolvidos nos predicados das consultas. Não foram criados índices para os campos envolvidos nos predicados de autorização.

Tabela 3 – Resultado dos testes de desempenho "a frio"

Consulta	Sem controle de autorização (seg.)	Regra	VPD tradicional (seg.)	Modelo flexível (seg.)
C1	20,91	T1	36,07	39,57
		T2	51,46	53,43
		T3	13,29	12,80
C2	14,36	T1	31,01	29,59
		T2	56,10	52,60
		T3	45,80	44,28
C3	15,73	T1	18,06	17,48
		T2	102,55	102,18
		T3	15,10	16,00

Contudo, também é importante avaliar o ganho de desempenho com a utilização de índices criados de acordo com os campos envolvidos nos predicados de autorização. Dessa forma, os testes foram repetidos após a criação dos índices para cada predicado. Os comandos de criação de todos os índices são apresentados no Apêndice 2. A Tabela 4 apresenta os tempos (em segundos) de execução “a frio” das consultas com influência das políticas de autorização nas duas abordagens e com índices criados para os predicados.

Tabela 4 - Resultado dos testes de desempenho "a frio" com índices de predicado

Consulta	Regra	VPD tradicional com índices de predicado (seg.)	Modelo flexível com índices de predicado (seg.)
C1	T1	37,04	38,90

	T2	53,54	52,15
	T3	12,16	13,18
C2	T1	32,10	29,99
	T2	54,51	52,43
	T3	46,97	53,05
C3	T1	19,62	20,59
	T2	100,58	109,42
	T3	15,89	15,45

A Figura 10 apresenta o gráfico comparativo dos tempos de execução de todos os testes realizados “a frio”.

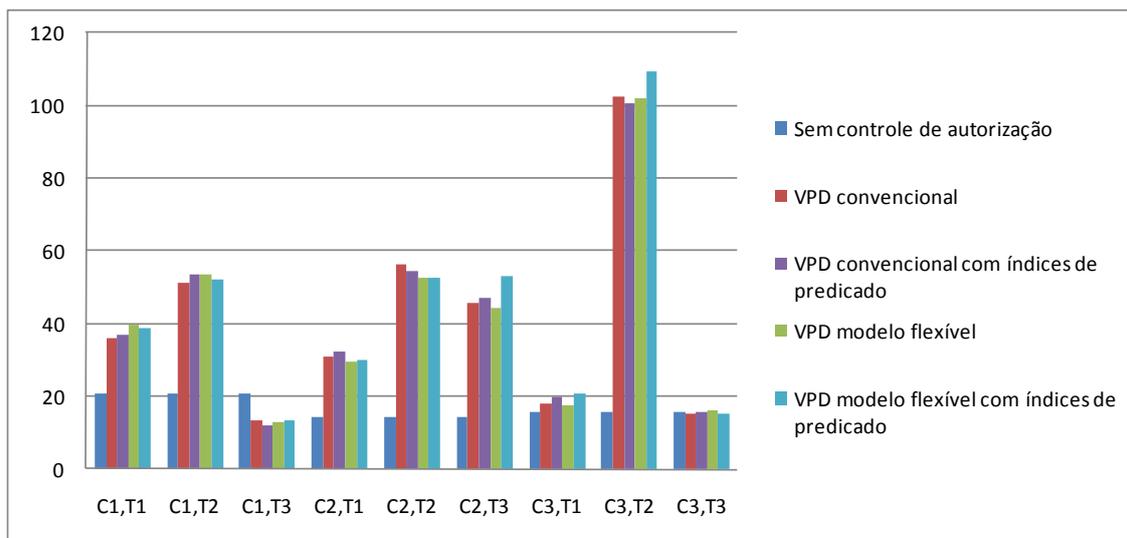


Figura 10 - Gráfico comparativo dos tempos de execução dos testes "a frio"

Outro aspecto importante analisado nos testes de desempenho é o tempo de execução das funções de autorização do VPD tradicional versus o tempo de execução da função genérica do modelo flexível para cada regra. Com esse teste podemos ter uma idéia do custo de acesso às tabelas do modelo flexível. A Tabela 5 apresenta os tempos (em segundos) de execução “a frio” das funções de autorização para cada regra.

Tabela 5 – Tempo de execução das funções de autorização para cada regra “a frio”

Regra	Função de autorização do VPD tradicional (seg.)	Função genérica do modelo flexível (seg.)	Diferença (seg.)	Diferença (%)
T1	0,105	0,390	0,285	271%
T2	0,100	0,430	0,330	330%
T3	0,095	0,680	0,585	616%

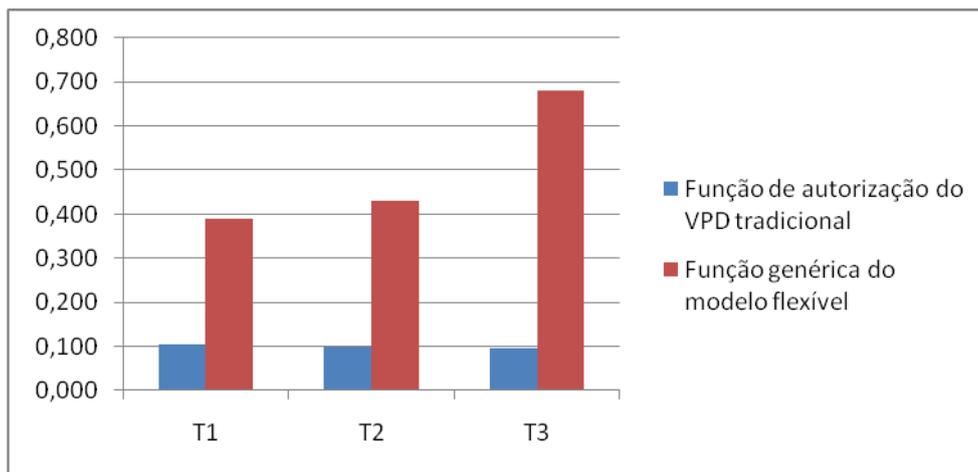


Figura 11 – Gráfico comparativo dos tempos de execução das funções de autorização para cada regra “a frio”

Através desses dados podemos observar que as funções de autorização do VPD tradicional levam aproximadamente o mesmo tempo de execução, afinal são muito parecidas. Em contrapartida, a função genérica do VPD com modelo flexível apesar de ser a mesma para todas as regras, apresenta uma considerável variação percentual. Isso acontece porque a função genérica faz acesso às tabelas do modelo flexível e, conseqüentemente, o seu tempo de execução será afetado pela quantidade de dados recuperada nas tabelas, assim como pela quantidade de dados cadastrados nas tabelas.

Apesar da variação percentual dos tempos de execução das funções de cada abordagem ser grande, a variação em segundos é ínfima. Logo, quanto menor o tempo de execução da consulta, maior é o impacto resultante da solução utilizando o modelo flexível, enquanto que quanto maior o tempo de execução da consulta menor é este impacto. Neste caso, podemos dizer que o tempo de execução das funções exerce pouca influência no desempenho do modelo.

Em seguida os testes foram realizados “a quente”, ou seja, para cada teste, a mesma consulta foi repetida dez vezes seguidas, e foi tomada a média dos tempos, descartando a primeira execução, e após este descarte, foram descartados o menor tempo e o maior tempo de execução.

