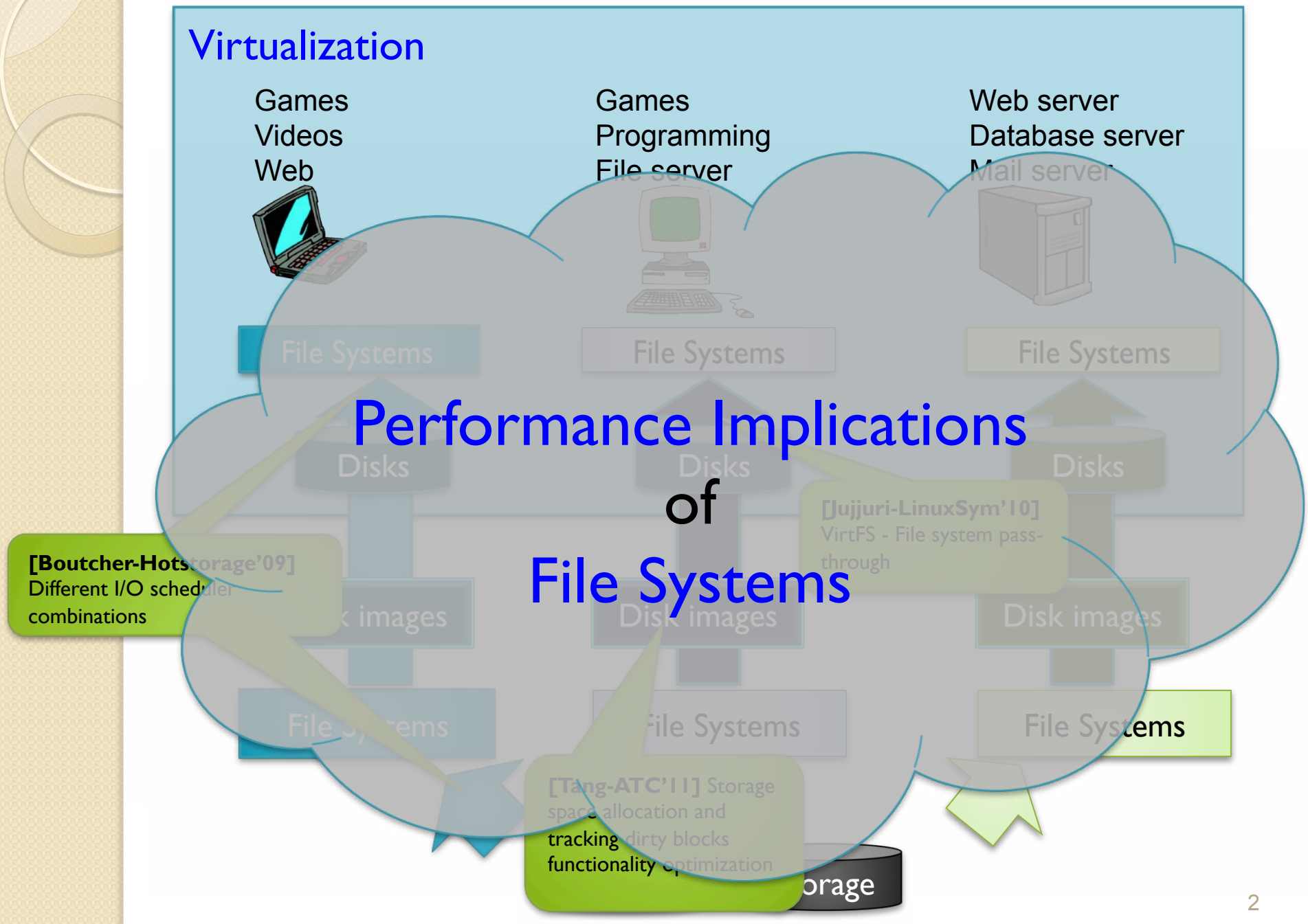
Understanding Performance Implications of Nested File Systems in a Virtualized Environment

Duy Le (Dan) - The College of William and Mary

Hai Huang - IBM T. J. Watson Research Center

Haining Wang - The College of William and Mary


Virtualization
Games
Videos
Web
Games
Programming
File server
Web server
Database server
Mail server
File Systems
File Systems
File Systems
Disks
Disks
Disks
Disk images
Disk images
Disk images
File Systems
File Systems
File Systems
[Boutcher-HotStorage'09]
Different I/O scheduler combinations
[Jujjuri-LinuxSym'10]
VirtFS - File system pass-through
[Tang-ATC'11] Storage space allocation and tracking dirty blocks functionality optimization
Performance Implications
of
File Systems

Nesting of File Systems

- "Selected file systems are based on workloads"
  - Only true in physical systems
- File systems for guest virtual machine
  - Workloads
  - Deployed file systems (at host level)
- **Investigation needed!**

Guest File Systems

- Ext2, Ext3, Ext4, ReiserFS, XFS, and JFS

Host File Systems

- Ext2, Ext3, Ext4, ReiserFS, XFS, and JFS

Understand nesting of file systems

- For the best performance?
  - ➔Best and worst Guest/Host File System combinations?
- Guest and Host File System Dependency
  - Varied I/Os and interaction
  - File disk images and physical disks

