Evaluating effects of memory compressed usage on MeeGo

LinuxCon Europe 2011
Anderson Briglia
anderson.briglia@indt.org.br
INdT Streams:

• SW & UI
• Network Technologies
• Manufacturing(logistics)
• Service experience

Independent and non-profit Research and Development Institute, founded by Nokia and focused on the generation of new concepts, products and solutions for areas related to mobile technologies and the internet.
Agenda

• Motivation
• Related work
• Compcache: some details about the project
• Experiments using N9 and N900 + MeeGo
• Final considerations
Why compressed memory?

• Generally, memory is scarce.

• Hardware changes bring more co$ts.

• A software solution can relieve the memory scarcity.

• Flash storage suffers from wear-leveling issues, so its useful if we can avoid using them as swap device.
• 1993 – Douglis with first comp. cache implementation.

• 1999 – Kaplan with new adaptive scheme.

• Following the same scheme, Rodrigo Castro released an implementation with new compressed storage structures approach and new adaptive re-sizing approach. It was for 2.4.x Linux kernel.

• CRAMES – Compressed RAM for Embedded Systems.
Compressed Cache

Previous version

• Changes to the swap write path, page fault handler and page cache lookup functions.

• Intrusive nature.
Compressed Cache

Ramzswap module

- RAM based block device used as swap disk.
- Swapped pages are compressed and stored in memory itself.
- Requested pages are decompressed before swapped-out.
Compressed Cache

Implementation design

- Far less intrusive than previous approach.
- Do not compress page cache (filesystem backed) pages.
- Compress anonymous pages only.
- Individual components:
  - LZO compressor.
  - xvMalloc memory allocator.
  - compcache block device driver: ramzswap.
- swapon /dev/ramzswap0
Compressed Cache

Memory management

• How manage the variable size of the compressed chunks?

• How reduce the fragmentation?
Compressed Cache

Memory management

• xvMalloc: $O(1)$ malloc/free.
• Very low fragmentation as presented on all tests.
• Can use highmem.
• Non-standard allocator interface.
Compressed Cache

Limitations

- ramzswap can never know when a compressed page is no longer required.
- Swap discard mechanism: BIO_RW_DISCARD
- If there is no swap operations the stale pages will remain.
- ramzswap can simply forward uncompresible pages to a backing swap disk, but it cannot swap out memory allocated by xvmalloc.
Test platforms

- N9 + Harmattan
  - has ramzswap as default swap device: ~ 256 MB
  - Swappiness = 30
  - RAM 1GB

- N900 running MeeGo by Community
  - has MMC block device as default swap device: ~ 75 MB
  - RAM 256 MB

- Tests consists in memory allocation speed (fillmem, scan utilities), and low memory situations: multiple applications running.
Compressed Cache Experiments

- Is ramzswap overhead affecting memory allocation speed?
- Is ramzswap enabling other use cases when memory consumption is critical?
N9 Experiments

• Performance tests
  – Memory allocation speed

• Memory consumption
  – Memory behavior when applications are running and ramzswap is being used or not
N9 Experiments

Performance test

• Time to allocate 550 MB

Swappiness = 60

Real swap + ramzswap

No ramzswap

Time (s)
N9 Experiments

Compressed memory after 550 MB allocated

The graph shows the number of stored pages and the saved memory for different configurations. The configurations include:
- No ramzswap
- Default config
- Swappiness = 60

The y-axis represents the number of stored pages and saved memory, ranging from 0 to 120000. The x-axis has two sections:
- Number of stored pages
- Saved memory

The graph visually compares the performance of each configuration in terms of memory compression and allocation.
N9 Experiments
Memory consumption behavior

• 8 browsers instances
• Calendar
• Video player
• Music player
N9 Experiments
Memory consumption behavior

MemFree

(default, no-ramzswap, swappiness 60)
## N9 Experiments

### Ramzswap memory consumption stats

<table>
<thead>
<tr>
<th>Metric</th>
<th>Default configuration</th>
<th>Swappiness = 60</th>
</tr>
</thead>
<tbody>
<tr>
<td>DiskSize:</td>
<td>262144 kB</td>
<td>262144 kB</td>
</tr>
<tr>
<td>NumReads:</td>
<td>648</td>
<td>897</td>
</tr>
<tr>
<td>NumWrites:</td>
<td>2183</td>
<td>10576</td>
</tr>
<tr>
<td>FailedReads:</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>FailedWrites:</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>InvalidIO:</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>NotifyFree:</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>ZeroPages:</td>
<td>68</td>
<td>419</td>
</tr>
<tr>
<td>GoodCompress:</td>
<td>87 %</td>
<td>76 %</td>
</tr>
<tr>
<td>NoCompress:</td>
<td>1 %</td>
<td>5 %</td>
</tr>
<tr>
<td>PagesStored:</td>
<td>2115</td>
<td>10157</td>
</tr>
<tr>
<td>PagesUsed:</td>
<td>502</td>
<td>3206</td>
</tr>
<tr>
<td>OrigDataSize:</td>
<td>8460 kB</td>
<td>40628 kB</td>
</tr>
<tr>
<td>ComprDataSize:</td>
<td>1979 kB</td>
<td>12682 kB</td>
</tr>
<tr>
<td>MemUsedTotal:</td>
<td>2008 kB</td>
<td>12824 kB</td>
</tr>
</tbody>
</table>
## Default configuration

