Filesystems in Linux

A brief overview and comparison of today's competing FSes.

Please save the yelling of obscenities for Q&A. **;-)**

Files and Directories

- Files and directories allow data to be
  - Grouped - files
  - Ordered hierarchically - directories
- Filesystems
  - Perform 2 functions, broadly speaking
    - Manage storage
    - Allow user to access data in a certain way
  - Evolution
    - Storage techniques have improved over the years
      - Database methodologies have seeped in
    - Users still have a 20-year old view
      - Files are, at best, random-access byte-streams

Filesystems Today

- Good ol' Ext2
- Ext3 is Ext2 with journaling added
- Reiser3 is making major inroads
- Reiser4 will be out eventually
- JFS from IBM
- XFS from SGI
- NTFS from...dial M for Microsoft
- All but Ext2/3 use B-Trees or variants
  - More complex
  - Faster
- All now provide ACLs

Journaling

- From databases
- Keep a log of “transactions”
  - All operations are atomic
  - Logging all data is expensive
  - Log metadata
- Crash? No problem
  - Replay log
  - FS is consistent
- Filesystem corruption is far less likely
- Startup time after a crash is much lower

Other stuff

- So how do I get this stuff working
  - Gentoo – comes with Ext2/3, Reiser3, XFS and JFS support (shameless plug)
  - Slackware – comes with Ext2/3, Reiser3 support
  - RedHat / Fedora Core– You're stuck with Ext3
  - Mandrake – At last count had Ext2/3, Reiser3, and JFS. Anybody know more?

Ext2

- Written by Theodore T'so
- Default for most distros
- Simple – no complex algorithms
- Rock-solid
- But not journaled
  - Fsck takes *ages*
- Reference implementation of an FS in Linux
- Use Ext3 over Ext2

Ext3

- Ext3 = Ext2 + Journaling
  - Nothing new / fancy
- Used to journal data and metadata
  - Expensive
  - Now journals metadata alone
- Migrating between Ext2 and Ext3 is a breeze
  - `tune2fs -j /dev/hdaX` to make hdaX journaled
  - Just mount as ext2 and the journal is ignored

More Ext3

- Use `tune2fs -J` to tune your journal
  - Set size, put journal on another device
- Advantages
  - Simple
  - Extremely robust
- Disadvantages
  - Simple
  - Not the first choice for high-performance machines
- Use it if robustness precedes performance

Reiser3

- Developed by Hans Reiser of Namesys
- Written from scratch
- Emerging champion, used by
  - Gentoo
  - Suse
  - Lindows
- Enhanced B+trees + assorted algorithms
- Tail packing

Tail Packing

- Files stored in *blocks*, usually 4Kb
  - 1Kb file takes 1 block
  - 41Kb file takes 11 blocks
  - The remaining 3Kb is wasted
  - These are called tails
- Reiser3 takes tails and packs them into a single block
  - Saves space
  - Repacking overhead when appending data
- Typical savings of about 5% (YM*WilN*)
- Don't want it? Mount with notail

The Reiser3 Verdict

- Advantages
  - Very fast
  - Relatively stable
  - Tail packing for the really stingy ;)
- Disadvantages
  - Tail repacking can be expensive
  - Not as solid as Ext3, supposedly
    - I have no complaints
- Use this at home, but not on your servers

JFS

- Developed by IBM for AIX
- Meant for high-end machines
  - High performance
  - Reliability
- Different block sizes supported
  - 512, 1024, 2048, 4096
- Directories stored in 2 ways
  - Small directories – in the block
  - Large directories – B+ trees
- Extents for large files

More JFS

- Dynamic inode allocation
  - Won't run out of inodes, or need to preallocate
- Online resizing
  - You can resize the *partition*
  - Good for Logical Volume Management
- Sparse files
  - Does any application support these?
- 64-bit ready

Still More JFS

- Advantages
  - Reliable
  - Good for large files
  - Only one to support online resizing
- Disadvantages
  - More tuned for large files and servers
    - Not too good for home use

XFS

- Made by SGI for Irix
- Industrial-strength FS
  - High-performance
  - Scalable – good for *huge* filesystems
- Extent-based - good for large files
- Extensive use of B+trees for speed
- Also a 64-bit FS
- Caches heavily
  - Minimize disk I/O
  - Great speed boost

Use XFS?

