IBM and Linux: Community Innovation for your Business

Introduction: CMM & Linux CPU Hotplug

2nd European IBM / GSE Conference for z/VSE, z/VM and Linux on System z, Leipzig, 28. Oct 2008

Hans-Joachim Picht
IBM Linux Technology Center, Germany
hans@linux.vnet.ibm.com

Agenda

- Introduction
- Linux and z/VM technology
- Cooperative Memory Management (CMM)
- Collaborative Memory Management Assist (CMMA)
- CPU Hotplug
- The Linux on System z CPU and Memory Hotplug Daemon
- Performance
  - CMM
  - CPU Hotplug

z/VM LPARs and Virtual Machines (Guests)

Can create multiple z/VM instances.

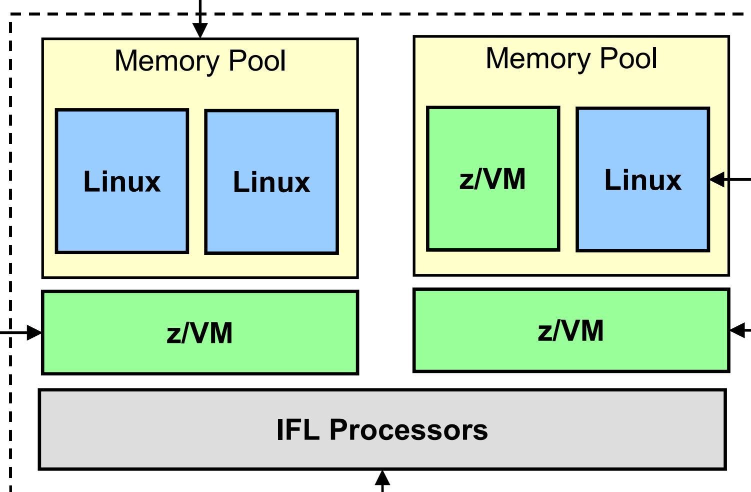
Can create separate CPU / memory pools

Separate by
*** Middleware**
*** Department**
*** Production/Dev/Etc.**
*** Performance**
*** Version / release**
*** Security domains**
*** Other ….**

z/VM instances can be at different release or patch levels.

CPU resources not used by one z/VM instance are available to another using LPAR settings.

Memory is not dedicated to a guest allowing whole guest instances to be paged out when not active.



System z - IBM's most reliable hardware

Low latency hardware virtualization

Guests can be created and started without a minimum CPU allocation.

Can run z/VM as a guest of z/VM

Guests can scale up to 64 virtual CPUs in size

New VMs can be defined easily

Problem statement

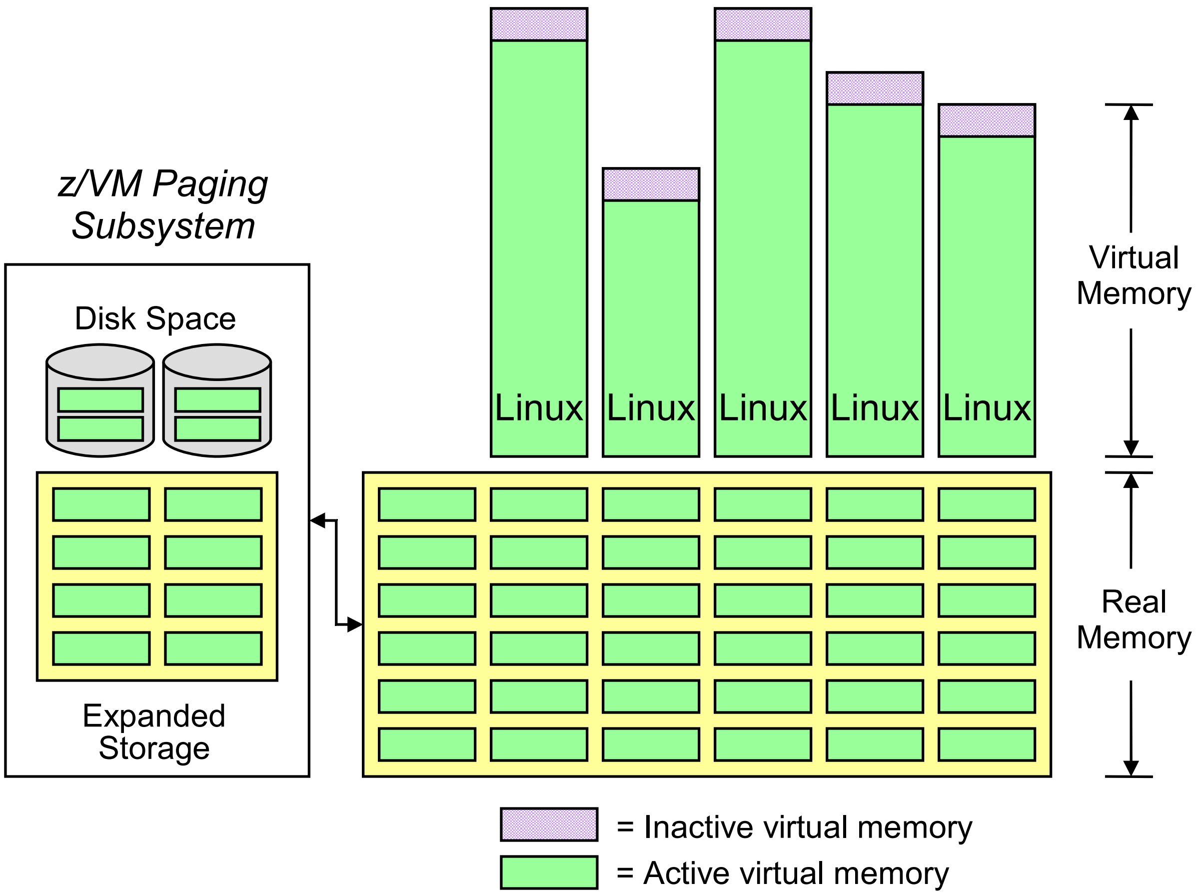
- Running Linux as a guest inside z/VM results in a different set of challenges compared to a Linux running on a discrete server
  - Multiple operating systems are hosted on top of the z/VM hypervisor
  - The problem of overcommitting physical memory has to be addressed
- The biggest problem is memory pressure: The lack of free, allocatable memory when needed
- Linux has the tendency to use all of its available memory.
  - Each guest OS assumes all resources are his alone
  - Any available memory will be used for things like filesystem cache
- As a result, the hypervisor (z/VM) might start swapping.
- To avoid excessive z/VM swapping the available Linux guest memory can be reduced.

Terminology & Technology

- Two methods available:
  - 9/10 4/5 VMRM-CMM (VM Resource Manager – Cooperative Memory Management) aka CMM1
    - Ballooning technique
    - The z/VM resource manager controls the size of the guests
    - Linux Support included in SLES 9, 10, RHEL 4 and 5
  - CMMA (Collaborative Memory Management Assist) aka CMM2
    - 10.1 Guest page hinting technique
    - Allows CP to "steal" pages based on the usage information
    - Target is to identify unused pages and non-dirty pages with a backing
- Both methods show performance improvements when z/VM hits a system memory constraint.

