CSC 2232: Topics in Computer System Performance and Reliability

Bianca Schroeder

Department of Computer Science
University of Toronto

SHORT ANNOUNCEMENT

- Lecture slides from last week are posted on course web page.
- First part of reading list is posted.
  - Volunteer now! :-)
- More project suggestions will be posted soon.

TODAY'S AGENDA

- Complete open versus closed systems
- Queueing Terminology
- First operational laws
- Little's law

PAPER PRESENTATIONS

PAPER PRESENTATIONS

- Each of you will present one paper in class
- Format of presentation:
  - 30 min presentation of paper contents
  - 5-15 min paper review
    - Good points
    - Bad points
  - 10 min class discussion

PURPOSE

- Wrong answers:
  - "To give a verbal version of the paper, cramming all its content into 30 min"
  - "To impress people with your technical depth and thoroughness"
- In fact, no one really cares about these things.
  - The goal is to filter out the main points of the paper and present them well.
  - By the end, everybody in the audience should remember 2-3 take-home messages

The Content

- Same basic topical structure as a paper:
  - Intro to the topic and problem
  - Brief preview of how the talk will progress
  - The solution
  - Evidence that the solution is good
  - Summary of the main points

Detail and time control

- Your main challenges:
  - Their limited time → You have exactly 30 min
  - Their limited attention → More than a little detail will tire and bore them
- These are related: The ways to limit detail also save time

2 main principles

- K.I.S.S. (keep it simple stupid).
- Repeat key insights: tell them what you're going to tell them (Forecast), tell them, and tell them what you told them (Summary).

How many slides?

- At the absolute most, you could have 1 slide per minute; 1 slide per 1.5 minutes is often about right
- This means you have about 20-30 slides over which to get across your message(s)

What's on each slide?

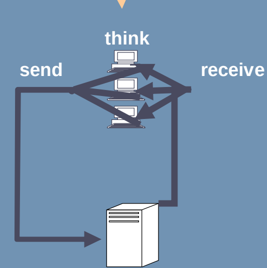
- Each slide should have one basic point
- There should NOT be tons of text
- Use sentence fragments
- Use big fonts
- Use a picture everywhere you possibly can!
  - Saves text and thus slides
  - Much more enjoyable to process

TODAY'S AGENDA

- ⇨ **Complete open versus closed systems**
- **Queueing Terminology**
- **First operational laws**
- **Little's law**

CLOSED SYSTEM MODEL

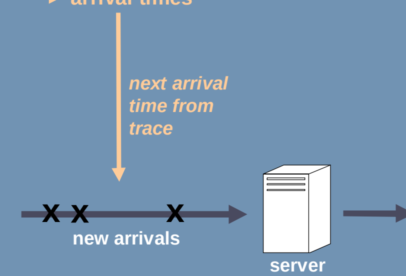
Model of user behavior → **User requests web page, receives page, reads page, clicks on new link**



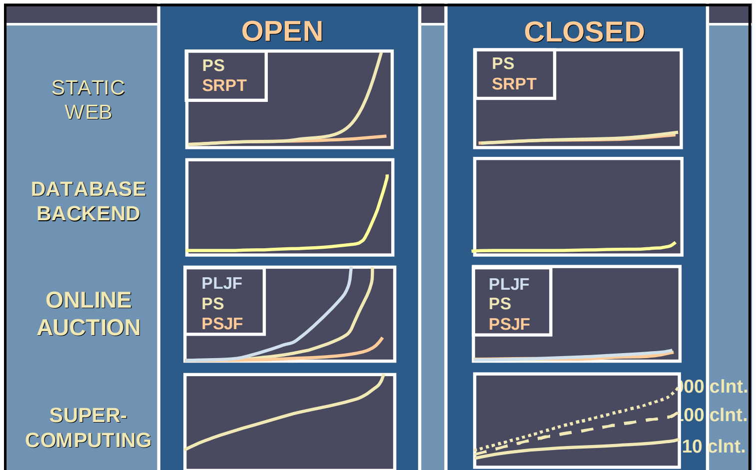
- **Fixed number of users, called the Multi-Programming-Level (MPL)**
- **Arrivals triggered by completions.**

OPEN SYSTEM MODEL

Trace / probability distribution → **arrival times**



- **There is no max number of simultaneous users**
- **Arrivals are independent of completions**

