pNFS Storage Device Preference

IETF 77 NFSv4 WG Meeting

March 23, 2010

Sorin Faibish sfaibish@emc.com
David Black - EMC
Peng Dai – VMware
Christos Karamanolis - VMware

Outline

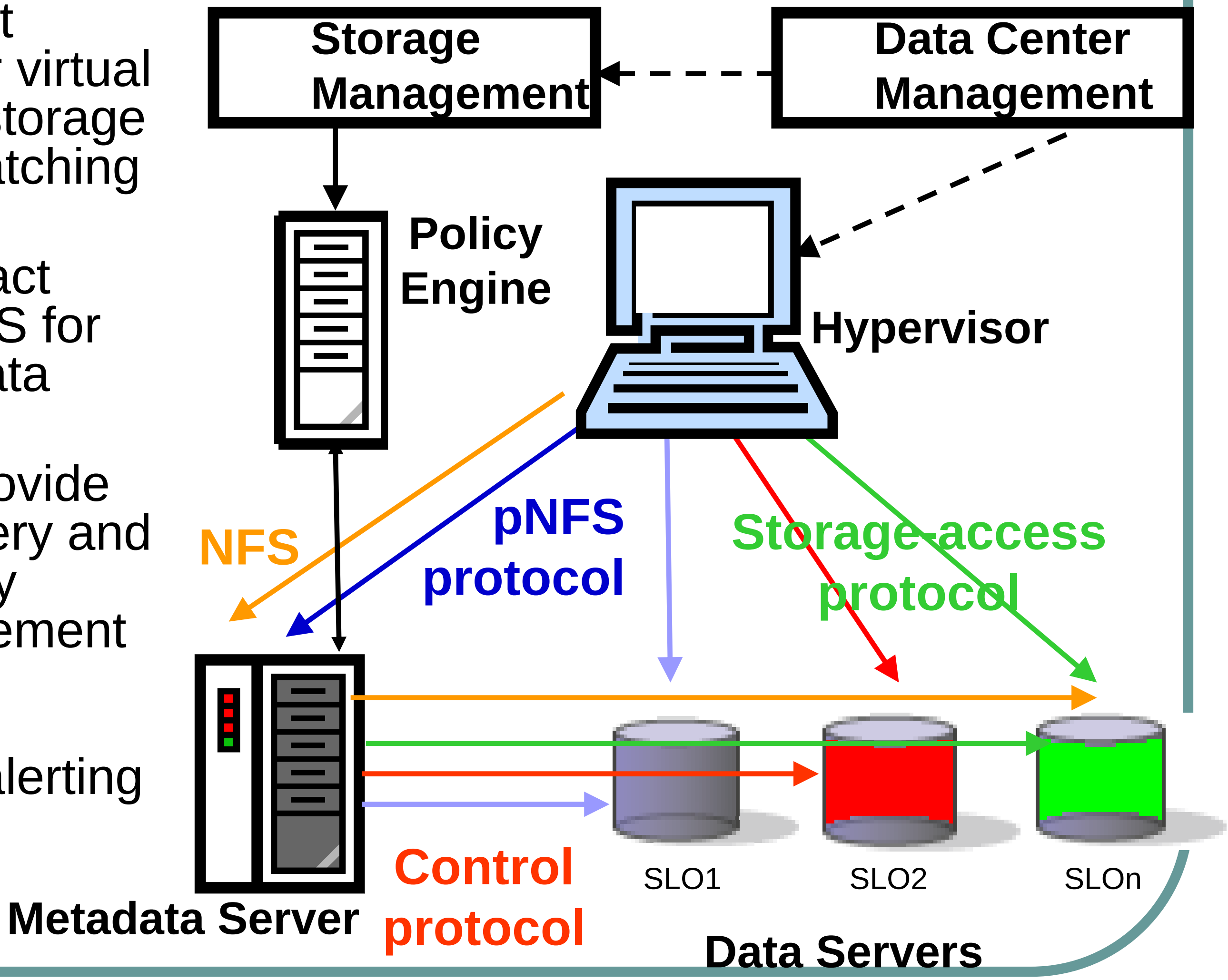
- pNFS protocol futures – where do we go
- Proposed new pNFS Architecture
- Virtualization requirements
- Policy-based management
- Hypervisor Use Cases (VMware view)
- Problem Statement
- Protocol Gaps
- Possible Remedies
- Next Steps
- Questions and Discussion

pNFS Quo Vadis (what pNFS.2)

