2007 Linux Storage & Filesystem Workshop

February 12-13, 2007, San Jose, CA

DualFS: A New Journaling File System for Linux

Juan Piernas <juan.piernascanovas@pnl.gov>

SDM Project

Pacific Northwest National Laboratory
http://www.pnl.gov

Sorin Faibish <sfaibish@emc.com>

EMC^2 Corporation
http://www.emc.com

Introduction

- **Meta-data management is a key design issue**
  - Especially important for recovery after a system crash
- **Traditional file systems:**
  - Write meta-data in a synchronous way
  - Use fsck-like tools
- **Current approaches:**
  - Log of last meta-data updates (e.g. XFS, JFS)
  - Asynchronous meta-data writes (e.g. Soft Updates)
- **Current approaches treat data and meta-data somewhat differently**
  - <u>But they are completely different</u>.

Introduction

- ❑ **DualFS: aimed at providing both good performance and fast consistency recovery through data and meta-data separation**
- ❑ **This separation is not a new idea:**
  - Muller and Pasquale (SOSP'91)
  - Cluster file systems (Lustre, PVFS)
- ❑ **DualFS proves, for the first time, that the separation can significantly improve file systems' performance without requiring several storage devices.**
- ❑ **Experimental results show that DualFS is the fastest file system in general (up to 98%)**

Outline

- ❑ Introduction
- ❑ Rationale
- ❑ DualFS
- ❑ Experimental Methodology and Results
- ❑ Conclusions

Rationale

Workload	I/O Requests (%)				I/O Time (%)	
	Data (R/W)		Meta-data (R/W)		Data	Meta-data
Root+Mail	28.41	(23.07/76.93)	71.59	(6.45/93.55)	20.47	79.53
Web+FTP	52.11	(63.37/36.63)	47.89	(23.45/76.55)	50.64	49.36
NFS	30.26	(63.06/36.94)	69.74	(27.14/72.86)	57.87	42.13
Backup	90.72	(99.94/00.06)	9.28	(71.08/28.92)	86.17	13.83

Distribution of the Data and Metadata Traffic for Different Workloads

Rationale

Workload	I/O Requests (%)				I/O Time (%)	
	Data (R/W)		Meta-data (R/W)		Data	Meta-data
Root+Mail	28.41	(23.07/76.93)	71.59	(6.45/93.55)	20.47	79.53
Web+FTP	52.11	(63.37/36.63)	47.89	(23.45/76.55)	50.64	49.36
NFS	30.26	(63.06/36.94)	69.74	(27.14/72.86)	57.87	42.13
Backup	90.72	(99.94/00.06)	9.28	(71.08/28.92)	86.17	13.83

Distribution of the Data and Metadata Traffic for Different Workloads

Rationale

Workload	Same-type Requests		Typeless Requests	
	Data (%)	Meta-data (%)	Data (%)	Meta-data (%)
Root+Mail	6.01	3.13	6.08	3.14
Web+FTP	42.48	6.43	43.10	7.01
NFS	11.25	10.86	11.47	10.89
Backup	77.25	1.20	79.92	25.14

Sequentiality of the Data and Metadata Requests for Different Workloads

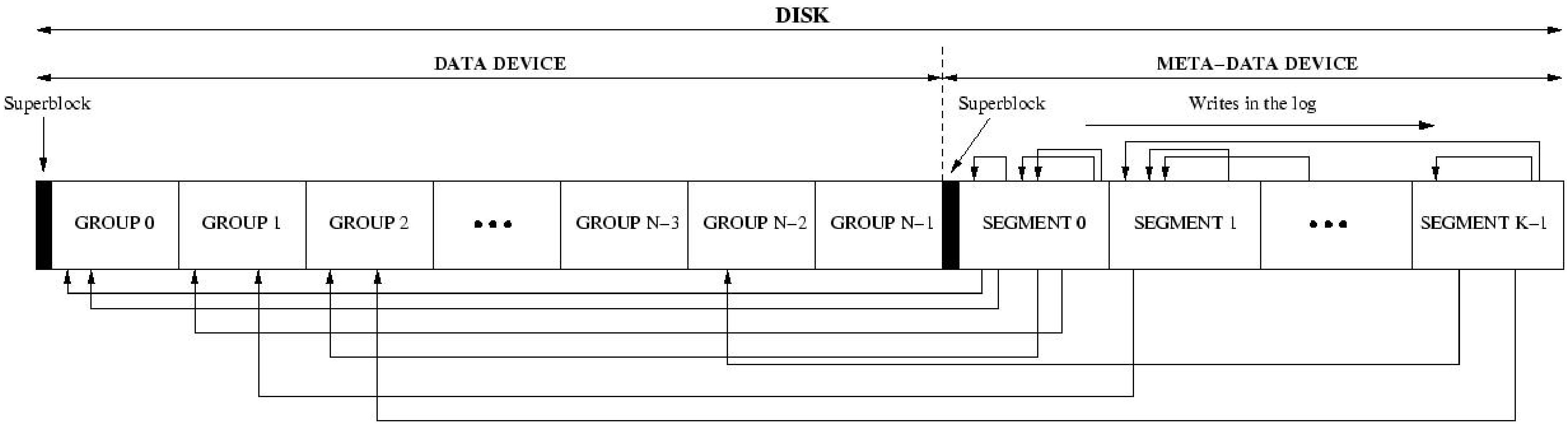
Rationale

- ❑ Our results confirm those obtained in previous works (Muller y Pasquale [1991], Ruemmler y Wilkes [1993], Vogels [1999])
- ❑ Our results also include disk I/O time, and sequentiality of data and meta-data requests
- ❑ Some conclusions about meta-data:
  - Meta-data represents a high percentage of the total I/O time in many workloads
  - Writes are predominant
  - Almost always, request are not sequential

Outline

- ❑ Introduction
- ❑ Rationale
- ❑ DualFS
- ❑ Experimental Methodology and Results
- ❑ Conclusions

