

Das optimale Dateisystem: Performance, Features, Optionen

# Qual der Wahl

Der Umstieg auf ein neues Dateisystem kostet Zeit und Nerven, lohnt sich aber, um die Stabilität und Performance des Rechners zu erhöhen. Dieser Artikel hilft bei der Wahl des optimalen Filesystems. Marcel Hilzinger



**Die Auswahl** des Dateisystems ist bei der Installation aktueller Linux-Distributionen zwar mit wenigen Mausklicks erledigt, die meisten Benutzer bleiben jedoch beim voreingestellten Filesystem und wechseln höchstens auf Grund schlechter Erfahrungen oder mit dem Umstieg auf eine andere Distribution. Es lohnt sich jedoch, bei der Entscheidung einen Blick über Ext 3 und ReiserFS hinaus zu riskieren.

Die Redaktion hat dafür einen erheblichen Testaufwand betrieben. Der eingesetzte leistungsstarke Server lief allein drei Wochen ununterbrochen (siehe [Kasten „So haben wir getestet“](#)). Laut Benchmarks mit den Filesystemen Ext 2, Ext 3, JFS, ReiserFS, Reiser 4, XFS und VXFS von Veritas haben JFS und XFS unter Kernel 2.6 an Performance gewonnen. Auch Reiser 4 will sich mit Rekordgeschwindigkeiten und neuem Design **[1]** sein Stück vom Dateisystem-Kuchen

abschneiden. Obwohl sich jedes Filesystem zu jedem Zweck verwenden lässt, gibt es enorme Unterschiede bei Schreibgeschwindigkeit, Datensicherheit und Feature-Angebot. Bei der Lese-Performance sind die Unterschiede geringer.

## Lesen ist langweilig

Die kleinsten Unterschiede weisen die einzelnen Dateisysteme beim Lesen auf. Hier haben in der Regel auch die verschiedenen Mount-Parameter kaum Einfluss. Die Abweichungen zwischen dem schnellsten und dem langsamsten Dateisystem liegen innerhalb der 20-Prozent-Grenze. Dieser Artikel geht deshalb überwiegend auf die Schreib-Performance ein.

Nur ein Lese-Ergebnis unter dem Suse Linux Enterprise Server 8 tanzt aus der Reihe (siehe [Abbildung 2b](#)). Auch hier zeigt sich das Feld zwar geschlossen,

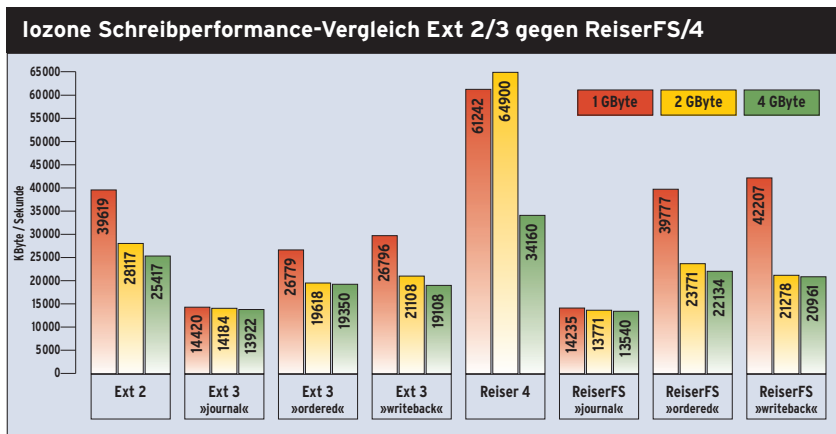
aber das Veritas-Dateisystem VXFS ist eine Ausnahme. Es kommt beim Lesen nur auf rund 60 MByte/s. Die Resultate der Konkurrenten mit Werten um 100 MByte/s lassen allerdings Zweifel am Messwert aufkommen, auf die der Artikel noch eingeht.

## Mainstream

Der überwiegende Teil der Linux-Rechner benutzt Ext 3 oder ReiserFS als Hauptdateisystem, da dies der Standardvorschlag der wichtigsten Distributionen ist. Der Test nimmt daher die zwei Dateisysteme genauer unter die Lupe. Vorweg: Weder sind ReiserFS und Ext 3 besonders schnell, noch weisen sie per Default ein besonders großes Feature-Spektrum auf. Da die einzelnen Distributoren aber ihr bevorzugtes Filesystem mit Patches versehen, gibt es große Unterschiede zwischen den Systemen. Die Testergebnisse sind deshalb unter diesen Vorbehalten zu betrachten.

