

Non-Uniform Memory Access

Christian Külker

2024-03-07

Contents

1	Installing	2
2	Installing NUMA Helper Tools For Debian	3
3	lscpu	3
4	Compiling And Installing numactl From Source	4
5	Numactl Commands	5
5.1	migratepages	5
5.2	numastat	5
5.3	numactl	5
5.4	memhog	6
5.5	numademo	6
6	Numatop	7
7	Further Reading	7
8	History	7
9	Disclaimer of Warranty	7
10	Limitation of Liability	8

Non-uniform memory access (NUMA) is a design specification of some modern multiprocessor architectures that, unlike **uniform memory access (UMA)**, does not allow all CPUs to access all memory equally. Usually this is due to the fact that each CPU has some memory attached to it. This does not necessarily mean that this memory cannot be accessed by processes from other CPUs, but accessing the memory has some performance penalty.

Because building a [NUMA](#) architecture is cheaper and still performs well if the programs are carefully designed, this design is quite popular in high performance computing ([HPC](#)).

The Linux 2.5 kernel started with basic NUMA support, which was improved in 3.8. Later, in 3.13, NUMA-related performance enhancements were added, such as memory pages shared between processes, huge pages, and `sysctl` support.

UMA

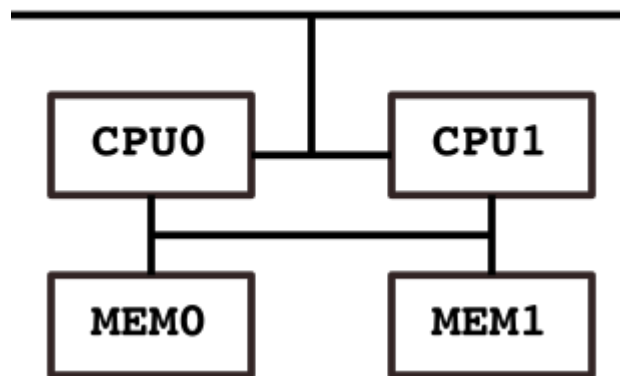


Figure 1: Uniform memory access

NUMA

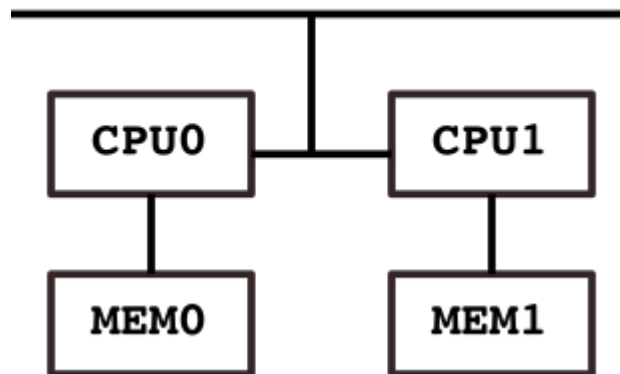


Figure 2: Non-uniform memory access

1 Installing

The content of the installation varies. It is recommended to either use a new distribution or even compile `numactl` from source, as this includes the `numademo` command.

2 Installing NUMA Helper Tools For Debian

Package	Stretch 9.12	Buster 10	Bullseye 11
numactl	2.0.11-2.1	2.0.12-1	2.12-1+b1
numad	0.5+20150602-5	0.5+20150602-5	0.5+20150602-7
numatop	1.0.4-3	2.1-2	2.1-4
util-linux	2.29.2-1+deb9u1	2.33.1-0.1	2.36.1-8+deb11u1

Tools and their package or source:

Command	Package	Source
lscpu	util-linux	util-linux
memhog	numactl	numactl
numastat	numactl	numactl
migspeed	numcatl	numactl
migratepages	numactl	numactl
numademo	n.a.	numactl
numatop	numatop	numatop
numad	numad	numad

- `numademo` is part of `numactl` but not packaged

3 lscpu

The tool `lscpu` can be used to understand the number of [NUMA](#) nodes.

```
# Laptop 2008
lscpu|grep NUMA
NUMA node(s):          1
NUMA node0 CPU(s):    0,1
# Desktop 2015
lscpu|grep NUMA
NUMA node(s):          1
NUMA node0 CPU(s):    0-7
```

```
# Raspberry Pi 4
lscpu|grep NUMA
NUMA node(s):                1
NUMA node0 CPU(s):          0-3
# Server 2022
lscpu|grep NUMA
NUMA node(s):                1
NUMA node0 CPU(s):          0-11
```

Standard hardware not used for HPC typically has only one NUMA node. Typical X86 NUMA hardware has 2 or more CPUs and 2 or more memory banks, one attached to each CPU.