Structure Overview



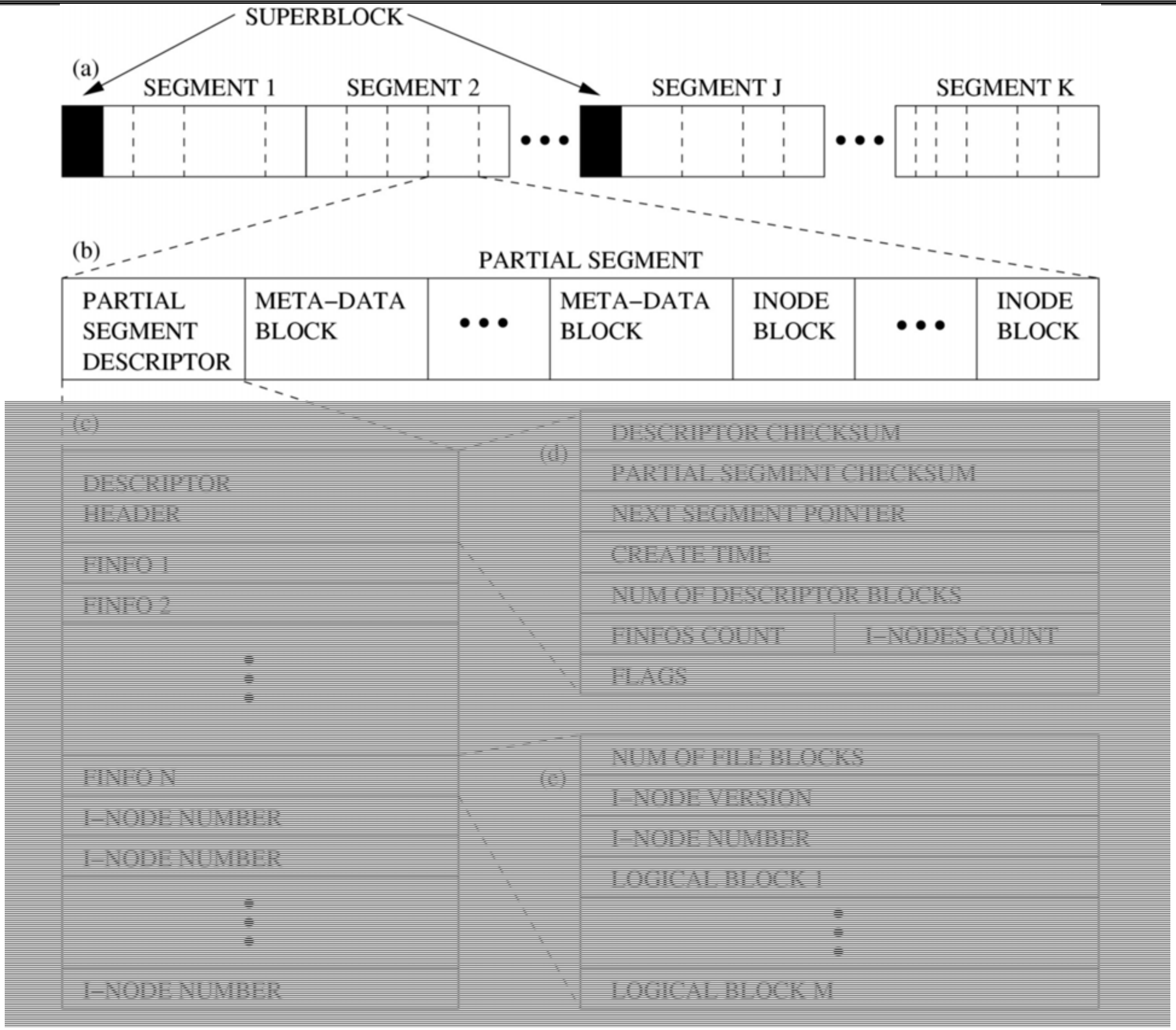
Data Device

- **Like Ext2 without meta-data blocks**
- **Groups:**
  - Grouping is performed in a per directory basis.
  - Related blocks are kept together.
  - File layout for optimizing sequential access.
  - DualFS selects the emptiest group with least associated i-nodes, in that order.
- **Directory affinity:**
  - Select the parent's directory if the best one it is not good enough (it does not have, at least, x% more free blocks)
- **Data blocks are not written synchronously**
  - However, new data blocks are written before the corresponding meta-data blocks (Ext3 "ordered" mode)

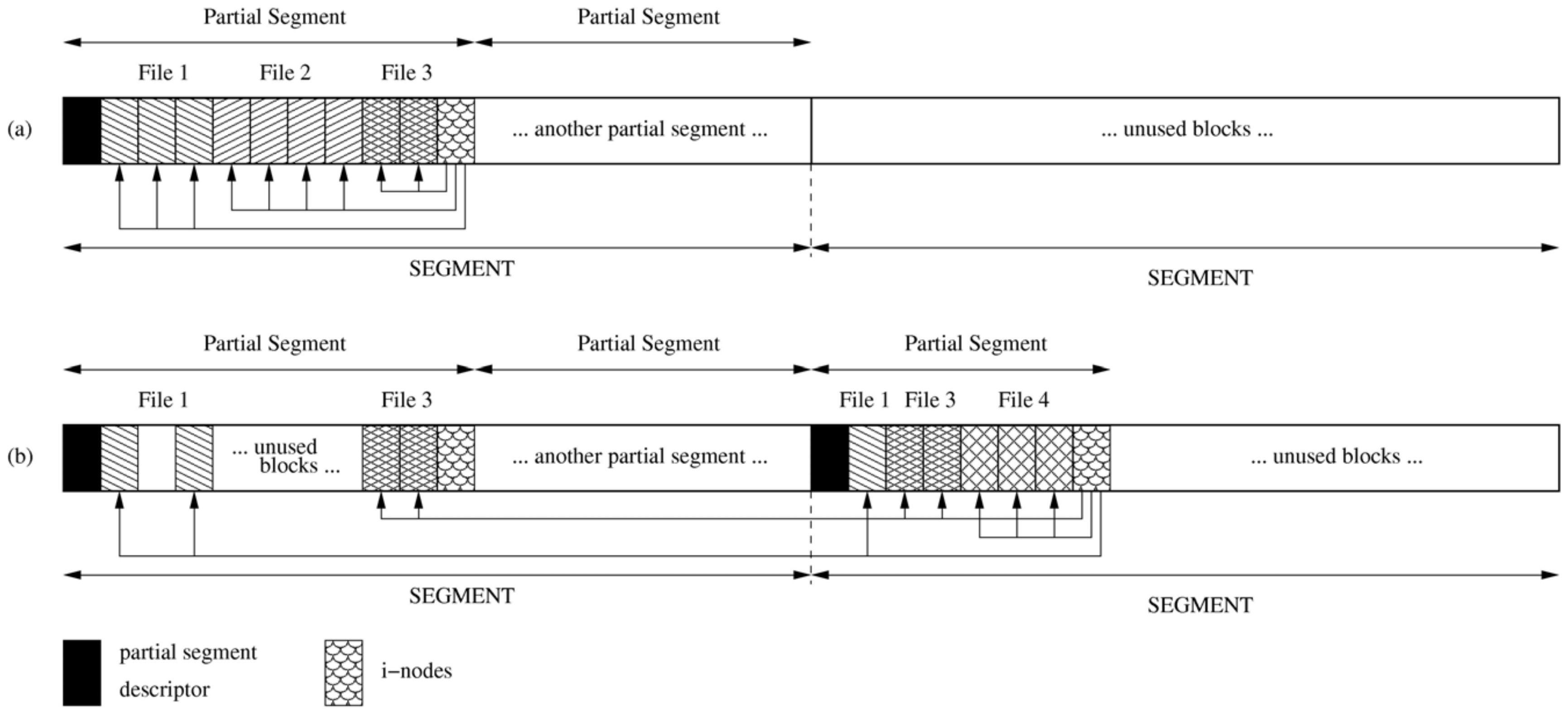
Meta-Data Device

- **We understand meta-data as all these items:**
  - i-nodes, indirect blocks, directory “data” blocks, and symbolic links
  - bitmaps, superblock copies
- **Organized as a log-structured file system**
  - Similar structure to that of BSD-LFS.
- **Almost all the meta-data elements have the same structure as that of their Ext2/Ext3 counterparts**
  - The main difference is how those elements are written to disk!!!