Da mesma forma que no caso anterior, os testes foram realizados sem aplicação política de autorização (ou seja sem a influência do VPD ou do modelo flexível), com a influência do VPD tradicional e com a influência do modelo flexível [Azevedo *et al.*, 2010a, 2010b]. É importante ressaltar que, para as consultas com a influência das políticas de autorização, o teste foi repetido para cada regra que as afetava. O resultado dos tempos de execução (em segundos) da segunda rodada de testes é apresentado na Tabela 6.

Tabela 6 – Resultado dos testes de desempenho “a quente”

Consulta	Sem controle de autorização (seg.)	Regra	VPD tradicional (seg.)	Modelo flexível (seg.)
C1	17,13	T1	32,61	33,22
		T2	41,95	42,29
		T3	10,45	10,59

C2	13,73	T1	29,64	28,35
		T2	44,02	42,93
		T3	2,53	3,04
C3	14,76	T1	14,95	14,98
		T2	16,77	18,73
		T3	15,18	15,29

Assim como no “a frio”, os testes foram repeditos após a criação dos índices para cada predicado. Os comandos de criação de todos os índices são apresentados no Apêndice 2.

A Tabela 7 apresenta os tempos (em segundos) de execução “a quente” das consultas com influência do VPD nas duas abordagens e com índices de predicado.

Tabela 7 - Resultado dos testes de desempenho "a quente" com índices de predicado

Consulta	Regra	VPD tradicional com índices de predicado (seg.)	Modelo flexível com índices de predicado (seg.)
C1	T1	33,27	33,17
	T2	41,56	41,29
	T3	10,2	10,72
C2	T1	28,92	27,87
	T2	44,06	44,57
	T3	0,27	0,33
C3	T1	15,09	14,73
	T2	17,25	16,27
	T3	14,76	15,19

A Figura 10 apresenta o gráfico comparativo dos tempos de execução de todos os testes realizados “a quente”.

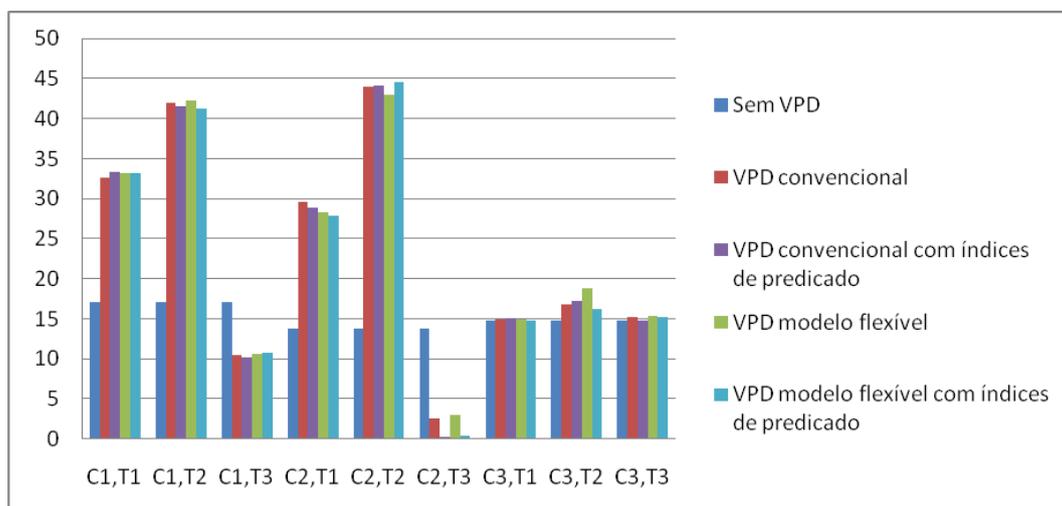


Figura 12 – Gráfico comparativo dos tempos de execução dos testes "a frio"

A Tabela 5 apresenta os tempos (em segundos) de execução “a quente” das funções de autorização para cada regra.

Tabela 8 – Tempo de execução das funções de autorização para cada regra “a quente”

Regra	Função de autorização	Função genérica do	Diferença	Diferença
-------	-----------------------	--------------------	-----------	-----------

	do VPD tradicional (seg.)	modelo flexível (seg.)	(seg.)	(%)
T1	0,048	0,060	0,012	20%
T2	0,071	0,117	0,046	39%
T3	0,034	0,089	0,089	61%

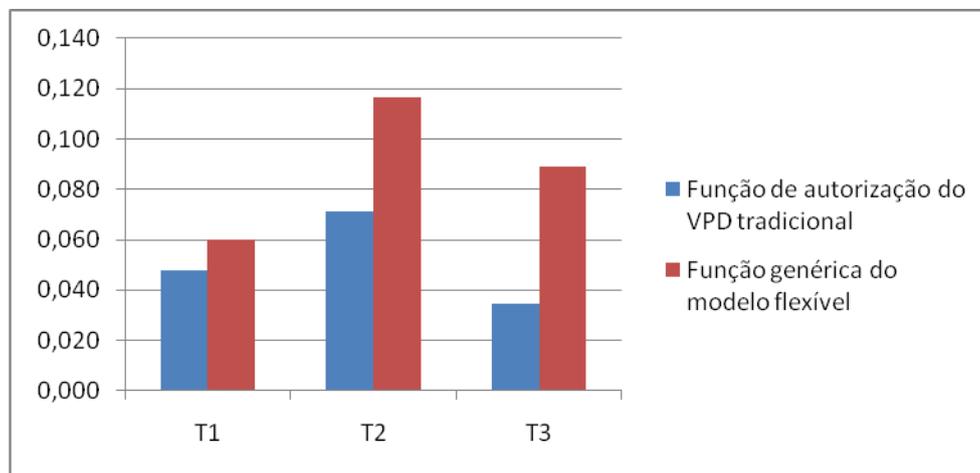


Figura 13 – Gráfico comparativo dos tempos de execução das funções de autorização para cada regra “a quente”

A variação percentual dos tempos de execução das funções de cada abordagem considerando o teste “a quente” é muito menor em relação ao tempo a frio, variando de 20 a 61%. Já a variação em segundos é ínfima. Logo, considerando o a diferença de tempo, podemos afirmar que o tempo de execução das funções exerce pouca influência no desempenho do modelo.

5 Análise dos resultados

Esta seção apresenta a análise dos resultados obtidos de acordo com as informações das Tabela 3, Tabela 4, Tabela 5, Tabela 6, Tabela 7 e Tabela 8.

As execuções das consultas sem controle de autorização nem sempre são mais rápidas do que as execuções com controle de autorização, como pode ser observado no gráfico da Figura 10. Isto ocorre porque o controle de autorização pode adicionar à consulta um predicado que restringe os dados sendo consultados (diminuindo a quantidade de linhas retornadas) de modo a acelerar o processamento. Como exemplo considere a consulta C1 e regra T3 (Tabela 4), cuja consulta sem controle de autorização demanda mais tempo de processamento do que a consulta com controle de autorização.

Como esperado, o teste a frio é mais lento do que o teste a quente, como ilustrado na Figura 14 e na Figura 15.