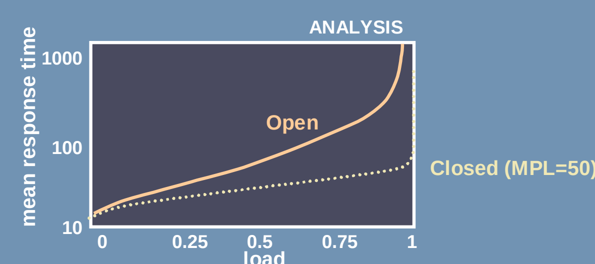


Tuning knob
- Request rate

Tuning knob
-Number of users
-Think time

PRINCIPLES FOR OPEN VS. CLOSED

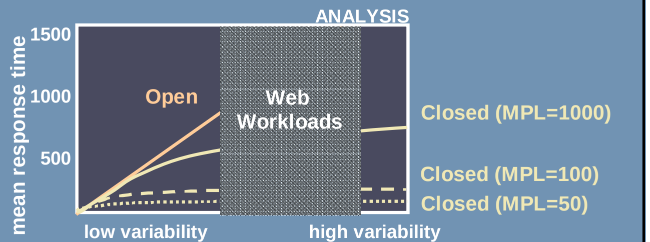
- **What is the magnitude in difference of response times?**
  - **Orders of magnitude!**



- **Why?**
  - ***Bounded number of jobs in closed system.***

PRINCIPLES FOR OPEN VS. CLOSED

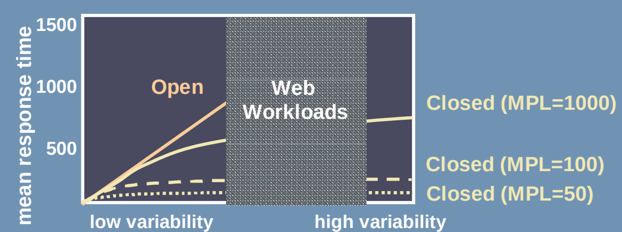
- How does variability affect open/closed response times?
  - Huge effect on open, limited effect on closed system.



- Why?
  - *Dependency between completions and arrivals in closed system reduces burstiness.*

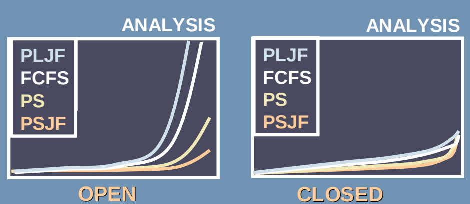
PRINCIPLES FOR OPEN VS. CLOSED

- Can we make closed look like open, by increasing MPL?



PRINCIPLES FOR OPEN VS. CLOSED

- What is the impact of scheduling?
  - Huge in open system, almost none in closed system.

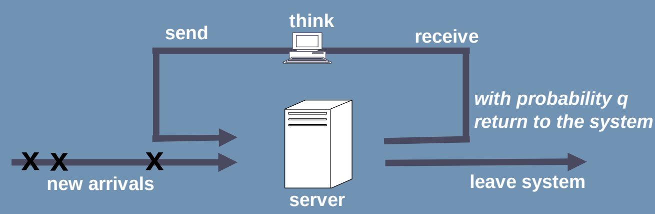


- Why?
  - *Scheduling takes advantage of variability in the system.*
  - *Closed systems reduce the effect of variability.*

HOW DO YOU CHOOSE THE RIGHT MODEL?

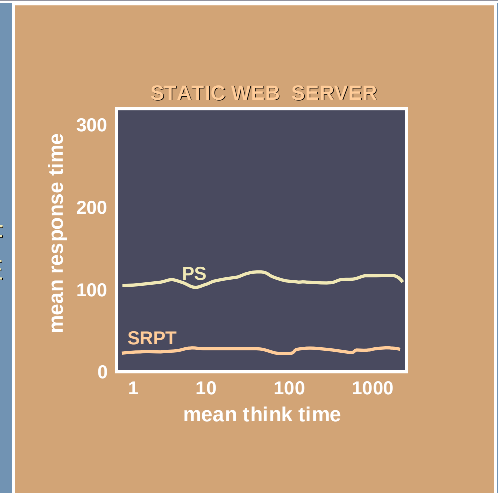
1. Is there a more realistic model?
1. What's most representative of real systems?

THE PARTLY-OPEN MODEL

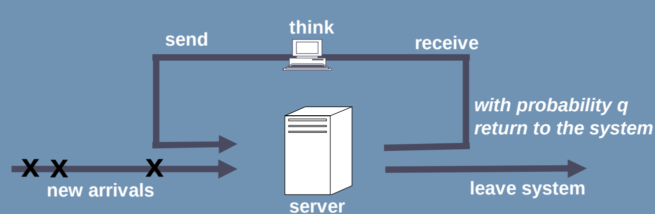


What parameters affect the load?
Does think time affect the load?
How do think times affect response times?