Meta-Data Device Structure

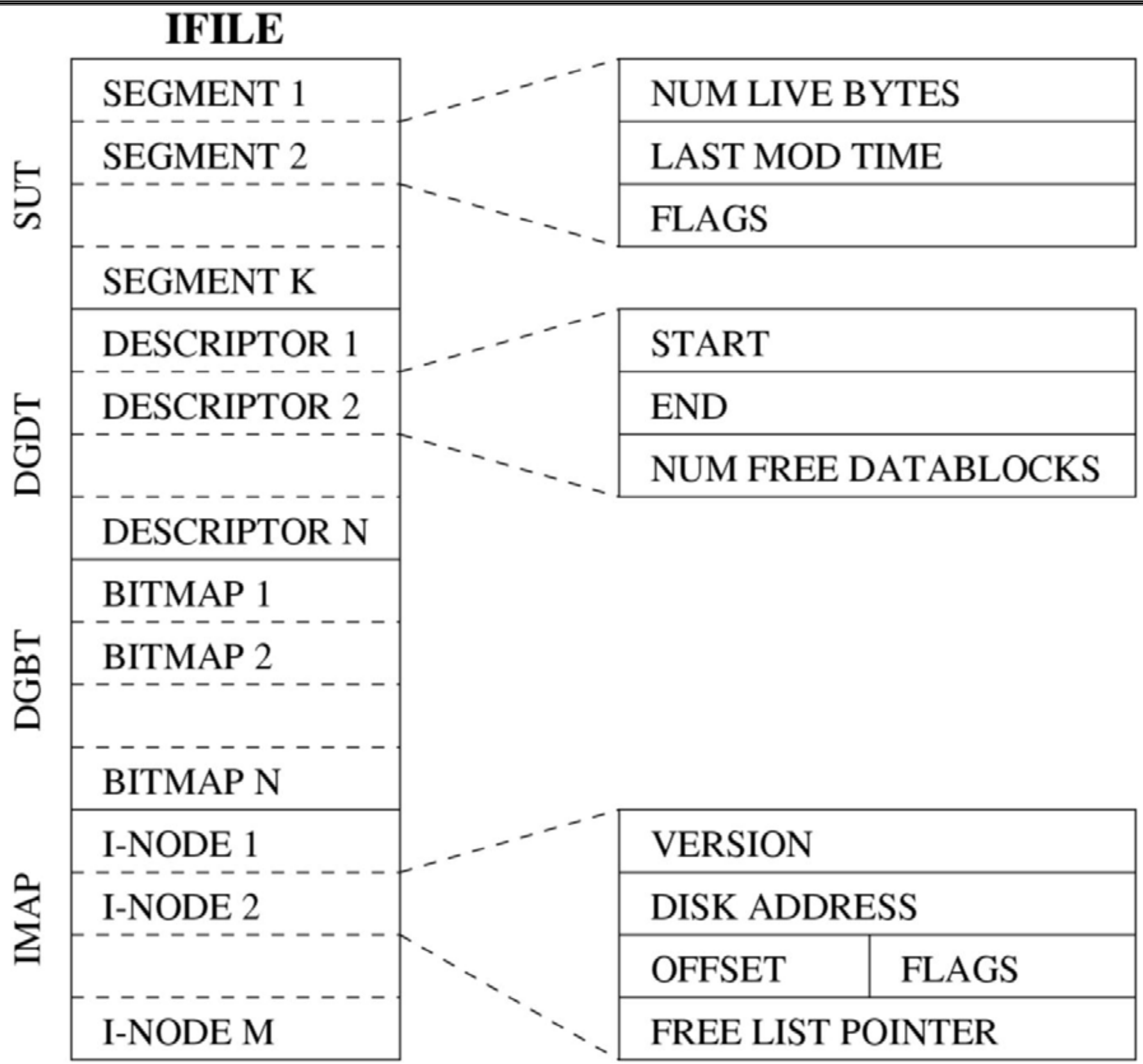


Meta-data Device's Operation



Changes in the meta-data device after modifying file 1, deleting file 2, adding two blocks to file 3, and creating a new file (file 4).

IFile



IFILE

SUT
SEGMENT 1
SEGMENT 2
SEGMENT K

NUM LIVE BYTES
LAST MOD TIME
FLAGS

DGDT
DESCRIPTOR 1
DESCRIPTOR 2
DESCRIPTOR N

START
END
NUM FREE DATABLOCKS

DGBT
BITMAP 1
BITMAP 2
BITMAP N

IMAP
I-NODE 1
I-NODE 2
I-NODE M

VERSION
DISK ADDRESS
OFFSET
FLAGS
FREE LIST POINTER

Meta-Data Prefetching

- **A solution to the read problem**
- **Simple: when the required meta-data block *B* is not in main memory, DualFS reads a group of consecutive blocks, from *B-j* to *B+i*, from the meta-data device**
- **Meta-data locality provided by “partial segments”:**
  - Temporal
  - Spatial
- **I/O-time efficient**
  - It does not produce further requests.
  - It takes advantage of the built-in disk cache.

On-Line Meta-Data Relocation

- The meta-data prefetching efficiency may deteriorate due to several reasons (changes in read patterns, file system aging, etc)
- Solution: on-line relocation of meta-data blocks
  - Every meta-data block which is read (from disk or main memory) is written again to the log.
- Relocation increases both spatial and temporal locality.
- More meta-data writes, but carried out efficiently
- Implicit relocation of i-nodes (atime updates)

Recovery

- ❑ **DualFS is considered consistent when information about meta-data is correct.**
- ❑ **We can recover the file system consistency very quickly from the last checkpoint.**
  - ● The length of time for recovery is proportional to the inter-checkpoint interval.
- ❑ **Recovering a DualFS file system means recovering its IFile.**

Outline

- ❑ Introduction
- ❑ DualFS
- ❑ Experimental Methodology and Results
- ❑ Conclusions

File Systems Compared

- **Ext2, no special mount options**
- **Ext3, “-o data=ordered” mount option**
- **XFS, “-o logbufs=8,osyncisdsync” mount options**
- **JFS, no special mount options**
- **ReiserFS, “-o notail” mount option**
- **DualFS, with:**
  - meta-data prefetching (16 blocks)
  - on-line meta-data relocation
  - directory affinity (10%).