- Advantages
  - Good for very large files and filesystems
  - Fast
- Disadvantages
  - Not quite as reliable as Ext3
  - Gentoo says it's flaky
  - Didn't work off my 2.6.5 kernel
    - No numbers in the benchmarks
- Use it if you've got a big server
  - With large files
  - Needing good throughput

Reiser4

- *Gleeful cackle*
- Currently in beta
- Should get into the 2.6 kernel soon
- Many new ideas, concepts
- Fast!!!
- But first...

Digressing...

- File system view for the user hasn't changed in 20 years
  - Very low level
  - Just a stream of bytes
  - Metadata is minimal
- Restrictions
  - Small files are **expensive**
  - Workarounds mean compromises
    - /etc/passwd
  - Files cannot have data added into the middle

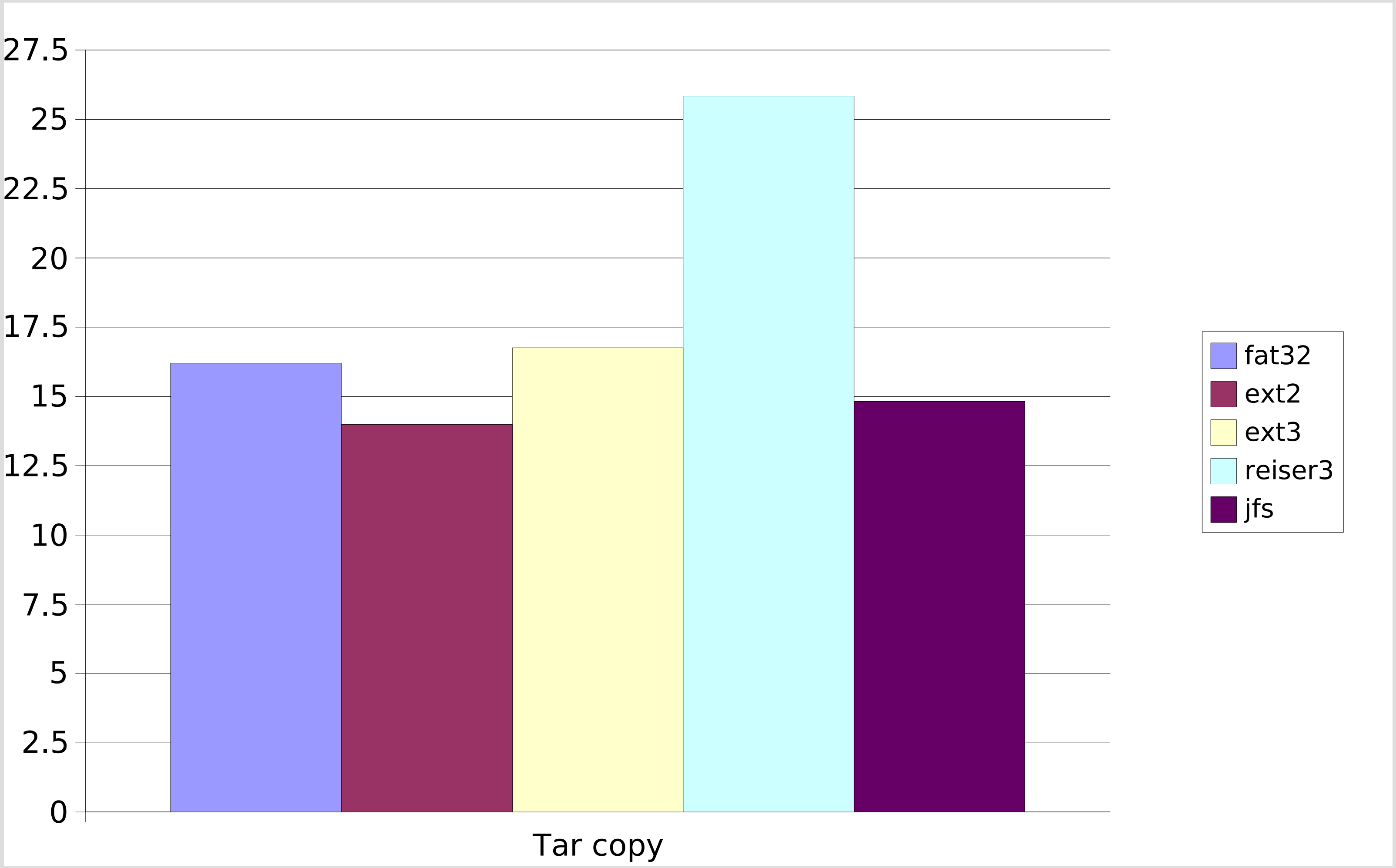
...Back to Reiser4

- Layering
  - Storage layer
    - Optimized for speed
    - Similar to the old implementation
  - Semantic layer
    - Offers a new range of options
    - Richer ways to access the data
- Plugins
  - Allow all sorts of extensions to the FS, *particularly* at the semantic layer
  - Example – cd into files for psedo-files
- WinFS intends to do similar things, in due time

