NUMA
Non-Uniform Memory Access

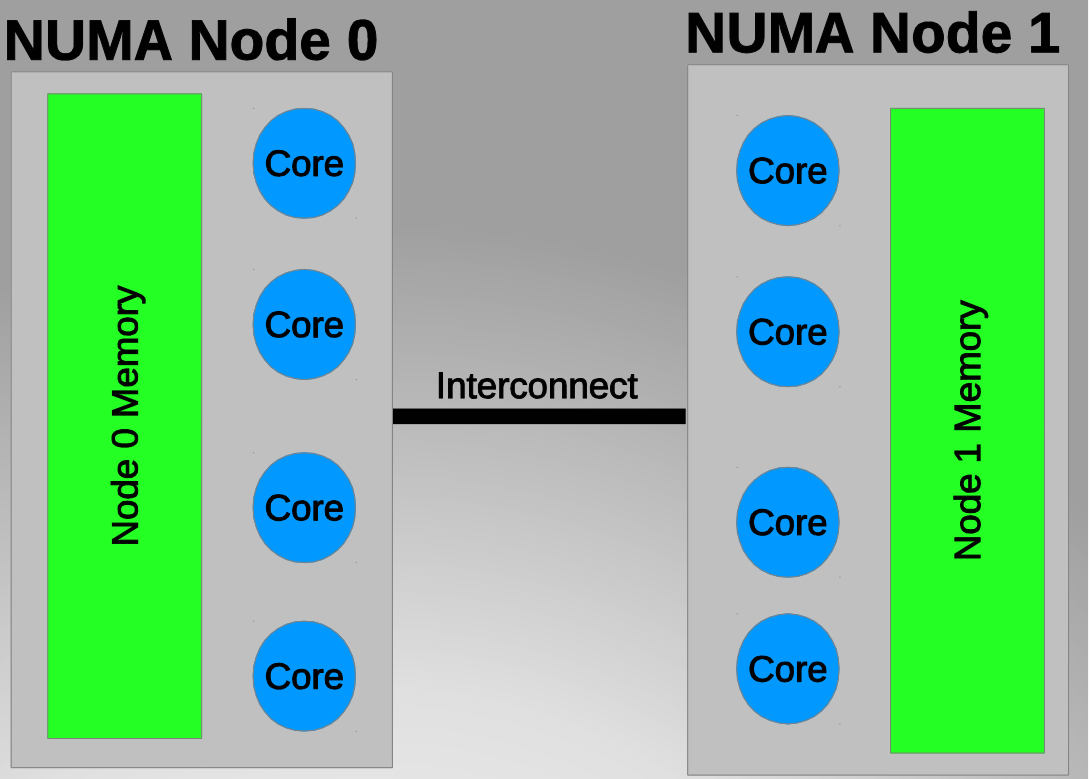
Numa becomes more common because memory controllers get close to execution units on microprocessors

(Article @ http://portal.acm.org/ft_gateway.cfm?id=2513149&type=pdf)

Christoph Lameter, Ph.D.
University of Illinois Urbana, September 30, 2013

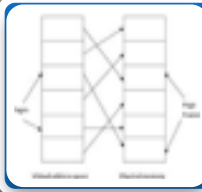
A NUMA System

- "Distance" of memory
- Started on HPC systems
- Linux modifications for NUMA support
- Local and Remote Memory
- Multi socket systems
- NUMA Nodes
- Interconnect



NUMA Placement

Refers to assigning physical memory nodes to memory structures



The virtual address space does nto change with placement. What changes is where the physical memory is located.

Latency of a memory access depends on placement in a NUMA system

Multiple strategies in hardware and software how to assign memory from the available NUMA nodes.

Operating System approaches to NUMA

- Ignoring NUMA effects
  - No OS changes needed. Random placement.
- Striping Memory accesses over all nodes
  - Possible with firmware changes. Balanced access.
- Heuristic memory placement
  - OS modifications to optimize placement
- Special configurations for applications
  - System Administrator sets application parameters
- Direct Application control of NUMA allocations
  - Developer adds NUMA support to the application

