HW1: 50/50

**Homework 1 – Mohammad Atif Faiz Afzal**

**HPC1**

**Date- 9/17/2014**

**Problem 1:**

For $f_c$=0.99,

1. 25/25

$$\frac{average\ cycles}{word\ access} = f_c.\frac{cache\ cycle}{word\ access} + (1 - f_c).\frac{main\ mem\ cycle}{word\ access}$$

$given\ \ f_c$=0.99, $\frac{cache\ cycle}{word\ access} = 2\frac{cycles}{word}, \frac{main\ mem\ cycle}{word\ access} = 100\frac{cycles}{word}$

$$\frac{average\ cycles}{word\ access} = 0.99 * 2\frac{cycles}{word} + 0.01 * 100\frac{cycles}{word}$$

$$\frac{average\ cycles}{word\ access} = 2.98\frac{cycles}{word}$$

$$Performamce = \frac{processor\ speed * \rho_{WM}}{average\ cycle}$$

$\rho_{WM} = \frac{floating\ point\ operations}{memory\ access}$, processor= 2GHz

$$Performamce = \frac{2X10^9 * 2}{2.98} = 1.342\ Giga\ flops$$

For $f_c$=0.01, similarly

$$\frac{average\ cycles}{word\ access} = 99.02\frac{cycles}{word}$$

$$Performamce = \frac{2X10^9 * 2}{99.02} = 40.4\ Mega\ flops$$

## Problem 2:

**(a)**

**Sample slurm script**

**For 8 core nodes**

```
#!/bin/sh
#SBATCH --partition=general-compute
#SBATCH --time=00:15:00
#SBATCH --job-name="prob2_8Core"
#SBATCH --output=prob2_8core.out
#SBATCH --mail-user=m27@buffalo.edu
#SBATCH --mail-type=ALL

#SBATCH --nodes=1

# **********************
# For 8-core nodes
# **********************
#SBATCH --constraint=CPU-L5520|CPU-L5630
#SBATCH --mem=24000
#SBATCH --tasks-per-node=8

tic=`date +%s`
echo "Start Time = "`date`

echo "SLURM job ID        = "$SLURM_JOB_ID
echo "Working Dir         = "`pwd`
echo "Compute Node        = "`hostname`

lstopo --of pdf 8core_topology

echo "All Done!"

echo "End Time = "`date`
toc=`date +%s`
elapsedTime=`expr $toc - $tic`
echo "Elapsed Time = $elapsedTime seconds"
```

Similar scripts were written for 12 core, 16 core and 32 core (both AMD and Intel) nodes were written.

Please note that all the cores available per node were used, therefore there is no red color seen on any cores shown in the graphical topologies

## Output files

### For 8 core node

Start Time = Tue Sep 16 19:30:41 EDT 2014
SLURM job ID = 2784297
Working Dir = /ifs/projects/hachmann/atif/hpc/ass1
Compute Node = d13n13
All Done!
End Time = Tue Sep 16 19:30:41 EDT 2014
Elapsed Time = 0 seconds

### For 12 core node

Start Time = Tue Sep 16 13:32:06 EDT 2014
SLURM job ID = 2781400
Working Dir = /ifs/projects/hachmann/atif/hpc/ass1
Compute Node = k14n13s02
All Done!
End Time = Tue Sep 16 13:32:06 EDT 2014
Elapsed Time = 0 seconds

### For 16 core node

Start Time = Tue Sep 16 19:21:04 EDT 2014
SLURM job ID = 2781398
Working Dir = /ifs/projects/hachmann/atif/hpc/ass1
Compute Node = f16n26
All Done!
End Time = Tue Sep 16 19:21:04 EDT 2014
Elapsed Time = 0 seconds

### For 32 core node (AMD)

Start Time = Wed Sep 17 01:48:51 EDT 2014
SLURM job ID = 2782407
Working Dir = /ifs/projects/hachmann/atif/hpc/ass1
Compute Node = k07n28
All Done!
End Time = Wed Sep 17 01:48:51 EDT 2014
Elapsed Time = 0 seconds

**For 32 core node (Intel)**

Start Time = Tue Sep 16 17:00:38 EDT 2014
SLURM job ID = 2783210
Working Dir = /ifs/projects/hachmann/atif/hpc/ass1
Compute Node = k06n05
All Done!
End Time = Tue Sep 16 17:00:39 EDT 2014
Elapsed Time = 1 seconds

The graphical topology outputs from the above jobs are attached at the bottom of this file.

