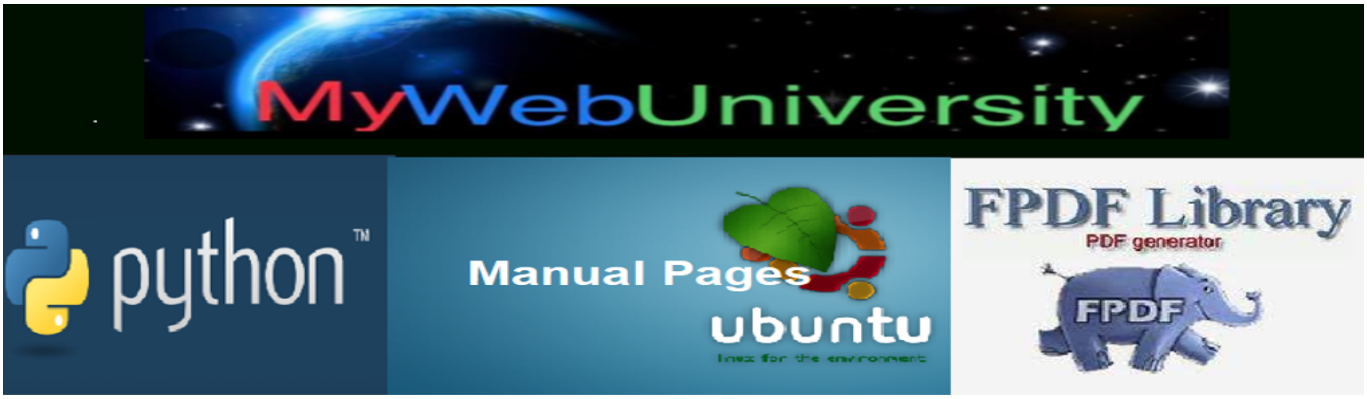

Rocky Enterprise Linux 9.2 Manual Pages on command 'numa.7'

$ man numa.7

NUMA(7) Linux Programmer's Manual NUMA(7)

NAME

numa - overview of Non-Uniform Memory Architecture

DESCRIPTION

Non-Uniform Memory Access (NUMA) refers to multiprocessor systems whose memory is divided into multiple memory nodes. The access time of a memory node depends on the relative lo? cations of the accessing CPU and the accessed node. (This contrasts with a symmetric mul? tiprocessor system, where the access time for all of the memory is the same for all CPUs.) Normally, each CPU on a NUMA system has a local memory node whose contents can be accessed faster than the memory in the node local to another CPU or the memory on a bus shared by all CPUs.

NUMA system calls

The Linux kernel implements the following NUMA-related system calls: get_mempolicy(2), mbind(2), migrate_pages(2), move_pages(2), and set_mempolicy(2). However, applications should normally use the interface provided by libnuma; see "Library Support" below.

/proc/[number]/numa_maps (since Linux 2.6.14)

This file displays information about a process's NUMA memory policy and allocation.

Each line contains information about a memory range used by the process, displaying?among other information?the effective memory policy for that memory range and on which nodes the pages have been allocated.

numa_maps is a read-only file. When /proc/<pid>/numa_maps is read, the kernel will scan the virtual address space of the process and report how memory is used. One line is dis? played for each unique memory range of the process.

The first field of each line shows the starting address of the memory range. This field allows a correlation with the contents of the /proc/<pid>/maps file, which contains the end address of the range and other information, such as the access permissions and shar? ing.

The second field shows the memory policy currently in effect for the memory range. Note that the effective policy is not necessarily the policy installed by the process for that memory range. Specifically, if the process installed a "default" policy for that range, the effective policy for that range will be the process policy, which may or may not be "default".

The rest of the line contains information about the pages allocated in the memory range, as follows:

N<node>=<nr_pages>

The number of pages allocated on <node>. <nr_pages> includes only pages currently mapped by the process. Page migration and memory reclaim may have temporarily un? mapped pages associated with this memory range. These pages may show up again only after the process has attempted to reference them. If the memory range represents a shared memory area or file mapping, other processes may currently have additional pages mapped in a corresponding memory range.

file=<filename>

The file backing the memory range. If the file is mapped as private, write ac? cesses may have generated COW (Copy-On-Write) pages in this memory range. These pages are displayed as anonymous pages.

heap Memory range is used for the heap.

stack Memory range is used for the stack.

huge Huge memory range. The page counts shown are huge pages and not regular sized pages.

anon=<pages>

The number of anonymous page in the range.

dirty=<pages>

Number of dirty pages.

mapped=<pages>

Total number of mapped pages, if different from dirty and anon pages.

mapmax=<count>

Maximum mapcount (number of processes mapping a single page) encountered during the scan. This may be used as an indicator of the degree of sharing occurring in a given memory range.

swapcache=<count>

Number of pages that have an associated entry on a swap device.

active=<pages>

The number of pages on the active list. This field is shown only if different from the number of pages in this range. This means that some inactive pages exist in the memory range that may be removed from memory by the swapper soon.

writeback=<pages>

Number of pages that are currently being written out to disk.

CONFORMING TO

No standards govern NUMA interfaces.

NOTES

The Linux NUMA system calls and /proc interface are available only if the kernel was con? figured and built with the CONFIG_NUMA option.

Library support

Link with -lnuma to get the system call definitions. libnuma and the required <numaif.h> header are available in the numactl package.

However, applications should not use these system calls directly. Instead, the higher level interface provided by the numa(3) functions in the numactl package is recommended.

The numactl package is available at ?ftp://oss.sgi.com/www/projects/libnuma/download/?. The package is also included in some Linux distributions. Some distributions include the development library and header in the separate numactl-devel package.

SEE ALSO

get_mempolicy(2), mbind(2), move_pages(2), set_mempolicy(2), numa(3), cpuset(7), nu? mactl(8)

COLOPHON

This page is part of release 5.10 of the Linux man-pages project. A description of the project, information about reporting bugs, and the latest version of this page, can be found at https://www.kernel.org/doc/man-pages/.