SLASH2

File System for Wide-Area Storage Management

Pittsburgh Supercomputing Center

Paul Nowoczynski

Jared Yanovich

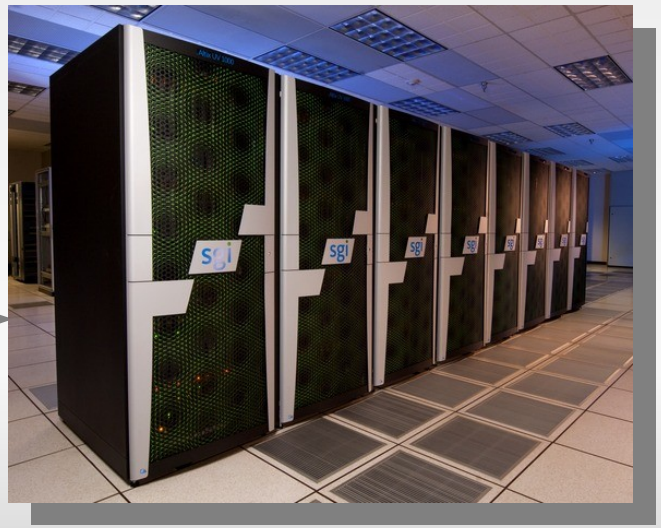
Zhihui Zhang

Need for Wide Area Storage Management

- Geographic replication for large, valuable datasets

  (when one site isn't enough!)

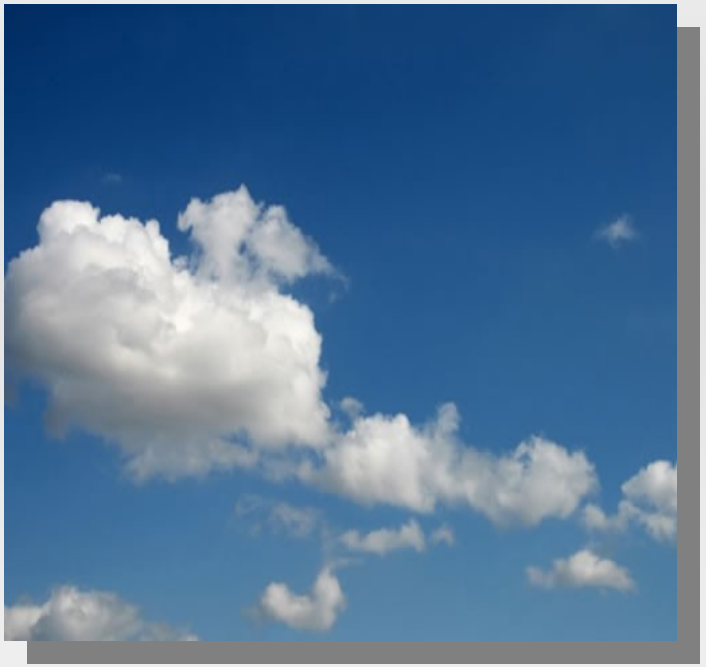
- Data generation and analysis often occur at different sites
  (LHC, LSST, Green Bank, etc.)



Need for Wide Area Storage Management

Cloud Computing

- Maintain storage environment for applications regardless of run-time locale
- Intelligently stage and ship input and output data



