Question 4:

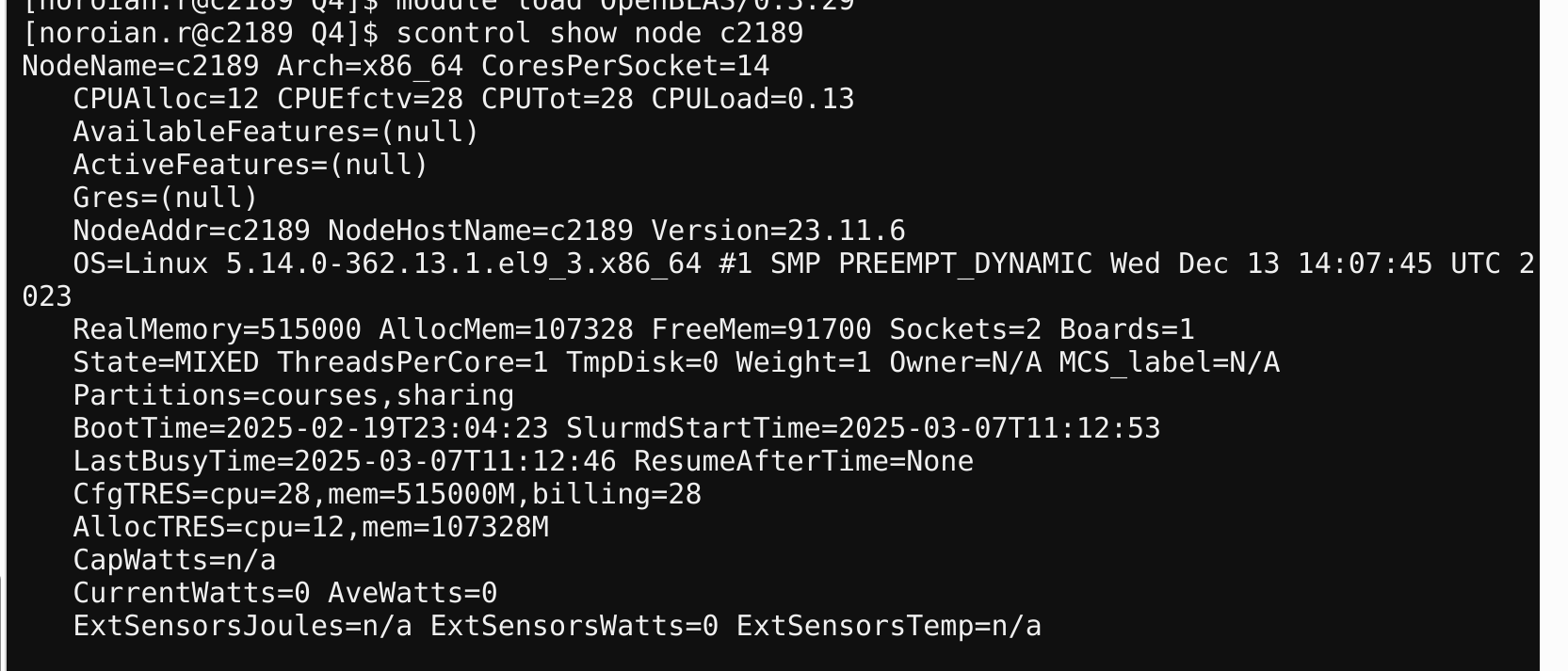
a.) Develop a matrix-matrix multiplication, multiplying two single-precision floating point matrices that are 256x256 elements. Compare your implementation to your dense implementation in problem 3. Discuss which result is faster and why.

For matrix multiplication in 256x256 matrices, I used the CPU 2 from part b below. The resulting time across multiple trials, which fluctuated from around 0.41 ms to 1.1 ms, averaged to about **0.68 ms.** For the dense implementation, performance was optimized from an initial 250 ms to **11.5 ms**. OpenBLAS still remains significantly faster, as it is able to utilize numerous optimizations to cleanup any potential overhead associated with linear algebra computation and find ways to get speedup at all parts of the program.