Ext 3 und ReiserFS gehören zur Familie der Journaling-Dateisysteme. Da beim Journaling streng genommen alle Daten zweimal geschrieben werden, ist nicht mit Spitzengeschwindigkeiten zu rechnen. In der Grundeinstellung benutzen Ext 3 und ReiserFS aber den Modus »data = ordered«, bei dem das Dateisystem nur die Metadaten – die Informationen über die Bewegungen im Dateisystem – im Journal speichert, die eigentlichen Daten nicht. Zu den Mount-Optionen siehe auch [Tabelle 1](#).

Steht nicht die Geschwindigkeit, sondern die Integrität der Daten an erster Stelle, ist die Option »data = journal« für das Einhängen der jeweiligen Partition zu empfehlen. Bei Ext 3 ist dabei mit einem Schreibperformance-Verlust bis zu



**Abbildung 1:** Schreibperformance-Vergleich mit dem Iozone-Benchmark unter Kernel 2.6.7-mm4. Im Journal-Modus sind Ext 3 und ReiserFS etwa gleich langsam. Writeback- und Ordered-Modus bringen bei Ext 3 bis zu 40 Prozent schnelleren Datentransfer. Bei ReiserFS macht der Unterschied bei den kleinsten Dateien bis zu 120 Prozent aus. Als Vergleich sind zusätzlich die Ergebnisse von Ext 2 und Reiser 4 angegeben.

50 Prozent zu rechnen. Bei größeren Dateien beträgt der Unterschied um die 20 Prozent (siehe [Abbildung 1](#)). Diese Option ist aber auch nicht für das Wurzelverzeichnis gedacht, sondern für separate Datenpartitionen.

Die Option »data = writeback« erlaubt dem Filesystem ins Journal zu schreiben, bevor alle Daten an ihrem Zielort angekommen sind. Sie ermöglicht einen Performancegewinn bis zu 10 Prozent und eignet sich gut für das Root-Dateisystem, da hier in der Regel kein Datenverlust zu befürchten ist. Ein separates »/home«- oder Dokumentenverzeichnis ist dann allerdings zu empfehlen.

## Ext 2 ganz flott

Erstaunlich gut für sein Alter schlägt sich auch Ext 2. Bei der Total-Run-time-Statistik (siehe [Abbildung 5](#)) muss es

sich nur von Reiser 4 und bei größeren Dateien von JFS und XFS geschlagen geben. Die Statistik berücksichtigt allerdings die »mkfs«-Zeiten nicht. Der Test zeigt auch, dass die Journaling-Fähigkeit von Ext 3 als Erweiterung zum Ext-2-System einen merklichen Performanceverlust mit sich bringt.

Für Partitionen ohne wichtige Daten, zum Beispiel eine separate »/tmp«-Partition, lohnt es sich deshalb allenfalls, Ext 2 statt Ext 3 einzusetzen, falls die Partitionsgröße nicht über die GByte-Marke steigt. Ansonsten ist Ext 2 nicht zu empfehlen, da der Dateisystem-Check unter Umständen länger dauert als das Einspielen eines Backups. Unter Kernel 2.6 lohnt sich aber der Einsatz von XFS, JFS oder Reiser 4 in jedem Fall.

Bei den zahlreichen Kerneloptionen ist es generell schwierig, zwei Kernel zu finden, die wirklich vergleichbar sind.

Da die Tester auch das proprietäre VXFS von Veritas einbeziehen wollten, mussten sie bei 2.4 einen United-Linux-Kernel oder einen Kernel von Red Hat Enterprise Linux nehmen, da Veritas nur diese Distributionen unterstützt. Sie entschieden sich für Kernel 2.4.21-138-smp von Suse Linux Enterprise Server 8 mit dem United-Linux-Servicepaket 3. Bei 2.6 kam der neueste, von Reiser 4 unterstützte Kernel mit den Patches von Andrew Morton (2.6.7-mm4) und den Reiser-4-Patches zum Einsatz.

## 2.4 oder 2.6

Der 2.4er Kernel von Suse Linux Enterprise Server 8 schreibt bei den meisten Dateisystemen schneller als Kernel 2.6.7-mm4 (siehe [Abbildung 2a](#)). Das kommt nicht ganz überraschend, da der SLES-Kernel bereits viele Features des offiziellen 2.6er Kernels enthält. Der Vorsprung ist vermutlich den Patches der Nürnberger zuzuschreiben. Erstaunen lösten hingegen die großen Unterschiede beim Lesen aus, kommt doch der Suse-Kernel der absoluten Durchsatzrate der SATA-Festplatten von 104 MByte/s (laut »hdparm«) bedrohlich nahe und ist damit gut 70 Prozent schneller als Kernel 2.6 (siehe [Abbildung 2b](#)).

