Dynamic Features of Linux on System z

Richard Young
IBM STG Lab Services

August 10th, 2011
Session Number 9878
The following are trademarks of the International Business Machines Corporation in the United States and/or other countries. For a complete list of IBM Trademarks, see www.ibm.com/legal/copytrade.shtml: AS/400, DB2, e-business logo, ESCON, eServer, FICON, IBM, IBM Logo, iSeries, MVS, OS/390, pSeries, RS/6000, S/390, System Storage, System z9, VM/ESA, VSE/ESA, WebSphere, xSeries, z/OS, zSeries, z/VM.

The following are trademarks or registered trademarks of other companies:

Java and all Java-related trademarks and logos are trademarks of Sun Microsystems, Inc., in the United States and other countries. LINUX is a registered trademark of Linux Torvalds in the United States and other countries. UNIX is a registered trademark of The Open Group in the United States and other countries. Microsoft, Windows and Windows NT are registered trademarks of Microsoft Corporation. SET and Secure Electronic Transaction are trademarks owned by SET Secure Electronic Transaction LLC. Intel is a registered trademark of Intel Corporation. * All other products may be trademarks or registered trademarks of their respective companies.

NOTES: Performance is in Internal Throughput Rate (ITR) ratio based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user’s job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput improvements equivalent to the performance ratios stated here.

IBM hardware products are manufactured from new parts, or new and serviceable used parts. Regardless, our warranty terms apply. All customer examples cited or described in this presentation are presented as illustrations of the manner in which some customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics will vary depending on individual customer configurations and conditions. This publication was produced in the United States. IBM may not offer the products, services or features discussed in this document in other countries, and the information may be subject to change without notice. Consult your local IBM business contact for information on the product or services available in your area.

All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only. Information about non-IBM products is obtained from the manufacturers of those products or their published announcements. IBM has not tested those products and cannot confirm the performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Prices subject to change without notice. Contact your IBM representative or Business Partner for the most current pricing in your geography. References in this document to IBM products or services do not imply that IBM intends to make them available in every country. Any proposed use of claims in this presentation outside of the United States must be reviewed by local IBM country counsel prior to such use. The information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice. Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.
Agenda

• Uses of Dynamic Resource Configuration
• Dynamically Adding Memory Resources to Linux
• Dynamically Adding Virtual CP Resources to Linux
• Automated Adjustment of CP and Memory Resources (CPU Hotplug)
• Linux on System z Suspend & Resume
Uses of dynamic resource configuration

• Helps to avoid Linux guest restarts and potential outage/downtime resource allocation changes
• Accommodate unplanned increases in application workload demands
• It can allow for more efficient overall Hypervisor operation (reduced overhead)
• Automated policy based reconfiguration more responsive than manual adjustments.
“Hotplug Memory”

- You can dynamically increase/decrease the memory for your running Linux guest system.
- To make memory available as hotplug memory you must define it to your LPAR or z/VM.
- Hotplug memory is supported by z/VM 5.4 with the PTF for APAR VM64524 and by later z/VM versions.
Dynamically Adding Memory

RGYLF0E4 DIRECT   AO   F 80   Trunc=72 Size=20 Line=0 Col=1 Alt=0

=====   *   **   Top of File   *   **   *
=====   USER RGYLF0E4 1GYLF0E4 1G 2G  G
=====   INCLUDE LINDFLT
=====   CPU 00
=====   CPU 01
=====   CRYPTO   APVIRTUAL
=====   IUCV ANY
=====   LOADDEV PORTNAME 5005076306138411
=====   LOADDEV LUN 4011402E000000000
=====   MACHINE ESA 4
=====   OPTION APPLMON MAXCONN 128
=====   DEDICATE 1000 3B46
=====   DEDICATE 2000 3B66
=====   DEDICATE 4000 1FF6
=====   NICDEF 0700 TYPE QDIO DEV 3 LAN SYSTEM NET172A
Dynamically Adding Memory

- This z/VM guest has a user directory entry with 1GB of initial memory and 2 GB of maximum memory
- In z/VM, changing the memory size or configuration of a guest causes a storage reset
- If you are running Linux natively in an LPAR without z/VM, you would use reserved storage in the LPAR definition to set aside potential additional memory
- In z/VM, define the memory to be dynamically enabled as “standby” storage
Dynamically Adding Memory

21:15:04 Ready; T=0.01/0.02 21:15:04
21:15:14 define storage 1G standby 1G
21:15:14 00: STORAGE = 1G MAX = 2G INC = 2M STANDBY = 1G RESERVED = 0
21:15:14 00: Storage cleared - system reset.
Dynamically Adding Memory

- “DEFINE STORAGE 1G STANDBY 1G” issued for this guest
- Issuing a DEFINE STORAGE command causes storage to be cleared
- Anything running at the time of the reset will be immediately terminated without running any shutdown procedures
- This means if you issued this command from a CMS EXEC, CMS is no longer running because storage has been cleared.
Dynamically Adding Memory

- Example of **IPL** and **define storage** commands in PROFILE EXEC:

```
IPLLNX:
    CALL DIAG 8,'DEFINE STORAGE 1G STANDBY 1G ' '15'X,
    'IPL 200 ' '15'X
    'CP MSG * IPL 200'
return
```
Dynamically Adding Memory

```bash
rgy1x0e4:~ # cat /proc/meminfo
MemTotal:       1021320 kB
MemFree:        17708 kB
Buffers:        192412 kB
Cached:         656340 kB
SwapCached:     0 kB
Active:         214908 kB
Inactive:       659924 kB
Active(anon):   2940 kB
Inactive(anon): 23256 kB
Active(file):   211968 kB
Inactive(file): 636668 kB
Unevictable:    0 kB
Mlocked:        0 kB
SwapTotal:      0 kB
SwapFree:       0 kB
Dirty:          16 kB
```
Dynamically Adding Memory

• After IPLing Linux in this guest, observe via /proc/meminfo that approximately 1GB of memory is available
• The “standby” memory is not reported by /proc/meminfo
• The /sys file system however has an awareness of this “standby” or “hot plug” memory
• With current level of s390-tools, Ismem can be used to report this information and chmem to bring elements online or offline
Dynamically Adding Memory

rgylx0e4:~ # cd /sys/devices/system/memory/
rgylx0e4:/sys/devices/system/memory # ls
block_size_bytes memory0 memory1 memory2 memory3
rgylx0e4:/sys/devices/system/memory # ls -la
total 0
drwxr-xr-x 6 root root 0 Apr 1 11:05 .
drwxr-xr-x 8 root root 0 Mar 28 01:03 ..
-r--r--r-- 1 root root 4096 Apr 1 11:05 block_size_bytes
drwxr-xr-x 2 root root 0 Apr 1 11:05 memory0
drwxr-xr-x 2 root root 0 Apr 1 11:05 memory1
drwxr-xr-x 2 root root 0 Apr 1 11:05 memory2
drwxr-xr-x 2 root root 0 Apr 1 11:05 memory3
rgylx0e4:/sys/devices/system/memory # cat block_size_bytes
10000000
rgylx0e4:/sys/devices/system/memory # ls memory0/
end_phys_index phys_device phys_index removable_state
rgylx0e4:/sys/devices/system/memory # grep -r --include "state" "line" /sys/devices/system/memory
/sys/devices/system/memory/memory0/state:online
/sys/devices/system/memory/memory1/state:online
/sys/devices/system/memory/memory2/state:online
/sys/devices/system/memory/memory3/state:online
rgylx0e4:/sys/devices/system/memory #
Dynamically Adding Memory

- When no standby memory is defined, only the 4 "core" memory sections exist
- No hotplug memory sections currently exist
- The next slide will show an example of /sys/devices/system/memory with **hotplug** memory sections available.
Dynamiically Adding Memory

```
rgylx0e4:/sys/devices/system/memory # ls -la
total 0
drwxr-xr-x 10 root root 0 Apr 1 13:05 .
drwxr-xr-x  8 root root 0 Apr 1 13:04 ..
-r-xr-xr--  1 root root 4096 Apr 1 13:05 block_size_bytes
drwxr-xr-x  2 root root 0 Apr 1 13:05 memory0
drwxr-xr-x  2 root root 0 Apr 1 13:05 memory1
drwxr-xr-x  2 root root 0 Apr 1 13:05 memory2
drwxr-xr-x  2 root root 0 Apr 1 13:05 memory3
drwxr-xr-x  2 root root 0 Apr 1 13:05 memory4
drwxr-xr-x  2 root root 0 Apr 1 13:05 memory5
drwxr-xr-x  2 root root 0 Apr 1 13:05 memory6
drwxr-xr-x  2 root root 0 Apr 1 13:05 memory7
```

Core Memory Sections

```
rgylx0e4:/sys/devices/system/memory # cat block_size_bytes
100000000
```

Hotplug Memory Sections

```
rgylx0e4:/sys/devices/system/memory # ls memory0/
end_phys_index phys_device phys_index removable state
rgylx0e4:/sys/devices/system/memory #
```
Dynamically Adding Memory

- /sys/devices/system/memory shows the eight “sections”.
- Linux allocates the initially allocated memory as “Core” memory. This is divided into 4 sections.
- The additional memory that can be added is “Hotplug” memory. This is also divided into 4 sections.
- The state of each memory section can be queried or set.
- The size of each section is documented in the “block_size_bytes” file.
Dynamically Adding Memory

```
rgylx0e4:~ # lsmem
Address Range
0x0000000000000000-0x0000000000000000 256 online no 0-127
0x000000000010000000-0x0000000000000002 512 online yes 128-383
0x000000000030000000-0x0000000000000003 256 online no 384-511
0x000000000040000000-0x0000000000000007 1024 offline - 512-1023

Memory device size : 2 MB
Memory block size : 256 MB
Total online memory : 1024 MB
Total offline memory: 1024 MB
```
Dynamically Adding Memory