Cooperative Memory Management (CMM)

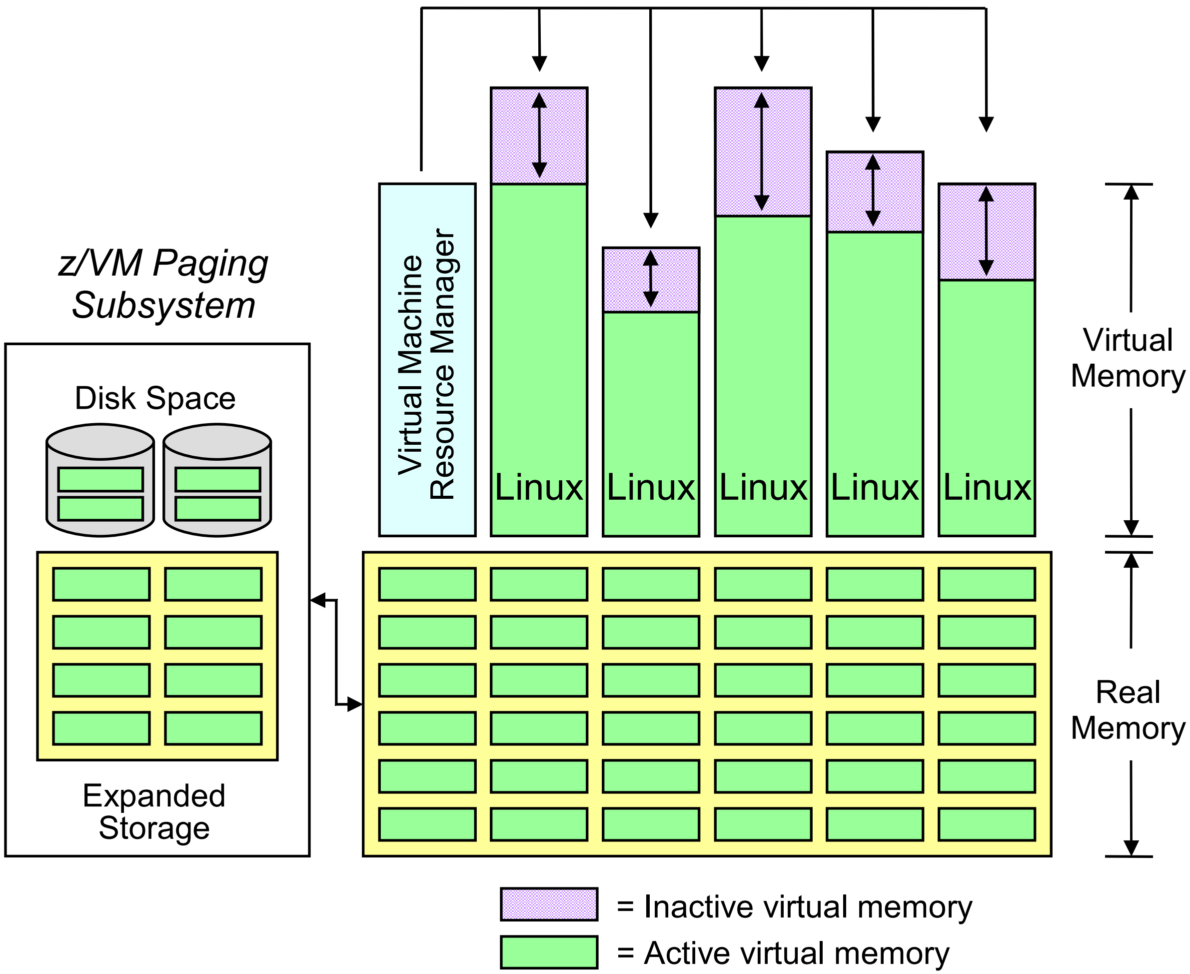
- The z/VM hypervisor maps guest virtual memory into the real memory storage of the System z machine.
- If there aren't enough real memory frames to contain all the active guests' virtual memory pages, some pages are moved to expanded storage
- When expanded storage is full, the pages of the guest are stored on the paging disk space (dasd)
- Inactive virtual memory pages inside the Linux guest must be recovered for use by other guest systems



Learn more at: http://ibm.com/servers/eserver/zseries/zvm/sysman/vmrm/vmrmcmm.html

Cooperative Memory Management (CMM)

- Solution: real memory constraint detected by z/VM Virtual Machine Resource Manager
- Linux images signaled to reduce virtual memory consumption
- Linux memory pages are released
- Demand on real memory and z/VM paging subsystem is reduced
- Helps improve overall system performance and guest image throughput

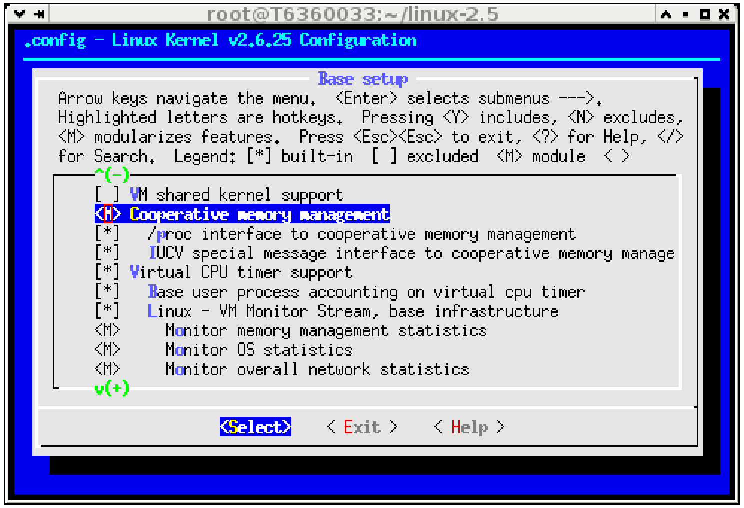


z/VM Command:

NOTIFY MEMORY <guestname1> <guestname2> ...

