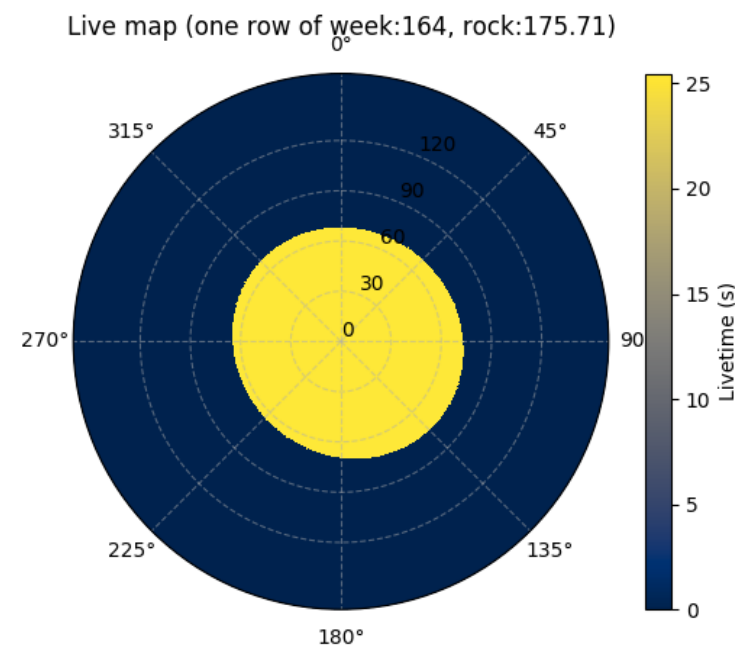
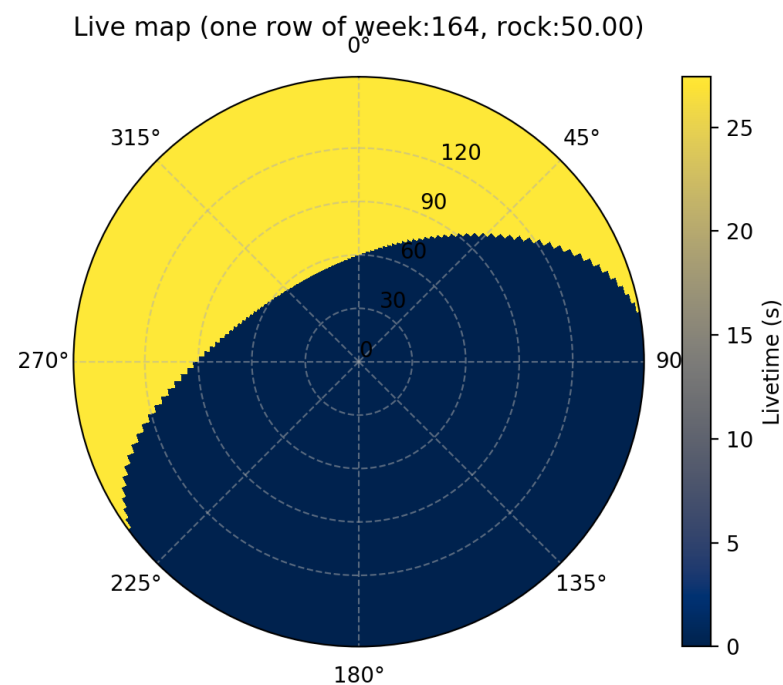


**Maximize the performance of CPU
to compute the exposure map by
using multiprocessing via MPI**

Patomporn

Problem: calculation of exposure map

- FT2 files is genuinely a log table of the position of the spacecraft with a step time
- One step time could calculate the exposure map where we consider the visible region as the example

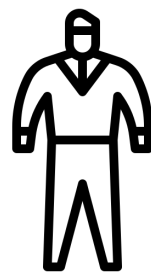
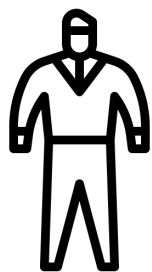
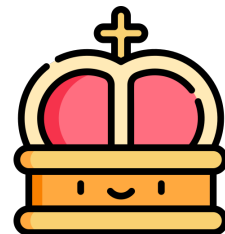


Elapsed time

- Calculate the exposure map of a single week (week:164)
- Performance
 - Python (coordinate TF with numpy): 1434.76 s
 - Plain C++: 11.85 s