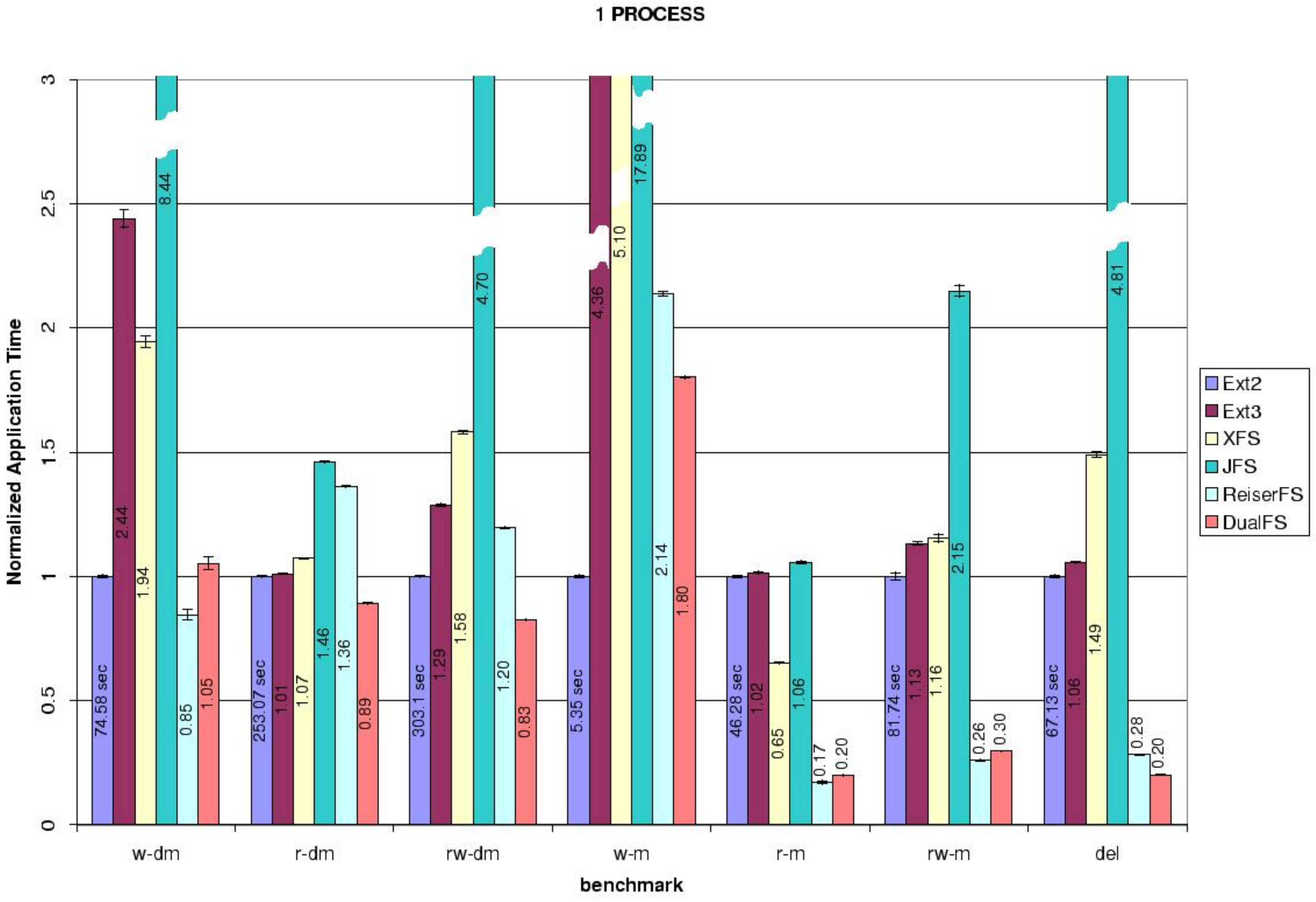
System Under Test

	Linux Platform
Processor	Two 450 Mhz Pentium III
Memory	256MB PC100 SDRAM
Disks	One 4 GB IDE 5,400 RPM Seagate ST34310A One 4 GB SCSI 10,000 RPM Fujitsu MAC3045SC SCSI disk: Operating system,swap and trace log. IDE disk: test disk
OS	Linux 2.4.19

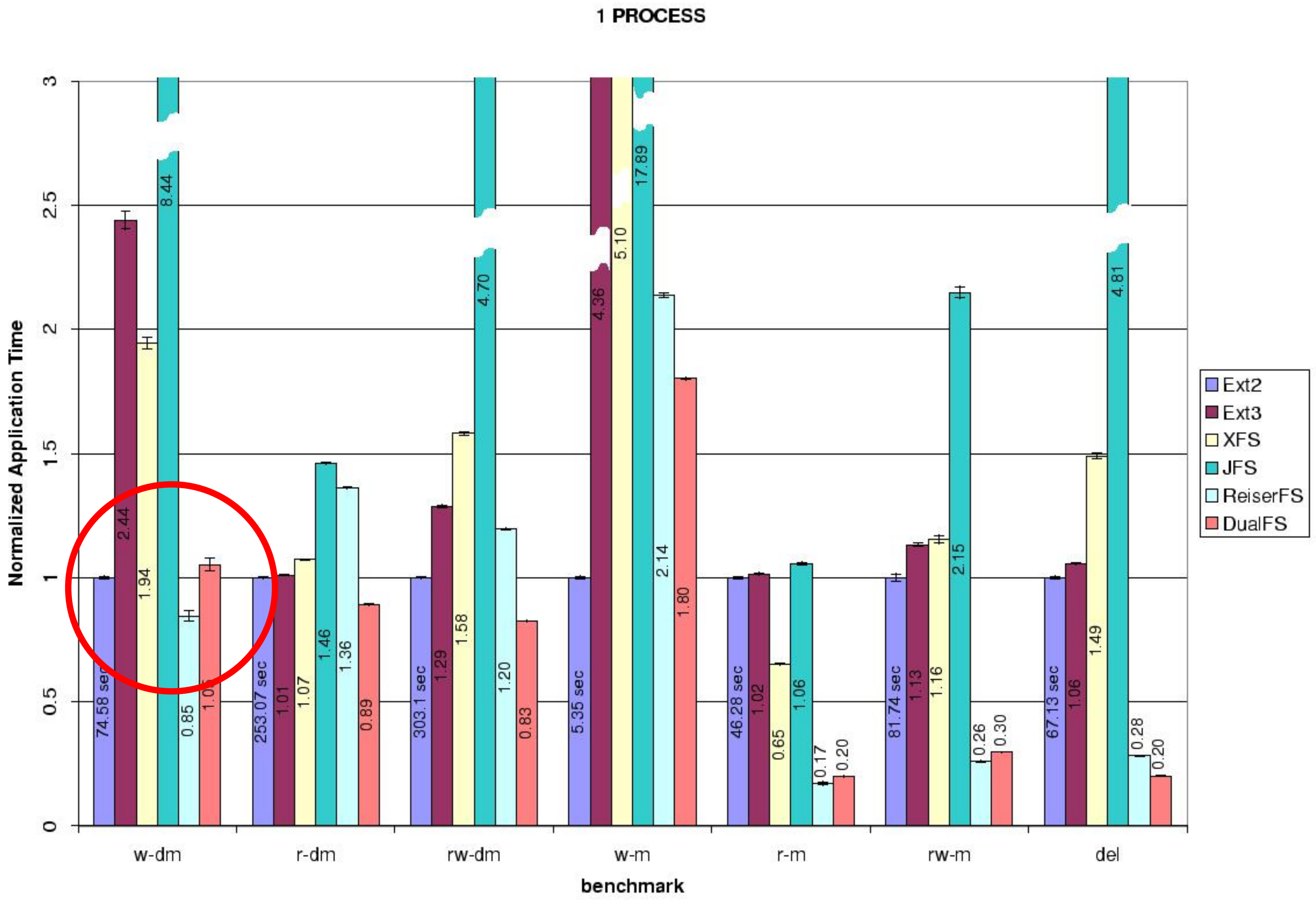
Microbenchmarks

- ❑ Read-meta (r-m): find files larger than 2 KB in a directory tree.
- ❑ Read-data-meta (r-dm): read all the regular files in a directory tree.
- ❑ Write-meta (w-m): create a directory tree with empty files
- ❑ Write-data-meta (w-dm): create a directory tree.
- ❑ Read-write-meta (rw-m): copy a directory tree with empty files
- ❑ Read-write-data-meta (rw-dm): copy a directory tree
- ❑ Delete (del): delete a directory tree