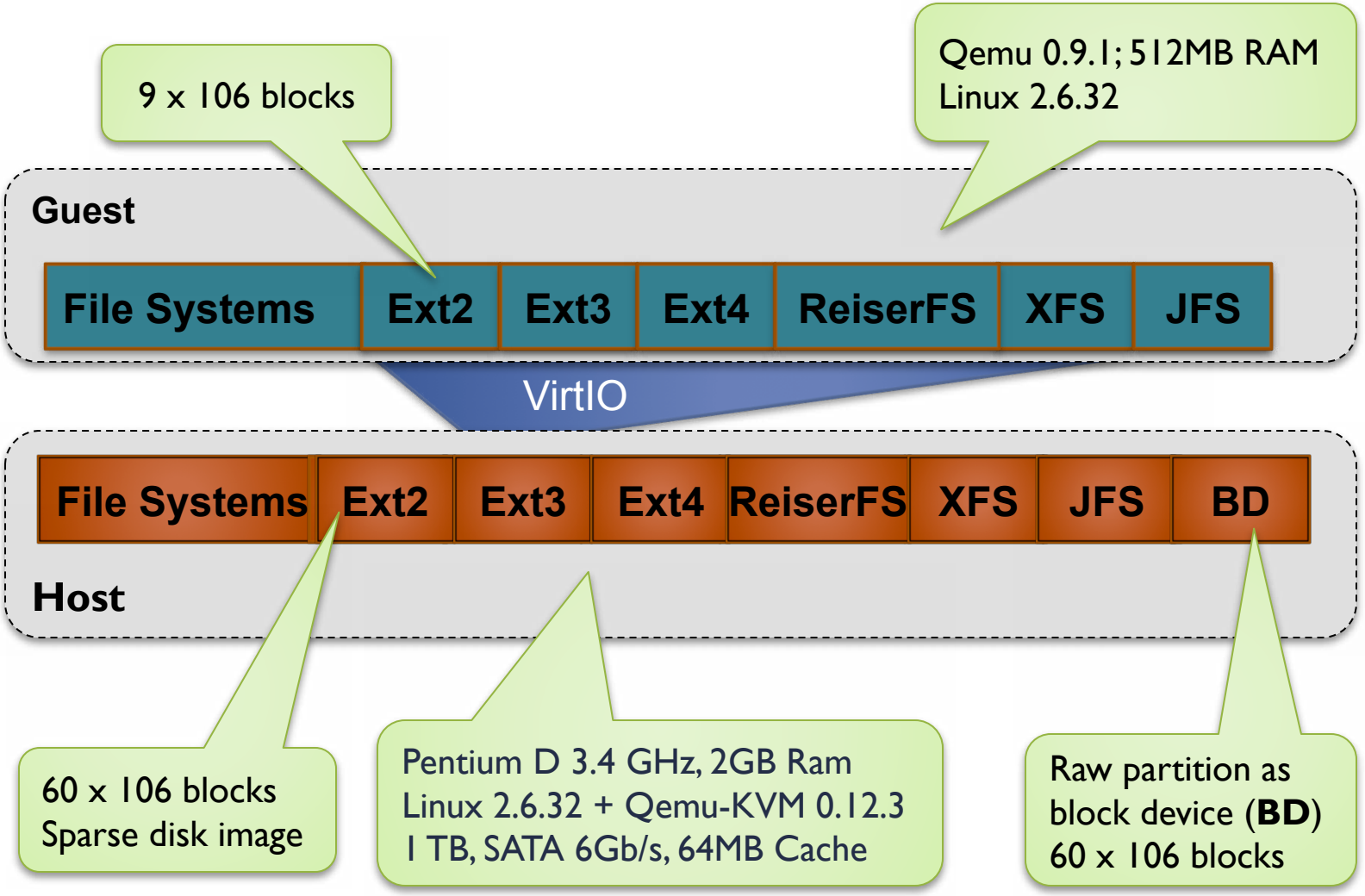
Outline

- Experimentations
  - Macro level
- Throughout analysis
  - Micro level
- Findings and Advice

Outline

- Experimentations
  - Macro level
- Throughout analysis
  - Micro level
- Findings and Advice

