



Numpy



Numpy

NumPy

Numpy

LAPACK

500

500

60000×60000

NumPy

Numpy

LAPACK

MKL

MKL OpenBLAS ATLAS

Intel

Intel CPU

MKL

conda

```
conda install numpy
```

Anaconda

CPU

Intel

MKL

AMD

OpenBLAS

conda-forge

conda-forge

pip

pip

```
pip install numpy
```

OpenBLAS

MKL

--	--	--

MKL ☐ Numpy ☐

MKL_NUM_THREADS

Numpy

A horizontal bar divided into four segments. The first segment is labeled '10%', the second '30%', the third 'CPU' followed by '10%', and the fourth '50%'.

MKL

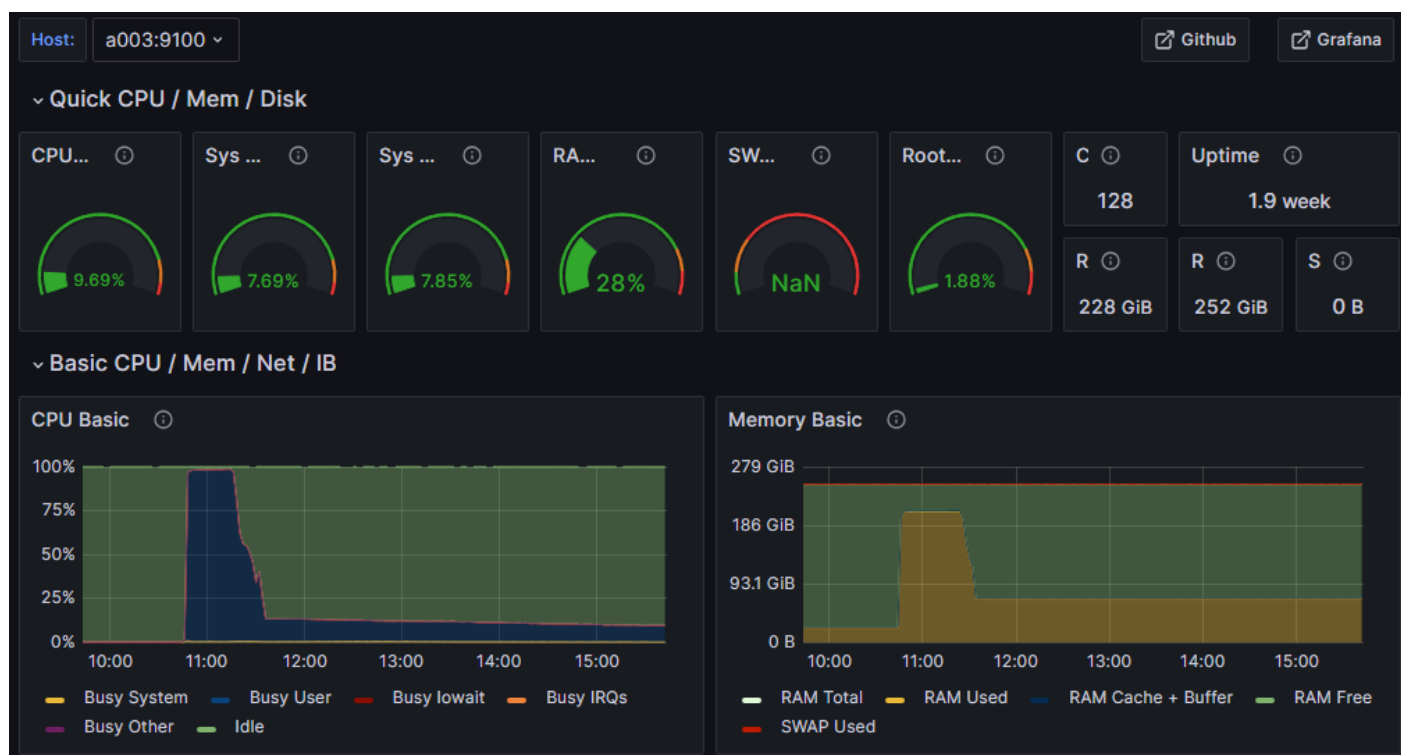
--	--	--	--	--	--	--	--

OpenBLAS[]

AMD

--	--	--	--	--	--	--	--

MKL_DEBUG_CPU_TYPE=5 [] [] [] []