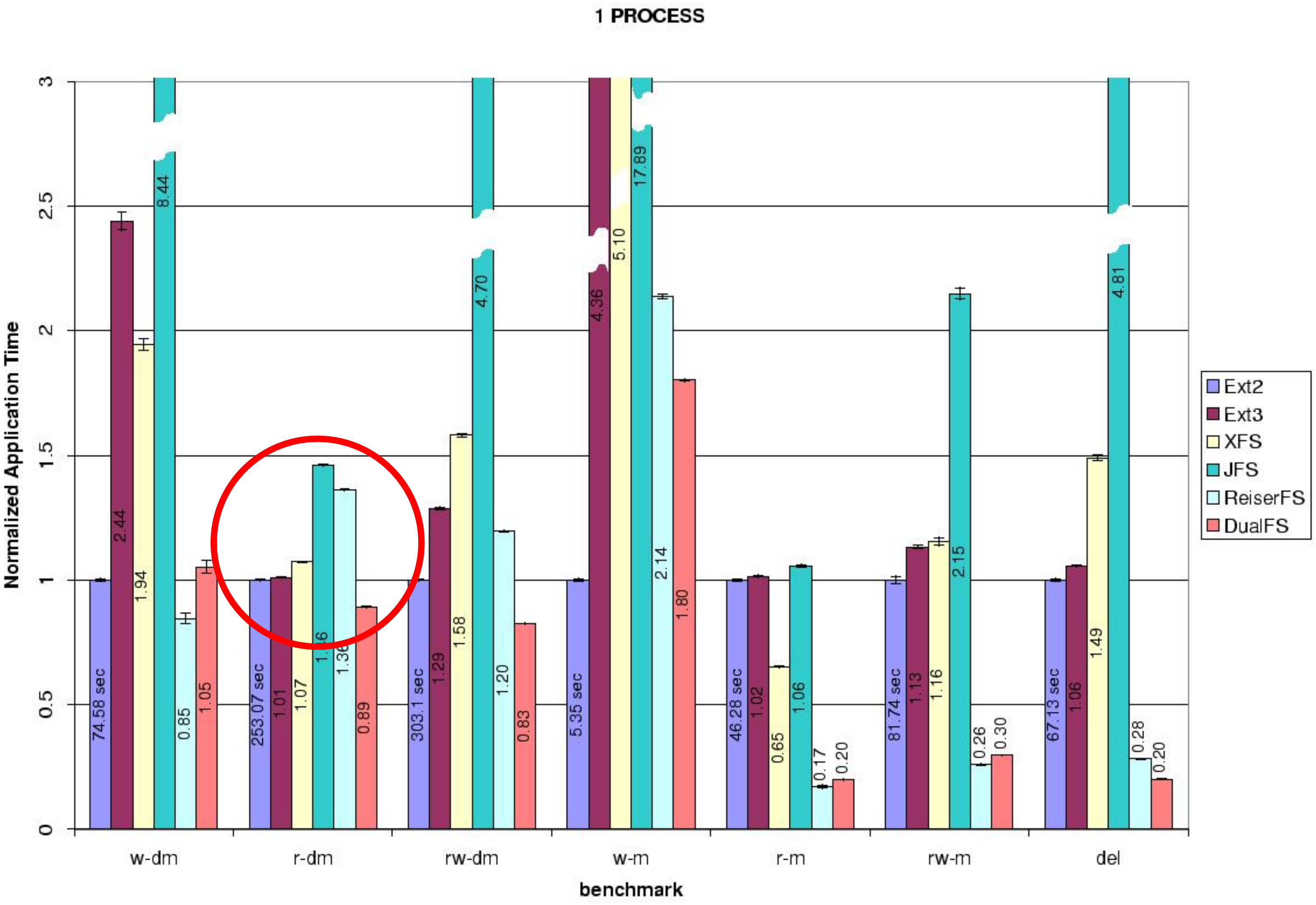
Microbenchmark (1 process)



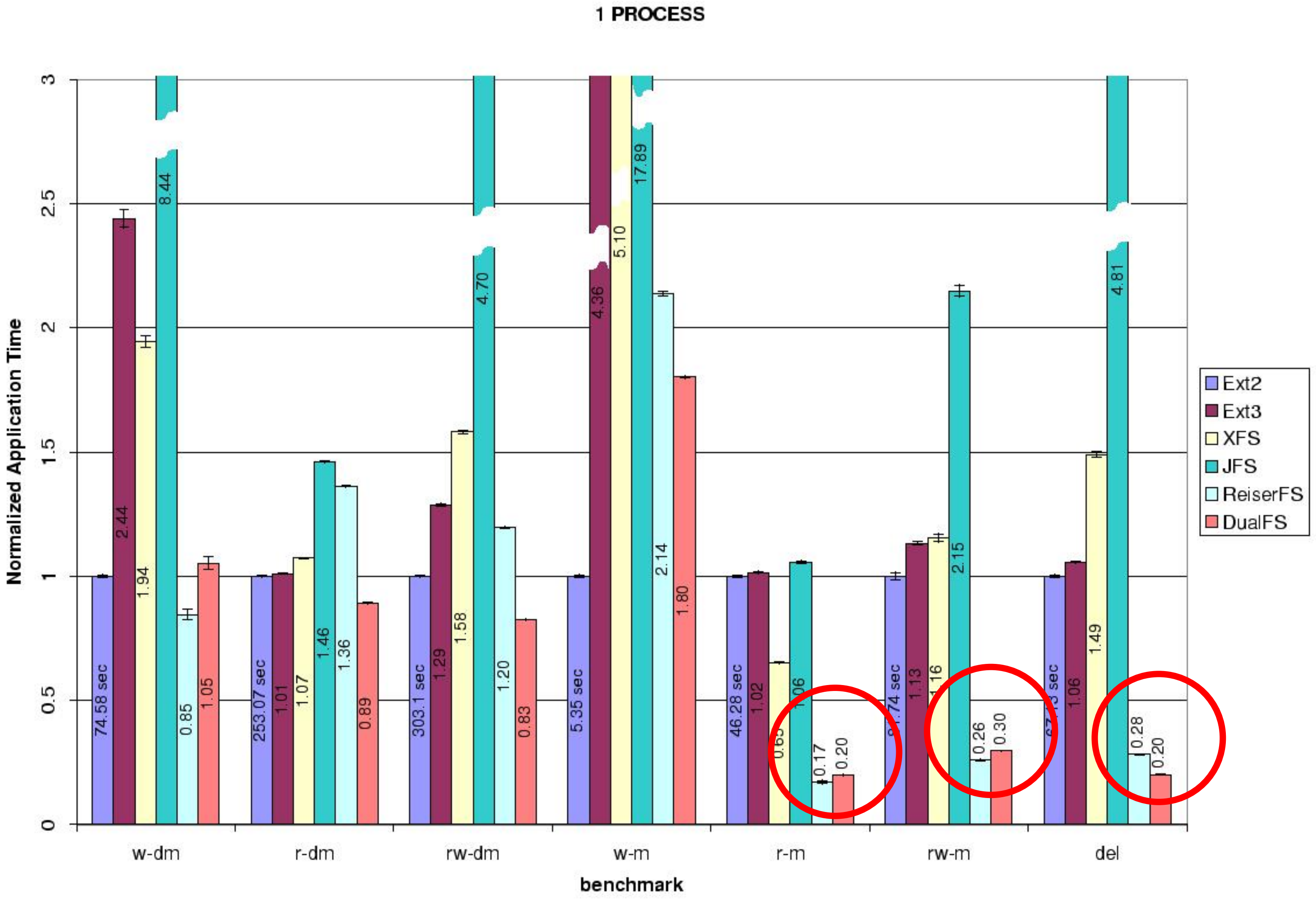
Microbenchmark (1 process)



