

Universidade do Minho
Licenciatura em Engenharia Informática
Laboratórios de Informática III



Analise de tempos de execução

Ano Lectivo de 2009/2010

GRUPO 9
54738 **João Gomes**
54745 **André Pimenta**
54825 **Daniel Santos**

16 de Maio de 2010

Resumo

Este relatório serve essencialmente para apresentar os resultados obtidos nas medições de tempos de operações de inserção e consulta da base de dados.

Nele vamos apresentar os resultados das medições do tempo de carregar a base de dados de clientes a partir de um ficheiro, inserir o registo de um novo cliente, procurar um cliente por nome e número de contribuinte , percorrer a estrutura e imprimir os dados dos utilizadores, inserir uma mensagem recebida de um amigo.

Para medir os tempos das diferentes operações usamos essencialmente a função `gettimeofday`.

Todos os dados que aqui vamos reportar foram obtidos através de uma máquina que mais à frente iremos descrever.

Conteúdo

Conteúdo	1
0.1 Motivação e objectivos	1
0.2 Estrutura	1
1 Desenvolvimento	2
1.1 Análise do Problema	2
1.2 Informações da máquina	3
1.3 Metodologia experimental	3
1.4 Tempos de execução	4
1.4.1 Tempo de inserção de um utilizador	4
1.4.2 Tempo de carregar bases de dados	5
1.4.3 Tempo de procura de utilizador por Nome e Nif	6
1.4.4 Tempo de percorrer e imprimir toda a estrutura	8
1.4.5 Tempo de inserir uma mensagem enviada por um amigo	8
2 Conclusão	10
3 Fotos	11

0.1 Motivação e objectivos

Podemos afirmar que foi bastante motivante verificar o quanto era eficaz ou não o projecto por nós desenvolvido.

Sendo que assim conseguimos ter uma ideia mais completa do trabalho que por nós foi desenvolvido, ao conhecer as suas capacidades de execução em diferentes contextos.

0.2 Estrutura

O presente relatório é constituído por três capítulos: **Introdução**; **Desenvolvimento**, dividido em duas secções, **Metodologia experimental**, **Tempos de execução** e, por fim, a **Conclusão**.

Capítulo 1

Desenvolvimento

1.1 Análise do Problema

Para a captura dos vários tempos de execução nos diversos cenários é preciso destacar as condições onde estas são feitas e de que forma são feitas.

É por tanto nítido o destaque que se tem de dar as condições da execução do programa para que sejam justas e iguais para os testes possam ser comparados e daí tirar conclusões.

Para realizar esta tarefa decidimos então usar algumas tarefas de medição de tempos de execução disponíveis pelos sistemas unix, ajustar os detalhes da maquina usada para testes.

1.2 Informações da máquina

Processador Intel Pentium Dual Core T2390 1.86Ghz - 1Mb cache L2 533Mhz
Disco Rígido - 250Gbs (5400RPM) SATA-II
Memória RAM - 2Gb DDR2 - 667Mhz
Sistema Operativo - Mint Elena 8

1.3 Metodologia experimental

Para a medição dos tempos era necessário usar mecanismos que permitissem medir o tempo com algum rigor como foi comunicado na palestra, e sendo a máquina acima descrita a usada nos testes verificamos que a função “gettimeofday” do unix nos permitia em muitos dos casos obter valores próximos do esperado. Esta é uma função torna-se especialmente útil pois não só nos devolve os resultados reportados em microsegundos mas também em segundos, que é a escala de leitura da base de dados de ficheiro.

Detalhes:

- Para cada experiencia foram realizados X medições.
- Cada medição foi registada com base na media de X operações.
- Para cada experiencia é reportada a média.
- Nos casos em que existe discrepância entre valores esses valores são apresentados assim como a devida justificação de tais medições.
- Os tempos são reportados em microsegundos.
- As medições são baseadas num processador com resolução de 1/1.86GHz (0.53ns).
- Medições efectuadas para base de dados com 5000, 10000, 15000 e 18000 registos .

1.4 Tempos de execução

1.4.1 Tempo de inserção de um utilizador

A inserção isersao de um utilizador na base de dados depende essencialmente de tres factores: o tempo de copiar os dados para a estrutura “Perfil”, o tempo necessário para calcular o código de hash para o nome e para o nif e percorrer uma posicao da tabela de hash do nif para verificar se o utilizador já está registado.