- Recent versions of s390-tools include the `lsmem` command
- `lsmem` provides a quick easily readable view of the same information that is in /sys/devices/system/memory directory. It details:
  - Which memory ranges are online or offline
  - Which memory is removable
  - The size of each range
  - The total memory online & offline
  - The memory section block size
Dynamically Adding Memory

rgylx0e4:/sys/devices/system/memory # grep -r --include="state" "line" /sys/devices/system/memory/
/sys/devices/system/memory/memory0/state:online
/sys/devices/system/memory/memory1/state:online
/sys/devices/system/memory/memory2/state:online
/sys/devices/system/memory/memory3/state:online
/sys/devices/system/memory/memory4/state:offline
/sys/devices/system/memory/memory5/state:offline
/sys/devices/system/memory/memory6/state:offline
/sys/devices/system/memory/memory7/state:offline
rgylx0e4:/sys/devices/system/memory # echo online > /sys/devices/system/memory/memory4/state
rgylx0e4:/sys/devices/system/memory # grep -r --include="state" "line" /sys/devices/system/memory/
/sys/devices/system/memory/memory0/state:online
/sys/devices/system/memory/memory1/state:online
/sys/devices/system/memory/memory2/state:online
/sys/devices/system/memory/memory3/state:online
/sys/devices/system/memory/memory4/state:online
/sys/devices/system/memory/memory5/state:offline
/sys/devices/system/memory/memory6/state:offline
/sys/devices/system/memory/memory7/state:offline
rgylx0e4:/sys/devices/system/memory # lsmem
<table>
<thead>
<tr>
<th>Address</th>
<th>Range</th>
<th>Size (MB)</th>
<th>State</th>
<th>Removable</th>
<th>Device</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x0000000000000000-0x0000000000000000</td>
<td>256</td>
<td>online</td>
<td>no</td>
<td>0-127</td>
<td></td>
</tr>
<tr>
<td>0x0000000001000000-0x0000000002000000</td>
<td>512</td>
<td>online</td>
<td>yes</td>
<td>128-383</td>
<td></td>
</tr>
<tr>
<td>0x0000000003000000-0x0000000003000000</td>
<td>256</td>
<td>online</td>
<td>no</td>
<td>384-511</td>
<td></td>
</tr>
<tr>
<td>0x0000000004000000-0x0000000004000000</td>
<td>256</td>
<td>online</td>
<td>yes</td>
<td>512-639</td>
<td></td>
</tr>
<tr>
<td>0x0000000005000000-0x0000000007000000</td>
<td>768</td>
<td>offline</td>
<td>-</td>
<td>640-1023</td>
<td></td>
</tr>
</tbody>
</table>

Memory device size : 2 MB
Memory block size  : 256 MB
Total online memory: 1280 MB
Total offline memory: 768 MB
rgylx0e4:/sys/devices/system/memory #
Dynamically Adding Memory

- One of the four hotplug memory sections is enabled by echoing “online” in to the state file.
- Ismem shows 256 MB of hotplug memory enabled and 1280MB now online
Dynamically Adding Memory

```
rallyx0e4:/sys/devices/system/memory # cat /proc/meminfo
MemTotal: 1283464 kB
MemFree: 1132460 kB
Buffers: 7296 kB
Cached: 51020 kB
SwapCached: 0 kB
Active: 30820 kB
Inactive: 52996 kB
Active(anon): 25508 kB
Inactive(anon): 120 kB
Active(file): 5312 kB
Inactive(file): 52876 kB
Unevictable: 0 kB
Mlocked: 0 kB
SwapTotal: 0 kB
SwapFree: 0 kB
Dirty: 4 kB
Writeback: 0 kB
AnonPages: 25504 kB
Mapped: 11032 kB
Shmem: 128 kB
```
Dynamically Adding Memory

- After enabling one memory section /proc/meminfo shows an additional 250MB of memory
- This is $\frac{1}{4}$th of our standby memory we defined with the DEFINE STORAGE command earlier
- Since we have 4 storage “sections” to represent the standby memory this amount is correct
Dynamically Adding Memory

```bash
dynamically adding memory

rgylx0e4:/sys/devices/system/memory # echo online > /sys/devices/system/memory/memory5/state
rgylx0e4:/sys/devices/system/memory # echo online > /sys/devices/system/memory/memory6/state
rgylx0e4:/sys/devices/system/memory # echo online > /sys/devices/system/memory/memory7/state
rgylx0e4:/sys/devices/system/memory # grep -r --include="state" "line" /sys/devices/system/memory/
```

<table>
<thead>
<tr>
<th>Address Range</th>
<th>Size (MB)</th>
<th>State</th>
<th>Removable</th>
<th>Device</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x0000000000000000-0x0000000000ffffff</td>
<td>256</td>
<td>online</td>
<td>no</td>
<td>0-127</td>
</tr>
<tr>
<td>0x0000000000000001-0x0000000000000002ffffff</td>
<td>512</td>
<td>online</td>
<td>yes</td>
<td>128-383</td>
</tr>
<tr>
<td>0x0000000000000003-0x0000000000000003ffffff</td>
<td>256</td>
<td>online</td>
<td>no</td>
<td>384-511</td>
</tr>
<tr>
<td>0x0000000000000004-0x0000000000000007ffffff</td>
<td>1024</td>
<td>online</td>
<td>yes</td>
<td>512-1023</td>
</tr>
</tbody>
</table>

Memory device size : 2 MB
Memory block size  : 256 MB
Total online memory: 2048 MB
Total offline memory: 0 MB

rgylx0e4:/sys/devices/system/memory #
```
Dynamically Adding Memory

- echo online is issued for the remaining 3 storage elements
- After enabling all the hotplug memory sections we should see a full 2GB of memory reported
- The full 2GB of memory is now reported by /proc/meminfo
**Dynamically Remove/Add Memory**

```bash
grylx0e4:/sys/devices/system/memory # chmem --disable 1024
grylx0e4:/sys/devices/system/memory # lsmem

<table>
<thead>
<tr>
<th>Address Range</th>
<th>Size (MB)</th>
<th>State</th>
<th>Removable</th>
<th>Device</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x0000000000000000-0x000000000fffffff</td>
<td>256</td>
<td>online</td>
<td>no</td>
<td>0-127</td>
</tr>
<tr>
<td>0x0000000001000000-0x000000002fffffff</td>
<td>512</td>
<td>online</td>
<td>yes</td>
<td>128-383</td>
</tr>
<tr>
<td>0x0000000003000000-0x000000003fffffff</td>
<td>256</td>
<td>online</td>
<td>no</td>
<td>384-511</td>
</tr>
<tr>
<td>0x0000000004000000-0x000000007fffffff</td>
<td>1024</td>
<td>offline</td>
<td>-</td>
<td>512-1023</td>
</tr>
</tbody>
</table>
```

- Memory device size : 2 MB
- Memory block size : 256 MB
- Total online memory : 1024 MB
- Total offline memory: 1024 MB

```bash
grylx0e4:/sys/devices/system/memory # grep -r --include="state" "line" /sys/devices/system/memory/
```

```bash
grylx0e4:/sys/devices/system/memory # chmem --enable 1024
grylx0e4:/sys/devices/system/memory # lsmem

<table>
<thead>
<tr>
<th>Address Range</th>
<th>Size (MB)</th>
<th>State</th>
<th>Removable</th>
<th>Device</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x0000000000000000-0x000000000fffffff</td>
<td>256</td>
<td>online</td>
<td>no</td>
<td>0-127</td>
</tr>
<tr>
<td>0x0000000001000000-0x000000002fffffff</td>
<td>512</td>
<td>online</td>
<td>yes</td>
<td>128-383</td>
</tr>
<tr>
<td>0x0000000003000000-0x000000003fffffff</td>
<td>256</td>
<td>online</td>
<td>no</td>
<td>384-511</td>
</tr>
<tr>
<td>0x0000000004000000-0x000000007fffffff</td>
<td>1024</td>
<td>online</td>
<td>yes</td>
<td>512-1023</td>
</tr>
</tbody>
</table>
```

- Memory device size : 2 MB
- Memory block size : 256 MB
- Total online memory : 2048 MB
- Total offline memory: 0 MB
Dynamically Adding Memory

- The memory sections can be set online or offline via the chmem command instead of echoing in to the “state” file
- lsmem will reported the memory sections in an accumulated fashion when the attributes are the same
- Not all memory sections will be removable, and the removable state can change over time
Summary of Memory Hotplug

• Utilizing hotplug memory does require some advanced planning:
  • z/VM 5.4 with VM64524 or above
  • DEFINE STORAGE STANDBY issued before Linux is IPLed
  • For native LPAR, RESERVED STORAGE must be defined
  • SLES 11 / RHEL 6

• Suspend/Resume restriction: The Linux instance must not have used any hotplug memory since it was last booted.
• You may not be able to disable hotplug memory that has been enabled
Summary of Memory Hotplug

- Can be very helpful when exact future memory need is unknown, without over allocating online memory from the start.
- After a Linux reboot core memory is made available again and hotplug memory is freed
Dynamically Managing Virtual CPs from Linux
The directory entry shows a guest with two initial virtual CPs.
The maximum potential virtual CPs shown is four.
z/VM does not make the additional potential virtual CPs available for Linux to enable on its own.
The additional potential virtual CPs must first be defined in the z/VM guest before dynamically enabling on Linux.
Dynamically Managing Virtual CPs

rgy1x0e4:~ # vmcp q v
STORAGE = 1G
XSTORE = none

CPU 00 ID FF12EBBE20978000 (BASE) CP CPUAFF ON
CPU 01 ID FF12EBBE20978000 CP CPUAFF ON

AP 51 CEX2A Queue 08 shared
CONS 0009 DISCONNECTED TERM START
  0009 CL T NOCONT NOHOLD COPY 001 READY FORM STANDARD
  0009 TO RGY1X0E4 RDR DIST RGY1X0E4 FLASHC 000 DEST OFF
  0009 FLASH CHAR MDFY 0 FCB LPP OFF
  0009 3215 NOEOF OPEN 0013 NOKEEP NOMSG NONAME
  0009 SUBCHANNEL = 000A

• Here the current z/VM guests virtual resources are displayed from within Linux
• The two initial and active virtual CPs are shown
• Notice there is no information displayed about the potential additional virtual CPs
**Dynamically Managing Virtual CPs**

