Sunday, November 24, 2024 09:30

- cache map:
  - possible servers for tasks
  - RAM & cores taken into account

- subtask 1:
  - only 1 server

- subtask 2:
  - no "complete by"

- subtask 3:
  - inf RAM, "complete by"

- when it reads, it has "minimum" turns to finish

- brute force sol'n:
  - compute

- must start by turn # = comple

next available turn = curr_turn + # of turns to complete

- reading takes 0 runtime
- once a task is finished you deallocate RAM & cores
- server only reads one task at a time ...
- server only reads one task at a time ...
  - you can allocate / store tasks as long as long as there's enough RAM.
  - you can append a task anywhere in the queue of a server (given space
  - RAM & cores differ only in queuing ... can queue if core < total cores, but cannot queue
  - read & run can occur same turn, but can only read 1 per turn
  - order matters
  - preprocessing takes time
  - 8,9 test case files

Potential Weighing Formula:

(Deadline - # of turns) - task order in csv - RAM → how much

can only read 1 per turn in order

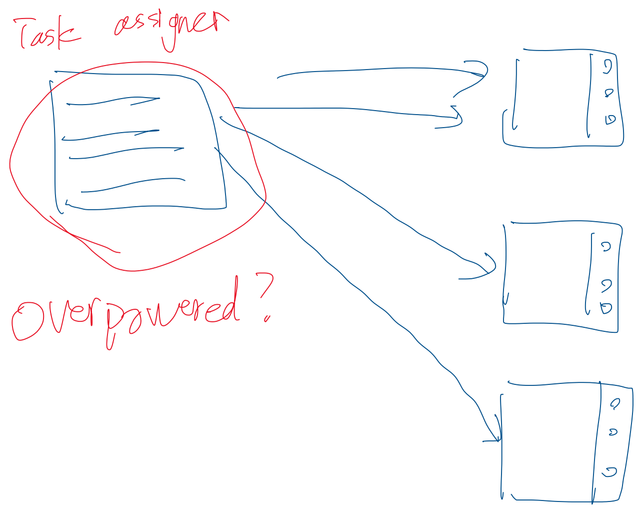
servers that support the task (RAM, Core)

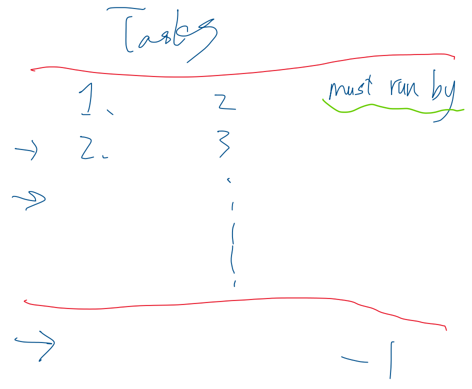
does it affect other tasks,

- can be seperate calculation, to determine if task is worth doing, and is possible

which servers can we use it on, and also their POWER USAGE

- preprocessing counts towards time

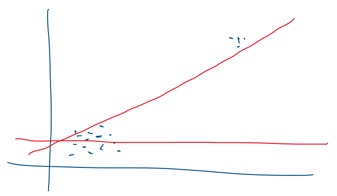




$$\frac{\text{tasks complete}}{\text{tasks total}} \times 25$$

make assumption

$$\frac{\text{turns complete}}{\text{median turns} \times \text{\# of tasks}} \times 15$$



use medians, since edge cases mess up averages

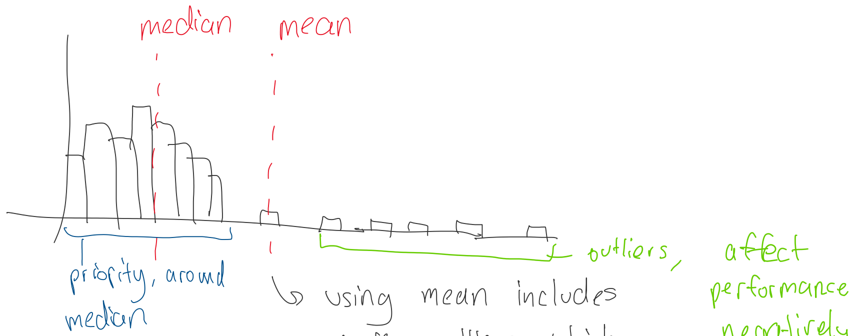
power used
(median power consumption

adjust for # of tas
completed, idk how yet

For presentation:

why we use median:

- the data will probably be skewed, we say the majority of data will be the same but there will be crazy outliers that raise the mean if we use average

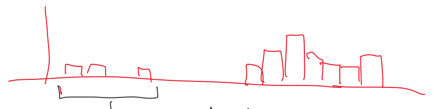


↳ using mean includes more outliers, which we should discard in favour of resources and targetting easier tasks

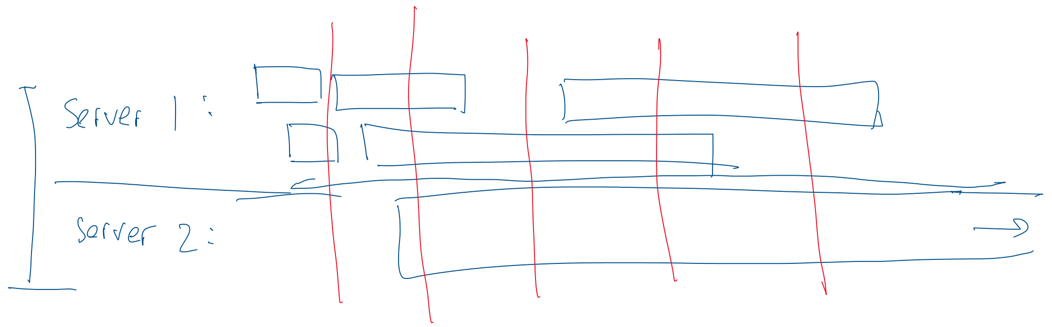
Example: having a server using 1 million Watts/turn would encourage the program to care less about power usage if we had a mean, but wouldn't greatly affect median

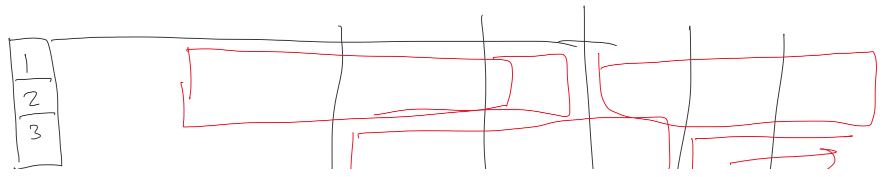
↳ If we can, generate ~~statistics~~/program output using mean and median

Note:



we actually prefer LOW OUTLIERS since they boost performance instead





[4]

turn	1	2	3	4

1				
2				
3				
4				