

# Linux Filesystems: Who, What and Where?

Stewart Smith [stewart@sun.com](mailto:stewart@sun.com)  
Sun Microsystems Inc

April 22<sup>nd</sup> 2009  
Percona Performance Conference 2009  
MySQL Conference and Expo 2009  
Santa Clara, California, USA

A few ground rules....

The following platforms are  
irrelevant:

Microsoft Windows  
Mac OS X  
Solaris  
FreeBSD

( at least in the context of this talk... )

**We're about filesystems on Linux**

Ext2

ext3

ext4

reiserfs

reiser4

jfs

xf

btrfs

nfs

tmpfs

**For database workloads**

**What is a database workload?**

**MyISAM:**

- Never fsync**
- slowly expanding files**

## InnoDB:

- `O_SYNC`, `fsync()` or `O_DIRECT`
- Extend files in chunks
- few, large files
- OR `file_per_table`

## Binlog:

- “slowly” (or quickly) expanding files
- periodically removed
- can have many
- up to a certain size

## ALTER TABLE:

- FRM file creation
- fsyncs
- lots of sequential IO

## Temp tables:

- short lived
- variable sized (probably small)
- don't care if survives crash

# Structure of a file system

# The i-node

- An i-node describes a file
- A directory is a special case of a file
  - Contains a list of name,i-node number pairs
- Superblock contains the i-node number of the root directory

# Data in an i-node

- Mode (chmod)
- owner, group
- timestamps
- size
- some directions to find out how to get the content of the inode
- extended attributes information

What do we care about?

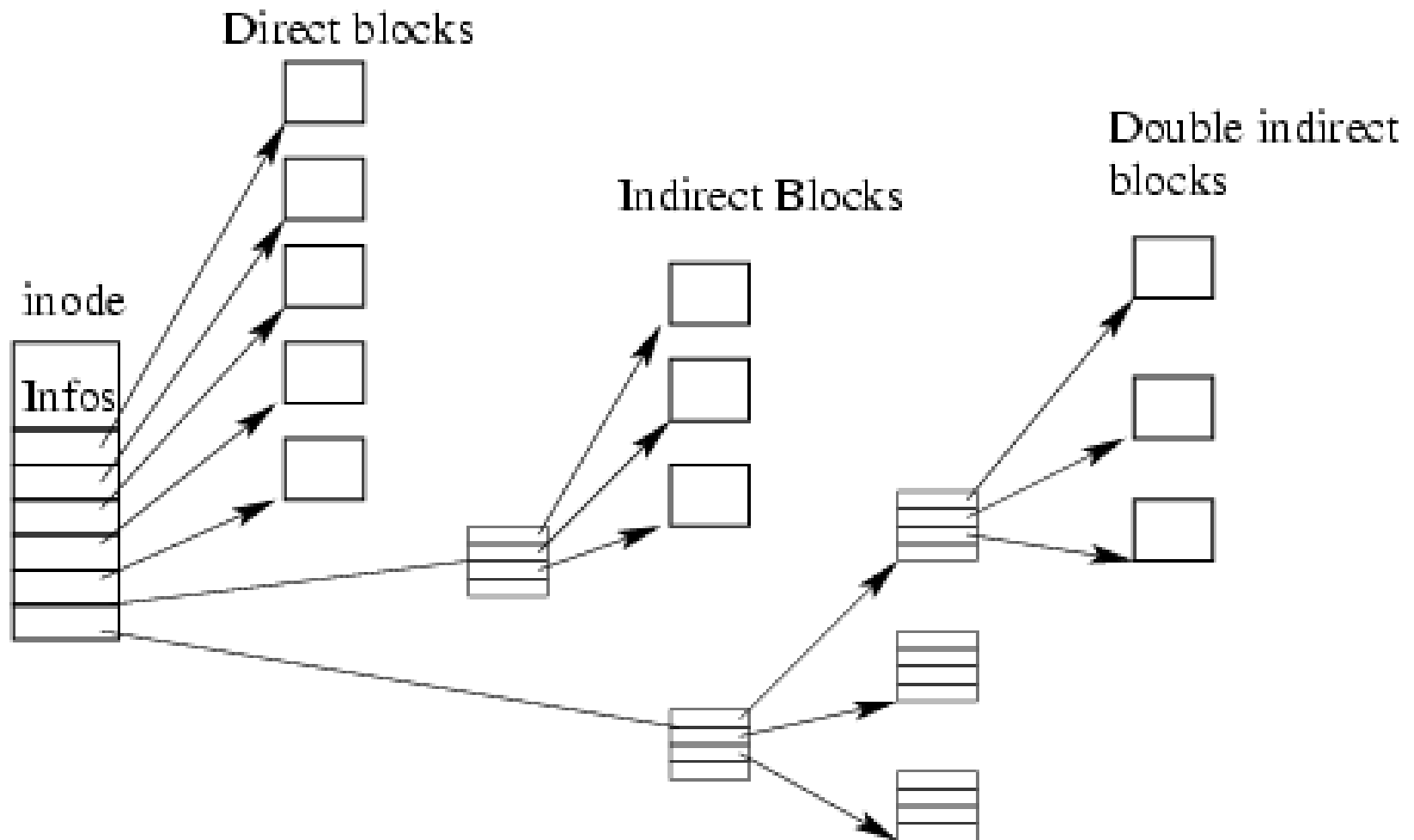
How directories are stored.

(we could have many databases,  
tables)

How blocks are referenced.

(big and small files,  
expanding/contracting files)

# Block Addressing



Crash-Safety, metadata  
performance and `fsync()` behaviour

**SSD vs spinning rust**

Ext2

ext3

ext4

reiserfs

reiser4

jfs

xf

btrfs

nfs

tmpfs