Master-Slave technique with MPI

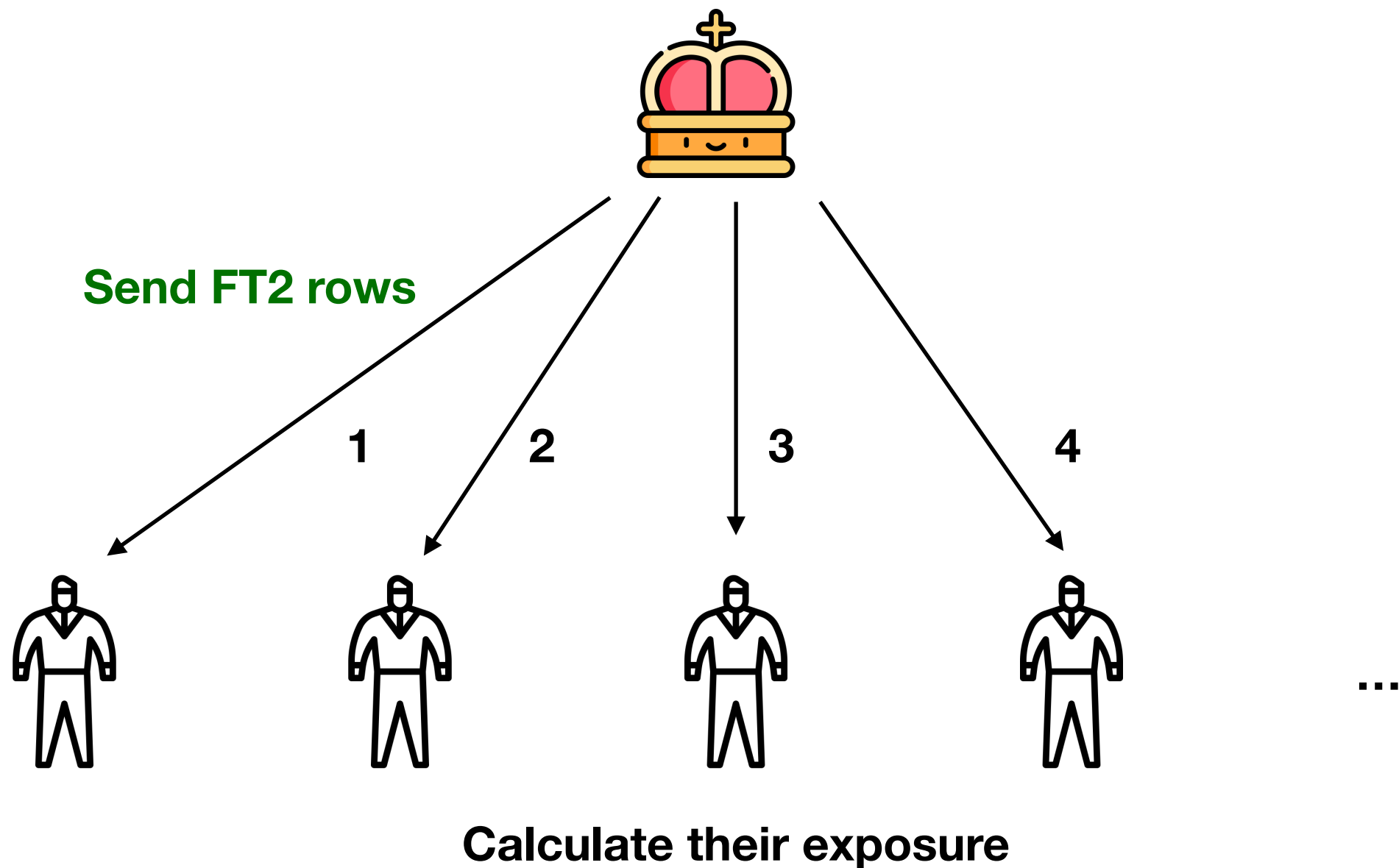
Spawn king and slave!



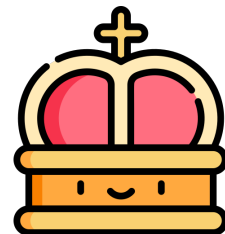
...