Getting Down 'n Dirty

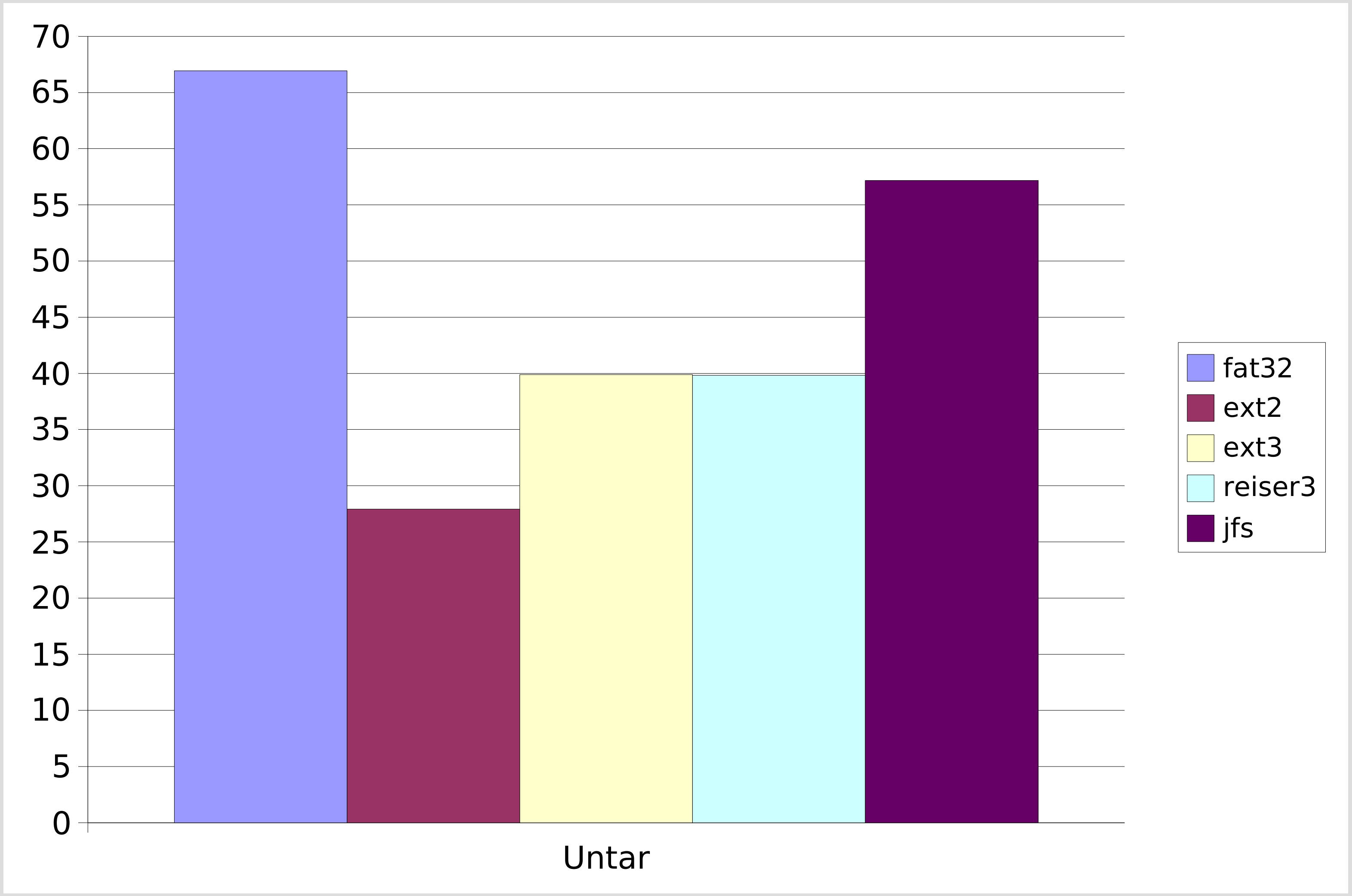
- Disclaimer
  - Take with a spoon of salt
  - Not very “real-world”
- The benchmark
  - Copy a tar file (Linux kernel tree, ~245MB)
  - Decompress tar
  - Delete tar
  - Recursive listing of kernel tree
  - Delete tree
- Fat32 numbers only for reference
  - No M$ bashing intended. ;)
  - No NTFS