```
rgylx0e4:~ # mpstat -A
Linux 2.6.32.29-0.3-default (rgylx0e4) 04/01/11 _s390x_

13:19:24   CPU   %usr  %nice  %sys  %iowait  %irq  %soft  %steal  %guest  %idle
13:19:24   all  1.43  0.00  0.65  0.30   0.00  0.02  0.06   0.00  97.53
13:19:24   0    1.62  0.00  0.67  0.29   0.00  0.02  0.03   0.00  97.37
13:19:24   1    1.25  0.00  0.64  0.30   0.00  0.02  0.08   0.00  97.70

13:19:24   CPU   intr/s
13:19:24   all  0.00
13:19:24   0    0.00
13:19:24   1    0.00
```

- Note the mpstat output from before defining the additional virtual CPs
- Observe the even distribution of idle time and usage
The Linux sysfs file system can access information about the two active virtual CPs.

- The kernel has a maximum potential of 64 processors.
- No information about the two potential additional virtual CPs is shown yet.
Dynamically Managing Virtual CPs

Using the `vmcp` command we pass the zVM CP DEFINE CPU commands on to our z/VM guest.

Remember this is a class G guest enabling the additional resources previously called out in the user directory.

After defining the additional virtual CPs in z/VM we still do not see them in the Linux /sysfs.
Dynamically Managing Virtual CPs

rgylx0e4:/sys/devices/system/cpu # ls
cpu0 cpu1 dispatching kernel_max offline online perf_events possible present rescann
rgylx0e4:/sys/devices/system/cpu # vmcp q v
STORAGE = 1G
XSTORE = none

CPU 00 ID FF12EBBE209780000 (BASE) CP CPUAFF ON
CPU 01 ID FF12EBBE209780000 CP CPUAFF ON
CPU 03 ID FF12EBBE209780000 STOPPED CP CPUAFF ON
CPU 02 ID FF12EBBE209780000 STOPPED CP CPUAFF ON

AP 51 CEX2A Queue 08 shared
CONS 0009 DISCONNECTED TERM START
  0009 CL T NOCONT NOHOLD COPY 001 READY FORM STANDARD
  0009 TO RGYLX0E4 RDR DIST RGYLX0E4 FLASHC 000 DEST OFF
  0009 FLASH CHAR MDY 0 FCB LPP OFF
  0009 3215 NOEOF OPEN 0013 NOKEEP NOMSG NONAME
  0009 SUBCHANNEL = 000A
RDR 000C CL * NOCONT NOHOLD EOF READY
  000C 2540 CLOSED NOKEEP NORESCAN SUBCHANNEL = 000E
FUN 000D CL A NOCONT NOHOLD COPY 001 READY FORM STANDARD
  000D TO RGYLX0E4 FUN DIST RGYLX0E4 DEST OFF
  000D FLASH 000 CHAR MDY 0 FCB
  000D 2540 NOEOF CLOSED NOKEEP NOMSG NONAME
  000D SUBCHANNEL = 000F
PRT 000E CL A NOCONT NOHOLD COPY 001 READY FORM STANDARD
  000E TO RGYLX0E4 PRT DIST RGYLX0E4 FLASHC 000 DEST OFF
  000E FLASH CHAR MDY 0 FCB LPP OFF
  000E 1403 NOEOF CLOSED NOKEEP NOMSG NONAME
  000E SUBCHANNEL = 0010
Dynamically Managing Virtual CPs

• By using the z/VM QUERY VIRTUAL command we can see the additional virtual CPs have been defined to the guest
• The new virtual CPs are in a “stopped” state
Dynamically Managing Virtual CPs

mpstat is only reporting two CPUs

The rescan operation is used to search for new available CPUs in the guest.

After rescan, additional /sysfs entries exist
Dynamically Managing Virtual CPs

```
rgylx0e4:/sys/devices/system/cpu # mpstat -A
Linux 2.6.32.29-0.3-default (rgylx0e4) 04/01/11 _s390x_

13:24:41   CPU  %usr  %nice  %sys  %iowait  %irq  %soft  %steal  %guest  %idle
13:24:41   all  0.43  0.00  0.21  0.09  0.00  0.01  0.02  0.00  99.23
13:24:41     0  0.49  0.00  0.22  0.09  0.00  0.01  0.01  0.00  99.18
13:24:41     1  0.37  0.00  0.21  0.09  0.00  0.01  0.02  0.00  99.29
13:24:41     2  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00
13:24:41     3  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00
```

- mpstat reports 0% use and 0% idle for the new CPUs. This is because they are stopped and offline
- The new CPUs must still be brought online to Linux
Dynamically Managing Virtual CPs

- Bring the new CPUs online to Linux by echoing 1 in to the “online” file for the given CPU