Each slave hold their zero exposure map when the process is begin

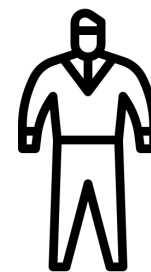
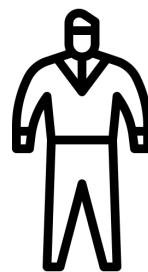
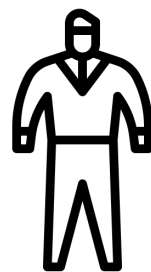
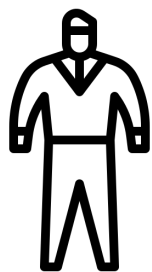
Master-Slave with MPI



Master-Slave with MPI



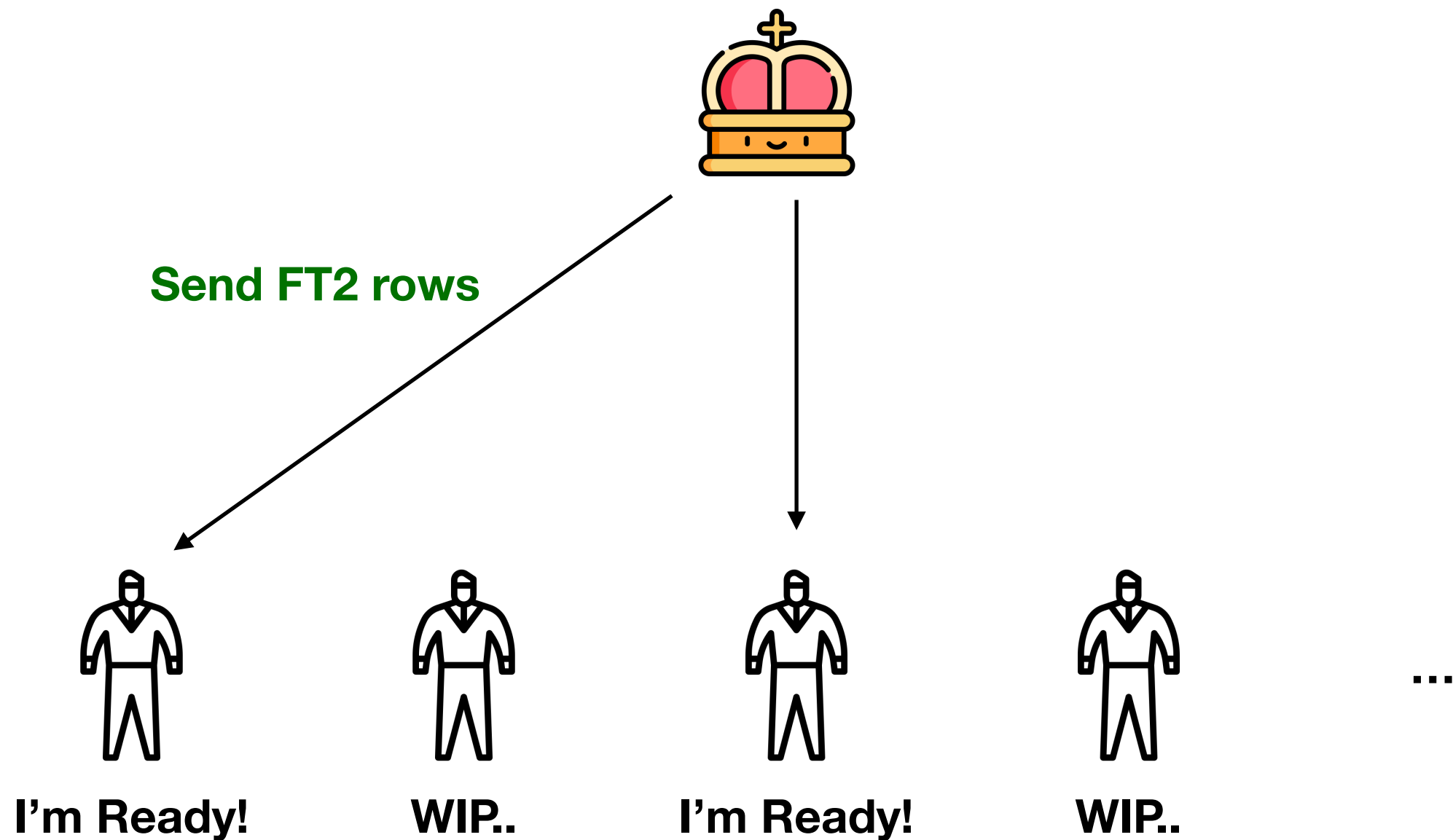
**Tell me when you
finish one FT2 row...**



...