MKL ☐ ☐ ☐

 eScience

40

16

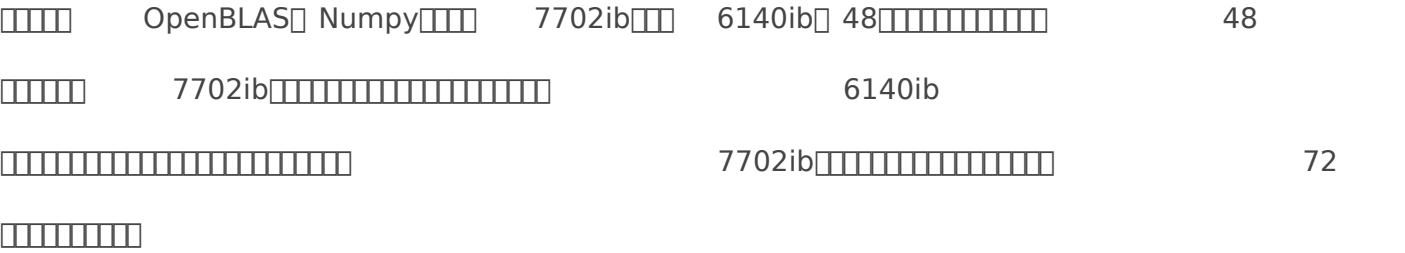
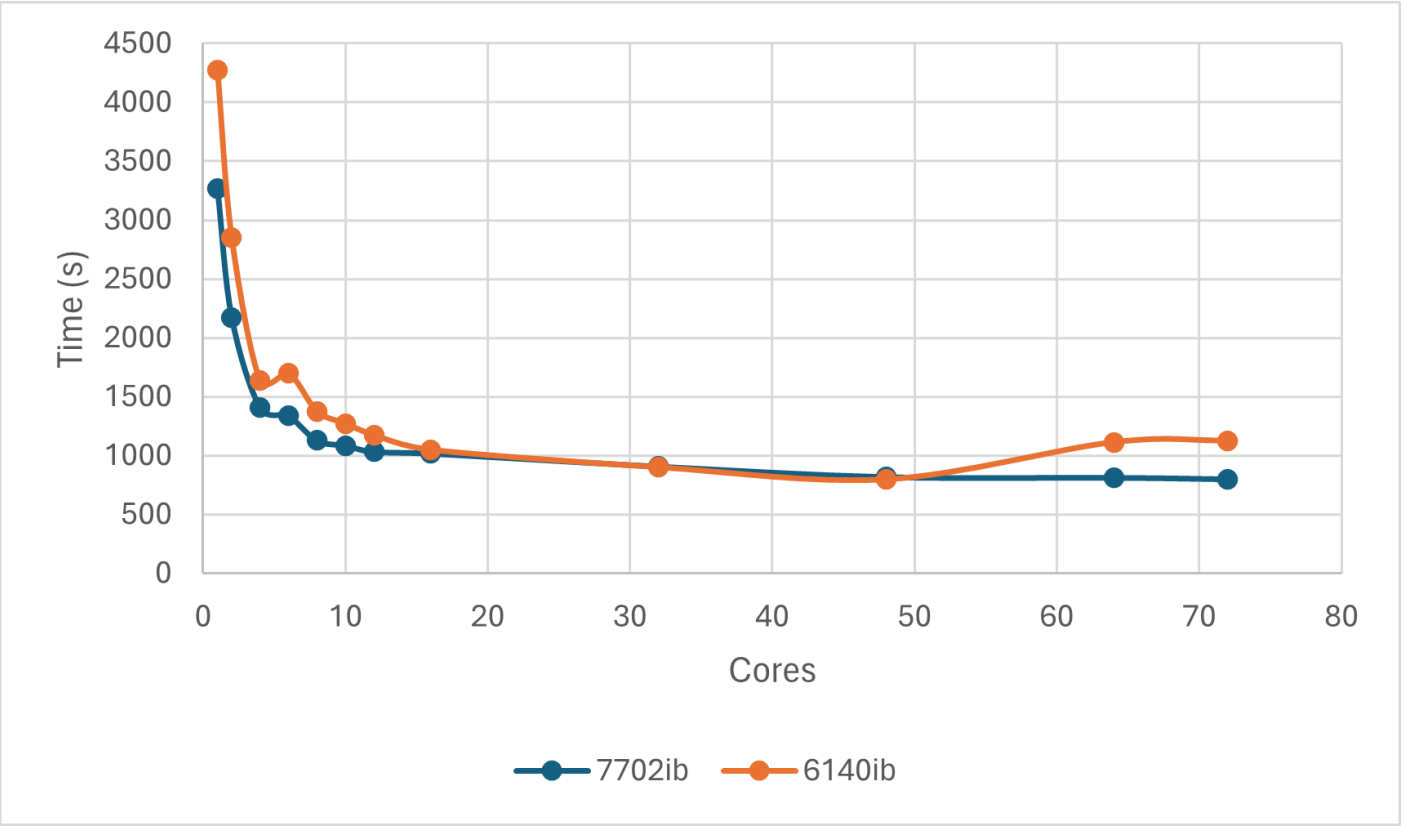
40