Distributed Research Collaboration

- Common work environment, regardless of location

Wide Area Storage Management in HPC

Data Set Migration

- **Archival storage**

  local or remote

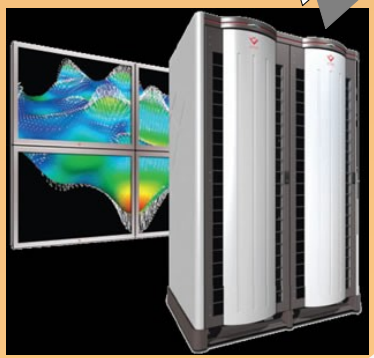
- **Post-processing systems**

  visualization

  data analysis

- **A supercomputer with available cycles**

Site A:
Compute and Archival Storage



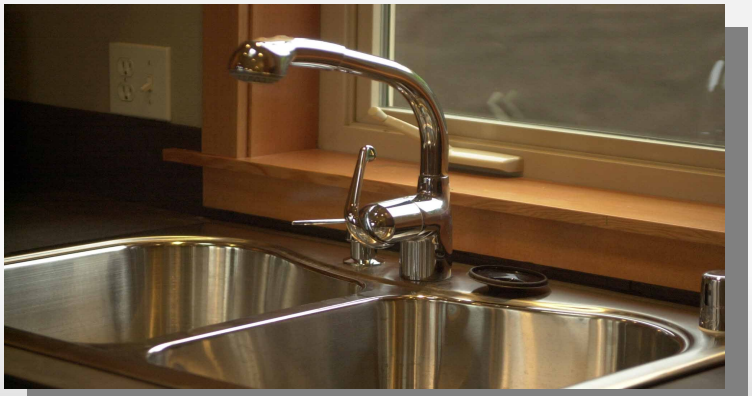
Site B:
Visualization/ post-processing

Site C:
Mass Storage

WAN FS Requirements (objectively speaking…)

What do HPC users/applications want?

- Universal namespace
- Global data access through standard APIs
  - POSIX
- Local performance (when necessary)
  - Implies tight integration with relevant storage resources
- System-managed data transfers
  - Fully utilize storage and network bandwidth
  - No babysitting!



WAN File Systems: The reality

- Providing system level uniformity across disparate storage resources is a tall order
- Many dimensions of heterogeneity exist
  - Varying from the technical to the political

WAN File Systems: The reality

- Heterogeneity across storage resources
- Access network may span institutions/administrative domains with differing:
  - Political requirements
    - Proprietary vendor deployments
  - Management philosophies
  - Technical requirements
    - Resource with characteristics tailored to suit role
    - Interoperability with other systems

Available Solutions and Methods

Either too much or not enough

- High-level data management interfaces
  - Data movers (e.g. GridFTP, `scp`)
  - External replica management
- Low-level interfaces
  - Adapting parallel filesystem technologies to the WAN (MC-GPFS, Lustre)

High-level Storage Management Tools

Burden is placed on the user

- Must learn interface to capabilities
- Dealing with system environment inconsistencies
- Managing data transfers – integrity, error detection & recovery
- Replica management – integrity, etc.
- Stale copies detection, avoidance, & garbage collection

High-level Storage Management Tools

Difficult to achieve good performance

- When transferring multi-terabyte data sets, good performance is critical
- Parallelization or striping across multiple endpoint nodes is necessary, this drastically complicates matters for the typical user
- Detailed knowledge of the network paths and storage architectures at the source and destination is usually required

Limited availability of transparent file operations

- No “global filesystem” which binds storage resources
- Files must be staged in from remote sites

Result: users must become system experts or system experts must aid users

WAN Parallel FS (low-level)

Possibility for system-managed operations exist

- Parallel data migration between sets of OSD's
- Namespace no longer an issue

However, a range of problems exist

- Vendor lock-in/licensing
- Requires strict systems administration procedures between sites
- Increased possibility of cascading outages
- Cannot be integrated with all types of compute or archival resources (portability)
- Provisions to prevent overload of network
- Manual old/new storage system migration

What is SLASH2?

"Portable filesystem layer designed to enable the inclusion of seemingly disparate storage systems into a common domain to provide system managed storage tasks."

or more simply put:

Data management system designed to aid users who frequently deal with large datasets in grid environments.

Key characteristics

- Highly portable – allow many types of storage systems to be integrated
- POSIX filesystem interface
  - Extensions for WAN-specific capabilities
- Object-based
- Distributed metadata realized, not actualized (yet)

SLASH2: Background

- Work started by researchers at the Pittsburgh Supercomputing Center in 2006
- Inspired by distributed archival caching system developed at PSC in 2004
  - MSST '05 paper on SLASH1
- Funding provided by National Archives (NARA) and National Science Foundation

SLASH2 and WAN Storage Management

- Data management activities are performed by the system
  - Monitored by administrators, eliminate users as middlemen
  - Avoid mistakes made by users
- Provides a common storage protocol which may be ported to a large array of systems
  - Regardless of vendor or storage system class
  - Minimizes dependence on proprietary solutions
  - Aid in integration of new storage and retirement of old
- System stored data checksums
  - Essential for detection of corrupt data
  - May be used for proactive scrubbing of data

