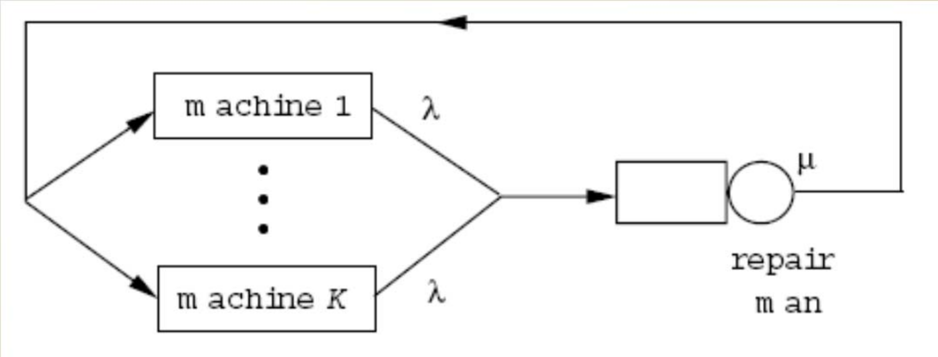
UCF Stands For Opportunity

CDA6530: Performance Models of Computers and Networks

Project 4 solution: realistic simulation using discrete event simulation (Fall, 2011)

Queuing Network: Machine Repairman Model



- c=3 machines
- Each fails at rate λ=0.2/min (expo. distr.)
- Single repairman, repair rate μ=0.5/min
- Define: N(t) – no. of machines working
  - $0 \leq N(t) \leq c$

Analytical Results

$$\sum_{i=0}^{c} \pi_i = 1 \Rightarrow \pi_0^{-1} = \sum_{k=0}^{c} \frac{1}{k!}\left(\frac{\mu}{\lambda}\right)^k$$

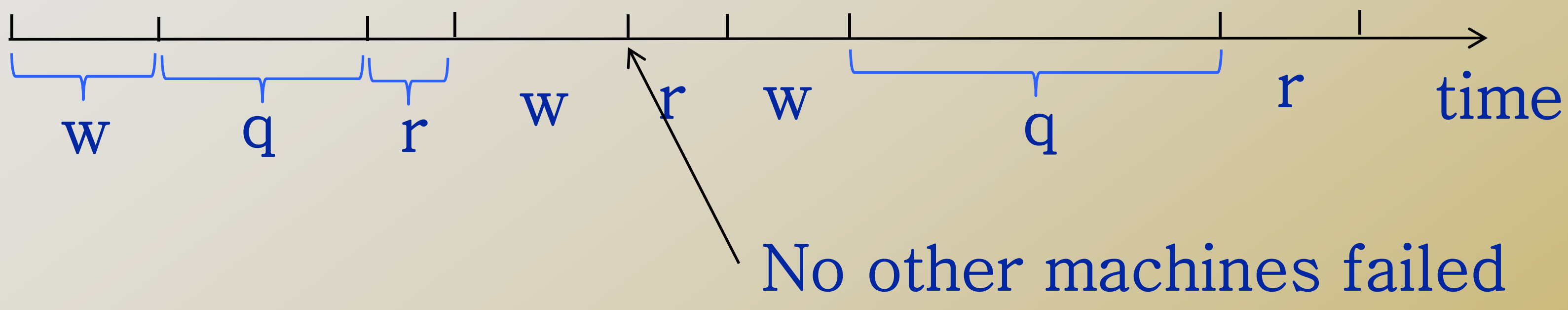
$$\pi_k = \frac{1}{k!}\left(\frac{\mu}{\lambda}\right)^k \pi_0$$

- Utilization rate? η=P(repairman busy) = 1-π_c
- E[N]? $$E[N] = \frac{\eta\mu}{\lambda}$$

Pre Simulation

- Physical entities to consider?
  - c=3 working machines
  - Do not need to consider repairman since state of machine can represent repairman's state
  - workingNum: number of working machine
- Each entity's data:
  - Status:
    - 'working', 'queuing', 'repairing' (w, q, r)
  - nextT: next event time
    - = expo. with λ (when 'working')
    - = ∞ (when 'queuing')
      - How long need to wait depend on others and the queue
    - = expo. with μ (when 'repairing')

