February 25, 2025

Announcements

- paper responses due Monday
- talk slides coll + perm.. to publish

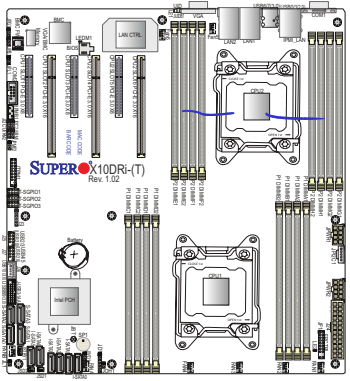
Goals

- no class Tue Mon 4
  <

Review

- multi processor memory

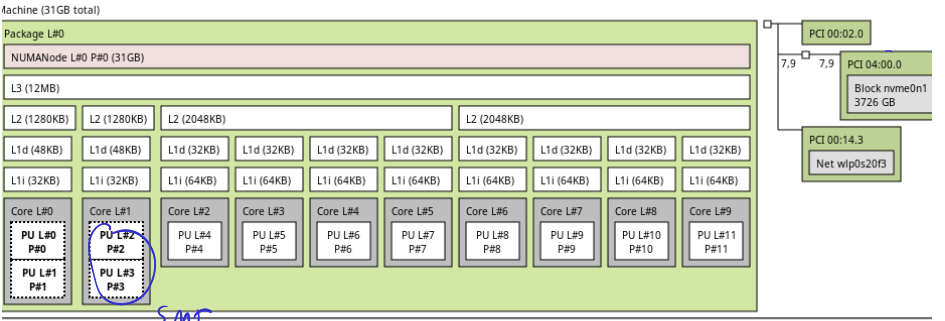
Topology and NUMA

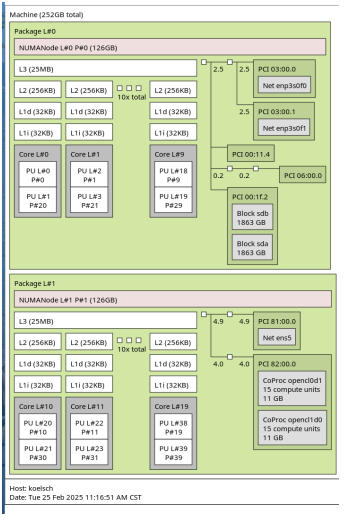


[SuperMicro Inc. '15]

Demo:

- Show `lstopo` on porter, from hwloc.
- `lstopo` on MI300


Machine (31GB total)
Package L#0
NUMANode L#0 P#0 (31GB)
L3 (12MB)
L2 (1280KB)
L2 (1280KB)
L2 (2048KB)
L2 (2048KB)
L1d (48KB)
L1d (48KB)
L1d (32KB)
L1d (32KB)
L1d (32KB)
L1d (32KB)
L1d (32KB)
L1d (32KB)
L1d (32KB)
L1d (32KB)
L1i (32KB)
L1i (32KB)
L1i (64KB)
L1i (64KB)
L1i (64KB)
L1i (64KB)
L1i (64KB)
L1i (64KB)
L1i (64KB)
L1i (64KB)
Core L#0
PU L#0 P#0
PU L#1 P#1
Core L#1
PU L#2 P#2
PU L#3 P#3
Core L#2
PU L#4 P#4
Core L#3
PU L#5 P#5
Core L#4
PU L#6 P#6
Core L#5
PU L#7 P#7
Core L#6
PU L#8 P#8
Core L#7
PU L#9 P#9
Core L#8
PU L#10 P#10
Core L#9
PU L#11 P#11
SMT
PCI 00:02.0
7,9
7,9
PCI 04:00.0
Block nvme0n1
3726 GB
PCI 00:14.3
Net wlp0s20f3
lost: arc
late: Di 25 Feb 2025 11:14:47 CST


