A Centralized Failure Handler for File Systems

Vijayan Prabhakaran

Andrea C. Arpaci-Dusseau

Remzi H. Arpaci-Dusseau

Failure handling diffusion

- Failure handling in file systems is broken
  - Assumes that disks fail in a fail-stop manner
  - Portions of a disk can fail: latent sector errors, block corruption

- File system I/O calls are distributed
  - System calls (open, stat, etc), flush daemons, journal

- Along with I/O, failure handling is also diffused
  - Detection and recovery for each I/O code

Problems due to diffusion

- Illogically inconsistent policies
  - Different techniques even under similar fault scenarios
- Tangled policies and mechanisms
  - Harder to separate failure policies from detection and recovery mechanisms
  - Policy decision: “To protect using parity or replica?”
  - Mechanisms: “How to implement parity protection?”
- Diffusion of bugs
  - Several bugs in failure handling code
  - Since bugs are repeated, hard to fix them all

Centralized Failure Handler

- Centralized failure handler
  - Detects and recovers with well defined failure policies
- Component of file system like cache manager or journaling layer
- Controls all I/O initiation and completion
- Detects I/O failures and invokes specified recovery policy

Benefits of Centralized Failure Handler

- Eliminate inconsistent policies
- Easy to add new functions
  - No need to write a failure handler for each function
- Can separate failure polices from mechanisms
- Fine grained failure policy: diff block types & I/O contexts
  - Applications can specify their own failure policies
  - E.g., “replicate an important directory but no need for temp file.”

Issues in Centralized Failure Handler

- Information
  - I/O for different block types and contexts
  - Failure handler needs semantic information about I/O
  - Maps: block types and I/O contexts to failure policies

- Architecture
  - Interacts with core file system, journal, cache
  - Two sub components: file system specific and generic

- Machinery
  - All I/O calls go through Centralized Failure Handler
  - I/O calls: time critical, completion specified in interrupt context
  - Contains machinery to separate completion path from failure handling