THE EFFECT OF THINK TIME

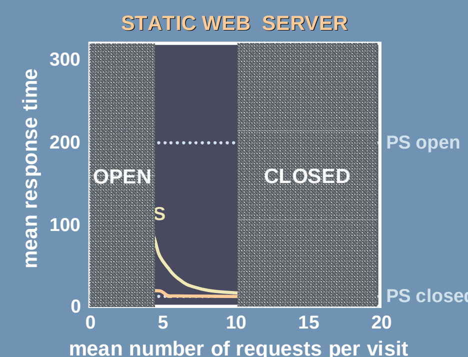


THE PARTLY-OPEN MODEL



How does this model compare to Open/Closed?

THE TRANSITION FROM OPEN → CLOSED



CHOOSING A SYSTEM MODEL

Real web workloads #req. / visit

- A site being "Slashdotted" (1.2)
- Financial service provider (1.4)
- CMU web server (1.8)
- Kasparov vs Deep Blue (2.4)
- Large corporate web site (2.4)
- Science Institute USGS (3.6)
- Online dept. store (5.4)
- Supercomp. site (6.0)
- World cup site (11.6)
- Online gaming site (12.9)

Open or Closed?

Use partly-open system to decide

PARTLY-OPEN

CLOSED

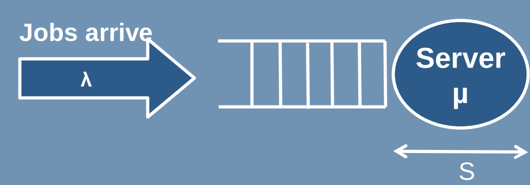
Motivating examples: conclusion

- Intuition is not always good enough
  - Need back-of-the envelope calculations and analytical tools to answer questions.
- Workload / fault load matters hugely
  - Important to understand what the real world looks like!

TODAY'S AGENDA

- Complete open versus closed systems
- Queueing Terminology
- First operational laws
- Little's law

THE SINGLE SERVER NETWORK



- Service Order: FCFS (First-Come-First-Serve)
- Average arrival rate: λ = 1 job / 6 sec
- Mean interarrival time: $1/\lambda$ = 6 sec
- Service Requirement S = 5 sec
- Mean service time: E[S] = 3 sec
- Average service rate: $\mu = 1/E[S]$ = 1 job / 3 sec

HOW IS THIS DIFFERENT FROM NORMAL CONCERSATIONS?

- Normal conversation:
  - The average arrival rate is 3 jobs / second
  - The CPU can provide 20,000 cycles per second
  - Jobs need on average 5,000 cycles to complete
- Translation into queueing lingo :

λ = 3 jobs / second

E[S] = 0.25 seconds

μ = 4 jobs / second (= 1/E[S])

PERFORMANCE METRICS

- Response time: T_S
  - Turnaround time, flow time, time in system
  - Main interest: E[T], also Var[T]
- Waiting time: T_Q
  - Time in queue
- Number of jobs in system: N_S
- Number of jobs in queue: N_Q

SOME OBSERVATIONS

Metrics:
T_S
N_S
T_Q
N_Q

- As λ increases:
  - All metrics increase
- As μ increases:
  - All metrics decrease
- Basic requirement: $\lambda \leq \mu$
  - What happens if $\lambda > \mu$? (Hint: what is E[Ns at time T])

MORE OBSERVATIONS

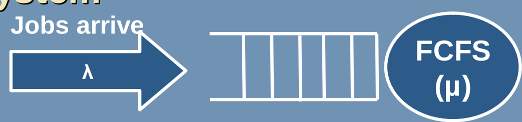
Metrics:
T_S
N_S
T_Q
N_Q

- Suppose $\lambda \leq \mu$ and interarrival distribution and service time distribution are constant:
  - What is T_Q ?
  - What is T_S ?
- So queueing comes from variability
  - So how does variability lead to queues?
  - E.g. assume λ = 1 job/sec and μ = 2 jobs /sec

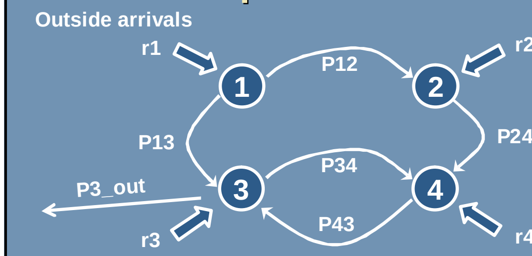
CLASSIFICATION OF QUEUEING NETWORKS

- OPEN versus CLOSED
- Let's start with OPEN:
  - Has external arrivals and departures

