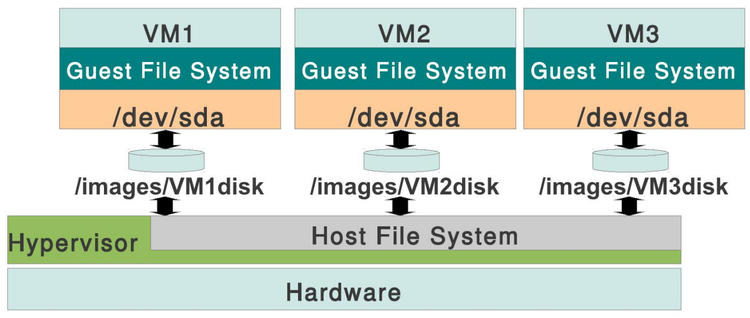
Understanding Performance Implications of Nested File Systems in a Virtualized Environment

Duy Le, Hai Huang, Haining Wang

Motivation and Goals



- **[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

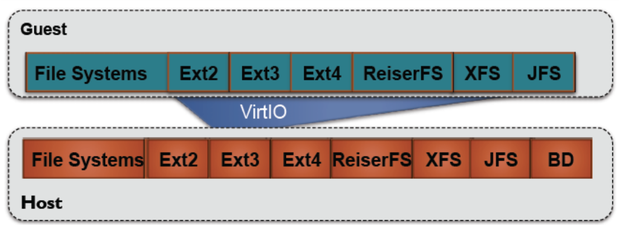
- "Selected file systems are based on workloads"
  - Only true in physical systems
- File system for guest virtual machine depends on
  - I/Os (varied workloads)
  - Deployed file systems at host (disk images, disks)
- What are ***best*** and ***worst*** Guest/Host File System combination?

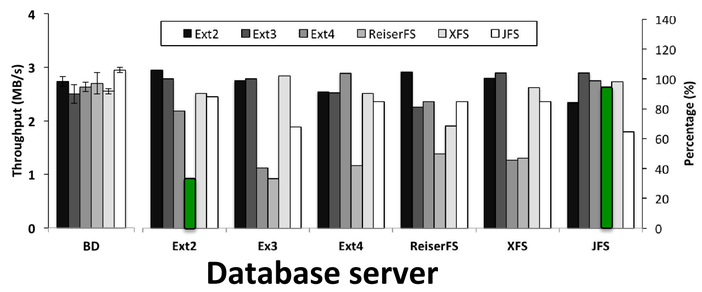
→ **Investigation needed!**

Multiple Guest File Systems / Multiple Host File Systems

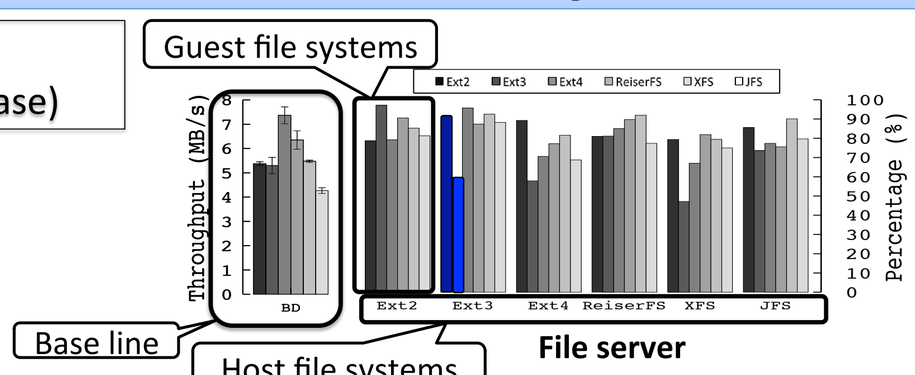
Macro-level Experimentations

- Filebench benchmark
- 4 services (NFS, Mail, Web, Database)



