

BENCHREL - BENCHMARK PARA A AVALIAÇÃO DE DESEMPENHO DE SGBD'S RELACIONAIS

Maria Antonieta Rossatto (*)

Programa de Engenharia de Sistemas - COPPE
Universidade Federal do Rio de Janeiro
Caixa Postal 68.511
21.945 - Rio de Janeiro - RJ, Brasil

ABSTRACT

This work describes some aspects of benchmarks. Static and dynamic aspects of performance evaluation and fundamental features of benchmarks are presented. A methodology for development of benchmarks is described and a comparative study of the DBMS's is developed to evaluate their functionality. Finally, some benchmarks are implemented to evaluate the performance of the DBMS's. The results are examined and the criticals features of the DBMS's are identified.

(*) Eletrobrás - Centrais Elétricas Brasileiras S/A

I. INTRODUÇÃO

A área de banco de dados tem se desenvolvido, nas últimas décadas, de forma marcante. Novos SGBD's têm sido lançados no mercado, tornando grande a quantidade de produtos disponíveis e forçando possíveis usuários a escolher entre os que se diferenciam no tipo, objetivos, características, capacidade, desempenho, funcionalidade e custo. Entretanto, dificilmente, encontra-se um SGBD "ótimo" que atende todos os requisitos e resolve todos os problemas da instalação.

A análise de desempenho de um SGBD não deve estar associada ao objetivo de escolher o melhor, no sentido absoluto da palavra. Deve-se interpretá-la como um modo de escolher o melhor SGBD que atenda os requisitos da instalação e objetivos da empresa.

O programa de Engenharia de Sistemas da COPPE/UFRJ desenvolveu um SGBD relacional COPPEREL, que entrou em operação em 1983. Este é utilizado em pesquisas na universidade e serve de base para vários projetos. Entretanto, está incompleto e precisa ser otimizado para incorporar novas características. O desenvolvimento de um benchmark para avaliar o desempenho de SGBD's Relacionais permite analisar o COPPEREL e identificar as otimizações que devem ser feitas no mesmo.

II. METODOLOGIA PARA ANÁLISE DE DESEMPENHO DE SGBD'S

O desempenho de um SGBD depende de alguns fatores. Deve-se considerar parâmetros estáticos e dinâmicos. Parâmetros estáticos são, geralmente, usados para analisar funcionalidade. Parâmetros dinâmicos são, geralmente, usados para analisar tempo e implementados através do uso de benchmarks.

Existem SGBD's que possuem resultados de desempenho muito semelhantes e funcionalidades muito diferentes. Logo, é necessário analisar não somente os parâmetros dinâmicos, mas também os parâmetros estáticos para escolher um SGBD. A análise dos parâmetros estáticos deve preceder a análise dos dinâmicos. Uma vez conhecida a funcionalidade de um SGBD, torna-se mais fácil avaliar seu desempenho, pois são conhecidos recursos disponíveis, componentes integrantes e formas de utilização.

Definiu-se, então, uma metodologia composta de parâmetros estáticos e dinâmicos que será utilizada para avaliar o desempenho dos SGBD's e será apresentada a seguir.

II.1 PARÂMETROS ESTÁTICOS

Foram definidos os seguintes parâmetros estáticos: **Estrutura física e organização do espaço de armazenamento, Estrutura de dados, Dicionário de dados, Linguagem de definição e manipulação de dados, Interface com o usuário, Integração com outros SGBD's, Qualidade da documentação, Segurança e proteção dos dados, Detecção de falhas e reconstrução do banco de dados, Concorrência, Portabilidade e flexibilidade, Gerador de aplicações e Ambiente operacional.** Em alguns casos, torna-se necessário analisar a arquitetura do SGBD, e utilitários disponíveis. Assim, permite-se que instalações usem a potencialidade máxima de cada produto, adaptem-o as suas necessidades e otimizem suas tarefas. Informações adicionais podem ser obtidas em [Ross91].

II.2 PARÂMETROS DINÂMICOS

Benchmarks tem sido utilizados, com sucesso, na avaliação de desempenho de SGBD's. Os resultados de desempenho representam o desempenho dos produtos em relação a um determinado tipo de aplicação e não devem ser generalizados para todos os tipos de aplicações, pois um SGBD não se comporta igual em todas. Estas devem representar situações reais que ocorrem no ambiente em teste, simular o funcionamento do SGBD e gerar resultados de desempenho representativos e próximos da realidade da empresa.

Caso não seja possível codificar subpartes de uma aplicação na linguagem do SGBD, deve-se excluí-las dos benchmarks de alto nível que testarão cada produto. Deste modo, evita-se que os resultados gerados fiquem distorcidos e insignificantes quando comparados entre si. Deve-se incluir, somente, as transações implementáveis em todos os SGBD's num benchmark de alto nível.

A execução dos benchmarks nos SGBD's gera resultados de desempenho. Estes variam com a aplicação, recursos e ambiente operacional. Logo, devem representar o mais fielmente possível o desempenho do SGBD, pois, geralmente, servem de base para os profissionais de banco de dados escolher seus produtos.