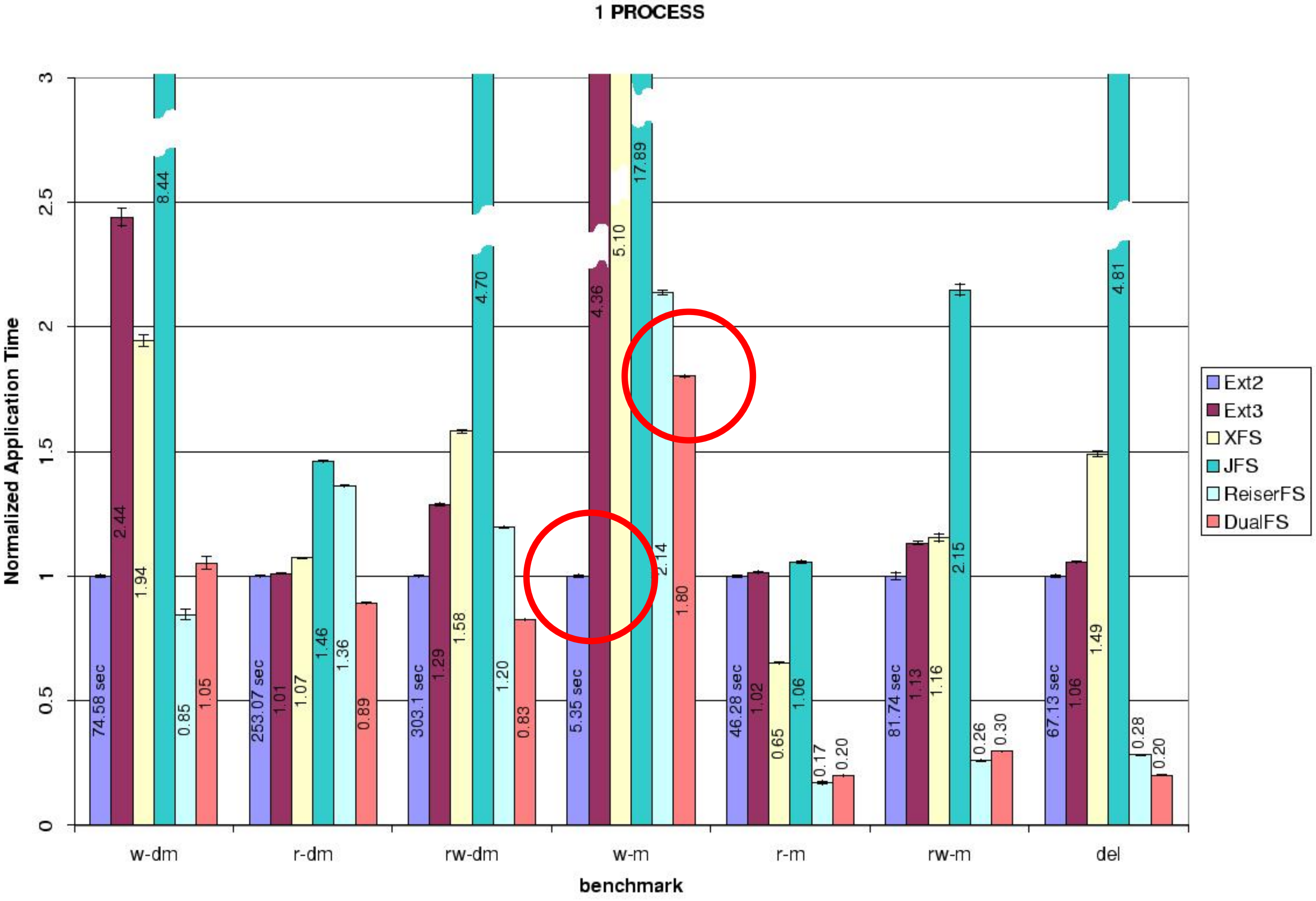
Microbenchmark (1 process)



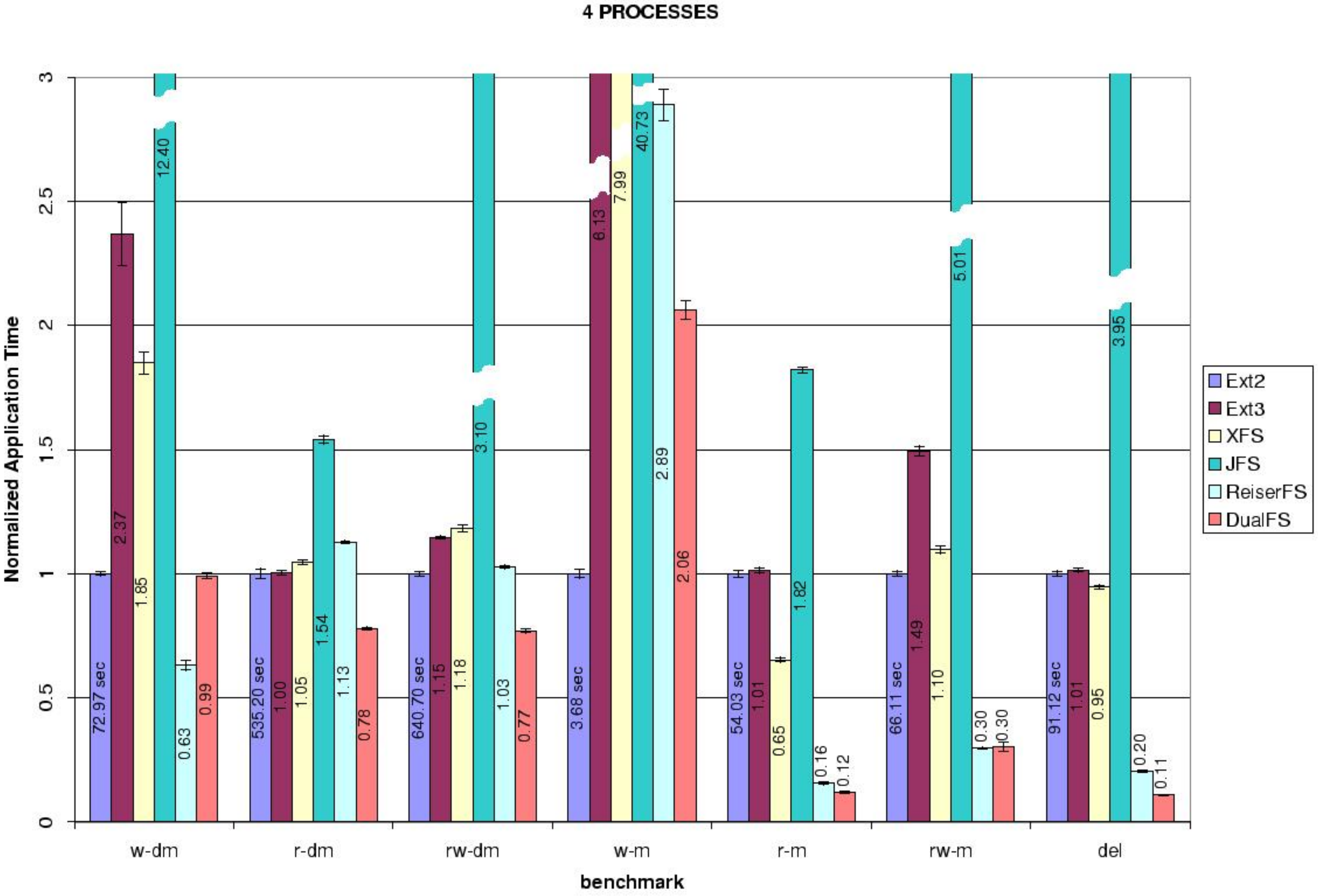
Microbenchmark (1 process)



Microbenchmark (1 process)