EXAMPLE 1: The single server system

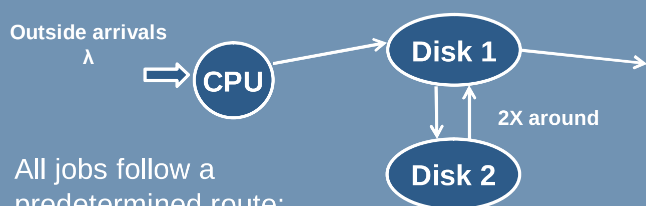


EXAMPLE 2: Network-of-queues with probabilistic routing



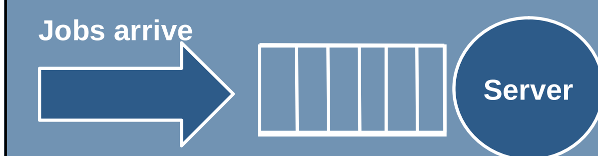
- Server *i* receives
  - external arrivals at rate r_i
  - and also internal arrivals from other servers
- Jobs finishing at server *i*, go to server *j* with prob. Pij

EXAMPLE 3: Network-of-queues with non-probabilistic routing



- All jobs follow a predetermined route:
  - CPU
  - Disk 1
  - Disk 2
  - Disk 1
  - Disk 2
  - Disk 1

EXAMPLE 4: Finite buffer

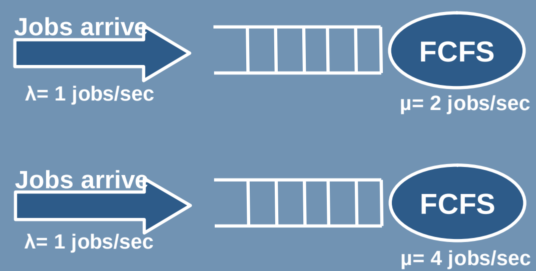


- Space in queue is limited to n jobs
- Arrivals that find no room are dropped

TWO MORE PERFORMANCE METRICS

Metrics:
T_S
N_S
T_Q
N_Q

- Throughput X
- Utilization U



TWO MORE PERFORMANCE METRICS

Metrics:
T_S
N_S
T_Q
N_Q

- Definitions:
- U_i: Fraction of time device i is busy
- X_i: The rate of completions at device i (in jobs/ sec)
- How does X_i relate to U_i?

HOW DOES X_I RELATE TO U_I

- Suppose we observe the system for T time units and see that
  - Device i is busy for B time units
  - C jobs are completed
  - U_i = B/T X_i = C/T
  - => X_i = C/(B/U_i) = C/B * U_i
  - What is C/B?
    - B/C = E[S] => C/B = 1/E[S] = μ

X_i = μ * U_i

HOW DOES X_I RELATE TO U_I

X_i = Mean rate of

= E{Rate of comp

P{server busy}

\+ E{Rate of compl

P{server is idle}

= μ * U_i

How does this change with changes in
- Job size distribution
- Interarrival time distribution
- Service order

X_i = μ * U_i

THE UTILIZATION LAW

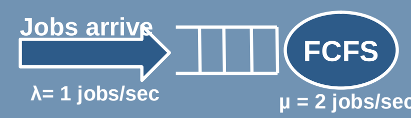
$X\_i = \mu * U\_i$

or equivalently

$U\_i = X\_i * E[S]$

WHAT IS THROUGHPUT?

$X\_i = \mu * U\_i$



- What is U?
  - Fraction of time server is bu[sy]
  - = mean service time / mean tim[e b]etw. Arrivals
  - = (1/μ)/(1/λ) = λ/μ

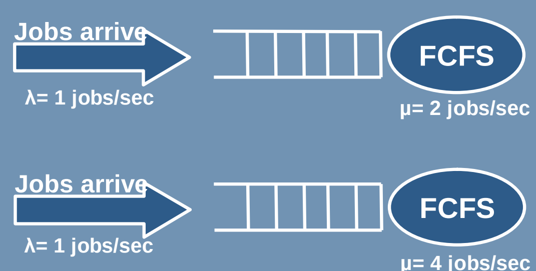
Throughput does not depend on service rate!

