

Verschiedene Dateisysteme im Vergleich

Renzo Lauper

Rechnergestützte Wissenschaften

D-MATH ETH Zürich - März 2002

1 Einleitung

Die Verwendung von normalen PCs findet eine immer grössere Verbreitung. PCs kommen in den verschiedensten Anwendungsbereichen vor, wie Fileserver, Printserver, Webserver, Mailserver, Heimcomputer usw. Dies ist nicht zuletzt möglich dank der rasanten Steigerung der Geschwindigkeit in den vergangenen Jahren.

Durch die wachsende Geschwindigkeit von CPU, RAM, Grafikkarten und die immer grösser werdenden Datenmengen wird eine Komponente immer mehr zum Flaschenhals vieler Anwendungen: Die Festplatte.

Es gibt viele Versuche die Datenspeicherung auf Festplatten zu steigern. Auf Softwareseite z.B. RAID, parallele Dateisysteme und auf der Hardwareseite schnellere Hardware, RAID, ...

In dieser Arbeit sollen verschiedene Dateisysteme - namentlich NTFS, FAT32, FAT, EXT2, und ReiserFS - in verschiedenen Anwendungsgebieten untersucht werden. Dabei geht es rein um die Dateisysteme, und nicht um ihre Integration in z.B. ein RAID-System.

2 Tests

2.1 Dateisysteme

1. FAT

Das FAT Dateisystem wurde in den DOS-Versionen bis Version 6.x verwendet und limitiert die Dateinamen auf eine Form von 8.3. Verschiedene Version verwendeten zwischen 12 und 16 Bytes für die Adressierung und hatten entsprechend tiefe Partitionsgrenzen für vernünftig kleine Blockgrössen.

2. FAT32

FAT32 löst als Nachfolge-Dateisystem von FAT einige der Probleme des letzteren und wird bei Win95SE/98/ME als Standard verwendet.

Somit verwendet es nun 32-bit Adressen, was es Partitionen bis zu 2 Terabyte verwalten lässt. Aber es verfügt über kein Journaling. In dieser Arbeit wurde das FAT32 eingesetzt, das nach wie vor mit Windows 2000 mitgeliefert wird.

3. NTFS

Das NTFS Dateisystem wurde parallel zum Betriebssystem MS NT entwickelt. Es beinhaltet unter anderem folgende Features: Change Journals, File compression, File encryption, 64-bit Adressen und Unicode 16-bit character set.

In dieser Arbeit wurde NTFS 5.0 unter Windows 2000 eingesetzt.

4. EXT2

EXT2 wurde 1993 als Erweiterung zu EXT vorgestellt und ist seither das Standard-Dateisystem von Linux. Es wird wohl in naher Zukunft durch EXT3 und/oder ReiserFS abgelöst werden. Dies nicht zuletzt, weil EXT2 über kein Journaling verfügt.

EXT2 ist dank einigen Optimierungen im Kernel sehr stabil und schnell. Die Partitionsgröße liegt bei 4 Terabyte.

5. ReiserFS

ReiserFS ist ein sehr junges Dateisystem, das auf balanced trees basiert. Es verfügt über ein schnelles Journaling und verfügt über Optimierungen bei vielen kleinen Dateien, die es nicht zwingend in einzelne Sektoren speichert und somit Leerstellen im Dateisystem vermindert.

Reiser wurde in der aktuellen Version verwendet. Die neue Version 4.0 ist fürs 3. Quartal 2002 angekündigt.

Bemerkung: Alle Dateisysteme wurden "out-of-the-box" verwendet, d.h. die einzige Einstellung, die jeweils vorgenommen wurde, ist die Blockgröße (4096 ausser bei FAT).

2.2 Betrachtete Faktoren

Der erste Faktor ist die **Dateigröße der Testdateien**. Um die verschiedenen Einsatzgebiete zu simulieren, wurde diese so geändert, dass es den entsprechenden realen Bedingungen so nahe wie möglich kommt. Dabei wurden die tatsächlichen Dateigrößen immer aus einem Bereich zufällig gewählt.

Der zweite, wichtige Faktor war die **Anzahl Transactions**, die an den vorhandenen Dateien vorgenommen wurden. Unter Transaction sind in dieser Arbeit Read, Append, Create und Delete gemeint.