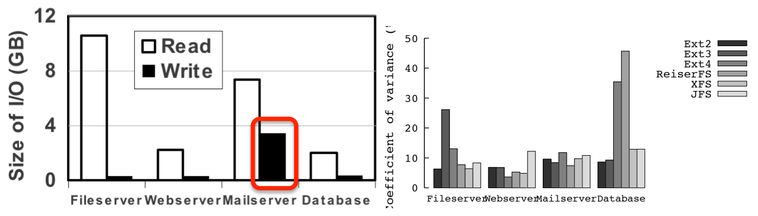


Database server



File server

Mail server



Observations

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

Micro-level Analysis

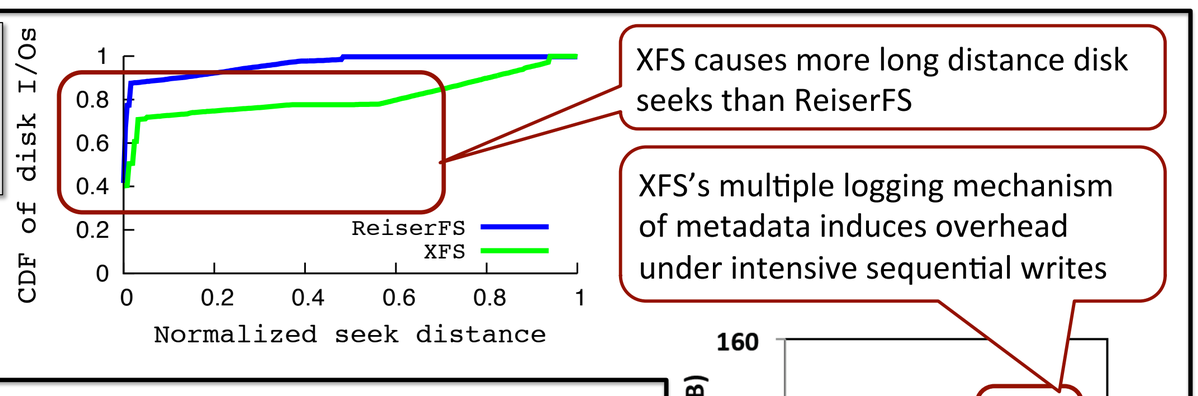
- **FIO** benchmark
- Primitive I/Os (Read/Write and Rand/Seq)

Observations

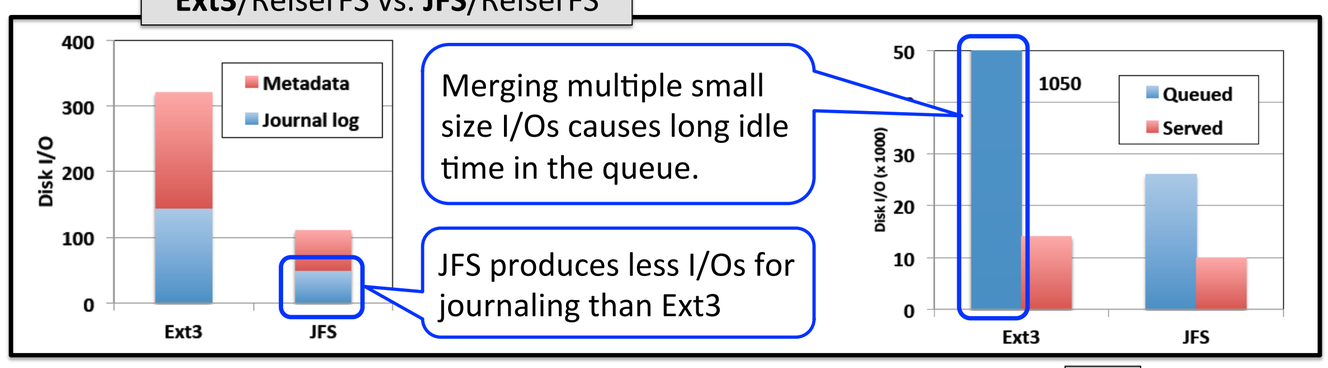
Read-dominated workloads
Unaffected performance by nested file systems

