

1597355 - Swap-space recommendation for Linux

Version	3	Type	SAP Note
Language	English	Master Language	English
Priority	Recommendations / Additional Info	Category	Performance
Release Status	Released for Customer	Released On	01.07.2011
Component	BC-OP-LNX (Linux)		

Please find the original document at <https://launchpad.support.sap.com/#/notes/1597355>

Symptom

You want to setup a SAP system on Linux and evaluate the amount of swap-space that is sufficient for your system.

Other Terms

Linux, Swap, swap space, swap-space, swap-space recommendation, paging, swapping

Reason and Prerequisites

Linux provides a special paging functionality to evict and reclaim memory pages of applications from the physical memory (RAM) to a dedicated area on a secondary storage (i. e. a disk). This functionality is called swap and the secondary storage area is referred to as swap-space. The Linux swap functionality can improve the overall system performance and system reliability. Not frequently used memory can be temporally moved to the swap-space and frees up physical memory that can be used for other purposes, i. e. page-cache. Furthermore it can prevent an out-of-memory situation, when a system runs out of physical memory. In this case Linux starts evicting memory-pages from the physical memory to the swap-space until it can serve all memory allocation requests from applications. The Linux swap-functionality is available on all Linux systems supported by SAP.

The swap functionality requires enough space on a secondary storage, typically a partition on a disk or a file on a filesystem.

Solution

SAP recommends to use an amount of swap-space based on the table illustrated below. Customers may decide to use more or less swap-space based on their individual system configuration and their own experience during the day-to-day usage of a system.

Physical Memory (RAM) Recommended Swap-Space

< 32 GByte 2 x RAM

32 - 63 GByte 64 GByte
 64 - 127 GByte 96 GByte
 128 - 255 GByte 128 GByte
 256 - 511 GByte 160 GByte
 512 - 1023 GByte 192 GByte
 1024 - 2047 GByte 224 GByte
 2048 - 4095 GByte 256 GByte
 4096 - 8191 GByte 288 GByte

> 8192 GByte 320 GByte

When running SAP systems, the use of Linux swap-space is mandatory. SAP recommends to setup a dedicated swap-partition on a fast storage, like a system internal HDD, RAID-Array, SSD or a LUN in a SAN.

In the past, SAP recommended to use a swap-space with a size of two times the amount of physical memory (RAM). For systems with up to 32 GByte RAM, this recommendation remains valid. For systems with more than 32 GByte RAM, the amount of swap-space in proportion to the physical memory can be reduced. This non-linear swap-space recommendation has several reasons:

Systems running SAP with more than 32 GByte RAM (64 GByte swap-space) are usually not able to efficiently use the high amount of swap-space without running into a performance degradation. The reason is, that a swap-space area on a disk is usually much slower than the physical RAM. Upon a certain swap-space usage, the swap-in and swap-out I/O activity may seriously slowdown or even completely freeze a system. A high swap-usage and increased disk I/O activity indicates that a system is already in an unhealthy state. This is slightly depending on the SAP workload (SAP system and database). Appropriate steps by the System-Administration have to be initiated in any case in order to secure the operation of the system.

Swap-space areas are typically stored on partitions residing on local disks. Huge amount of disk-space for the exclusive swap-space usage can be quite expensive on systems with large amounts of memory. Many blade servers do not have enough slots to provide sufficient disk-space capacity. Furthermore, modern servers are often equipped with smaller Solid-State Drives, that doesn't provide enough disk capacity for large swap-space partitions.

Overview of the Linux swap mechanism (2.6.x based kernel versions)

During normal operation a Linux system tries to keep all application data in physical memory. Linux uses swap-space to store inactive memory pages, evicted from the systems physical memory, in order to free up physical memory capacity. The operation of evicting memory pages from the physical memory to the swap-space is called "swap-out". The operation of restoring pages from the swap-space into the physical memory is called "swap-in". The swap-space is a memory area on an external device, typically a partition on a harddisk or a file on a filesystem. It is usually much slower than the physical RAM. Linux can use several swap-space locations in parallel and merges all swap-space locations together to one large swap-space area. Memory pages that are swapped-out into the swap-space, free up physical memory that can be re-used for other purposes. Swap-space is not a full-qualified extension to the physical RAM, because it is not possible to read or modify swapped-out memory pages directly. Instead, swapped-out memory pages have to be read back into the physical memory when an application tries to access these memory pages again.

