Solaris pNFS Server
Works In Progress

Jeff Smith
Sun Microsystems

Overview

- How does MDS create an optimal layout?
- When could MDS create a layout?
- When could MDS store a layout?
- What does Solaris MDS favor?

How can MDS create optimal layout?

- Depend on the client admin!
  - > Client SPE policies determine layout hint attribute
- Depend on the server admin!
  - > Server SPE uses policies and layout hint

When to create...

- Speculatively -- when layout hint provided.
  - > OP_OPEN(createattrs) / OP_SETATTR
  - > Asynchronous creation can limit latency impact
- Just-in-time – when layout requested.
  - > OP_LAYOUTGET
  - > Implication for create-time policies
    - >layout hint is not part of OP_LAYOUTGET
    - >MDS needs to cache hint (or ignore it).

When to write...

- When layout is created?
  - Layout data written soon after file creation
  - Minimizes “reboot window”
  - Layout creation latency added to open path.
- When layout is requested?
  - Slightly larger reboot window.
  - Layout creation latency stays in LAYOUTGET
  - Defers layout storage allocation until client requests the layout.

When to write...

- When first state checked by DS?
  - Ultimate lazy approach.
  - MDS avoids allocating storage for layout data until client starts doing IO.
  - Maximizes “reboot window”. YIKES!
    - Implications for layout creation policies.

What does Solaris MDS favor?

- Extend VOP_CREATE() interface
  - > Layout tightly coupled with inode creation.
  - > Reboot window closed.
  - > Layout has has “free ride” to stable storage.
- Async create/write layout data when hint received.
  - > Minimizes latency in open/create path
  - > Makes reboot window very small.
  - > Additional IO needed to store layout data.

Questions?

nfsv41-discuss@opensolaris.org

Solaris pNFS Server Works In Progress

Jeff Smith
Sun Microsystems