jSCSI

A Java iSCSI Initiator

Marcel Waldvogel
University of Konstanz
1340
Agenda

> Introduction
  – iSCSI revisited
  – YAlI: yet another iSCSI initiator?

> Current work
  – Device interface
  – Implemented devices
  – Whiskas
  – Lucene backend
  – Benchmarks

> Future work
iSCSI revisited

> “SCSI over TCP/IP”

> RFC 3720 (April 2004)

> cheaper than Fibre Channel
  – Fibre Channel going IP

> no infrastructure changes
iSCSI revisited
YAll: yet another iSCSI initiator?

- First Java iSCSI initiator
- Platform-independent device
- Avoids kernel loop
- Education
- Terabyte-sized iSCSI RAID on your LAN
- Multiple Connections per Session
  - No target available
Device interface

- Multithreaded
- Synchronous execution
- Stackable devices

### Device

+ open() : void
+ getName() : String
+ getBlockSize() : int
+ getBlockCount() : long
+ read(address : long, data : byte[ ] ) : void
+ write(address : long, data : byte[ ] ) : void
+ close() : void
RAID 0, 1

- Raid0Device
  - JSCSIDevice
  - JSCSIDevice

- Raid1Device
  - JSCSIDevice
  - JSCSIDevice
Write Buffering, Prefetching

- PrefetchDevice
  - JSCSIDevice

- WriteBufferDevice
  - JSCSIDevice
Example of a Device Tree
Lucene Backend

> Apache Lucene
  - text search engine library
  - high-performance, full-featured

> Device-based backend for Lucene

> Full-text index on raw iSCSI target

> jSCSI vs. filesystem backend
  - use
  - performance
  - complexity of setup and maintenance
### Benchmarks

<table>
<thead>
<tr>
<th>Description</th>
<th>unit</th>
<th>min</th>
<th>max</th>
<th>avg</th>
<th>stddev</th>
<th>conf95</th>
<th>runs</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>... Read 40kB</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>jSCSI</td>
<td>ms</td>
<td>2</td>
<td>14</td>
<td>3,34</td>
<td>1,32</td>
<td>[3,08;3,60]</td>
<td>100</td>
</tr>
<tr>
<td>Open-iSCSI</td>
<td>ms</td>
<td>4</td>
<td>15</td>
<td>5,29</td>
<td>1,32</td>
<td>[5,03;5,55]</td>
<td>100</td>
</tr>
<tr>
<td><strong>... Read 400kB</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>jSCSI</td>
<td>ms</td>
<td>22</td>
<td>24</td>
<td>22,48</td>
<td>0,52</td>
<td>[22,38;22,58]</td>
<td>100</td>
</tr>
<tr>
<td>Open-iSCSI</td>
<td>ms</td>
<td>5</td>
<td>16</td>
<td>6,79</td>
<td>3,19</td>
<td>[6,17;7,41]</td>
<td>100</td>
</tr>
<tr>
<td><strong>... jSCSI write 8MB</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>No RAID, 1 target</td>
<td>ms</td>
<td>3889</td>
<td>6987</td>
<td>4347,55</td>
<td>398,42</td>
<td>[4322,86;4372,24]</td>
<td>1000</td>
</tr>
<tr>
<td>RAID 0, 2 targets</td>
<td>ms</td>
<td>2346</td>
<td>4947</td>
<td>2803,46</td>
<td>330,12</td>
<td>[2783,00;2823,92]</td>
<td>1000</td>
</tr>
<tr>
<td>RAID 1, 2 targets</td>
<td>ms</td>
<td>4663</td>
<td>9447</td>
<td>5313,72</td>
<td>487,65</td>
<td>[5283,50;5343,95]</td>
<td>1000</td>
</tr>
<tr>
<td><strong>... Lucene &quot;Getting started&quot; [9]</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Build file system index</td>
<td>ms</td>
<td>523</td>
<td>1270</td>
<td>676,40</td>
<td>153,80</td>
<td>[609,00;743,80]</td>
<td>20</td>
</tr>
<tr>
<td>Build jSCSI index</td>
<td>ms</td>
<td>6363</td>
<td>7889</td>
<td>6927,95</td>
<td>468,25</td>
<td>[6722,73;7133,17]</td>
<td>20</td>
</tr>
<tr>
<td>Search file system index</td>
<td>ms</td>
<td>2</td>
<td>68</td>
<td>7,05</td>
<td>14,09</td>
<td>[0,87;13,23]</td>
<td>20</td>
</tr>
<tr>
<td>Search jSCSI index</td>
<td>ms</td>
<td>5</td>
<td>10</td>
<td>9,45</td>
<td>1,36</td>
<td>[8,85;10,05]</td>
<td>20</td>
</tr>
</tbody>
</table>
Summary

> Device hierarchy with Stream flexibility
> Multithreading
> Avoids kernel loop
> Extensible platform
> Performance OK
Additional features for jSCSI 2.0

- Asynchronous I/O semantics for the Device interface
- Improved prefetcher and write buffer
- Support for multiple pending operations per connection
- Smaller memory footprint and less CPU consumption
- Resizable devices (ZFS storage pool)
- Write balancing
  - space usage
  - activity
  - latency
Marcel Waldvogel
waldvogel@uni-konstanz.de

University of Konstanz
http://www.inf.uni-konstanz.de/disy/