$X\_i = \lambda$

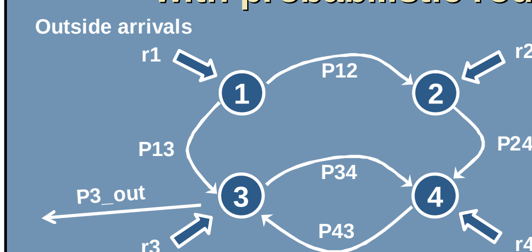
BACK TO OUR OLD EXAMPLE:

Metrics:
T_S
N_S
T_Q
N_Q
X
U

$X\_i = \mu * U\_i = \lambda$

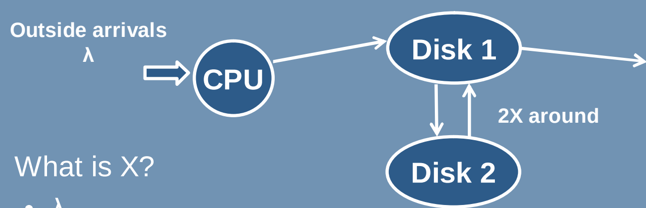


EXAMPLE 2: Network-of-queues with probabilistic routing



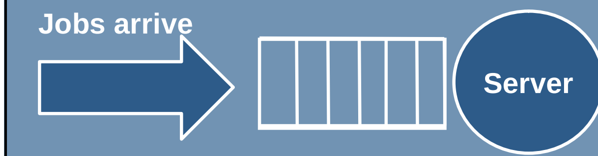
- What is X?
  - ∑ r_i
- What is X_i?
  - $X\_i = \lambda\_i = r\_i + \sum \lambda\_j * Pji$

EXAMPLE 3: Network-of-queues with non-probabilistic routing



- What is X?
  - λ
- What is X_disk1
  - 3λ
- What is X_disk2
  - 2λ

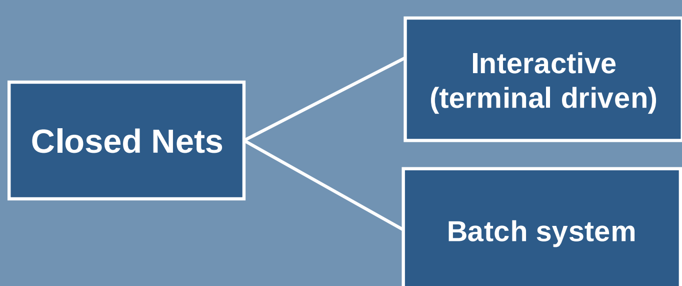
EXAMPLE 4: Finite buffer



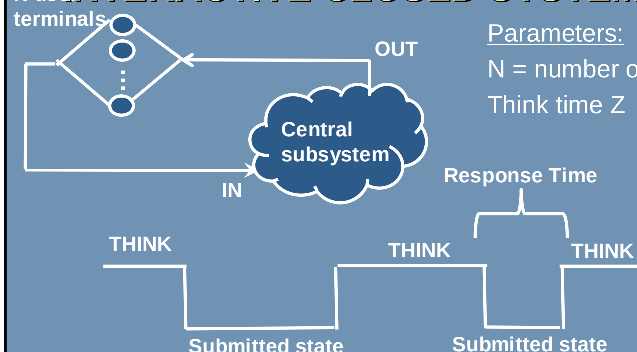
- Space in queue is limited to n jobs
- What is X?
  - X = U*μ
- But U is no longer λ/μ
- Need Markov chains …

CLOSED SYSTEMS

- Closed networks have no external arrivals or departures



INTERACTIVE CLOSED SYSTEMS



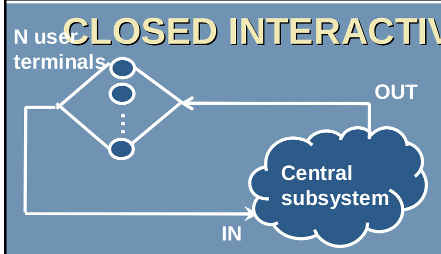
Parameters:

N = number of terminals

Think time Z

- How is response time defined?
  - Time it takes a job to go from IN to OUT

CLOSED INTERACTIVE SYSTEMS



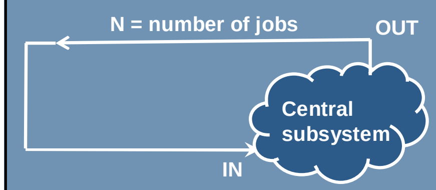
Parameters:

N = number of terminals

Think time Z

- Examples?
- Typical goals:
  - How high can we make N while keeping response times reasonably low?
  - Given a fixed N, what changes to central subsystem will improve response time the most?

CLOSED BATCH SYSTEMS



Parameters:

N = number of jobs

- As soon as one job completes, another one is started
- So there's always exactly N jobs in central subsystem
- Typical Goal?
  - Throughput!
- How is throughput defined?