Die Tester haben daher die Resultate mit Bonnie ++ und dem »slow.c«-Benchmark überprüft, mit den gleichen Ergebnissen zu Gunsten des SLES-8-Kernels. Bei der Schreibperformance zeigt jedoch der »slow.c«-Benchmark ein leicht differenziertes Bild (siehe [Abbildung 3a](#) und [3b](#)). Nach diesem Test, der die Durchsatzrate bei gleichzeitigem Schreiben

**Tabelle 1:** Mount-Optionen von Ext 3 und ReiserFS

Option	Erklärung
data=journal	Diese Option, bei der alle Daten zunächst in den Journal-Bereich kopiert werden, erst dann an den eigentlichen Bestimmungsort, garantiert höchste Datensicherheit. Mit ihr sinkt aber die Datendurchsatzrate sowohl bei Ext 3 als auch bei ReiserFS beinahe auf die Hälfte, da jeder Schreibvorgang doppelt so lange dauert.
data=ordered	Dies ist die Standardoption. Das Dateisystem schreibt zunächst die Daten an den Bestimmungsort und trägt dann die Informationen über den abgeschlossenen Vorgang im Journal ein.
data=writeback	Mit dieser Mount-Option, die bei Ext 3 einen Performancegewinn von etwa 10 Prozent, bei ReiserFS bis zu 30 Prozent gegenüber der Default-Option bringt, darf das Dateisystem bereits in das Journal schreiben, bevor alle Daten an ihrem Bestimmungsort angekommen sind. Bei einem Crash kann es vorkommen, dass durch den Dateisystemcheck alte Daten in Dateien auftauchen. Die Option ist für ReiserFS nur unter Kernel 2.6 verfügbar.
data=notail	Nur bei ReiserFS. ReiserFS benutzt Leerraum in Blöcken, um darin Teile von Daten zu speichern, die nicht in einen Block passen. Der Schwanz (engl. Tail) der Datei wird also abgeschnitten und in einem anderen Block gespeichert. ReiserFS speichert damit 10 bis 20 Prozent mehr Dateien auf derselben Partition als beispielsweise Ext 3. Weil damit ein leichter Performanceverlust bis zu 5 Prozent verbunden ist, lässt sich dieses Feature mit der Option »data=notail« abschalten.

mehrerer Datenströme misst, haben JFS und XFS in Kernel 2.6 viel an Performance gewonnen. Zumindest für XFS bestätigen dies auch die Benchmark-Ergebnisse von Bonnie++ und Iozone.

## Pole Position

Steht Performance an erster Stelle auf der Dateisystem-Wunschliste, ist Reiser 4 klarer Favorit. Kein anderes Dateisystem steht bei den verschiedenen Tests so oft auf dem ersten Platz. Alle Tests untermauern auch die Behauptung auf der Namesys-Homepage [2], Reiser 4 sei doppelt so schnell wie ReiserFS. Beim sequenziellen Anlegen von 50000 Dateien ist es sogar dreimal so flink wie das zweitschnellste Dateisystem XFS (siehe [Abbildung 4](#)).

Hier machen sich die Schreib-Optimierungsarbeiten an Reiser 4 deutlich bemerkbar. Auch bei den übrigen Schreibtests ist Reiser 4 zumindest 10 bis 20 Prozent flotter als das nächstschnellste

Dateisystem. Es darf sich somit zu Recht insgesamt als schnellstes Linux-Dateisystem bezeichnen.

Mit Blick auf die Ressourcen sieht es für Reiser 4 hingegen weniger gut aus. So liegt zum Beispiel die CPU-Belastung beim erwähnten sequenziellen Anlegen von 50000 Dateien für Reiser 4 bei 26 Prozent. Die übrigen Dateisysteme geben sich hier schon mit 1 bis 4 Prozent zufrieden. Reiser 4 liegt bei der Prozessorbelastung durchschnittlich um 10 Prozentpunkte über allen andern Dateisystemen, manchmal sind es aber auch 50 oder mehr. Nur ReiserFS überbietet Reiser 4 in einigen Testkategorien in dieser Hinsicht.

Auch auf der Feature-Liste von Reiser 4 fehlt es zurzeit noch an grundlegenden Funktionen, etwa ACLs (Access Control Lists) oder Quotas (siehe [Tabelle 2](#)). Der erst für Version 4.1 angekündigte Repacker zur Defragmentierung des Filesystems ist hingegen bereits im Betazustand. Laut ersten Testberichten von der

Reiser-Mailingliste bringt er nach mehreren Durchläufen die volle Performance von Reiser 4 zurück.

## Veritas VXFS

Das proprietäre Journaling-Dateisystem von Veritas [3] hinterlässt einen gemischten Eindruck. Da es nur für Kernel 2.4 erhältlich ist, sind direkte Geschwindigkeitsvergleiche mit Reiser 4 leider nicht möglich, VXFS gehört aber zu den schnellsten Systemen. In den meisten Fällen dürfte für den Einsatz von Veritas nicht die Geschwindigkeit der ausschlaggebende Punkt sein, sondern dessen Feature-Vielfalt.