Tar Copy

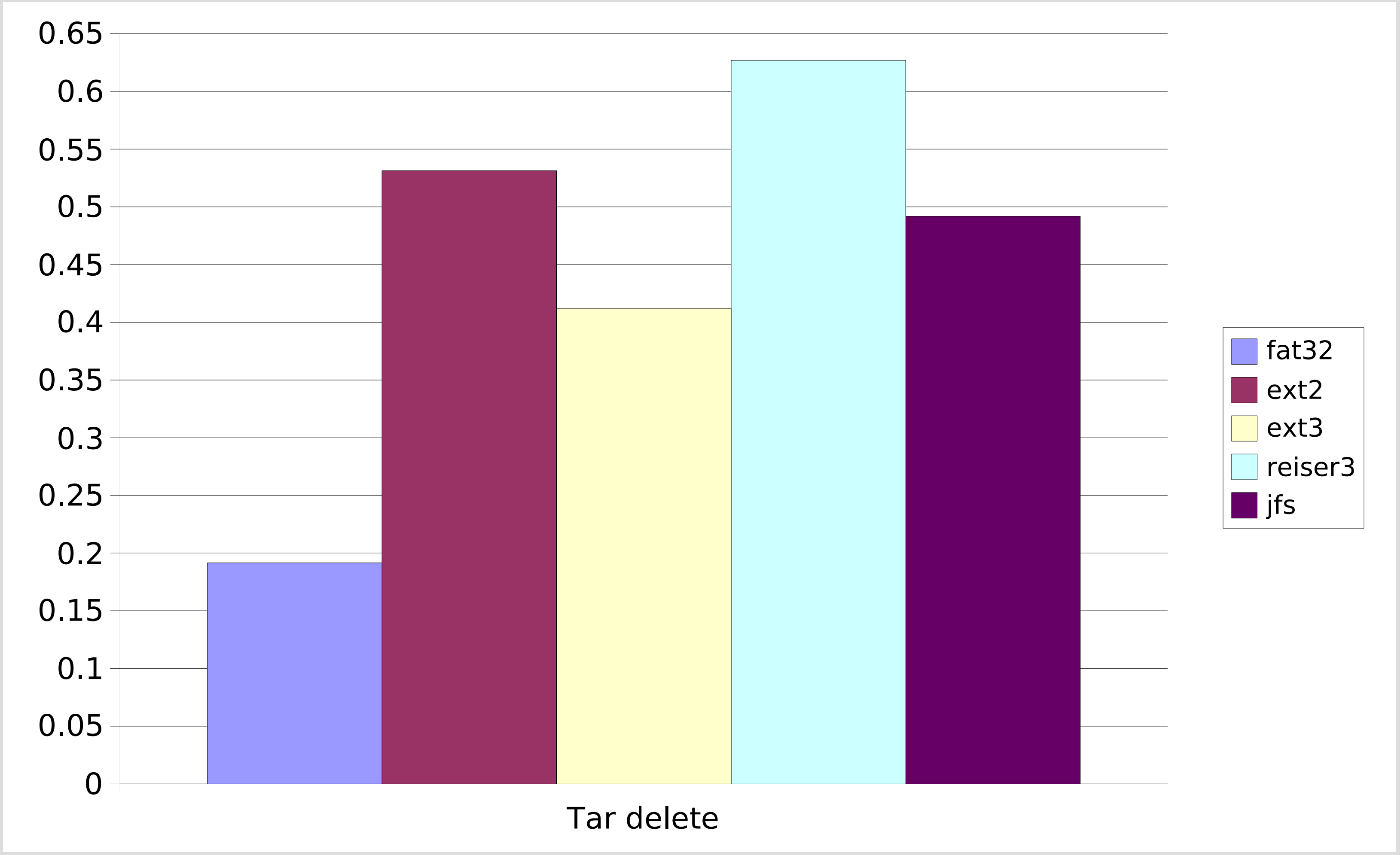


27.5
25
22.5
20
17.5
15
12.5
10
7.5
5
2.5
0
Tar copy
fat32
ext2
ext3
reiser3
jfs

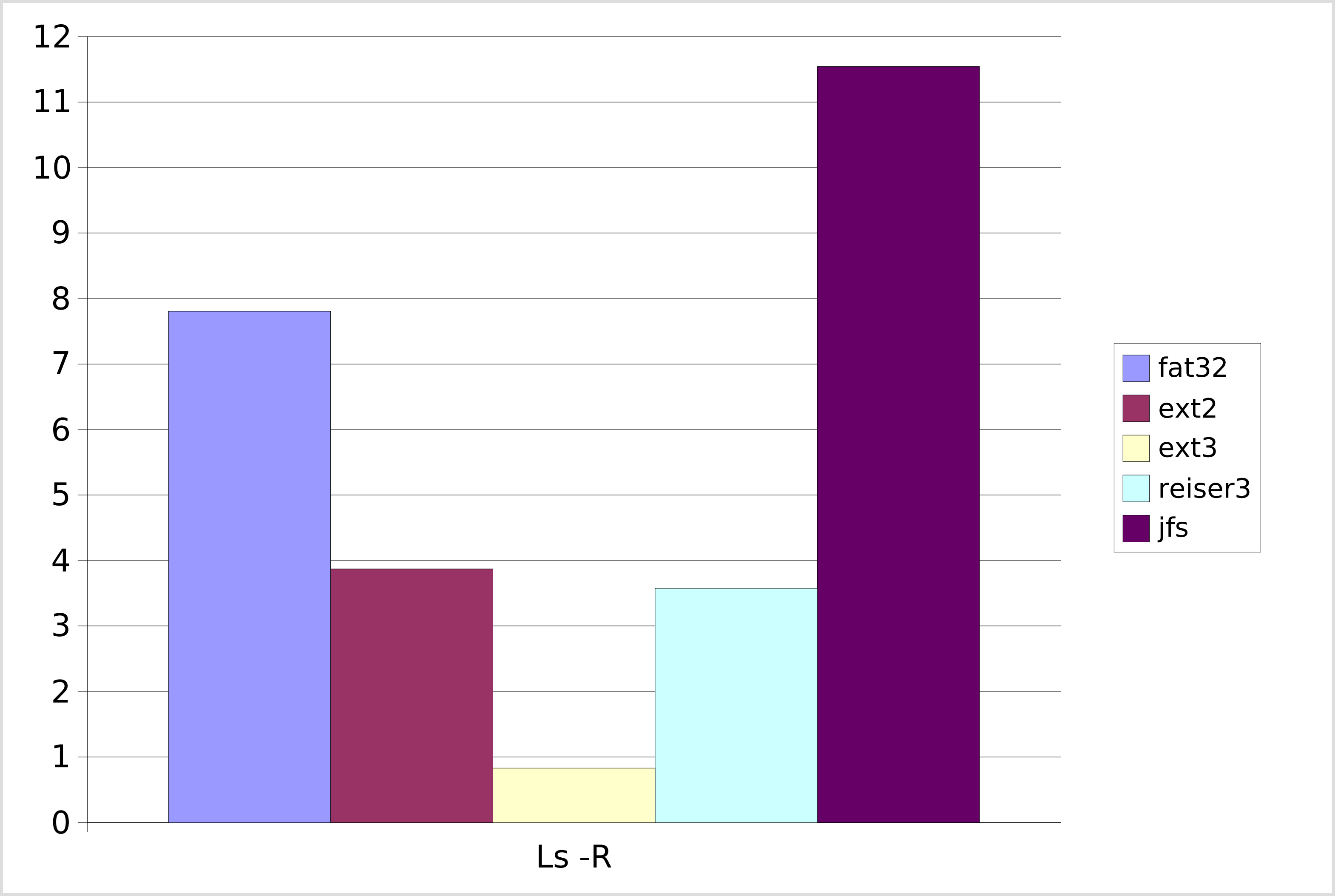
Untar


70
65
60
55
50
45
40
35
30
25
20
15
10
5
0
Untar