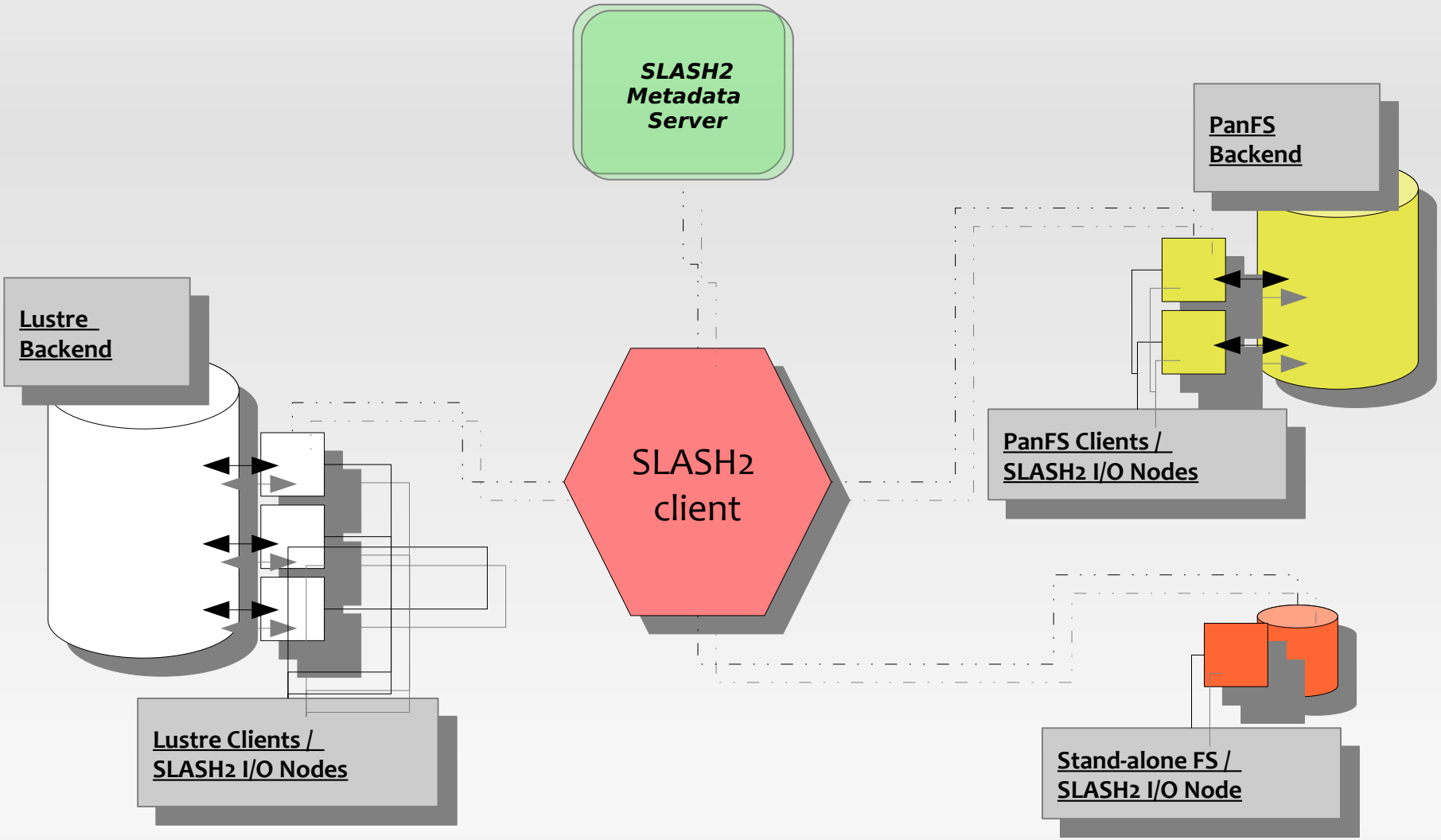
SLASH2 and WAN Storage Management

- Incorporates essential features into the system

  Move complexity away from the user and into filesystem

  - File replication
  - Integrated replica tracking
  - Inline data verification and rectification
  - Detection of faulty storage endpoints
- Provides a single namespace

SLASH2 Architecture Example



SLASH2 Architecture Overview

- Files are divided into 128M (by default) regions called *bmaps* (block maps)
- No locking mechanisms for I/O to avoid scaling limitations
  - Caveat: multiple writers in overlapping file regions may produce undefined results