So kennt VXFS zwei zusätzliche Journal-Optionen: »mincache = closesync« und »tmplog«. Die erste Option ist für Desktopsysteme gedacht. Veritas schreibt damit die Daten beim Speichern durch den Benutzer sofort auf die Platte. Es verhindert so, dass bei einem versehentlichen Abstellen des Rechners Daten verloren

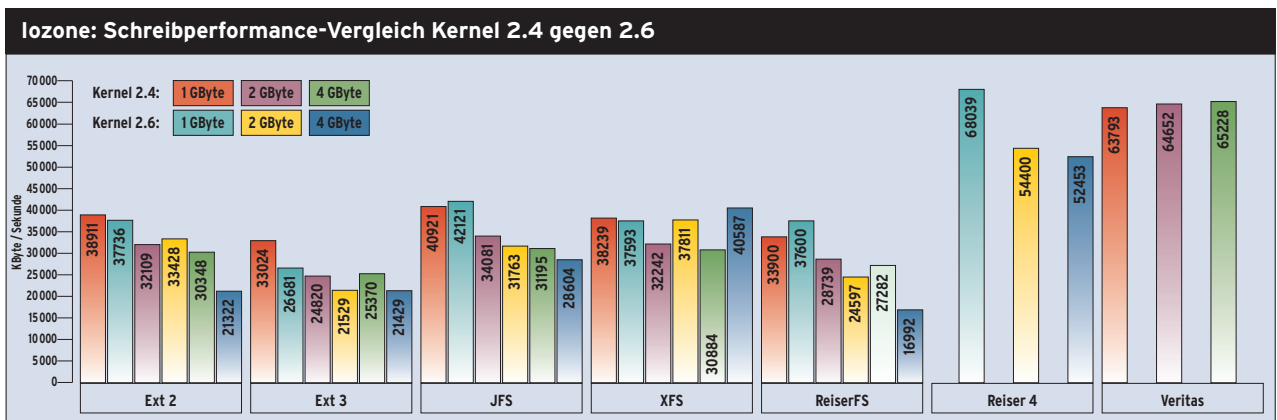


Abbildung 2a: Schreibperformance-Vergleich mit dem Iozone-Benchmark zwischen dem SLES-8-SPM-Kernel und Kernel 2.6.7-mm4 mit 4-KByte-Blöcken. Außer XFS sind alle Dateisysteme unter dem Suse-Kernel schneller. Reiser 4 und Veritas sind deutlich schneller als alle anderen Dateisysteme.

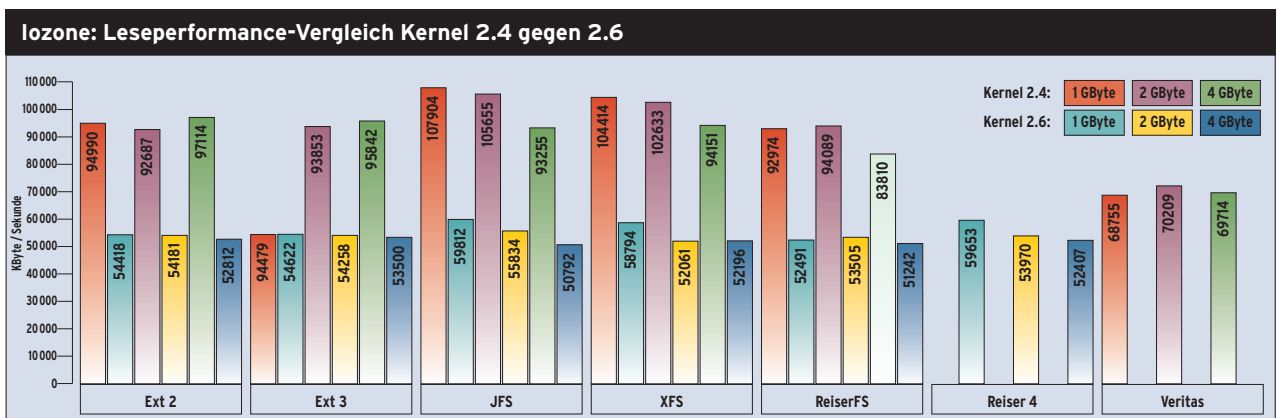
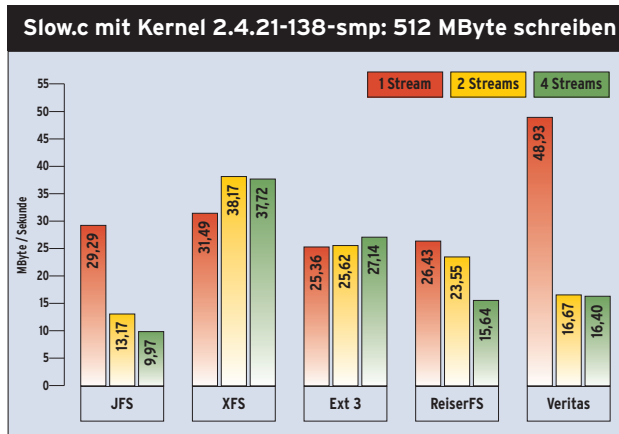
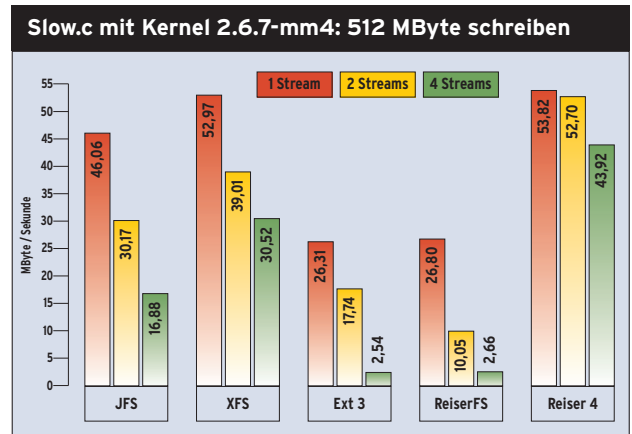