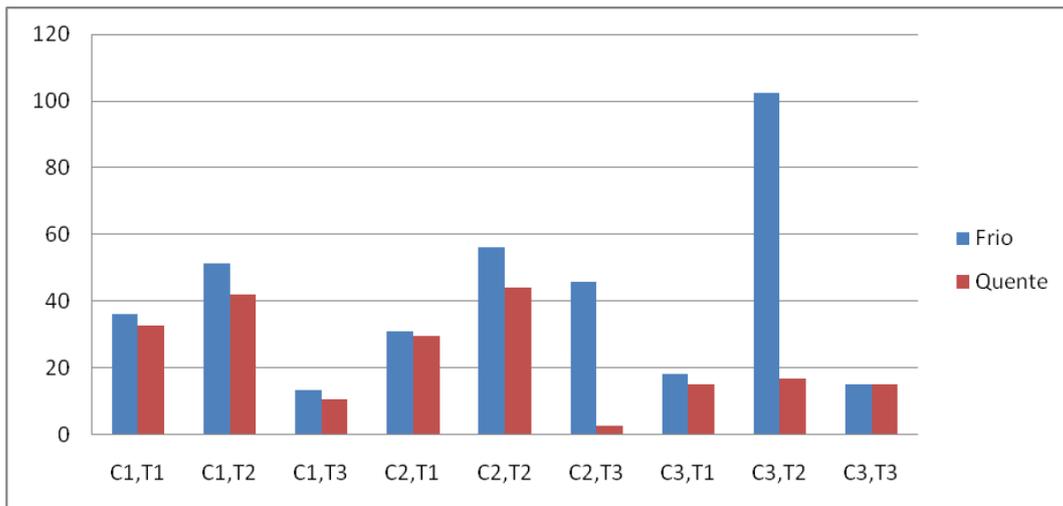


Figura 14 – Gráfico da execução frio x quente com VPD tradicional

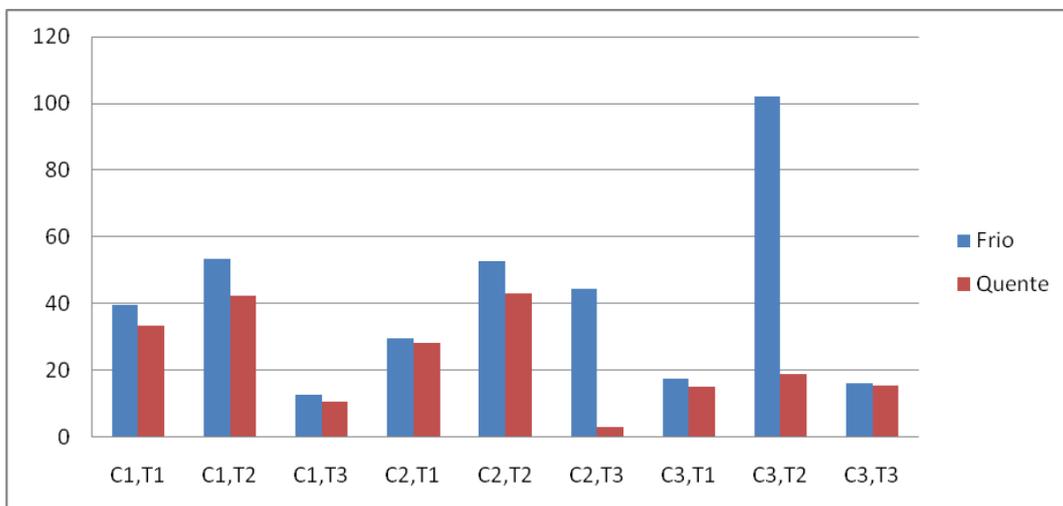


Figura 15 - Gráfico da execução frio x quente com modelo flexível

Comparando os testes a frio, a Tabela 9 (e no respectivo gráfico da Figura 16) mostra que o modelo flexível foi mais lento do que o VPD tradicional em três casos e mais rápido em seis casos, como apresentado na Tabela 9 e na Figura 16. Analisando estes resultados em relação ao tempo de execução das funções apresentados na Tabela 5 (e no respectivo gráfico da Figura 11), podemos afirmar que o tempo de execução das regras de autorização utilizando o modelo flexível não influencia o desempenho das consultas. Esta afirmação é baseada no fato de que o tempo de execução das regras de autorização é percentualmente muito maior do que o tempo de execução das regras de autorização com o VPD tradicional e mesmo assim houve mais casos das execuções serem mais rápidas com o modelo flexível do que com o VPD tradicional. Isto se dá porque a diferença considerando o tempo (propriamente dito) é muito pequena (0,285 segundos a 0,585 segundos).

Ainda em relação a Tabela 9, apesar dos tempos apresentarem uma grande variação percentual (de -6,24% a 9,7%), a variação em segundos (de -3,5 segundos a 3,5 segundos) é pequena considerando o tempo total tomado por cada consulta. Considerando a diferença de tempo em segundos (coluna 4) e em valor absoluto, dividido pelo menor tempo de execução da consulta com VPD tradicional e com o modelo flexível (menor valor entre as colunas 2 e 3), o acréscimo de uma abordagem em relação a ou-

tra varia de 0% a 10%. Em média esta diferença é de 5%. Considerando os benefícios que se pode alcançar com o controle de autorização a nível de dados em operações executadas em ambientes transacionais, este acréscimo de tempo é muito pequeno.

Tabela 9 - Comparação do VPD tradicional x modelo flexível no teste a frio

Regra	VPD tradicional (seg.)	Modelo Flexível (seg.)	Diferença (seg.)	Diferença (%)	Diferença em relação menor tempo de execução (%)
C1, T1	36,07	39,57	3,50	9,70%	10%
C1, T2	51,46	53,43	1,97	3,83%	4%
C1, T3	13,29	12,80	-0,49	-3,69%	4%
C2, T1	31,01	29,59	-1,42	-4,58%	5%
C2, T2	56,10	52,60	-3,50	-6,24%	7%
C2, T3	45,80	44,28	-1,52	-3,32%	3%
C3, T1	18,06	17,48	-0,58	-3,21%	3%
C3, T2	102,55	102,18	-0,37	-0,36%	0%
C3, T3	15,10	16,00	0,90	5,96%	6%
Média	41,05	40,88	-0,17	-0,21%	5%

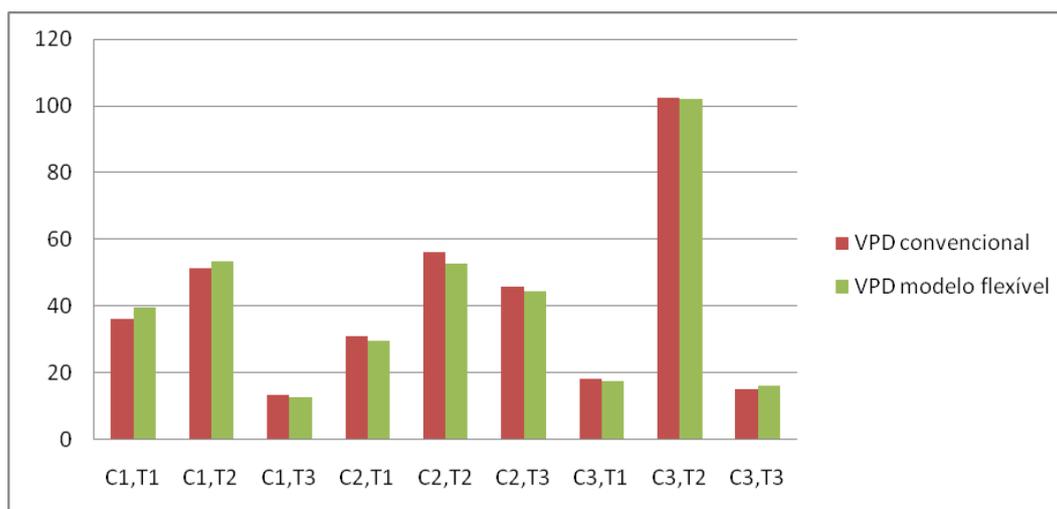


Figura 16 – Comparação do VPD tradicional x modelo flexível no teste a frio

Comparando os testes a quente, a Tabela 10 (e no respectivo gráfico da Figura 17) mostra que o modelo flexível foi mais lento do que o VPD tradicional em sete casos e mais rápido em dois casos. Analisando estes resultados em relação ao tempo de execução das funções apresentados na Tabela 8 (e no respectivo gráfico da Figura 12), podemos afirmar que o tempo de execução das regras de autorização utilizando o modelo flexível não influencia o desempenho das consultas. Esta afirmação é baseada no fato de que o tempo de execução das regras de autorização é percentualmente maior do que o tempo de execução das regras de autorização com o VPD tradicional e mesmo assim houve casos das execuções serem mais rápidas com o modelo flexível do que com o VPD tradicional. Isto se dá porque a diferença considerando o tempo (propriamente dito) é muito pequena (0,012 segundos a 0,117 segundos).