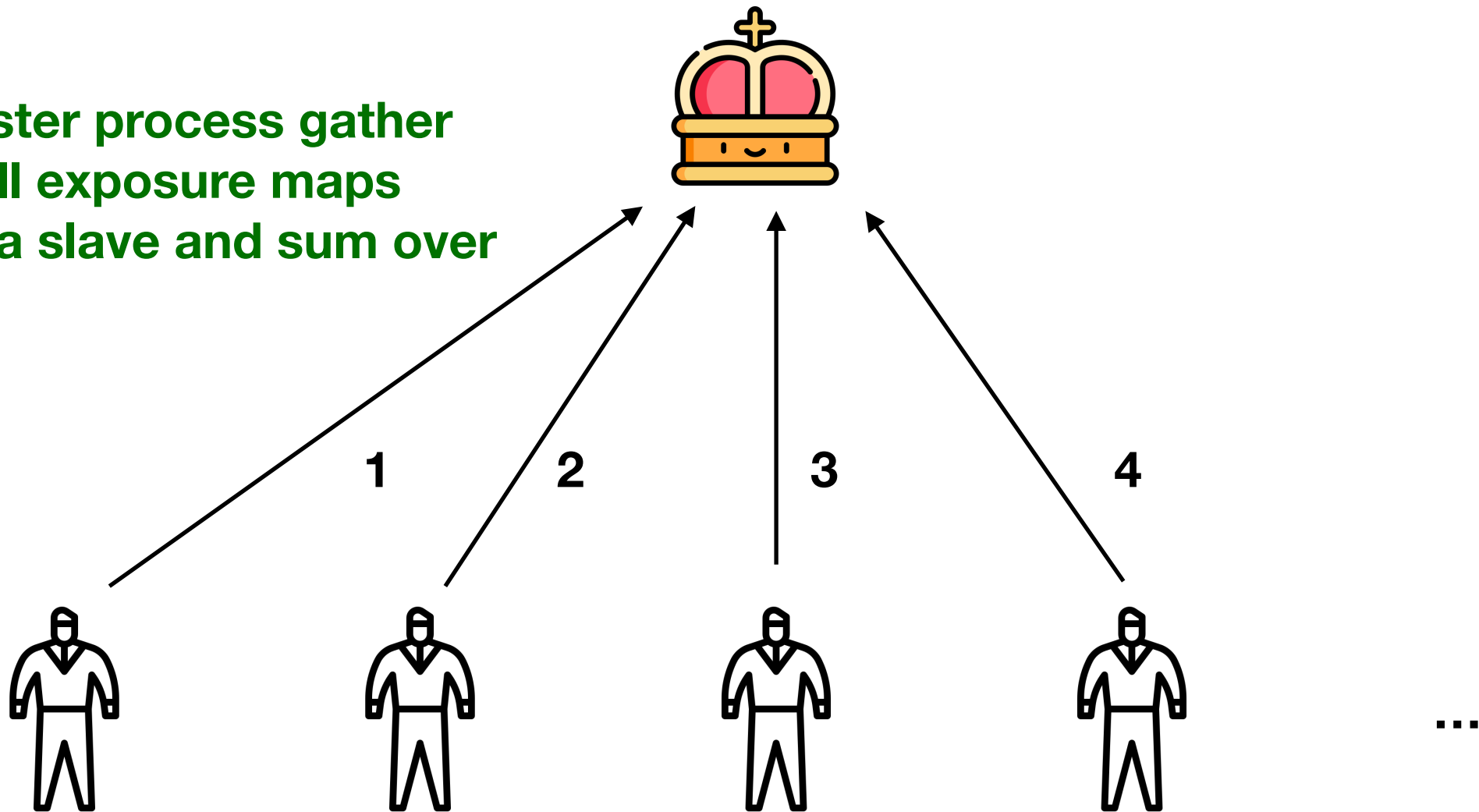
Hold on.. we are working on it

Master-Slave with MPI



When the calculation is done

