SENG 533 Notes

Brian Pho

April 20, 2018

Contents

Chapter 1

Introduction to Performance Evaluation

- MAT = Maximum Achievable Throughput
- **Problem**: Adding more cores: use of memory increases the number of read cycles, running threads on different CPUs means the thread has to get the data back into the current CPU cache that it's on
- **Solution**: Don't allow the same thread to run on different CPUs or split the workload into the number of CPUs available and only allow them to run on that CPU

Chapter 2

Performance Steps

- Determine system boundaries (what you can and cannot control)
- Determine bottlenecks in the system
- Have a clearly defined goal (E.g. optimize some metric)
- User classes: not all users are the same thus treat them differently
- Use different metrics to better evaluate the metric
- Two types of workload parameters: arrival and max
- Scenario A: | | | | |(1 request per second)
- Scenario B: |||||(1 request per second)
- Even though the average number of requests are the same in both scenarios, the density/distribution are different. Scenario B would be considered a "flash crowd" and cause queuing
- $R_{CPU} = \frac{D_{CPU}}{1-\lambda D_{CPU}}$ where D is the processing cost and R is the response time
- When λ approaches 0, the response time of the CPU approaches the processing time of the CPU.
- A load generator has virtual users and those users send synthetic requests to the system for testing.
- A scaled down system is not the same as a model as a model will abstract away details while a scaled down system still retains such details
- A problem with measurement is that virtual users may not match what real users do
- Sometimes the measurement isn't a measure of the system but of the load generator (if the load generator is the bottleneck)
- When evaluating using measurement, avoid having external aspects of the system as a bottleneck

Chapter 3

Performance-Oriented Review

- RPC: Remote Procedure Call

3.1 Memory

- Latency depends on the startup time of the device
- Latency: x improvement
- Bandwidth (b/w): x^2 improvement
- This translates to writing software that does bulk transfers or caching to improvement performance (to prevent the latency hit)
- Cannot make a very large cache because it isn't cost effective and isn't necessary (spatial and temporal locality)
- Distributed memory makes memory access times variable
- Use prefetching to improve performance

3.2 Processor

- Processors are now multicore so exploit parallelization
- Performance of memory lags behind the performance of processors
- Thread driven processes vs multiple processes: the threaded model is less reliable since their is a single point of failure (the process)
- Talking between processes: pipes / network (messaging) / shared memory
- Critical sections should be treated as a resource (due to queuing)

Chapter 4

Models

- Multiple Resource Multiple Queue: recurrent user (user performs repeated actions)
  - $V_{Processor} = 1 + V_{Disk\ A} + V_{Disk\ B}$
  - $V_{Disk\ A} = P_2 * V_{Processor}$
  - $V_{Disk\ B} = P_3 * V_{Processor}$
  - Where V is the number of visits
  - $D_{Processor} = V_{Processor} * S_{Processor}$
  - $D_{Disk\ A} = V_{Disk\ A} * S_{Disk\ A}$
  - $D_{Disk\ B} = V_{Disk\ B} * S_{Disk\ B}$
  - Where D is the resource demand and S is the service time
- Open Model
  - Workload intensity specified by an arrival rate
  - Unbounded number of customer in the system
  - Throughput is an input parameter
- Closed Model
  - Workload intensity specified by the customer population
  - Bounded number of customers in the system
  - Throughput is an output parameter
  - Closed Model - Batch system: has not think time/terminals
  - Closed Model - Interactive systems: has think time/terminals
- S = service time
- Be careful of assumptions when modelling and watch for tradeoffs