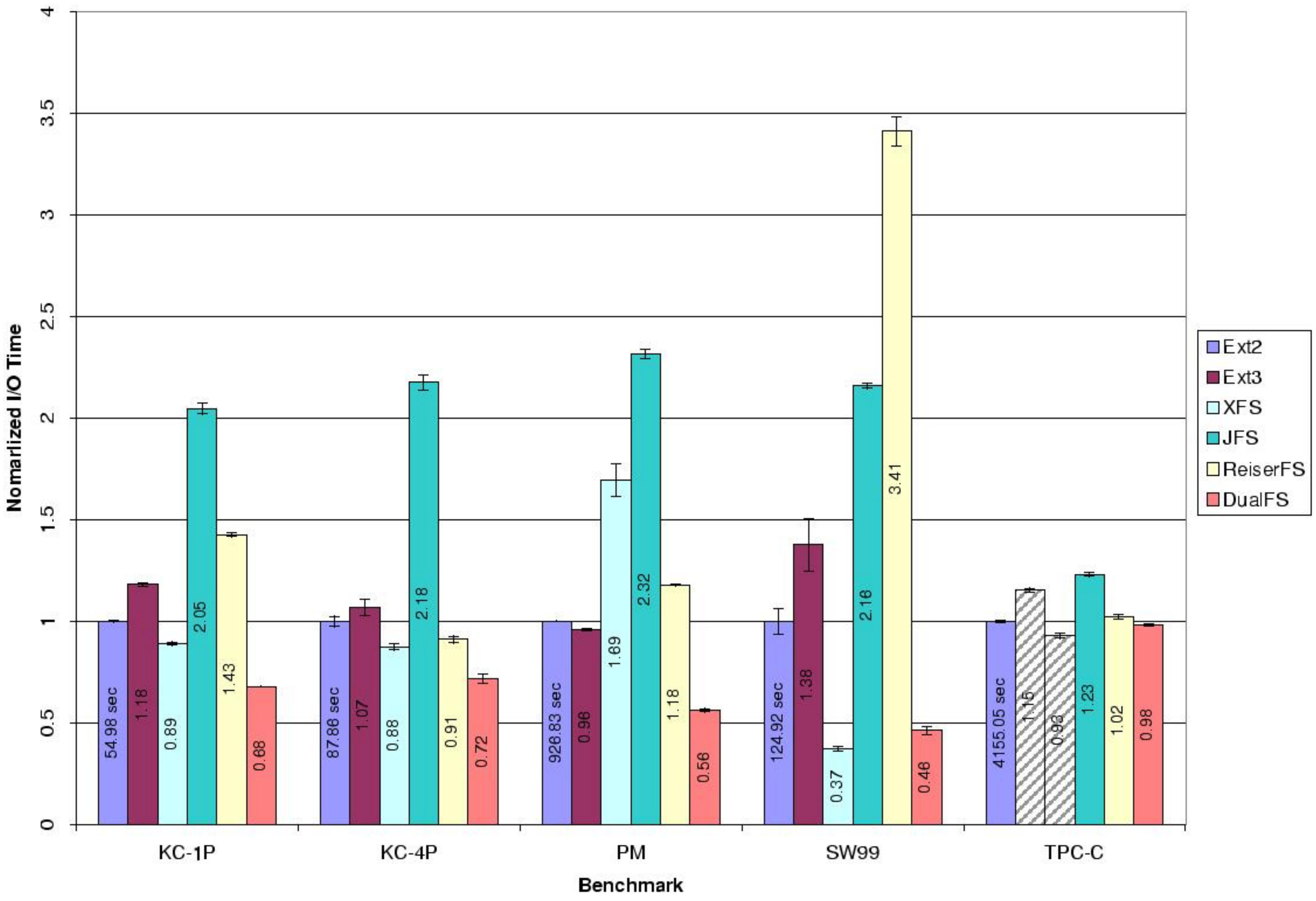
Microbenchmark (4 processes)



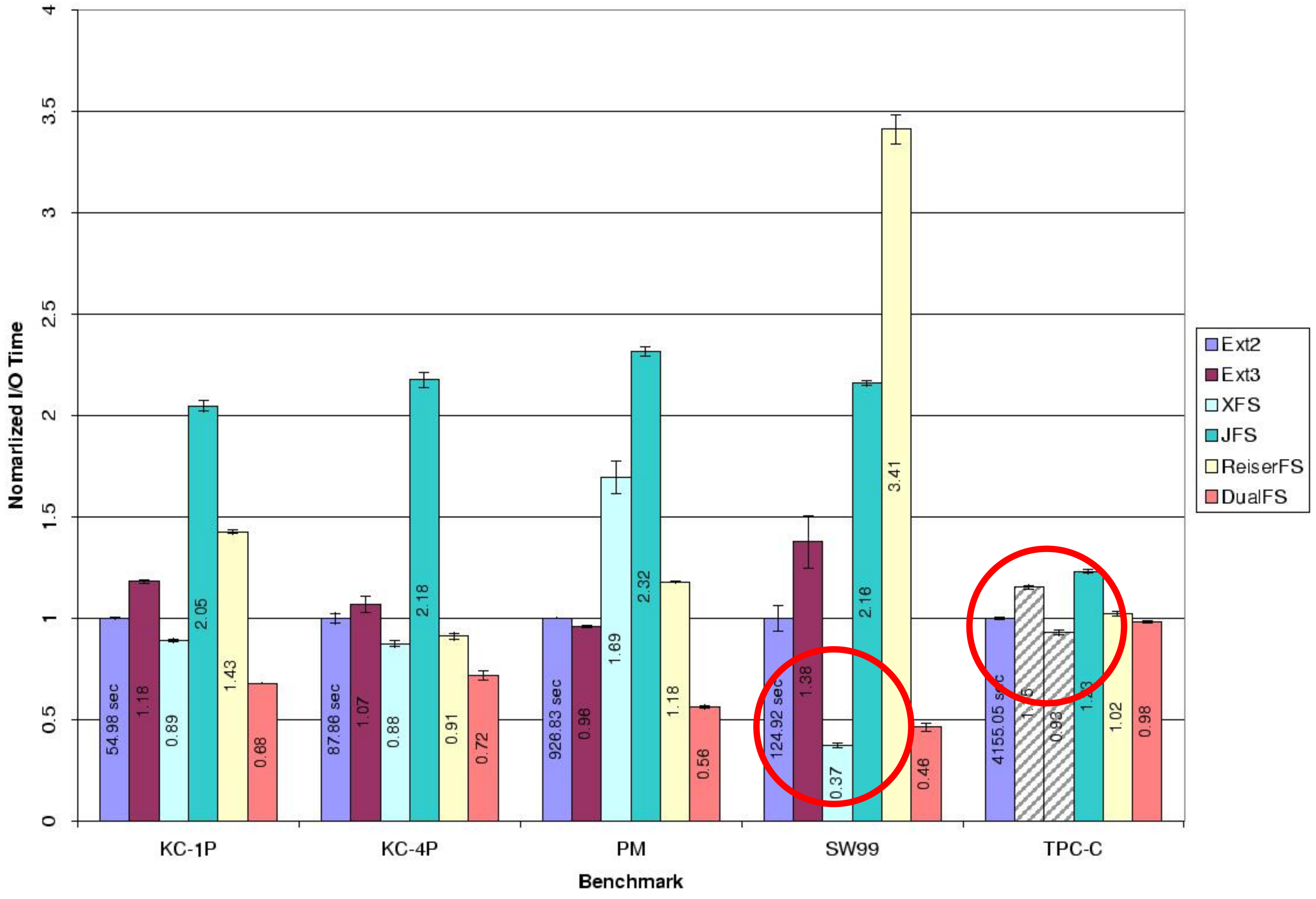
Macrobenchmarks

- ❑ Compilation of the Linux kernel 2.4.19, for 1 and 4 processes
- ❑ Specweb99
- ❑ Postmark v1.5
- ❑ TPC-C
- ❑ All but Postmark are CPU-bound in our system.