Read und Append sowie Create und Delete bilden je zusammen eine Gruppe, für die einzeln das **Verhältnis des Auftauchens der einen oder der anderen Transaction** eingestellt wurde.

Als feste Parameter wurden folgende Werte gewählt:

- Bei allen Tests wurde jeweils in Blöcken von 4096 Bytes geschrieben und gelesen.
- Die Anzahl Bytes pro logischem Sektor wurde auf 4096 eingestellt. Ausnahme bildet das FATFS, weil dieses die Testpartition von 2GB nur mit einer Blockgrösse von 32kB adressieren konnte.
- Die Dateigrößen und die Art der Transaction wurden zufällig gewählt. Als Seed des Zufallszahlengenerators wurden für alle Dateisysteme die selben verwendet.

2.3 Einsatzgebiete

1. Fileserver

Ein typischer Fileserver ist der eigene Desktoprechner oder Laptop. Ein Fileserver zeichnet sich dadurch aus, dass er sehr unterschiedliche Aufgaben in beliebiger Reihenfolge an stark unterschiedlichen Dateien ausführen muss.

Die Dateigrößen wurden aus einem breiten Bereich von **10Byte bis 10MByte** gewählt und die **Verhältnisse von Read zu Append und von Create zu Delete beide auf 1** gesetzt, d.h. sie treten alle mit gleicher Wahrscheinlichkeit auf.

Dabei entspricht Read dem Öffnen einer Datei und Append und Create dem Speichern.

2. Mailserver (z.B. qmail)

Als erster Mailserver wurde eine Umgebung simuliert wie sie z.B. vom Mailserver qmail unter Linux verwendet wird: Jedes Mail wird in einer eigenen Datei gespeichert.

Dazu wurden die Dateigrößen aus dem Bereich **100Byte bis 500kByte** gewählt und **Append als Transaction ausgeschlossen**. D.h. Dateien können nur erstellt, gelesen und gelöscht werden.

3. Mailserver MBOX (z.B. sendmail)

Als zweite Mailserver-Umgebung wurde das System von z.B. sendmail simuliert, das alle Mails eines Accounts in einer Datei speichert. An diese Datei werden neue Mails mittels Append angefügt.

Der Bereich der Dateigrößen wurde auf **100kByte bis 100MByte** gesetzt. Das **Verhältnis von Read zu Append wurde auf 1** gesetzt und **Create und Delete wurde ausgeschlossen**, weil diese Tests von einem laufenden Mailserver ausgingen und da das Erstellen resp. Löschen eines Accounts im Verhältnis zu den vorhandenen Accounts keinen Einfluss auf die Performance hat.

2.4 Software

- Benchmark-Software *Postmark*
Als Benchmarksoftware wurde Postmark 1.5 verwendet. Dies vor allem, weil Postmark in C geschrieben ist und sowohl unter Linux als auch Windows läuft. Somit liessen sich gut vergleichbare Tests durchführen.
Für jedes Einsatzgebiet wurden 3x5 Tests mit 5 verschiedenen Konfigurationen durchgeführt. Die Tests wurden durch Skripte sequentiell immer im selben Verzeichnis auf der selben Partition durchgeführt.
- Betriebssysteme *Windows und Linux*
Windows wurde in der Version 2000 verwendet und bei Linux kam Slackware 8.0 mit dem Kernel 2.4.5 zum Einsatz.

2.5 Verwendete Hardware

Als Testhardware kam ein Desktop-Rechner mit folgenden Spezifikationen zum Einsatz:

- Mainboard: ASUS A7M266D 760MPX
- CPU: 1xAMD MP 1.6+
- RAM: 768MB
- HDD: IDE, Fujitsu MPD3064AT, 6.48GB (Testpartition: 2GB)