Ainda em relação à Tabela 10, assim como no teste anterior, os tempos apresentaram uma grande variação percentual (de -4,35% a 20,16%), porém uma pequena variação em segundos (de -1,29 segundos a 1,96 segundos). Considerando a diferença de tempo em segundos (coluna 4) e em valor absoluto, dividido pelo menor tempo de execução da consulta com VPD tradicional e com o modelo flexível (menor valor entre as colunas 2 e 3), o acréscimo de uma abordagem em relação a outra varia de 0% a 20%. Sendo 12% o segundo maior valor. Em média esta diferença é de 5%. Dado que o caso em que a variação foi de 20% foi para uma consulta com tempo de execução muito pequeno e considerando os benefícios que se pode alcançar com o controle de autorização a nível de dados em operações executadas em ambientes transacionais, este acréscimo de tempo é muito pequeno.

Tabela 10 – Comparação do VPD tradicional x modelo flexível no teste a quente

Regra	VPD tradicional (seg.)	Modelo Flexível (seg.)	Diferença (seg.)	Diferença (%)	Diferença em relação menor tempo de execução (%)
C1, T1	32,61	33,22	0,61	1,87	2%
C1, T2	41,95	42,29	0,34	0,81	1%
C1, T3	10,45	10,59	0,14	1,34	1%
C2, T1	29,64	28,35	-1,29	-4,35	5%
C2, T2	44,02	42,93	-1,09	-2,48	3%
C2, T3	2,53	3,04	0,51	20,16	20%
C3, T1	14,95	14,98	0,03	0,20	0%
C3, T2	16,77	18,73	1,96	11,69	12%
C3, T3	15,18	15,29	0,11	0,72	1%
Média	23,12	23,27	0,15	3,33	5%

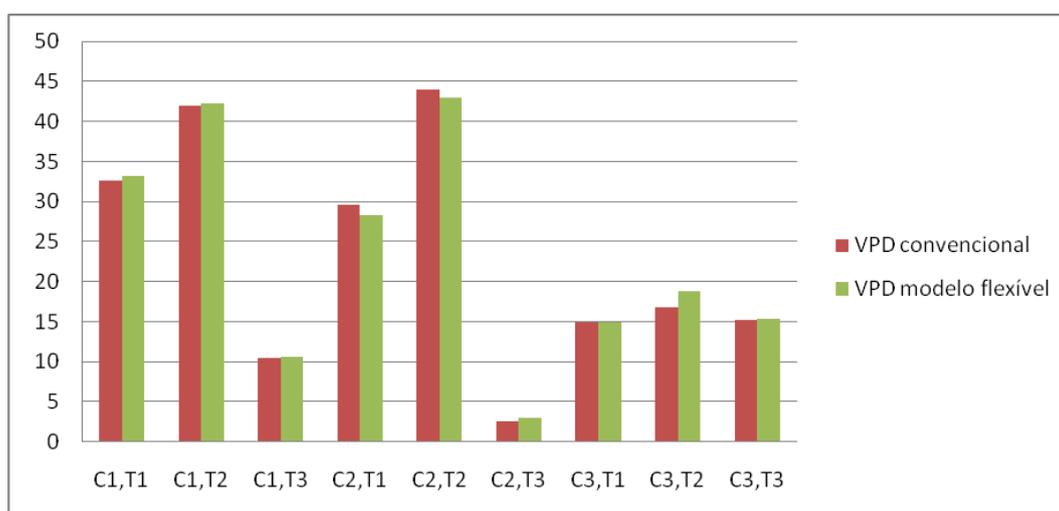


Figura 17 - Comparação do VPD tradicional x modelo flexível no teste a quente

Os dados dos testes a frio e a quente nos mostram que, na maioria das vezes, a diferença entre os tempos das consultas com o VPD tradicional e com o modelo flexível não passa de 10% do tempo total de cada consulta, ficando muitas vezes abaixo de 1 segundo. Podemos observar ainda que isso é verdade tanto para os casos em que o

VPD tradicional é mais rápido, quanto para os casos onde o modelo flexível é mais rápido. Dessa forma podemos considerar as duas abordagens equivalentes em relação ao desempenho.

Comparando a utilização do VPD tradicional com e sem os índices de predicado, a Tabela 11 mostra que nos testes a frio a utilização do índice de predicado foi mais lenta em seis casos e mais rápida em três casos. Logo, a inclusão de índices nem sempre resulta em aceleração do processamento. A variação percentual dos tempos de execução ficou entre -8,5% e 8,64%, enquanto a variação em segundos ficou entre -1,97 segundos e 2,08 segundos. Na maioria dos testes a diferença de tempo foi muito pequena considerando a execução da consulta considerando política sem índice para os predicados e com índice para os predicados.

Tabela 11 - Comparação da utilização x não utilização de índices de predicados com VPD tradicional no teste a frio

Regra	Sem índice de predicado (seg.)	Com índice de predicado (seg.)	Diferença (seg.)	Diferença (%)
C1, T1	36,07	37,04	0,97	2,69
C1, T2	51,46	53,54	2,08	4,04
C1, T3	13,29	12,16	-1,13	-8,50
C2, T1	31,01	32,10	1,09	3,51
C2, T2	56,10	54,51	-1,59	-2,83
C2, T3	45,80	46,97	1,17	2,55
C3, T1	18,06	19,62	1,56	8,64
C3, T2	102,55	100,58	-1,97	-1,92
C3, T3	15,10	15,89	0,79	5,23

