Active Storage using OSD

John A. Chandy
Department of Electrical and Computer Engineering
Active Disks

• We already have intelligence at the disk
  – Block management
  – Arm scheduling

• Can we use that intelligence for computation?
  – Disk controller can potentially optimize data layout and retrieval since it knows how the data is stored
  – Avoid extra data transfer across networks, buses, or interconnect
  – Parallel disks get you concurrency
Active Disks

Active Disks

- Riedel et al.
  - Data mining
  - Image processing

- Acharya et al.
  - Database queries
  - Streaming model
Active Disks

- Acharya et al.
  - Code fragments are called disklets and they operate on streams of data
  - Three types of streams:
    - Disk-resident streams
    - Host-resident streams
    - Pipe streams
  - Communication between a disklet and its environment is restricted to its input and output streams.
  - The source and sinks for these streams are specified by the host-resident program as a part of the installation of the disklet.
Performance of active disks

From Acharya et al, *ASPLOS, 1998*
Active Disks

- Vendors haven’t provided mechanisms for active disks yet
- Complexity not worth it in commodity disks
- Clusters provide the parallelism right now
Object Disks

• Intelligence at the disk can also be used to offload some of the metadata processing
  – Allocation of disk blocks is easier at the drive
  – The focus of interest is an object
    • Analogous to an inode
    • Not named like a file, instead identified by 128-bit ID

• Separate metadata server provides
  – Object location (filesystem tree)
  – UNIX level permissions

• CMU Network Attached Secure Disk (NASD)
Object Based Storage Architecture

From “Working draft SCSI object-based storage device commands (OSD-2)”
Object Disks

- OSD type cluster storage nodes
  - Panasas, Lustre, PVFS

- Unlike active disks, object disks are now a reality
  - SCSI T-10 OSD (Object-Based Storage Device)
  - Reference implementations from Intel, DISC, and IBM
  - No real disks yet, but coming
Active Disks

• Can we use OSDs to make Active Disks a reality?
  – Application-aware storage
    • Object attributes can give hints to the disk
    • Application specific
  – Parallel File Systems
    • Felix et al. added a filtering layer to Lustre to provide active processing
  – T10 OSD?
Challenges

- What is the programming model?
- How do you download code to the OSD target?
- How do you execute code on the OSD?
- What are the security/capability implications?
Active Disks using OSD

• Programming Model
  – Object-oriented
    • Attach object types to storage objects
    • Define methods for object types
  – RPC
    • Call methods on OSD remotely
Active Disks using OSD

• Example:
  – Record list object

List {
    addRecord();
    sortRecords();
    searchRecord(string);
};
Active Disks using OSD

- Distribute List across multiple objects - one per OSD target
- Client has a single unified List view
  - Proxy class coordinates methods on client’s List with method calls on OSD Lists
  - Client proxy manages objects in distributed List
Active Disks Using OSD

• Code is written in Java

• Advantages
  – Object-oriented
  – Machine-independent
  – Secure/Safe
  – Easy to move from client to target
Active Disks using OSD

• How do you move Java code from client to target within OSD framework?

• OSD objects allow users to set attributes on an object

• We set an attribute on each object where the attribute is the .class file or .jar file associated with the object methods
  – One code attribute page with multiple code attributes
  – Allows different users to define their own code
Active Disks using OSD

How do you execute the method remotely within the OSD framework?

- Modify the READ command so that we can invoke a method
- Recent change to OSD spec allows for command attribute pages - i.e. attribute pages that are only valid for the life of the command

- Introduced by Pete Wyckoff/OSC for support of scatter/gather
- We use the command attribute page to specify the method and parameters

From T10/08-091r0 proposed changes to OSD-2

<table>
<thead>
<tr>
<th>Bit</th>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>Reserved</td>
<td>DPO(^a)</td>
<td>FUA(^a)</td>
<td>ISOLATION (see 5.2.5)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>CAP(^b)</td>
<td>Reserved</td>
<td>GET/SET CDBFMT(^c)</td>
<td>Command specific options</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>(MSB)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>23</td>
<td></td>
<td>PARTITION_ID (see 5.2.7)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>31</td>
<td>(MSB)</td>
<td></td>
<td>USER_OBJECT_ID (see 5.2.11)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>(MSB)</td>
<td></td>
<td>LENGTH (see 5.2.6) or ALLOCATION LENGTH (see 5.2.2)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>40</td>
<td>(MSB)</td>
<td></td>
<td>STARTING BYTE ADDRESS (see 5.2.9)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>47</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>48</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\(^a\) Introduced to support remote execution
\(^b\) Introduced to support scatter/gather
\(^c\) Allowed to add additional options to commands