Existem dois tipos de benchmarks para SGBD's: baixo nível e alto nível. Os de alto nível analisam SGBD's através da execução de uma aplicação completa e coleta dos resultados. Os de baixo nível analisam SGBD's através da execução de partes isoladas de uma aplicação e coleta dos resultados. Em SGBD's

relacionais, deve-se dividir uma aplicação em transações onde cada uma é composta de uma ou mais operações relacionais. Para cada transação, executa-se o benchmark de baixo nível, obtêm-se resultados de desempenho e compara-se os SGBD's.

O benchmark deve ser simples, fácil de utilizar e requerer um esforço mínimo do usuário para sua execução. Simplicidade não implica em perda de poder como muitas pessoas pensam. É errado pensar que para analisar um SGBD complexo, deve-se utilizar um benchmark complexo. Existem benchmarks simples que analisam sistemas complexos de forma segura e eficiente.

III. METODOLOGIA PARA DESENVOLVIMENTO DE BENCHMARKS

A construção de benchmarks é uma tarefa difícil e complexa, que envolve diversos fatores. A metodologia apresentada em [Ston85] serviu de base para a construção da metodologia que será apresentada. Esta metodologia será composta de três fases:

a. Projeto

Estabelece-se configuração e condições ambientais nos SGBD's em teste, aplicação e dados a serem utilizados, e parâmetros de comparação. Estes parâmetros contabilizam os desempenhos obtidos durante a execução do benchmark e medem tempo.

Após a definição dos parâmetros de comparação, estabelece-se uma forma real de medir tempo para computar os parâmetros, que permita a interpretação rápida e sem ambiguidade dos resultados de desempenho. Geralmente, utiliza-se o relógio da máquina hospedeira. A leitura deve ocorrer imediatamente antes e depois da execução de cada evento a ser analisado, e os resultados anotados para evitar perda de informações.

b. Execução

Executa-se o benchmark e obtêm-se os resultados de desempenho. Inicialmente, deve-se executar o benchmark em modalidade monousuária, mesmo que o ambiente operacional em uso seja multiusuário. O processo deve ser repetido várias vezes, de forma a confirmar e garantir os resultados de desempenho.

Finalmente, caso o ambiente operacional seja multiusuário, executa-se o benchmark em modalidade multiusuária e obtêm-se o desempenho dos SGBD's em condições reais. Repete-se o processo diversas vezes, para confirmar os resultados. Assim, reduz-se a influência de fatores externos como compartilhamento dos dados do banco de dados, compartilhamento da CPU e dispositivos de E/S, multiprogramação, concorrência, paralelismo e deadlock.

c. Análise

Analisa e compara os resultados de desempenho de cada SGBD. Em ambientes multiusuário, deve-se considerar a influência de fatores externos nos resultados de desempenho, pois estes podem diferenciar o comportamento dos SGBD's nos dois ambientes.

Finalmente, escolhe-se o "melhor" SGBD para a instalação, com base nos parâmetros estáticos e dinâmicos. Durante a análise, o usuário deve ter clareza dos objetivos do benchmark e dos parâmetros de comparação. Assim, gera-se medidas de tempo que retratam interesses e necessidades do usuário, e servem para comparar características e desempenhos reais de SGBD's.

III.1 PARÂMETROS DE COMPARAÇÃO

Foram definidos alguns parâmetros que serão utilizados pelo benchrel para avaliar o desempenho dos SGBD's. A medida dos tempos depende do critério estabelecido na fase de projeto.

a. Tempo de resposta

É o intervalo de tempo entre a submissão da aplicação e a apresentação dos resultados, isto é, o intervalo de tempo entre a chamada do SGBD e apresentação dos resultados das consultas.

b. Tempo de Execução

É o intervalo de tempo entre o início e o fim do processamento. É o somatório do tempo de CPU com o de E/S. Não é considerado o tempo de espera por recursos em ambientes multiusuário.

c. Tempo de CPU

É o tempo de processamento.

d. Tempo de E/S

É o tempo de acesso aos dados situados no banco de dados.

e. Tempo de operação

É o intervalo de tempo entre o início e o fim do processamento de uma operação. Deve-se construir vários benchmarks de baixo nível que executem cada operação em separado, garantindo-se que cada operação não sofre influência de outras operações.

f. Tempo de improdutividade ("overhead")

É o tempo gasto pelo sistema para executar tarefas que não contribuem de forma líquida para o funcionamento do SGBD. Em [Turb87], a improdutividade é calculada executando-se uma "consulta vazia", isto é, sem comandos de CPU e E/S.

g. Produtividade ("Throughput")

É o volume de processamento realizado pelo SGBD num intervalo de tempo. Geralmente, a produtividade é medida em função do número de operações ou transações realizadas por segundo.