fat32
ext2
ext3
reiser3
jfs

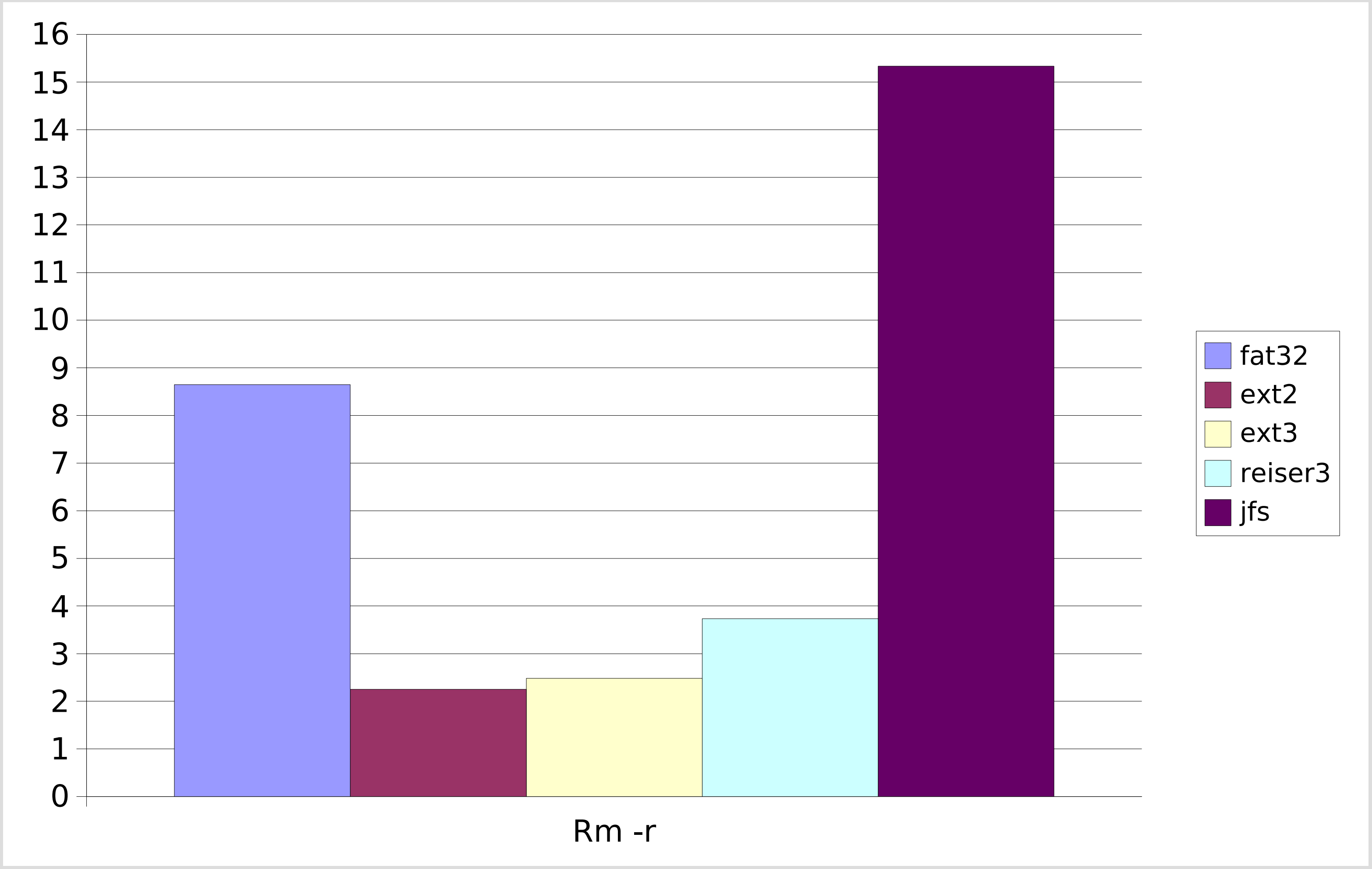
Delete Tar


0.65
0.6
0.55
0.5
0.45
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
Tar delete
fat32
ext2
ext3
reiser3
jfs

Recursive Listing



Recursive Deletion



The Verdict

- Don't use Fat32
- If you want to use Ext2, use Ext3
- Reiser can be faster under load
- JFS and XFS
  - Supposed to be industry-grade
  - Not performing all that well
    - More research before commitment