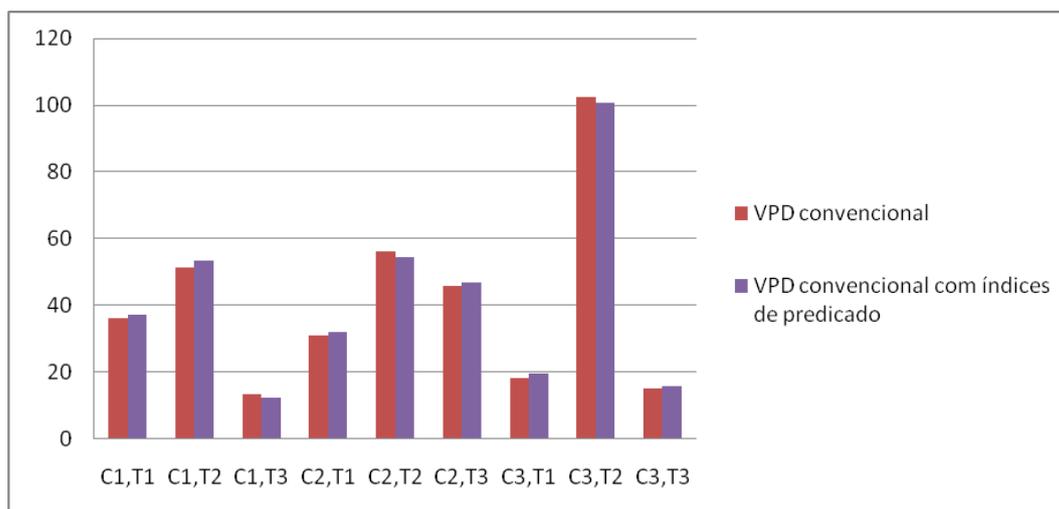


Figura 18 - Comparação da utilização x não utilização de índices de predicados com VPD tradicional no teste a frio

Em contrapartida, a Tabela 12 mostra que nos testes a quente a utilização do índice de predicado foi mais lenta em quatro casos e mais rápida em três casos. Novamente, a inclusão de índices nem sempre resulta em aceleração do processamento.

A variação percentual dos tempos de execução ficou entre -89,33% e 2,86%, enquanto a variação em segundos ficou entre -2,26 segundos e 0,66 segundos. O maior ganho no uso de índice apareceu na execução da consulta C2 com a regra T3, isto se deve ao fato de que a tabela consultada é *Orders*, a qual tem 1.500.000 e o predicado sobre a coluna *o_totalprice* acelera muito a consulta.

Tabela 12 - Comparação da utilização x não utilização de índices de predicados com VPD tradicional no teste a quente

Regra	Sem índice de predicado (seg.)	Com índice de predicado (seg.)	Diferença (seg.)	Diferença (%)
C1, T1	32,61	33,27	0,66	2,02
C1, T2	41,95	41,56	-0,39	-0,93
C1, T3	10,45	10,20	-0,25	-2,39
C2, T1	29,64	28,92	-0,72	-2,43
C2, T2	44,02	44,06	0,04	0,09
C2, T3	2,53	0,27	-2,26	-89,33
C3, T1	14,95	15,09	0,14	0,94
C3, T2	16,77	17,25	0,48	2,86
C3, T3	15,18	14,76	-0,42	-2,77

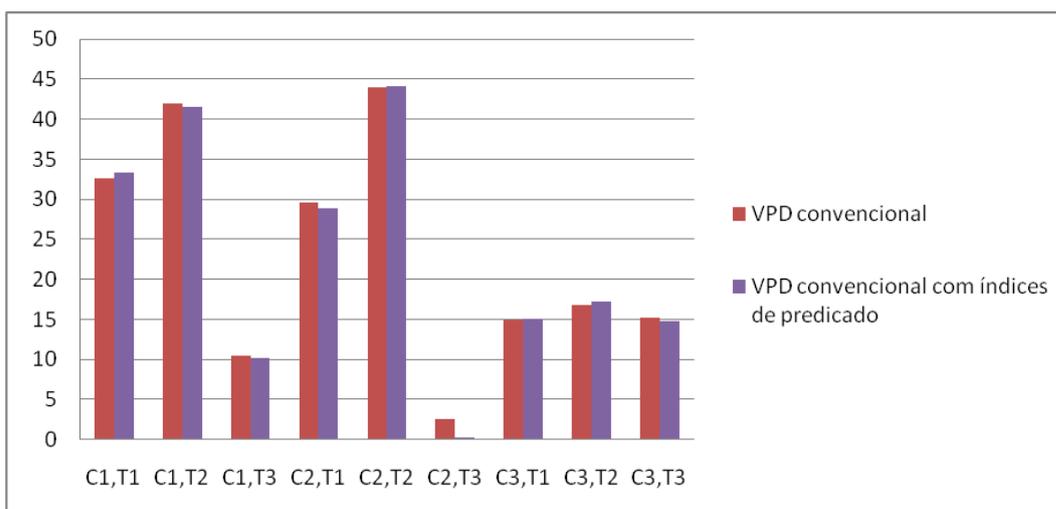


Figura 19 - Comparação da utilização x não utilização de índices de predicados com VPD tradicional no teste a quente

Comparando a utilização do modelo flexível com e sem os índices de predicado, a Tabela 13 mostra que nos testes a frio a utilização do índice de predicado foi mais lenta em cinco casos e mais rápida em quatro casos. Assim como nos casos anteriores, a inclusão de índices nem sempre resulta em aceleração do processamento. A variação percentual dos tempos de execução ficou entre -3,44% e 19,81%, enquanto a variação em segundos ficou entre -1,28 segundos e 8,77 segundos.

Tabela 13 - Comparação da utilização x não utilização de índices de predicados com modelo flexível no teste a frio

Regra	Sem índice de	Com índice de	Diferença	Diferença
-------	---------------	---------------	-----------	-----------

	predicado (seg.)	predicado (seg.)	(seg.)	(%)
C1, T1	39,57	38,90	-0,67	-1,69
C1, T2	53,43	52,15	-1,28	-2,40
C1, T3	12,80	13,18	0,38	2,97
C2, T1	29,59	29,99	0,40	1,35
C2, T2	52,60	52,43	-0,17	-0,32
C2, T3	44,28	53,05	8,77	19,81
C3, T1	17,48	20,59	3,11	17,79
C3, T2	102,18	109,42	7,24	7,09
C3, T3	16,00	15,45	-0,55	-3,44

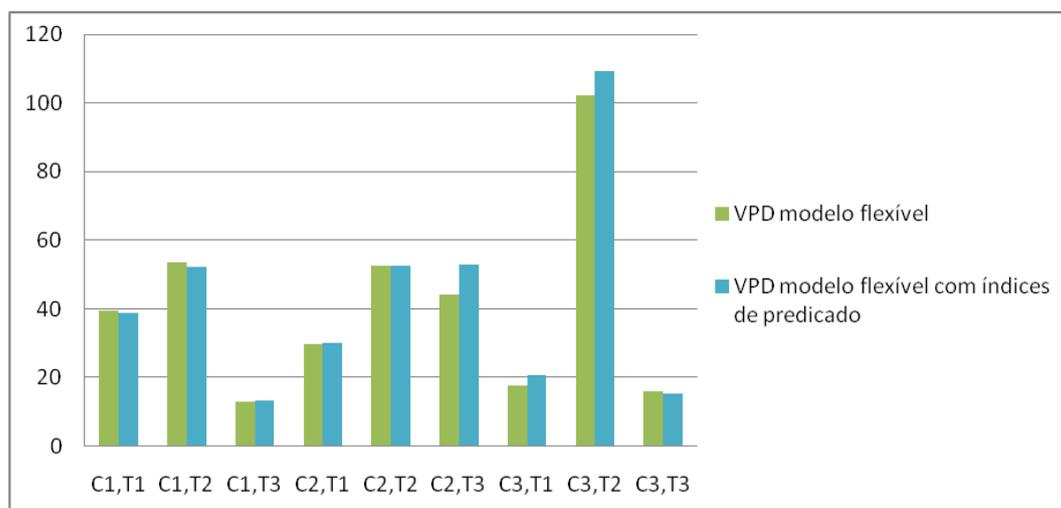


Figura 20 - Comparação da utilização x não utilização de índices de predicados com modelo flexível no teste a frio

Em contrapartida, a Tabela 14 mostra que nos testes a quente a utilização do índice de predicado foi mais lenta em dois casos e mais rápida em sete casos. Mais uma vez a inclusão de índices não resultou sempre em aceleração do processamento. A variação percentual dos tempos de execução ficou entre -89,14% e 3,82%, enquanto a variação em segundos ficou entre -2,71 segundos e 1,64 segundos. Assim como no teste do modelo tradicional, o maior ganho no uso de índice apareceu na execução da consulta C2 com a regra T3.