A utilização de tabelas hash permite tempos de acesso e inserção próximos de tempo constante no entanto à medida que o número de utilizadores vai aumentando começam a surgir colisões que por sua vez começam a afectar o tempo de inserção como veremos.

Através dos tempos observados conseguimos apurar os seguintes resultados:

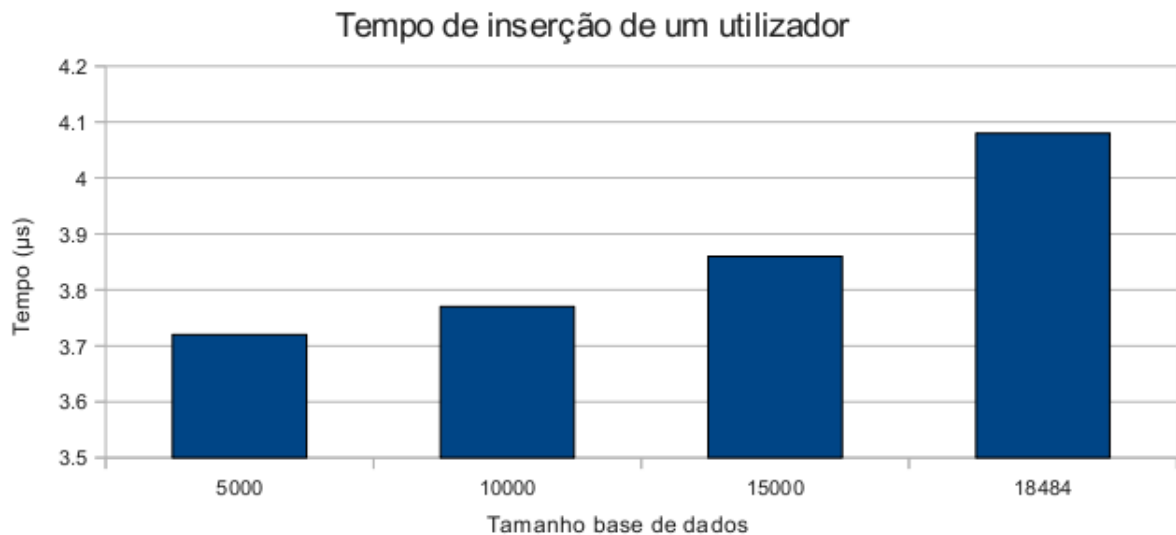


Figura 1.1: Grafico do tempo de inserção de um utilizador

<i>Tempo de Inserção de um Utilizador</i>				
Tamanho	5000	10000	150000	18484
Tempo (µs)	3.72	3.77	3.86	4.08
Máximo (µs)	88	204	97	262
Mínimo (µs)	3	3	3	3

Figura 1.2: Tabela do tempo de inserção de um utilizador

Como podemos verificar através dos resultados, em todas as amostras vemos que existem outliers.

Estes valores podem ser explicados pela distribuição que as funções de hash fazem. Em alguns casos o mesmo código hash é atribuído a mais do que um utilizador o que faz com que o programa tenha de percorrer a lista de users para não permitir que existam dois utilizadores repetidos como já explicamos acima.

Podemos observar que à medida que a base de dados aumenta o tempo de inserção aumenta também ligeiramente. Isto é explicado com o facto de o tamanho da tabela de hash permanecer constante mas o número de utilizadores ir aumentando o que faz com que cada vez ocorram mais colisões o que provoca mais perda de tempo.

1.4.2 Tempo de carregar bases de dados

A leitura de uma base de dados de ficheiro é um processo pode ser demoroso pois trata-se da leitura e inserção milhares de utilizadores nas tabelas de hash e como tal é necessário que essa leitura seja o mais eficiente possível.

É natural que à medida que a base de dados vai crescendo e principalmente se estivermos a utilizar um ficheiro muito grande não conseguimos ter todo na memória cache. Isto vai implicar perdas de tempo significativas visto que aceder à memória RAM tem um custo muito superior, em termos de tempo dispendido, em relação ao acesso à memória cache.

Sendo assim a principal condicionante a uma rápida leitura será a velocidade a que conseguimos aceder aos dados em memória.