Chapter 5

Operational Laws

5.1 Definitions

- Operational = measurements
- Assumption: Input equals output (arrivals equals completion)
- k = number of resources
- $C_k \neq C$ where C is the number of completions of the entire system and C_k is the number of completions of that resource
  - $\lambda = \frac{A}{T}$ Arrival rate
  - $X = \frac{C}{T}$ Throughput
  - $X_k = \frac{C_k}{T}$ Device/resource throughput
  - $U_k = \frac{B_k}{T}$ Utilization
  - $S_k = \frac{B_k}{C_k}$ Service Time (sec)
  - $V_k = \frac{C_k}{C}$ Number of times a resource is visited
  - $U_k = X * D_k$
  - All of the following equations work for multiple customer class

5.2 Utilization Law

- $U_k = S_k * X_k$
- Useful for verifying measurements using the last equation (Equation is always true)
- Useful for obtaining a variable that you aren't able to measure

5.3 Forced Flow Law

- $X_k = X * V_k$
- Resources are visited many/few times

5.4 Little's Law

- $N = X * R$
- Where N is average number of requests in the system, X is the throughput, and R is the average time spent in the system or response time
- When applied to only the resource without the queue line, it is the same as the utilization law
- $N_{Queue-CPU} = X_{CPU} * (R_{CPU} - S_{CPU})$
- $N_{Queue+CPU} = X_{CPU} * R_{CPU}$

5.5 Utilization Analysis

- $\frac{U_{threshold}}{D_k} = X_{actual}$
- If $X_{actual} < X_{expect}$ (not meeting expectation) then either increase the threshold or decrease the demand
- The demand can be decreased by either decreasing the visits to that resource or by decreasing the service time of that resource
- $D_A = D_{A,CPU} + D_{B,CPU} + D_{C,CPU} + D_{Network}$
- The throughput is still optimistic because it hasn't taken into account the network

5.6 Bottleneck Analysis

5.7 Open System

- Best case R (response time)

  R = D
- Worst case R

  R = (N x D) + D = ∞

5.8 Closed System

- Best case X (throughput)

  $X(1) = \frac{1}{D+Z}$

  $X(2) = \frac{2}{D+Z}$

  $X(N) = min(\frac{N}{D+Z}, \frac{1}{D_{max}})$

- Worse case X

  $X(1) = \frac{1}{D+Z}$

  $X(2) = \frac{2}{2D+Z}$

  $X(N) = \frac{N}{ND+Z}$

5.9 Graphs

- 3 benefits
- D line moves down
- ND1-Z lines move to the right
- ND2-Z line has a less steep slope

5.10 In Class Problem

- Answer - Disk3 is the bottleneck since it has the highest demand. To fix without adding hardware load balance the system by reducing the load on disk3 to 0.12 second (the average of demands on the disk)
- $0.1 * V_{cpu} = 1$ by definition since 10% of the time the visit will exit to the terminal, but when it enters it is always 1. $V_{cpu} = 10$, $V_{net} = 18$, $V_{d1} = 2.7$, $V_{d2} = 6.3$
- $\frac{200{,}000}{10{,}000} = 20 = X_{cpu}$. $\frac{20}{10} = 2 = X$. Little's Law: $2 * 5 = X * R = 10$ This takes into account the disk. $N_{think} = N - N_{[CPU/Disk]} = 40$. $N_{think} = X * Z = 20$ *seconds* where $X = 2$

5.11 Variables

- $C_k - device\ completions$
- $C - system\ completions$
- $S_k - service\ time\ per\ visit$
- $D_k - device\ demand\ (seconds)$
- $X - system\ throughput\ (transactions/second)$
- $X_k - throughput\ (completions/second)$
- $B_k - busy\ time$
- $T - observation\ period$
- $U - utilization\ (max\ value\ is\ 1)$
- $Z - think\ time$

5.12 Equations

- $U_k = X_k * S_k$
- $X_k = X * V_k$
- $U_k = X * D_k$
- $D_k = V_k * S_k$
- $N = XR$
- $N = X(R + Z)$
- $S_k = \frac{B_k}{C_k}$
- $X = \frac{C}{T}$
- $U_k = \frac{B_k}{T}$
- $X_k = \frac{C_k}{T}$

