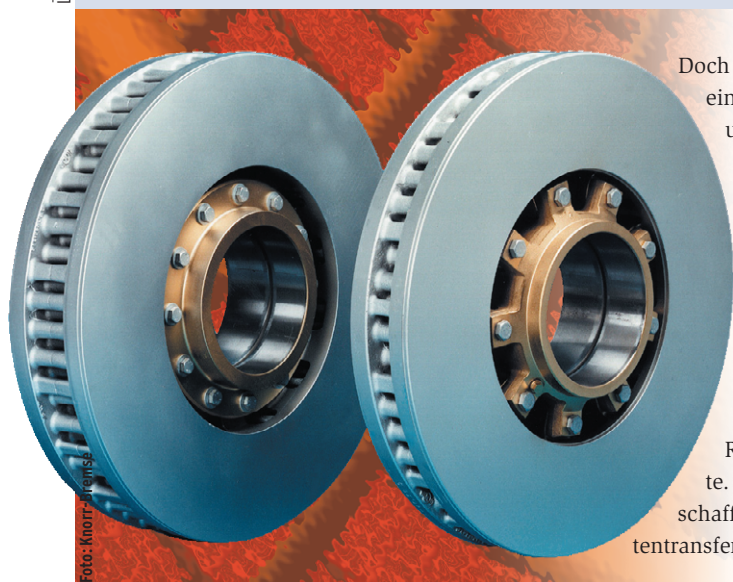


Workshop: Datendurchsatz bei EIDE-Festplatten verbessern

Wer bremst, verschenkt

Die EIDE-Festplatten in vielen Linux-Rechnern bleiben weit hinter ihrer Leistungsfähigkeit zurück. Mit Hdparm löst der Anwender die von den gängigen Distributionen eingestellte Handbremse. Thomas Wölfer



Doch Vorsicht: Festplattentuning ist ein bisschen riskant. Sie sollten unbedingt eine Datensicherung durchführen, bevor Sie die Experimente starten. Zudem ist es sinnvoll, Hdparm zunächst im Testbetrieb zu probieren. Booten Sie den Rechner dazu im Single-User-Modus. Als Beispielsystem dient ein AMD Athlon-XP 2600+ mit 256 MByte RAM und einer EIDE-Festplatte. Nach Angaben des Herstellers schafft die Platte eine maximale Datentransferrate von 66 MByte/s.

Werte liefert, wenn der Computer während der Tests nicht belastet ist.

Bei weiteren Testläufen auf dem Beispielsystem zeigte sich, dass Hdparm nur ganz leicht voneinander abweichende Ergebnisse ermittelte, und zwar in hohem Maß ernüchternde: Die Festplatte kommt lediglich auf magere 3,8 MByte/s – das ist Steinzeit.

Einstellungen einsehen

Um die Festplatte zur Höchstleistung zu treiben, sehen Sie sich zuerst die aktuellen Einstellungen an. Dazu starten Sie Hdparm einfach ohne Optionen und übergeben nur das entsprechende Device, wie in [Abbildung 2](#) zu sehen. Die meisten Optionen sind ausgeschaltet, obendrein behandelt der Kernel den Festplatten-Controller lediglich als 16-Bit-Multi-IO-Baustein (»I/O Support«). Kein Wunder, dass bei solchen Einstellungen nur ein magerer Durchsatz von knapp 4 MByte/s rauskommt. Dies gilt es, im Folgenden zu ändern. Die einzelnen Hdparm-Schalter sind aber nicht auf Anhieb zu verstehen.

Optionen-Dschungel

Die Option »multcount« steht für Multiple Sector Count. Sie bestimmt, wie viele Sektoren ein I/O-Interrupt von der Platte liest. Die meisten Platten unterstützen zwei, vier, acht oder 16 Sektoren.

Schlechter Datendurchsatz auch bei modernsten Festplatten ist eine unangenehme und zudem unnötige Folge so ziemlich jeder Standardinstallation. Debian, Suse & Co. sorgen mit äußerst konservativen Standardeinstellungen für maximale Kompatibilität während des Setups – zu Lasten der Leistung. Das Programm Hdparm [1] löst diese künstliche Bremse einfach und schnell. Unser Workshop zeigt, wie Sie sich mit Hdparm Schritt für Schritt zur eigentlichen Transferleistung der Festplatte vorstasten. Es lohnt sich: Je nach Hardware kann die I/O-Performance sogar um das Zehnfache ansteigen. Alle aktuellen Festplatten unterstützen diverse PIO-Modi (Programmed Input Output), Ultra-DMA (Direct Memory Access, [2]) und viele weitere Besonderheiten und Funktionen – Standardinstallationen berücksichtigen die wenigsten davon. Mit Hdparm übergeben Sie der Festplatte die entsprechenden Werte und holen alles an Leistung heraus, was die Platte zu bieten hat.

Behäbige Leistung

Wie schnell Ihre Festplatte momentan arbeitet, finden Sie mit den beiden Optionen »-T« und »-t« heraus ([Abbildung 1](#)). Geben Sie diese Optionen bei Tests immer zusammen an. Hdparm misst die Geschwindigkeit des Cache (Speicher, CPU und Buffer-Cache) bei Lesezugriffen und ermittelt die Leistung der Platte, ohne die Daten im Cache zu berücksichtigen. Damit erhalten Sie eine effektive Aussage zur Performance der angegebenen Festplatte.