Os tempos que obtivemos na leitura das bases de dados com diferentes tamanhos:

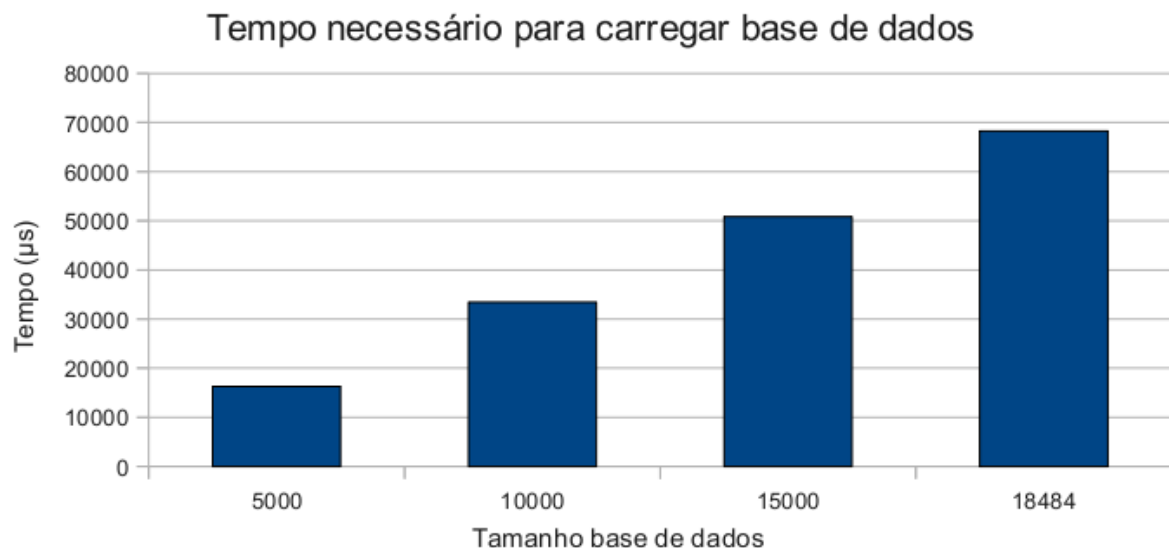


Figura 1.3: Grafico do tempo necessário para carregar base de dados

Podemos observar através dos resultados que o tempo dispendido aumenta linearmente com o aumento da base de dados. Isto deve-se ao facto de esta ser um comando que envolve muitas operações de I/O.

Estando as operações de I/O entre as que consomem mais tempo podemos antever que aqui as colisões não terão um papel tão fundamental em relação ao tempo total gasto como tinham na inserção de um utilizador.

1.4.3 Tempo de procura de utilizador por Nome e Nif

Procurar um utilizador por nome e por nif é uma operação bastante simples. Consiste em calcular o código hash do nome/nif introduzido e percorrer a lista ligada que se encontra na posição do código calculado.

Esta é uma operação que tem no entanto algumas condicionantes. Se existirem duas pessoas com o mesmo nome, o tempo na depende só da máquina depende também do utilizador pois ele (utilizador) o utilizador que pretendemos pode não ser o primeiro da lista e então teremos de ir percorrendo a lista. Sendo que esta decisão de escolher se é ou não o utilizador que de facto pretendíamos ser uma operação de I/O será impossível fazer qualquer análise neste caso pois estes tempos ficarão sempre dependentes da velocidade a que o utilizador responde.

Sendo assim os dados aqui apresentados serão apenas referentes a utilizadores com nome unico na base de dados.