Chapter 6

Mean Value Analysis (MVA)

6.1 Single Class Open Model

- Negative exponential: $P(X = x) = \lambda e^{-\lambda x}$ where X is the service time and x is the individual service time sample. $Mean = Standard\ Deviation = \frac{1}{\lambda}$.
- Useful to built upon.
- Doesn't take into account queue interaction. Allows analysis of system with independent queues.
- X = arrivals per second
- $R_k(\lambda)$ is the response rate as a function of the arrival rate
- $A_k(\lambda)$ is Arrival Instant Queue Length
- A is an inside view of the queue (number of people in front of me) and Q is the outside view of the queue (how many people in queue)
- A may not be equal to Q without the assumptions.

6.2 Single Class Closed Model

- Answer: $R(2) = R(2)_{CPU} + R(2)_{Disk}$
- Calculate X(1), R(1) at each resource to get Q(1)

6.3 Multi Class Open Model

- We want $R_{c,k}(\lambda)$ at the resource k
- We now have a vector of lambda
- $R_{c,k}(\lambda) = (A_{c,k} + 1)$
- $R_{c,k}(\lambda) = (A_{c,k} * S_k + S_k)V_{c,k}$ The problem with this is that we don't know the type of class the people in the queue are

- Assumption: the mean service time is the same regardless of the class type (but visit count can be different thus different demand)
- $R_{c,k} = D_{c,k}(1 + A_{c,k}(\lambda))$
- In class exercise: $R_U = R_{U,CPU} + R_{U,DISK}$ and $R_{U,CPU} = \frac{D_{U,CPU}}{1-U_{CPU}}$
- In an open system, throughput is equal to the arrival rate
- Answer: $(R_{U,CPU} = \frac{2}{1-\frac{7}{19}}) + R_{U,DISK}$

6.4 Multi Class Closed Model

- In class exercise: $Q_{Q,CPU}(0,0) = 0$, $Q_{DISK}(0,0) = 0$. What is $R_Q(2,1)$?
- $R_{Q,CPU}(1,0) = D_{Q,CPU}(1 + Q_{CPU}(0,0)) = 1$
- $R_{Q,DISK}(1,0) = D_{Q,DISK}(1 + Q_{DISK}(0,0)) = 3$
- Need Q before next step. Working towards finding the queue length
- $X_Q(1,0) = \frac{1}{4+0} = 0.25$
- Little's Law: $Q_{Q,CPU}(1,0) = X_Q(1,0) * R_{Q,CPU}(1,0) = 0.25$ do it 3 more times for the disk and second class of customer
- Next step: $R_{U,CPU}(0,1) = D_{U,CPU}(1 + Q_{CPU}(0,0)) = 2$
- $R_U(0,1) = R_{U,CPU}(0,1) + R_{U,DISK}(0,1) = 6$
- $X_U(0,1) = \frac{1}{6}$
- $Q_{CPU}(1,0) = 1/4,\ Q_{DISK}(1,0) = 3/4$
- $Q_{CPU}(0,1) = 1/3,\ Q_{DISK}(0,1) = 2/3$
- $R_{Q,CPU}(1,1) = D_{Q,CPU}[1 + Q_{CPU}(0,1)] = 4/3$
- $R_{Q,DISK}(1,1) = D_{Q,DISK}[1 + Q_{DISK}(0,1)] = 5$
- $R_Q(1,1) = 4/3 + 5 = 19/3$
- $X_Q(1,1) = \frac{1}{19/3} = \frac{3}{19}$
- $Q_{Q,CPU}(1,1) = X_Q(1,1) * R_{Q,CPU}(1,1) = 4/19$
- $Q_{Q,CPU}(1,1) = X_Q(1,1) * R_{Q,CPU}(1,1) = 15/19$
- $R_{U,CPU}(1,1) = D_{U,CPU}[1 + Q_{CPU}(1,0)] = 5/2$
- $R_{U,DISK}(1,1) = D_{U,DISK}[1 + Q_{DISK}(1,0)] = 7$
- $R_U(1,1) = 5/2 + 7 = 19/2$

