

# Wayback: A User-level Versioning File System For Linux

**Brian Cornell,**  
Peter Dinda, and Fabián Bustamante

Computer Science Department  
Northwestern University

# Really excited last night

- Great idea for this presentation
- 3d rotating slides...
- Woke up this morning, Oh no!
- Too excited: no backup copy
- No RCS<sup>1</sup> or CVS<sup>2</sup> check-in
- Lucky, Wayback was there!
- Reverted back to yesterday afternoon, and here we are, aren't you glad?

1: F. Tichy, 1980.

2: B. Berliner, 2001.

# Outline

- Overview
- Related work
- How it works
  - User's perspective
- Behind the scenes
  - System's perspective
- Design decisions
- Performance
- Future work

# Wayback: Automatic Versioning

- Automatic: Versions files and directories without user interaction
- Universal: Versions files of all types
- Comprehensive: Never loses data
- Generic: Works over any directory on any base FS
- User-space: Implemented in user-space

Free (GPL): <http://wayback.sourceforge.net>

# Related Work

- VMS operating system [K McCoy, 1990]
  - Version on close
- Elephant [D Santry, et al, 1999]
  - Exploit massive disks
- VersionFS stackable file system [Muniswamy-Reddy, et al, 2004]
  - Contemporaneous to this project
  - Implements different features
- Cedar [D. K. Gifford, et al, 1998], 3DFS [D. G. Korn, et al, 1989], CVFS [C. A. Soules, et al, 2003]

# How does it work?

- Three simple steps
- Remount folder
- Edit and work as normal
- Revert or extract old versions

# Remount folder

- Run wayback executable to remount
  - `wayback /mnt/data /home/brian/Projects`
- Use mount.wayback script to simplify and mount for all users
  - `mount.wayback /mnt/data /home/brian/Projects`
- Files still at `/mnt/data`, but if you edit them at `/home/brian/Projects`, they are versioned

# Work normally

- Edit files with any editor
- Modify binary files too
- Move, rename, delete files
- Edit directories as much as you want
- Everything you do is recorded and remembered



# Revert or extract versions

- List the versions of a file
  - `vstat Wayback.sxi`
- Revert a file to an old version
  - To a time: `vrevert -d 15:00:00 Wayback.sxi`
  - To a date:  
`vrevert -d 2004:06:29:15:00:00 Wayback.sxi`
  - To a numbered version as listed by `vstat`:  
`vrevert -n 5 Wayback.sxi`
- Revert a directory to a date/time
  - `vrevert -d 2004:06:29:15:00:00 Documents`
- Note: Even reverting can be undone

# Revert or extract versions

- Extract versions to a different file
  - Same format as `vrevert` with a destination
  - `vextract -d 2004:06:29:15:00:00 Wayback.sxi`  
`GoodWayback.sxi`
- Delete the versioning if you're sure you don't need it anymore
  - `vrm Wayback.sxi`
  - All versioning info for the file is permanently deleted
  - File is not deleted. To delete file and versioning, first delete file, then versioning.

# Behind the Scenes

- FUSE<sup>1</sup> module sends all FS calls to Wayback
- Everything kept track of transparently
- Hidden log files, one per file or directory
- File operations recorded
- Directory operations recorded

1: M. Szeredi,, *Filesystem in USEr space*,  
<http://sourceforge.net/projects/avf>

# Log Files

- Every versioned file has a hidden log file
  - `Wayback.sxi. versionfs! version`
- Every directory has a hidden catalog
  - `Documents/. versionfs! version`
- Logs created as soon as needed
- Logs omitted from directory listing by Wayback
- Logs in binary custom format

# File Operations

- On every write or truncate, entry appended to log
- Data needed to undo write or truncate is written