- NFS and for this purpose v4.2, pNFS.2 (call it next) needs to address the new technological trends of Virtualization, Cloud Computing and Storage Tiering.
- NFSv4.1 and pNFS is addressing very little for the new needs
- Need to add features that will support cloud storage and virtualization to be a viable protocol
- Hypervisors today use non-standard solutions to cope with the virtualization needs; let’s standardize them
- Clouds don't look at using pNFS storage as it has limitations of scalability and management

Proposed pNFS architecture

- VM management specify policy for virtual disk and select storage instance with matching capabilities
- Hypervisor interact with MDS and DS for metadata and data access
- Policy engine provide resource discovery and cataloging, policy based data placement and migration, conformance monitoring and alerting



Virtualization requirements

- Policy-based management of data objects
  - Policy enforcement at virtual disk granularity during the VM’s lifecycle
  - Policy enforcement and monitoring as part of the control protocol
  - Manage policy enforcement metadata
- Scale, distribution and heterogeneity management
  - Single-image system of loosely coupled, distributed, heterogeneous system components
  - Scalable metadata and namespace management
- Generic hypervisor support
  - VMware use cases here, but more generic requirements

Policy-based management

- Descriptive management of application requirements
  - Policy = storage properties ‘required’ by application/VM
  - Capability = range of properties ‘offered’ by storage system
  - End-to-end contract between client and storage
  - Policies semantics opaque to protocol
- In scope for protocol:
  - In-band policy specification per file
  - Grouping hints for files (e.g., files of one virtual disk)
- Out of scope for protocol (at least for now):
  - Expose storage capabilities of any type
  - Monitoring and compliance reporting
  - Policy and metadata consistency (proprietary control protocol)

Hypervisor use cases

- **Provisioning**: lifecycle of a virtual disk starts with the initial provisioning
  - facilitate policy implementation from the onset to minimize data movement
  - hypervisor policy for virtual disks has to be communicated to the storage system at creation time
  - the storage system cannot satisfy the policy, the creation may be failed (Iyer draft is addressing this partially)

To support this use case pNFS needs layout hints specified during file creation to be binding as attributes

Hypervisor use cases (cont.)

- **Data Access**: no data loss tolerated neither data access unavailability
  - write access to the disk has to be synchronous to simulate the real disk semantics
  - write completion response implies the data and related MD on stable storage
  - frequency of metadata operations on virtual disks are different/lower than that of regular user files
  - separated data and metadata paths such that each can scaled independently

Hypervisor use cases (cont.)

- **Compliance**: on-going monitoring/alerting of the policy status and layout change according to SLO
  - storage system to persistently store the policy settings, as part of virtual disk metadata
  - layout change must be implemented seamlessly to the application (in VM)
  - layout change must be selective according to external policy engines requirements
  - include in the policy the action hints for the relevant attributes

Hypervisor use cases (cont.)

- **Change Management**: when business requirements change, the resource demand need to change
  - allocate more or higher quality resources to meet the new service level
  - in the case of down-sizing, ability to free unnecessary resources
  - changing application requirements with no significant application downtime

To support this use case pNFS needs to support changing file attributes on the fly