```
rgylx0e4:/sys/devices/system/cpu/cpu2 # echo 1 > online
rgylx0e4:/sys/devices/system/cpu/cpu2 # ls
address capability configure crash_notes idle_count idle_time_us online polarization topology
rgylx0e4:/sys/devices/system/cpu/cpu2 # cat online
1
rgylx0e4:/sys/devices/system/cpu/cpu2 # echo 1 > ..../cpu3/online
```
Dynamically Managing Virtual CPs

• On a idle system, the new CPUs momentarily show 100% idle after being brought online
• Once a little bit of workload hits the system, this quickly changes
Dynamically Managing Virtual CPs

rgylx0e4:/sys/devices/system/cpu # ls
cpu0 cpu1 cpu2 cpu3 dispatching kernel_max offline online perf_events possible
rgylx0e4:/sys/devices/system/cpu # echo 0 > cpu1/online
rgylx0e4:/sys/devices/system/cpu # echo 0 > cpu3/online
rgylx0e4:/sys/devices/system/cpu # mpstat -A
Linux 2.6.32.29-0.3-default (rgylx0e4) 04/01/11 _s390x_

13:27:53   CPU %usr %nice %sys %iowait %irq %soft %steal %guest %idle
13:27:53   all  0.27  0.00  0.14  0.06  0.00  0.01  0.01  0.00  99.52
13:27:53     0  0.35  0.00  0.16  0.06  0.00  0.01  0.01  0.00  99.40
13:27:53     1  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  100.00
13:27:53     2  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  100.00
13:27:53     3  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  100.00
Dynamically Managing Virtual CPs

• You can take offline CPUs that were initially online as well

• Some Considerations
  • Obviously multithreaded application or multiple applications in a single virtual server could potentially benefit from additional virtual CPs
  • Could impact monitor applications or middleware that might query the number of processors on startup (i.e., the Java Virtual Machine)
  • zVM “DEFINE CPU” is a Class G command
  • This does NOT add additional capacity to the LPAR, it simply makes resources available to the guest
  • (R.O.T.) Don’t add unnecessary virtual CPs or more virtual CPs than logical processors.
Automated Policy Based Adjustment of CPs and Memory
(The CPU Hotplug Daemon)
Automated Adjustment of CPs and Memory

• The hot plug daemon (cpuplugd) can dynamically offline and re-online processors in Linux
• The hot plug daemon can also add and remove memory over time via CMM
• The cpuplug daemon checks the system at configurable intervals
• You must configure the plug and unplug rules for it to operate
• You must activate the cpuplug daemon to use it, by default it is inactive
Automated Adjustment of CPs and Memory

- The default rules are NOT recommendations
- You should customize the rules/configuration to fit your environment
- `cpuplugd -V -f -c /etc/sysconfig/cpuplugd` - This invokes cpuplugd in the foreground with verbose messaging to help you understand its operation
- It is highly recommended you customize its operation before enabling the cpuplug daemon
- It is important to understand what state you will be in after you execute a “plug” or “unplug” operation when writing the rules.
Automated Adjustment of CPs

More virtual CPs

Excessive available CP capacity

Desired CP capacity

Inadequate available CP capacity

Less virtual CPs
Automated Adjustment of CPs

Excessive available CP capacity

Desired CP capacity

Inadequate available CP capacity

Desired Action –

- Remove enough capacity so you are in the “green zone” after the plug rule triggers
- If resource demand is unchanged, subsequent intervals should not undo your action
Automated Adjustment of CPs

- Excessive available CP capacity
- Desired CP capacity
- Inadequate available CP capacity

Very likely NOT your optimal configuration

Step 1
Step 2
Step 3
Step 4
Step 5
Step 6
Step N
Step N+1
Automated Adjustment of CPs

- You can only add/remove a full virtual CP of capacity.
- This means at times you might have 1.25 or more virtual CPs of idle capacity as an acceptable state.
- Understand the range in which your rules are plugging and unplugging virtual CPs. It should be at least the size of one virtual CP, since that is the minimum granularity you can add or remove.
What happens if I run with the default rules?

- CPU_MIN = 1
- CPU_MAX = 0 (maximum available)
- UPDATE = 10
- HOTPLUG = "$\text{loadavg} > \text{onumcpus} + 0.75 \& \text{(idle} < 10.0\text{)}$"
- HOTUNPLUG = "$\text{loadavg} < \text{onumcpus} - 0.25 | \text{(idle} > 50\text{)}$"
- Defined As:
  - loadavg: The current loadaverage
  - onumcpus: The actual number of cpus which are online
  - runnable_proc: The current amount of runnable processes
  - idle: The current idle percentage
What happens if I run with the default rules?

- **loadavg**: the current load average – Comes from the first /proc/loadavg value. The average number of runnable process. Not average CPU utilization! One looping process on a system would cause this to approach 1.0 Five looping processes on a single CPU system would cause this to approach 5.0

- **onumcpus**: the actual number of cpus which are online 
  (Via: /sys/devices/system/cpu/cpu/cpu%d/online)

- **runable_proc**: the current amount of runnable processes 
  (The 4th /proc/loadavg value)

- **idle**: the current idle percentage – Where 1 idle processor = 100 and 4 idle processors = 400 (/proc/stat 4th value)
Specific cpuplugd examples for CPU
Automated Adjustment of CPs
Automated Adjustment of CPs

- The initial state of the system is:
  - 4 virtual CPs
  - System is currently completely idle and has more processor capacity than it currently needs
Automated Adjustment of CPs

^Crgylx0e4:~ # cpuplugd -V -f -c /etc/sysconfig/cpuplugd
found cpu_min value: 1
found cpu_max value: 0
found update value: 10
found cmm_min value: 0
found cmm_max value: 8192
found cmm_inc value: 256
found the following rule: HOTPLUGIN = (loadavg+0.75>onumcpus) | (idle<25.0)
found the following rule: HOTUNPLUGIN = (loadavg<onumcpus-0.25) | (idle>50)
found the following rule: MEMPLUGIN = freemem<250
found the following rule: MEMUNPLUGIN = freemem>750 | swaprate>1
Detected System running in z/VM mode
Valid CPU hotplug configuration d=ct=d.
Can not open /proc/sys/vm/cmm_pages
The memory hotplug function will be disabled.

update_interval: 10 s
cpu_min: 1
cpu_max: 4
loadavg: 2.470000
idle percent = 0.100000
onumcpus 4
runable_proc: 1

onumcpus: 4

hotplug: (((loadavg) + (0.750000)) > (onumcpus)) | ((idle) < (25.000000))
hotunplug: ((loadavg) < (((onumcpus) - (0.250000))) | ((idle) > (50.000000))

maximum cpu limit is reached
Automated Adjustment of CPs

• The cpu hotplug daemon is started in the foreground with cpuplugd –V –f –c /etc/sysconfig/cpuplugd

• Active rules echoed
  • HOTPLUG = (loadavg+0.75>onumcpus)|(idle<25.0)
  • HOTUNPLUG=(loadavg<onumcpus-.25)|(idle>50)

• Memory hotplug currently disabled, no /proc/sys/vm/cmm_pages. This will be covered later

• First interval
  • loadavg = 2.47
  • Idle percent = 0.1
  • Max CPU limit reached (all 4 are active)
Automated Adjustment of CPs

update_interval: 10 s
cpu_min: 1
cpu_max: 4
loadavg: 2.090000
idle_percent = 399.800000
numcpus 4
runable_proc: 1

onumcpus: 4

hotplug: (((loadavg) + (0.750000)) > (onumcpus)) | ((idle) < (25.000000))
hotunplug: ((loadavg) < ((onumcpus) - (0.250000))) | ((idle) > (50.000000))

-----------------------------------------
cpu with id 3 is currently online and will be disabled
-----------------------------------------

update_interval: 10 s
cpu_min: 1
cpu_max: 4
loadavg: 1.770000
idle_percent = 306.200000
numcpus 4
runable_proc: 1

onumcpus: 3

hotplug: (((loadavg) + (0.750000)) > (onumcpus)) | ((idle) < (25.000000))
hotunplug: ((loadavg) < ((onumcpus) - (0.250000))) | ((idle) > (50.000000))

-----------------------------------------
cpu with id 2 is currently online and will be disabled
-----------------------------------------
Automated Adjustment of CPs

• 2\textsuperscript{nd} Interval
  • Loadavg = 2
  • Idle = 399 (out of 4 online CPUs)
  • Action: CPU ID 3 disabled

• 3\textsuperscript{rd} Interval
  • Loadavg = 1.77
  • Idle = 306 (out of 3 online CPUs)
  • Action: CPU ID 2 disabled
Automated Adjustment of CPs

update_interval: 10 s
cpu_min: 1
cpu_max: 4
loadavg: 1.500000
idle_percent = 203.800000
numcpus: 4
runable_proc: 1
----------------------------------------
onumcpus: 2
----------------------------------------
hotplug: (((loadavg) + (0.750000)) > (onumcpus)) | ((idle) < (25.000000))
hotunplug: (((loadavg) < (((onumcpus) - (0.250000))) | ((idle) > (50.000000)))
----------------------------------------
cpu with id 2 is currently offline and will be enabled
cpu with id 2 enabled
----------------------------------------
update_interval: 10 s
cpu_min: 1
cpu_max: 4
loadavg: 1.270000
idle_percent = 303.500000
numcpus: 4
runable_proc: 1
----------------------------------------
onumcpus: 3
----------------------------------------
hotplug: (((loadavg) + (0.750000)) > (onumcpus)) | ((idle) < (25.000000))
hotunplug: (((loadavg) < (((onumcpus) - (0.250000))) | ((idle) > (50.000000)))
----------------------------------------
cpu with id 2 is currently online and will be disabled
Automated Adjustment of CPs

- Interval 4
  - Loadavg = 1.5
  - Idle % = 203
  - Action = Enable CPU ID 2 (because of loadavg part of rule, not idle%)

- Interval 5
  - Loadavg = 1.27
  - Idle % = 303
  - Action = Disable CPU ID 2 (because of both parts of the unplug rule)

- Load has stayed the same thru all of the intervals, yet we are adding and removing the same CPU
Automated Adjustment of CPs

- Messages about processors being enabled or disabled by CPU hotplug will appear in /var/log/messages.
- In this example 3 of 4 virtual CPs were stopped
- This information could easily be captured for reporting or alerting
Automated Adjustment of CPs

- Two processes running in a CPU loop on a 4 way system
- Lets take a look at the impact to CPU Hotplug

```
rgy1x0e4:~ # ps -ef | grep loop
root  3336  3200  71 13:49 pts/3 00:01:18 /bin/sh ./loopme.sh
root  3337  3200  74 13:49 pts/3 00:01:21 /bin/sh ./loopme.sh
root  3371  3200  0 13:51 pts/3 00:00:00 grep loop
```
Automated Adjustment of CPs

Apr 1 13:53:54 rgylx0e4 kernel: cpu.17772b: Processor 2 started, address 0, identification 12EBB5
Apr 1 13:54:04 rgylx0e4 kernel: cpu.f76a91: Processor 2 stopped
Apr 1 13:54:15 rgylx0e4 kernel: cpu.17772b: Processor 2 started, address 0, identification 12EBB5
Apr 1 13:54:25 rgylx0e4 kernel: cpu.f76a91: Processor 2 stopped
Apr 1 13:54:36 rgylx0e4 kernel: cpu.17772b: Processor 2 started, address 0, identification 12EBB5
Apr 1 13:54:46 rgylx0e4 kernel: cpu.f76a91: Processor 2 stopped
Apr 1 13:54:56 rgylx0e4 kernel: cpu.17772b: Processor 2 started, address 1, identification 12EBB5
Apr 1 13:55:06 rgylx0e4 kernel: cpu.17772b: Processor 3 started, address 0, identification 12EBB5
Apr 1 13:55:17 rgylx0e4 kernel: cpu.f76a91: Processor 3 stopped
Apr 1 13:55:27 rgylx0e4 kernel: cpu.17772b: Processor 3 started, address 0, identification 12EBB5
Apr 1 13:55:37 rgylx0e4 kernel: cpu.f76a91: Processor 3 stopped
Apr 1 13:55:47 rgylx0e4 kernel: cpu.f76a91: Processor 2 stopped
Apr 1 13:55:58 rgylx0e4 kernel: cpu.17772b: Processor 2 started, address 0, identification 12EBB5
Apr 1 13:56:08 rgylx0e4 kernel: cpu.f76a91: Processor 2 stopped
Apr 1 13:56:18 rgylx0e4 kernel: cpu.17772b: Processor 2 started, address 0, identification 12EBB5
Apr 1 13:56:28 rgylx0e4 kernel: cpu.f76a91: Processor 2 stopped
Apr 1 13:56:39 rgylx0e4 kernel: cpu.17772b: Processor 2 started, address 0, identification 12EBB5
Apr 1 13:56:49 rgylx0e4 kernel: cpu.f76a91: Processor 2 stopped
Apr 1 13:56:59 rgylx0e4 kernel: cpu.17772b: Processor 2 started, address 0, identification 12EBB5
Apr 1 13:57:10 rgylx0e4 kernel: cpu.f76a91: Processor 2 stopped
Apr 1 13:57:20 rgylx0e4 kernel: cpu.17772b: Processor 2 started, address 0, identification 12EBB5
Apr 1 13:57:30 rgylx0e4 kernel: cpu.f76a91: Processor 2 stopped
Apr 1 13:57:41 rgylx0e4 kernel: cpu.17772b: Processor 2 started, address 0, identification 12EBB5
Apr 1 13:57:51 rgylx0e4 kernel: cpu.f76a91: Processor 2 stopped
Apr 1 13:58:01 rgylx0e4 kernel: cpu.17772b: Processor 2 started, address 0, identification 12EBB5
Apr 1 13:58:11 rgylx0e4 kernel: cpu.f76a91: Processor 2 stopped
Apr 1 13:58:22 rgylx0e4 kernel: cpu.17772b: Processor 2 started, address 1, identification 12EBB5
Apr 1 13:58:32 rgylx0e4 kernel: cpu.f76a91: Processor 2 stopped
Apr 1 13:58:43 rgylx0e4 kernel: cpu.17772b: Processor 2 started, address 1, identification 12EBB5
Apr 1 13:58:53 rgylx0e4 kernel: cpu.f76a91: Processor 2 stopped
Apr 1 13:59:03 rgylx0e4 kernel: cpu.17772b: Processor 2 started, address 0, identification 12EBB5
Apr 1 13:59:14 rgylx0e4 kernel: cpu.f76a91: Processor 2 stopped
Automated Adjustment of CPs

• **Summary of our little experiment**
  - Under a steady load to 2 CPU bound processes, CPs zero and one stay online.
  - CP two oscillates between online and offline
  - CP three stays offline
  - Suggests the plug/unplug rules should be refined, since you are unable to add a virtual CP without removing it on the next interval.
Automated Adjustment of CPs

• Given:

\[
\text{HOTPLUG } (\text{loadavg}+0.75>\text{onumcpus}) \text{ | } (\text{idle}<25.0)
\]

\[
\text{HOTUNPLUG}= (\text{loadavg}<\text{onumcpus}-.25) \text{ | } (\text{idle}>50)
\]

• The idle part of the rules requires the system be between 25 and 50% idle not to take action. However adding or removing any CP will change this by a value of 100. This is not likely what you want.

• Unplugging a CPU when it is 51% idle could impact your application. What handles the 49% of the CP that was not idle?
Automated Adjustment of CPs

- Processor status change messages appear on the Linux console
- z/VM also issues HCPGSP2629I
Next let's look at the memory management features
Automated Adjustment of Memory

- cpuplugd memory management utilizes CMM (CMM1)
- The cpuplug daemon determines how much memory to add or remove based upon the rules you put in place
- It is based upon a configurable interval you set
- The memory increment added or removed is also configurable
- Separate plug and unplug rules are used for memory
- There are NO default memory plug and unplug rules
- If you start cpuplugd without any configuration changes it will manage CPUs but NOT memory.
Automated Adjustment of Memory

- Writing memory plug and unplug rules
  - `apcr`: the amount of page cache reads as listed in `vmstat bi/bo`
  - `freemem`: the amount of free memory (in megabyte)
  - `swaprate`: the number of swapin and swapout operations

- CMM pool size and increment
  - `CMM_MIN`: min size of the static page pool (default 0)
  - `CMM_MAX`: max size of the static page pool (default 8192 pages)
  - `CMM_INC`: amount added/removed (default 256 pages or 1MB)

- `apcr` can be used to gauge the IO load on Linux system. With heavier IO rates you may want to allow the system to utilize more memory to help improve performance. This memory would get utilized by pagecache.
Automated Adjustment of Memory

- Cpuplugd and CMM1 currently will NOT release pagecache memory
- With the default interval of 10 seconds, in a memory constrained situation you will only add 6MB/min or 360MB/hr
- With instantaneous allocations in GB by some application environments this has the potential to impact application performance, unless increased
- Lets take a brief look at an example
Automated Adjustment of Memory

- This guest currently only has a small amount of memory resident
- In order to see the impact of CPU hotplug we will make more memory resident
Automated Adjustment of Memory

rgylx0e4:/etc # free -m