Linux swap-space extends the physical memory in the way, that it is used to store **inactive** memory pages, which are evicted from the physical memory in favor of **active** memory pages or even memory pages from the page-cache. According to that, Linux may prefer a bigger page-cache over least used memory pages of an application. If a system runs out of physical memory, Linux evicts as many inactive memory pages as needed from the physical memory into the swap-space in order to be able to fulfill all memory allocation requests from applications. If a system runs out of physical memory and out of swap-space, the OOM killer (out of memory killer) starts to kill processes based on a selection-algorithm in order to free up memory and to keep a system alive.

Reasons for swapping out physical memory to the swap-space

Linux starts to evict memory pages from the physical memory to the swap-space under certain conditions:

- **Memory pressure:** If Linux runs out of physical memory, e. g. if an applications allocates more memory than it is physically available, Linux starts to swap-out memory pages to the swap-space to free-up physical memory. Before that happens, it will try to shrink its caches, like the page-cache, buffer-cache and slab-cache.
- **Large amount of inactive memory and a growing page-cache:** If the RAM contains a lot of inactive memory pages from applications (for example SAP shared memory areas) and a growing page-cache fills up the remaining memory, Linux may decide to page-out some or all of the inactive memory pages into the swap-space. This behavior improves the overall-performance in many cases, but may also lead into performance problems under some circumstances. More details about this behavior are explained in the SAP Note #1557506

Description of the Linux swap mechanism

The Linux kernel 2.6 uses the kswapd daemon thread for swap operations. Additionally a direct-memory reclaim path exists, bypassing the kswapd, when a system gets under heavy memory pressure that can not be solved by the kswapd anymore. Kswapd only evicts pages from the physical memory that are marked as inactive. The kernel scans memory areas in all memory zones and decides, based on a LRU (least recently used) and a balance algorithm, whether a scanned memory page gets active or inactive. The more memory pressure exists, the higher is the probability for a memory page to get marked as inactive. Not all inactive memory pages are swapped-out immediately. Another algorithm decides how much of the inactive memory is required to be swapped-out to the swap-space. The kswapd is invoked by the Linux memory allocator if the available free memory of a zone gets below a certain value.

In opposite of swap-out, the swap-in mechanism restores a page from the swap-space into the physical memory. This is being done, if a process tries to access a memory page in the virtual memory address space, that is currently not resident in the physical RAM, a so called page-fault. After a page has been read back into the physical memory, it is not being deleted immediately from the swap space. Instead, the page-table information of this page is stored in the **Swap-Cache**. If this recently restored memory page has to be swapped-out again and has not been modified in the meanwhile, it is still identical with the already existing copy in the swap-space. In this case, the systems decides, based on the information stored in the swap-cache, that this page can just be removed from the physical memory without having the need to do an expensive disk write operation.

The difference between paging and swapping

In Linux, the operation that moves a memory page from the physical memory (RAM) to a secondary device (e. g. a disk) is called page-out. The operation of reading a memory-page from a secondary device into the physical memory is called page-in. This does not only include the eviction of memory pages from the physical memory to the swap-space and reading it back into the physical memory. For example, it also includes the synchronization of dirty memory pages stored in the page-cache with their counterparts on the disk. Swap-space read and write operations are referred to as swap-in and swap-out.

Other Components

Component	Description
BC-OP-LNX-SUSE	SUSE Linux
BC-OP-LNX-RH	Red Hat Linux
BC-OP-LNX-OLNX	Oracle Linux

This document refers to

SAP Note/KBA	Title
171356	SAP Software on Linux: General information
1647204	BYD: Database is stopped due to insufficient memory

This document is referenced by

SAP Note/KBA	Title
2944508	All server memory being consumed by Linux buffers and cache
2489136	DB6 error ADM0503C and Linux OOM (Out-Of-Memory)
2369910	SAP Software on Linux: General information
1944799	SAP HANA Guidelines for SLES Operating System Installation
1647204	BYD: Database is stopped due to insufficient memory

[Terms of use](#) | [Copyright](#) | [Trademark](#) | [Legal Disclosure](#) | [Privacy](#)