It also can be useful to understand the architecture in general as such, because “CPU’s” are not equal.

Desktop:

```
lscpu | grep -E '^Thread|^Core|^Socket|^CPU\('|grep -v scaling
CPU(s):                4
Thread(s) per core:    2
Core(s) per socket:    2
Socket(s):              1
```

Raspberry Pi:

```
lscpu | grep -E '^Thread|^Core|^Socket|^CPU\('|grep -v scaling
CPU(s):                4
Thread(s) per core:    1
Core(s) per socket:    4
Socket(s):              1
```

4 Compiling And Installing numactl From Source

```
aptitude install autoconf automake
git clone https://github.com/numactl/numactl.git
cd numctl
./autogen.sh
./configure
make
make install
```

5 Numactl Commands

5.1 migratepages

The man pages says:

“migratepages moves the physical location of a processes pages without any changes of the virtual address space of the process. Moving the pages allows one to change the distances of a process to its memory. Performance may be optimized by moving a processes pages to the node where it is executing.”

5.2 numastat

The `numastat` command shows per-NUMA statistics for processes and the operating system.

Example for one CPU:

```
numastat
          node0
 numa_hit      366309
 numa_miss      0
 numa_foreign   0
interleave_hit    7846
 local_node     366309
 other_node      0
```

5.3 numactl

The `numactl` command controls the [NUMA](#) policy for processes or shared memory.

Example for one CPU:

```
numactl --show
policy: default
preferred node: current
physcpubind: 0 1 2 3 4 5
cpubind: 0
nodebind: 0
membind: 0

numactl --hardware
available: 1 nodes (0-0)
```

```
node 0 size: 6025 MB
node 0 free: 5612 MB
node distances:
node 0
  0: 10
```

5.4 memhog

The `memhog` command allocates memory with a policy for testing. For some reason, the Debian 10 Buster release does not include a man page. However, there is a [page] online (<http://man7.org/linux/man-pages/man8/memhog.8.html>).

Allocate a 1G region, (implicit) default policy, repeat test 4 times

```
memhog -r4 1G
```

5.5 numademo

The `numademo` command is **not** available as a Debian package, it is available as `numactl` source code.

On an old laptop from 2008 with Debian 8.11 Jessie (used with a compiled `numademo` executable from `numactl` source from before 2020, maybe in 2016):

```
./numademo -S 100M
1 nodes available
memory with no policy memset          Avg 2304.98 MB/s Max 2310.25 MB/s
↔ ...
local memory memset                  Avg 2302.23 MB/s Max 2310.05 MB/s
↔ ...
memory interleaved on all nodes memset Avg 2295.28 MB/s Max 2307.35
↔ MB/s ...
memory on node 0 memset              Avg 2303.58 MB/s Max 2306.69 MB/s
↔ ...
[...]
```

Running this test on Debian 11 Bullseye on a 2013 desktop or 2015 laptop, or on the same 2008 laptop (Debian 8.11 Jessie) the newer version gives:

```
/numademo -S 100M
A minimum of 2 nodes is required for this test.
```

It seems that the minimum requirement for the test changed.

6 Numatop

This utility requires a supported CPU. If executed on an unsupported CPU, it will indicate that:

```
numatop -s low -l 2 -f /tmp/warn.log
CPU is not supported!
```

7 Further Reading

- [NUMA - hpc wiki info](#)
- [NUMA - wikipedia](#)
- [github](#)

8 History

Version	Date	Notes
0.1.5	2024-03-07	Add architecture example for lscpu
0.1.4	2023-03-11	Linux note, minor improvements in typeface
0.1.3	2023-03-10	Improve writing, move history
0.1.2	2022-05-17	Change shell blocks to bash block, history, dots description, Debian helper tools table, +lscpu Update for Debian 11 Bullseye
0.1.1	2020-05-01	Update for Debian 10 Buster
0.1.0	2016-03-24	Initial release

9 Disclaimer of Warranty

THERE IS NO WARRANTY FOR THIS INFORMATION, DOCUMENTS AND PROGRAMS, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE INFORMATION, DOC-

UMENT OR THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE INFORMATION, DOCUMENTS AND PROGRAMS IS WITH YOU. SHOULD THE INFORMATION, DOCUMENTS OR PROGRAMS PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

10 Limitation of Liability

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE INFORMATION, DOCUMENTS OR PROGRAMS AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE INFORMATION, DOCUMENTS OR PROGRAMS (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE INFORMATION, DOCUMENTS OR PROGRAMS TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.