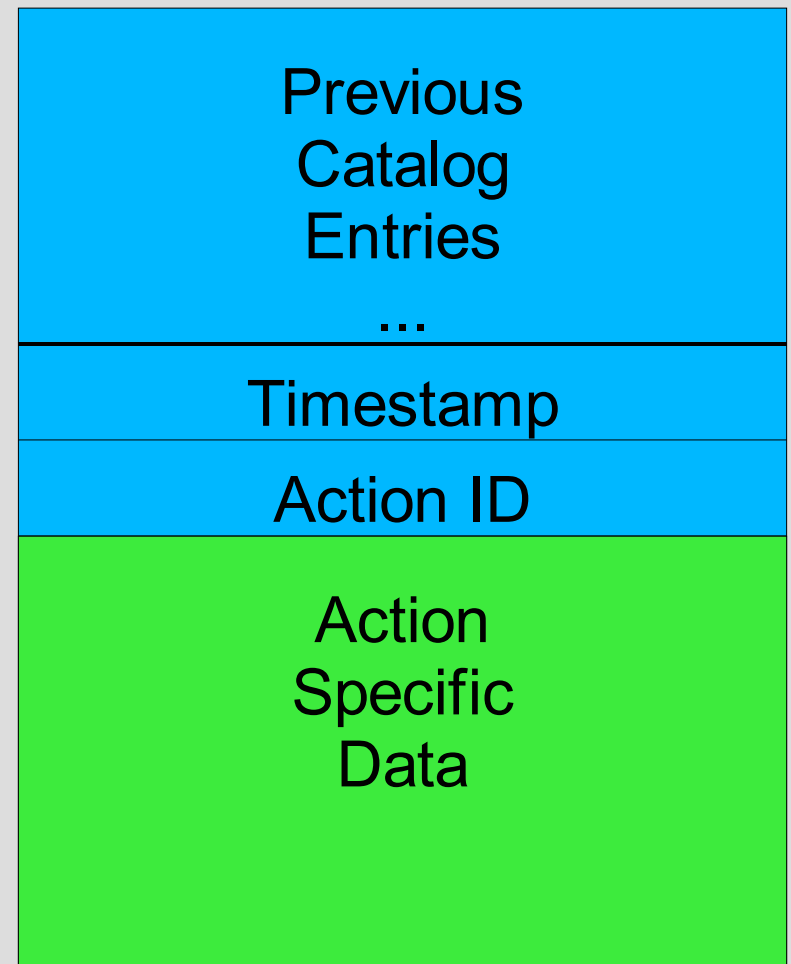
Log file



# Directory Operations

- When files are changed in a way that changes their directory entries, directory catalog is updated
- Every operation writes timestamp and action ID
- Specific data per operation

## Catalog file



# Dir Ops: Create, Mkdir

- Just needs to know name so it can be deleted on revert

Length of  
Name

File or  
Directory  
Name

# Dir Ops: Rm

- First, file is truncated to 0 bytes
  - File version log now has backup
- Record file attributes
  - Mode, owner, times
- Record filename
- Record destination for symbolic link

Attributes Contents:

Mode	UID	GID
Access Time	Modified Time	Create Time

Size of Data
Attributes
Filename [Link dest]



# Dir Ops: Rmdir, Rename

- Rmdir actually does rename and omits from directory listing
  - Don't want to lose logs for files in directory
- Record old and new names

Length of  
Names

Old and  
New  
Names

# Dir Ops: Chmod, Chown, Utime

- Just need to know the filename and the old attributes

Size of Data

Attributes

Filename

# Design Decisions

- **User** vs kernel level
  - User: Remount any base FS
  - User: Development easier, more stable
  - Kernel: Faster
- **Undo** vs redo log
  - Undo: Everything always retained as long as current version is not lost
  - Undo: No overhead of initial file copy
  - Redo: Files recoverable even if the current version is lost

# Design Decisions

- **FUSE** vs other user level modules
  - It's modern and actively developed
  - It works with kernels 2.4 and 2.6
  - It gives complete but simple access to FS operations
  - It's written in C (with which we are more familiar)

# Design Decisions

- **Comprehensive versioning** vs others
  - Record every operation that ever happens
  - Periodic: Choose time-granularity of versioning
  - Version on close: Assume one logical operation per file open
- **Individual log per file** vs central log
  - Individual: No central point of failure
  - Individual: Simple, logs stay with files
  - Central: Doesn't consume inodes as fast

# What's the cost of Wayback?

- 3 performance tests
  - Bonnie<sup>1</sup>: Large file reads and writes
  - Andrew<sup>2</sup>: Typical use and compilation estimation
  - RCS: Versioning comparison with RCS
- Tested with ext3, FUSE, and Wayback

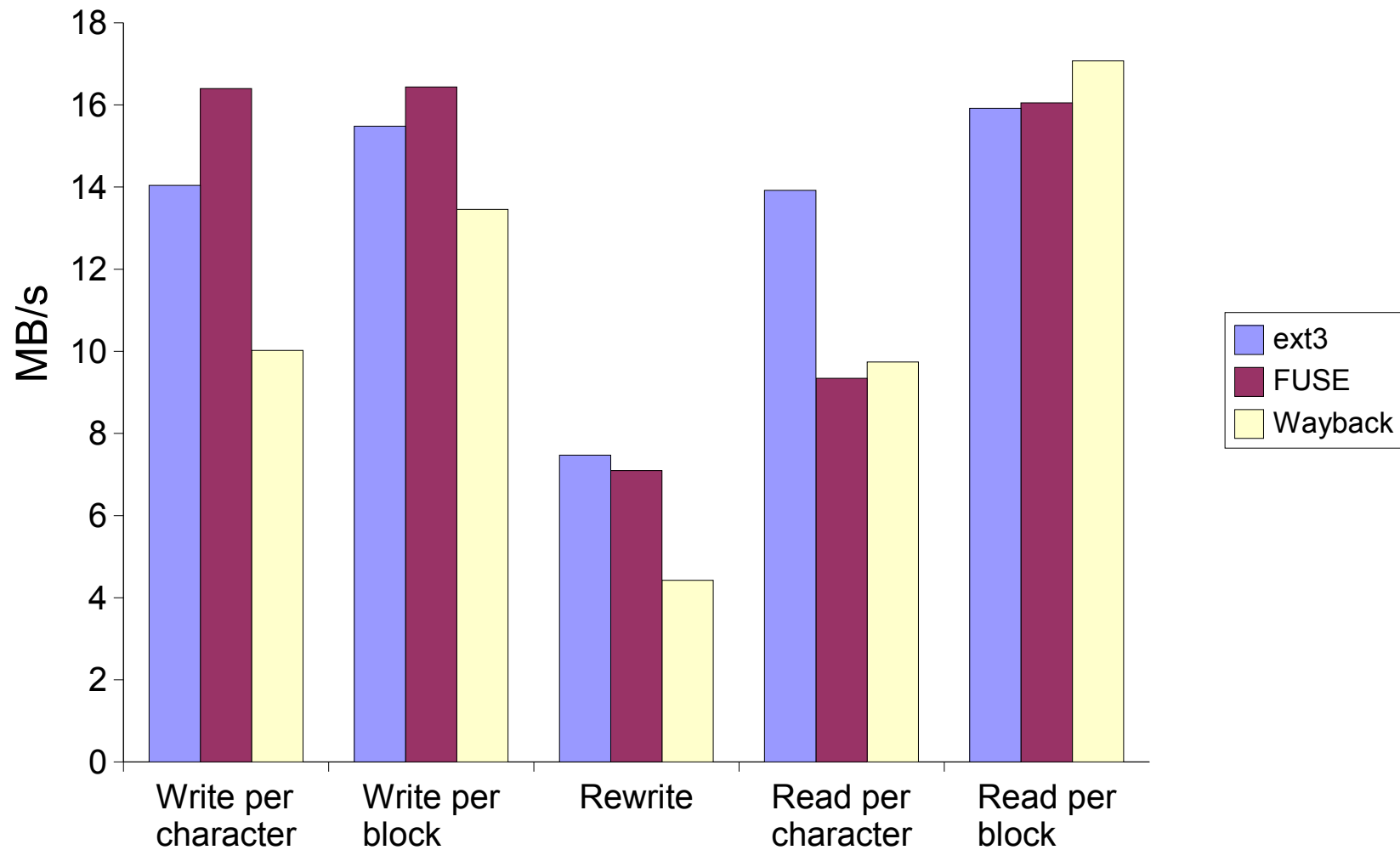
1: T. Bray, <http://www.textuality.com/bonnie>