CMM: Linux Implementation

- The cmm Linux device driver receives the SHRINK request from VMRM
- Dynamic memory resizing is implemented via a “balloon”
- The cmm device driver allocates memory inside the Linux guest and informs CP that it no longer has to manage these pages
- Benefits:
  - This reduces the guest's memory footprint
  - The guest is forced to reclaim pages currently used for read and write cache



- By growing and shrinking the balloon, CP can deal with changing memory requirements by different Linux guests within z/VM
- The cmm device driver can either be directly compiled into the Linux Kernel or loaded as a module

[root@t6360033 ~]# **modprobe cmm**

CMM1 user interfaces within a Linux Guest

- `sysctl` or `/proc` user interface
- `/proc/sys/vm/cmm_pages`
  - Read to query number of pages permanently reserved
  - Write to set new target (will be achieved over time)
- `/proc/sys/vm/cmm_timed_pages`
  - Read to query number of pages temporarily reserved
  - Write increment to add to target
- `/proc/sys/vm/cmm_timeout`
  - Holds pair of N pages / X seconds (read/write)
  - Every time X seconds have passed, release N temporary pages
- IUCV special message interface
  - CMMSHRINK/CMMRELEASE/CMMREUSE
  - Same as cmm_pages/cmm_timed_pages/cmm_timeout write

Configuring Linux for CMM

- z/VM

  NOTIFY MEMORY T6360033 LNX00005

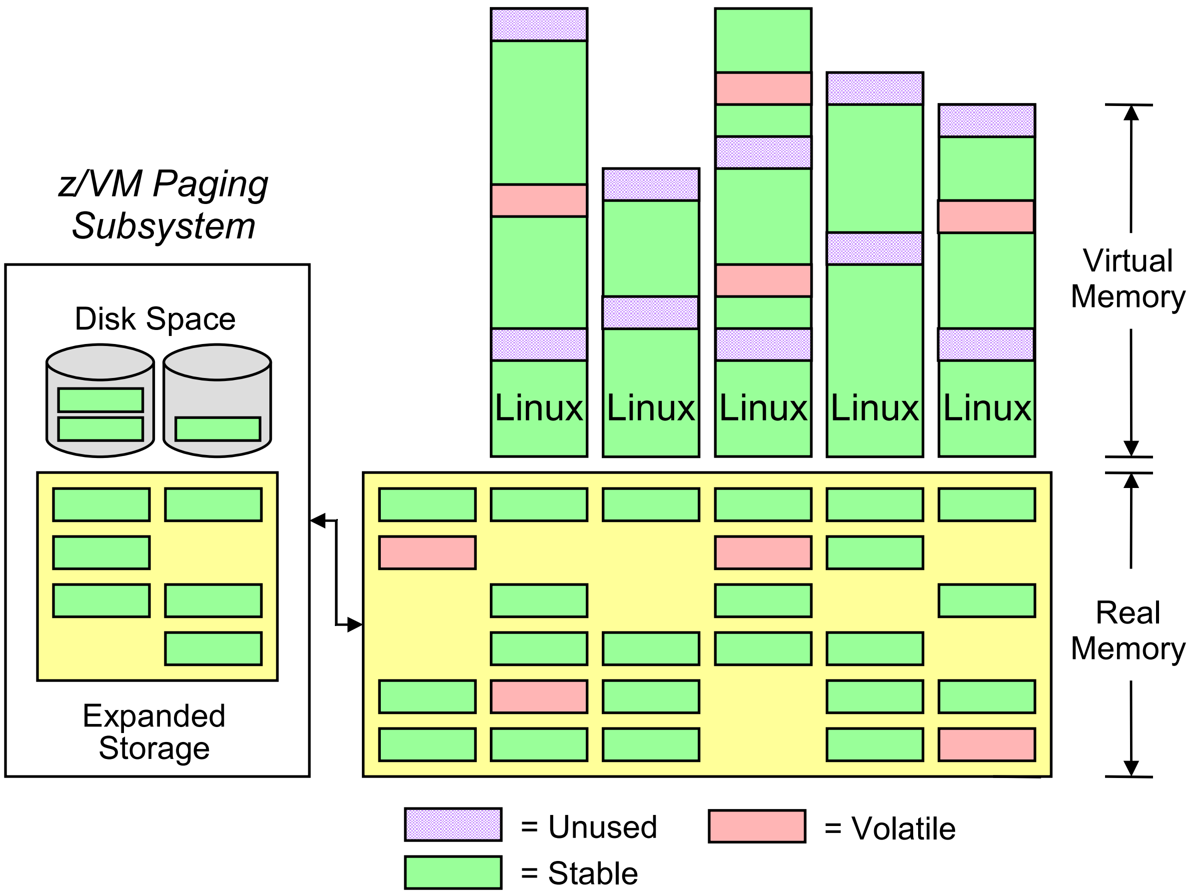
- Linux
  - If cmm is configured as a building component
    - Add a parameter to the kernel parameter line (in `/etc/zipl.conf`)
    - `cmm.sender=<guest name>`
    - <guest name> is the name of the z/VM guest thats allowed to send messages to the CMM module.
    - The default guest name is VMRMSVM. This is the default name of the VMRM Service Virtual Machine (SVM)
    - Example: `cmm.sender=T6360033`
  - If cmm is available as a loadable kernel module
    - `modprobe cmm sender=T6360034`

Collaborative Memory Management Assist (CMMA) aka CMM2

- While CMM1 formerly known as VMRM-CMM is a software only solution which only requires a resource manager like VMRM and a Linux guest with a loaded CMM driver, CMMA is build in hardware
- CMMA adds new hardware functions to the z9
- The hardware support known as Host Page Management Assist (HPMA) is able to let Linux guests and z/VM modify and keep track of the using state of each 4K page of the Linux Kernel
- CP is now able to determine when an application inside the Linux guest releases memory and can select those for removal without paging-out to the expanded storage
- Using this information the guest and the hypervisor try to optimize their memory usage
- Requirements: z/VM 5.2 with APAR VM63856 and z9 BC/EC or newer