IV. ESCOLHA DOS SGBD's

Os benchmarks propostos serão utilizados para testar alguns SGBD's existentes no mercado de microcomputadores. Foram escolhidos 7 SGBD's relacionais: Copperel, dBase III/Plus, Dialog Plus/c, Oracle, Paradox, Rbase V, e Zim. A escolha resultou de pesquisas sobre os SGBD's. Foram mantidos contatos com representantes nacionais que se mostraram interessados na utilização destes em [Ross91].

Os resultados de desempenho gerados pelo benchmark precisam ser analisados. Para tanto, é necessário conhecer com precisão a funcionalidade de cada SGBD. Estudos dos produtos, permitiram avaliar suas características e capacidades, obedecendo os parâmetros estáticos fixados na metodologia. A seguir será apresentado um quadro comparativo da funcionalidade dos SGBD's.

QUADRO COMPARATIVO DA FUNCIONALIDADE DOS SGBD's

	COPPEREL	DBASEIII/PLUS	DIALOG PLUS/C	ORACLE
a.	!Indexação por !"Hashing"	!Indexação por !Árvore B+	!Indexação por !Árvore B+	!Indexação por !Árvore B+ e !Concentração
b.	!Caracter !Numérico !Inteiro !Natural !Data	!Caracter !Numérico !Lógico !Memo !Data	!Caracter !Numérico !Lógico !Memo !Data !Inteiro	!Caracter !Numérico !Data !Binário
c.	!Esquema	!Catálogo	!Catálogo	!DD ativo
d.	!Doperel	!dBase	!Dialog	!SQL*PLUS
e.	!Interativa !sem menus	!Interativa !por menus	!Interativa !por menus	!Interativa !por menus
f.	!nenhuma	!nenhuma	!c/ DBASE	!total c/SGBDs !que usam SQL,
g.	!português	!inglês	!português	!inglês
h.	!senhas !níveis acesso	!senhas !níveis acesso	!senhas !níveis acesso	!senhas !níveis acesso
i.	!Journal Image !AI e BI files	!sem mecanismo !restaurador	!sem mecanismo !restaurador	!Journal Image !AI e BI files
j.	!Bloqueio	!Bloqueio	!Bloqueio	!Bloqueio
k.	!Mono e !Multiusuário	!Mono e !Multiusuário	!Mono e !Multiusuário	!Mono e !Multiusuário
l.	!PC, UNIX, !IBM, UNISYS,	!PC	!PC, UNIX, !Cobra	!Todos os !ambientes

Figura IV.1

QUADRO COMPARATIVO DA FUNCIONALIDADE DOS SGBD's (Cont.)

	PARADOX	IRBASE SYSTEM V	ZIM
a.	Identificação por chaves	Indexação por Árvore B+	Indexação por Árvore B+
b.	Caracter Numérico Inteiro Data Monetário	Caracter Numérico Inteiro Data Hora Monetário	Caracter Numérico Inteiro Data
c.	IDD ativo	IDD ativo	IDD não ativo
d.	PAL	Rbase/SQL	ADL
e.	Interativa sem menus	Interativa por menus	Interativa por menus
f.	p/DBASE	c/DBASE c/SGBDs que usam SQL	c/DBASE c/SGBDs que usam SQL
g.	Inglês	Inglês	Inglês
h.	senhas níveis acesso	senhas níveis acesso	senhas níveis acesso
i.	restauração parcial	sem mecanismo restaurador	Journal Image! AI e BI files!
j.	Bloqueio	Bloqueio	Bloqueio
k.	Mono e Multiusuário	Mono e Multiusuário	Mono e Multiusuário
l.	PC, IBM, VAX	PC	PC, IBM, VAX, COBRA, UNIX,

Figura IV.2

V. ESPECIFICAÇÃO, PROJETO E EXECUÇÃO DOS BENCHMARKS

A especificação, projeto e desenvolvimento dos benchmarks tomou como base a metodologia proposta. Estes utilizaram aplicações reais em seus testes para melhor simular o funcionamento dos SGBD's na instalação em uso e, gerar resultados de desempenho mais representativos e próximos da realidade. Entretanto, em ambientes com vários tipos de aplicações, a construção de um benchmark para cada tipo, é uma tarefa custosa. A solução foi a construção de uma aplicação única, simples e genérica que englobou os requisitos e objetivos daquelas da instalação.

De forma a permitir o desenvolvimento de um benchmark representativo de ambientes reais de banco de dados, foram analisadas as aplicações desenvolvidas na Eletrobrás Centrais Elétricas Brasileiras S. A., e observadas as operações mais frequentes. Apesar dos SGBD's utilizados não serem exatamente os mesmos que serão testados nesta tese, as transações envolvidas em aplicações de banco de dados independem do SGBD em uso e empregam combinações de operações de forma a atender seus objetivos. Entretanto, a amostra obtida é restrita e dependente das aplicações analisadas, e contém um conjunto reduzido de transações, sendo representativa somente para os ambientes considerados, não devendo ser generalizada para todos os ambientes de banco de dados.

Para construir a amostra de transações foram estabelecidas classes de transações típicas, de forma a distribuir os valores encontrados por classe e apurar a frequência de cada uma dentro da amostra. Para tanto, definiu-se uma transação típica T_i como uma combinação de operações sobre relações. Cada T_i usa relações como operandos e produz uma relação como resultado. A seguir, serão definidas as transações T_i .

