# Scaling policy

# Terms

- A/I/O/T : num of allocated/idle/other/total nodes
- R: num of reserved nodes
- J: job in the queue
- P:job is running
- Min: the minimum number of nodes reserved

# Assumption

- All jobs will take a reasonable LONG time
- Jobs won’t come and go very frequently(slow change)

# Case 1 simple increase

- I>0
- No J in the queue
- R increases until I=0 by submitting job

# Case 2 make a way

- Assume I==0
- One or more jobs are waiting for resources
- Pick the job with minimum resource requirement $J_{min}$
- If $J_{min}$ <R-Min -> which means there are enough resources can be shared to this job
- Then kill $J_{min}$ reserved nodes( can be nodes fetching task most recent)
- $J_{min}$ will be on running due to the backfilling policy

# Case 3 (Other job) finished

- Let’s say T=15, R=8,P1=3,P2=4,Min=7,J1=5
- Assume P2 finished, then I==4
  - At this time J1 require 1 more nodes to become running while R-Min=1 then kill one node
- Assume P1 finished, then I==3
  - At this time J1 require2 more nodes to become running while R-Min=1<2 then R increase to absorb the 3 idle nodes, and decrease next time when P2 finished
- Over all assumption, no jobs will require too much resources which will led to: J>(T-Min), then these jobs will never be executed