3 Resultate

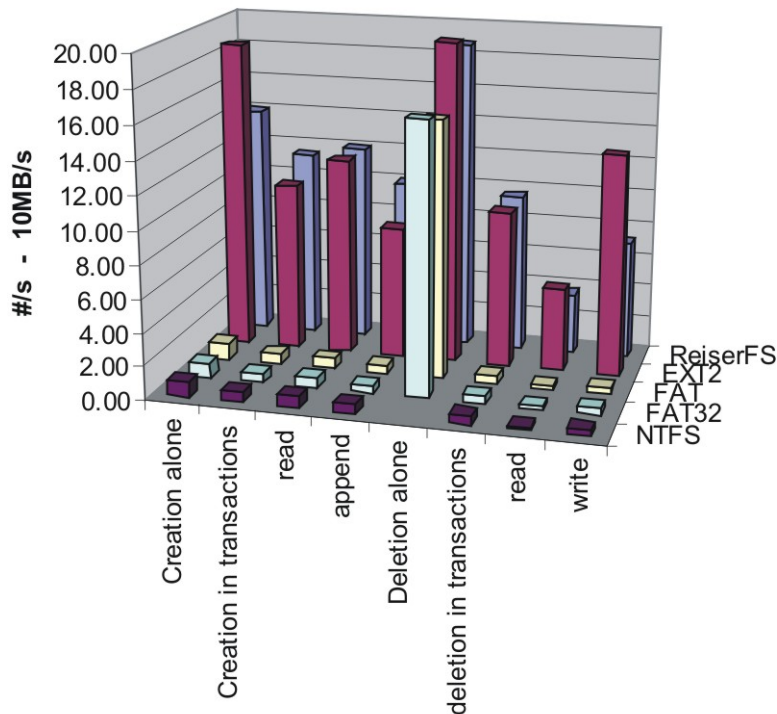
Die folgenden Grafiken zeigen folgende Daten:

- **X-Achse:**
 - *Creation alone*: Anzahl Dateien pro Sekunde, die sequentiell angelegt werden.
 - *Creation in transactions*: Anzahl Dateien pro Sekunde, die im Rahmen von Transactions angelegt wurden, also nicht sequentiell.
 - *read*: Anzahl Dateien pro Sekunde, die während den Transactions gelesen wurden, also nicht sequentiell.
 - *append*: Anzahl Dateien pro Sekunde, die während den Transactions an die bestehenden Dateien angehängt wurden.
 - *Deletion alone*: Anzahl Dateien pro Sekunde, die nach Abschluss aller Transactions sequentiell gelöscht wurden.

- *deletion in transactions*: Anzahl Dateien pro Sekunde, die während den Transactions gelöscht wurden, also nicht sequentiell.
- *read*: Totaler Datendurchsatz beim Lesen in MB/s oder 10MB/s.
- *write*: Totaler Datendurchsatz beim Schreiben in MB/s oder 10MB/s.

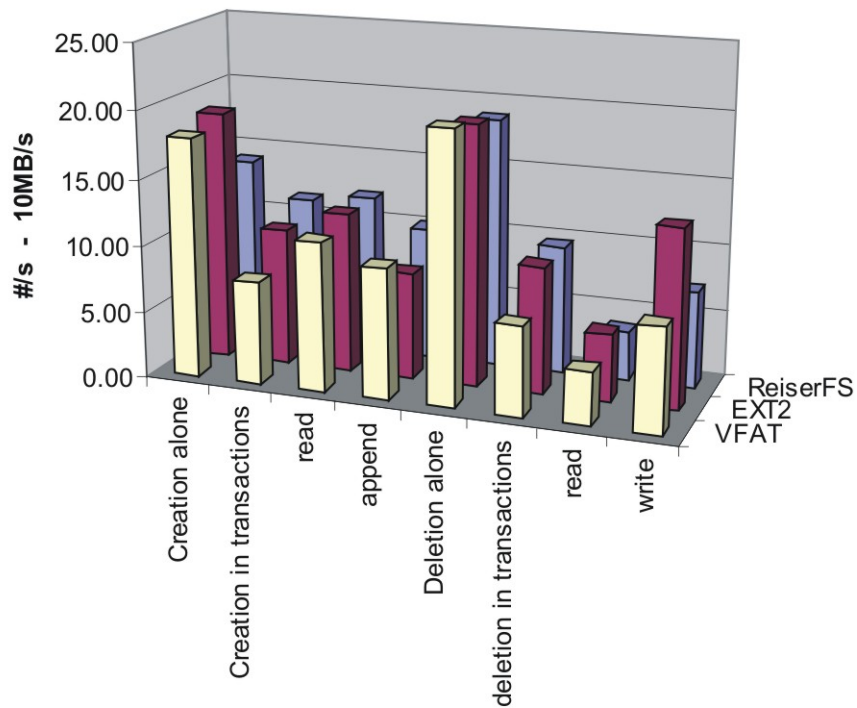
- **Y-Achse**: Dateisystem
- **Z-Achse**: Anzahl Dateien pro Sekunde resp. MB/s oder 10MB/s

