Physical Disentanglement in a Container-Based File System

A Presentation for CS854

Lanyeu Lue,
Yupu Zhang,
Thanh Do,
Samer Al-Kiswany,
Andrea C. Arpaci-Dussueau,
Remzi H. Arpaci-Dussueau

Presented by Krishna Vaidyanathan

March 4, 2016

Table of Contents

Introduction

- Isolation is central to increased reliability and improved performance of modern computer systems.
- We say, for example, that two files are *entangled* if their blocks are allocated using the same bitmap.
- Entanglement mainly arises because:
  - Logically-independent file system entities are not physically independent.

Motivation

- Entanglement can cause three main problems:
  1. Global failure
  2. Slow recovery
  3. Bundled performance

Global Failure

- Single fault leads to a global failure.
- Current file systems crash entire system or mark whole file system read -only.
- For example:
  - Btrfs crashes entire OS when invariant is violated.
  - ext3 marks whole file-system read-only when it detects corruption in single inode bitmap.

Global Failure

- Single fault leads to a global failure.
- Current file systems crash entire system or mark whole file system read -only.
- For example:
  - Btrfs crashes entire OS when invariant is violated.
  - ext3 marks whole file-system read-only when it detects corruption in single inode bitmap.

Global Failures	Ext3	Ext4	Btrfs
Crash	129	341	703
Read-only	64	161	89

Global Failure

- Single fault leads to a global failure.
- Current file systems crash entire system or mark whole file system read -only.
- For example:
  - Btrfs crashes entire OS when invariant is violated.
  - ext3 marks whole file-system read-only when it detects corruption in single inode bitmap.

Global Failures	Ext3	Ext4	Btrfs
Crash	129	341	703
Read-only	64	161	89

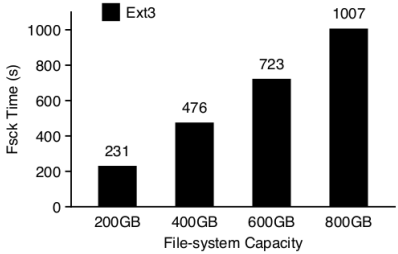
Fault Type	Ext3	Ext4
Metadata read failure	70 (66)	95 (90)
Metadata write failure	57 (55)	71 (69)
Metadata corruption	25 (11)	62 (28)
Pointer fault	76 (76)	123 (85)
Interface fault	8 (1)	63 (8)
Memory allocation	56 (56)	69 (68)
Synchronization fault	17 (14)	32 (27)
Logic fault	6 (0)	17 (0)
Unexpected states	42 (40)	127 (54)

Slow Recovery

- After failure, offline file-system checker scans whole file system.
- Checkers are pessimistic: entire file system checked, when only a small piece of corrupted data.
- Not scalable.

Slow Recovery

- After failure, offline file-system checker scans whole file system.
- Checkers are pessimistic: entire file system checked, when only a small piece of corrupted data.
- Not scalable.

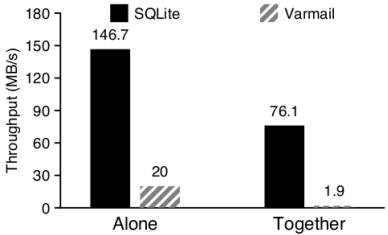


Bundled Performance

- File-systems, like ext3, use a journal to keep track of uncommitted changes, committing at periodic intervals.
- All updates within a short period of time are grouped together.
- Performance of independent processes are bundled.

Bundled Performance

- File-systems, like ext3, use a journal to keep track of uncommitted changes, committing at periodic intervals.
- All updates within a short period of time are grouped together.
- Performance of independent processes are bundled.



Current Solutions

- Namespaces: defines subset of files and directories to be made visible.

Current Solutions

- Namespaces: defines subset of files and directories to be made visible.
  - Fails to address above problems.
  - Files from different namespaces may still share metadata, system states etc.,
- Static disk partitions: Multiple file systems can be created on separate partitions.

Current Solutions

- Namespaces: defines subset of files and directories to be made visible.
  - Fails to address above problems.
  - Files from different namespaces may still share metadata, system states etc.,
- Static disk partitions: Multiple file systems can be created on separate partitions.
  - Single `panic()` or `BUG_ON()` can still crash entire OS.
  - Not flexible, and number of partitions limited.

Usage Scenarios

- Virtual Machines:

Usage Scenarios

- Virtual Machines:
  - Fault isolation of paramount importance.
  - Single fault triggered by one virtual disk can cause host file system to become read-only.
  - Redeployment and recovery require considerable downtime.

Usage Scenarios

- Virtual Machines:
  - Fault isolation of paramount importance.
  - Single fault triggered by one virtual disk can cause host file system to become read-only.
  - Redeployment and recovery require considerable downtime.
- Distributed File Systems:

Usage Scenarios

- Virtual Machines:
  - Fault isolation of paramount importance.
  - Single fault triggered by one virtual disk can cause host file system to become read-only.
  - Redeployment and recovery require considerable downtime.
- Distributed File Systems:
  - Physical entanglement negatively impacts distributed file systems, especially multi-tenant settings.
  - Specifically, HDFS does not provide fault isolation for applications.
  - Scenario: four clients concurrently read different files, and the machine which stores the data blocks crashes.