Reliability Algorithms for Network Swapping Systems with Page Migration
Benjamin Mitchell, Julian Rosse, Tia Newhall, Swarthmore College, Swarthmore PA

www.cs.swarthmore.edu/~newhall/nswap.html

The Problem

Reliability for Network Swapping Systems with:

(1) migration of remotely swapped pages
(2) dynamic re-sizing of nodes’ available RAM for storing remotely swapped pages

Reliability schemes that require fixed placement of page & reliability data or fixed size “backing store” are not applicable

(if pages in same parity group migrate to same remote server, then lose reliability for that parity group)

Network Swapping: Cluster nodes use the remote idle memory of other nodes as their “swap device” rather than their own disk
+ network speeds faster than disk speeds
+ usually a significant amount of idle RAM in cluster

Reliability: recovery for swapped page data necessary:
• node failure is common in larger clusters
• a single node crash can affect programs on other nodes by losing their remotely swapped page data

Mirroring Algorithm

• Each swap-out results in 2 pages being sent, 1 to primary and 1 to backup server (any 2 servers will do, no static pairing)
• On swap-in, if primary server unreachable, get backup from other server
• On page migration care must be taken to ensure that both copies don’t end up on the same remote server

Nswap servers use LOW PRIORITY MIGRATE (less time critical than normal MIGRATE, so care can be taken to migrate pages to restore reliability)
+ easy to implement
+ avoids using disk for reliability data
+ little time overhead to normal swapping - uses twice as much network RAM to store remotely swapped page data

Dynamic Parity Algorithm

RAID-based Reliability Solution
+ Time Efficient: uses remote idle RAM to store reliability data
+ Space Efficient: uses much less idle RAM than mirroring

Architecture

Dedicated Parity Server for regular Nswap Client-Server nodes
Large Clusters into Parity Sets, each w/dedicated Parity Server

Dynamic Parity Groups

Parity Group membership determined by page’s swap slot & host IP
+ can be computed on-the-fly: client/server don’t need to store
The size of a parity group changes as pages are swapped in/out:
(if 3 pages in same group are currently swapped out, size is 3 if later only 2 in group swapped, then size is now 2)

On Swap Out:

Nswap server: sends swapped page & metadata to Parity Server
+ does not slow down the client-side of swapping
Parity Server: updates Parity Page for received page & metadata
• May increase Parity group size by one
• Must detect and handle inconsistencies in parity info.

Page Migration

Pages in same parity group can end up on same node:
(1) Bad server choice by client swapping page
(2) During page migration from one server to another
Nswap servers use LOW PRIORITY MIGRATE to restore reliability

Recovery when Swap-In Fails:

Client sends RECOVERY msg to Parity Server which restores page

Preliminary Results

<table>
<thead>
<tr>
<th>Workload</th>
<th>Swap to Disk</th>
<th>Nswap No Reliability</th>
<th>Nswap Mirroring</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sequential R&amp;W</td>
<td>361.5 sec</td>
<td>328.0 sec (speedup 1.1)</td>
<td>424.6 sec (0.9)</td>
</tr>
<tr>
<td>Random R&amp;W</td>
<td>1519.4</td>
<td>184.1 (8.3)</td>
<td>371.6 (4.1)</td>
</tr>
<tr>
<td>Random R&amp;W &amp; File I/O</td>
<td>2701.5</td>
<td>276.2 (9.8)</td>
<td>328.0 (8.2)</td>
</tr>
</tbody>
</table>

For more information about Nswap see: Newhall, Francis, Ganachev, and Speleg. "Nswap: A Network Swapping Module for Linux Clusters." In the Proceedings of Euro-Par'95.