**(b)**

All the nodes have non-uniform shared memory access (NUMA). It can be seen in the graphical topologies (attached) that all the nodes are mentioned as NUMA. From the topology it can be seen that a memory is associated with each socket, which is why this architecture is NUMA. In case of UMA (uniform shared memory) architecture, the access to memory for every socket is the same.

**(c)** All the nodes have L1d, L1i and L2 cache partitions and a shared L3 cache partition. All these caches are sources of contention for accessing memory.

especially L3 since it is shared, as well as cachce coherency issues

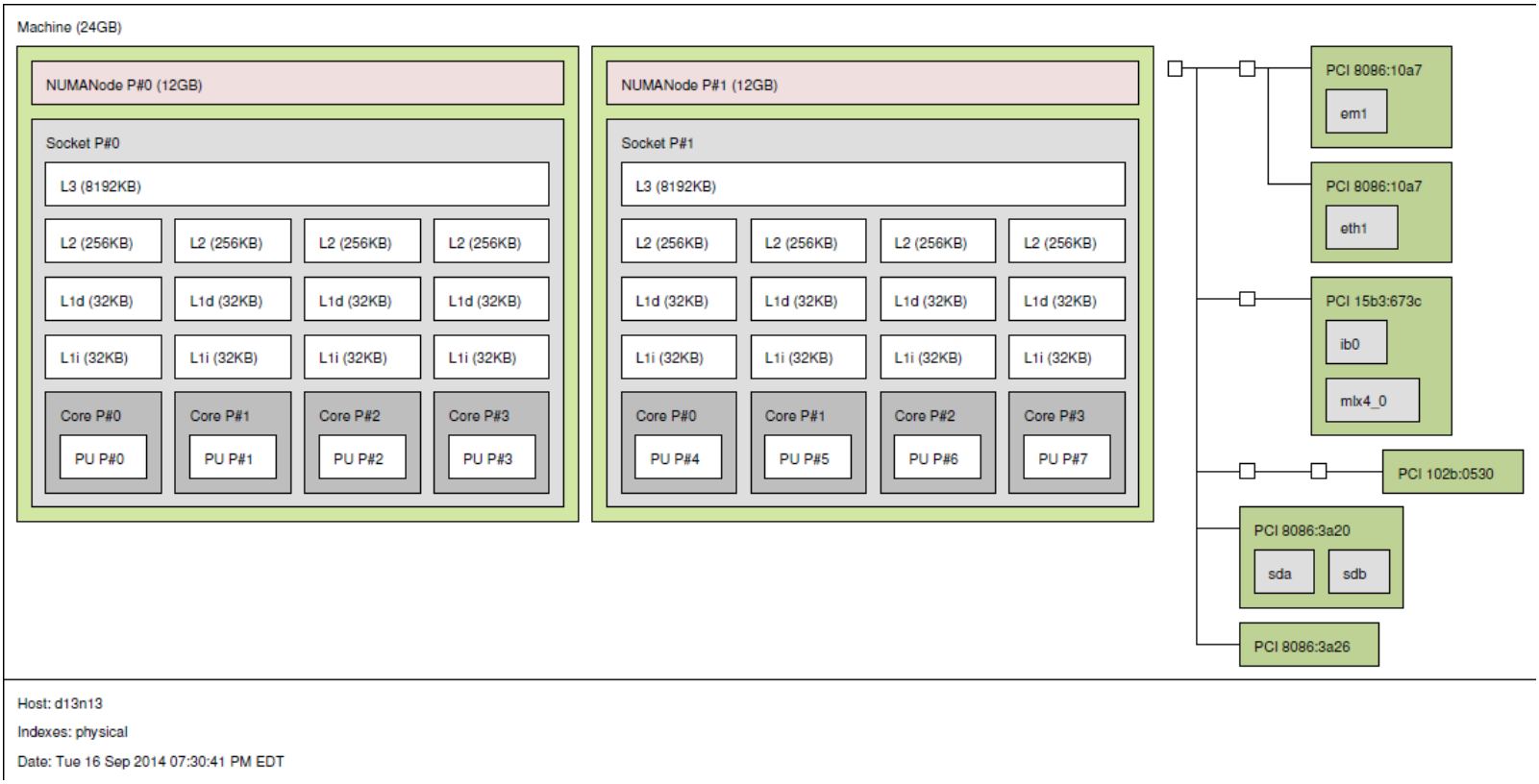


**Figure 1 : 8 core node topoly**

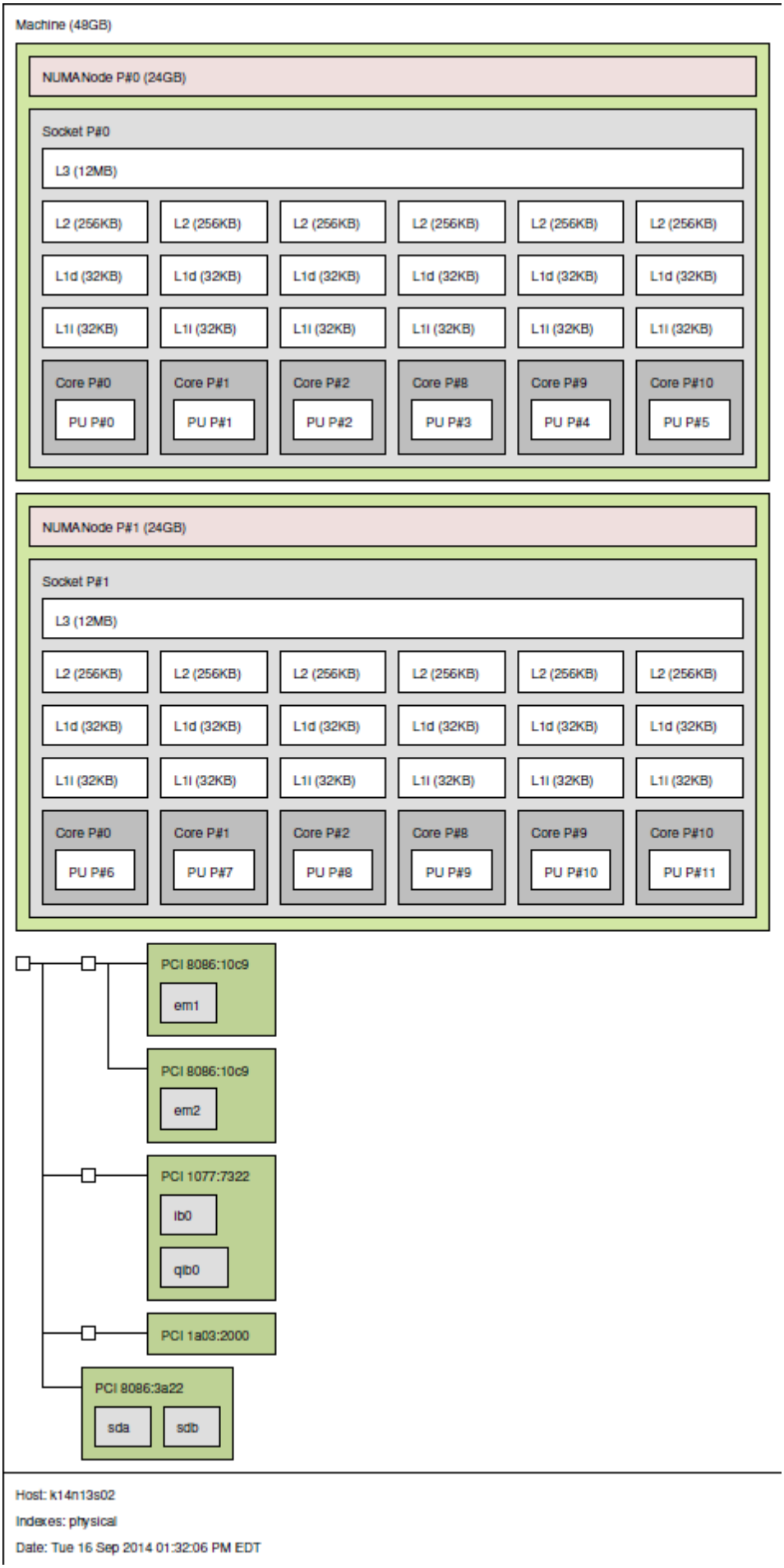


**Figure 2: 12 core node topoly**

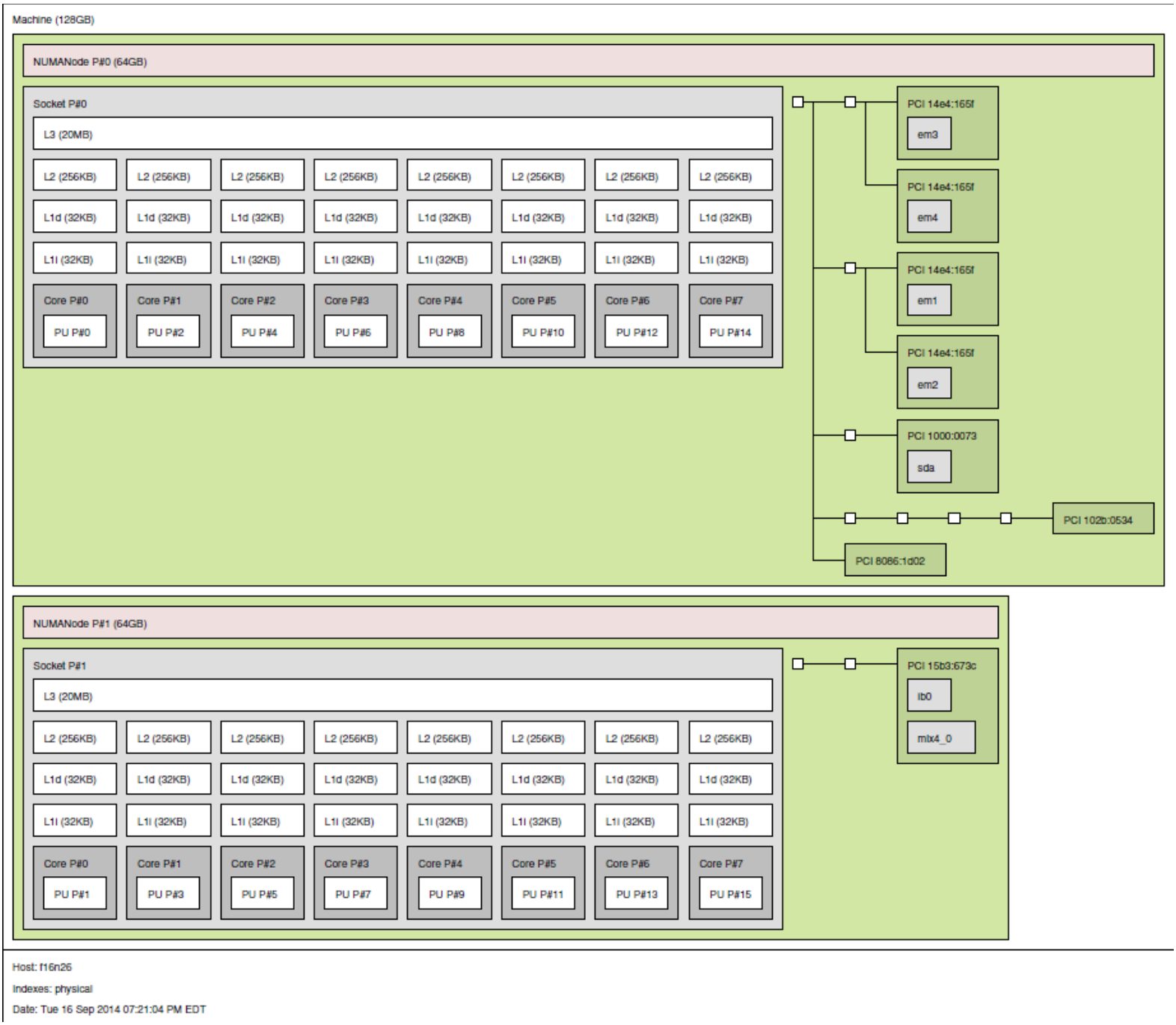