Tabela 14 - Comparação da utilização x não utilização de índices de predicados com modelo flexível no teste a quente

Regra	Sem índice de predicado (seg.)	Com índice de predicado (seg.)	Diferença (seg.)	Diferença (%)
C1, T1	33,22	33,17	-0,05	-0,15
C1, T2	42,29	41,29	-1,00	-2,36
C1, T3	10,59	10,72	0,13	1,23
C2, T1	28,35	27,87	-0,48	-1,69
C2, T2	42,93	44,57	1,64	3,82

C2, T3	3,04	0,33	-2,71	-89,14
C3, T1	14,98	14,73	-0,25	-1,67
C3, T2	18,73	16,27	-2,46	-13,13
C3, T3	15,29	15,19	-0,10	-0,65

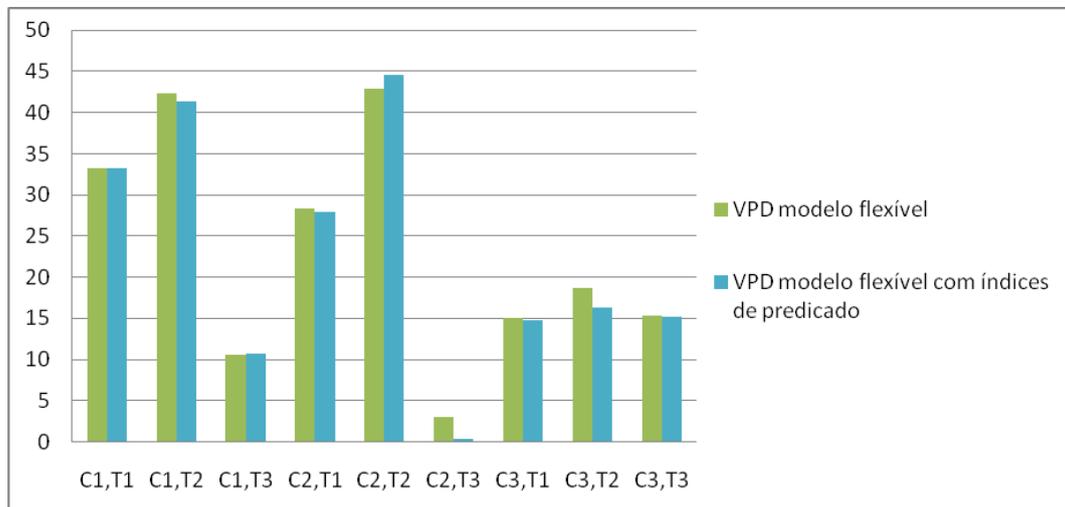


Figura 21 - Comparação da utilização x não utilização de índices de predicados com VPD tradicional no teste a quente

Os dados dos testes de comparação da utilização ou não dos índices de predicado nos mostram que, excetuando-se o caso citado da consulta C2 com a política T3, a maioria das consultas apresentou uma diferença pequena entre a utilização ou não do índice. Através da análise dos planos de consulta, foi possível observar que isso acontece porque o otimizador de plano de consulta do Oracle não utilizou os índices definidos no Apêndice 2.

Levantou-se então a questão de que em alguns predicados os índices não foram utilizados por estar sendo usado o operador IN. Para verificar essa possibilidade, a regra da política T1 foi reescrita de duas formas que evitassem o operador IN nos parâmetros da regra. A Figura 22 e a Figura 23 apresentam as duas novas versões da regra.

```

1=2 OR (l_suppkey in (
  select s_suppkey
  from tpch.supplier, tpch.nation, tpch.region
  where s_nationkey = n_nationkey
    and n_regionkey = r_regionkey
    and n_hemisphere = 1
    and (r_regionkey = 1 or r_regionkey = 2)
    and s_acctbal >= (9000))

```

Figura 22 - Regra T1a

```

1=2 OR (l_suppkey in (
  select s_suppkey
  from tpch.supplier, tpch.nation, tpch.region
  where s_nationkey = n_nationkey
    and n_regionkey = r_regionkey
    and n_hemisphere = 1
    and r_regionkey = 1
    and s_acctbal >= (9000)

```

```

union
select s_suppkey
from tpch.supplier, tpch.nation, tpch.region
where s_nationkey = n_nationkey
and n_regionkey = r_regionkey
and n_hemisphere = 1
and r_regionkey = 2
and s_acctbal >= (9000))

```

Figura 23 - Regra T1b

Em seguida os testes a quente foram repetidos para essas duas novas regras utilizando a abordagem do VPD convencional. A regra T1 original obteve uma média de tempo de 32,61 segundos, enquanto a regra T1a obteve uma média de tempo de 34,77 e a regra T1b uma média de tempo de 77,93 segundos. Portanto podemos concluir que o operador IN não influenciou na escolha do otimizador do Oracle.

Observe que, apesar das adaptações não utilizarem o operador IN nos parâmetros da regra, ele ainda é utilizado na definição da regra (`l_suppkey in`). Esse operador IN especificamente é muito difícil de ser substituído, pois como a regra é dinâmica, ela depende diretamente de outra consulta. Uma forma alternativa de escrever o predicado é utilizar o operador EXISTS em detrimento do IN. Para verificar a eficiência dessa alteração, a regra T1 foi novamente reescrita como é apresentada na Figura 24.

```

1=2 OR exists (
select *
from tpch.supplier, tpch.nation, tpch.region
where l_suppkey = s_suppkey
and s_nationkey = n_nationkey
and n_regionkey = r_regionkey
and n_hemisphere = 1
and (r_regionkey = 1 or r_regionkey = 2)
and s_acctbal >= (9000))

```

Figura 24 – Regra T1c

A consulta C1 foi então executada com a regra T1c a quente, obtendo uma média de tempo de execução de 34,59 segundos, muito próxima dos 32,61 segundos obtidos pela execução com a regra T1 original. Dessa forma podemos dizer que, na questão do tempo de execução, a utilização do operador EXISTS é equivalente à utilização do operador IN.

Para verificar o desempenho caso os índices fossem utilizados, a consulta C3 foi executada novamente, porém com a adição dos comandos de sugestão de utilização de índice. Esses comandos fazem com que o otimizador do Oracle utilize os índices sugeridos para realizar a consulta. A Figura 25 apresenta o comando de sugestão de índice para regra T1, a Figura 26 apresenta o comando de sugestão de índice para a regra T2 e a Figura 27 apresenta o comando de sugestão de índice para a regra T3.

```

/*+ index(SUPPLIER S ACCTBALL_IDX) */

```

Figura 25 - Comando de sugestão de utilização de índice para a regra T1

```

/*+ index(CUSTOMER C ACCTBAL_IDX) */

```

Figura 26 - Comando de sugestão de utilização de índice para a regra T2

```

/*+ index(ORDERS O_TOTALPRICE_IDX) */
/*+ index(LINEITEM L_EXTENDEDPRI_PRICE_IDX) */

```

Figura 27 - Comando se sugestão de utilização de índice para a regra T3

A consulta C3 foi executada duas vezes para cada regra, e para cada execução foi adicionado o comando de sugestão de utilização de índice para a regra em questão. Esta consulta foi escolhida por envolver várias junções entre tabelas. A abordagem utilizada foi a do modelo flexível, e elas foram executadas a frio. A Tabela 15 apresenta a comparação da média do tempo em segundos de execução das consultas sem índice, com índice e com sugestão de utilização de índice.

