



# Mini-Projeto 1

## Análise de Desempenho

**Andrus Sandres – 254231**  
**Hugo Poloni – 254150**  
**Luiz Carlos Irber Jr. – 254258**  
**Mário Clemêncio da Silva Jr. – 254355**

**03/04/2009**

### Ambiente utilizado:

Computador: Laptop Lenovo T61 Core 2 Duo T8300 @ 2.40 GHz, 2,00 GB de RAM DDR2 667 MHz

SO: Ubuntu Linux 8.10 64bits com kernel otimizado para a CPU utilizada

Sistema de arquivos: Ext3

- **Teste: “stress --cpu”**

- **CPU:** A utilização se mantém em 100% quando o número de threads é superior a 2 e em torno de 50% quando o thread é igual a 1. Isso ocorre devido à CPU utilizada possuir dois núcleos.
- **I/O:** O teste não faz referência à I/O, sendo assim o gráfico referente apresenta apenas leituras e escritas de processos descorrelatados.
- **Memória:** A quantidade de memória utilizada se mantém estável, pois da mesma forma que o I/O, o teste não faz referência à alocação de memória, além disso, o computador está somente executando os testes de stress.
- **Interrupções:** Apresentaram pouca variação podendo ser considerada estável.
- **Troca de contexto:** Pouca quantidade devido ao baixo número de processos e threads. Mesmo utilizando 10 threads que foi o caso mais extremo avaliado, o número de trocas de contexto foi baixo.

- **Teste: “stress --hdd”**

- **CPU:** A maior parte de sua utilização deve-se à espera de I/O, pois no teste não há carga no sistema. No primeiro teste apresentou picos, pois o dispositivo de armazenamento conseguiu lidar melhor com as requisições, a partir do segundo teste o dispositivo está saturado, sendo assim a espera é quase constante.
- **I/O:** Observa-se bursts no disco que se referem também à alocação e liberação de memória e também a variação da taxa de dados enviados ao dispositivo se manteve a mesma independente do parâmetro passado para o hdd denotando assim a saturação do disco já no primeiro teste.
- **Memória:** Com o aumento de requisições de I/O observa-se a tendência de estabilizar a quantidade de memória alocada nas requisições, até que a velocidade limite do disco seja alcançada e a alocação de memória fique estável.
- **Interrupções:** Houve pouca variação devido o disco estar saturado, notando-se uma forte tendência à estabilização no número de interrupções.
- **Troca de contexto:** Mesmo sendo difícil observar variações no gráfico, é sabido que o Ubuntu utiliza o scheduler de I/O CFQ que aumenta as fatias de tempo com o intuito de aumentar a vazão dos dados.

- **Teste: “stress --vm”**

- **CPU:** Com somente uma thread sendo executada, a utilização de CPU gira em torno de 50%, dando a entender que a alocação e liberação de memória estão

utilizando somente um núcleo, nos outros testes a utilização observada é de 100%.

- **I/O:** Os dados aferidos não se mostraram relevantes pois as taxas são pequenas com alguns picos e não são devidas ao teste em questão, uma vez que este não realiza I/O.
- **Memória:** A execução de malloc() e free() não pode ser observada devido a rapidez das operações (que acontecem sequencialmente) ser relativamente grande quando comparada à velocidade de aferição dos dados.
- **Interrupções:** Apresentaram pouca variação podendo ser considerada estável da mesma forma que testes anteriores.
- **Troca de contexto:** Analogamente ao teste anterior, a troca de contexto não foi expressiva.

- **Teste: “stress --cpu --hdd”**

- **CPU:** Da mesma forma que o stress de CPU com apenas uma thread, esta utiliza somente um núcleo fazendo com que o processamento da CPU como um todo gire em torno de 50%, enquanto o outro núcleo está a maior parte do tempo sem carga o que denota a espera de I/O apresentada no gráfico.
- **I/O:** Idem ao teste “stress --hdd”.
- **Memória:** Seguiu o padrão dos testes anteriores, quanto maior o número de requisições de disco, mais memória é alocada e maior é o tempo que demora a ser liberada. Isso ocorre devido ao disco não conseguir dar vazão. A memória é estabilizada quando a vazão do disco é igual à alocação de memória relativa à I/O, lembrando que não fará diferença alocar mais memória quando o disco já está saturado.
- **Interrupções:** Mais testes são necessários para retirar uma conclusão precisa sobre este dado.
- **Troca de contexto:** Seguiu o padrão observado anteriormente, Em teoria, quanto maior o número de processos e/ou threads maior a quantidade de trocas de contexto, porém verifica-se como anteriormente um trade-off promovido pelo CFQ a medida que as operações de IO aumentam. Nota-se claramente a enorme diferença de tempo decorrido a medida que as transações são processadas na CPU e o tempo que elas levam em IO.

## Bônus

### Ambiente utilizado:

Computador: Desktop Pentium 4 HT 3.2 @ 3.6, 1,00 GB de RAM DDR 400 MHz

SO: Ubuntu Linux 8.10 32bits com kernel não otimizado para a CPU utilizada

Sistema de arquivos: Ext3

As diferenças dos dados aferidos nestes testes em comparação aos testes anteriores são:

- **CPU:** No teste com somente uma thread a CPU também se manteve em 50% mesmo possuindo apenas um núcleo, isso ocorre devido à tecnologia Hyper-Threading da Intel, nos outros testes a CPU se comportou de forma semelhante à outra.
- **I/O:** A vazão do disco do desktop é superior devido principalmente a velocidade de rotação do mesmo, mas o comportamento é semelhante ao do laptop.
- **Memória:** Houve uma mudança no modo de utilização da memória pelo segundo computador, uma vez que a quantidade de memória física disponível era menor. Nota-se a maior utilização da memória virtual e de um padrão recorrente a variação de quantidade da memória "idle" sugerida pelo início e término de jobs entregues ao processador.
- **Troca de contexto:** Dados interessantes pois mostram como a vazão do desktop possibilita uma menor troca de contextos em função da taxa de interrupções. O processador mesmo sendo de 2 gerações anteriores e apresentando apenas 1 núcleo físico pode ser aproveitado de maneira mais eficaz que o processador do laptop, gerando um número maior de interrupções.