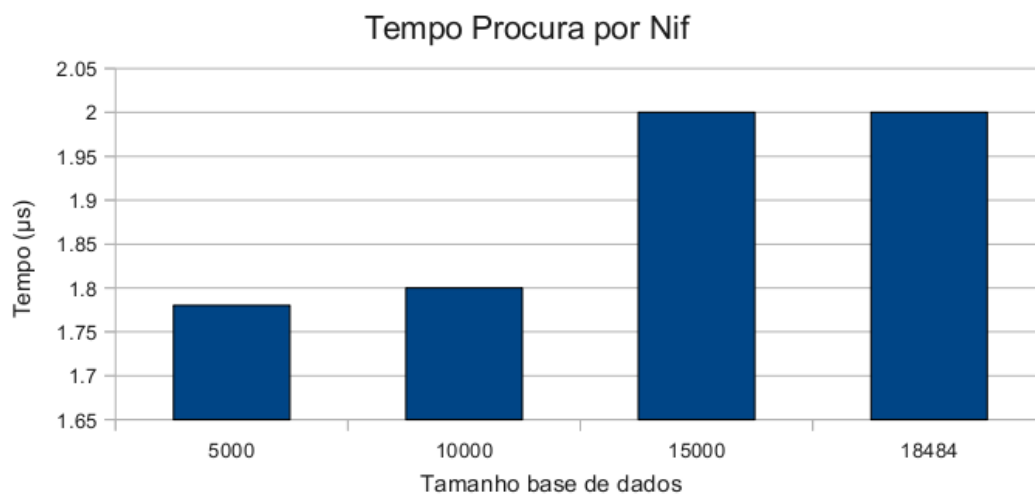


Figura 1.4: Grafico do tempo procura por Nif

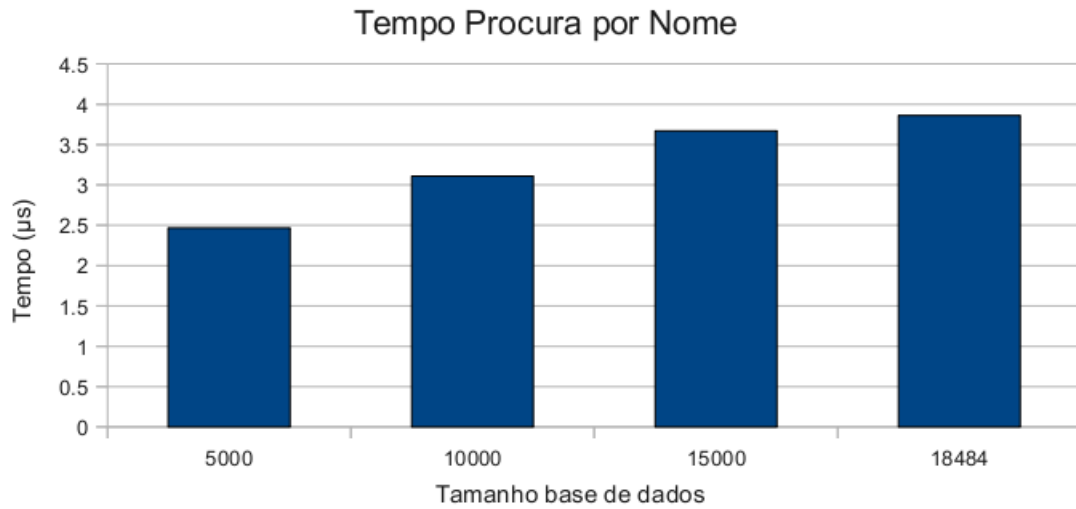


Figura 1.5: Grafico do tempo procura por Nome

O facto de a pesquisa para o nome ser mais lenta do que a pesquisa para o nif deve-se ao facto de a função que calcula os códigos de hash para o nif ser mais eficiente do que a função que calcula os códigos de hash para o nome. Posto isto é normal que a procura por nome demore mais tempo visto que em principio existirão mais colisões.

1.4.4 Tempo de percorrer e imprimir toda a estrutura

O tempo de percorrer toda a estrutura será o tempo de percorrer cada posição da tabela de hash e imprimir todos os utilizadores.

Está será uma tarefa que levará um pouco de tempo a executar pois para cada utilizador temos de fazer operação da I/O.

Os tempos observados para as bases de dados com diferentes tamanhos foram as seguintes:

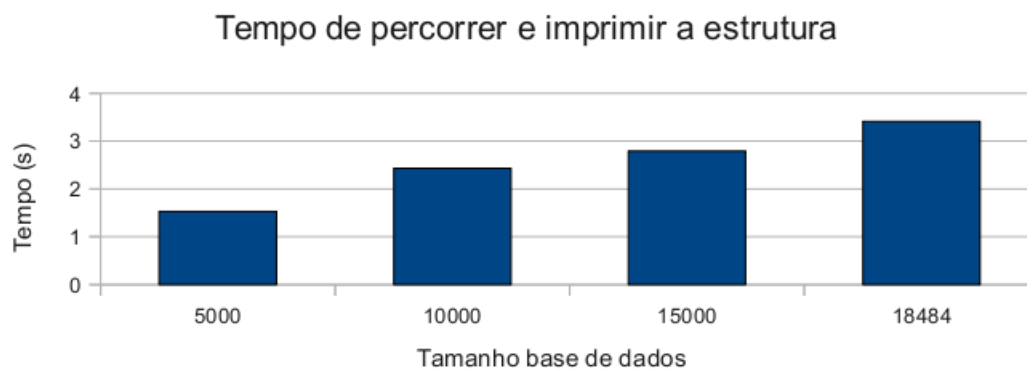


Figura 1.6: Grafico do tempo de percorrer e imprimir a estrutura

<i>Tempo de percorrer e imprimir estrutura</i>				
Tamanho	5000	10000	15000	18484
Tempo (s)	1.53	2.43	2.78	3.41

Figura 1.7: Tabela do tempo de percorrer e imprimir a estrutura

1.4.5 Tempo de inserir uma mensagem enviada por um amigo

Esta é sem dúvida a operação que traz maior desequilíbrio entre os valores registados nas diferentes observações.