Collaborative Memory Management Assist (CMMA) aka CMM2

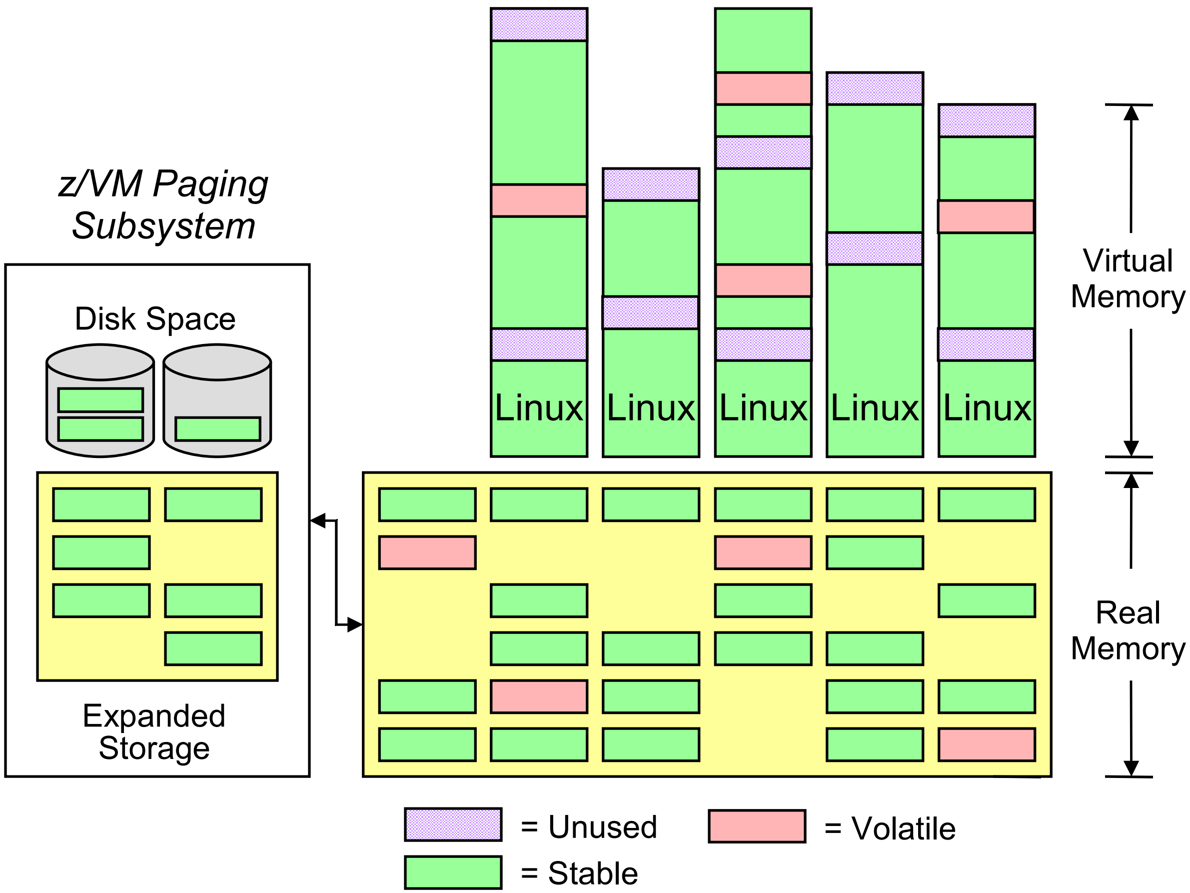
- Extends coordination of memory and paging between Linux and z/VM to the level of individual pages using a new hardware assist (CMMA)
- z/VM knows when a Linux application has released a page of memory
- Host Page-Management Assist (HPMA), in conjunction with CMMA, further reduces z/VM processing needed to resolve page faults
- Can help z/VM host more virtual servers in the same amount of memory
- Supported by System z9 and z/VM V5.3



Simplified overview

Collaborative Memory Management Assist (CMMA)

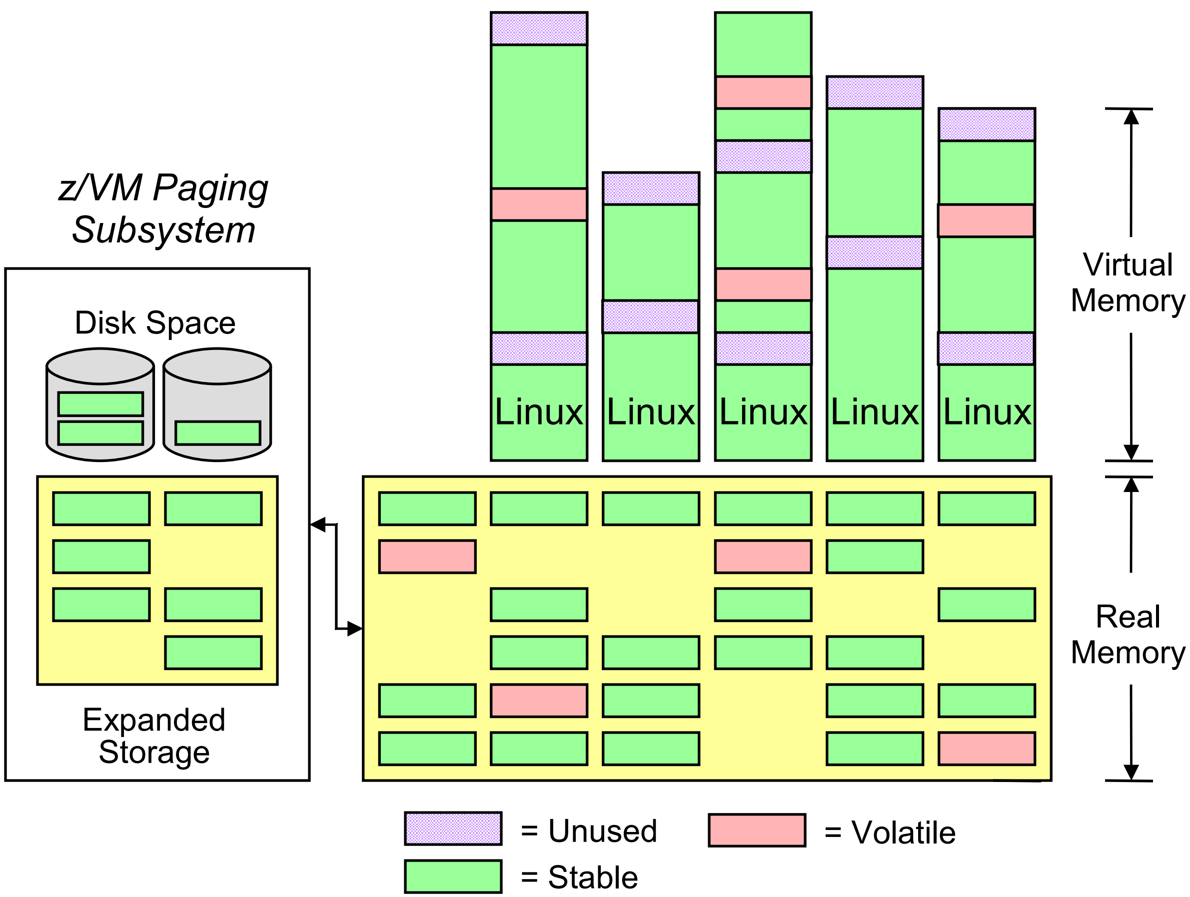
- **stable**: Linux requires the content of the memory
- **unused**: Linux no longer requires the information stored in these pages
- **volatile**: Linux can recover the content stored in these pages, but it is helpful, to be able to access the content



Simplified overview

Collaborative Memory Management Assist (CMMA)

- If z/VM requires a page, it can use unused pages.
  - This prevents z/VM from unnecessary paging
- I case no unused pages are available, z/VM will delete the content of volatile pages and redistribute them
  - This prevents z/VM from unnecessary paging
- When Linux accesses an old page, z/VM sends a fault and Linux will reconstruct the content (which is read from storage)
  - Benefit: Read once inside the guest and don't perform „page out“ + „page in“ operations in z/VM



- stable pages can the paged normally by z/VM (same behavior as in a non-cmm env.)

CMM2 activation

- CMM2
  - z/VM: **`CP SET MEMASSIST ON ALL`**
  - Linux: Set kernel parameter **`cmma=on`** in **`/etc/zipl.conf`**

CPU Hotplug

- During system runtime cpus of a Linux guest and be activated and deactivated via entries in the `/sys` filesystem
- Deactivate

```
[root@T63~]# echo 0 > /sys/devices/system/cpu/cpu8/online
```

- Activate

```
[root@T63~]# echo 1 > /sys/devices/system/cpu/cpu8/online
```

Activating standby CPUs and deactivating operating CPUs

- Activating standby CPUs and deactivating operating CPUs (2.6.25)
  - A CPU in an LPAR can be in a configured, standby, or reserved state.
  - Under Linux, on IPL only CPUs that are in a configured state are brought online and used.
  - The kernel operates only with configured CPUs.
  - To configure or deconfigure a CPU its physical address needs to be known.