Abbildung 2b: Leseperformance-Vergleich mit dem Iozone-Benchmark zwischen dem SLES-8-SPM-Kernel und Kernel 2.6.7-mm4 mit 4-KByte-Blöcken. Unter dem Suse-Kernel lesen die Dateisysteme um zirka 70 Prozent schneller. Veritas macht hier allerdings einen schlechten Eindruck.



**Abbildung 3a:** Nach den Ergebnissen des »slow.c«-Benchmark arbeitet das Veritas-Dateisystem sehr schnell. Beim Schreiben von mehreren Datenströmen verliert es jedoch enorm an Geschwindigkeit. Überraschend sind die - mehrfach überprüften - Resultate von Ext 3.



**Abbildung 3b:** JFS und XFS schneiden unter 2.6 deutlich besser ab als unter dem Suse-Kernel. XFS kommt dabei fast an Reiser 4 heran. Das schlechte Abschneiden von ReiserFS und Ext 3 lässt vermuten, dass die Nürnberger viel Energie in das Patchen der Dateisysteme in ihrem Kernel investieren.

gehen. Die zweite Option ist für temporäre Verzeichnisse gedacht und bewirkt, dass VXFS das Journal so selten wie möglich nachträgt. Veritas kann zudem Momentaufnahmen von VXFS-Dateisystemen speichern (Snapshots), um trotz laufenden Betriebs ein Backup zu erstellen. Die Preise für VXFS 2.2 samt Volume Manager und Administrationstools beginnen bei 1570 Euro.

Der Versuch, die verschiedenen Mount-Optionen von VXFS zu differenzieren, scheiterte, da »vxfs« bei allen Optionen ähnliche Zeiten lieferte. Da die Optionen laut Veritas-Dokumentation Auswirkung auf die Performance des Dateisystems haben sollen, gingen die Tester zunächst davon aus, dass der Benchmark nicht zuverlässig lief. Aber auch das Überprüfen der Ergebnisse mit dem »slow.c«-Benchmark lieferte keine abweichenden

Resultate. Eine Nachfrage der Redaktion konnte Veritas Deutschland bis zur Drucklegung des Magazins nicht befriedigend beantworten.

Zum gemischten Eindruck trägt auch bei, dass Bonnie++ beim Test mit 1 GByte Daten bei 50000 respektive 100000 Testdateien abstürzte. Der Effekt ist reproduzierbar, der Fehler ließ sich allerdings nicht ausmachen. Schließlich gehören auch die Total-Run-time-Zeiten von VXFS zu den schlechtesten (siehe [Abbildung 5](#)), was die Zweifel an der Geschwindigkeit des Dateisystems nährt.

## XFS und JFS

XFS hat das Zeug zum Dateisystem erster Wahl aufzusteigen. Im Gegensatz zu Reiser 4 kann es schon jetzt auf einen aktiven Benutzerkreis zurückblicken