T_1 = Seleção
 T_2 = Projecção
 T_3 = União
 T_4 = Intersecção
 T_5 = Diferença
 T_6 = Divisão
 T_7 = Produto Cartesiano
 T_8 = Junção
 T_9 = $T_1 + T_8$ = Seleção + Junção
 T_{10} = $T_5 + T_3$ = Subtração + União
 T_{11} = $T_1 + T_5 + T_3$ = Seleção + Subtração + União
 T_{12} = $T_1 + T_2$ = Seleção + Projecção
 T_{13} = $T_1 + T_5$ = Seleção + Subtração
 T_{14} = $T_1 + T_2 + T_8$ = Seleção + Projecção + Junção
 T_{15} = $T_1 + T_3$ = Seleção + União
 T_{16} = $T_{14} + T_{10}$ = Seleção + Projecção + Junção + Subtração + União
 T_{17} = $T_{14} + T_5$ = Seleção + Projecção + Junção + Subtração
 T_{18} = $T_{14} + T_3$ = Seleção + Projecção + Junção + União

A escolha destas transações deve-se ao fato de serem estas as mais utilizadas pelas aplicações da instalação analisada. Uma análise detalhada das aplicações da Eletrobrás permitiu gerar uma distribuição de frequência. A seguir, serão apresentados quadros com o número de transações encontradas por aplicação, num total de 3.165 transações e 17 aplicações ($P_1 \dots P_{17}$).

Os resultados obtidos permitiram concluir que as transações T_1 , T_3 e T_{14} foram as mais frequentes, com percentagens de 30.4%, 20.6% e 11.8%, respectivamente. A seguir, será apresentado um gráfico comparativo acumulativo (figura V.3) da frequência das transações no ambiente da Eletrobrás. No eixo das abscissas, serão representadas as classes de transações, que são 18, e no eixo das ordenadas serão marcadas as frequências acumuladas, cuja maior ocorrência é 963 e corresponde a 30.4% do total.

DISTRIBUIÇÃO DE FREQUÊNCIA DAS TRANSAÇÕES TÍPICAS POR APLICAÇÃO

T/A	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
T1	23	103	2	50	80	54	91	9	18	20
T2	3	2	3	2	2	13	8	3	6	8
T3	10	44	4	83	20	45	141	33	81	21
T4	6	10	2	12	8	9	16	1	3	11
T5	2	9	4	0	7	1	3	3	0	2
T6	0	0	0	0	0	0	0	0	0	0
T7	0	0	0	0	0	0	0	0	0	0
T8	61	4	0	7	0	10	0	3	0	1
T9	2	2	0	5	2	2	0	3	0	1
T10	0	0	0	0	0	0	0	0	0	0
T11	21	8	0	0	20	3	11	37	0	4
T12	3	1	1	1	0	0	5	1	0	7
T13	1	3	1	0	20	3	1	5	0	1
T14	0	35	2	0	7	0	11	30	1	62
T15	3	11	0	13	0	1	16	0	1	2
T16	11	1	0	7	0	2	9	0	2	13
T17	7	1	2	0	3	0	11	0	0	1
T18	5	1	0	5	0	1	7	1	1	13
TOTAL	158	235	21	185	167	144	330	129	113	167

Figura V.1

DISTRIBUIÇÃO DE FREQUÊNCIA DAS TRANSAÇÕES TÍPICAS POR APLICAÇÃO (Cont.)

T/A	P11	P12	P13	P14	P15	P16	P17	%	TOTAL
T1	15	63	121	31	189	46	40	33,4	963
T2	5	15	12	5	6	25	9	4,0	127
T3	73	40	32	2	2	3	17	20,6	651
T4	5	12	20	1	13	12	6	4,6	147
T5	3	6	2	0	1	4	2	1,6	49
T6	0	0	0	0	0	0	0	0,0	0
T7	0	0	0	0	0	0	0	0,0	0
T8	0	10	4	5	82	13	0	6,3	200
T9	0	1	9	0	4	0	1	1,0	32
T10	0	0	0	0	0	0	0	0,0	0
T11	0	21	41	0	3	10	17	6,2	196
T12	3	11	6	1	3	13	2	1,8	58
T13	0	6	15	0	21	3	1	2,6	91
T14	0	1	20	0	184	17	2	11,8	372
T15	0	63	4	4	12	3	7	4,4	140
T16	0	3	1	3	0	5	1	1,8	58
T17	0	6	10	0	1	0	1	1,4	43
T18	0	6	0	3	0	4	1	1,5	48
TOTAL	104	264	297	55	521	159	115	100	3165