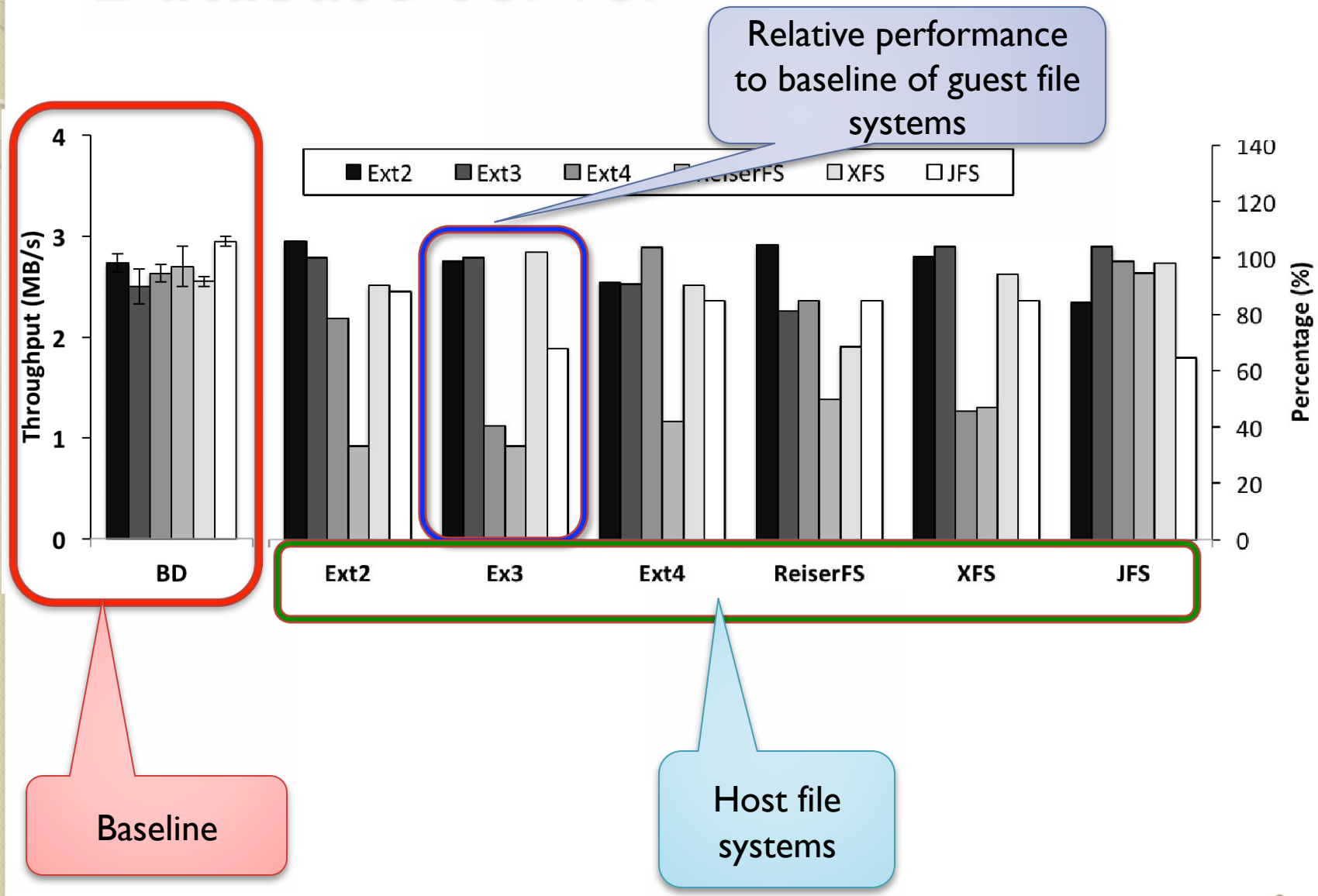
Experimental Setup



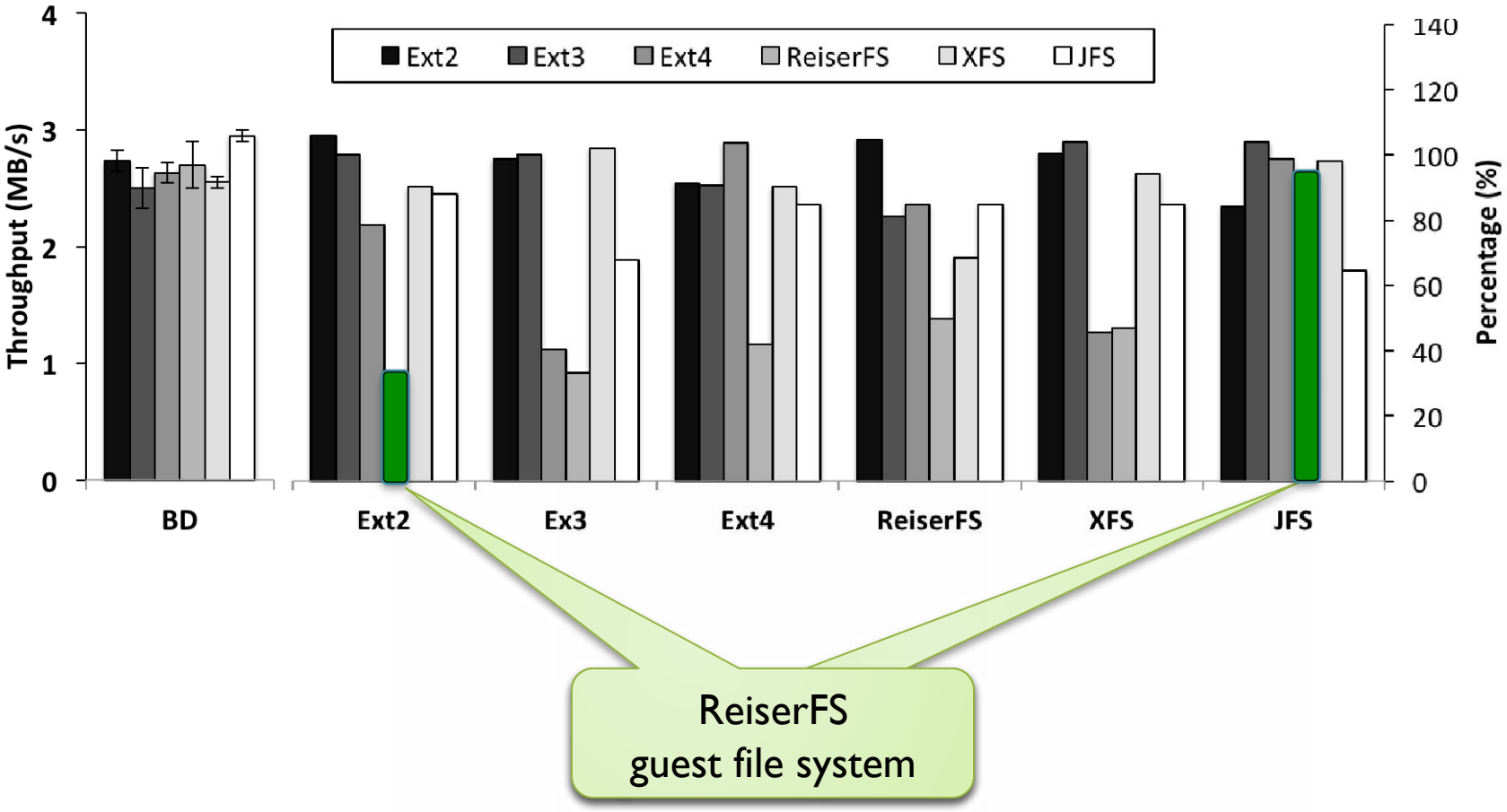
Performance features

- Filebench
  - File server, web server, database server, and mail server.
- Throughput
- Latency
- I/O Performance
  - Different abstraction consideration
    - Via block device (BD)
    - Via nested file systems
  - Relative performance variation
    - BD as baseline