- $X_U(1,1) = \frac{1}{19/2} = \frac{2}{19}$
- $Q_{CPU}(1,1) = 9/19$
- $Q_{DISK}(1,1) = 29/19$
- $R_{Q,CPU}(2,1) = D_{Q,CPU}[1 + Q_{CPU}(1,1)]$
- $R_{Q,DISK}(2,1) = D_{Q,DISK}[1 + Q_{DISK}(1,1)]$
- $\mathbf{R_Q(2,1) = R_{Q,CPU}(2,1) + R_{Q,DISK}(2,1) = 28/19 + 144/19}$
- Steps: $Q \rightarrow R \rightarrow X \rightarrow Repeat$
- The initial Q is 0. When calculating R for a certain class, subtract 1 from that class of the $Q_{CPU}()$

6.5 In Class Exercise

6.5.1 Question 1

- Check stable with formula in slides (< 1, the utilization of each resource must not exceed 1). Each row multiplied by its respective class lambda must be less than 1.
- $U_{CPU} = \lambda_A D_{A,CPU} + \lambda_B D_{B,CPU} + \lambda_C D_{C,CPU} < 1$
- Answer: Is stable
- $R_A = R_{A,CPU} = \frac{D_{A,CPU}}{(1-U_{CPU})}$
- $R_B = R_{B,CPU} + R_{B,disk1} + R_{B,disk2}$
- $R_C = R_{C,CPU} + R_{C,disk2} + R_{C,disk3}$
- Think time/num of users = closed. Lambda = open

6.5.2 Question 2

- $S_1 = 0.20, S_2 = 0.25, S_3 = 100, N = 2$
- First step: Need to know demand times D_1, D_2, D_3
- By definition, $V_1 = 1, V_2 = 0.2V_1, V_3 = 0.8V_1$
- Then demand is $D = VS$ so $D_1 = 0.2, D_2 = 0.05, D_3 = 0.8$
- Next step: $N = 0 \rightarrow N = 1 \rightarrow N = 2$
- $R_1(1) = D_1[1 + Q_1(0)]$
- $R_2(1) = D_2[1 + Q_2(0)]$
- $R_3(1) = D_3[1 + Q_3(0)]$ Except R_3 doesn't have a queue so $R_3(1) = D_3$

- Question is asking: $Q_2(2) =? = X(2)R_2(2)$
- $R(1) = R_1(1) + R_2(1) + R_3(1)$
- $X(1) = \frac{1}{R(1)}$
- $Q_1(1) = X(1)R_1(1)$

Chapter 7

Quiz + Midterm

- From start to multi class, closed MVA (inclusive). Mon, Feb 26
- Allowed to bring formula sheet (2) (4 sides), no examples. Can define variables.

Chapter 8

MVA with Load Dependent Resources

- Have assumed up to this point that the service time, S_k, is unchanging. However, service time is now a function of the number of customers, $S_k(n)$ where n is the number of customers
- Previously, $R_k(N) = V_k S_k[1 - Q_k(N-1)]$
- Queue length distribution matters because the service time will change depending on the queue length.
- Intuition behind $R_i(n)$ equation: the equation performs a summation over all possibilities given n customers. E.g. n = 3, then either there are 0 customers, 1 of each, or both. Thus $P_i(0|2), P_i(1|2), P_i(2|2)$. You use n - 1 in the formula because you can calculate n given the response time with one customer removed.
- Example

j	1	2	3
$\mu_{CPU}(j)$	$\frac{1}{0.1}$	$\frac{1.8}{0.1}$	$\frac{1.8}{0.1}$

  $V_{CPU} = 1. D_{disk} = 0.06, N = 3, Z = 0, R(3) =?$
- For N = 0, $Q_{disk} = 0, P_{CPU}(0|0) = 1$
- For N = 1, $R_{disk}(1) = D_{disk}[1 + q_{disk}(0)] = 0.06, D_{CPU}(1) = P_{CPU}(0|0) * \frac{1 * V_{CPU}}{\mu_{CPU}(1)} = 0.1$
- $R(1) = R_{disk}(1) + R_{CPU}(1) = 0.16$