```
Mem:   5018 167 4850 0 6 56
-/+ buffers/cache: 104 4913
Swap:   0 0 0
```

rgylx0e4:/etc # dd if=/dev/zero of=/mnt/testfile bs=1M count=10000

- The entire 5GB of memory is almost all free
- Only 5MB used as cache
- The “dd” command is used in this example to populate page cache and consume memory
Automated Adjustment of Memory

```
rgylx0e4:/etc # free -m

    total  used  free   shared   buffers  cached
Mem:       5018  167  4850     0        6       56
-/+ buffers/cache:    104  4913
Swap:       0     0       0

rgylx0e4:/etc # dd if=/dev/zero of=/mnt/testfile bs=1M count=10000
  dd: writing `/mnt/testfile': No space left on device
2085+0 records in
2084+0 records out
2185232384 bytes (2.2 GB) copied, 127.398 s, 17.2 MB/s

rgylx0e4:/etc # free -m

    total  used  free   shared   buffers  cached
Mem:       5018  2260  2757     0        7       2067
-/+ buffers/cache:    106  4913
Swap:       0     0       0
```
Automated Adjustment of Memory

- The memory consumption has more than doubled.
Automated Adjustment of Memory

rgylx0e4:/etc # cpuplugd -V -f -c /etc/sysconfig/cpuplugd
found cpu_min value: 1
found cpu_max value: 0
found update value: 10
found cmm_min value: 0
found cmm_max valus: 8192000
found cmm_inc value: 25600
found the following rule: HOTPLUG = (loadavg+0.75>onumcpus)||(idle<25.0)
found the following rule: HOTUNPLUG = (loadavg<onumcpus-0.25)||(idle>50)
found the following rule: MEMPLUG = freemem<250
found the following rule: MEMUNPLUG = freemem>1750|swaprate>1
Detected System running in z/VM mode
Valid CPU hotplug configuration detected.
Can not open /proc/sys/vm/cmm_pages
The memory hotplug function will be disabled.
----------------------------------
update_interval: 10 s
cpu_min: 1
cpu max: 2
Automated Adjustment of Memory

rgylx0e4:/etc # modprobe cmm
rgylx0e4:/etc # cpuplugd -V -f -c /etc/sysconfig/cpuplugd
found cpu_min value: 1
found cpu_max value: 0
found update value: 10
found cmm_min value: 0
found cmm_max value: 8192000
found cmm_inc value: 25600
found the following rule: HOTPLUG = (loadavg+0.75>onumcpus) | (idle<25.0)
found the following rule: HOTUNPLUG = (loadavg<onumcpus-0.25) | (idle>50)
found the following rule: MEMPLUG = freemem<250
found the following rule: MEMUNPLUG = freemem>1750 | swaprate>1
Detected System running in z/VM mode
Valid CPU hotplug configuration detected.
Valid memory hotplug configuration detected.
Automated Adjustment of Memory

```plaintext
maximum cpu limit is reached
update_interval: 10 s
cpu_min: 0
cpu_max: 8192000
swappage: 0
acpr: 0
cmm_inc: 25660
free memory: 2758 MB
------------------------------------------------------------
cmm_pages: 0
------------------------------------------------------------
memplug: (freemem) < (250.000000)
memunplug: ((freemem) > (1750.000000) | (swappage) > (1.000000))
changed number of pages permanently reserved to 25660

update_interval: 10 s
cpu_min: 1
cpu_max: 2
loadavg: 0.040000
idle percent = 199.900000
numcups 2
runnable proc: 1
------------------------------------------------------------
onumcups: 2
------------------------------------------------------------
hotplug: ((loadavg) + (0.750000)) > (onumcups)) | ((idle) < (25.000000))
hotunplug: ((loadavg) < ((onumcups) - (0.250000)) | ((idle) > (50.000000)))
------------------------------------------------------------
cpu with id 1 is currently online and will be disabled
update_interval: 10 s
cpu_min: 0
cpu_max: 8192000
swappage: 0
acpr: 9
cmm_inc: 25600
free memory: 2659 MB
------------------------------------------------------------
cmm_pages: 25600
------------------------------------------------------------
memplug: (freemem) < (250.000000)
memunplug: ((freemem) > (1750.000000) | (swappage) > (1.000000))
changed number of pages permanently reserved to 51200
```

~ 100MB reserved

~ 200MB reserved
Automated Adjustment of Memory

~ 1.1GB reserved

Page reservation stabilized
Automated Adjustment of Memory

- Stabilized 281600 page of memory
- Rules say to unplug memory while freemem > 1750 MB
- The trace shows it is down to 1655 MB
Automated Adjustment of Memory

```
rgylx0e4:~ $ free -m

    total used  free shared buffers cached
Mem:    5018  3363  1655  0      7  2147
-/+ buffers/cache:  1208  3810
Swap:    0      0      0
```

- Note that the “cached” memory is still 2147. cpuplugd does not currently act upon “cached” memory.
- “used” memory has increased. The pages we reserved with CMM are considered “used”.
Automated Adjustment of Memory

- The size of the memory reserved from CMM can be queried by reading `/proc/sys/vm/cmm_pages`
- A trace is not required to obtain that point in time value
Automated Adjustment of Memory

- A 3 is echoed into drop_caches to cause the current page_cache to be dropped
- This decreased the “used” total and increases the free memory total
- Since our cpuplugd memory rule is a function of “freemem” we can now return even more real memory to the hypervisor
Automated Adjustment of Memory

minimum cpu limit is reached

update_interval: 10 s
cmm_min: 0
cmm_max: 9192000
swaprate: 0
apcr: 1
cmm inc: 25600
free memory: 2492 MB

cmm pages: 614400

memplug: (freesmem) < (250.000000) |
memunplug: ((freesmem) > (1750.000000)) | ((swaprate) > (1.000000))

changed number of pages permanently reserved to 640000

update interval, 10 s

cpu_min: 1
cpu_max: 2
loadavg: 0.000000
idle percent = 99.000000
numcpus: 2
runable_proc: 1
numcpus: 1

hotplug: ((loadavg) + (0.750000)) > (onumcpus)) | ((idle) < (25.000000))
hotunplug: ((loadavg) < (onumcpus) - (0.250000)) | ((idle) > (50.000000))

minimum cpu limit is reached

update interval: 10 s
cmm_min: 0
cmm_max: 9192000
swaprate: 0
apcr: 0
cmm inc: 25600
free memory: 2392 MB

cmm pages: 640000

memplug: (freesmem) < (250.000000)
memunplug: ((freesmem) > (1750.000000)) | ((swaprate) > (1.000000))

changed number of pages permanently reserved to 665600

~ 2.5 GB reserved

~ 2.6 GB reserved
Automated Adjustment of Memory