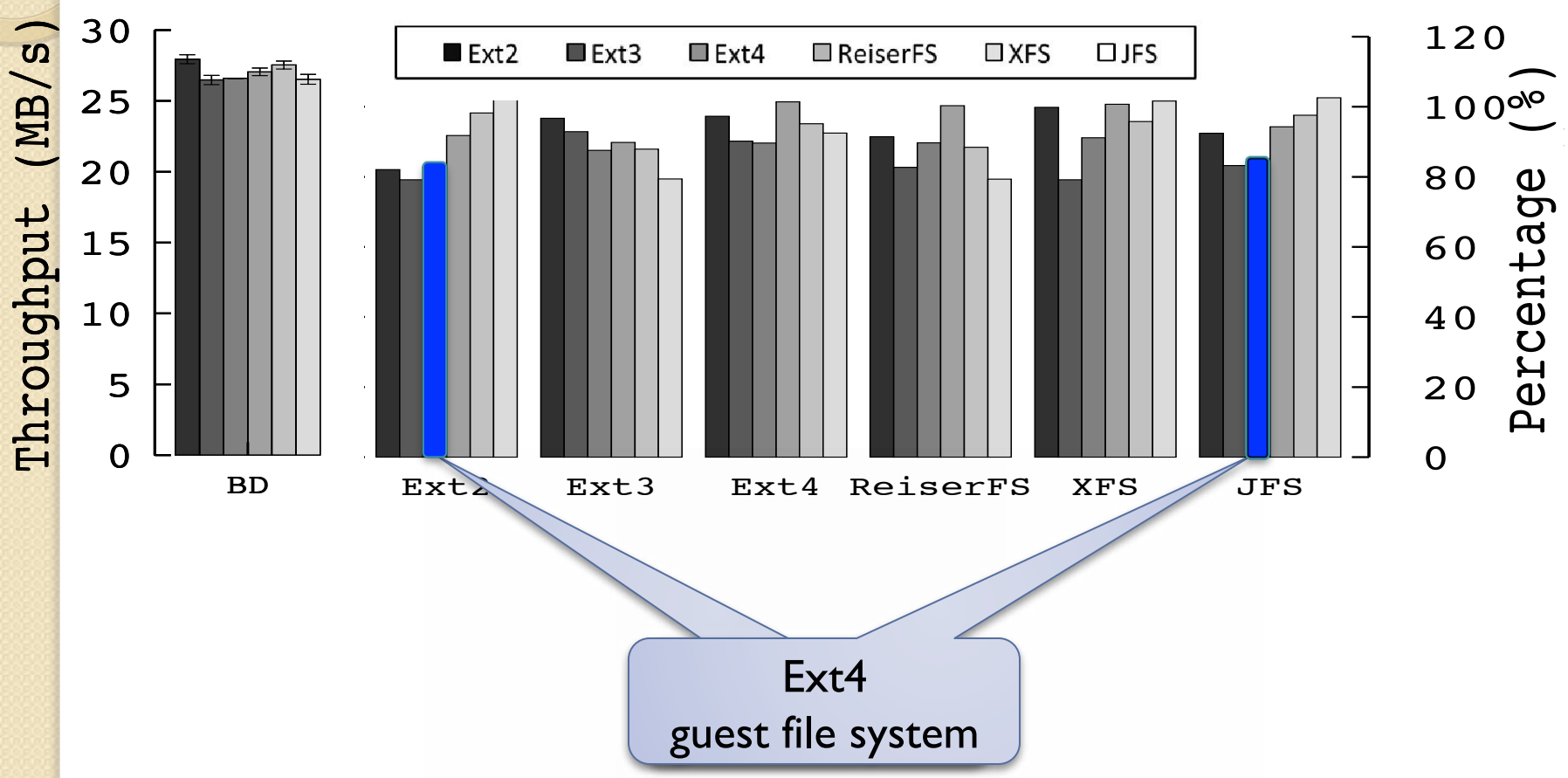
Database server



Database server



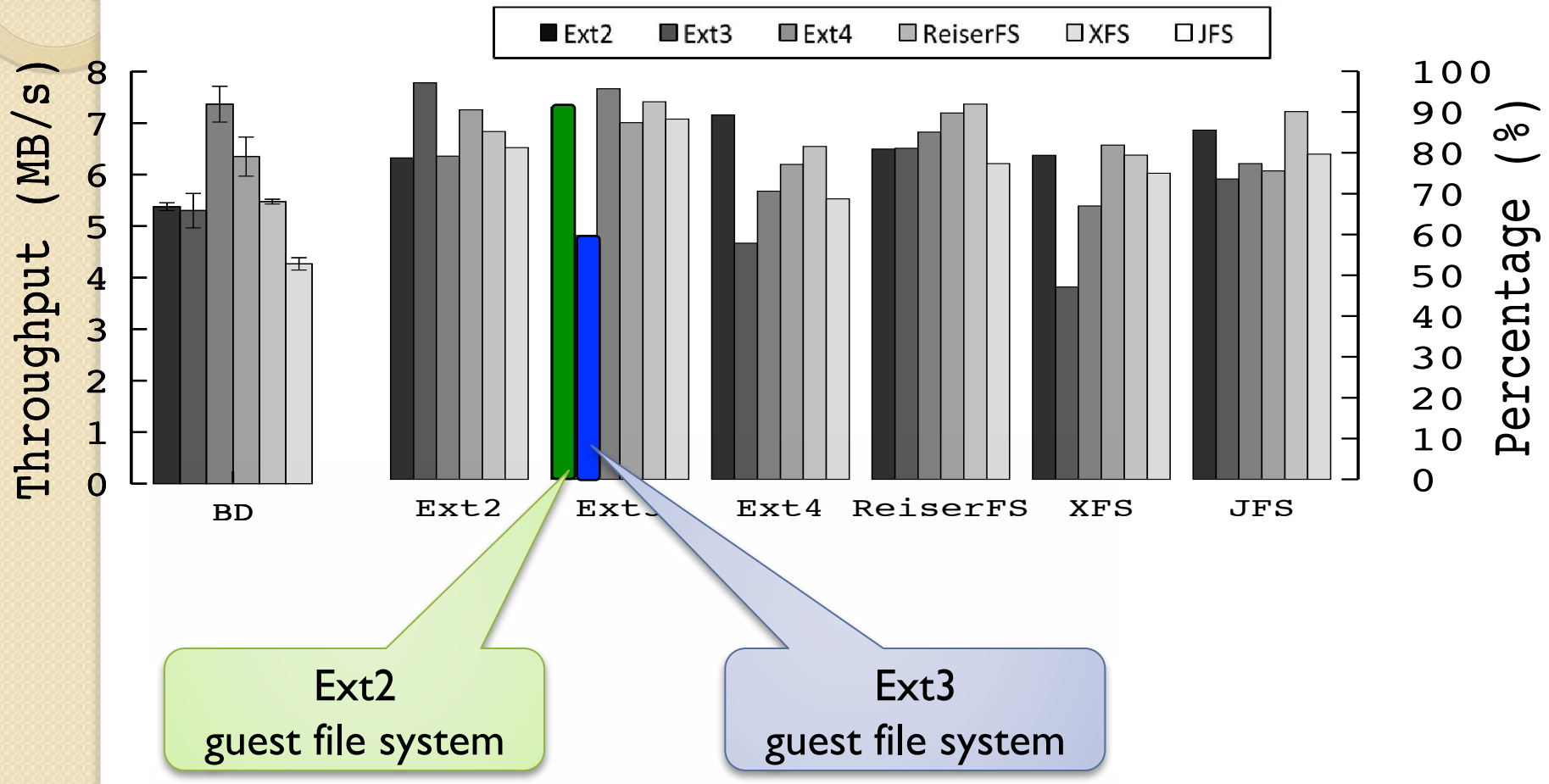
Web server


Throughput (MB/s)
30
25
20
15
10
5
0
Ext2
Ext3
Ext4
ReiserFS
XFS
JFS
BD
Ext2
Ext3
Ext4
ReiserFS
XFS
JFS
120
100
80
60
40
20
0
Percentage (%)
Ext4
guest file system

Macro level observations

- Guest file system → Host file systems
  - Varied performance
- Host file system → Guest file systems
  - Impacted differently
- Right and wrong combinations
  - Bidirectional dependency
- I/Os behave differently
  - Writes is more critical than Read (mail server)

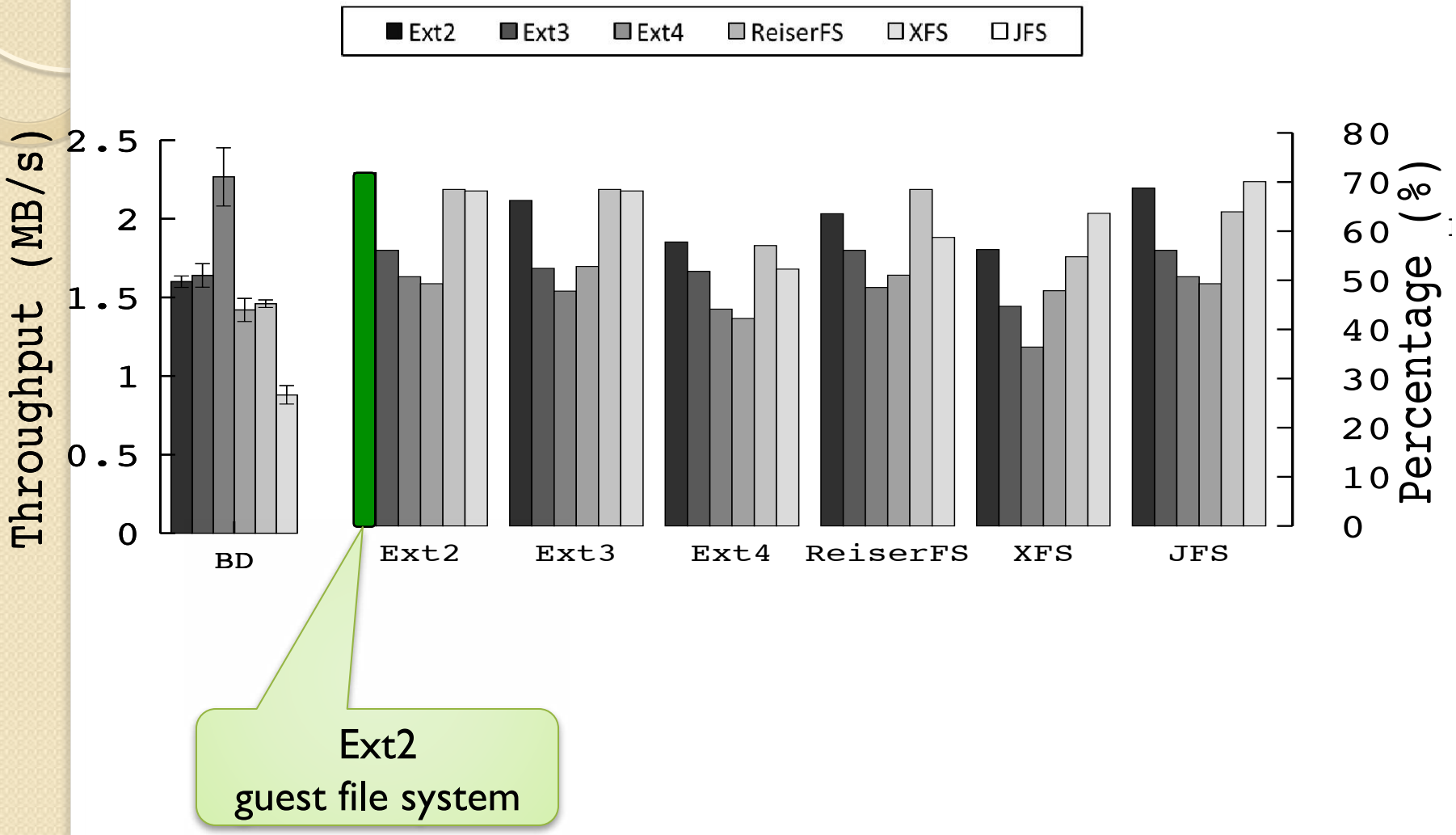
File server


Ext2
Ext3
Ext4
ReiserFS
XFS
JFS
Throughput (MB/s)
0
1
2
3
4
5
6
7
8
Percentage (%)
0
10
20
30
40
50
60
70
80
90
100
BD
Ext2
Ext4
ReiserFS
XFS
JFS
Ext2
guest file system
Ext3
guest file system

Macro level observations

- Guest file system → Host file systems
  - Varied performance
- Host file system → Guest file systems
  - Impacted differently
- Right and wrong combinations
  - Bidirectional dependency (mail server)
- I/Os behave differently
  - Writes is more critical than Read (mail server)