SLASH2 Architecture: Client

- Runs in userspace via FUSE
  - Easy to debug, non system critical failure, simpler upgrading
- I/O page memcache supporting asynchronous, coalesced writes
  - Multiple writers or single writer with readers switches to direct I/O
- Location aware data retrieval
  - Transparent access to remote I/O servers if no residency
- `msctl` program to access WAN-specific capabilities
- Replication policies: one-time and persistent
- Cached and timeout metadata
- Stateless design: persistent storage only needed for optional local cache

SLASH2 Architecture: I/O Server

- SLASH2 object data stored directly in local filesystem (POSIX)
  - Archiver support needs more work
  - Otherwise agnostic (MacOSX, Linux, BSD, and soon Windows)
- Data checksums fetched from MDS and checked on read
- Checksums computed and send to MDS on write
- SLASH2 I/O nodes mounting same backend file system may be used in parallel
  - File I/O
  - Data Replication
- Low overhead, does not jeopardize system stability
- Stateless design

SLASH2 Architecture: Metadata Server

- Provides name → object ID mapping
- Data residency to bmap level
- Sliver checksums (64-bit CRC per 1MB of data)
- Metadata stored in underlying ZFS file system
- Operation journal for graceful failure recovery
- I/O leases issued on bmaps for coherency with hard timeouts for unresponsive clients
- Write leases invalidate other replicas
- Replica scheduling engine

SLASH2: System-Managed File Replication

Ease of use

- Users specify target files and destination system in a single request:
  "copy file set D to Site C archiver"
- Tools to check status

Behind the scenes

- If a non-recoverable failure occurs, a system administrator is notified – user intervention is not required
- Parallel transfer is used to saturate links when necessary
- Uses an effective load-balancing scheme to ensure that a single slow endpoint does not disproportionally affect performance
- Respect administrative policy of network resource usage
- Tuning parameters are held in the system, users are agnostic
- Automatic integrity verification & continuous retry until user cancel

SLASH2: Integrated Replica Management

- Data replicas are systematically maintained
  - Completed replication update a file's residency state
  - Upon overwrite of file data, old replicas are automatically invalidated by the system
- Replication occurs at block or chunk level, not file level
  - Partial file overwrites will not invalidate entire file replicas, only the affected block(s)
- Intelligent data retrieval (should data replicas exist)
  - Data retrieval requests are sent to most appropriate storage system
  - Corrupt or missing data may be retrieved from alternate replicas transparently to the application

SLASH2: Replication UsageExample

Create a persistent replication policy for all portions of all files under the directory `mydataset` to the resources `io0` and `io1`:

```
$ msctl -R bmap-repl-policy=persist:* mydataset
$ msctl -R new-bmap-repl-policy=persist mydataset
$ msctl -R repl-add:io0@SITE:*,io1@SITE:* mydataset
```

SLASH2 Security

- Currently UID/GID based
- Nominal integrity provided by cryptographic hash signatures
- Authentication details yet to be finalized
  - Kerberos?
- Node authentication
- Needs to support untrusted networks

Ongoing Work: Distributed Metadata

- Preliminary implementation of a distributed metadata system prototype using e*ventual consistency*
  - Aimed at providing reliable but asynchronous namespace mirroring between metadata servers
  - Implements a new algorithm where the metadata servers may modify a single namespace simultaneously without creating conflicts
    - Modification logs are passed amongst the metadata servers and applied in a deterministic fashion
    - Ensures that metadata servers are always "approaching synchronization," though they may not be fully in sync at any given moment

Ongoing Work: Read-only Import

- Provide capability to import data local to an I/O server into the SLASH2 namespace
  - Data is marked read-only
  - Modifications to dataset are lost and SLASH2 may harbor stale data
  - If full SLASH2 features are required, data must be copied

Ongoing Work: Client Local Cache

- Important for WAN environments
  - One copy over the WAN
  - Subsequent accesses to local copy
- Cooperative caching?
  - Nearby clients access each other's caches
  - Likely better to setup local I/O server, but same principle (one copy, modify, push back)

Related Work

- iRODS
- XtreemFS
- DMOVER

Additional Resources

- Web site: *http://speedpage.psc.teragrid.org/slash2*
- Contact e-mail: *advsys@psc.edu*
- Thanks
- Questions?