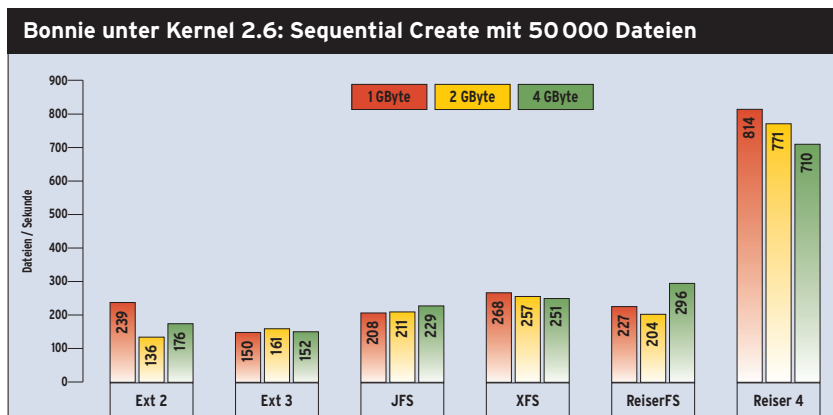
und unterstützt Quotas, ACLs sowie erweiterte Attribute. Es ist das schnellste Dateisystem nach Reiser 4 und macht ihm in einigen Kategorien den ersten Rang streitig. XFS schneidet in erster Linie bei großen Datenmengen am besten ab. Als einziges Dateisystem kann es seine Performance auch bei 4 GByte Testdaten noch steigern.

Durchgehend im Mittelfeld bewegt sich JFS. Da Features wie Quota-Support und erweiterte Attribute immer noch auf der Todo-Liste der Entwickler stehen, gibt es auf den ersten Blick keinen überzeugenden Grund für das von IBM stammende Journaling File System. Laut Bonnie++ ist JFS jedoch das Journaling-Dateisystem, das im Verhältnis zur Performance die wenigsten Ressourcen beansprucht (siehe [Abbildung 6](#)).

## ACL und erweiterte Attribute

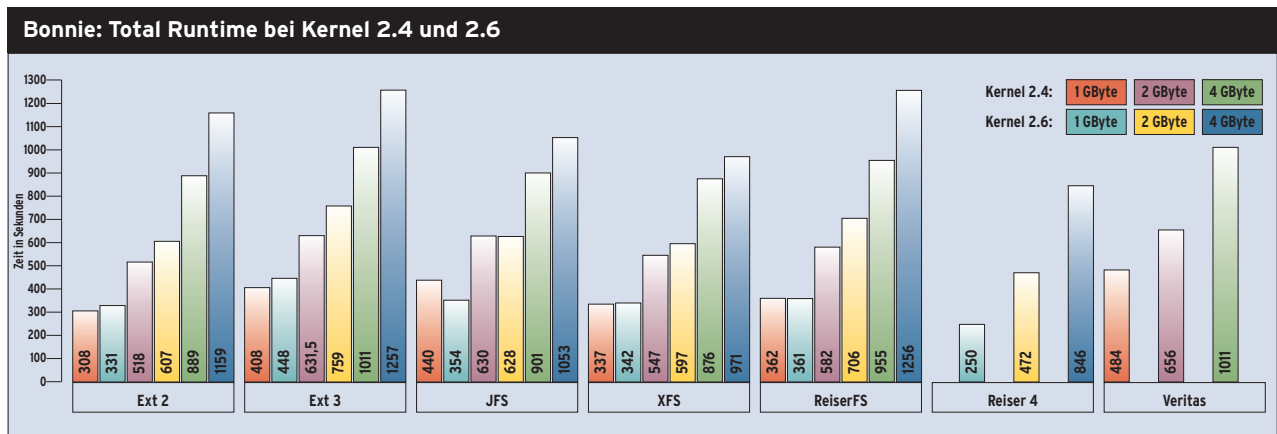
Ein wichtiger Gesichtspunkt bei der Auswahl des perfekten Dateisystems sind auch Features wie ACL, Quota und erweiterte Attribute. Auch hier gibt es große Unterschiede in puncto Dateisystem und Kernelversion. Die Übersicht in [Tabelle 2](#) bezieht sich auf die zur Drucklegung aktuelle Version des jeweiligen Dateisystems, beschränkt sich aber nicht nur auf den offiziellen Kernel, sondern bezieht auch Patches mit ein.

Access-Control-Listen führen alle der getesteten Dateisysteme außer Reiser 4. Ursprünglich als Samba-Erweiterung ge-



**Abbildung 4:** In der Kategorie Sequential Create von Bonnie++ unter Kernel 2.6.7-mm4 schlägt Reiser 4 die Konkurrenz um Längen. Hier machen sich die Schreib-Optimierungsfunktionen von Reiser 4 am meisten bemerkbar. ReiserFS liegt auf der Höhe von XFS, kann jedoch die kleinsten Dateien schneller anlegen.





**Abbildung 5:** Total Runtime misst die Zeit für alle Tests pro Kategorie. Das Veritas-Dateisystem schnitt hier überraschenderweise am schlechtesten ab. Unter Kernel 2.6.7-mm4 schlägt Reiser 4 einmal mehr die Konkurrenz deutlich. Den zweiten Platz belegt zumindest bei kleinen Dateien Ext 2.

dacht lassen sich mit ACLs nun auch in nativen Linux-Dateisystemen differenziertere Zugriffsberechtigungen einstellen. Leider ist dieses Feature de facto auf die Textebene beschränkt und es gibt weder für KDE noch für Gnome grafische Dateimanager, die nach dem Bearbeiten einer Datei auch die ACLs oder erweiterten Attribute wieder korrekt speichern. Auch diesen Aspekt darf der Sysadmin bei der Dateisystemauswahl nicht außer Acht lassen.