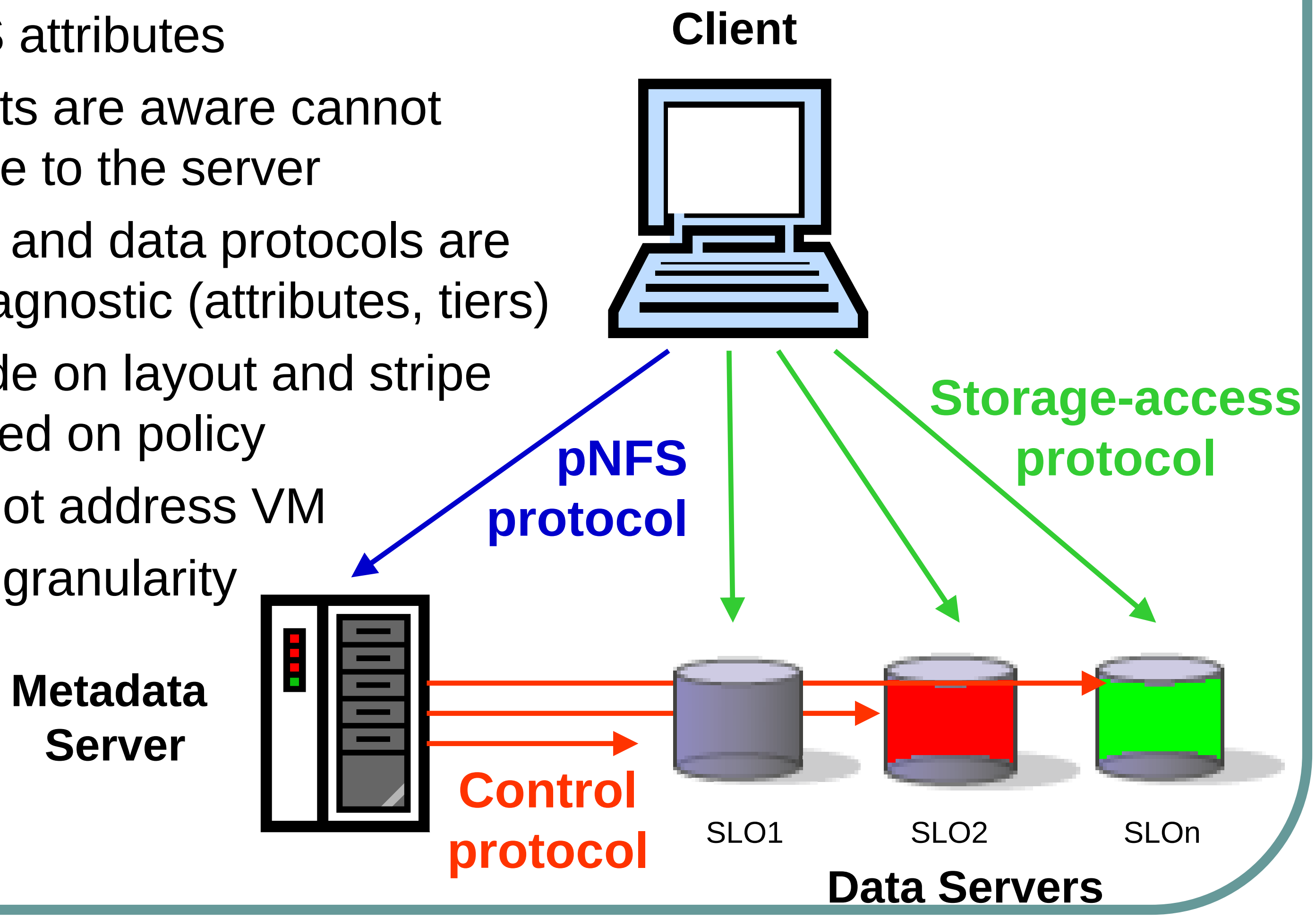
Hypervisor use cases (cont.)

- **Namespace**: separate namespace structure from storage capabilities and access methods
  - namespace structure may be driven by different and sometimes competing priorities
  - virtual disks with drastically different SLOs co-located under same namespace data set
  - files with different access methods co-located under same namespace data set

To support this use case pNFS needs new namespace granularity (VM level)

Problem Statement

- pNFS clients and servers are not aware of DS attributes
- Even if clients are aware cannot communicate to the server
- Both control and data protocols are SLO/policy agnostic (attributes, tiers)
- Server decide on layout and stripe size not based on policy
- FedFS cannot address VM namespace granularity