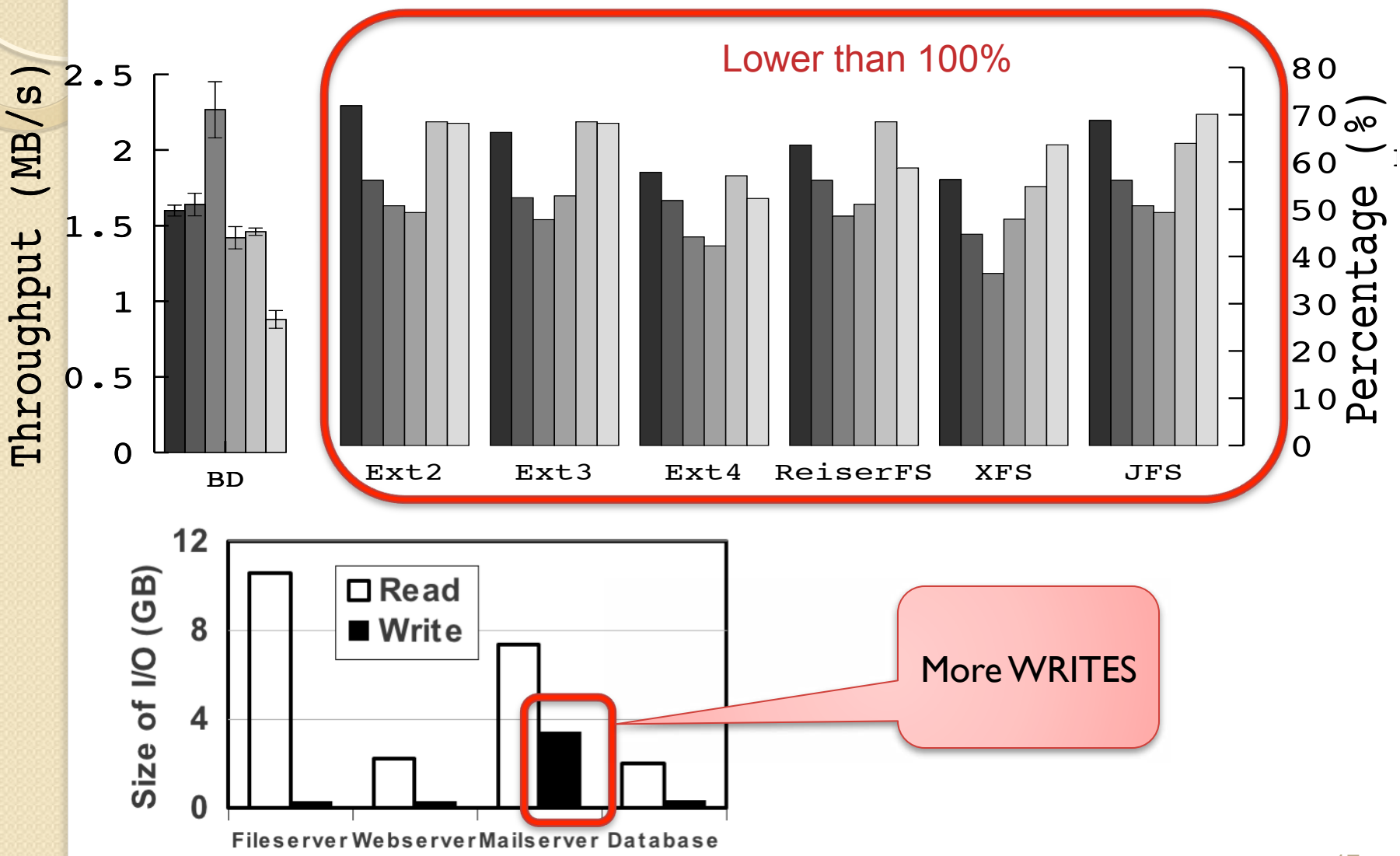
Mail server


Ext2
Ext3
Ext4
ReiserFS
XFS
JFS
Throughput (MB/s)
2.5
2
1.5
1
0.5
0
Percentage (%)
80
70
60
50
40
30
20
10
0
BD
Ext2
Ext3
Ext4
ReiserFS
XFS
JFS
Ext2
guest file system

Macro level observations

- Guest file system → Host file systems
  - Varied performance
- Host file system → Guest file systems
  - Impacted differently
- Right and wrong combinations
  - Bidirectional dependency
- I/Os behave differently
  - Writes is more critical than Reads

Mail server


Throughput (MB/s)
2.5
2
1.5
1
0.5
0
BD
Lower than 100%
Ext2
Ext3
Ext4
ReiserFS
XFS
JFS
80
70
60
50
40
30
20
10
0
Percentage (%)
Size of I/O (GB)
12
8
4
0
Read
Write
Fileserver
Webserver
Mailserver
Database
More WRITES

Macro level observations

- Guest file system → Host file systems
  - Varied performance
- Host file system → Guest file systems
  - Impacted differently
- Right and wrong combinations
  - Bidirectional dependency
- I/Os behave differently
  - WRITES are more critical than READS

Macro level observations

- Guest file system → Host file systems
  - Varied performance
- Host file system → Guest file systems
  - Impacted differently
- Right and wrong combinations
  - Bidirectional dependency
- I/Os behave differently
  - WRITES are more critical than READS
- Latency is sensitive to nested file systems

Outline

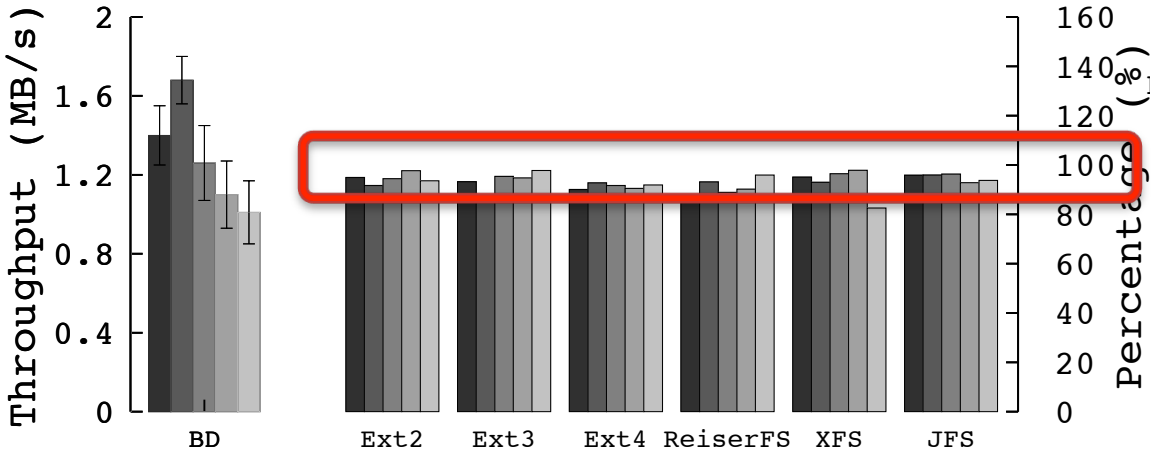
- Experimentations
  - Macro level
- Throughout analysis
  - Micro level
- Findings and Advice

Micro-level analysis

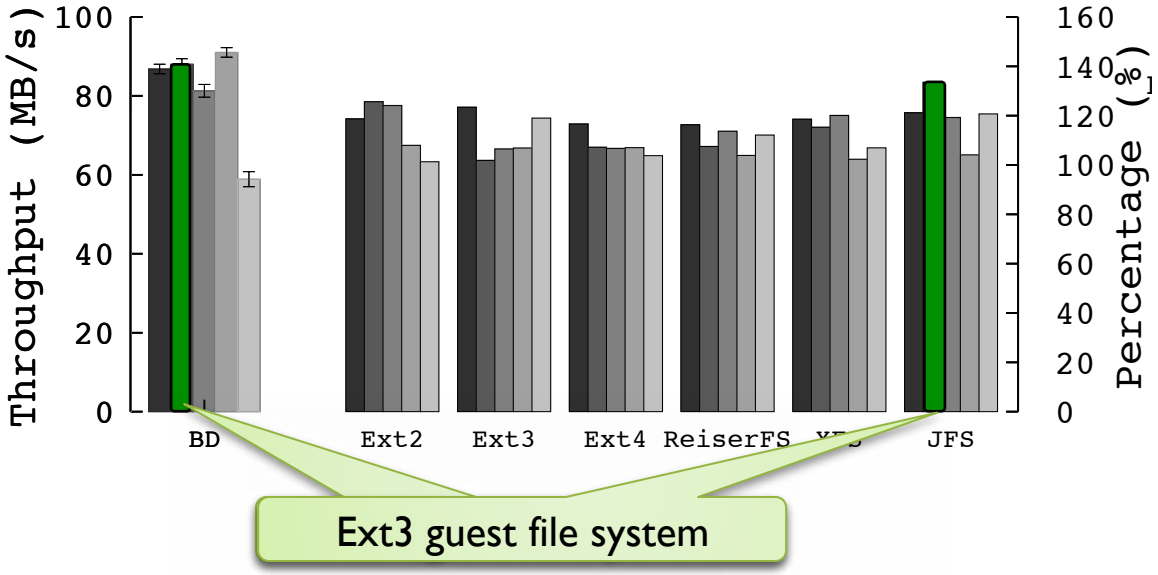
- Same testbed
- Primitive I/Os
  - Reads or Writes
  - Random or Sequential
- FIO benchmark