Figura V.2

DISTRIBUIÇÃO DE FREQUENCIA Transações na Eletrobrás

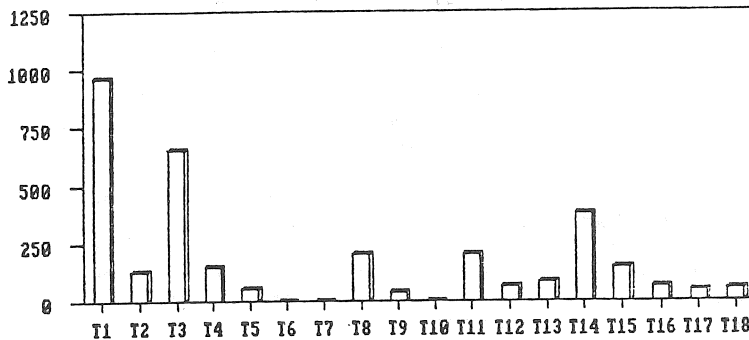


Figura V.3

Tornou-se necessário, então, definir uma aplicação para fazer parte do benchmark, que utilizasse todas as transações típicas definidas anteriormente, com distribuição de frequência igual a apresentada na amostra. A solução foi utilizar uma aplicação da Eletrobrás, pois esta estaria projetada e especificada no mesmo ambiente operacional e seria representativa da amostra obtida. Entretanto, suas aplicações são muito extensas e um benchmark requer uma aplicação pequena, simples e concisa que consiga realizar todos os testes de desempenho necessários a avaliação. Logo, decidiu-se trabalhar com uma subparte de uma aplicação que atendesse os objetivos e requisitos obrigatórios.

A aplicação escolhida foi a AEI (Aquisição de Equipamentos de Informática). Esta controla a aquisição de equipamentos de informática. A seguir, será apresentada a subparte do diagrama entidade-relacionamento (E-R) estritamente necessária a demonstração dos conceitos deste trabalho. Foram suprimidos os atributos e chaves para não sobrecarregá-lo.

DIAGRAMA ENTIDADE-RELACIONAMENTO DA APLICAÇÃO AEI

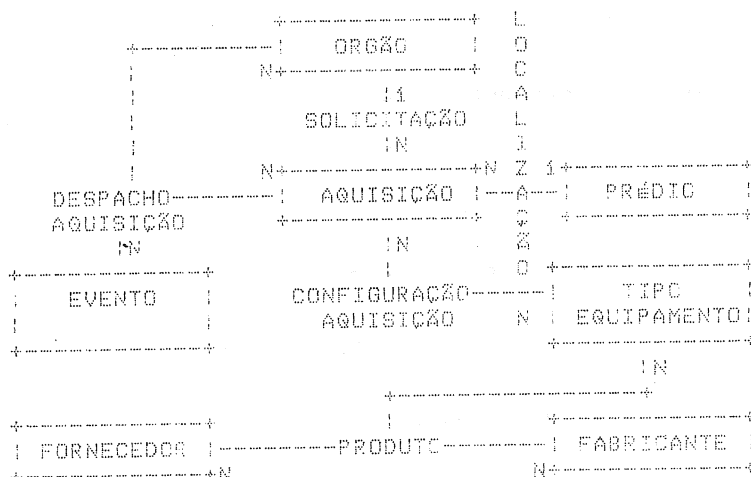


Figura V.4

As transações típicas T_i foram implementadas nas linguagens dos SGBD's em teste, utilizando as entidades da aplicação AEI. Cada T_i associada aos parâmetros de comparação fixados na metodologia, originou um benchmark de baixo nível, que foi executado nos SGBD's usando o módulo NI ("Norton Integrator") do "Norton Utilities" para medir o tempo imediatamente antes e depois da execução de cada evento a ser analisado.

Estes benchmarks foram executados juntamente com os SGBD's, em um microcomputador do tipo IBM-PC/AT operando na modalidade monousuária. As limitações no escopo do ambiente considerado visou isolar os efeitos causados pelas configurações de hardware, características do sistema operacional, softwares utilizados e fatores externos.

O processo de execução dos benchmarks foi repetido 5, 10, 100 e 1000 vezes, para confirmar os resultados e gerar parâmetros mais precisos e seguros. A execução de cada benchmark 1000 vezes verificou a estabilização dos parâmetros a medida que o número de execuções aumentou e, obteve um tempo de operação mais próximo do real. A cada mudança de SGBD e benchmark, foi zerado o buffer de memória para evitar a propagação de sujeiras e a influência de outros fatores no teste corrente.

A entrada dos dados da aplicação e dos comandos dos benchmarks foi feita em massa através da leitura de blocos de registros via arquivo externo, visando reduzir o tempo de improdutividade que seria gasto para selecionar opções em menus e, digitar e fornecer informações para os SGBD's via tela.