Activating standby CPUs and deactivating operating CPUs (cont)

- Only present CPUs have a sysfs entry. If you add a CPU to the system the kernel will not automatically detect it.
- To force the detection of the CPU use the rescan attribute.

```
[root@T63~]# echo 1 > /sys/devices/system/cpu/rescan
```

- If new CPUs are found new sysfs entries will be created and they will be in the configured or standby state depending on how the hypervisor added them.

```
[root@T63~]# echo 1 >
/sys/devices/system/cpu/cpu2/configure
```

- Bring the CPU online by writing "1" to its online attribute:

```
[root@T63~]# echo 0 > /sys/devices/system/cpu/cpu2/online
```

- To deactivate an operating CPU bring the CPU offline by writing "0" to its online attribute and change the state of the CPU to standby by writing "0" to its configure attribute

The Linux on System z CPU and Memory Hotplug Daemon (cpuplugd)

* Changes the number of used processors on the fly, depending on the current overall utilization and load
* Is available with SLES10 SP2
* Expectation:
  - **Increases the performance of single threaded applications within a z/VM or LPAR environment with multiple CPUs**
* Enables or disables CPUs based on a set of rules
* Is enabled in the kernel configuration by setting

```
Base setup --->
--- Processor type and features ---
64 bit kernel (CONFIG_64BIT)
Symmetric multi-processing support (CONFIG_SMP)
[] Support for hot-pluggable CPUs (CONFIG_HOTPLUG_CPU)
```

Exemplary /etc/sysconfig/cpulugd configuration file

```
CPU_MIN="2"
CPU_MAX="0"

UPDATE="10"

CMM_MIN="0"
CMM_MAX="8192"
CMM_INC="256"

HOTPLUG="(loadavg > onumcpus + 0.75) & (idle < 10.0)"
HOTUNPLUG="(loadavg < onumcpus - 0.25) | (idle > 50)"

# Memplug and memunplug can contain the following keywords:
#  - apcr:          the amount of page cache reads
#  - freemem:       the amount of free memory (in megabyte)
#  - swaprate:      the number of swapin and swapout operations
#
# Per default this function is disabled, because this rule has to be
# adjusted by the admin for each production system, depending on the
# environment

MEMPLUG="0"
MEMUNPLUG="0"
```

cpuplugd parameters

- The control information is stored at **/etc/sysconfig/cpuplugd**
- Minimum number of CPUs is set with **cpu_min="<number>"**
- Maximum number of CPUs is set with **cpu_max="<number>"**
- The update interval is set with **update="<value in seconds>"**
- Consider the effect of kernel "cpu" parameters:
  - **maxcpus=<n>** sets the number of processors which will be active after system boot.
  - **possible_cpus=<n>** is the upper limit for hotpluggable CPUs.
  - If **possible_cpus** is not specified but **maxcpus**, then **maxcpus** is the upper limit for hotpluggable CPUs

cpuplugd rules

- The default rule for increasing the number of CPUs is

```
HOTPLUG="(loadavg > onumcpus + 0.75) & (idle < 10.0)"
```

  - An additional CPU is enabled, if the loadaverage is greater than the the number of active (online) CPUs plus 0.75 and the current idle percentage is below 10 percent.

- The default rule for decreasing the number of CPUs is

```
HOTUNPLUG="(loadavg < onumcpus - 0.25) | (idle > 50)"
```

  - A CPU is disabled, either if the current load is below the number of active CPUs minus 0.25 or if the idle percentage is above 50%.

- The formulas for these rules can be modified. See “Device Drivers, Features and Commands” for valid expressions.

- Note:
  - `loadavg` is a value that changes slowly
  - `idle` changes fast
  - Increments and decrements of active CPUs are done in steps of 1 every time when the rules are checked.