Description	Parameters
Total I/O size	5 GB
I/O parallelism	255
Block size	8 KB
I/O pattern	Random/Sequential
I/O mode	Native async I/O

Read dominated workloads



Random

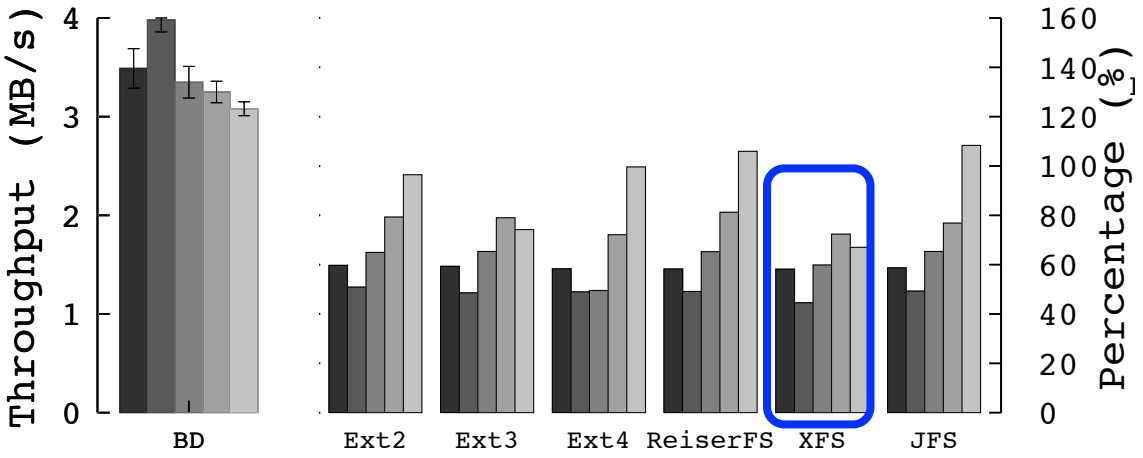


Sequential

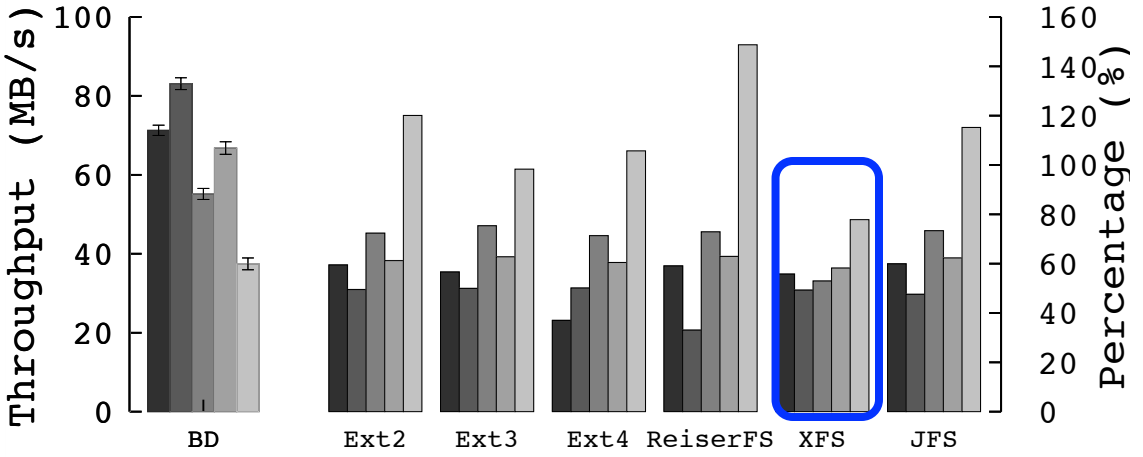
Observations

- Read dominated workloads
  - Unaffected performance by nested file systems
- Write dominated workloads
  - Heavily affected performance by nested file systems