Como cada utilizador pode receber 20 mensagens de cada utilizador e a base de dados foi construída para cerca de 10000 utilizadores é fácil prever que o número de mensagens que podem existir seja um número muito grande. Sendo o suporte para mensagens uma condição do

projecto e sendo que seria incomportável ter um número tão grande de mensagens em memória decidimos que trabalhar com as mensagens através de ficheiro.

Cada utilizador que pretenda enviar ou então que receba uma mensagem fará com que o programa verifique se as suas mensagens já se encontram em memória. Se o utilizador ainda não tem as suas mensagens em memória então o programa será responsável por verificar se o utilizador possui mensagens em ficheiro e caso verifique que existem carrega-las para memória.

Outra condição que influencia o tempo de inserção das mensagens é o facto de cada utilizador só poder receber 20 mensagens de cada utilizador. Assim quando o utilizador receber a vigésima mensagem, e daí em diante, teremos de percorrer toda a lista de mensagens para apagar a mensagem mais antiga, o que fará com que seja dispendido mais algum tempo na operação de inserção.

Tempo necessário para enviar uma mensagem

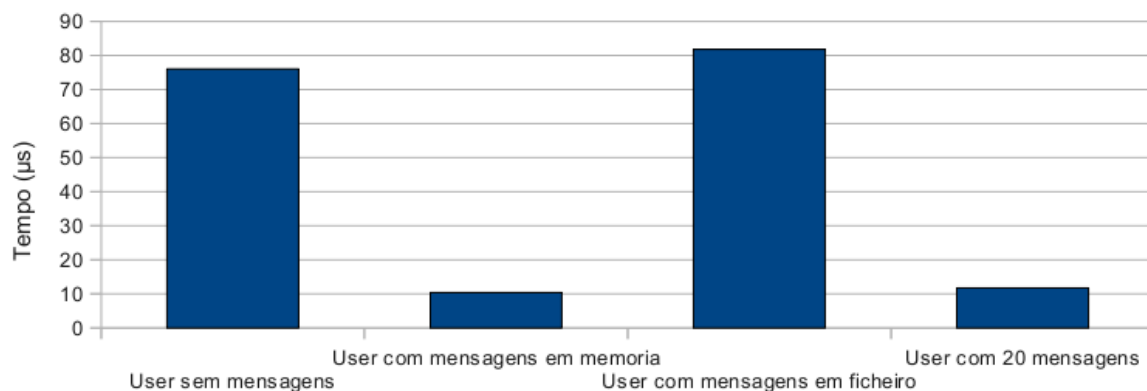


Figura 1.8: Grafico do tempo necessário para enviar uma mensagem

<i>Tempo necessário para enviar mensagem</i>				
	User sem mensagens	User com mensagens em memoria	User com mensagens em ficheiro	User com 20 mensagens
Tempo (µs)	76	10.4	81.8	11.75

Figura 1.9: Tabela do tempo necessário para enviar uma mensagem

Capítulo 2

Conclusão

Através deste trabalho e depois de medir os tempos das diferentes operações pudemos observar mais detalhadamente as implicações que traz uma deitura de dados de ficheiro, ou uma utilização de tabelas de hash ou ainda a utilização de listas ligadas entre outros mecanismos utilizados para guardar as informações dos utilizadores da nossa rede social.

Este trabalho veio por isso dar-nos mais alguma experiencia e novos conhecimentos, nomeadamente em relação a algumas escolhas que fizemos e que prejudicaram o desempenho do nosso programa, e outras que podemos considerar que foram boas escolhas e que levaram a que o programa tivesse um bom desempenho em outros aspectos.

Ficamos por isso satisfeitos com o trabalho realizado.

Capítulo 3

Fotos



Figura 3.1: João Miguel



Figura 3.2: André Pimenta

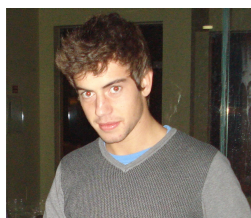


Figura 3.3: Daniel Santos