NAME

NUMA - Non-Uniform Memory Access

SYNOPSIS

options MAXMEMDOM
options NUMA

DESCRIPTION

Non-Uniform Memory Access is a computer architecture design which involves unequal costs between processors, memory and IO devices in a given system.

In a **NUMA** architecture, the latency to access specific memory or IO devices depends upon which processor the memory or device is attached to. Accessing memory local to a processor is faster than accessing memory that is connected to one of the other processors. FreeBSD implements NUMA-aware memory allocation policies. By default it attempts to ensure that allocations are balanced across each domain. Users may override the default domain selection policy using cpuset(1).

NUMA support is enabled when the **NUMA** option is specified in the kernel configuration file. Each platform defines the **MAXMEMDOM** constant, which specifies the maximum number of supported NUMA domains. This constant may be specified in the kernel configuration file. **NUMA** support can be disabled at boot time by setting the *vm.numa.disabled* tunable to 1. Other values for this tunable are currently ignored.

Thread and process **NUMA** policies are controlled with the cpuset_getdomain(2) and cpuset_setdomain(2) syscalls. The cpuset(1) tool is available for starting processes with a non-default policy, or to change the policy of an existing thread or process. See SMP(4) for information about CPU to domain mapping.

Systems with non-uniform access to I/O devices may mark those devices with the local VM domain identifier. Drivers can find out their local domain information by calling bus_get_domain(9).

MIB Variables

The operation of **NUMA** is controlled and exposes information with these sysctl(8) MIB variables:

vm.ndomains
The number of VM domains which have been detected.

vm.phys_locality
A table indicating the relative cost of each VM domain to each other. A value of 10 indicates equal cost. A value of -1 means the locality map is not available or no locality information is

available.

vm.phys_segs
The map of physical memory, grouped by VM domain.

IMPLEMENTATION NOTES

The current **NUMA** implementation is VM-focused. The hardware **NUMA** domains are mapped into a contiguous, non-sparse VM domain space, starting from 0. Thus, VM domain information (for example, the domain identifier) is not necessarily the same as is found in the hardware specific information. Policy information is available in both struct thread and struct proc.

SEE ALSO

cpuset(1), cpuset_getaffinity(2), cpuset_setaffinity(2), SMP(4), bus_get_domain(9)

HISTORY

NUMA first appeared in FreeBSD 9.0 as a first-touch allocation policy with a fail-over to round-robin allocation and was not configurable. It was then modified in FreeBSD 10.0 to implement a round-robin allocation policy and was also not configurable.

The numa_getaffinity(2) and numa_setaffinity(2) syscalls and the numactl(1) tool first appeared in FreeBSD 11.0 and were removed in FreeBSD 12.0. The current implementation appeared in FreeBSD 12.0.

AUTHORS

This manual page written by Adrian Chadd <*adrian@FreeBSD.org*>.

NOTES

No statistics are kept to indicate how often **NUMA** allocation policies succeed or fail.