3.1 Fileserver



Die Resultate zeigen einen enormen Unterschied zwischen den Dateisystemen unter Windows und denen unter Linux. Die selben Tests wurden zum Gegenvergleich noch auf einer anderen Festplatte durchgeführt, führten aber auf die selben Größenordnungen der Ergebnisse, vor allem die selben markanten Unterschiede zwischen FAT/NTFS und EXT2/ReiserFS.

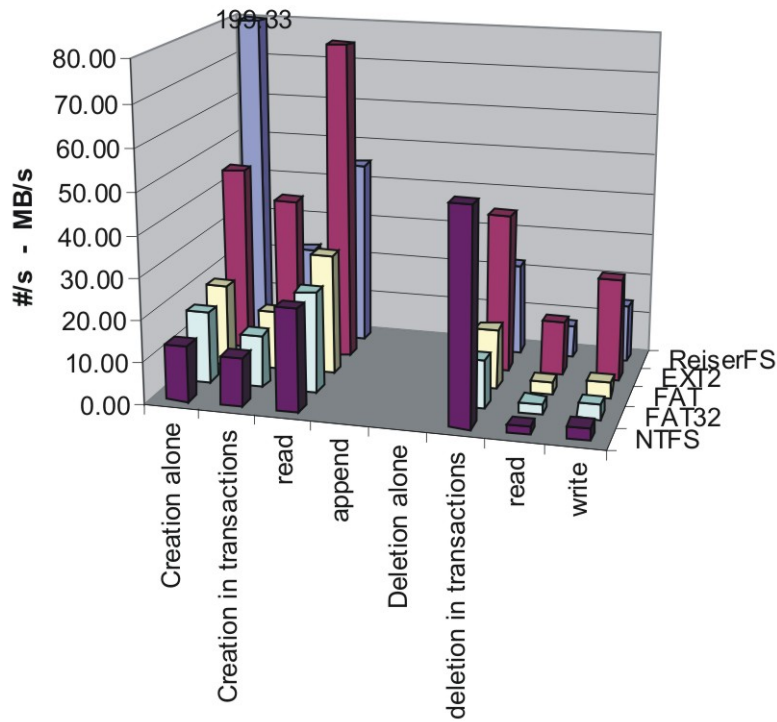
Als weiterer Test wurde die FAT32-Partition, die für die oben gezeigten Messungen verwendet wurde, unter Linux als VFAT-Partition gemountet und die selben Tests so ausgeführt. Dies ergab folgende Ergebnisse im Vergleich zu EXT2 und ReiserFS:



Alle mir bekannten und zugänglichen Faktoren wurden unter Linux und Windows identisch eingestellt, z.B. sind auch die Zufallszahlen, die zur Entscheidung über die Dateigrößen und die Transactions gebraucht werden, die selben unter Linux und Windows.

Für mich gibt es keine sinnvolle Erklärung dieses markanten Unterschiedes des Dateizugriffes auf die selbe FAT32-Partition unter Windows und unter Linux, ausser dass der FAT-Treiber unter Linux besser implementiert ist als unter Windows...