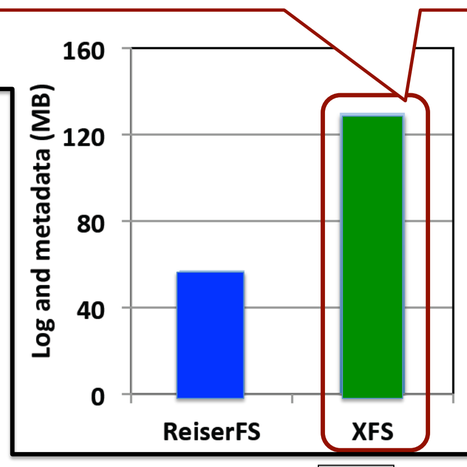
Write-dominated workloads
Heavily affected performance by nested file systems

Sequential Write
JFS/**ReiserFS**
vs.
JFS/**XFS**

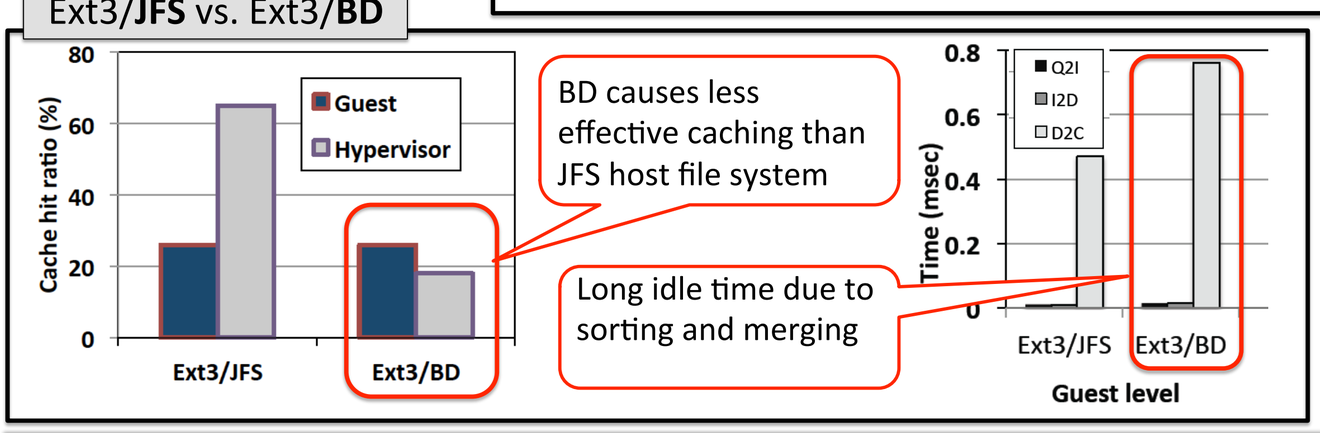


Sequential Write
Ext3/ReiserFS vs. **JFS**/ReiserFS





Sequential Read
Ext3/**JFS** vs. Ext3/**BD**



Findings

- Sequential **Read**
  Readahead at host when file systems are nested
- Sequential **Write**
  I/O scheduler is NOT good for all nested file systems
  Journal logging on disk images lowers the performance
  Effectiveness of guest FS's block allocation is NOT guaranteed

Advice

#1 – Read-dominated workloads
Minimum impact on I/O throughput
Sequential reads: even improve the performance

#2 – Write-dominated workloads
Nested file system should be avoided
Journaling degrades the performance for most workloads

#3 – I/O sensitive workloads
I/O latency increased by 10-30%

#4 – Data allocation scheme
Impossibility to classify guest's data and metadata at host
Pass-through host file system sometimes is good

#5 – Tuning file system parameters
"Discard" disk or access time (noatime and nodiratime)
Data allocation and balancing tasks