A cada execução de um benchmark em um SGBD, foram analisados os parâmetros dinâmicos fixados na metodologia. Entretanto, neste ambiente operacional deve-se fazer algumas considerações:

a. O **tempo de execução** coincidiu com o tempo de resposta, pois este ambiente é monousuário e não existe espera por recursos.

b. Durante a análise do tempo de execução, não foi possível distinguir o tempo de CPU do tempo de E/S, por não existir um procedimento para interceptar os acessos a disco e medir a E/S.

c. O **tempo de improdutividade** retratou o intervalo de tempo entre a chamada do SGBD para inicialização e carga do banco de dados, e o término da execução dos procedimentos necessários a construção do ambiente para teste dos benchmarks. Para tanto, foi executada uma consulta vazia, de forma a melhor retratar tempos improdutivos envolvidos na execução de uma transação.

d. Fixou-se o **tempo de operação** (Toper) como o intervalo de tempo necessário para processar uma transação T_i . Logo, Toper coincidiu com o tempo de execução de cada benchmark de baixo nível (descontado o tempo de improdutividade), pois cada benchmark representa uma transação típica T_i .

e. Fixou-se a **produtividade** como o número de transações T_i realizadas por segundo. Logo, esta foi calculada pela fórmula $1/Toper$ (1 dividido pelo tempo de operação).

f. Para a maioria das transações Ti, aumentando-se o número de execuções, o tempo total de execução e operação tendeu a aumentar uniformemente de t e o tempo médio tendeu a um valor constante e próximo do T anterior. Os tempos que serão apresentados nos quadros, são tempos médios.

g. Seja produtividade esperada, (PEsp) o produto da frequência pela produtividade de cada transação Ti em cada SGBD. Assim, definiu-se o parâmetro produtividade média esperada que é obtido pelo somatório do PEsp de cada transação típica Ti.

A execução dos benchmarks em cada SGBD gerou resultados de desempenho. A partir dos tempos de operação de cada transação Ti em cada SGBD gerou-se um quadro comparativo, mostrado a seguir, para analisar o desempenho de cada produto em Ti.

Os resultados obtidos permitiram concluir que:

i. Tia é a mais rápida na maioria dos SGBD's e T6 pode ser implementada, somente, no COPPEREL e ORACLE.

ii. O COPPEREL e o ORACLE são os únicos SGBD's com recursos que permitem implementar todas as operações relacionais. Não houve necessidade de simular as operações através da construção de procedimentos em linguagem de terceira geração. Os comandos disponíveis foram desenvolvidos com propósitos relacionais e, assim, foram suficientes para a implementação de todas as Ti.

QUADRO COMPARATIVO DO TEMPO DE OPERAÇÃO DE CADA TRANSAÇÃO TÍPICA EM CADA SGBD (em segundos)

Ti/SGBD	COPPEREL	DBASE	DIALOG	ORACLE	PARADOX	RBASE	ZIM
T1a	0,486	0,143	0,235	0,563	0,400	0,180	0,613
T1b	0,807	0,603	0,468	3,200	1,200	1,357	0,957
T2	0,878	0,532	0,487	2,400	-	1,592	-
T3	2,317	20,920	15,171	17,343	1,321	4,563	1,378
T4	0,745	4,026	5,672	1,995	-	3,387	1,378
T5	2,325	-	-	13,678	0,817	2,137	1,378
T6	2,985	-	-	0,380	-	-	-
T7	1,792	7,286	3,881	347,580	-	33,527	-
T8	1,593	1,593	1,600	7,727	-	2,138	2,006
T9	6,973	75,844	53,018	4,235	-	5,674	3,201
T10	0,801	0,368	0,266	0,451	-	1,158	1,433
T11	0,801	1,111	0,811	0,745	-	0,957	1,001
T12	1,371	0,610	0,600	0,917	-	0,563	-
T13	0,403	0,979	0,641	0,611	0,526	0,956	1,285
T14	6,981	75,830	53,018	1,454	-	5,676	1,823
T15	2,527	3,000	1,400	10,213	1,961	8,237	1,433
T16	7,912	133,611	71,999	0,577	-	7,273	2,375
T17	3,417	-	-	4,222	-	5,853	-
T18	4,678	3,251	2,999	14,743	-	5,861	-
Total	49,792	329,707	212,266	433,034	6,225	91,089	20,261

Figura V.5

iii. O PARADOX é o SGBD cuja linguagem possui menor poder relacional. Consequentemente, conseguiu-se implementar, somente, seis (6) transações em PAL.

iv. O COPPEREL é um SGBD que possui desempenho semelhante em todas as transações. Seu desempenho não ficou prejudicado em nenhuma delas. Por exemplo, na base de dados em teste, o tempo de operação foi inferior a 8 (oito) segundos.

v. O ORACLE teve seu tempo de operação prejudicado pela incorporação do tempo de improdutividade relativo a impressão dos resultados das consultas. Não desenvolveu-se procedimentos para interceptar as impressões e medir este tempo.

A partir da aplicação AEI, da distribuição de frequência das aplicações da Eletrobrás e dos benchmarks de baixo nível, foi definido e implementado um benchmark de alto nível T que englobou as transações T_i em um único programa. O benchmark T foi, também, executado várias vezes. Assim, analisou-se os resultados de desempenho de cada SGBD.

A partir destes resultados, construiu-se um quadro comparativo do tempo de operação médio de T em cada SGBD. A seguir, será apresentado este quadro com os valores de T_{oper} em segundos. Deve-se considerar que algumas transações não puderam ser implementadas em T devido a deficiências de alguns produtos, e geraram tempos de operação distorcidos e insignificantes.

