

## Aumente a taxa de transferência de dados de seus discos IDE com o `hdparm`



# Pé na tábuca!

O desempenho de discos rígidos em sistemas Linux recém-instalados fica normalmente aquém das possibilidades oferecidas pelo hardware. O `hdparm` solta o “freio-de-mão” puxado pelas distribuições na instalação padrão.

POR THOMAS WÖLFER

A baixa velocidade na transferência de dados - mesmo com os mais modernos discos rígidos disponíveis no mercado - é o preço que atualmente se paga por uma instalação padrão de distribuições Linux como o Debian, SuSE e outros, já que elas são extremamente conservadoras na configuração de hardware, de olho na compatibilidade máxima durante o processo de instalação. O utilitário `hdparm` solta esse “freio de mão” artificial de modo simples e eficiente. Este artigo vai lhe mostrar como utilizar, em doses homeopáticas e de forma segura, o `hdparm` para tirar o máximo que seu disco rígido é capaz de oferecer. Dependendo do hardware, isso pode significar um ganho de desempenho até dez vezes maior.

### O mapa da mina

Quase todos os discos rígidos atualmente no mercado oferecem suporte a diversos modos de operação, como PIO (*Program-*

*med Input and Output - Entrada e Saída Programada*) e UltraDMA (*Direct Memory Access - Acesso Direto à Memória*), entre outras tantas funções e características especiais; nenhuma delas é, entretanto, considerada pelas rotinas de instalação padrão de várias distros. O programa `hdparm` permite que modifiquemos a configuração do disco rígido, possibilitando que obtenhamos dele o máximo em performance.

Mas cuidado: otimizar o desempenho do HD pode ser um pouco arriscado. É imprescindível fazer um backup dos dados em disco antes de começar a realizar experimentos. Além disso, recomenda-se realizar uma fase de testes antes de proceder à configuração propriamente dita. Para isso, inicialize o computador em “Single User Mode”.

Utilizamos em nossos testes um AMD AthlonXP 2600+ com 256 MBytes de RAM e um disco rígido EIDE. Segundo as especificações do fabricante, o disco

rígido utilizado oferece uma taxa de transferência de dados máxima de 66 MByte/s. Para saber qual é a taxa de transferência atual do disco rígido, utiliza-se as opções `-T` e `-t`. O utilitário `df` pode ser usado para mostrar quais unidades de disco estão instaladas em seu sistema. Use as opções `-T` e `-t` sempre juntas. `hdparm` mede a

### Mostrando as possíveis configurações

Testes repetidos com o sistema exemplo mostraram que o `hdparm` calculou valores quase idênticos para as taxas de transferência de dados, com uma pequena variação. A surpresa foi que ao invés dos 65 Mbyte/s de desempenho que deveriam ser atingidos pelo disco rígido, de acordo com o fabricante, obtivemos apenas 3,9 MByte/s. Ou seja, sem nenhuma otimização, o desempenho do disco rígido é aproximadamente 17 vezes menor do que aquilo que ele realmente pode oferecer.

Para otimizar a performance do disco rígido, vamos primeiramente verificar qual é a sua configuração atual. Para tanto, basta executar o `hdparm` sem opções, indicando apenas o dispositivo a ser analisado (Figura 2). A maioria das opções está desativada e, além disso, o kernel trata as controladoras de disco como dispositivos de entrada e saída de 16 bits (“I/O Support”). Não é de se admirar que em uma configuração como essa a taxa de transferência não passe de

meros 4 MByte/s.

```

jrtstl:~ # hdparm -Tt /dev/hda

/dev/hda:
Timing buffer-cache reads: 1164 MB in 2.00 seconds = 581.42 MB/sec
Timing buffered disk reads: 12 MB in 3.21 seconds = 3.73 MB/sec
jrtstl:~ #
  
```

Figura 1: Desempenho miserável, pois a configuração do disco rígido não está otimizada.

```

jrtstl:~ # hdparm /dev/hda

/dev/hda:
multcount      = 0 (off)
IO_support     = 0 (default 16-bit)
unmaskirq     = 0 (off)
using_dma     = 0 (off)
keepsettings  = 0 (off)
readonly      = 0 (off)
readahead     = 8 (on)
geometry      = 7473/255/63, sectors = 120060864, start = 0
jrtstl:~ #
  
```

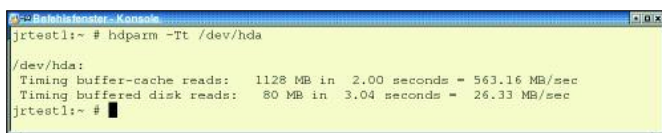
Figura 2: Executar o `hdparm` sem opções fornece uma lista que, além dos parâmetros físicos (“geometry”) do disco, mostra também se ele se encontra em modo DMA.

## Dobrando o desempenho com valores seguros

Todo mundo quer melhorar o desempenho do sistema. Não é possível, entretanto, dominar todas as opções do *hdparm* logo de início. Desse modo, vamos primeiro dar uma boa olhada nas possibilidades que o comando oferece. A palavra *multcount* é uma abreviação para “Multiple Sector Count”, que especifica quantos setores são lidos em uma única interrupção de I/O. A maioria dos discos rígidos oferecem suporte para 2, 4, 8 ou 16 setores. É possível que as cabeças de leitura dos discos leiam até mais que 16 setores. A quantidade de setores que um disco rígido pode ler pode ser verificada com a opção *-i* (*Max-MultSect*). Entretanto, não se deve confiar cegamente nos valores fornecidos pelo *hdparm*: alguns discos simplesmente indicam poder suportar a leitura de um número de setores que, na prática, supera a sua capacidade real. O resultado é a perda certa de dados! Em alguns casos mais graves até o sistema de arquivos é corrompido - o manual do *hdparm*, inclusive, traz um alerta a esse respeito. O ganho de desempenho tende a ser muito grande, mas o risco de perda de dados é diretamente proporcional a ele. A melhor proteção é fazer primeiro um backup do sistema e somente depois realizar testes para se chegar ao valor ideal de *multcount*, que pode ser fixado com a opção *-m*. Por fim, a utilização de suporte a I/O indica ao *hdparm* como os dados que vêm do “bus” PCI são manipulados pela controladora de disco. Controladoras modernas não precisam rodar em modo 16 bits. O suporte a I/O pode ser configurado com a opção *-c*, conforme segue:

```
hdparm -c1 -m16 /dev/hda
```