**Figure 3: 16 core node topoly**

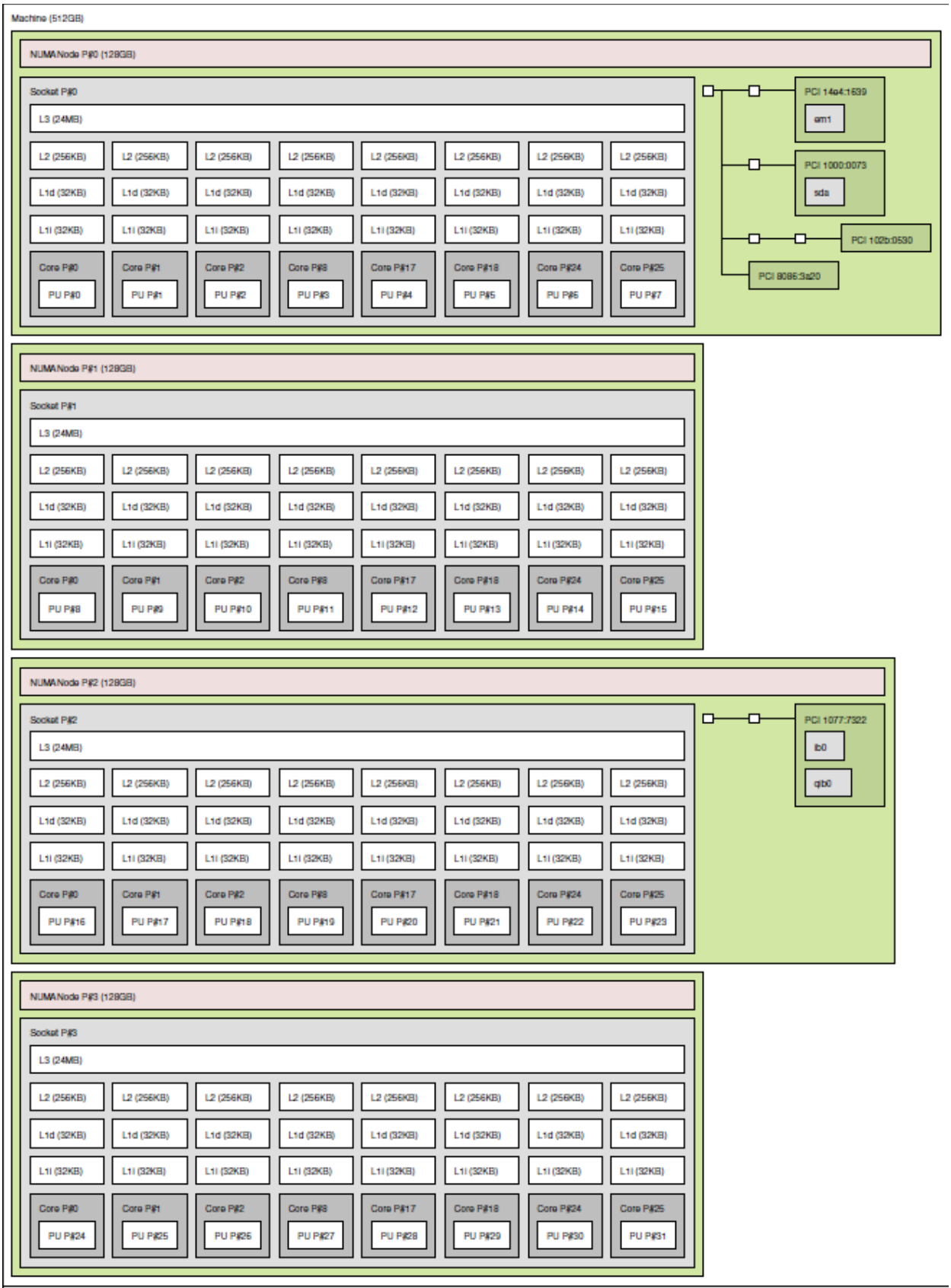


**Figure 4: 16 core node topoly (Intel)**

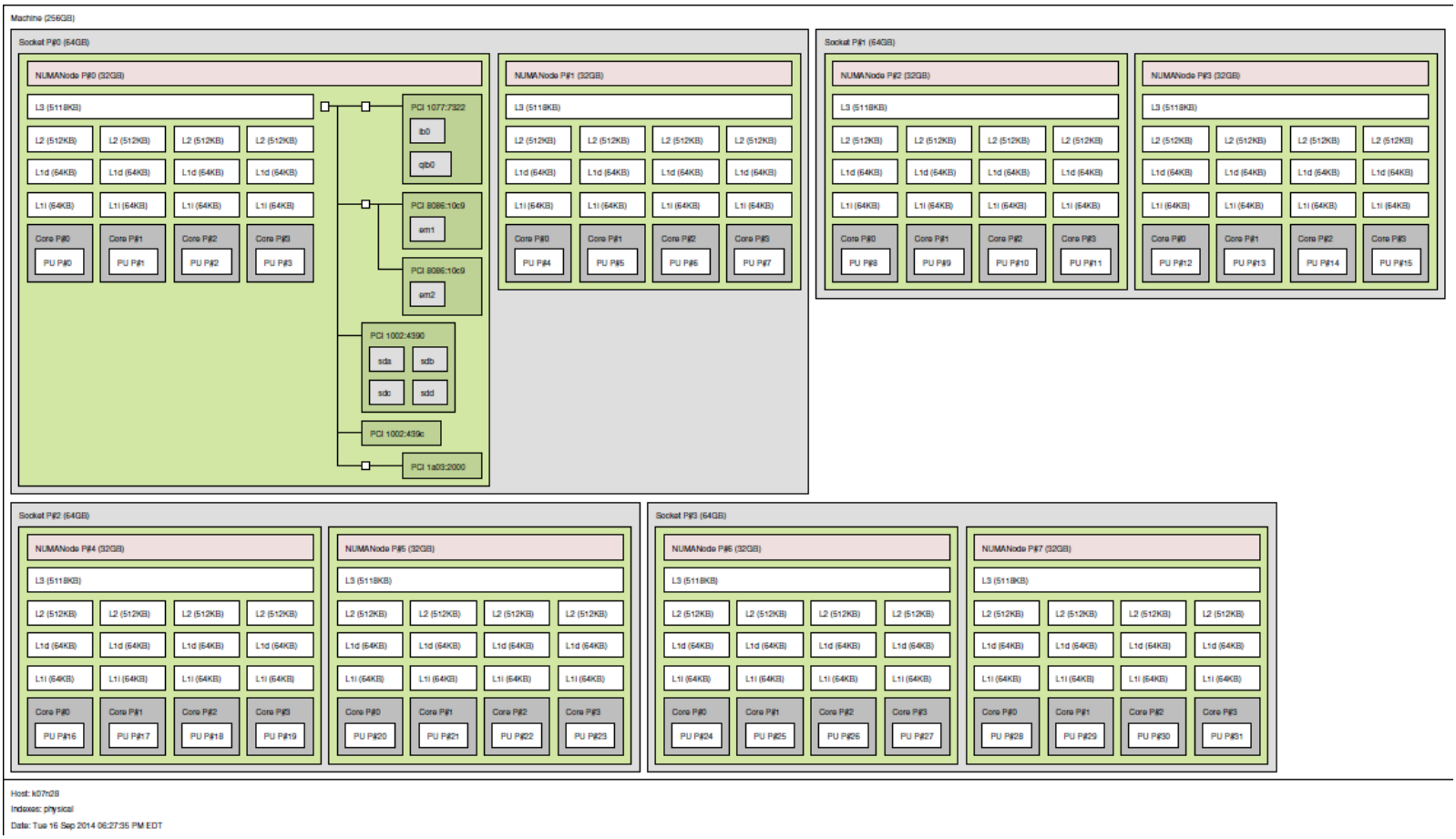


**Figure 5: 16 core node topoly (AMD)**