QUADRO COMPARATIVO DO T_{OPER} DE T EM CADA SGBD (em segundos)

	COPPEREL	DBASE	DIALOG	ORACLE	PARADOX	SBASE	ZIM
T	1113,102	1073,735	831,773	393,983	57,549	206,636	81,878

Figura V.6

Os resultados obtidos permitiram construir um gráfico comparativo do desempenho de T em cada SGBD. A seguir, este será apresentado. No eixo das abscissas, serão representados os SGBD's, que são 7(sete), e no eixo das ordenadas será marcado o tempo de operação de cada SGBD com relação a T.

GRÁFICO DO TEMPO DE OPERAÇÃO DE T EM CADA SGBD (em segundos)

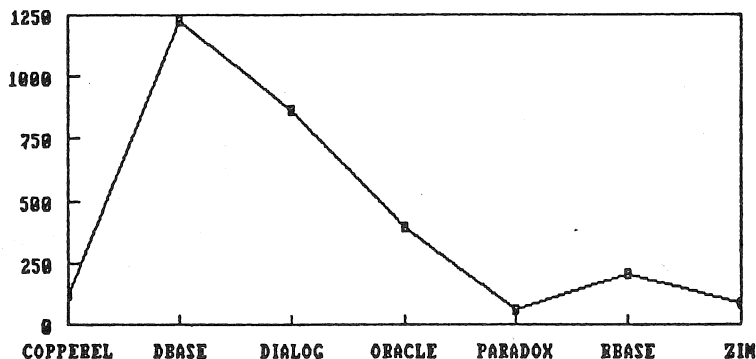


Figura V.7

A construção e implementação do benchmark de alto nível T ficou limitada pelas restrições impostas por alguns produtos. A falta de recursos de linguagem impediram a incorporação de algumas T_i em T. A seguir, será apresentado um quadro do número de transações implementadas e não implementadas em cada SGBD.

QUADRO COMPARATIVO DA IMPLEMENTAÇÃO DAS TRANSAÇÕES POR SGBD

SGBD	NO. TRANSAÇÕES IMPLEMENTADAS	NO. TRANSAÇÕES NÃO IMPLEMENTADAS	TRANSAÇÕES NÃO IMPLEMENTADAS
COPPEREL	19	00	---
DBASE	16	03	T5, T6, T17
DIALOG	16	03	T5, T6, T17
ORACLE	19	00	---
PARADOX	06	13	T2, T4, T6, T7, T8, T9, T10, T11, T12, T14, T16, T17, T18
RBASE	18	1	T6
ZIM	13	6	T2, T6, T7, T12, T17, T18

Figura V.8

Pelo fato do PARADOX permitir a implementação de apenas 6 transações, a execução do benchmark de alto nível neste SGBD apresentou o menor tempo de operação. Este fato comprova que, em alguns casos, um benchmark de alto nível pode gerar resultados de desempenho distorcidos. Neste caso, o produto com menor capacidade relacional apresentou-se mais eficiente. Consequentemente, estes resultados e o gráfico não são válidos para analisar o desempenho dos SGBD's.

Deste modo, conclui-se que a análise de desempenho destes SGBD's no ambiente em questão deve considerar, principalmente, os resultados gerados pelos benchmarks de baixo nível e ignorar aqueles gerados pelo de alto nível T. Pode-se comparar, analisar e validar os resultados de desempenho obtidos de T, somente, para o COPPEREL E ORACLE, pois estes produtos são os únicos capazes de implementar e executar todas as transações T_i, gerando resultados comparáveis e significativos.

Adicionalmente, decidiu-se construir um benchmark de alto nível T_{min} contendo o subconjunto máximo de T_i implementáveis em todos os SGBD's. Este é formado por 5 das 6 transações implementadas no PARADOX. Não foram utilizados recursos externos, mas somente recursos e linguagens disponíveis nos produtos, para implementar os benchmarks. Assim, o benchmark T_{min} incorporou apenas as transações T1a, T1b, T3, T13 e T15.

O benchmark T_{min} ficou limitado e não retratou com exatidão a aplicação AEI e a distribuição de frequência das aplicações na Eletrobrás. Entretanto, gerou resultados de desempenho comparáveis e significativos. A seguir, será apresentado um quadro comparativo do tempo médio de T_{min} em cada produto.

QUADRO COMPARATIVO DO TOPER DE T_{min} EM CADA SGBD (em segundos)

	COPPEREL	DBASE	DIALOG	ORACLE	PARADOX	RBASE	ZIM
T	53,243	326,576	216,239	321,531	53,427	112,999	53,666

Figura V.9

Estes resultados mostraram que o toper obtido por T_{min} foi bem próximo ao Toper obtido por T no PARADOX. Assim, comprovou-se que os números de T estavam distorcidos e insignificantes. A seguir, será apresentado um gráfico comparativo do desempenho de T_{min} nos SGBD's. No eixo das abscissas serão representados os SGBD's e no eixo das ordenadas os toper médios.