44

[illegible]

"

[illegible][illegible]



Progress bar (25 segments)

Progress bar (10 segments)

Progress bar (6 segments)

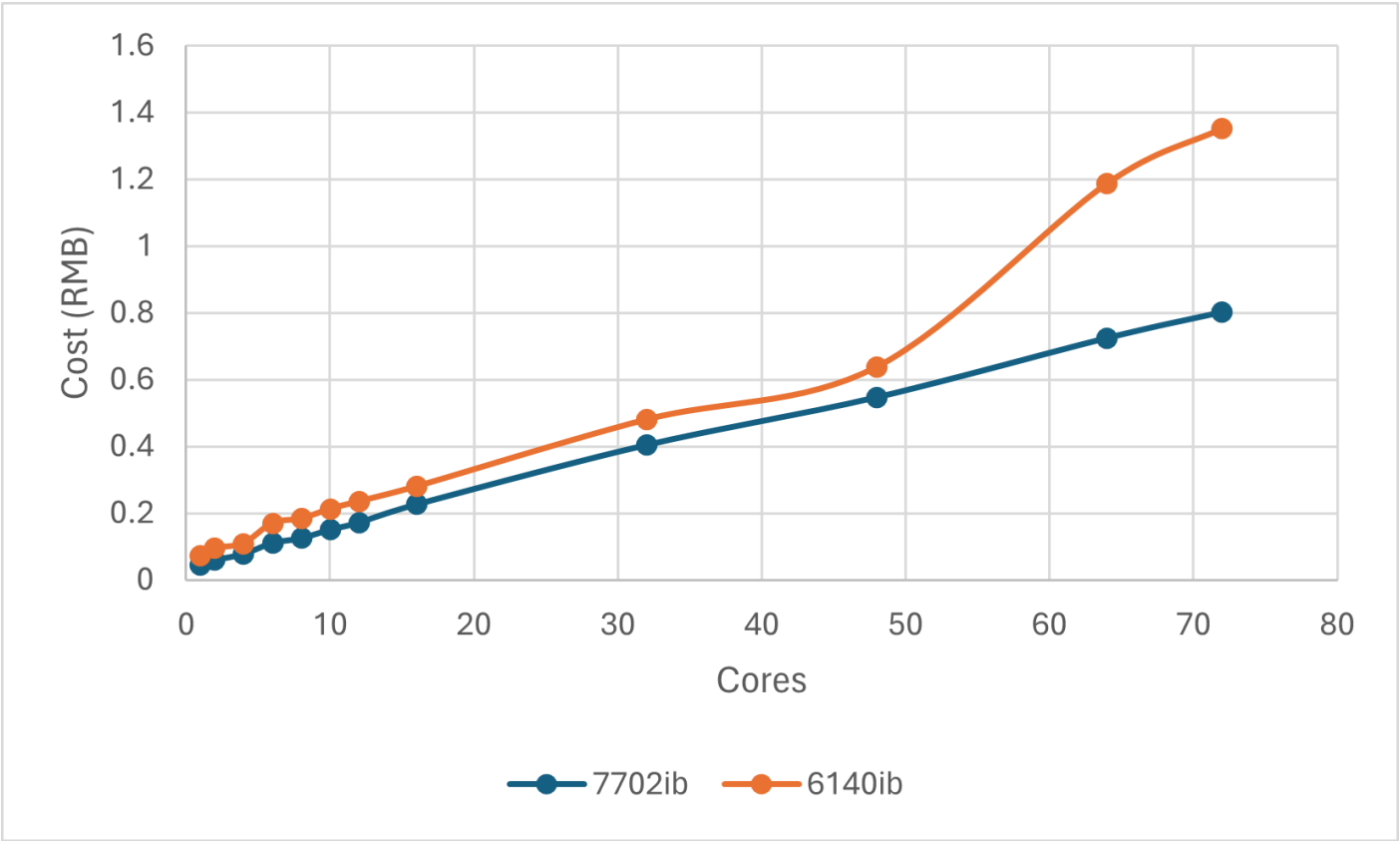
Progress bar (30 segments)

CPU Progress bar (25 segments)

Progress bar (4 segments)

Progress bar (30 segments)

Progress bar (3 segments)