2: J. Howard, et al, 1988.

# Test machines

- Tests on 3 machines
  - Machine A: Normal
    - Athlon XP 2400
    - 512 Mb memory
    - 2.5 in notebook hard drive
  - Machine B: Slow disk
    - Pentium IV 2.2 Ghz
    - 512 Mb memory
    - USB 1.1 hard drive
  - Machine C: Slow processor
    - Celeron 500 Mhz
    - 128 Mb memory
    - 2.5 in notebook hard drive

# Bonnie: 30-70% max overhead

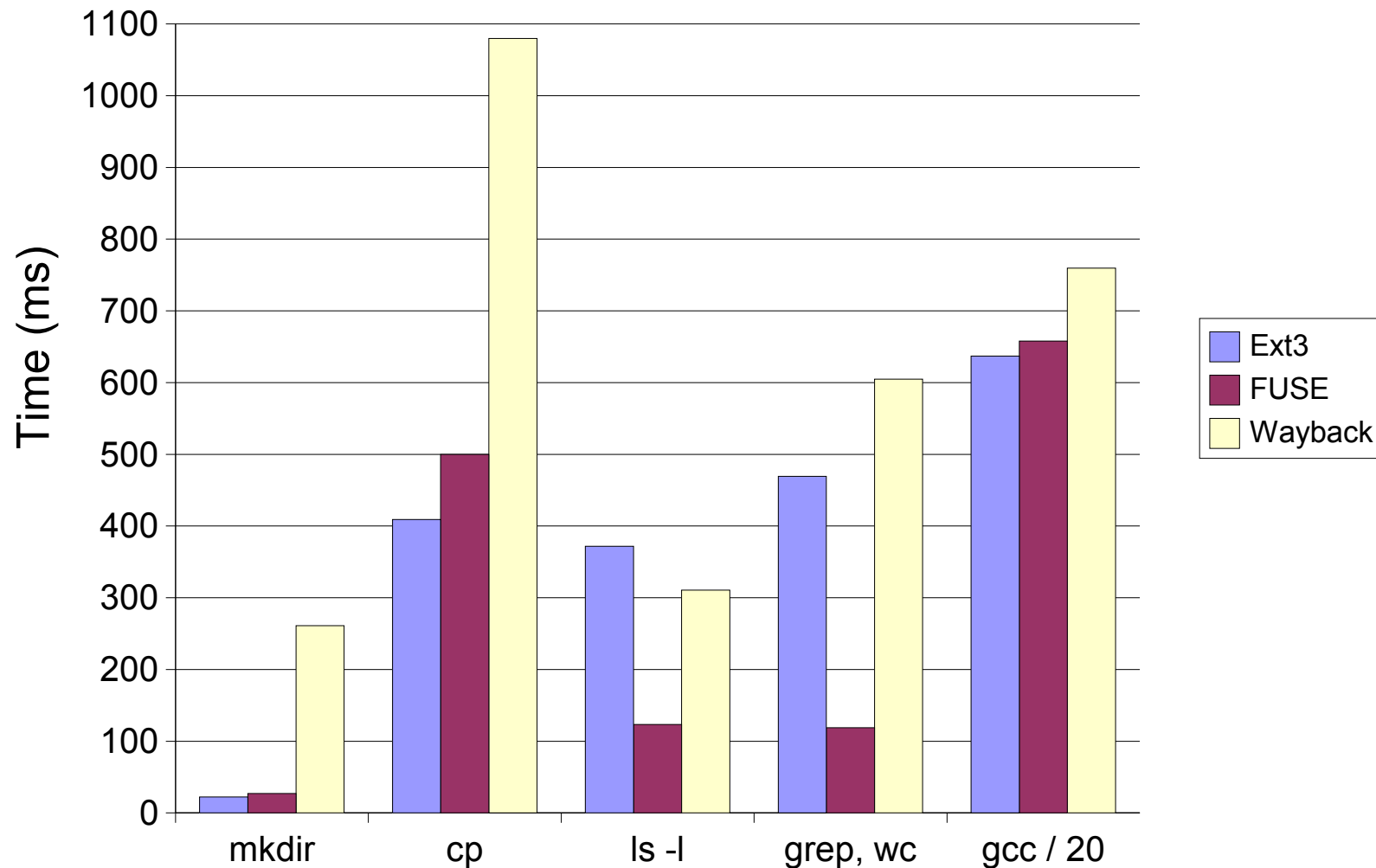




# Modified Andrew Benchmark

- Andrew uses Makefile to run five phases with Sun graphical source code
- Modification uses Perl and Linux `wm`
- Five phases:
  - 1: Directory creation (`mkdir`)
  - 2: Copying files (`cp`)
  - 3: Stating files (`ls -l`)
  - 4: Reading files (`grep` and `wc`)
  - 5: Compilation (`gcc`)

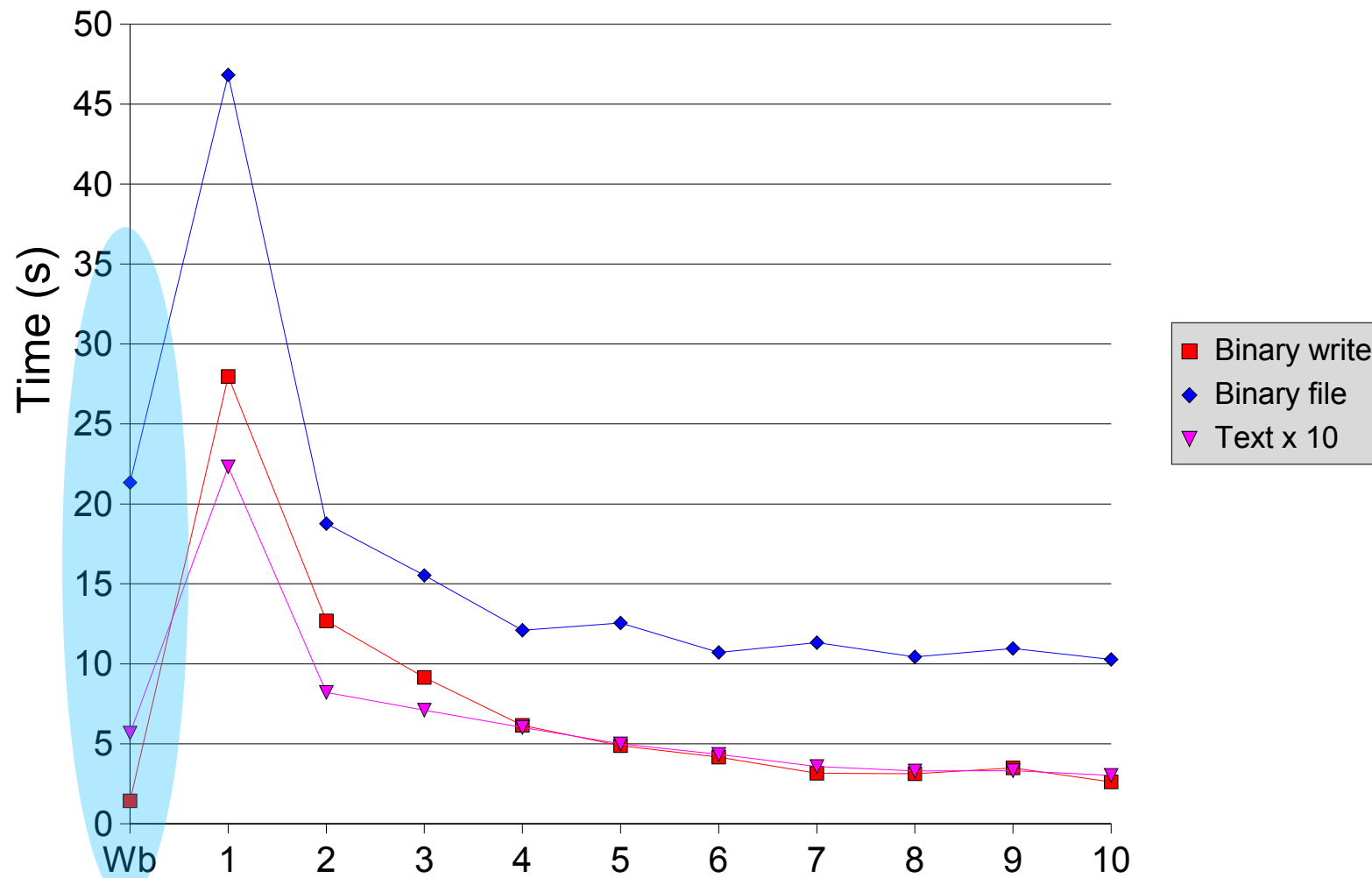
# Andrew: Compile 15% slower



# RCS Comparison

- 3 traces
  - Binary write:
    - Random small changes with write system call
  - Binary file:
    - Random small changes, but rewrite whole file
  - Text:
    - Random changes to lines in text file
- 11 configurations
  - Wayback
  - RCS checkin after every k changes,  $k=1..10$