Com esse comando o suporte a I/O é configurado em modo 1, que é um dentre vários modos de 32 bits. Todos os modos estão listados no manual do comando sob a opção *-c*. A opção *-m16* configura o sistema para leitura de 16



```

jrttest1:~ # hdparm -Tt /dev/hda
/dev/hda:
Timing buffer-cache reads: 1128 MB in 2.00 seconds = 563.16 MB/sec
Timing buffered disk reads: 80 MB in 3.04 seconds = 26.33 MB/sec
jrttest1:~ #
  
```

Acima de tudo o uso do modo UltraDMA acelera em muito a transferência de dados.

setores por interrupção de I/O, o que leva a resultados muito melhores na próxima bateria de testes

Já conseguimos praticamente dobrar a taxa de transferência de dados do disco rígido: de 3,9 MByte/s chegamos a 7,6 MByte/s. No entanto, ainda estamos bem longe da performance prometida pelo fabricante, que é de 66 MByte/s. Vamos ver o que é possível fazer para conseguir melhorar este cenário.

## Quadruplicando a taxa de transferência com DMA

Como próxima medida de otimização vamos permitir o não-mascaramento de interrupções (*umaskirq*). Além disso, vamos ativar o modo de acesso direto à memória (DMA: *using\_dma*). Normalmente o Linux entra em modo “IDLE” enquanto espera por uma interrupção - ou seja, ele não faz nada. É óbvio que com isso ele desperdiça tempo de transferência de dados, caso ele não esteja ocupado com outras tarefas. E via de

regra esse comportamento faz sentido, visto que isso contribui para aumentar a estabilidade do sistema. Contudo, se desativarmos esta proteção, aumentamos a responsividade do sistema. O problema é que, na maioria dos casos, ele se torna tão instável que nem dá mais para usar. De qualquer modo, realizar alguns experimentos com o não-mascaramento de interrupções não envolve riscos. O mesmo vale para o uso do modo DMA: temos opções para “brincar” à vontade. Mas o sistema tende a travar quando insistimos em tentar ativar as opções erradas de acesso direto à memória. Felizmente, basta reiniciar o computador para que tudo volte ao normal, de modo que podemos realizar os testes sem medo de seqüelas ou perda de dados.

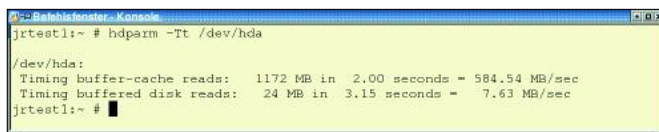
Caso a documentação do disco rígido ou da controladora estejam à mão, use os modos de operação de acesso direto à memória indicados nela. Sem a documentação, será necessário realizar experimentos com o sistema (e

testá-lo) até que a configuração ideal seja encontrada. Tivemos que nos esforçar bastante até encontrar a configuração ideal de nosso sistema de testes, mostrada a seguir:

```
hdparm -d1 -X66 -u1 -m16 -cd 2 /dev/hda
```

A opção *-d1* ativa o “flag” de utilização de acesso direto à memória (*using\_dma*). Além disso, também é necessário indicar qual modo de acesso deve ser utilizado. Em nosso exemplo usamos UltraDMA modo 2, configurado através da opção *-X66*. A opção *-u1* desativa o mascaramento de outras interrupções (*umaskirq*). Os resultados alcançados são surpreendentes.

A taxa de transferência de dados quintuplica com o uso dos novos parâmetros, se a compararmos com a configuração padrão. Entretanto, a nossa nova configuração não irá resistir à próxima reinicialização do sistema.



```

jrttest1:~ # hdparm -Tt /dev/hda
/dev/hda:
Timing buffer-cache reads: 1172 MB in 2.00 seconds = 584.54 MB/sec
Timing buffered disk reads: 24 MB in 3.15 seconds = 7.63 MB/sec
jrttest1:~ #
  
```

Figura 3: Basta configurar a controladora para realizar leituras de 32 setores por interrupção de I/O, ao invés de 16, para dobrar a taxa de transferência de dados.

## Salvando a nova configuração

Como mencionado anteriormente, a configuração só permanece válida até a próxima reinicialização do sistema. Para salvá-la, precisamos adicionar o comando *hdparm* mostrado anteriormente a um dos scripts de inicialização do sistema, *rc\**. Entretanto, apenas isso não basta: em determinadas circunstâncias, a controladora de disco realiza um chamado “IDE reset”. Quando isso ocorre, perde-se a configuração atual. A opção *-k1* evita que tal reset seja efetuado. Mas ela só deve ser usada após nos certificarmos de que todas as opções passadas para o *hdparm* funcionam sem problemas - ou seja, quando podemos ler e escrever dados do disco sem que nenhum alerta (“warning”) apareça no arquivo */var/log/messages*.

## INFORMAÇÕES

[1] Homepage do *hdparm*: <http://www.ibiblio.org/pub/Linux/system/hardware/>