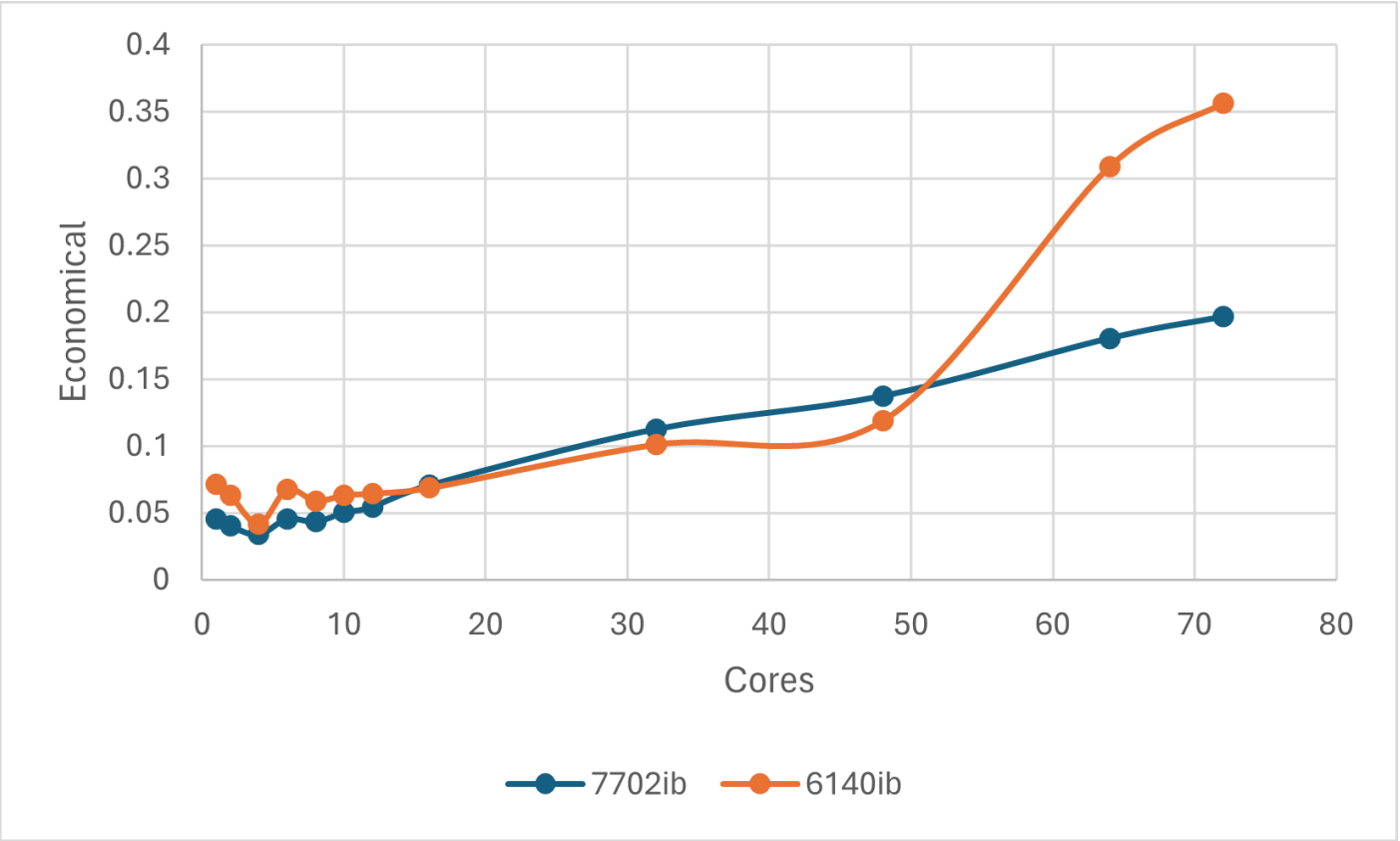


Progress bar (30 segments)

Progress bar (4 segments)

$$T_{total} = N_{task} \times \left(\frac{N_{max}}{N_{cores}} \right)^{-1} \times T_p(N_{cores})$$
$$= \frac{N_{task}}{N_{max}} \times N_{cores} \times T_p(N_{cores})$$

```
def main(N_task, N_max, N_cores, T_p):
    T_total = (N_task / N_max) * N_cores * T_p(N_cores)
    return T_total
```





████████ Numpy██

1. Numpy

██

███

2. MKL Numpy████ Numpy

██ LAPACK████ MKL

OpenBLAS

██

██

3. Numpy██ 4~5

██

████████ 4~16████ 4

██