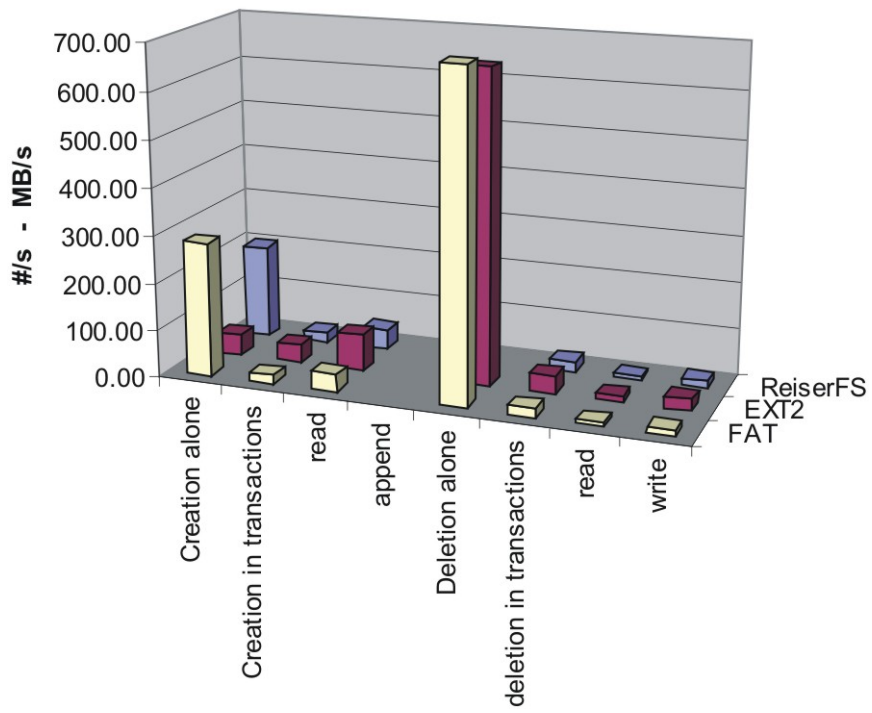
3.2 Mailserver



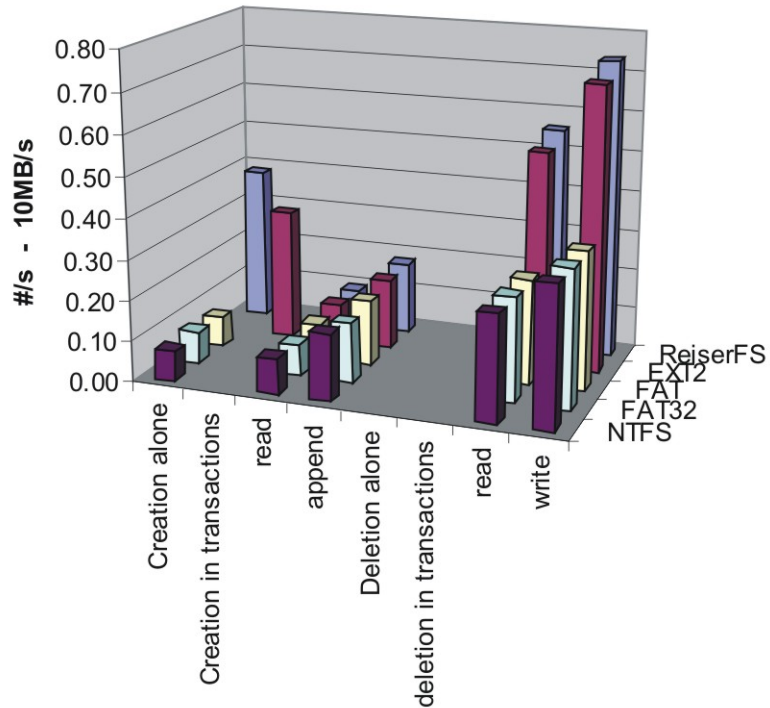
Auch hier liegen EXT2 und ReiserFS eindeutig vorne. Interessant ist, dass bei FAT, FAT32 und NTFS mit der Zunahme an Features die Geschwindigkeit nicht zunimmt, obwohl z.B. FAT32 eine Weiterentwicklung von FAT ist. Es scheint aber, als ob der Overhead durch die neuen Features nicht oder nur knapp durch neue Technologie gutgemacht werden kann.

Das selbe erkennt man bei EXT2 und ReiserFS. Obwohl letzteres das neuere Dateisystem ist, vermag es wegen der Vielzahl der neuen Features wie Journaling, Encryption (hier nicht aktiviert) nicht an die Leistungen des einiges älteren EXT2 herankommen.