Machine (252GB total)
Package L#0
NUMANode L#0 P#0 (126GB)
L3 (25MB)
L2 (256KB)
L2 (256KB)
10x total
L2 (256KB)
L1d (32KB)
L1d (32KB)
L1d (32KB)
L1i (32KB)
L1i (32KB)
L1i (32KB)
Core L#0
PU L#0 P#0
PU L#1 P#20
Core L#1
PU L#2 P#1
PU L#3 P#21
Core L#9
PU L#18 P#9
PU L#19 P#29
2.5
2.5
PCI 03:00.0
Net enp3s0f0
2.5
PCI 03:00.1
Net enp3s0f1
PCI 00:11.4
0.2
0.2
PCI 06:00.0
PCI 00:1f.2
Block sdb 1863 GB
Block sda 1863 GB
Package L#1
NUMANode L#1 P#1 (126GB)
L3 (25MB)
L2 (256KB)
L2 (256KB)
10x total
L2 (256KB)
L1d (32KB)
L1d (32KB)
L1d (32KB)
L1i (32KB)
L1i (32KB)
L1i (32KB)
Core L#10
PU L#20 P#10
PU L#21 P#30
Core L#11
PU L#22 P#11
PU L#23 P#31
Core L#19
PU L#38 P#19
PU L#39 P#39
4.9
4.9
PCI 81:00.0
Net ens5
4.0
4.0
PCI 82:00.0
CoProc opencl0d1 15 compute units 11 GB
CoProc opencl1d0 15 compute units 11 GB
Host: koelsch
Date: Tue 25 Feb 2025 11:16:51 AM CST

Placement and Pinning

Who decides on what core my code runs? How?

OMP_PLACES = cores
pthread_setaffinity_np

Who decides on what NUMA node memory is allocated?

First touch, libnuma

Demo: intro/NUMA and Bandwidths

What is the main expense in NUMA?

Cache Coherence

What is *cache coherence*?

How is cache coherence implemented?

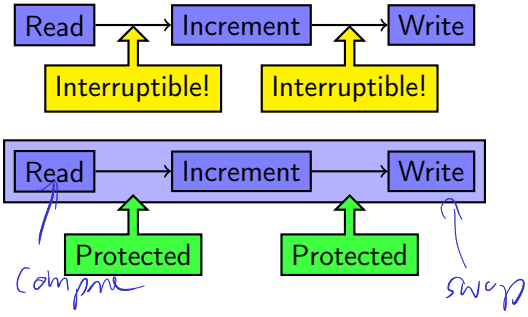
Cache coherence simulation

What are the performance impacts?

- Demo: intro/Threads vs Cache
- Demo: intro/Lock Contention

MESI Sequence to Try

'Conventional' vs Atomic Memory Update



Outline

Outline

Atomic Operations: Compare-and-Swap

```
#include <stdatomic.h>
_Bool atomic_compare_exchange_strong(
        volatile A* obj, C* expected, C desired );
```

What does `volatile` mean?

What does this do?

How might you use this to implement atomic FP multiplication?

read, op, CAS

Memory Ordering

Why is Memory Ordering a Problem?

e.g. Lock vs compiler / processor reordering

What are the different memory orders and what do they mean?

- seq_cst
- acquire : no r/w in this thread reorder before all releasing writes elsewhere are visible!
- release : no r/w in this thread reorder after all writes here are visible to acquiring elsewhere

Example: A Semaphore With Atomics

```
#include <stdatomic.h> // mo_->memory_order, a_->atomic
typedef struct { atomic_int v;} naive_sem_t;
void sem_down(naive_sem_t *s)
{
  while (1) {
    while (a_load_explicit(&(s->v), mo_______) < 1)
        spinloop_body();
    int tmp=a_fetch_add_explicit(&(s->v), -1, mo_________rel);
    if (tmp >= 1)
        break; // we got the lock
    else // undo our attempt
      a_fetch_add_explicit(&(s->v), 1, mo_______);
  }
}
void sem_up(naive_s_t *s) {
  a_fetch_add_explicit(&(s->v), 1, mo_______);
}
```

[Cordes '16] — Hardware implementation: how?