```c
hotunplug: ((loadavg) < ((onumcpus) - (0.250000))) | ((idle) > (50.000000))
minimum cpu limit is reached

update_interval: 10 s
cmm_min: 0
cmm_max: 8192000
swaprate: 0
apcr: 2
num_inc: 25600
free memory: 1690 MB

memplug: (freemem) < (250.000000)
memunplug: ((freemem) > (1750.000000)) | ((swaprate) > (1.000000))

update_interval: 10 s
cpu_min: 1
cpu_max: 2
loadavg: 0.000000
idle_percent = 100.000000
numcpus 2
runable_proc: 1

onumcpus: 1

hotplug: (((loadavg) + (0.750000)) > (onumcpus)) | ((idle) < (25.000000))
hotunplug: ((loadavg) < ((onumcpus) - (0.250000))) | ((idle) > (50.000000))
minimum cpu limit is reached

update_interval: 10 s
cmm_min: 0
cmm_max: 8192000
swaprate: 0
apcr: 2
num_inc: 25600
free memory: 1690 MB

memplug: (freemem) < (250.000000)
memunplug: ((freemem) > (1750.000000)) | ((swaprate) > (1.000000))
```

~ 3.3 GB reserved

Reserved page count stabilized
CPU Hotplug Summary

• CPU Hotplug memory management will NOT release page cache memory on its own
• In our example, the CMM module had to be loaded before starting cpuplugd
• Understand how much memory you want to allow CMM to claim and the rate at which you will return memory to the system for use. The last thing you want is a failing memory allocation, or adverse performance impact.
CPU Hotplug Summary

- Under heavier IO load you might want to make more free memory available to Linux.
- The goal is to allow the Linux to dynamically return pages of memory to z/VM when they are not in use, and to allow the entire system to operate more efficiently.
- The amount of memory required an application to run is a function of the application program code, the workload volume, and any other software added to monitor or manage the environment.
Linux on System z Suspend and Resume
Suspend and Resume - Uses

- Possible Uses:
  - Linux instance with middleware that has long startup or initialization time.
  - Instances with long idle periods during the day where the server is not used. Use to free memory and processor resources while suspended.
  - Resume a guest to central storage, moments before it is needed. (Assuming you know when it will be needed again)
  - Provide consistency? Suspend, FlashCopy, and Resume?
Suspend and Resume - Planning

- Planning for Suspend and Resume
  - Kernel 2.6.31 or higher
  - RHEL 6 / SLES 11 or higher
  - Suspended Linux is written to the designed swap disk
  - Must be large enough to hold the memory footprint of the Linux server

- Restrictions
  - No hotplug memory since the last boot
  - No CLAW Device Driver
  - All tape devices closed and unloaded
  - No DCSS with exclusive writable access
Suspend and Resume – Planning

• While suspended:
  • Don’t alter the data on the swap device with the suspend Linux
  • DCSSs and NSSs used must remain unchanged
  • Avoid real and virtual hardware configuration changes

• For all the restrictions and configuration information see:
  • Linux on System z Device Drivers, Features, and Commands SC33-8411-x
Suspend and Resume - Planning

• Kernel Parameters
  • resume=<device node for swap partition>
  • no_console_suspend - Allows you to see console messages longer in to the suspend process
  • noresume - Skip resume of previously suspended system

• Consider swap file priorities
  • You might want to make swap partition for suspend the lowest priority

• Utilize echo disk > /sys/power/state
• Utilize SIGNAL SHUTDOWN and /etc/inittab CTRL-ALT-DELETE to suspend your system
Suspend and Resume - Preparing

rgy1xd85:/etc # cat /etc/zipl.conf
# Modified by YaST2. Last modification on Sat Apr 23 15:48:27 EDT 2011
[defaultboot]
defaultmenu = menu

###Don't change this comment - YaST2 identifier: Original name: linux###
[SLES11_SP1V1]
  image = /boot/image-2.6.32.29-0.3-default
  target = /boot/zipl
  ramdisk = /boot/initrd-2.6.32.29-0.3-default,0x2000000
  parameters = "root=/dev/disk/by-path/ccw-0.0.0200-part1 resume=/dev/sda2 no_console_suspend"
Suspend and Resume - Preparing

rgylxd85:/etc/sysconfig # zipl
Using config file '/etc/zipl.conf'
Building bootstrap in '/boot/zipl'
Building menu 'menu'
Adding #1: IPL section 'SLBS11_SP1V1' (default)
Adding #2: IPL section 'FailsafeV2'
Adding #3: IPL section 'ipl'
Preparing boot device: dasda (0200).
Done.
rgylxd85:~ # uname -a
Linux rgylxd85 2.6.32.29-0.3-default #1 SMP 2011-02-25 13:36:59 +0100 s390x s390x
rgylxd85:~ # cat /proc/swaps
Filename Type Size Used Priority
/dev/sda1 partition 5237148 0 -1
/dev/sda2 partition 5245212 0 1
rgylxd85:~ # vmstat 1
procs memory swap-- io-- system-- cpu------
 r  b  swpd  free  buff  cache  si  so  bi  bo  in  cs us sy id wa st
 0  0   0  2956988  6488  44796  0  0  272 19  0 108  2  1 95  2  0
 0  0   0  2956988  6488  44804  0  0  0  0  0 19  0  0 100  0  0
 0  0   0  2957004  6488  44852  0  0  0  0  0 10  0  0 100  0  0
^C
rgylxd85:~ # echo disk > /sys/power/state
Suspend and Resume - Suspending

16:10:35 qdio: 0.0.0602 OSA on SC e using AI:1 QEBSM:0 PCI:1 TDD:1 SIGA:RW A0
16:10:35 qeth.736dae: 0.0.0600: Device is a Guest LAN QDIO card (level: V611)
16:10:35 with link type GuestLAN QDIO (portname: )
16:10:35 qeth.47953b: 0.0.0600: Hardware IP fragmentation not supported on eth0
16:10:35 qeth.066069: 0.0.0600: Inbound source MAC-address not supported on eth0

16:10:35 qeth.d7fdb4: 0.0.0600: VLAN enabled
16:10:35 qeth.e90c78: 0.0.0600: Multicast enabled
16:10:35 qeth.5a9d02: 0.0.0600: IPV6 enabled
16:10:35 qeth.184d8a: 0.0.0600: Broadcast enabled
16:10:35 qeth.dac2aa: 0.0.0600: Using SW checksumming on eth0.
16:10:35 qeth.9c4c89: 0.0.0600: Outbound TSO not supported on eth0
16:10:35 PM: Saving image data pages (45435 pages) ... 0% 1%
2% 3% 4% 5% 6% 7% 8% 9% 10%
11% 12% 13% 14% 15% 16% 17% 18% 19% 20%
21% 22% 23% 24% 25% 26% 27% 28% 29% 30%
31% 32% 33% 34% 35% 36% 37% 38% 39% 40%
41% 42% 43% 44% 45% 46% 47% 48% 49% 50%
51% 52% 53% 54% 55% 56% 57% 58% 59% 60%
61% 62% 63% 64% 65% 66% 67% 68% 69% 70%
71% 72% 73% 74% 75% 76% 77% 78% 79% 80%
81% 82% 83% 84% 85% 86
16:10:50 PM: **Wrote 181740 kbytes in 1.22 seconds (148.96 MB/s)**

16:10:50 PM: S

16:10:50 md: stopping all md devices.

16:10:57 sd 1:0:1:1077035025: [sdb] Synchronizing SCSI cache

16:10:57 sd 0:0:0:1077035025: [sda] Synchronizing SCSI cache

16:10:57 Disabling non-boot CPUs ...

16:10:57 01: HCPGSP2629I The virtual machine is placed in CP mode due to a SIGP stop from CPU 01.

16:10:57 00: HCPGSP2629I The virtual machine is placed in CP mode due to a SIGP stop from CPU 00.
Suspend and Resume – Resume Attempt

16:11:43 io scheduler cfq registered
16:11:43 cio.b5d5f6: Channel measurement facility initialized using format extended (mode autodetected)
16:11:43 TCP cubic registered
16:11:43 registered taskstats version 1
16:11:43 Freeing unused kernel memory: 228k freed
16:11:43 doing fast boot
16:11:43 Creating device nodes with udev
16:11:43 udevd version 128 started
16:11:43 dasd-eckd.90fb0d: 0.0.0200: New DASD 3390/0A (CU 3990/01) with 3336 cylinders, 15 heads, 224 sectors
16:11:43 dasd-eckd.412b53: 0.0.0200: DASD with 4 KB/block, 2403360 KB total size, 48 KB/track, compatible disk layout
16:11:43 dasda:VOL1/ 0X0200: dasda1
16:11:43 mount: devpts already mounted or /dev/pts busy
16:11:43 mount: according to mtab, devpts is already mounted on /dev/pts
16:11:43 Boot logging started on /dev/ttyS0 (/dev/console) at Sat Apr 23 16:11:26 2011
16:11:43 kjournald starting. Commit interval 15 seconds
16:11:43 EXT3 FS on dasda1, internal journal
16:11:43 EXT3-fs: mounted filesystem with ordered data mode.
16:11:53 Trying manual resume from /dev/sda2
Suspend and Resume – Resume Attempt

16:11:53 resume device /dev/sda2 not found (ignoring)
16:11:53 Trying manual resume from /dev/sda2
16:11:53 resume device /dev/sda2 not found (ignoring)