Chapter 9

In Class Problems 3

9.1 Question 1

- Start with the simplest technique and then move to more complicated methods
- Could try using Little's law, but we have no R so can't use.
- No R hints at MVA, arrivals per second hints at open system MVA.
- $\lambda_{1-7} = 0.1 * 8L = 0.8L$, $\lambda_8 = 0.3 * 8L = 2.4L$
- $S_{1-8} = \frac{4}{5L}$ Use open, single class MVA to solve
- $V_{1-8} = 1$ Use $Q_{1-7} = \frac{U_{1-7}}{1-U_{1-7}}$
- $U_{1-7} = \lambda_{1-7} * D_{1-7} = 0.8L * (1 * \frac{4}{5L}) = 0.64$
- $Q_{1-7} = \frac{0.64}{1-0.64} = 1.78$
- $U_8 = \lambda_8 * D_8 = 2.4L * 1 * \frac{4}{5L} = 1.92$ Utilization is over 1 thus we know that the arrival rate exceeds what the system can handle, aka the system is unstable. Then the queue link will be infinity.
- The probability of empty queue is one minus utilization (aka when the queue isn't utilized, it's empty) so 36% chance the queues 1 to 7 are empty. Since queue 8 is unstable it has no probabilitiy.

9.2 Question 2

- $R = \frac{D}{1-U}$
- Don't make decisions based on the average
- Ambiguous

9.3 Question 3

- $D_{SRV} = 0.5, V_{SRV} = 20, S_{MOV1} = 50ms, S_{MOV2} = 30ms, X_{MOV1} = 15/sec$
- $V_{MOV1} = 9.5, V_{MOV2} = 9.5$ **from** $1 + 2V_{MOV1/2} = V_{SRV} = 20$
- $X_{SRV} = 31.58, X_{MOV2} = 15$ **from** $X_{MOV2} = X * V_{MOV2} = 1.58 * 9.5$
- $X = 1.58$ **from** $X = \frac{X_{MOV1}}{V_{MOV1}} = \frac{15}{9.5}$
- $U_{SKV} = 0.79, U_{MOV1} = 0.75, U_{MOV2} = 0.45$ **from** $U_k = X_k * S_k$
- b) $Q_{SRU} = 8.5, Q_{MOV1} = 35, Q_{MOV2} = 1.5$
- Response time ambiguous as to over total visits or per visits
- $R_{SRV} = 5.38, R_{MOV1} = 2.21, R_{MOV2} = 1.27$ **from** $R_k = \frac{Q_k}{X_k}(per\ visit)\ or\ \frac{Q_k}{X}(all\ visits)$

Chapter 10

Practice Problems 1

1.

2.

3.

4. $S_{DISK1} = 0.3/32, S_{DISK2} = 0.41/36, S_{DISK3} = 0.54/50$ **from** Utilization Law
   $X = 13680/3600$ **from** completions per seconds
   $V_{DISK1} = 32/3.8, V_{DISK2} = 36/3.8, V_{DISK3=50/3.8}$