# RCS: Wayback Faster



# Future Work

- Log compression
  - Remove unnecessary granularity
- Garbage collection
  - Remove old versions
- Limiting
  - Limit number or size of versions
- Customizable versioning
  - Filter to only version some files, not others
- Migration of virtual machines
  - Undo/redo to sync virtual machines

# Summary

- Don't lose important data!
- RCS, CVS, etc require user action
- Wayback automatically versions
  - Remount directory with versioning
  - Edit files as normal
  - Writes are written as undo information
  - Easy to revert or extract to any time
- Wayback faster than RCS even excluding user time

# Get it!

It's free! (Gnu General Public License)

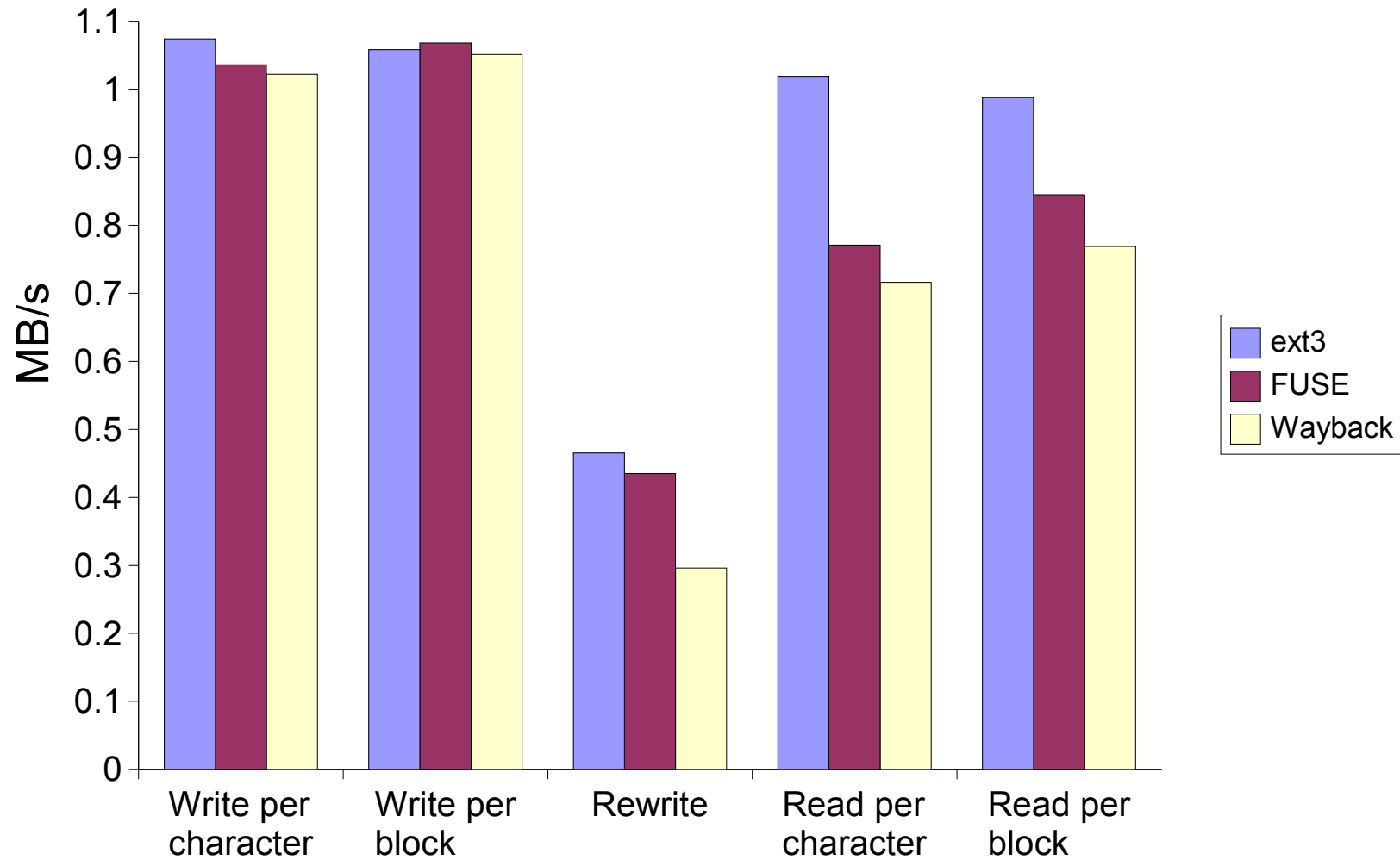
<http://wayback.sourceforge.net>

# RCS Size Comparison (MB)

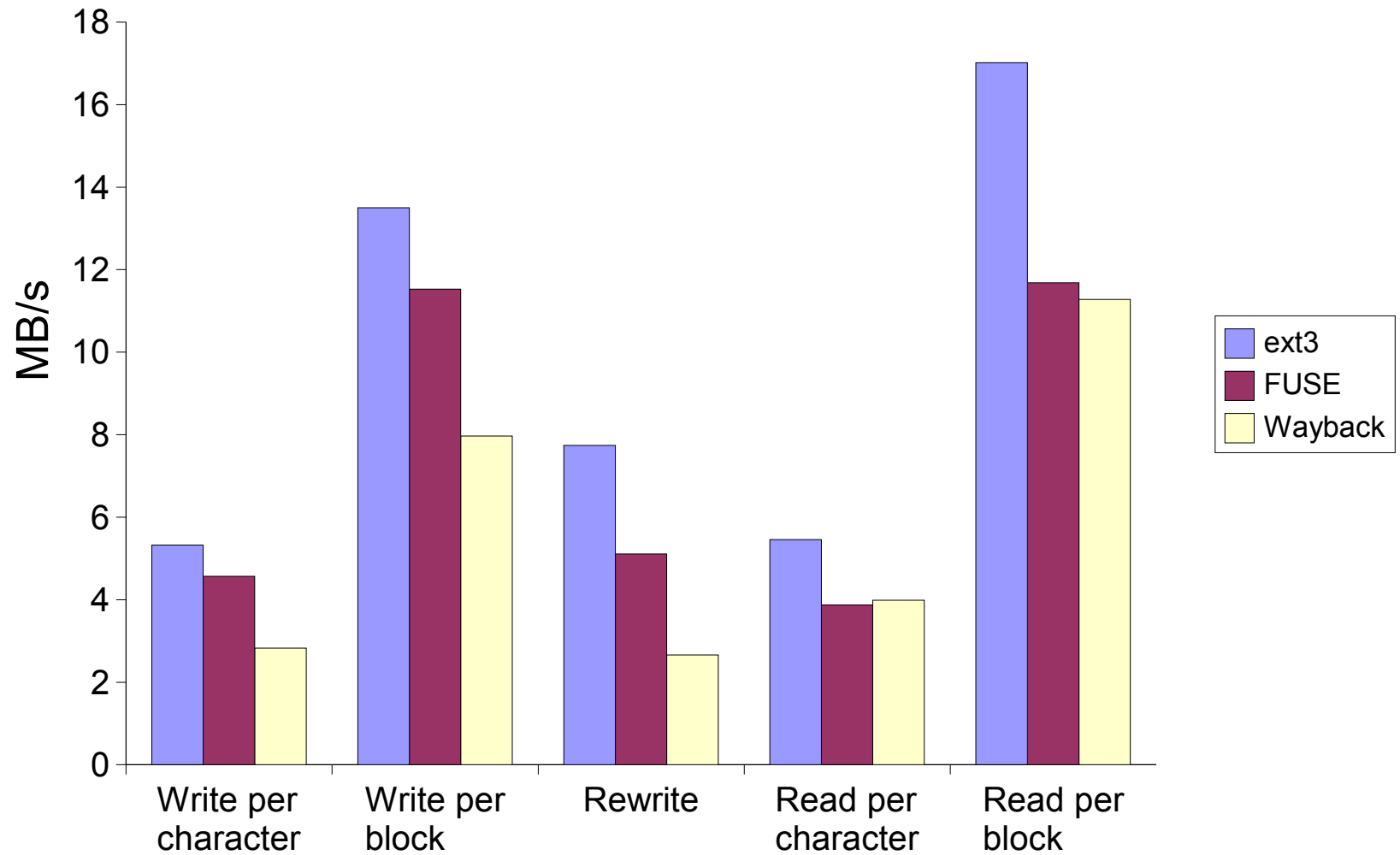
File Type	Binary Write	Binary File	Text
Wayback	1157456.4	106347428.0	2182218.0
RCS Period 1	2242325.4	3856521.8	101062.2
RCS Period 2	2237020.7	3779180.4	96134.2
RCS Period 3	2233854.1	3731427.8	94336.4
RCS Period 4	2234384.4	3719578.2	93597.1
RCS Period 5	2233716.4	3700853.4	93095.3
RCS Period 6	2227924.3	3621657.1	92375.5
RCS Period 7	2230300.3	3635107.2	92321.2
RCS Period 8	2227552.2	3590195.5	91960.0
RCS Period 9	2231124.4	3625548.8	92060.8
RCS Period 10	2232218.7	3629717.4	92045.2