Pre Simulation



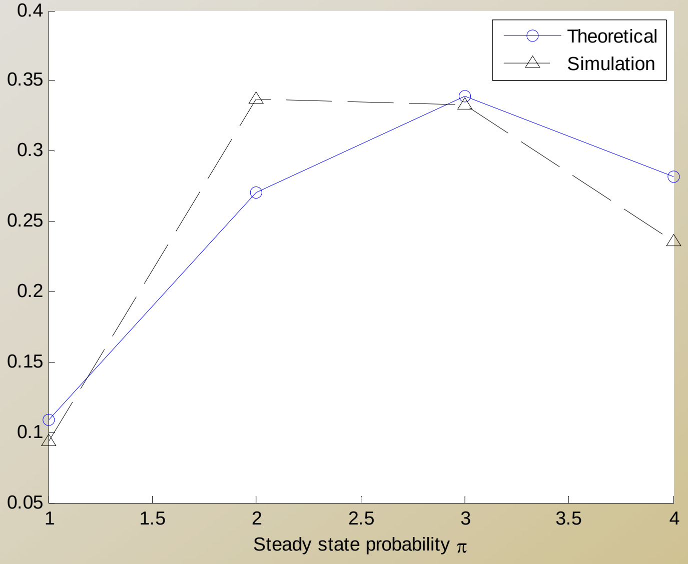
- Each machine's state transition (above)
- Need to record the repairman queue
  - Queue (1:3)
    - Queue(1)=k: the k-th machine is in repair
    - Queue(2)=k: the first job in the queue is the k-th machine (k=1,2,…, c)
    - queueNum: number of machines in the queue

Pseudo Code

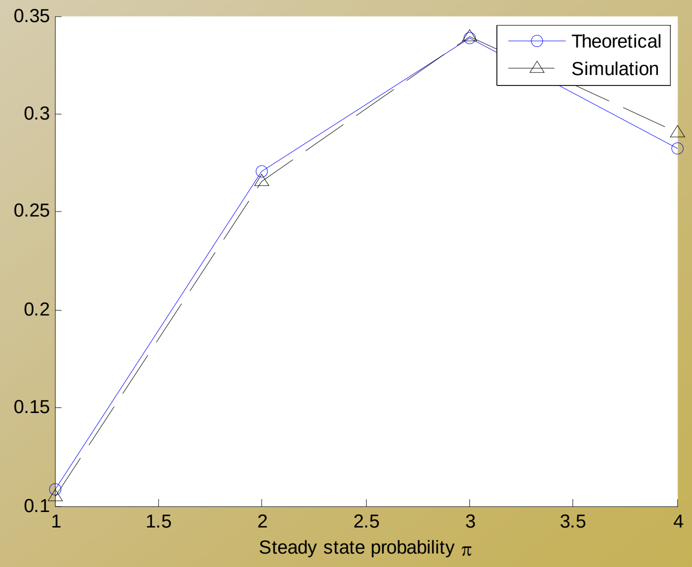
- Event List:
  - EventList (1:3): next event time of each node
- Next event: suppose it is EventList(k)
  - Means k-th machine acts, action determined by its current status
  - currentTime = EventList(k)
  - If Status(k) == ‘w’   (broken event)
    - Update: Queue(), queueNum, workingNum
    - If k is the first job in queue
      - Status(k) = ‘r’, EventList(k) = currentTime + expo(μ) (next event: finish repair)
    - Else: Status(k) = ‘q’, EventList(k) = ∞
  - If Status(k) == ‘q’   (your code is wrong!)
  - If Status(k) == ‘r’   (finish repair)
    - Update: Queue(), if machine j moves to repair, update EventList(j)
    - Update: queueNum, workingNum, repairNum
    - Status(k) = ‘w’,
    - EventList(k) = currentTime + expo(λ)

Post Simulation

- Update Output Data: (i-th event)
  - Tran(i) = currentTime
  - SystemState(i) = workingNum



100 repaired



10000 repaired

What we learned?

- Three elements in discrete event simulation:
  - Objects, events, event list
- Coding lesson:
  - Writing out a code outline is absolutely necessary!
    - Don’t try to write code directly from scratch!