Die Manpages von Hdparm weisen darauf hin, dass der Test mehrmals laufen sollte, um einen Mittelwert zu erhalten. Beachten Sie außerdem, dass Hdparm nur dann aussagekräftige (korrekte)

```

Befehlsfenster - Konsole
jrttestl:~ # 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
jrttestl:~ # █

```

Abbildung 1: Ohne Konfiguration mit Hdparm bringt die Festplatte nur einen Bruchteil ihrer Leistung. Statt des vom Hersteller angegebenen theoretischen Durchsatzes von 66 MByte/s erreicht sie gerade mal 3,8 MByte/s.

```

jrttest1:~ # 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
jrttest1:~ # █

```

Abbildung 2: Ohne Optionen ausgeführt liefert Hdparm eine Liste zurück, die neben den physikalischen Kenngrößen (»geometry«) auch zeigt, ob sich die Festplatte im DMA-Modus befindet.

ren. Möglicherweise lesen die Köpfe sogar noch mehr als 16 Sektoren. Wie viele Ihr Festplattenmodell bewältigt, ermittelt die Option »-i«. Den Wert liefert Hdparm neben »MaxMultSect« zurück.

Doch können Sie sich nicht immer auf diesen Wert verlassen. Einige Platten behaupten einfach, eine bestimmte Anzahl von Sektoren zu verkraften, sind aber im Betrieb überfordert. Dann ist Datenverlust so gut wie sicher. Der beste Schutz: erst Backup anlegen, dann testen, dann festschreiben. Sind Sie und Ihre Daten sicher, setzen Sie den Wert für »multcount« mit der Option »-m«. Über I/O-Support teilen Sie Hdparm mit, wie die Daten vom PCI-Bus zum Controller gelangen. Moderne Festplatten-Controller müssen bestimmt nicht im 16-Bit-Modus fahren. Den I/O-Support konfigurieren Sie mit der Option »-c«:

```
hdparm -c3 -m16 /dev/hda
```

Damit ist der I/O-Support auf Mode »3« gestellt. Es handelt sich dabei um einen von mehreren 32-Bit-Modi. Alle Modi listet die Manpage bei der Option »-c« auf. Mit »-m16« stellen Sie die 16-Sektoren-Reads ein. Der nächste Test des Bei-

spielsystems liefert schon deutlich bessere Werte (Abbildung 3). Mit nur wenig Aufwand hat sich der Durchsatz der Festplatte fast verdoppelt: von 3,8 auf 7,6 MByte/s. Trotzdem sind Sie aber immer noch ein gutes Stück von dem theoretisch möglichen Durchsatz von 66 MByte/s entfernt.

Aktiver DMA vervielfacht den Durchsatz

Als nächste Tuningmaßnahme lassen Sie jetzt das Unmasking von Interrupts zu: »umaskirq«. Normalerweise macht der Kernel nichts anderes, während er auf einen Interrupt wartet. Das trägt wesentlich zur Stabilität des Systems bei. Fällt dieser Schutz, erhöht sich jedoch die Antwortfreudigkeit des Systems. Das System reagiert darauf in den meisten Fällen so instabil, dass es nicht mehr zu gebrauchen ist.

Ähnliches gilt, wenn Sie zudem den DMA-Modus einschalten: »using_dma«. Auch hier gibt es eine Fülle von Optionen. Doch das System bleibt gerne stehen, wenn es mit den falschen Einstellungen laufen soll. Zum Glück: Ein Re-

boot rettet die Lage immer, sodass Sie auch hier testen können, ohne mit bleibenden Problemen rechnen zu müssen. Falls Ihnen die Dokumentation von Festplatte und Controller beziehungsweise Motherboard vorliegt, entnehmen Sie dort die zulässigen DMA-Modi. Ohne Dokumentation müssen Sie so lange mit den Optionen spielen und immer wieder testen, bis Sie die idealen Einstellungen gefunden haben. Für das Testsystem erwiesen sich folgende Werte als passend:

```
hdparm -d1 -X66 -u1 -m16 -c3 /dev/hda
```

Die Option »-d1« schaltet das »using_dma«-Flag ein. Zudem geben Sie über die Option »-X66« an, welchen Modus Sie benutzen möchten. In unserem Beispiel ist dies der Ultra-DMA-Modus »2«. Mit der Option »-u1« erlauben Sie dem Festplatten/Controller-Treiber, die Maskierung anderer Interrupts aufzuheben (»umas-kirq«).

Der Test bringt erstaunliche Ergebnisse (Abbildung 4): Der Durchsatz steigt mit den neuen Parametern im Vergleich zur Standardeinstellung um mehr als das Fünffache. Allerdings halten die neuen Werte nur bis zum nächsten Reboot.

Einstellungen übernehmen

Um die Parameter dauerhaft einzustellen, tragen Sie sie in eines der Startskripte »rc*« ein. Das allein reicht aber noch nicht aus. Falls der Controller einen IDE-Reset durchführt, gehen die vorgenommenen Einstellungen verloren. Die Option »-k1« verhindert dies. Sie sollten die Option allerdings erst dann nutzen, wenn Sie sicher sind, dass Ihr Rechner mit den aktuellen Einstellungen fehlerfrei läuft. Das ist der Fall, wenn das System von der Platte lesen und schreiben kann und keine Fehler in »/var/log/messages« auftauchen. (jre) ■

Infos

[1] Hdparm: <http://www.ibiblio.org/pub/Linux/system/hardware/>

[2] IDE-Kurs: <http://www.tomshardware.de/storage/20020730/ide-04.html>

Der Autor

Thomas Wölfer ist Software-Entwickler und administriert die Server des IT-Portals Nickles.de.

```

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:~ # █

```

Abbildung 3: Den Controller auf 32-Bit-I/O-Support schalten – schon verdoppelt sich der Durchsatz.

```

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:~ # █

```

Abbildung 4: Vor allem der richtigen Ultra-DMA-Modus steigert die Performance.