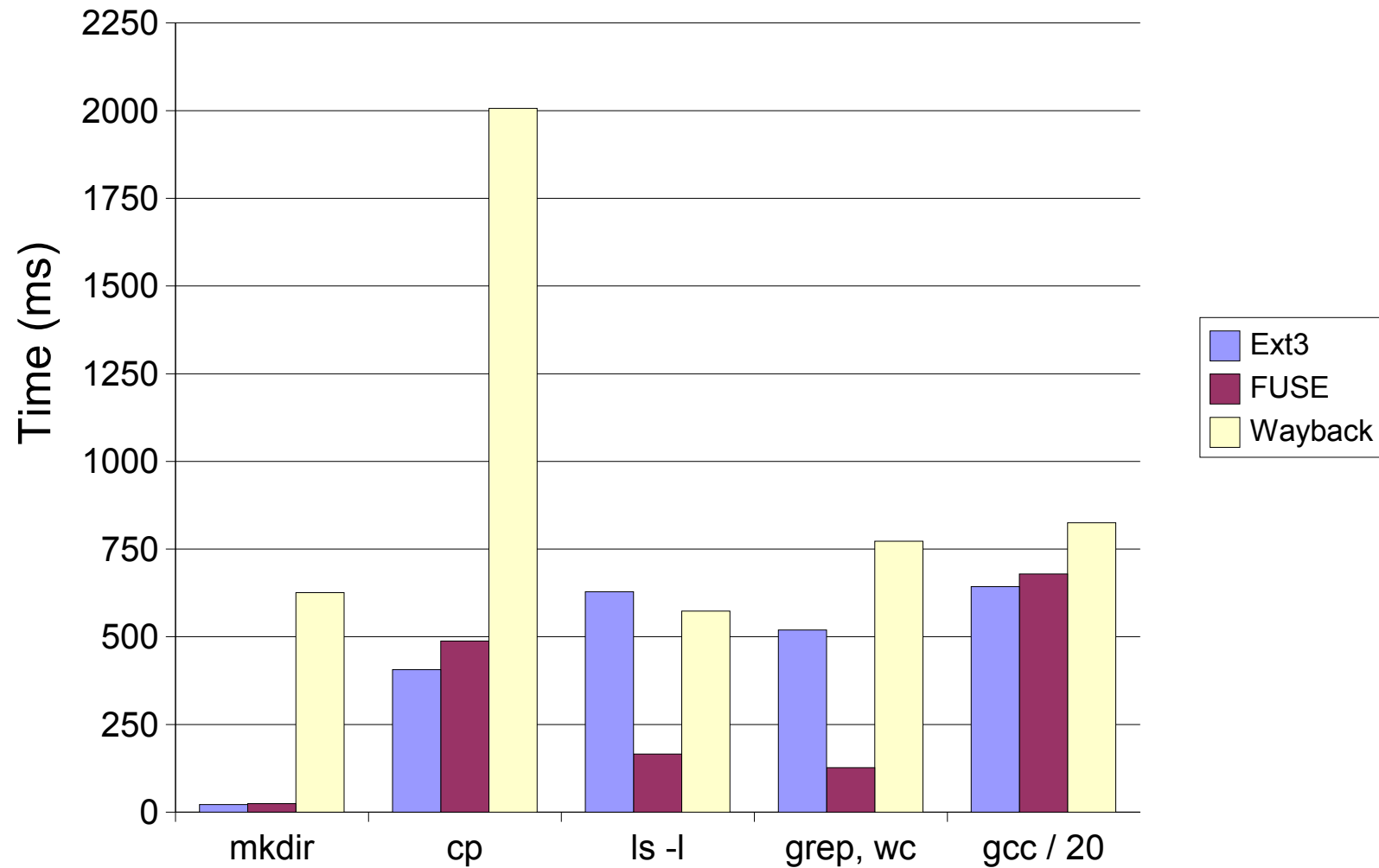
## Slow Disk: Bonnie Performance



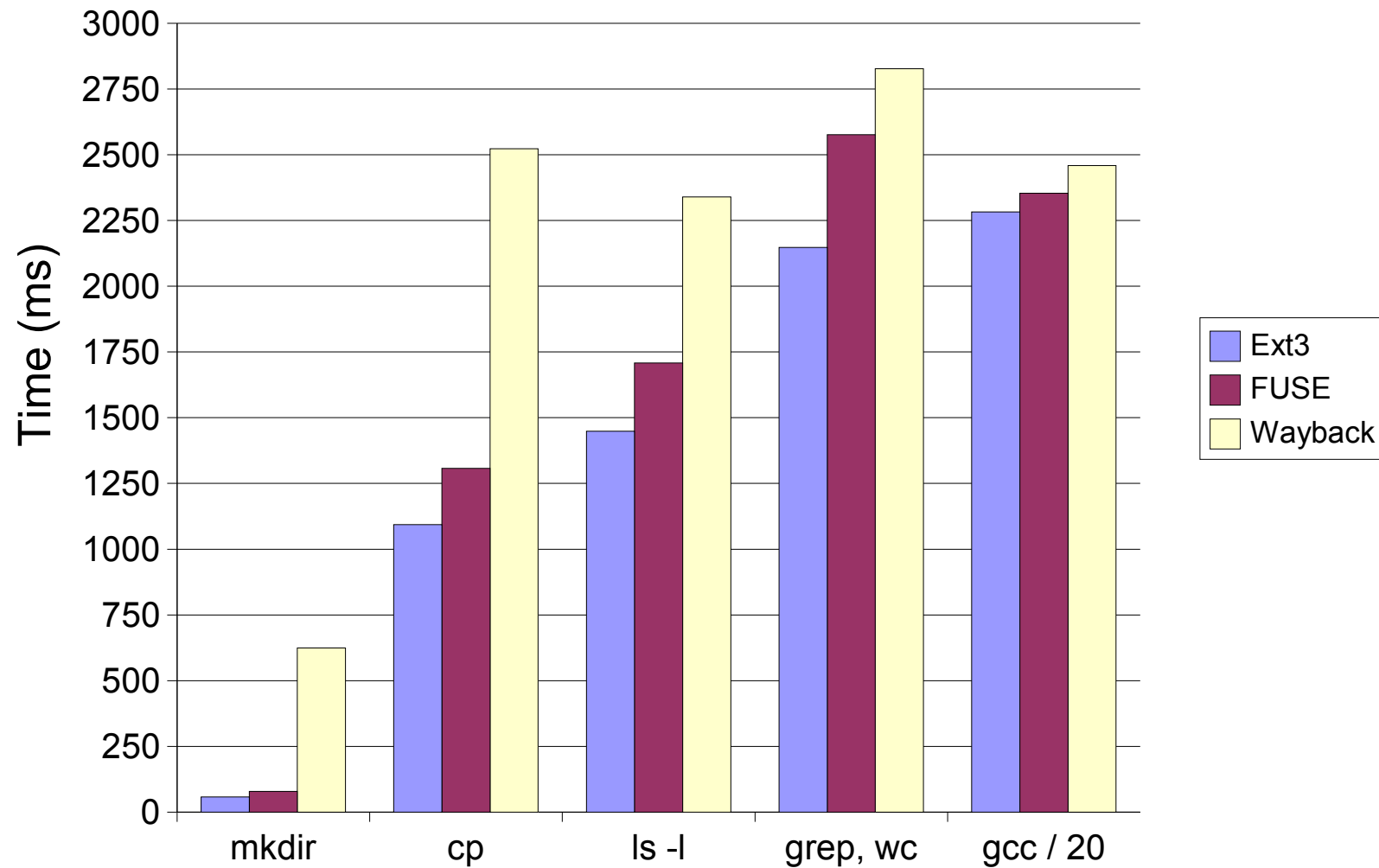
## Slow Processor: Bonnie Performance