<table>
<thead>
<tr>
<th>Metric</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>DiskSize</td>
<td>262144 kB</td>
</tr>
<tr>
<td>NumReads</td>
<td>648</td>
</tr>
<tr>
<td>NumWrites</td>
<td>2183</td>
</tr>
<tr>
<td>FailedReads</td>
<td>0</td>
</tr>
<tr>
<td>FailedWrites</td>
<td>0</td>
</tr>
<tr>
<td>InvalidIO</td>
<td>0</td>
</tr>
<tr>
<td>NotifyFree</td>
<td>0</td>
</tr>
<tr>
<td>ZeroPages</td>
<td>68</td>
</tr>
<tr>
<td>GoodCompress</td>
<td>87 %</td>
</tr>
<tr>
<td>NoCompress</td>
<td>1 %</td>
</tr>
<tr>
<td>PagesStored</td>
<td>2115</td>
</tr>
<tr>
<td>PagesUsed</td>
<td>502</td>
</tr>
<tr>
<td>OrigDataSize</td>
<td>8460 kB</td>
</tr>
<tr>
<td>ComprDataSize</td>
<td>1979 kB</td>
</tr>
<tr>
<td>MemUsedTotal</td>
<td>2008 kB</td>
</tr>
</tbody>
</table>

## Swappiness = 60

<table>
<thead>
<tr>
<th>Metric</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>DiskSize</td>
<td>262144 kB</td>
</tr>
<tr>
<td>NumReads</td>
<td>897</td>
</tr>
<tr>
<td>NumWrites</td>
<td>10576</td>
</tr>
<tr>
<td>FailedReads</td>
<td>0</td>
</tr>
<tr>
<td>FailedWrites</td>
<td>0</td>
</tr>
<tr>
<td>InvalidIO</td>
<td>0</td>
</tr>
<tr>
<td>NotifyFree</td>
<td>0</td>
</tr>
<tr>
<td>ZeroPages</td>
<td>419</td>
</tr>
<tr>
<td>GoodCompress</td>
<td>76 %</td>
</tr>
<tr>
<td>NoCompress</td>
<td>5 %</td>
</tr>
<tr>
<td>PagesStored</td>
<td>10157</td>
</tr>
<tr>
<td>PagesUsed</td>
<td>3206</td>
</tr>
<tr>
<td>OrigDataSize</td>
<td>40628 kB</td>
</tr>
<tr>
<td>ComprDataSize</td>
<td>12682 kB</td>
</tr>
<tr>
<td>MemUsedTotal</td>
<td>12824 kB</td>
</tr>
</tbody>
</table>
N9 Experiments

Some conclusions

- Memory consumption
  - Default configuration
    - Good responsiveness but the video did not play.
  - No swap
    - After the six browser instance the responsiveness was decreasing and the video player did not start.
  - With ramzswap
    - Responsiveness ok, all applications started and worked properly.
N900 Experiments

• Performance tests
  – Memory allocation speed

• Memory consumption
  – Memory behavior when applications are running and ramzswap is being used or not
N9 Experiments

Performance: Time to allocate 90 MB

- Swappiness = 60
- Real swap + ramzswap
- No ramzswap

Time (s)
N900 Experiments

- Time to allocate 90 MB
  - System was able to allocate 90 MB bunch of memory in three different swap configurations.
  - Swappiness = 60 and ramzswap size = 15% of total memory seems to be the best choice.
N900 Experiments
Memory consumption behavior

- 8 browsers instances
- Video player
- Photo gallery
N900 Experiments
Memory consumption behavior
N900 Experiments

Results

• Memory consumption
  – Default configuration
    • No responsiveness after video play, not possible to finish the test.
  – With ramzswap (swappiness = 30 or 60)
    • No possible to finish the test as well. The device rebooted.
  – Ramzswap + real swap (mmc blk device)
    • Best choice.
    • Ramzswap was configured to have higher priority. MMC blk device as second swap area.
As presented, memory compressing is an alternative for embedded and memory limited devices.

The newer Compressed Cache version (using ramzswap), is pretty mature and had abandoned the intrusive nature from previous version.
Final considerations

- N9 experiments showed that the best choice is to combine the fs back storage swap and the ramzswap. Its default configuration already set this.

- The default value for swappiness on N9 is 30. Maybe we should review this since we could have more pages going earlier to compressed memory, increasing the performance.
Final considerations

• N900 experiments showed that the ramzswap usage could bring more benefits than the current configuration.

• Before ramzswap load, we need to configure the amount of memory used since it is not a good idea to have big ramzswap partitions.
Presentation resources

- Compcache project website:  
  http://code.google.com/p/compcache

- LWN.net article about in-memory compressed swapping:  
  http://lwn.net/Articles/334649/

- Documentation about Virtual Memory Management on kernel.
Thanks

anderson.briglia@indt.org.br
anderson.briglia@openbossa.org