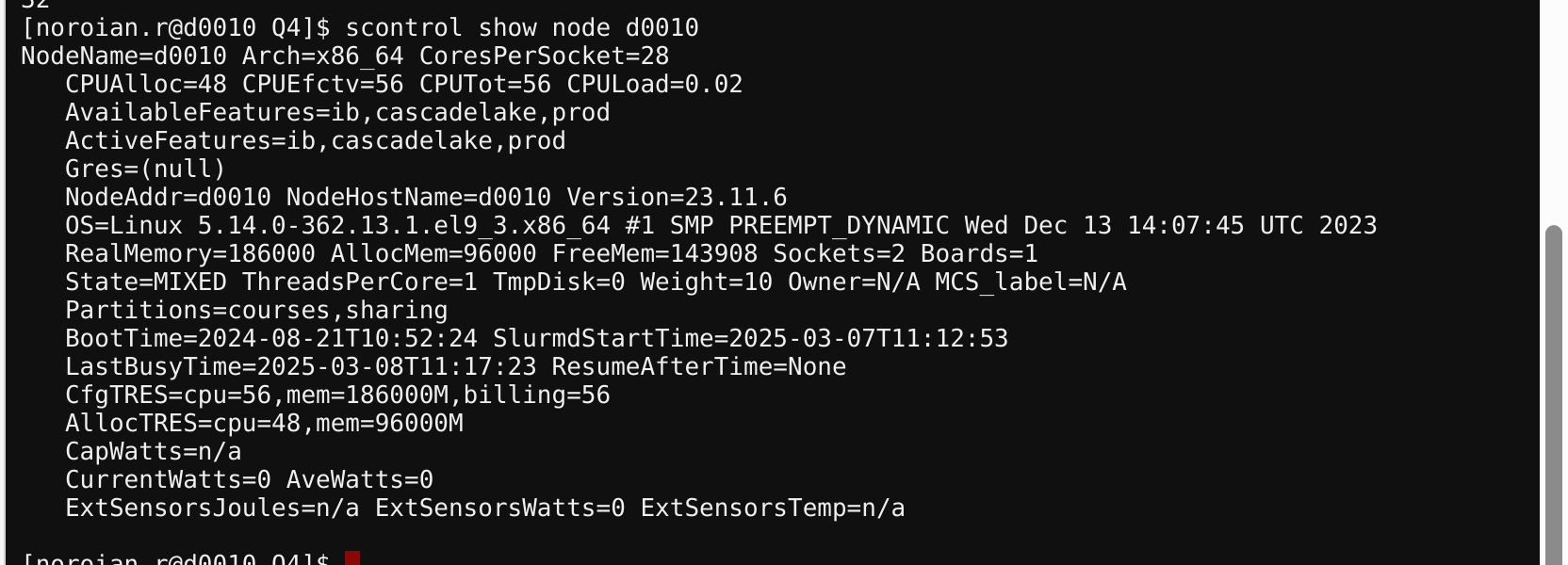
I also tested 512x512 matrix multiplication performance and found that OpenBLAS is superior here as well. The dense implementation in problem 3 has a runtime of about **97.32 ms,** while OpenBLAS runs in about **3.12 ms**. This demonstrates the value of OpenBLAS in computationally intensive programs such as LLM training and other AI applications.

b.) Run your program OpenBLAS accelerated program on two different CPU platforms in Explorer. Discuss the CPUs you are using, and the performance differences you obtain.

CPU 1 Node c2189:

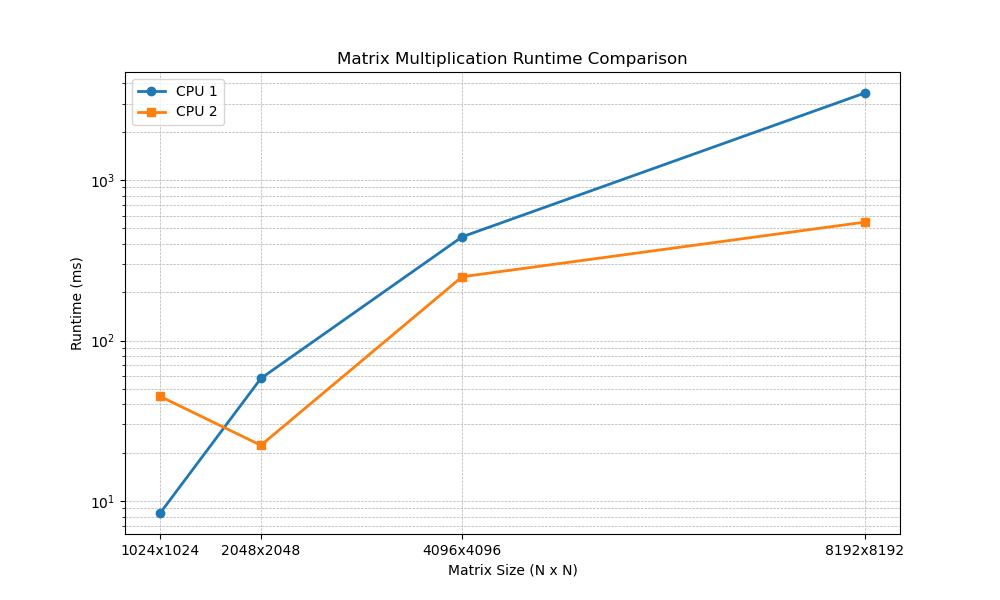
In addition to this, I requested only 4 CPUs per task, so I received a node with 2 sockets of 14 cores each for a total of 28 cores.

CPU 2 Node d0010:

In this implementation, I requested 32 CPUs per task and received a node with 2 sockets each with 28 cores, for a total of 56 cores.

Performances:

|  |  |  |
| --- | --- | --- |
| Matrix Size | CPU 1 runtime (ms) | CPU 2 runtime (ms) |
| 1024x1024 | 8.42 | 44.85 |
| 2048x2048 | 58.08 | 22.24 |
| 4096x4096 | 443.22 | 249.83 |
| 8192x8192 | 3489.8 | 547.07 |

OpenBLAS is a library designed to optimize linear algebra computations such as matrix multiplication. For smaller matrix sizes, allocating CPUs with a greater number of cores showed greater fluctuations in runtime with OpenBLAS and led to worse efficiency compared to fewer cores. This is shown by the greater runtime of CPU 2 despite containing more cores that one would assume greatly improve performance. As the matrix size increases, the runtime scales far better with CPU 2, which possesses more cores. For larger matrices, the performance of CPU 2 is much better, shown by the about 7 times speed difference for an 8192x8192 matrix.