Mit dem Einsatz von ACLs und erweiterten Attributen ist zudem bei allen Dateisystemen ein geringer Performanceverlust verbunden. Schließlich ist noch die Anzahl der Benutzer zu berücksichtigen, da XFS höchstens 25 und Ext 2/3 höchstens 32 ACL-Einträge pro Datei zulassen. Bei ReiserFS und JFS liegt diese Grenze theoretisch bei 8191 Einträgen, in der Praxis sind jedoch bei entsprechenden Gruppenaufteilungen fünf bis zehn ACL-Einträge pro File meist ausreichend.

## Backup

Benutzer von Ext 2 und Ext 3 hängen oft an den Tools »dump« und »restore«. Ein ähnliches Werkzeug kennt auch XFS, die Funktionalität der Hilfsprogramme ist jedoch nicht identisch. Dump für Ext 2 oder Ext 3 speichert weder ACL-Informationen noch erweiterte Attribute.

Besser sieht es bei »xfsdump« aus. Es speichert laut Homepage [4] sowohl die erweiterten Attribute als auch ACLs – ein weiterer Pluspunkt für das ehemalige Irix-Dateisystem. ReiserFS und Reiser 4 verzichten ganz auf Dump und

Restore, da Tools wie »star« diese Funktionen besser wahrnehmen. Standard Tar von Jörg Schilling speichert ACLs und erweiterte Attribute [5].

## Speicherbedarf

Linux-Benutzer wissen, wie wichtig genügend Hauptspeicher für die Performance des Rechners ist. Das Dateisystem macht da keine Ausnahme. So benutzen zum Beispiel Reiser 4 und XFS eine besondere Technik, um Daten erst dann auf die Platte zu schreiben, wenn dies wirklich nötig ist. Damit unterlässt es das Dateisystem, Datenmüll, etwa von Temporärdateien, auf die Platte zu schreiben. Je größer der Hauptspeicher, desto effizienter schreibt das Dateisystem auf die Platte, da es mit weniger Schreibzugriffen auskommt und mehr Daten am Stück überträgt.

Schließlich kann auch der Code-Umfang des Dateisystems ein Entscheidungsgrund sein. Das Kernelmodul für XFS beansprucht stolze 511 KByte. Auch Reiser 4 dürfte mit 367 KByte eine eigene Moduldiskette beanspruchen, wenn per Floppy gebootet werden soll. Gemäßig-

ter geben sich ReiserFS mit 205 KByte und JFS mit 165 KByte. Sieger in dieser Kategorie ist Ext 3, das nur 106 KByte im Kernel beansprucht.

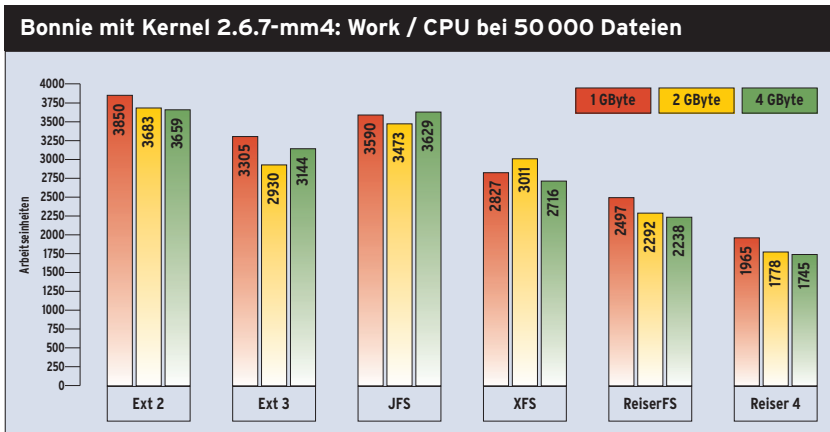
## Fazit

Für ältere Rechner und Festplatten mit kleinen Partitionen eignet sich nach wie vor Ext 2 am besten. Hier Ext 3 einzusetzen lohnt sich nicht, da Ext 3 wesentlich langsamer als Ext 2 ist und Abstürze wegen Ext 2 sehr unwahrscheinlich sind, womit auch der Filesystem-Check keine entscheidende Rolle spielt.

Auch in Veritas steckt viel Know-how. Das Snapshot-Feature, das eine Art Live-Backup erstellt, hat nicht umsonst auch im Windows-2003-Server Einzug gehalten. Als Nebeneffekt treten aber hohe Kosten für das proprietäre Dateisystem und die entsprechende Serversoftware von Suse oder Red Hat auf.