Zum Vergleich auch hier die Werte für FAT32 unter Linux getestet:



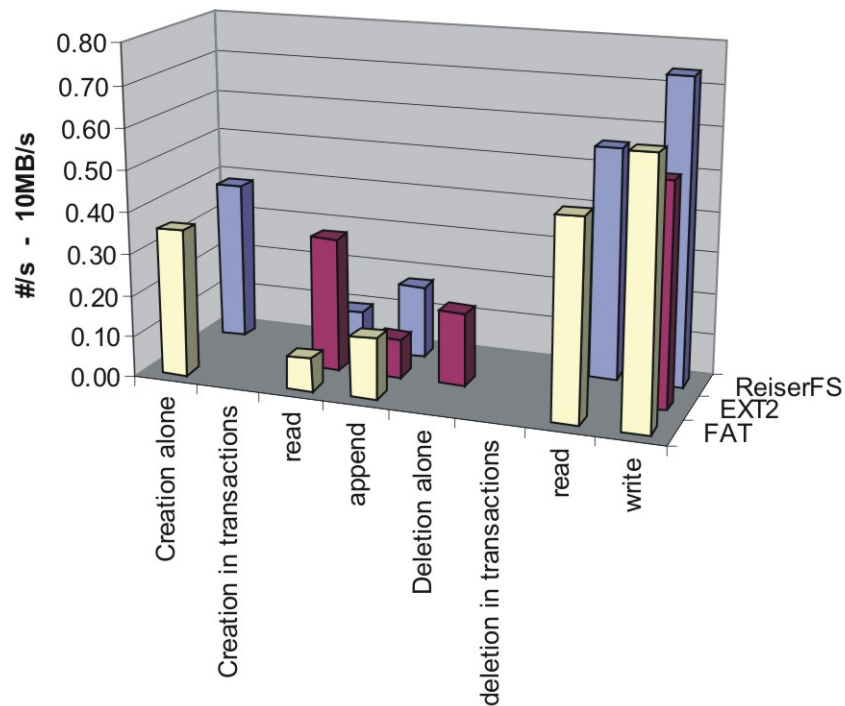
3.3 Mailserver MBOX



Die Aussagekraft dieser Daten ist wohl die kleinste dieser Arbeit, da die Tests schwierig durchzuführen waren. Die Dateianzahl war wegen ihrer Grösse stark beschränkt und auch die Transactions hochzuschrauben war nur bedingt möglich. Zum einen, weil diese ja die Dateien nochmals vergrösserten (append) und zum zweiten, weil diese Operationen teilweise sehr lange dauerten.

Ansonsten zeichnet sich in etwas das selbe Ergebnis ab wie bei den anderen beiden Fällen.

Zum Vergleich auch hier die Werte für FAT32 unter Linux getestet:



4 Schlussfolgerungen

Um pure Performance zu erhalten muss man auf einiges an Funktionalität verzichten. Dies ist ein allgemeines Prinzip, das sich in der Informatik immer wieder finden lässt und es scheint auch auf Dateisysteme zuzutreffen.

Was diese Tests natürlich nicht miteinbeziehen können, ist das ganze Umfeld der Dateisysteme und deren Handling. Z.B. bemerkt man nach einem Absturz eines ReiserFS-Dateisystems einiges an Performanceunterschied zu z.B. EXT2, weil letzteres die ganze Festplatte durchchecken muss um wieder ein intaktes Dateisystem sicherzustellen. Daneben muss ReiserFS dank seinem Journal nur die nötigen Dateien checken.

Auch Aspekte wie Sicherheit können in dieser Arbeit nicht miteinbezogen werden. Muss z.B. bei einem FAT-System die Verschlüsselung oder nur schon der Zugriffsschutz durch zusätzliche Software sichergestellt werden, verliert man dadurch einiges an Speed und eas-of-use im Gegensatz zu z.B. NTFS, welches solche Dinge bereits implementiert hat.

Nichts desto trotz vermag diese Arbeit eine Aussage über mögliche Einsatzgebiete aufzuzeigen und die Performance von "out-of-the-box" Dateisystemen zu vergleichen, was einem bei der Wahl des richtigen Dateisystems durchaus helfen kann.

Contents

1	Einleitung	1
2	Tests	1
2.1	Dateisysteme	1
2.2	Betrachtete Faktoren	2
2.3	Einsatzgebiete	3
2.4	Software	4
2.5	Verwendete Hardware	4
3	Resultate	4
3.1	Fileserver	5
3.2	Mailserver	7
3.3	Mailserver MBOX	9
4	Schlussfolgerungen	10