16:11:53 Waiting for device /dev/disk/by-path/ccw-0.0.0200-part1 to appear: ok
16:11:53 fsck from util-linux-ng 2.16
16:11:53 [/sbin/fsck.ext3 (1) -- /] fsck.ext3 -a /dev/dasdal
16:11:53 /dev/dasdal: recovering journal
16:11:53 /dev/dasdal: clean, 4239/150176 files, 67293/600276 blocks
16:11:53 Mounting root /dev/disk/by-path/ccw-0.0.0200-part1
16:11:53 mount -o rw,acl,user_xattr -t ext3 /dev/disk/by-path/ccw-0.0.0200-part1
  /root
16:12:01 INIT: version 2.86 booting
16:12:01 System Boot Control: Running /etc/init.d/boot
16:12:01 Mounting sysfs at /sys..done
16:12:01 Mounting debugfs at /sys/kernel/debug..done
16:12:01 Copying static /dev content..done
16:12:01 Mounting devpts at /dev/pts..done
16:12:01 Boot logging started on /dev/ttyS0(/dev/console) at Sat Apr 23 16:12:0
1 2011
16:12:01 Starting udevd: udevd version 128 started
16:12:01 dasd-eckd.90fb0d: 0.0.0202: New DASD 3390/0A (CU 3990/01) with 3338 cy
Suspend and Resume – Attempt Summary

• The resume on the previous page failed
• The initial ram disk did not include zfcp, however the swap file on the SCSI device is required for the resume operation
• This example only had 3390 model 3 volumes available and needed to be able to suspend guests larger than 2.2 GB
• This issue is easily resolved by adding zfcp to the initrd
Suspend and Resume – Preparing zfcp

## Path: System/Kernel
## Description:
## Type: string
## Command: /sbin/mkinitrd

# This variable contains the list of modules to be added to the initial ramdisk by calling the script "mkinitrd"
# (like drivers for scsi-controllers, for lvm or reiserfs)
# INITRD_MODULES="jbd ext3 zfcp"
Suspend and Resume - Preparing

rgylxd85:/etc/sysconfig # mknitrd

Kernel image: /boot/image-2.6.32.29-0.3-default
Initrd image: /boot/initrd-2.6.32.29-0.3-default
Root device: /dev/disk/by-path/ccw-0.0.0200-part1 (/dev/dasda1) (mounted on / as ext3)
Resume device: /dev/sda2
Kernel Modules: jbd mbcache ext3 scsi_mod scsi_tgt scsi_transport_fc qdio zfcp dasd_mod dasd_ec
Features: block dasd zfcp resume.userspace resume.kernel

27394 blocks
rgylxd85:/etc/sysconfig # zipl
Using config file '/etc/zipl.conf'
Building bootstrap in '/boot/zipl'
Building menu 'menu'
Adding #1: IPL section 'SLES11_SP1V1' (default)
Adding #2: IPL section 'FailsafeV2'
Adding #3: IPL section 'ipl'
Preparing boot device: dasda (0200).
Done.
rgylxd85:/etc/sysconfig #
Suspend and Resume - Suspending

rgylxd85:~ # cat /proc/swaps
Filename          Type    Size    Used  Priority
/dev/sda1        partition 5237148 0    -1
/dev/sda2        partition 5245212 0    1

rgylxd85:~ # vmstat 1
procs ------------memory----------- ---swap-- -----io---- -system-- -----cpu-----
 r  b  swpd  free  buff  cache  si  so  bi  bo  in  cs  us  sy  id  wa  st
 0  0   0    2957980 6424 43892   0   0   390  23   0 164  2  1 94  2  0
 0  0   0    2957980 6424 43892   0   0     0    0   0   0  0  0 100  0  0
 0  0   0    2957964 6424 43932   0   0     0    0   0 10  0  0 100  0  0

^C
rgylxd85:~ # echo disk > /sys/power/state
Suspend and Resume - Suspending

16:21:15 PM: Freezing user space processes ... (elapsed 0.00 seconds) done.
16:21:15 PM: Freezing remaining freezable tasks ... (elapsed 0.00 seconds) done.
16:21:15 PM: Preallocating image memory... 16:21:15 done (allocated 45601 pages)

16:21:15 PM: Allocated 182404 kbytes in 0.12 seconds (1520.03 MB/s)
16:21:15 sd 1:0:3:1077035025: [sdb] Synchronizing SCSI cache
16:21:15 sd 0:0:5:1077035025: [sda] Synchronizing SCSI cache
16:21:16 01: HCPGSP2629I The virtual machine is placed in CP mode due to a SIGP
stop from CPU 01.
16:21:16 01: HCPGSP2627I The virtual machine is placed in CP mode due to a SIGP
initial CPU reset from CPU 00.
16:21:16 Disabling non-boot CPUs ...
16:21:16 cpu.f76a91: Processor 1 stopped
16:21:16 PM: Creating hibernation image:
16:21:16 PM: Need to copy 45066 pages
16:21:16 PM: Hibernation image created (45066 pages copied)
16:21:16 Enabling non-boot CPUs ...
16:21:16 cpu.17772b: Processor 1 started, address 0, identification 12EBBE
16:21:16 CPU1 is up
16:21:16 qdio: 0.0.0.2000 ZFCP on SC 1 using AI:1 QEBSM:1 PCI:1 TDD:1 SIGA: W AO
16:21:16 qdio: 0.0.1000 ZFCP on SC 0 using AI:1 QEBSM:1 PCI:1 TDD:1 SIGA: W AO
Suspend and Resume - Suspending

16:21:16 qdio: 0.0.0602 OSA on SC e using AI:1 QEBSM:0 PCI:1 TDD:1 SIGA:RW AO
16:21:16 qeth.736dae: 0.0.0600: Device is a Guest LAN QDIO card (level: V611)
16:21:16 with link type GuestLAN QDIO (portname: )
16:21:16 qeth.47953b: 0.0.0600: Hardware IP fragmentation not supported on eth0
16:21:16 qeth.066069: 0.0.0600: Inbound source MAC-address not supported on eth0

16:21:16 qeth.d7fdb4: 0.0.0600: VLAN enabled
16:21:16 qeth.e90c78: U.U.U6UU: Multicast enabled
16:21:16 qeth.5a9d02: 0.0.0600: IPV6 enabled
16:21:16 qeth.184d8a: 0.0.0600: Broadcast enabled
16:21:16 qeth.dac2aa: 0.0.0600: Using SW checksumming on eth0.
16:21:16 qeth.9c4e88: O.O.0800: Outbound TSO not supported on eth0

16:21:16 PM: Saving image data pages (45155 pages) ...

<table>
<thead>
<tr>
<th>%</th>
<th>2%</th>
<th>3%</th>
<th>4%</th>
<th>5%</th>
<th>6%</th>
<th>7%</th>
<th>8%</th>
<th>9%</th>
<th>10%</th>
</tr>
</thead>
<tbody>
<tr>
<td>11%</td>
<td>12%</td>
<td>13%</td>
<td>14%</td>
<td>15%</td>
<td>16%</td>
<td>17%</td>
<td>18%</td>
<td>19%</td>
<td>20%</td>
</tr>
<tr>
<td>21%</td>
<td>22%</td>
<td>23%</td>
<td>24%</td>
<td>25%</td>
<td>26%</td>
<td>27%</td>
<td>28%</td>
<td>29%</td>
<td>30%</td>
</tr>
<tr>
<td>31%</td>
<td>32%</td>
<td>33%</td>
<td>34%</td>
<td>35%</td>
<td>36%</td>
<td>37%</td>
<td>38%</td>
<td>39%</td>
<td>40%</td>
</tr>
<tr>
<td>41%</td>
<td>42%</td>
<td>43%</td>
<td>44%</td>
<td>45%</td>
<td>46%</td>
<td>47%</td>
<td>48%</td>
<td>49%</td>
<td>50%</td>
</tr>
<tr>
<td>51%</td>
<td>52%</td>
<td>53%</td>
<td>54%</td>
<td>55%</td>
<td>56%</td>
<td>57%</td>
<td>58%</td>
<td>59%</td>
<td>60%</td>
</tr>
<tr>
<td>61%</td>
<td>62%</td>
<td>63%</td>
<td>64%</td>
<td>65%</td>
<td>66%</td>
<td>67%</td>
<td>68%</td>
<td>69%</td>
<td>70%</td>
</tr>
<tr>
<td>71%</td>
<td>72%</td>
<td>73%</td>
<td>74%</td>
<td>75%</td>
<td>76%</td>
<td>77%</td>
<td>78%</td>
<td>79%</td>
<td>80%</td>
</tr>
<tr>
<td>81%</td>
<td>82%</td>
<td>83%</td>
<td>84%</td>
<td>85%</td>
<td>86</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Suspend and Resume – Suspended/Resume

% 87% 88% 89% 90% 91% 92% 93% 94% 95% 96
% 97% 98% 99% 100% done
16:21:21 PM: Wrote 180620 kbytes in 1.18 seconds (153.06 MB/s)
16:21:21 PM: $|
16:21:25 sd 1:0:3:1077035025: [sdb] Synchronizing SCSI cache
16:21:25 sd 0:0:5:1077035025: [sda] Synchronizing SCSI cache
16:21:25 01: HCPGSP2629I The virtual machine is placed in CP mode due to a SIGP stop from CPU 01.
16:21:25 00: HCPGSP2629I The virtual machine is placed in CP mode due to a SIGP stop from CPU 00.
16:21:33 00: IPL 200 CIFAR
16:21:33 00: zIPL v1.8.0-44.45.2 interactive boot menu
16:21:33 00:
16:21:33 00: 0. default (SLES11_SP1V1)
16:21:33 00:
16:21:33 00: 1. SLES11_SP1V1
16:21:33 00: 2. FailsafeV2
16:21:33 00: 3. ipl
16:21:33 00:
16:21:33 00: Note: VM users please use 'cp vi vmsg <number> <kernel-parameters>'
Suspend and Resume - Resuming

16:21:54 cio.b5d5f8: Channel measurement facility initialized using format extended (mode autodetected)
16:21:54 TCP cubic registered
16:21:54 registered taskstats version 1
16:21:54 Freeing unused kernel memory: 228k freed
16:21:54 doing fast boot
16:21:54 SCSI subsystem initialized
16:21:54 Creating device nodes with udev
16:21:54 udevd version 128 started
16:21:54 scsi0 : zfcp
16:21:54 qdio: 0.0.1000 ZFCP on SC 0 using AI:1 QEBSM:1 PCI:1 TDD:1 SIGA: W A0
16:21:54 dasd-eckd.90fb0d: 0.0.0200: New DASD 3390/QA (CU 3990/01) with 3338 cylinders, 15 heads, 224 sectors
16:21:54 dasd-eckd.412b53: 0.0.0200: DASD with 4 KB/block, 2403360 KB total size, 48 KB/track, compatible disk layout