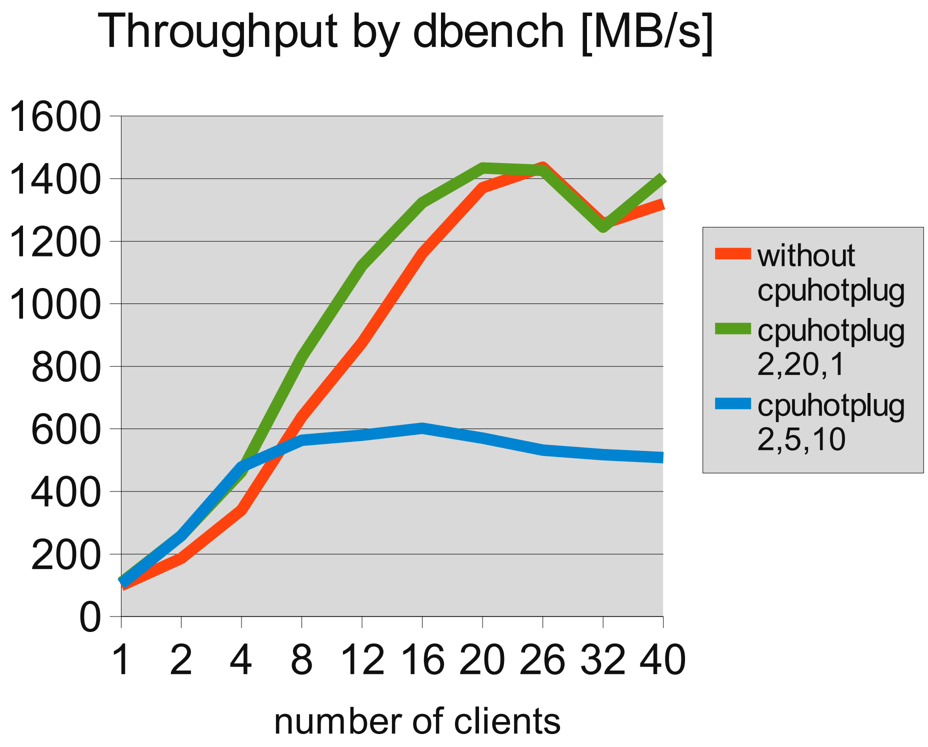
Performance

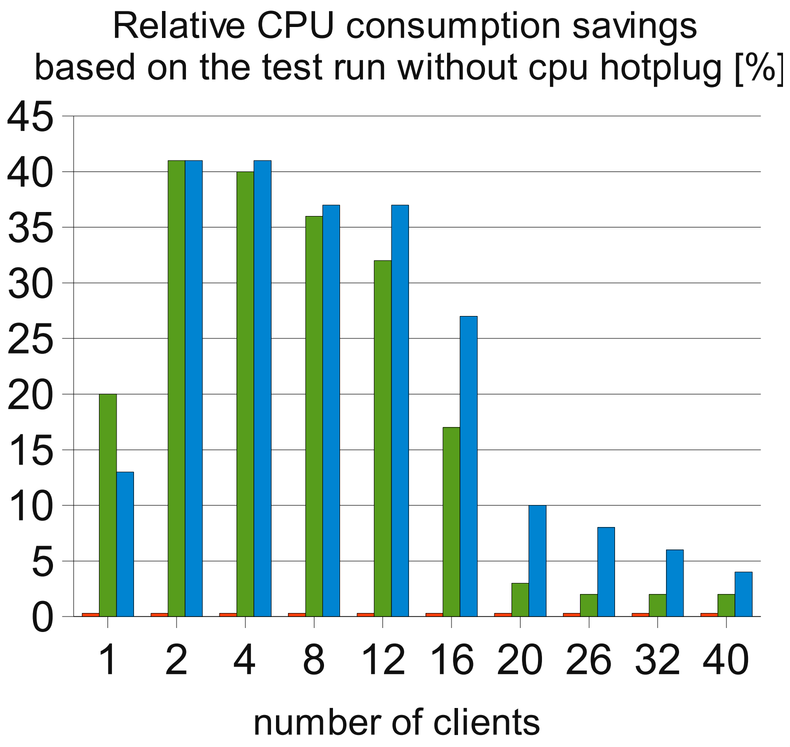
Cpuplugd test workload

- dbench 3
  - Emulation of Netbench benchmark, rates windows file servers
  - Mainly memory operations
  - Mixed file operations workload for each process: create,write,read,append, delete
  - Scaling with 1,2,4,8,16 CPUs and 1,4,8,12,16,20,26,32 and 40 clients
  - 2 GB memory
- Modification to the standard code:
  - Purpose: Need more interaction between clients
  - Create two processes per client and communicate with POSIX message queues.
  - First process:
    - Read the I/O commands from the control file
    - Pass this information to the second process
  - Second process:
    - Performs the execution of this command
    - Reports the end of the operation back to the first process

cpuplugd performance results

- Improvements in case where the default (high) number of CPUs is not needed
- Up to 40% more throughput, up to 40% CPU cost savings





cpuplugd summary

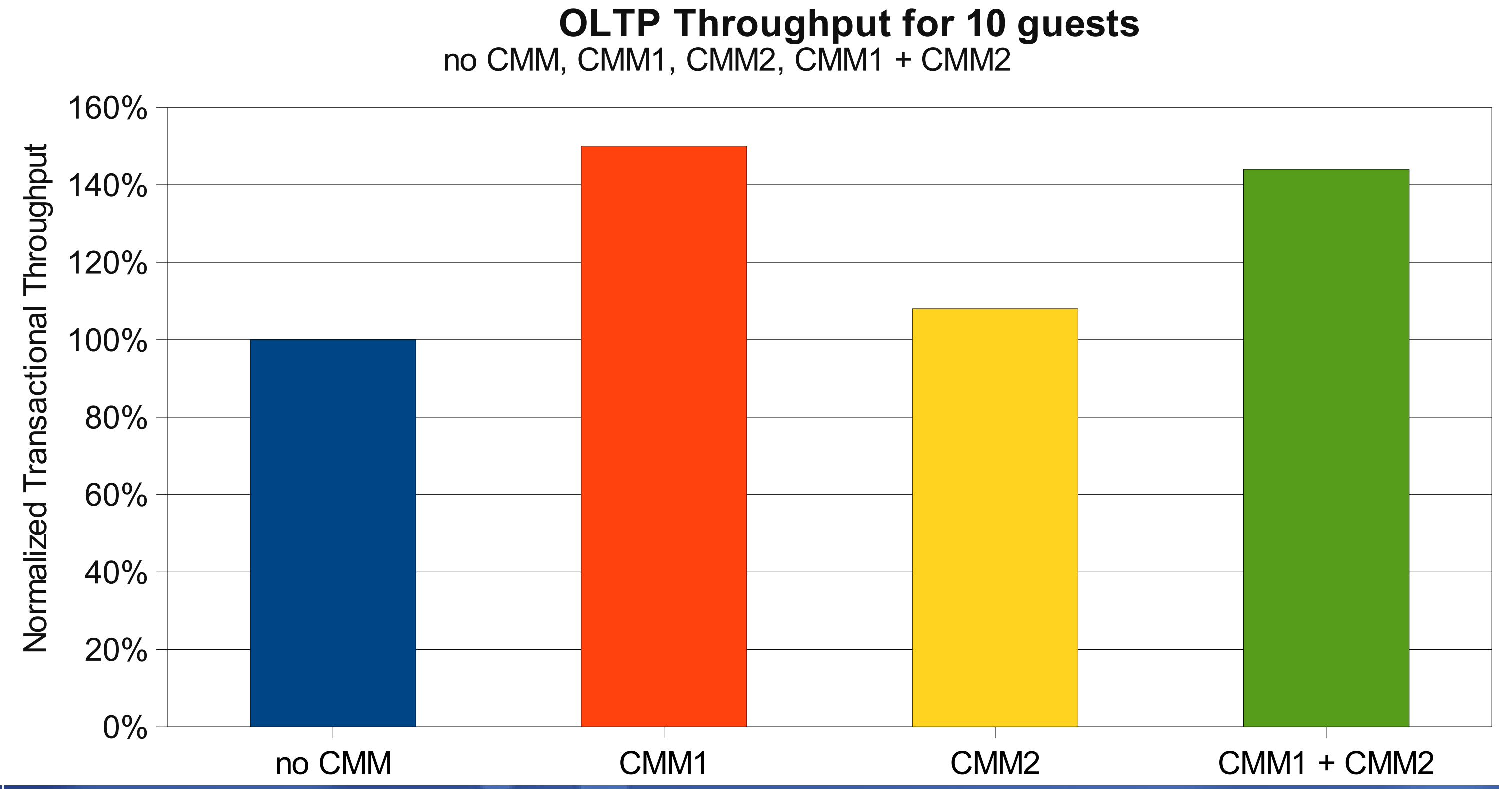
- This feature improves the performance by
  - Sizing the correct amount of processors for a Linux system depending on its current load.
  - Avoiding the Linux scheduler queue balancing in partial load situations
- Set the minimum and maximum number of CPUs to values which apply to the real workload:
  - Setting `cpu_min` to 2 may be too high
  - `cpu_max` should be set so that it really covers the peaks
- Linux guests under z/VM: use z/VM 5.4
  - Guarantees that stopped processors are no longer included in virtual processor prioritization calculations
  - Ensures share redistribution

CMM large application scenario - test setup

- Requirements
  - A software product should be tested which is frequently used by customers
  - The application should require and use large quantities of memory
  - The memory should be overcommitted
- Test environment
  - LPAR with z/VM, 10 CPUs, 80 GB central, 4 GB expanded storage
  - 10 SLES10 SP1 Linux guests, each with 3 virtual CPUs and 16 GB memory
  - All Linux guests require 3x of the available CPU resources and 2x of the available z/VM guest storage.
  - OLTP database workload was chosen as Linux application
  - Tests with CMM1, CMM2 and a combination of both
- Expectation
  - Both features should improve the overall system performance and increase the overall throughput

CMM real life scenario - measurement results

- 50% throughput improvement with CMM1
- No big improvement, when using only CMM2



CMM real life scenario - conclusions

- Why did CMM1 help to improve the performance?
  - Initially the memory is overcommitted by 2x
  - Then VMRM instructs the Linux guests to shrink their page cache, which contains the database buffer pool and the file system cache
  - The reduction is done in the file system cache, keeping the full amount of database buffer pool
  - Each Linux guest reduced its memory to approx. 8.5 GB, overcommitment drops.
  - Disk I/O is not suffering here significantly from a smaller file cache
  - The transaction throughput is not impacted heavily

CMM real life scenario – conclusions (cont.)