Deste modo, pode-se concluir que o COPPEREL é o SGBD com maior poder relacional e melhor desempenho nos parâmetros dinâmicos, apesar de possuir deficiências nos parâmetros estáticos.

GRÁFICO DO TEMPO DE OPERAÇÃO DE T_{min} EM CADA SGBD (em segundos)

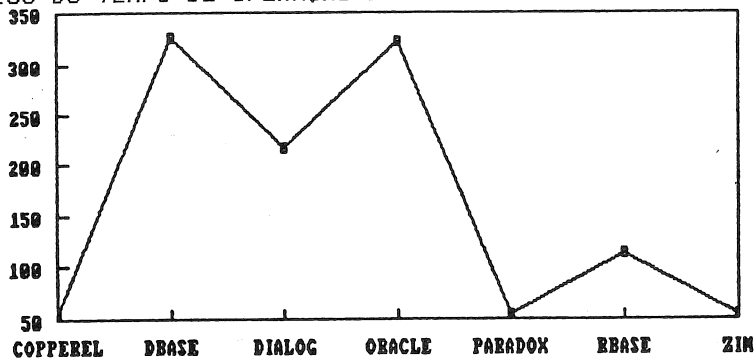


Figura V.10

VI. CONCLUSÃO

Não existe benchmark perfeito, que possa ser usado para avaliar o desempenho de todos os SGBD's em todas as instalações e não há receita para um benchmark bem sucedido. Não conseguiu-se desenvolver uma fórmula capaz de funcionar, perfeitamente, em todos os SGBD's, instalações e ambientes operacionais. Contudo, existem benchmarks suficientemente rigorosos para garantir algum grau de sucesso na maioria dos produtos analisados. Adicionalmente, deve-se mencionar que benchmarks permanecem como a única maneira prática de estimar o desempenho de SGBD's.

A implementação e execução dos benchmarks permitiu identificar características e deficiências de cada SGBD. Não há um que seja o "melhor" em todos os parâmetros estáticos e dinâmicos.

Finalmente, deve-se concluir que um SGBD gerado no curso de informática da U.F.R.J., sem fins comerciais, possui desempenho muito bom e funcionalidade relativamente boa. Não encontrou-se um produto que fosse muito superior ao COPPEREL. Portanto, após

algumas otimizações, este estaria em condições de competir com os melhores SGBD's relacionais no mercado de informática.

VII. FUTUROS TRABALHOS

Poderiam ser desenvolvidas novas estatísticas baseadas em amostras maiores de transações reais em uso em várias empresas. A nova amostra deveria conter um conjunto vasto de transações, seria representativa de ambientes reais de banco de dados e permitiria o desenvolvimento de benchmarks de alto nível mais completos. Estes poderiam ser utilizados para analisar o desempenho de grande parte dos SGBD's em vários ambientes operacionais com diversos tipos de aplicações. Para tanto, deveriam ser contactadas diversas empresas, de forma a analisar suas instalações e aplicações, observar as transações, montar a estatística e identificar transações típicas mais frequentes.

Os testes de desempenho poderiam ser extendidos para ambientes multiusuário. O benchmark deveria ser adaptado para considerar e analisar a influência dos fatores externos. Provavelmente, novos parâmetros dinâmicos deveriam ser definidos. Os testes precisariam ser regenerados e os resultados reanalisados.

A metodologia ("benchmark") desenvolvida para avaliação do desempenho de SGBD's foi testada em 7 produtos. Contudo, esta poderia ser usada para avaliar outros SGBD's. Logo, deveria-se implementá-la utilizando a linguagem do produto em teste.

VIII. REFERÊNCIAS BIBLIOGRÁFICAS

- [Bit83] Bitton, Dina & DeWitt, David J. & Turbyfill, Carolyn, "Benchmarking Database Systems: A Systematic Approach", Proceedings of the 1983 Very Large Database Conference, Florence, Italy, 1983.
- [Bora84] Boral, Haran & DeWitt, David J., "A Methodology for Database System Performance Evaluation", Proceedings of the 1984 ACM Sigmod International Conference on the Management of Data, Boston, June 18 - 21, 1984.
- [DeWi85] DeWitt, David J., "Benchmarking Database Systems: Past efforts and Future Directions", Database Engineering, V0, N1, March 1985.
- [Ston85] Stonebraker, Michael, "Tips on Benchmarking Data Base Systems", Database Engineering, V8, N1, March 1985.
- [Ross91] Rossatto, Maria Antonieta Rossatto, "BENCHREL - Benchmark para a Avaliação de Desempenho de SGBD's Relacionais - e Propostas de Melhorias para o COPPEREL", Tese de Mestrado em Engenharia de Sistemas e Computação, COPPE/U.F.R.J., Rio de Janeiro, 1991.
- [Turb87] Turbyfill, Carolyn, "Comparative Benchmarking of Relational Database Systems", Ph. D. Thesis, Number 87 - 871, Department of Computer Science, Cornell University, Ithaca, New York, September 1987.