Tabela 15 -Comparação do tempo de execução das consultas sem índice, com índice e com sugestão de índice de predicado

Regra	Sem índice de predicado (seg.)	Com índice de predicado (seg.)	Com sugestão de índice de predicado (seg.)
C3, T1	17,48	20,59	19,51
C3, T2	102,18	109,42	103,84
C3, T3	16,00	15,45	813,5

Tanto para a regra T1 quanto para a regra T2 (segunda e terceira linhas da tabela), os tempos de execução com a sugestão de utilização de índice foi equivalente aos tempos sem a sugestão. Por outro lado, para a regra T3 (quarta linha da tabela) o tempo de execução com a sugestão de utilização de índice foi muito pior que os tempos sem a sugestão. Dessa forma podemos concluir que o otimizador fez a escolha certa ao não utilizar os índices.

Quatro casos interessantes devem ser ressaltados:

- A consulta C2 com a política T3 teve um percentual superior a 1700% maior para a execução a frio em relação à execução a quente, o que é observado comparando-se a sexta linha da Tabela 9 com a sexta linha da Tabela 10. Isto se deve ao fato de que na execução da consulta C2 o Oracle utilizou *full scan* na tabela. Como a tabela consultada é a tabela *Orders* que tem 1.500.000 registros, esta operação é mais lenta na execução a frio (quando o *cache* está vazio para cada execução) do que na execução a quente (quando os dados estão no *cache*).
- A consulta C3 com a política T2 teve um percentual superior a 500% maior para a execução a frio em relação à execução a quente, o que é observado comparando-se a penúltima linha da Tabela 9 com a penúltima linha da Tabela 10. Isto se deve ao fato de que na execução da consulta C3 o Oracle utilizou *full scan* nas tabelas *Customer*, *Supplier*, *Orders* e *Lineitem*. Considerando as tabelas *Orders* e *LineItem* que têm 1.500.000 e 6.001.204 registros, respectivamente, esta operação é mais lenta na execução a frio (quando o *cache* está vazio para cada execução) do que na execução a quente (quando os dados estão no *cache*).
- A consulta C2 com a política T3 teve um ganho de mais de 80% com a utilização de índices de predicado nos testes a quente, enquanto nos testes a frio chegou a ter uma perda de mais de 19%. Novamente, esta diferença de tempo pode ser atribuída aos dados estarem armazenados no *cache*.

6 Conclusões

Este relatório apresentou a avaliação experimental do modelo flexível proposto por Azevedo *et al.* [2010a, 2010b] em relação à implementação do Virtual Private Database [Jeloka *et al.*, 2008].

O modelo flexível monta o predicado de autorização a partir de dados cadastrados em um modelo genérico, enquanto que no VPD tradicional as funções de autorização tem o código todo definido (estático), sem ocorrer construção do predicado em tempo de execução. O modelo flexível é mais lento que o VPD tradicional em alguns casos e mais rápido em outros casos. Como a diferença de tempo é muito pequena, as vantagens de flexibilidade de uso da proposta para gestão das regras de autorização em um local centralizado (todas as funções são armazenadas em um único modelo) demonstra ser viável o uso do modelo flexível perante o VPD tradicional.

Como trabalho futuro, propomos os testes da proposta em relação aos outros tipos de políticas, além do tipo dinâmico: *Static*, *Context-Sensitive*, e *Shared*. Os tipos de políticas *static* e *context-sensitive* otimizam o desempenho do VPD.

- Políticas estáticas mantêm o mesmo predicado para consultas, atualizações, inserções e remoções através de uma sessão. Entretanto o contexto da aplicação ou atributos como SYSDATE podem alterar o valor retornado pelo predicado. Elas são úteis para os ambientes sempre que é necessário aplicar o mesmo predicado.
- Políticas sensíveis ao contexto podem ser alteradas depois da compilação, mas o VPD reexecuta a política apenas se o contexto da aplicação é alterado. Isto garante que qualquer mudança que ocorra no predicado será aplicada. Políticas sensíveis ao contexto são úteis quando políticas VPD deve garantir predicados diferentes para diferentes usuários ou grupos.

Políticas estáticas e sensíveis ao contexto podem ser compartilhadas através de múltiplos objetos de bancos de dados, a fim de que a política aplicada sobre outro objeto use o mesmo predicado que está no *cache*.

Além disso, propomos a execução dos testes experimentais sobre dados reais, como trabalho futuro.

7 Referências Bibliográficas

AZEVEDO, L. G., PUNTAR, S., THIAGO, R., BAIÃO, F., CAPPELLI, C. **A Flexible Framework for Applying Data Access Authorization Business Rules**. In: 12th International Conference on Enterprise Information Systems (ICEIS 2010), Funchal, Madeira, Portugal, June 8-12, 2010a.

AZEVEDO, L.; PUNTAR, S.; MELO, R.; BAIÃO, F.; CAPPELLI, C. **Avaliação Prática de Funcionalidades para Autorização de Informações (Label Security e Virtual Private Database)**. Relatórios Técnicos do DIA/UNIRIO (RelaTe-DIA), RT-0002/2010, 2010b. Disponível em: <http://seer.unirio.br/index.php/monografiasppgi>.

JELOKA, S.; MULAGUND, G.; LEWIS N. *et al.*. **Oracle Database 10gR2 Security Guide**. Oracle, 2008.

TPCH, **TPC Benchmark H**. Transaction Processing Performance Council. Disponível em <http://www.tpc.org/tpch/>. Acesso em: 10 set. 2009.

Apêndice 1 – Otimização de desempenho

A execução das funções das políticas pode consumir uma quantidade significativa de recursos do sistema. Se for possível diminuir o número de vezes que essas funções devem ser executadas, então também é possível otimizar o desempenho do servidor do banco de dados.

Para evitar execuções desnecessárias das funções das políticas, o Oracle apresenta cinco tipos diferentes de políticas [JELOKA *et al.*, 2008], o que nos permite especificar como e com que frequência o predicado da política deve mudar. A Tabela 16 apresenta os tipos de política disponíveis, o momento de execução da função e a propriedade de compartilhamento de predicado em cada uma.

Tabela 16 - Tipos de política VPD

Tipo de política	Execução da função	Predicado compartilhado
STATIC	Uma vez, o predicado é então armazenado na SGA (System Global Area).	Não
SHARED_STATIC	Uma vez, o predicado é então armazenado na SGA (System Global Area).	Sim
CONTEXT_SENSITIVE	O objeto protegido é acessado pela primeira vez. O contexto de aplicação se alterou desde a última execução da função.	Não
SHARED_CONTEXT_SENSITIVE	O objeto protegido é acessado pela primeira vez. O contexto de aplicação se alterou desde a última execução da função.	Sim
DYNAMIC	Sempre que o objeto protegido é acessado.	Não

As políticas estáticas e as sensíveis ao contexto permitem a otimização de desempenho, pois não executam a função cada vez que um objeto protegido é acessado. Contudo, antes de habilitar a política como estática ou sensível ao contexto, é recomendável testá-la como dinâmica para ter certeza que a função funciona corretamente. Além disso, se nenhum tipo for especificado para uma política, ela assume por padrão o tipo dinâmico, onde a função é executada cada vez que o objeto protegido é acessado.