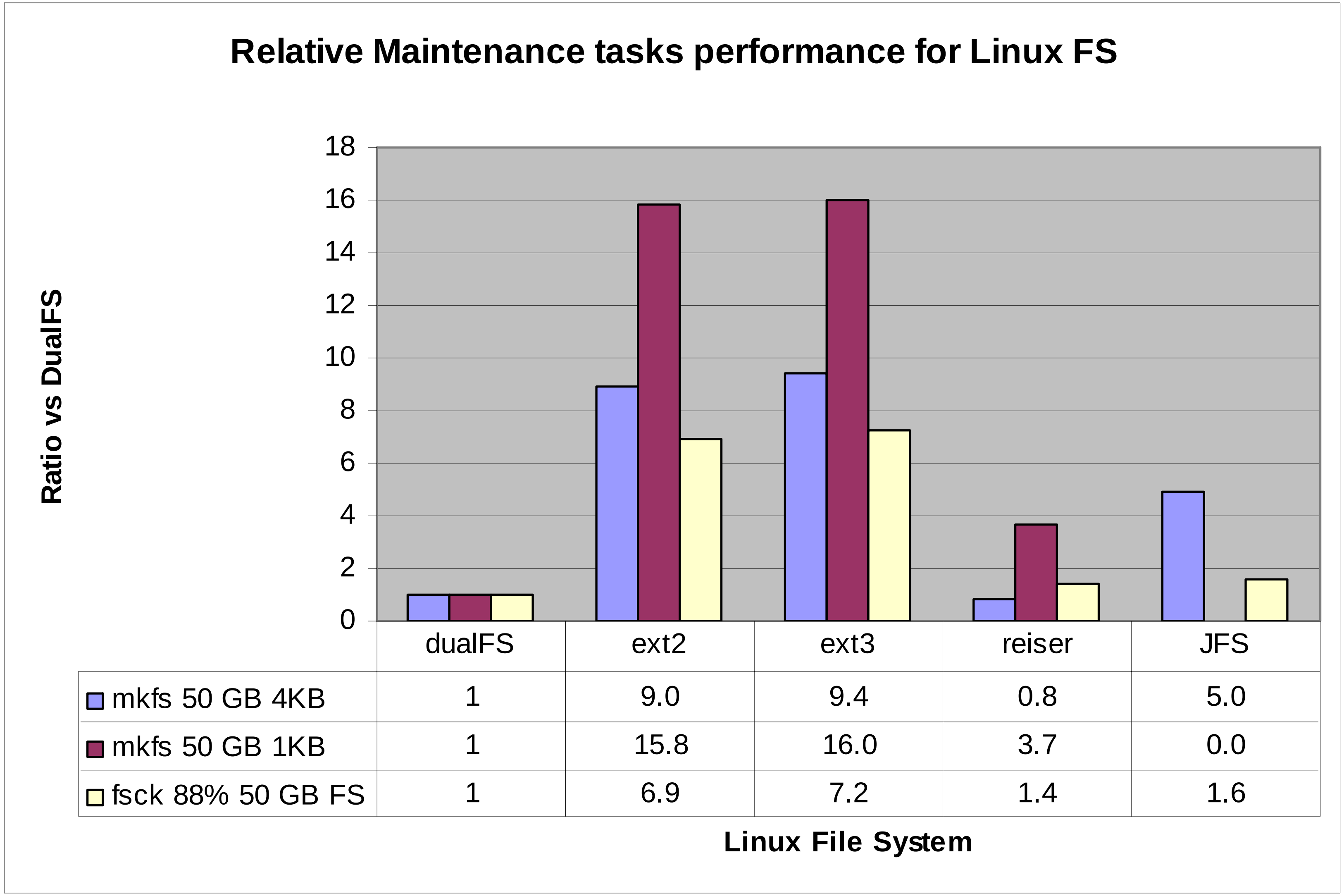
Macrobenchmarks (Disk I/O Time)



Macrobenchmarks (Disk I/O Time)



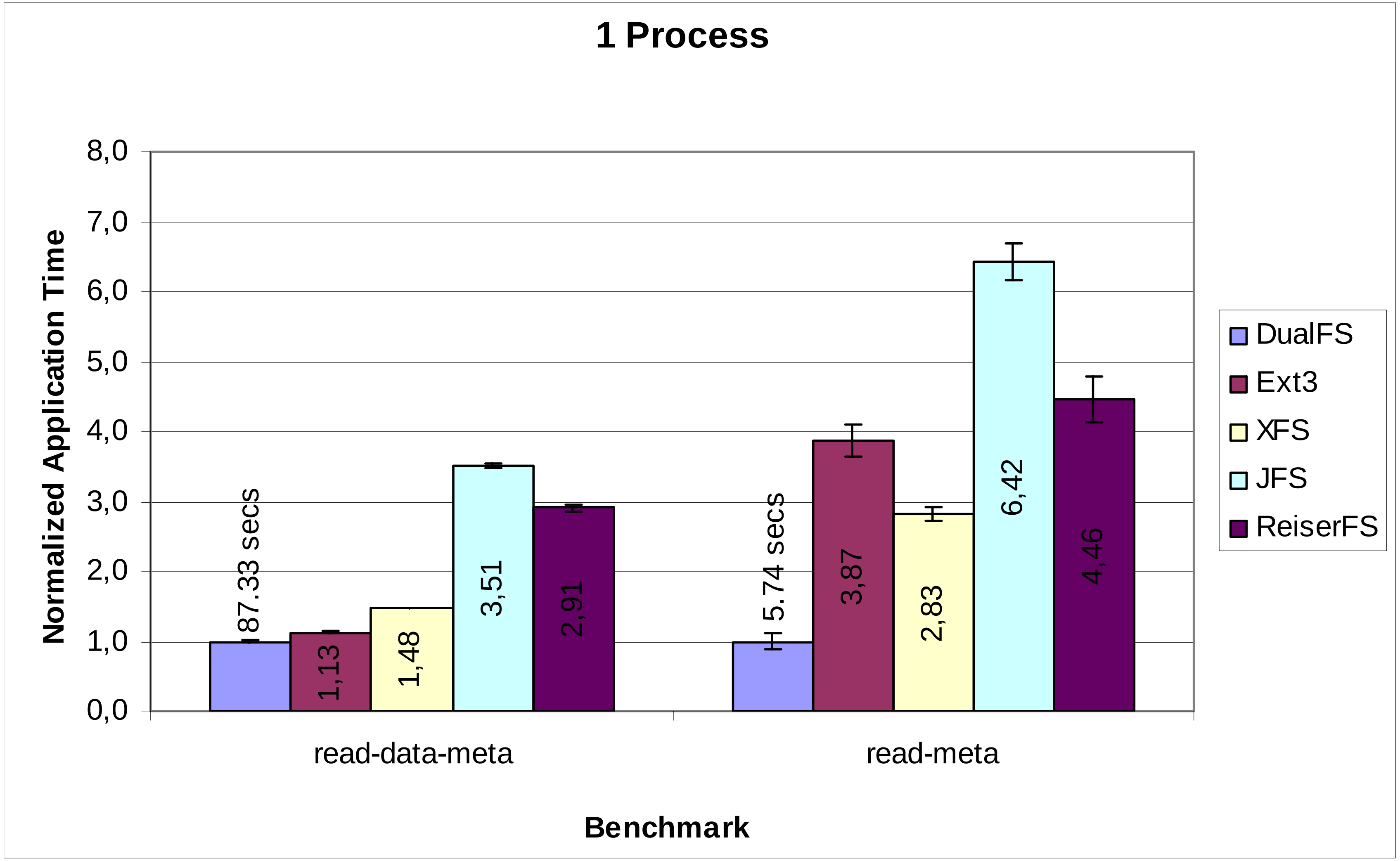
Maintenance Tasks



Relative Maintenance tasks performance for Linux FS

	dualFS	ext2	ext3	reiser	JFS
🟦 mkfs 50 GB 4KB	1	9.0	9.4	0.8	5.0
🟥 mkfs 50 GB 1KB	1	15.8	16.0	3.7	0.0
🟨 fsck 88% 50 GB FS	1	6.9	7.2	1.4	1.6

Some Results with Linux 2.6.11



Outline

- □ **Introduction**
- □ **DualFS**
- □ **Experimental Methodology and Results**
- □ **Conclusions**

Conclusions

- **DualFS is a new journaling file system with:**
  - data and meta-data managed in very different ways
  - one-copy meta-data blocks
  - large meta-data requests
  - quick consistency recovery
- **Compared six journaling and non-journaling file systems:**
  - DualFS is the best file system in most cases
  - DualFS reduces total I/O time up to 98%
- A new journaling file-system design based on data and meta-data separation, and special meta-data management, is desirable

Future work

❑ **To improve the design and the implementation:**

- Deferred block allocation and extensions.
- Better directory structure (B+ tree, ….).
- Data and meta-data devices in the same partition.
- Dealing with bad blocks.
- Meta-data device as generic LFS.

❑ **To explore new storage models:**

- Object Storage Devices (OSD)

❑ **To complete port to Linux 2.6.x:**

- This can not be the effort of just one man.
- DualFS is an open-source project now!!!

Questions?

DualFS: A New Journaling File System for Linux

Juan Piernas, and Sorin Faibish

DualFS Documentation

http://ditec.um.es/~piernas/dualfs

Source Code

http://dualfs.sourceforge.net