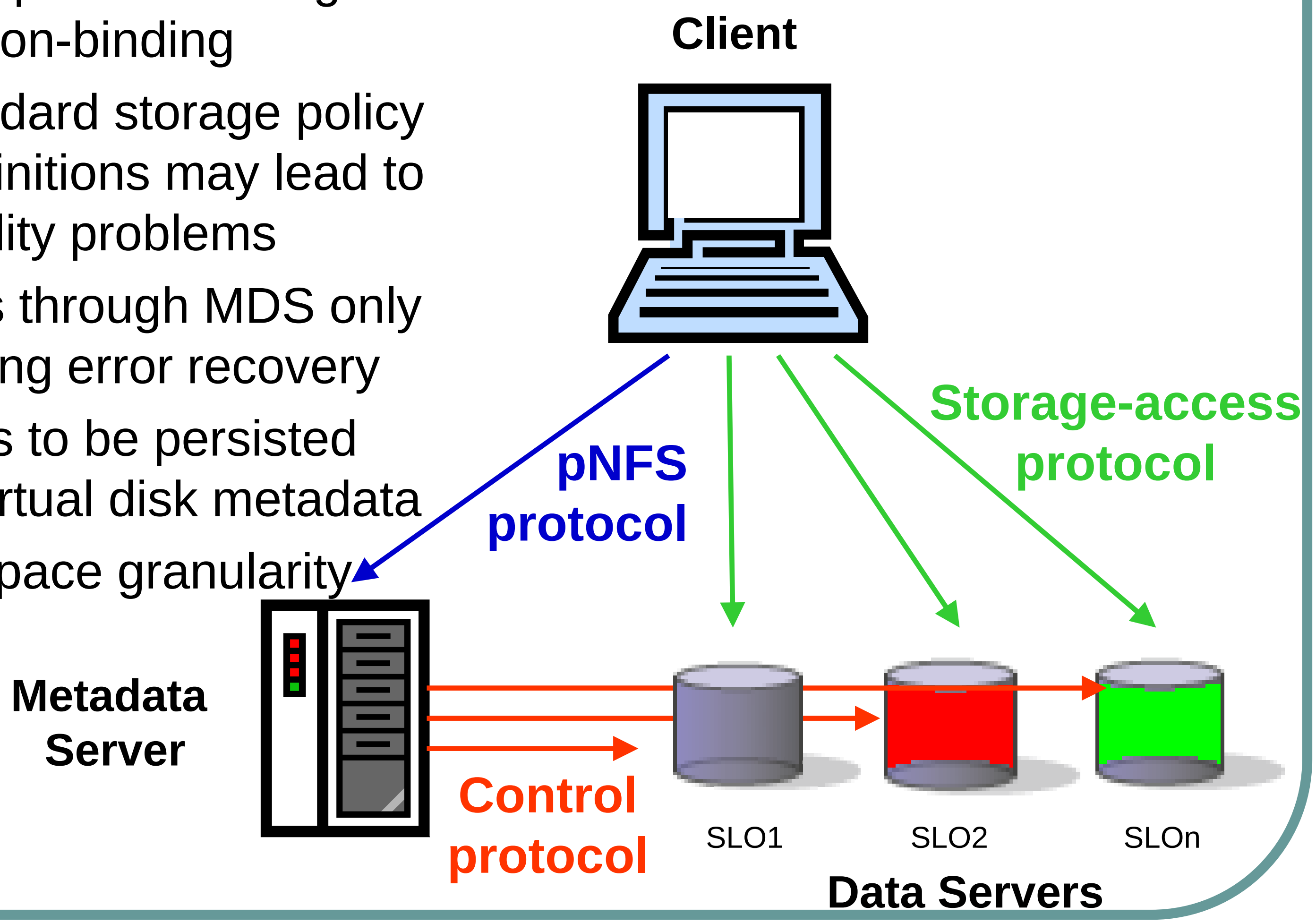
Problem Statement (Summary)

- pNFS allows host applications to support SLO by multiple mounts
- pNFS allows host application SLO when running in VMs with multiple mounts
- Hypervisors can support SLOs using multiple MDS mounts, but
- vMotion cannot work using multiple mounts as not all clients mount same mounts

Need a single mount to support SLOs

Protocol gaps

- Layout hint specified during file creation is non-binding
- Lack of standard storage policy attribute definitions may lead to interoperability problems
- Data access through MDS only allowed during error recovery
- Policy needs to be persisted with other virtual disk metadata
- New namespace granularity



Possible Remedies (new I-D)

- Extend use of hints and extending attributes to include DS SLOs and policy
- Extended use of opaque in all layouts to multi-tier
- Client control of devices for layout - discretionary layout
- Enhance server DB using inputs from external policy engine
- Add special namespace structures aware of file SLOs associated to VMs
- Add mechanism to allow pNFS client to define file stripe for performance SLOs
- Allow client to control layout decisions

Next Steps

Conclusion: we must move the NFSv4.2 in the right direction not only for fixing NFSv4.1 lack of features but support future storage needs.

- Some content for v4.2 is indented to address these needs but is it enough to justify a new minor of v4 (see Dave Noveck presentation).
- Should we only have pNFS.2? (stupid, maybe)
- Adding features needed by hypervisors should be a base for the decision to go to minor 2 or pNFS.2.

Questions and Discussion