- **Políticas estáticas**

Nas políticas estáticas a função é executada apenas uma vez, e o predicado é então armazenados na SGA (System Global Area). Nesse caso o mesmo predicado é aplicado a todos os usuários da instância do banco.

Geralmente o predicado de uma política estática é baseado em atributos recuperados pela função SYS_CONTEXT, por isso, apesar do predicado ser estático, o resultado da consulta será diferente de acordo com o valor dos atributos de contexto de cada usuário.

Exemplo de uso: Considere um *data warehouse* que armazena dados de pesquisa de mercado para organizações concorrentes entre si. A *warehouse* deve definir uma política onde cada organização deve ter acesso somente aos seus próprios dados, que é expressa pelo predicado `WHERE subscriber_id=SYS_CONTEXT('customer', 'cust_num')`. O uso da função

SYS_CONTEXT permite que o banco altere dinamicamente o conjunto de registros retornado, sem a necessidade de gerar outro predicado.

A política estática compartilhada funciona da mesma forma que a política estática comum, contudo permite que o predicado da política seja compartilhado com outros objetos do banco.

- **Políticas sensíveis ao contexto**

Nas políticas sensíveis ao contexto a função é executada da primeira vez que o objeto protegido é acessado, e o predicado é então armazenado na UGA (User Global Area). Nesse caso cada sessão terá o seu predicado armazenado na sua UGA.

O predicado será reutilizado até que haja uma alteração no contexto de aplicação. Nesse caso, a função será executada novamente e o novo predicado será armazenado na UGA para ser reutilizado até que o contexto mude mais uma vez. Esse tipo de política é útil quando os predicados diferem de acordo com o usuário que executa a consulta.

Exemplo de uso: Considere uma tabela SALES_HISTORY com uma política que define que analistas só podem ver seus próprios produtos e empregados regionais só podem ver suas próprias regiões. Nesse caso, o servidor deve re-executar a função somente quando o usuário mudar. O ganho de desempenho se dá quando um mesmo usuário executa várias consultas seguidas sem a necessidade de gerar o predicado em cada uma delas.

Assim como no caso anterior, a política sensível ao contexto compartilhada funciona da mesma forma que a política sensível ao contexto comum, contudo permite que o predicado da política seja compartilhado com outros objetos do banco.

Apêndice 2 – Índices criados nas tabelas consultadas

A seguir são listados os comandos de criação dos índices das tabelas do modelo flexível e das tabelas do TPC-H.

- **Índices do modelo flexível**

Os índices das tabelas do modelo flexível foram criados de acordo com a forma que as tabelas deste modelo são acessadas.

```
CREATE INDEX "PERFIL"."ATR_NOME_IDX" ON "PERFIL"."ATRIBUTO"  
("ATR_NOME")  
PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS  
STORAGE(INITIAL 65536 NEXT 65536 MINEXTENTS 1 MAXEXTENTS  
2147483645 PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1  
BUFFER_POOL DEFAULT)  
TABLESPACE "PERFIL";
```

```
CREATE INDEX "PERFIL"."PRFI_DT_EXPIRACAO_IDX" ON  
"PERFIL"."PERFIL" ("PRFI_DT_EXPIRACAO")  
PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS  
STORAGE(INITIAL 65536 NEXT 65536 MINEXTENTS 1 MAXEXTENTS  
2147483645 PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1  
BUFFER_POOL DEFAULT)  
TABLESPACE "PERFIL";
```

```
CREATE INDEX "PERFIL"."PU_DT_EXPIRACAO_IDX" ON
"PERFIL"."PERFIL_USUARIO" ("PU_DT_EXPIRACAO")
PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS
STORAGE(INITIAL 65536 NEXT 65536 MINEXTENTS 1 MAXEXTENTS
2147483645 PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1
BUFFER_POOL DEFAULT)
TABLESPACE "PERFIL";
```

```
CREATE INDEX "PERFIL"."TB_NOME_IDX" ON "PERFIL"."TABELA"
("TB_NOME")
PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS
STORAGE(INITIAL 65536 NEXT 65536 MINEXTENTS 1 MAXEXTENTS
2147483645 PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1
BUFFER_POOL DEFAULT)
TABLESPACE "PERFIL";
```

- **Índices das tabelas do TPC-H para as consultas**

Os índices das tabelas do TPC-H foram criados primeiramente de acordo com as consultas que seriam realizadas.

```
CREATE INDEX "TPCH"."L_SHIPDATE_IDX" ON "TPCH"."LINEITEM"
("L_SHIPDATE")
PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS
STORAGE(INITIAL 65536 NEXT 65536 MINEXTENTS 1 MAXEXTENTS
2147483645 PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1
BUFFER_POOL DEFAULT)
TABLESPACE "TPCH";
```

```
CREATE INDEX "TPCH"."O_ORDERDATE_IDX" ON "TPCH"."ORDERS"
("O_ORDERDATE")
PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS
STORAGE(INITIAL 65536 NEXT 65536 MINEXTENTS 1 MAXEXTENTS
2147483645 PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1
BUFFER_POOL DEFAULT)
TABLESPACE "TPCH";
```

```
CREATE INDEX "TPCH"."R_NAME_IDX" ON "TPCH"."REGION" ("R_NAME")
PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS
STORAGE(INITIAL 65536 NEXT 65536 MINEXTENTS 1 MAXEXTENTS
2147483645 PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1
BUFFER_POOL DEFAULT)
TABLESPACE "TPCH";
```

- **Índices das tabelas do TPC-H para os predicados VPD**

Novos índices para as tabelas do TPC-H foram criados de acordo com os predicados VPD das regras definidas.

```
CREATE INDEX "TPCH"."S_ACCTBAL_IDX" ON "TPCH"."SUPPLIER"
("S_ACCBAL")
PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS
STORAGE(INITIAL 65536 NEXT 65536 MINEXTENTS 1 MAXEXTENTS
2147483645 PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1
BUFFER_POOL DEFAULT)
```

```
TABLESPACE "TPCH";
```

```
CREATE INDEX "TPCH"."C_ACCTBAL_IDX" ON "TPCH"."CUSTOMER"  
("C_ACCTBAL")  
PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS  
STORAGE(INITIAL 65536 NEXT 65536 MINEXTENTS 1 MAXEXTENTS  
2147483645 PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1  
BUFFER_POOL DEFAULT)  
TABLESPACE "TPCH";
```

```
CREATE INDEX "TPCH"."O_TOTALPRICE_IDX" ON "TPCH"."ORDERS"  
("O_TOTALPRICE")  
PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS  
STORAGE(INITIAL 65536 NEXT 65536 MINEXTENTS 1 MAXEXTENTS  
2147483645 PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1  
BUFFER_POOL DEFAULT)  
TABLESPACE "TPCH";
```

```
CREATE INDEX "TPCH"."L_EXTENDEDPRICE_IDX" ON "TPCH"."LINEITEM"  
("L_EXTENDEDPRICE")  
PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS  
STORAGE(INITIAL 65536 NEXT 65536 MINEXTENTS 1 MAXEXTENTS  
2147483645 PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1  
BUFFER_POOL DEFAULT)  
TABLESPACE "TPCH";
```