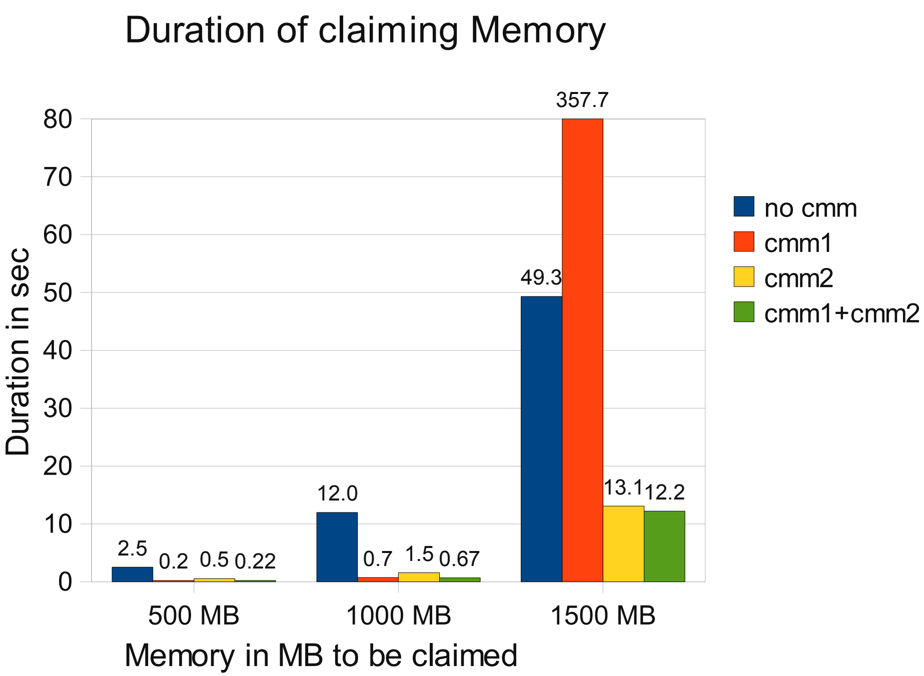
- Why did CMM2 not help to improve the performance?
  - The memory is always overcommitted by 2x
  - The Linux guests always want their allocated memory, because the database workload occupies all available file cache with its disk I/O.
  - Each Linux guest continues to occupy 16 GB memory
  - With CMM2 Linux sets the page status and z/VM can select the non-dirty pages of the page cache to reuse for another guest
  - Each Linux guest claims memory as soon as he is scheduled

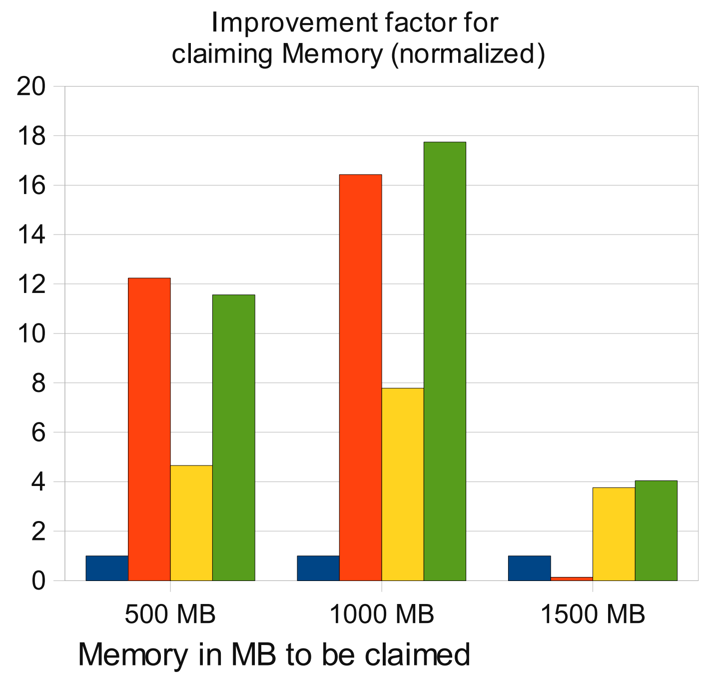
Special CMM2 scenario - setup

- Idea
  - In the real life scenario all Linux guests were permanently busy
  - Memory requirements were constantly on the same level
  - A situation, where a guest suddenly claims a big number of pages was not yet tested
- Test environment
  - 15 guests, touching all their memory, all z/VM storage used.
  - A guest now claims 500MB, 1000MB or 1.5GB of memory
  - We measure the duration of these operations
- Expectation
  - Both features should perform this exercise faster than a setup where no CMM is used

Special CMM2 scenario – measurement results

- In case of sudden memory claims CMM2 is the better choice
- CMM1 is doing well if the amount of requested pages is not too high