Linux and NUMA

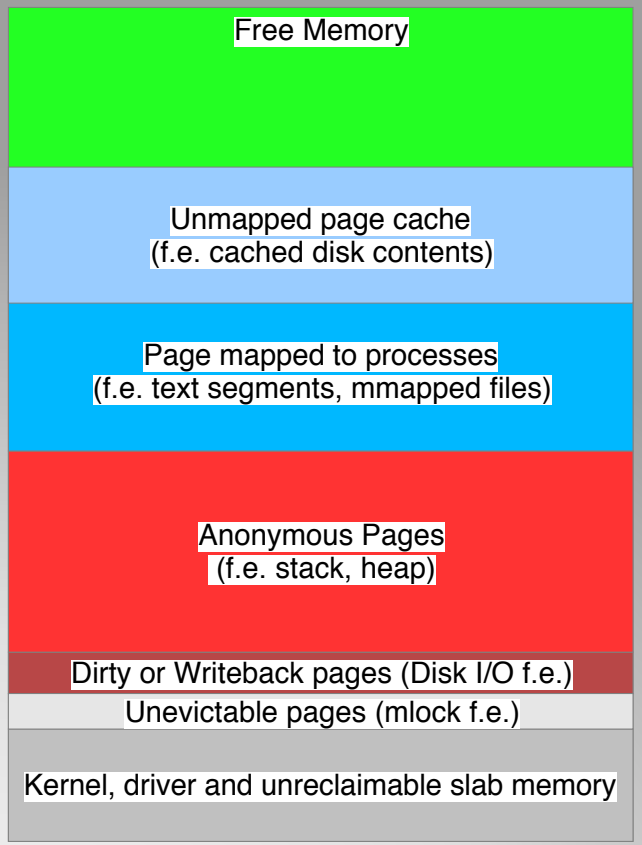
Memory Zones DMA, HIGHMEM, MOVABLE

Replication for each NUMA Node

To see the zones on a Linux system do
cat /proc/zoneinfo

Memory Reclaim

- Memory is continuously allocated
- Reclaim (freeing of memory) occurs when memory is running low.
- LRU freeing scheme.
- Memory allocation under NUMA is determined by **Memory Policies**
- **NODE LOCAL**
- **INTERLEAVE**
- Boot default is **INTERLEAVE**
- Running system defaults to **NODE LOCAL**
- See **/proc/<pid>/numa_maps** for active memory policies on a certain process.
- Processes inherit memory policies from their parents



NUMA Hardware information

- Display the hardware characteristics
- NUMA Distances (SLIT table)
- ACPI (System Vendor Firmware) sets these parameters up

```
$ numactl --hardware
available: 2 nodes (0-1)
node 0 cpus: 0 2 4 6 8 10 12 14 16 18 20
22 24 26 28 30
node 0 size: 131026 MB
node 0 free: 588 MB
node 1 cpus: 1 3 5 7 9 11 13 15 17 19 21 23
25 27 29 31
node 1 size: 131072 MB
node 1 free: 169 MB
node distances:
node 0 1
0: 10 20
1: 20 10
```

NUMA Counters

- Basic NUMA counters
- NUMA_MISS shows allocations that were not optimal
- Good for troubleshooting

$ **numastat**

	node0	node1
numa_hit	13273229839	4595119371
numa_miss	2104327350	6833844068
numa_foreign	6833844068	2104327350
interleave_hit	52991	52864
local_node	13273229554	4595091108
other_node	2104327635	6833872331

The Memory Map of a process

- Numa_maps allows to see what memory has been allocated to a process.
- Node allocation
- File backed vs. Anonymous pages

```
# cat /proc/1/numa_maps
7f830c175000 default anon=1 dirty=1 active=0 N1=1
7f830c177000 default file=/lib/x86_64-linux-gnu/ld-2.15.so anon=1 dirty=1 active=0 N1=1
7f830c178000 default file=/lib/x86_64-linux-gnu/ld-2.15.so anon=2 dirty=2 active=0 N1=2
7f830c17a000 default file=/sbin/init mapped=18 N1=18
7f830c39f000 default file=/sbin/init anon=2 dirty=2 active=0 N1=2
7f830c3a1000 default file=/sbin/init anon=1 dirty=1 active=0 N1=1
7f830dc56000 default heap anon=223 dirty=223 active=0 N0=52 N1=171
7fffb6395000 default stack anon=5 dirty=5 active=1 N1=5
```

Allocate now
Reclaim later

Reclaim first
Allocate on the desired node

- Zone reclaim allows tuning OS behavior for what is wanted.
- Enabling zone reclaim means more reclaim but less NUMA misses.
- Kernel autoconfigures based on the NUMA distances in the SLIT table.

Zone Reclaim

First Touch Policy

- A physical page can be mapped into multiple address spaces
- The policy of the process that touches the page first determines where the page will be allocated.
- Pages may be cached from past use. Memory policies do not relocate pages.
- Numa_maps may show allocation on unexpected nodes.

Memory Migration

- Memory is moved between NUMA nodes
- Can be used to clean up misallocated pages
- Improves performance
- The virtual addresses visible from user space do not change.
- Uses the command **migratepages**. This can be run while the application is accessing the pages.

NUMA Scheduling

- Problem of the scheduler not being aware of where the pages of a process have been allocated.
- It is customary to pin processes to avoid scheduler interference.
- Recently NUMA scheduling features are being added to the Linux kernel but that work is not yet fully mature.

- Questions?
- Comments?
- Opinions?

Conclusion