Write dominated workloads



Random

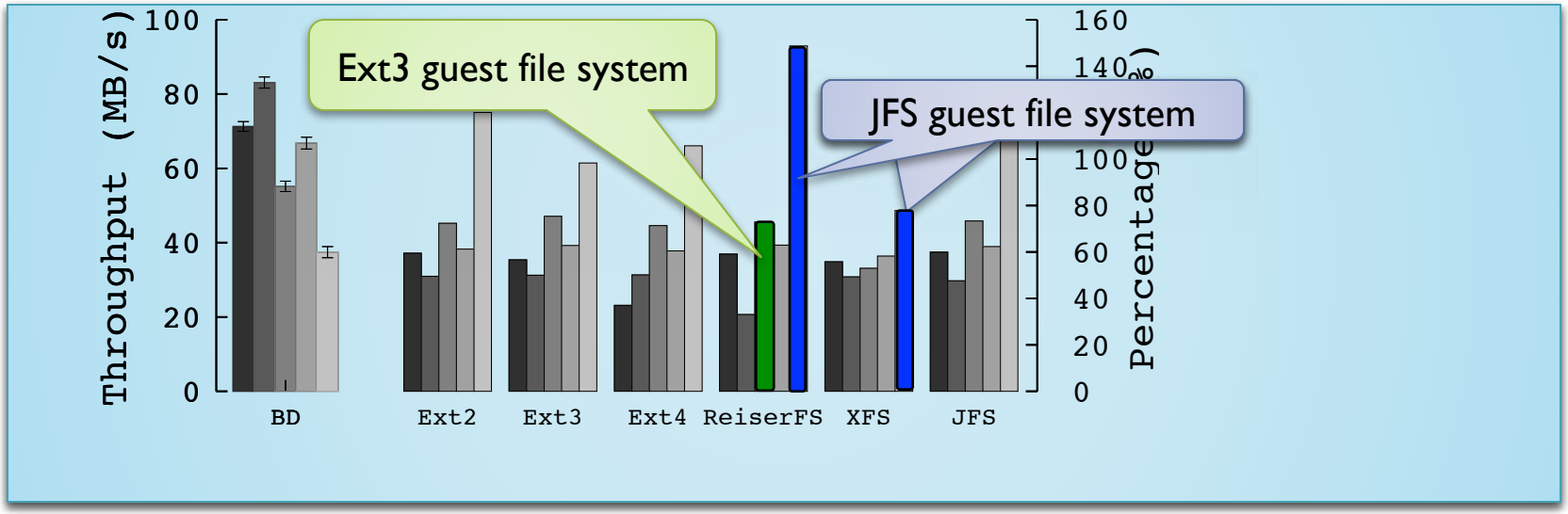


Sequential

Observations

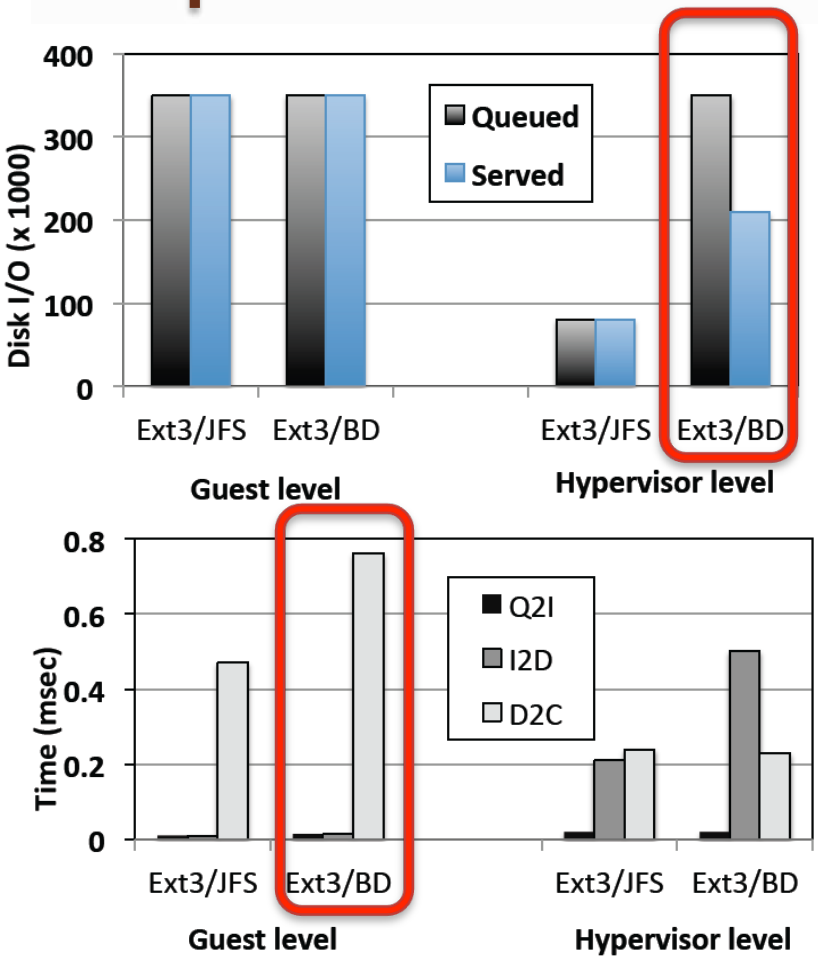
- Read dominated workloads
  - Unaffected performance by nested file systems
- Write dominated workloads
  - Heavily affected performance by nested file systems

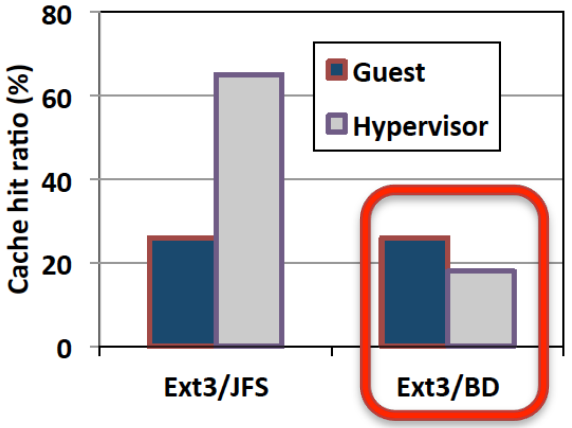
Observations



- Sequential Reads: Ext3/JFS vs. Ext3/BD
- Sequential Writes:
  - Ext3/ReiserFS vs. JFS/ReiserFS (same host file systems)
  - JFS/ReiserFS vs. JFS/XFS (same guest file systems)
- I/O analysis using **blktrace**

Sequential Read Workload



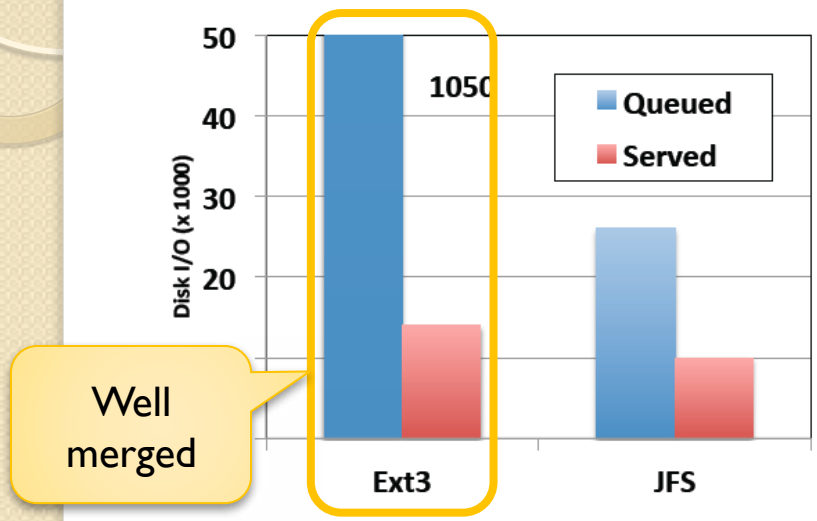


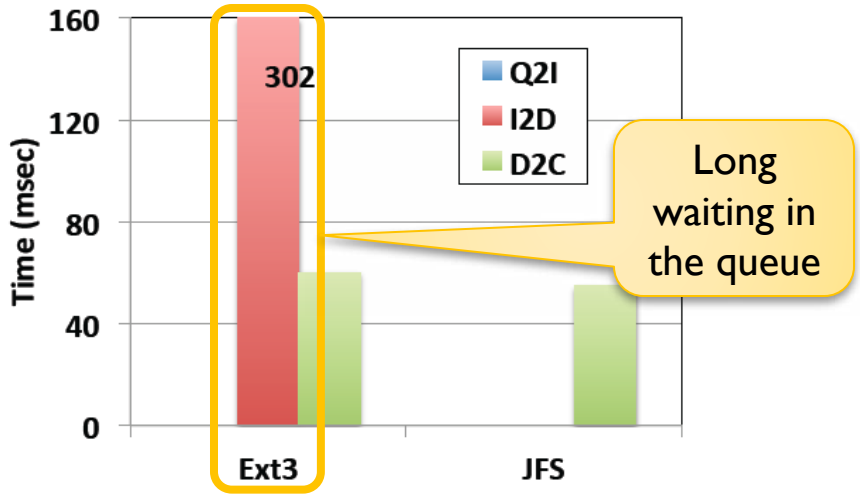
- Findings:
  - Readahead at the hypervisor when nesting FS
  - Long idle times for queuing

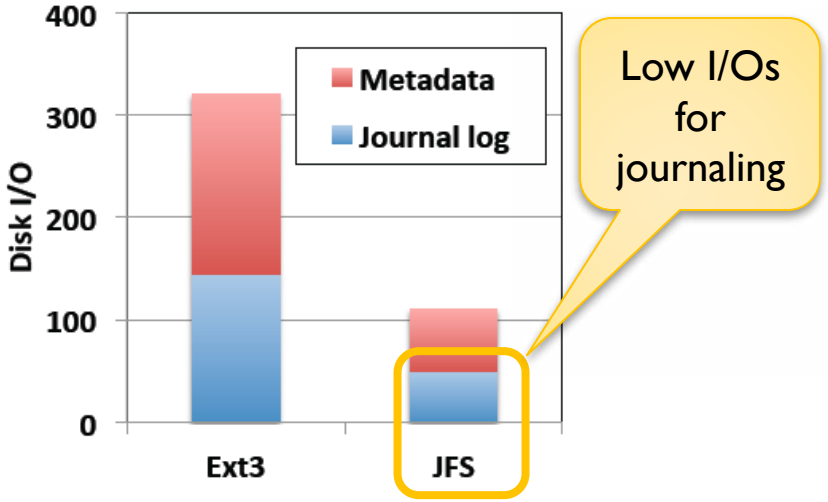
Sequential Write Workload

- Different guests (Ext3, JFS) same host (ReiserFS)
  - I/O scheduler and Block allocation scheme