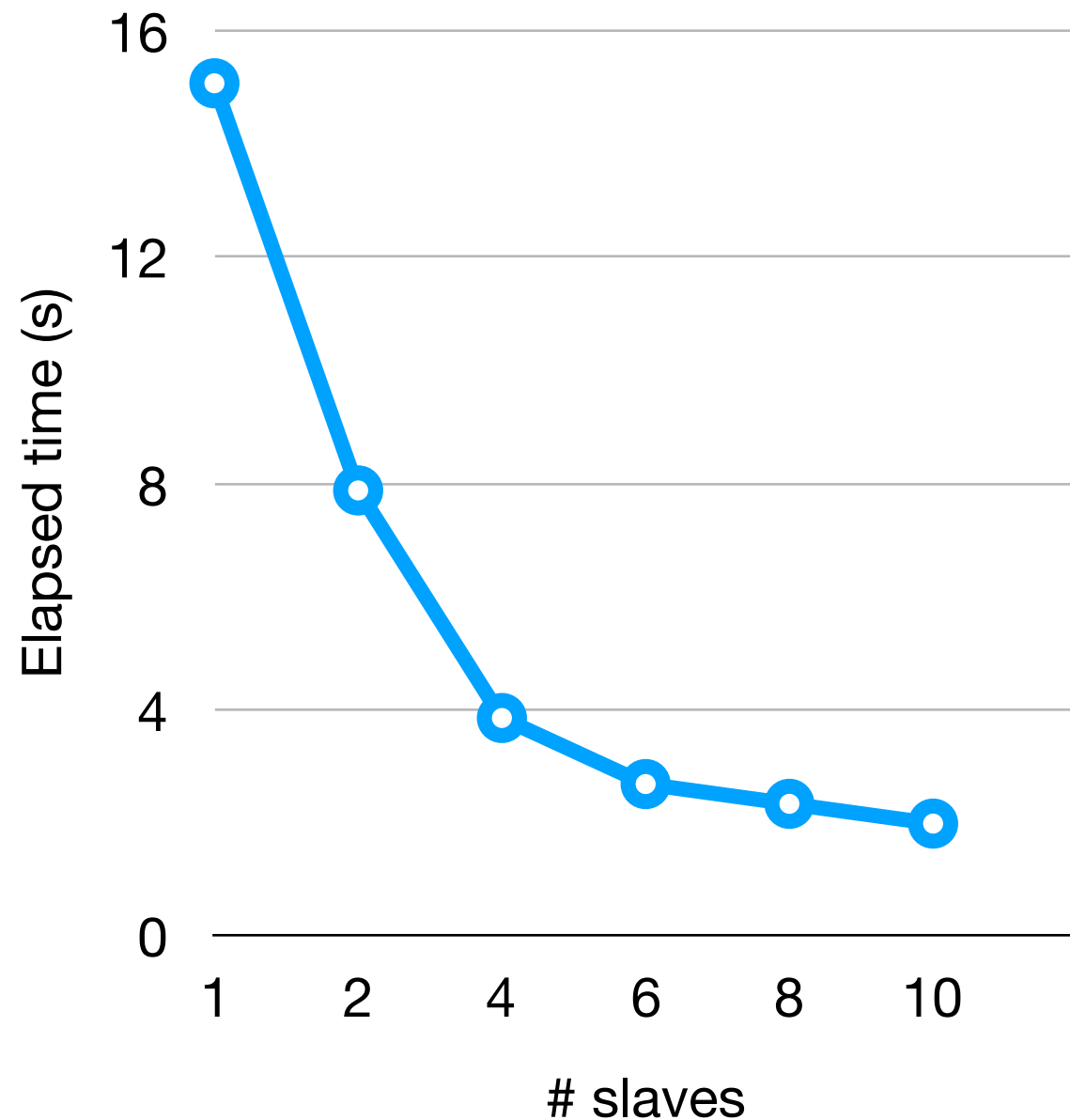
Master process gather
all exposure maps
from a slave and sum over