16:21:54 dasda:VOL1/ 0x0200: dasda1
16:21:54 scsi 0:0:5:1077035025: Direct-Access IBM 2107900 .204 PQ: 0 ANSI: 5
16:21:54 sd 0:0:5:1077035025: [sda] 20971520 512-byte logical blocks: (10.7 GB/10.0 GiB)
16:21:54 sd 0:0:5:1077035025: [sda] Write Protect is off
16:21:54 sd 0:0:5:1077035025: [sda] Write cache: enabled, read cache: enabled,
Suspend and Resume - Resuming

do esn't support DPO or FUA
16:21:54 sda: sda1 sda2
16:21:54 sd 0:0:5:1077035025: [sda] Attached SCSI disk
16:21:54 mount: devpts already mounted or /dev/pts busy
16:21:54 mount: according to mtab, devpts is already mounted on /dev/pts
16:21:54 Boot logging started on /dev/ttyS0 (/dev/console) at Sat Apr 23 16:21:45 2011
16:21:54 PM: Starting manual resume from disk
16:21:54 Freezing user space processes ... (elapsed 0.00 seconds) done.
16:21:54 Freezing remaining freeable tasks ... (elapsed 0.00 seconds) done.
16:21:54 PM: Loading image data pages (45155 pages) ... 0% 1%
  2%  3%  4%  5%  6%  7%  8%  9% 10% 11%
 12% 13% 14% 15% 16% 17% 18% 19% 20% 21%
 22% 23% 24% 25% 26% 27% 28% 29% 30% 31%
 32% 33% 34% 35% 36% 37% 38% 39% 40% 41%
 42% 43% 44% 45% 46% 47% 48% 49% 50% 51%
 52% 53% 54% 55% 56% 57% 58% 59% 60% 61%
 62% 63% 64% 65% 66% 67% 68% 69% 70% 71%
 72% 73% 74% 75% 76% 77% 78% 79% 80% 81%
 82% 83% 84% 85% 86% 87% 88% 89% 90% 91%
 92% 93% 94% 95% 96% 97% 98% 99% 100% done
16:21:54 PM: Read 180620 kbytes in 1.31 seconds (137.87 MB/s)
Suspend and Resume - Resuming

16:21:54 sd 0:0:5:1077035025: [sda] Synchronizing SCSI cache
16:22:06 01: HCPGSP2629I The virtual machine is placed in CP mode due to a SIGP stop from CPU 01.
16:22:06 01: HCPGSP2627I The virtual machine is placed in CP mode due to a SIGP initial CPU reset from CPU 00.
16:22:07 Disabling non-boot CPUs ...
16:22:07 cpu.f78a91: Processor 1 stopped
16:22:07 PM: Creating hibernation image:
16:22:07 PM: Need to copy 45066 pages
16:22:07 Enabling non-boot CPUs ...
16:22:07 cpu.17772b: Processor 1 started, address 0, identification 12EBBE
16:22:07 CPU1 is up
16:22:07 qdio: 0.0.2000 ZFCP on SC 1 using AI:1 QEBSM:1 PCI:1 TDD:1 SIGA: W AO
16:22:07 qdio: 0.0.1000 ZFCP on SC 0 using AI:1 QEBSM:1 PCI:1 TDD:1 SIGA: W AO
16:22:07 qdio: 0.0.0602 OSA on SC e using AI:1 QEBSM:0 PCI:1 TDD:1 SIGA:RW AO
16:22:07 qeth.736dae: 0.0.0600: Device is a Guest LAN QDIO card (level: V611)
16:22:07 with link type GuestLAN QDIO (portname: )
16:22:07 qeth.47953b: 0.0.0600: Hardware IP fragmentation not supported on eth0
16:22:07 qeth.066069: 0.0.0600: Inbound source MAC-address not supported on eth0

16:22:07 qeth.d7f9d4: 0.0.0600: VLAN enabled
16:22:07 qeth.e90c78: 0.0.0600: Multicast enabled
Suspend and Resume - Resuming

16:22:07 qeth.5a9d02: 0.0.0600: IPV6 enabled
16:22:07 qeth.184d8a: 0.0.0600: Broadcast enabled
16:22:07 qeth.dac2aa: 0.0.0600: Using SW checksumming on eth0.
16:22:07 qeth.9c4c89: 0.0.0600: Outbound TSO not supported on eth0
16:22:07 Restarting tasks ... done.
16:22:11 Apr 23 16:22:07 rgylxd85 kernel: Freezing user space processes ... (elapsed 0.00 seconds) done.
16:22:11 Apr 23 16:22:07 rgylxd85 kernel: Freezing remaining freezable tasks ... (elapsed 0.00 seconds) done.
16:22:11 Apr 23 16:22:07 rgylxd85 kernel: Disabling non-boot CPUs ...
16:22:11 Apr 23 16:22:07 rgylxd85 kernel: Enabling non-boot CPUs ...
16:22:11 Apr 23 16:22:07 rgylxd85 kernel: CPU1 is up
Suspend and Resume

If the suspend and resume are completed fast enough your TCP connections may not even drop. The above ssh session is an example of that.
Using SIGNAL SHUTDOWN to trigger a suspend
Suspend and Resume - /etc/inittab

#3:2345:respawn:/sbin/mingetty --noclear /dev/3270/ttycons dumb
# KVM hypervisor console:
#1:2345:respawn:/sbin/mingetty --noclear /dev/hvc0 linux

# what to do when CTRL-ALT-DEL is pressed
<F12>:
ca::ctrlaltdel:/sbin/shutdown -r -t 4 now
ca::ctrlaltdel:/bin/sh -c "/bin/echo disk > /sys/power/state || /sbin/shutdown -t3 -h now"

# not used for now:
pf::powerwait:/etc/init.d/powerfail start
pn::powerfailnow:/etc/init.d/powerfail now
#pn::powerfail:/etc/init.d/powerfail now
po::powerokwait:/etc/init.d/powerfail stop
sh:12345:powerfail:/sbin/shutdown -h now THE POWER IS FAILING

- By adding the modified ctrlaltdel entry to /etc/inittab you can suspend your Linux guest to a swap file when it receive a “Signal shutdown”.
- In the event the suspend fails, a “regular” shutdown would occur.
Suspend and Resume - signal

signal shutdown user rgylxd85 within 60
Ready; T=0.01/0.01 17:02:06

• Trigging a suspend from z/VM is easy once the Linux
  inittab update is in place.
• The standard signal shutdown command should very
  quickly suspend the guest
17:02:07 PM: Syncing filesystems ... 17:02:07 done.
17:02:07 Freezing user space processes ... (elapsed 0.00 seconds) done.
17:02:07 Freezing remaining freezable tasks ... (elapsed 0.00 seconds) done.
17:02:07 PM: Preallocating image memory... 17:02:07 done (allocated 45739 pages)

17:02:07 PM: Allocated 182956 kbytes in 0.12 seconds (1524.63 MB/s)
17:02:07 sd 1:0:2:1077035025: [sdb] Synchronizing SCSI cache
17:02:07 sd 0:0:0:1077035025: [sda] Synchronizing SCSI cache
17:02:07 01: HCPGSP2629I The virtual machine is placed in CP mode due to a SIGP stop from CPU 01.
17:02:07 01: HCPGSP2627I The virtual machine is placed in CP mode due to a SIGP initial CPU reset from CPU 00.
17:02:07 Disabling non-boot CPUs ...
17:02:07 cpu.f76a91: Processor 1 stopped
17:02:07 PM: Creating hibernation image:
17:02:07 PM: Need to copy 45190 pages
17:02:07 PM: Hibernation image created (45190 pages copied)
17:02:07 Enabling non-boot CPUs ...
17:02:07 cpu.17772b: Processor 1 started, address 0, identification 12EBBE
17:02:07 CPU1 is up
17:02:08 qdio: 0.0.1000 ZFCP on SC 0 using AI:1 QEBSM:1 PCI:1 TDD:1 SIGA: W AO
17:02:08 qdio: 0.0.2000 ZFCP on SC 1 using AI:1 QEBSM:1 PCI:1 TDD:1 SIGA: W AO
Suspend and Resume - Suspended

%  87%  88%  89%  90%  91%  92%  93%  94%  95%  96
%  97%  98%  99% 100% done
17:02:12 PM: Wrote 181116 kbytes in 1.12 seconds (181.71 MB/s)
17:02:12 PM: S |
17:02:12 md: stopping all md devices.
17:02:14 sd 1:0:2:1077035025: [sdb] Synchronizing SCSI cache
17:02:14 sd 0:0:0:1077035025: [sda] Synchronizing SCSI cache
17:02:14 Disabling non-boot CPUs ... 
17:02:15 01: HCPGSP2629I The virtual machine is placed in CP mode due to a SIGP stop from CPU 01.
17:02:15 00: HCPGIR450W CP entered; disabled wait PSW 00020001 80000000 00000000 000000FF
Suspend and Resume

- After the signal is received by the Linux guest we see that a sync is issued for the file systems.
- User space and other freezable tasks are then frozen.
- The hibernation image is created.
- The image is written to the swap partition.
- The CPUs and devices are stopped.
Suspend and Resume - Summary

• Great option for middleware with long startup times
• Linux hotplug memory should currently be avoided with suspend / resume
• Ensure your initial ramdisk has all the device drivers you need to access the swap file and /boot partition for resume
• Ensure your swap file has adequate space to store the Linux instance
• If the resume fails, a normal IPL will occur
References

- Linux on System z Device Drivers, Features, and Commands
  - SC33-8411-09
- z/VM CP Commands and Utilities Reference
  - SC24-6175-01
- z/VM Directory Maintenance Facility Commands Reference
  - SC24-6188-01
Thank You

Merci

Grazie

Danke

Obrigado

Gracias

Japanese

Spanish

Brazilian Portuguese

Italian

German

Arabic

Russian

Tamil

Traditional Chinese

Simplified Chinese

Korean

English

French

Thai

Hindi

多謝
Questions?

IBM

Richard G. Young
Certified I/T Specialist
IBM STG Lab Services
zVM & Linux on z Team Lead

777 East Wisconsin Ave
Milwaukee, WI 53202

Tel 414 921 4276
Fax 414 921 4276
Mobile 262 893 8662
Email: ryoung1@us.ibm.com