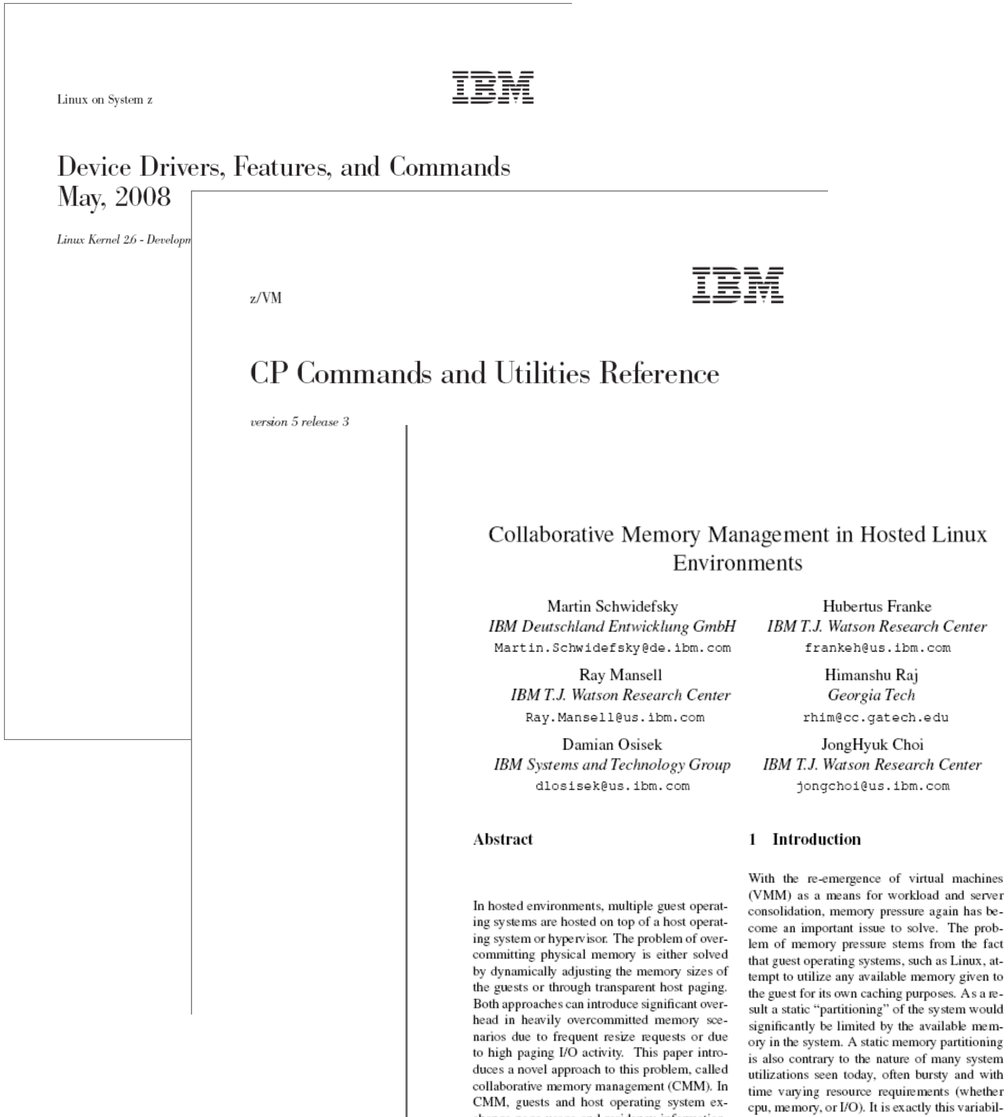
Special CMM2 scenario - conclusion

- Why is CMM2 good in all test cases?
  - z/VM can see which pages are clean
  - There is a big amount of non-dirty pages, since we did not modify them
  - z/VM simply takes enough of these pages and assigns them to the requesting Linux guest
- Why is CMM1 good in the small and medium request test case and bad in the large request test case?
  - Before pages can be given to the requesting Linux guest we first have to process the shrink step to provide enough pages
  - Shrinking is done in Linux in increments of 1 MB
  - In the small and medium request test case there was still enough memory left over that the shrink operation could complete in a short time
  - In the large request test case the Linux guests were busy to find free pages and the shrinking took extremely long
  - Finally the Linux guests started swapping

CMM overall conclusion

- CMM1
  - In cases of memory overcommtiment CMM1 shrinks the Linux guests
  - If the initial memory definition and allocation for the Linux guest was roughly sized too high CMM1 will correct this
  - The memory reduction in the Linux guests avoids frequent claims when each guest is dispatched
  - The effectiveness depends on the amount of page cache that can be removed from the Linux guests without impacting the guest performance too much
- CMM2
  - Linux guests provide z/VM the page states do that z/VM can “steal” guest pages which can easily be recreated in situations of memory claims
  - The effectiveness depends on the amount of unused or non-dirty pages with backing that can be identified
- CMM1 + CMM2
  - Is the best choice to be prepared for all situations

More Information



Linux on System z

Device Drivers, Features, and Commands
May, 2008

Linux Kernel 2.6 - Developm

z/VM

CP Commands and Utilities Reference

version 5 release 3

Collaborative Memory Management in Hosted Linux Environments

Martin Schwidefsky
IBM Deutschland Entwicklung GmbH
Martin.Schwidefsky@de.ibm.com

Hubertus Franke
IBM T.J. Watson Research Center
frankeh@us.ibm.com

Ray Mansell
IBM T.J. Watson Research Center
Ray.Mansell@us.ibm.com

Himanshu Raj
Georgia Tech
rhim@cc.gatech.edu

Damian Osisek
IBM Systems and Technology Group
dlosisek@us.ibm.com

JongHyuk Choi
IBM T.J. Watson Research Center
jongchoi@us.ibm.com

Abstract

In hosted environments, multiple guest operating systems are hosted on top of a host operating system or hypervisor. The problem of overcommitting physical memory is either solved by dynamically adjusting the memory sizes of the guests or through transparent host paging. Both approaches can introduce significant overhead in heavily overcommitted memory scenarios due to frequent resize requests or due to high paging I/O activity. This paper introduces a novel approach to this problem, called collaborative memory management (CMM). In CMM, guests and host operating system exchange page usage and residency information.

1 Introduction

With the re-emergence of virtual machines (VMM) as a means for workload and server consolidation, memory pressure again has become an important issue to solve. The problem of memory pressure stems from the fact that guest operating systems, such as Linux, attempt to utilize any available memory given to the guest for its own caching purposes. As a result a static "partitioning" of the system would significantly be limited by the available memory in the system. A static memory partitioning is also contrary to the nature of many system utilizations seen today, often bursty and with time varying resource requirements (whether cpu, memory, or I/O). It is exactly this variability that virtualization tends to exploit

Thank you for your interest !

Questions?

Hans-Joachim Picht

Linux Technology Center

Linux on System z Kernel Development & Red Hat Liaision

IBM Deutschland Research & Development GmbH
Schönaicher Strasse 220
71032 Böblingen, Germany

Phone +49 (0)7031-16-1810
Mobile +49 (0)175 - 1629201
hans@linux.vnet.ibm.com

Disclaimer / Trademarks

The following are trademarks of the International Business Machines Corporation in the United States, other countries, or both.

Not all common law marks used by IBM are listed on this page. Failure of a mark to appear does not mean that IBM does not use the mark nor does it mean that the product is not actively marketed or is not significant within its relevant market.

Those trademarks followed by ® are registered trademarks of IBM in the United States; all others are trademarks or common law marks of IBM in the United States.

For a complete list of IBM Trademarks, see www.ibm.com/legal/copytrade.shtml:

*, AS/400®, e business(logo)®, DBE, ESCO, eServer, FICON, IBM®, IBM (logo)®, iSeries®, MVS, OS/390®, pSeries®, RS/6000®, S/30, VM/ESA®, VSE/ESA, WebSphere®, xSeries®, z/OS®, zSeries®, z/VM®, System i, System i5, System p, System p5, System x, System z, System z9®, BladeCenter®

The following are trademarks or registered trademarks of other companies.

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.
Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license therefrom.
Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.
Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.
Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.
UNIX is a registered trademark of The Open Group in the United States and other countries.
Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.
ITIL is a registered trademark, and a registered community trademark of the Office of Government Commerce, and is registered in the U.S. Patent and Trademark Office.
IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency, which is now part of the Office of Government Commerce.

* 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.