Steht die Integrität der Daten im Vordergrund, empfiehlt es sich, den Modus »data = journal« von Ext 3 oder ReiserFS zu verwenden. Will JFS Systemadministratoren und Benutzer überzeugen, müssen IBM und die JFS-Entwickler in ihr

Tabelle 2: Dateisystem-Features							
	Ext 2	Ext 3	JFS	ReiserFS	Reiser 4	Veritas	XFS
Journaling		✓	✓	✓	✓	✓	✓
ACL	✓	✓	✓	✓		✓	✓
Erweiterte Attribute	✓	✓	✓	✓			✓
Quota	✓	✓		✓		✓	✓
Online resize	✓	✓	✓	✓		✓	
Defrag					✓	✓	✓
Dump/Restore	✓	✓				✓	✓



**Abbildung 6:** Mit Arbeit pro CPU misst Bonnie++, welche Prozessorleistung ein Dateisystem verbraucht. Die auf der Abbildung zu sehenden Ergebnisse für Kernel 2.6 decken sich weitgehend mit denen von Kernel 2.4.

Dateisystem investieren, um ihm zu größerer Geschwindigkeit und mehr Möglichkeiten zu verhelfen. Zählt nur die Geschwindigkeit des Dateisystems, ist Reiser 4 mit Kernel 2.6 ein Muss. Da die Software noch sehr jung ist, dürfte es allerdings nicht schaden, wichtige Daten auf einer Nicht-Reiser-4-Partition zu speichern.

Einen guten Kompromiss zwischen Stabilität und Performance verspricht XFS. Mit seinen zahlreichen Tools bietet es dem Systemadministrator zurzeit die meisten Möglichkeiten. Auch vermochte es seine Geschwindigkeit unter Kernel 2.6 zu verbessern.

Die ausführlichen Testergebnisse stehen unter: [[http://www.linux-magazin.de/Service/Listings/2004/11/fs\\_bench](http://www.linux-magazin.de/Service/Listings/2004/11/fs_bench)] ■

### Infos

- [1] Marcel Hilzinger, „IV. Symphonie von Reiser“: Linux-Magazin 09/04, S. 38
- [2] Namesys: [<http://www.namesys.com>]
- [3] Veritas: [<http://www.veritas.com/de>]
- [4] XFS: [<http://oss.sgi.com/projects/xfs/>]
- [5] Star: [<http://www.fokus.gmd.de/research/cc/glone/employees/joerg.schilling/privat/star.html>]
- [6] Fsbench: [<http://fsbench.netnation.com>]
- [7] Bonnie++: [<http://www.cocke.com.au/bonnie++/>]
- [8] lozone: [<http://www.iozone.org>]
- [9] Slow.c Benchmark: [<http://www.jburgess.uklinux.net/slow.c>]
- [10] Mirco Dölle, Jörg Reitter, „Pisa-Studie - Performance-Vergleich Ext 2/3, JFS, ReiserFS und XFS“: Linux-Magazin 03/04, S. 38

### So haben wir getestet

Für die Tests benutzten wir einen 3-HE-Server RM312 von Thomas-Krenn.com, der mit einem auf 2,8 GHz getakteten Pentium-4-Prozessor und 512 MByte Hauptspeicher ausgestattet ist. Für die Tests standen zwölf SATA-Festplatten mit einem Raid-Verbund von knapp 2 TByte zur Verfügung. Davon belegte die Testpartition 200 GByte.

Als Betriebssystem für die Benchmarks mit dem 2.4er Kernel diente Suse Linux Enterprise Server 8 mit Kernel 2.4.21-138-smp. Für die Tests unter Kernel 2.6 benutzten wir Suse Linux 9.1 mit Kernel 2.6.7-mm4 und den entsprechenden Patches für Reiser 4.

Über drei Wochen arbeitete der Testrechner ununterbrochen an den Dateisystem-Benchmarks. Die Testarbeit übernahmen die »fs\_bench«-Perl-Skripte von [6] mit den Benchmarks Bonnie++ [7] und lozone [8]. Sie testeten die Durchsatzrate des Dateisystems bei

verschiedenen Schreib- und Lesemethoden. lozone prüfte 1,2 und 4 GByte große Dateien mit unterschiedlicher Blockgröße. Bei Bonnie bildeten jeweils 10, 25, 50 und 100 000 Dateien das Testvolumen, wobei Bonnie auch die Prozessorbelastung maß.

Zur Überprüfung der Ergebnisse benutzten wir den »slow.c«-Benchmark [9]. Alle Tests verliefen grundsätzlich wie bei [10], allerdings testeten wir bei den Journaling-Dateisystemen auch verschiedene Mount-Optionen.



**Abbildung 7:** Unser Pentium-4-Testsystem von Thomas-Krenn.com.