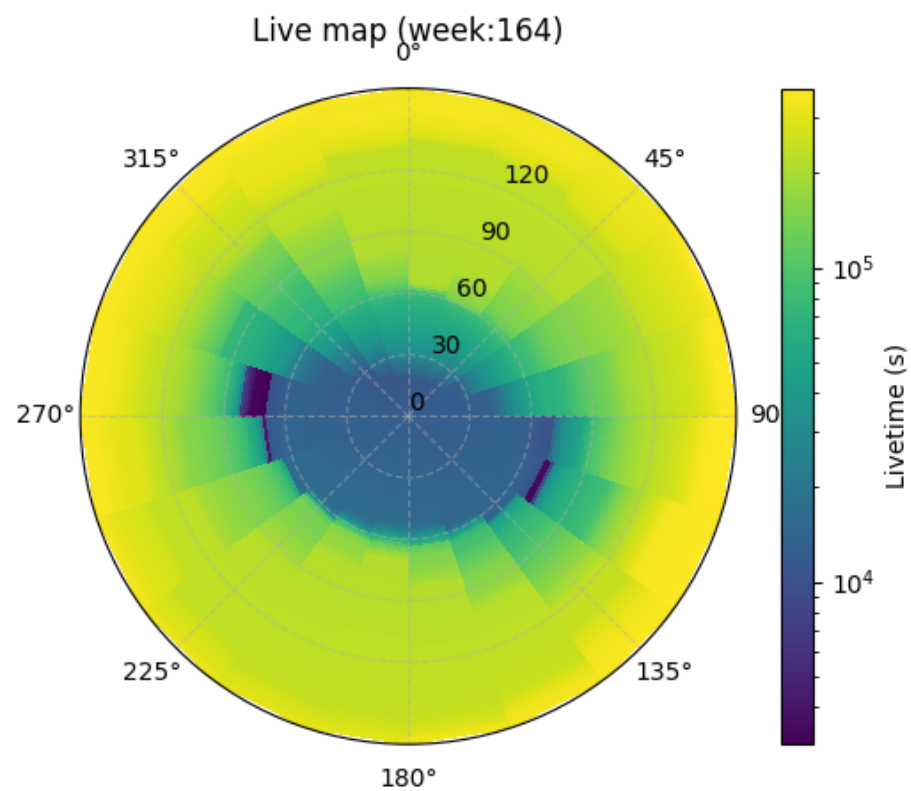
Performance

Testing CPU: 6 cores 12 threads

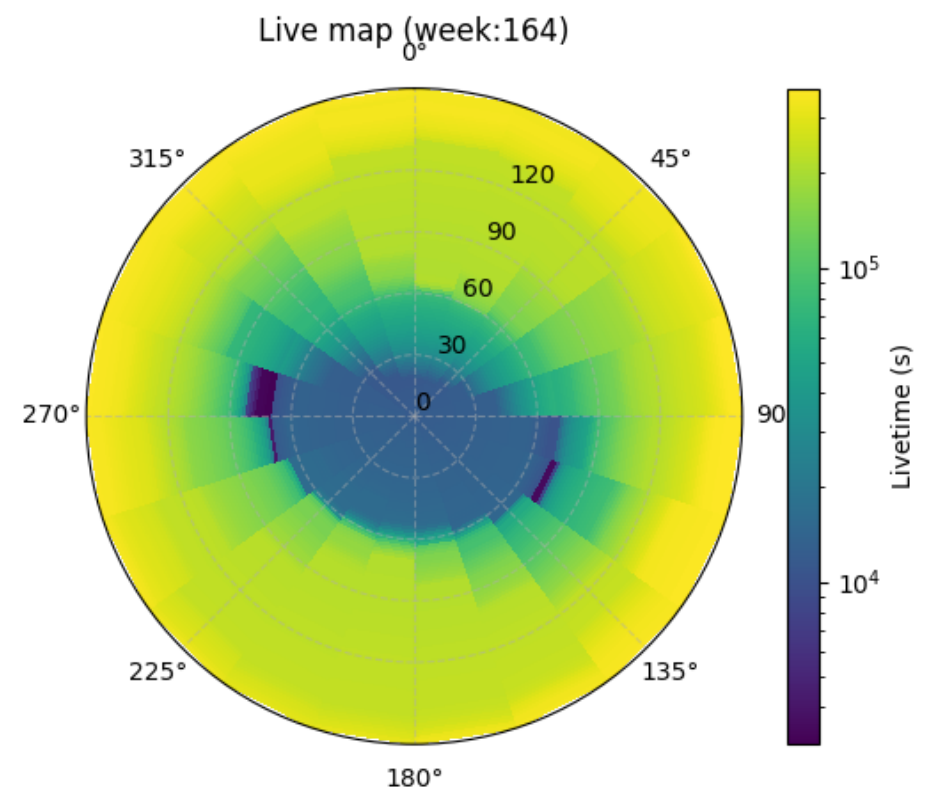
- Plain C++: 11.85 s
- C++ with MPI
 - 1 slave: 15.08 s
 - 2 slaves: 7.87 s
 - 4 slaves: 3.84 s
 - 6 slaves 2.67 s
 - 8 slaves: 2.32 s
 - 10 slaves: 1.97 s



Comparison



Python



C++ with MPI