From T10/08-091r0 proposed changes to OSD-2
### Command attribute page

<table>
<thead>
<tr>
<th>CODE ATTRIBUTE NUMBER</th>
<th>METHOD IDENTIFIER</th>
<th>NUMPARAMS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PARAM_1_LEN</td>
<td></td>
</tr>
<tr>
<td></td>
<td>PARAM_1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>...</td>
<td></td>
</tr>
<tr>
<td></td>
<td>PARAM_N_LEN</td>
<td></td>
</tr>
<tr>
<td></td>
<td>PARAM_N</td>
<td></td>
</tr>
</tbody>
</table>
Simple List example

- Sorting on up to 12 OSDs

![Graph showing comparison between OSP and TCP/IP for sorting up to 12 OSDs]
Active Disks using OSD

- **Issues**
  - What if we don’t know the amount of data that the method will return?
  - SCSI requires that the client request a fixed amount of data and the device must return just that amount of data - no more, no less
  - Probably require two READ commands
    - First command returns size and a cookie
    - Second command uses cookie
    - Requires state at the target
Active Disks using OSD

• Issues
  – Safety
    • Should we place limits on active disk computation?
      – CPU, memory, disk
  – Security
    • Capabilities can be used to protect access to commands
      – Application-specific
      – User-specific
      – System-wide
Active Disks using OSD

- What about other command data patterns?
  - Read only
  - Write only
  - Read/write
  - Read object - write to another object
  - Read collection - write to another collection
Active Disks using OSD

• Other issues
  – Is OO RPC the right model?
    • We are looking into functional models like MapReduce/Hadoop
  – What system services should the OSD target provide?
    • Local file system access?
    • Process/Thread management?
  – Application adoption is needed
    • Database and search are the most promising
Further directions

• Hardware Acceleration
• Active Storage Networks
Hardware acceleration

- Is it possible to dedicate some reconfigurable hardware space on the object disk to do hardware acceleration
  - Encryption functions
  - Image processing
  - Other application-specific hardware
Hardware acceleration

Hardware acceleration

• Use same OSD mechanisms to
  – Download hardware bitstreams
  – Execute hardware methods

• Issues
  – No standard for hardware bitstreams
  – Are there security issues?
Active Storage Networks

- Can intelligence in a storage network
  - Accelerate computation?
  - Accelerate storage access?

- Motivation
  - Active Disks
    - Intelligence at the disk can distribute computation to parallel disks
    - Process data in streams
    - Disks only have local view of data
  - Active Storage Network
    - Network has a global view of data
    - Distributed caching of file system metadata and data
    - Redundancy optimizations
Active Storage Network
Active Storage Networks

• Application operations
  – Reduction operations
    • Database queries
      – SELECT … ORDER BY .. LIMIT $k$
      – $O(nm)$ for normal disks
      – $O(n + km)$ for active disks
      – $O(n + k \log m)$ for active disks with ASN

• Scientific applications
  – MPI_SUM, MPI_MAX, etc.
Active Storage Networks

• Application operations
  – Transformational operations
    • Sorting
      – $O(mn \log mn)$ for normal disks
      – $O(n \log n) + O(mn \log m)$ for active disks
      – $O(n \log n) + O(n \log m)$ for active disks with ASN

• Scientific applications
  – Matrix transformations

• Stream-based
  – Video editing
File System Caching

- Centralized cache frees up memory at the clients
- Metadata caching reduces access to metadata server
  - File layouts
  - Directory lookups
  - File access attributes
ASN Switch Implementation

- Embedded nodes in the switch
  - Processor directly on the backplane
  - FPGAs

- Downloadable functions (netlets)
  - Software functions on embedded processors
  - Hardware functions on FPGAs

- Can we do this in OSD?
  - Current implementation is very system specific

- Can we process iSCSI packets on the fly?

- How do we handle OSD capabilities?
Summary

- OSD implementation of Active Storage
- Shows promise
- Future directions in hardware acceleration and ASN
- Students
  - Tina John and Anu Thiruvenkata Ramani