Ext3/ReiserFS vs. JFS/ReiserFS







- **Ext3** causes multiple back merges
- **JFS** coalescences multiple log entries

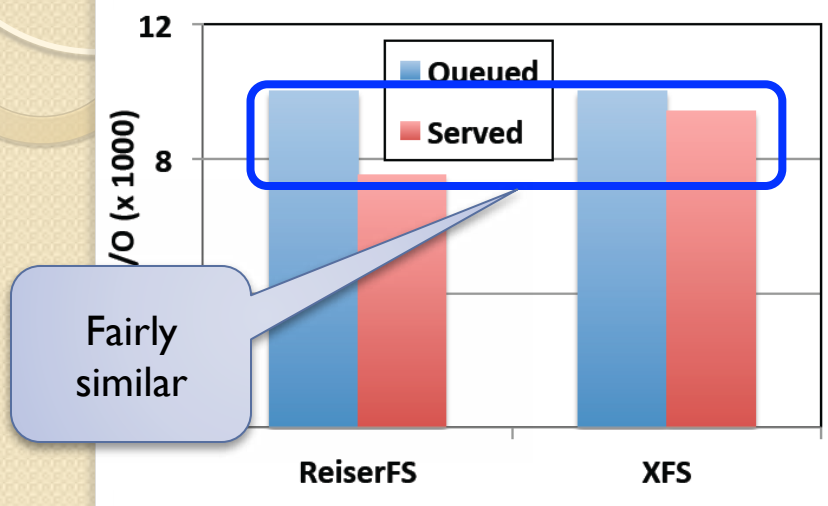
Sequential Write Workload

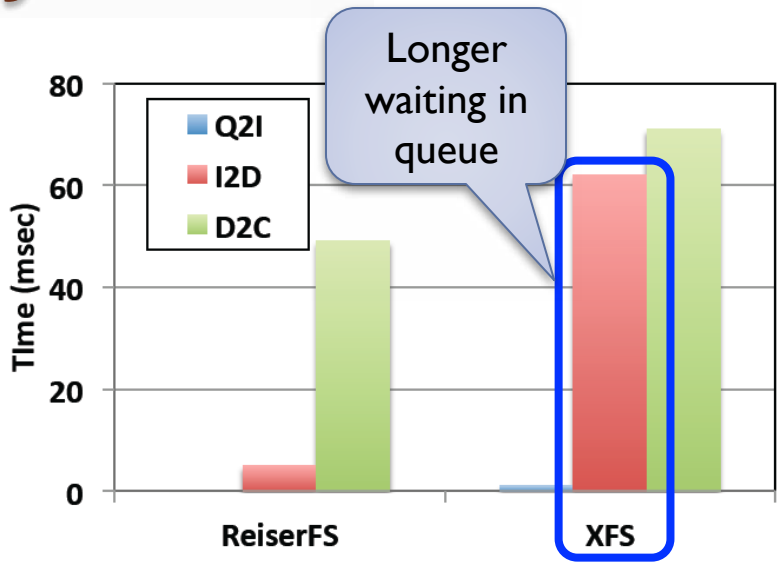
- Different guests (Ext3, JFS) same host (ReiserFS)
  - I/O scheduler and Block allocation scheme
  - Findings
    - I/O schedulers are NOT effective for ALL nested file systems
    - I/O scheduler's effectiveness on block allocation scheme

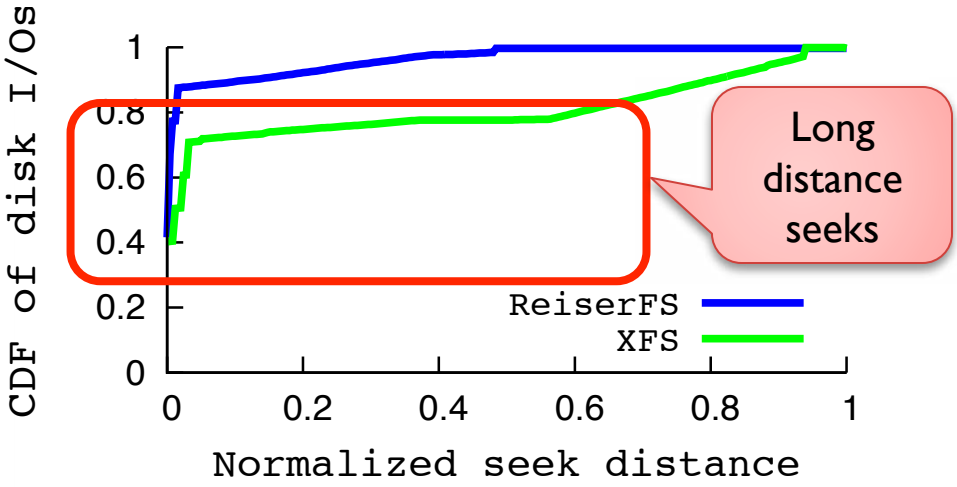
Sequential Write Workload

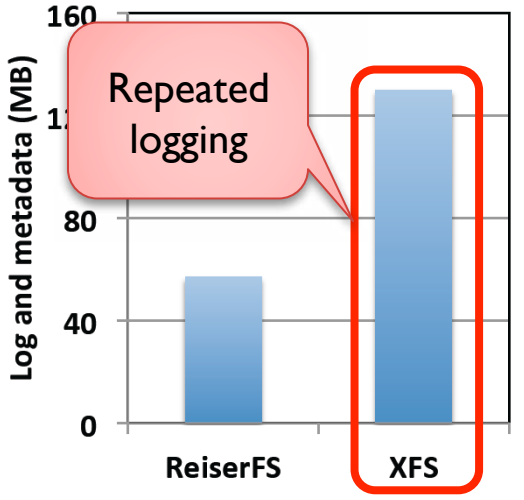
- Different guests (Ext3, JFS) same host (ReiserFS)
- Same guest (JFS) different hosts (ReiserFS, XFS)
  - Block allocation schemes

JFS/ReiserFS vs. JFS/XFS









Sequential Write Workload

- Different guests (Ext3, JFS) same host (ReiserFS)
- Same guest (JFS) different hosts (ReiserFS, XFS)
  - Block allocation schemes
  - Findings:
    - Effectiveness of guest file system's block allocation is NOT guaranteed
    - Journal logging on disk images lowers the performance

Outline

- Experimentations
  - Macro level
- Throughout analysis
  - Micro level
- Findings and Advice

Findings and Advice

- Advice 1 – Read-dominated workloads
  - Minimum impact on I/O throughput
  - Sequential reads: even improve the performance
- Advice 2 – Write-dominated workloads
  - Nested file system should be avoided
    - One more pass-through layer
    - Extra metadata operations
  - Journaling degrades performance

Findings and Advice

- Advice 3 – I/O sensitive workloads
  - I/O latency increased by 10-30%
- Advice 4 – Data allocation scheme
  - Data and Metadata I/Os of nested file systems are not differentiated at host
  - Pass-through host file system is even better!
- Advice 5 – Tuning file system parameters
  - "Non-smart" disk
  - Noatime and nodiratime

Physical disk partitions

Devices	Blocks ($x10^6$)	Speed (MB/s)	Type
sdb2	60.00	127.64	Ext2
sdb3	60.00	127.71	Ext3
sdb4	60.00	126.16	Ext4
sdb5	60.00	125.86	ReiserFS
sdb6	60.00	123.47	XFS
sdb7	60.00	122.23	JFS
sdb8	60.00	121.35	Block Device