5. $R = 16/3.8$ **from** Little's Law

Chapter 11

Modelling Software Contention

- The $D_{client,server}$ is optimistic because it doesn't assume there is queuing at the CPU and disk.
- Try using MVA to solve the software resources (ignore CPU and disk demand). Need N, Z, and D. Have N and Z, but no D (don't have demand of client on server).
- Try using MVA to solve the hardware resources (ignoring clients). Need N, Z, D. Have N and D, but no Z (don't have server think time).
- To compute the software model, you need the output of the hardware model. And to compute the hardware model, you need the output of the software model.
- To solve this issue, make a guess.
- The CPU, DISK, and think time are abstracted away to be "not competing time". It is the time the customer does not interact with the critical section directly. Treat that time as a delay. Not competing = think time + time at devices
- The "not competing" time of the hardware model is when the critical section doesn't bother the CPU/disk, aka the idle time.There should be two "not competing" time, one for the CS, one for the user.
- $U = X * R$ since D = R
- Not needed to memorize MOL algorithm for final

11.0.1 Software Blocking Example

- The decrease the response time is due to having more server threads.
- But the response time flattens out since queuing happens at the remote server now (and at the server's own resources)
- The bottleneck shifts
- The response time of the remote server doesn't change since it only has 1 thread

Chapter 12

Introduction to Discrete Event Simulation

- Not going into details of simulation, more of avoiding common errors with doing simulations.
- What are the state variables that I need to keep of (state variables)
- Do not run only one simulation, run the simulation multiple times to get more confident with our results

Chapter 13

Simulation Input Analysis

-

Chapter 14

Analysis of Results

-

Chapter 15

System Level Performance Models

- Modelling that MVA can't do
- Difference between System-level models and other models

  The previous models have been really detailed, this model only cares about the input-output behaviour
- MVA can't handle rejected requests
- First come up with a notion of state
- The number of states is equal to the minimum of either the system throughput or request arrival rate
- "Birth" rates are the arrival transitions
- "Death" rates are the completion transitions
- Each transition is associated with a rate
- Find the probability of the system being in that state (state probabilities)
- Probability of rejection is equal to the probability of the highest state (E.g. P_3)
- Utilization is equal to $1 - P_0$
- Ignore previous states when calculating probability due to memoryless property
- Can integrate system-level models with MVA
- LQM assumes the system is closed
- Problem with this technique is state space explosion (I.e. too many computations needed)

15.1 Answers to Questions

- Q1: P_3
- Q2: Do weighted average. I.e. multiply each state probability with the number in that state
- Q3: Do weighted average. I.e. multiply each state probability with it's respective throughput
- Q4: Use Little's Law (N = XR)

15.2 In-Class Example

.

- Suppose arrival rate is λ with M threads in the system. Open system
- States = number of requests in the system
- Total Number of States = infinity
- Birth transitions are λ
- Can solve for the throughput using MVA (consider threads + CPU + Disk as closed subsystem)
- Because M is an upper bound, eventually throughput maxes at X(M)

15.3 In Class Practice 4

- Q1. a) Use the CI formula ($x - z_{1-\alpha/2} * s/\sqrt{n}$). Find the z value from the table = 1.96. Answer = [6.432, 9.568]. Samples need to be independent.
- Q1. b) Let $\alpha/2$ be the area to the right of where 10 lies on the z-curve. Let the right side of CI be 10 ($(x + z_{1-\alpha/2} * s/\sqrt{n}) = 10$). Then $z_{1-\alpha/2} = 2.5$. Do the reverse z table lookup so when z = 2.5, then $1 - \alpha/2 = 0.9938$. Then solve for $\alpha/2 = 0.0062$.
- Q2. State space diagram = state transition diagram. Use number of users in the system over the number of CPU cores in use for states because if more users than CPUs, can't model. Open system so infinite number of states. All of the arrows on the top are α, the arrows on the bottom start at μ, then 2μ, then 3μ, up to 4μ, then it can no longer go higher.
- Q3. MVA can't answer this question but advance modelling can. The states are the number of users in the system (1, 2, ..., N). The death rates are the throughput X(1), X(2), ... X(N).The birth rates are N/Z, (N - 1)/Z, ... 1/Z. If there are 0 users in the system, then that means all of them are thinking, the rate that one of them will exit thinking is N/Z. Then get P_k using a system of equation.

15.4 LQM Practice

- Q1: