MDPs for Energy-Aware Multiple Server Systems

Vincent J. Maccio, Douglas G. Down

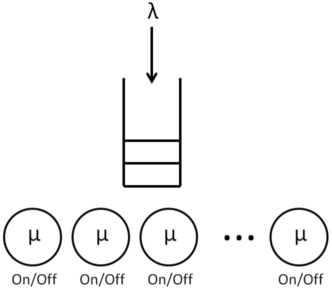
Department of Computing and Software
McMaster University
Hamilton, Ontario, Canada

Motivation

- Server farms use a lot of energy
- Systems are designed to handle peak loads
  - Some servers spend a lot of their time idle
- When a server idles it *still* uses a lot of energy
- Why not just turn them off?
  - Overhead, system performance suffers, could use more energy in the long run etc.

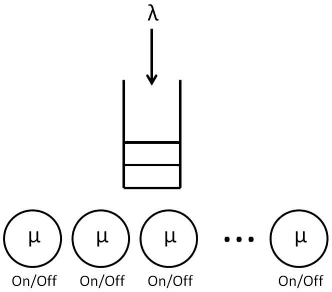
The System of Interest

- c homogeneous servers, each can be in one of four states
  - *Busy, Idle, Off, Setup*
- System parameters rate, all underlying distributions are exponentially distributed
  - Jobs arrive at rate λ
  - Jobs are processed at rate μ
  - Servers turn on at rate γ (turn offs happen instantly)
- System parameters are assumed to be constant and outside the control of the manager



The System of Interest (cont)

- Associated metrics, $\mathbb{E}[R]$ (expected response time), $\mathbb{E}[E]$, (expected amount of energy being used)
  - We can also consider the rate at which servers are turning on/off
- Servers consume energy at different rates while in their different states
  - Ratios relative to when the server is *Busy* denoted r_{Idle}, and r_{Setup}
- Cost function is monotonically increasing with respect to the system metrics
  - $\mathbb{E}[E]\mathbb{E}[R]$, $\mathbb{E}[R] + \beta\mathbb{E}[E]$, etc.

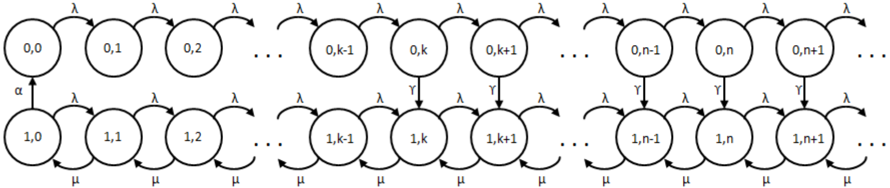


The Optimal Policy

- When does one turn a server off? When does one turn a server on? How many?
- Choices are optimally made only the moment an event occurs
  - Consequence of the exponential distributions
- The system can be described by a three dimensional state space
  - (# jobs, # servers on, # servers in setup)
- In the optimal policy, N servers will always be left on (N may be 0)

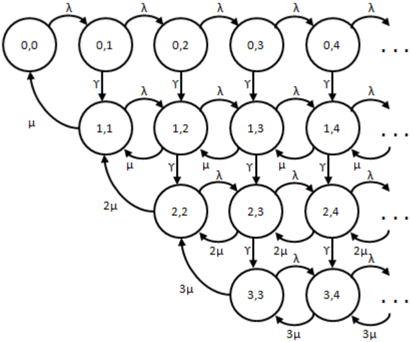
Previous Work

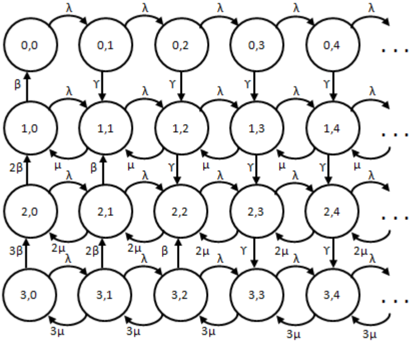
- The optimal policy for a single on/off server system is known ($c = 1$)
  - Steady state and closed form system metrics solved for
- 2 Decision variables



Related Work

- Ghandi et al. applied their RRR technique to specific policies
  - Staggered setup (0 decision variables), Delayed off (1 decision variable)

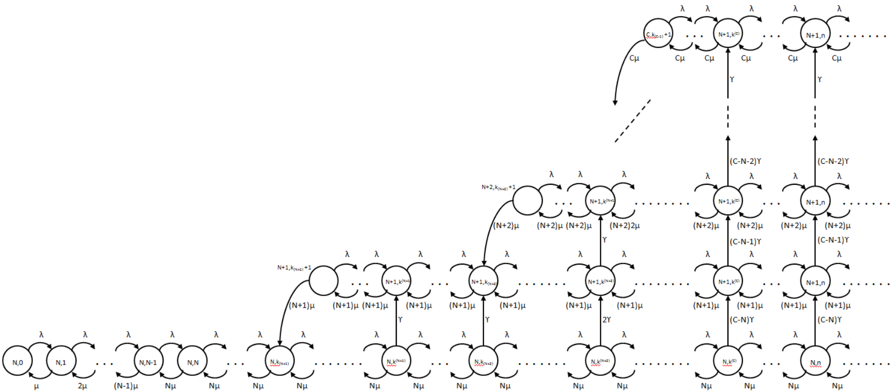




Optimal Policy Decision Variables

- How many decision variables does the optimal policy have with c servers?
- How many jobs need to be in the system before a server starts turning on?
    - What if there is a server already turning on? What if there are two?
- There is a geometric explosion in decision variables
- Same holds for deciding when to turn the servers off

Complex CTMC



Approach

- Generic Markov chains are too complex to analyse
- Examine these systems as MDPs to hopefully find *exploitable* structural properties and patterns
  - Verify them analytically via policy iteration if possible
- Simplify the Markov chain to make them more reasonable to solve
- Verify analytic results against the optimal policy found in the MDP

System as MDP

- $S = \{n_1, n_2, n_3 \mid n_1, n_2, n_3 \geq 0,\ \ n_2 + n_3 \leq c\}$
- $A_{n_1,n_2,n_3} = \{-n_2, -n_2 - 1, ..., -1, 0, 1, ..., c - n_2 - n_3 - 1, c - n_2 - n_3\}$
- Perform uniformization, switching the CTMC to a DTMC relative to the maximum possible rate, and rescale time so the max rate equal 1
  - Done without loss of generality
  - Max rate is $c\max(\mu, \gamma) + \lambda$
- Transition probabilities are easily found from μ, λ, and γ
  - For example: $p((n_1 + 1, 0, n_3 + a)|(n_1, 0, n_3), a) = \lambda$
- $r((n_1, n_2, n_3), a) = n_1 + \beta(min(n_1, n_2) + (n_2 - min(n_1, n_2) r_{Idle} + n_3 r_{Setup})$

Preliminary Observations - Bulk Setup

- When γ sufficiently large (at least on the order of μ and λ) it is often optimal to start turning all available server on once a job arrives which cannot be served
  - One decision variable (N)
  - $\mu = \lambda = \gamma = 1, c = 5, \beta = 3$

# of Server On ->

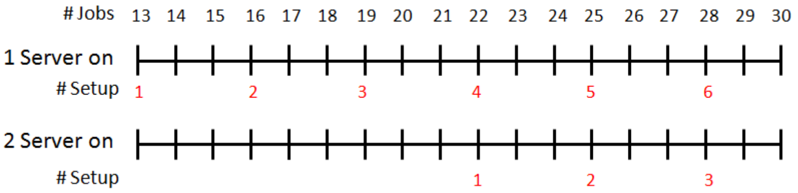
# of Jobs	0	1	2	3	4	5
0	5	0	-1	-2	-3	-4
1	5	4	0	-1	-2	-3
2	5	4	3	0	-1	-2
3	5	4	3	2	0	-1
4	5	4	3	2	1	0
5	5	4	3	2	1	0
6	5	4	3	2	1	0
7	5	4	3	2	1	0
8	5	4	3	2	1	0

Preliminary Observations - k Staggered Setup

- As setup times increase, threshold values increase
- Amount of excess jobs needed before it's optimal to turn the next server on remains approximately constant
    - If $N = 2$ and it's optimal to turn the $(N+1)^{th}$ server on when there's $N + k$ jobs in system, then it is also optimal to turn the $(N+2)^{th}$ server on when there are $N + 2k$ jobs in the system
    - Two decision variables (N, k)

Preliminary Observations - Nested Staggered Setup

- Under most configurations where staggering appears, a nested stagger is a better fit
- Three decision variables (N, k, k_{nested})



Conclusion

- These systems are incredibly complex to analyse the exact optimal policy
  - Number of decision variables gets large with respect to the number of servers
- MDPs reveal certain patterns and structures of the optimal policy
- Analyse these approximation, in hopes of getting closed form solutions
  - Test the models against the optimal policy

[1] A. Gandhi, S. Doroudi, M. Harchol-Balter and A. Scheller-Wolf. "Exact Analysis of the M/M/k/setup Class of Markov Chains via Recursive Renewal Reward." *ACM SIGMETRICS*, 2013.

[2] A. Gandhi, V. Gupta, M. Harchol-Balter, and M. Kozuch. "Optimality Analysis of Energy-Performance Trade-off for Server Farm Management." *Performance Evaluation*, vol. 67, 2010, pp. 1155-1171.

[3] A. Gandhi and M. Harchol-Balter. "M/M/k with Exponential Setup." *CMU Technical Report CMU-CS-09-166*, 2010.

[4] V. J. Maccio. "On Optimal Policies for Energy-Aware Servers" Master's thesis, McMaster University, Hamilton, Ontario, Canada. September 2013

[5] V. J. Maccio, Douglas G. Down. "On Optimal Policies for Energy-Aware Servers", MASCOTS, August, 2013

Questions?

Thank you.