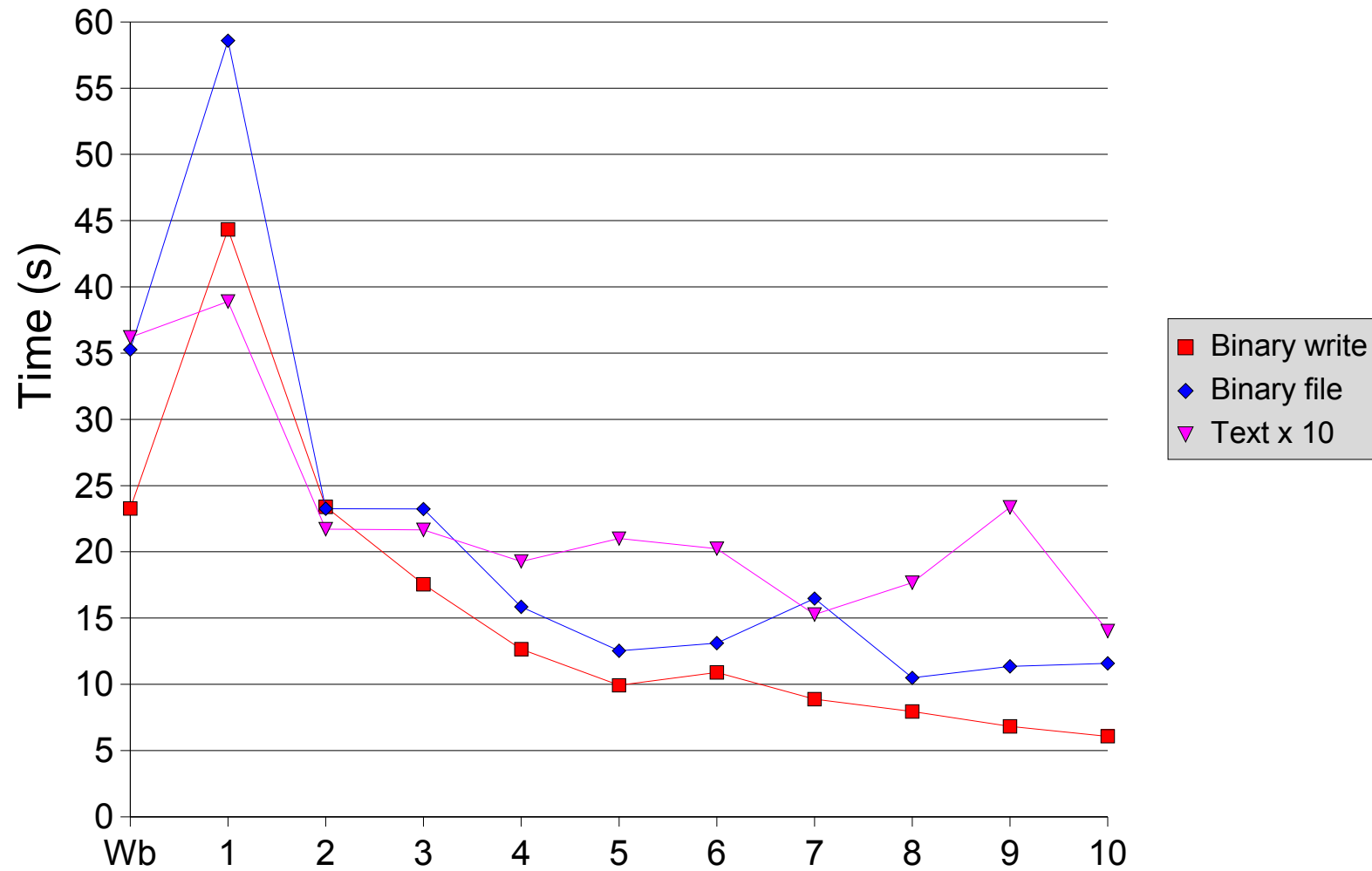
## Slow Disk: Andrew Performance



## Slow Processor: Andrew Performance



## Slow Disk: RCS Performance



# Slow Processor: RCS Performance

