Queing part.

LECTURE 8

**What is m/m/k?**

Poisson arrivals = exponentioal distribution inter-arrivals

1 queue, k servers.

Utilization for this system = lambda / (mu \*k)

Use experiments to disapprove the theory.

Is a K-small server better than 1 big server.

**Scheduling policy?**FCFS and Last come Last Serve.

round robin, smallest job first(you need job size)  
shortest remaining time

Linux used by policy? Processor sharing. Most insensitive to job size.

Tell me your epoch, then I can estimate how much time you need.

Stuck with 1 big job, or stuck with many small jobs.

Avoid starvation.

**“load balancing” is important:**

If one server is vey loaded, it is very problematic: heat, server outage.

Provider can not peek into application, due to confidentiality.

Make sure it is not not overheated, temperature has to maintin a certain range, no extra pumping cold air.

If you have a hot spot is very difficult.

Server is overheated, it burns out 🡪 reliability issues.

Overheated 🡪 super slow

Effective balancing depends on scheduling:

Random is the worst policy.

Then round robin, join shortest queue, least work left, central queue, size interval splitting

(the last 3 is hard to track the state and implement).

What low balancing polocy would be the winner?

**shortest-queue,** is the best when we use processor sharing.

This week is Data Science Track

Thesis: self-driving cars

Optimization courses are very important to ML.

Collect Data 🡪 feature engineering 🡪 standard ML 🡪 deep ML

Which features is the most important to predict? But you can also use domain knowledge.

Standard ML and deep ML difference?

Deep: hard to explain, more than 2 layers

Standard: easy to explain

How does ML use in system?

When the server is going to fail.

Collect utilization of clusters of years, then build a prediction model.

If you do Software Engineering, you collect lots of Logs.

How much computation overhead.

Worked for IBM.

Random Forest is very simple to use, powerful, quick, no need much data.

Try benchmark against Random Forest.

Do you need linear boundaries.

Decision Tree.

When you go deep in a Deceision tree: bias decrease, and variance increase.

**Homework Tip**

1) average number of jobs in system

2) use little to find average response time

Service time increase by x2, service times increases more than 2x

Last part of lec 8.

Keepresponse time constant, the arrival increase x4, then service rate must decrease lss then 4

Processor sharing is INSENSITIVE

Lower response tive when load is low?

A single server with faster service rate is better than a smaller one.

There is a point.

If service time has High variance and load is medium? Many small serves or big servers

Load is rho = lambda/mu.

Tandem queue?

Lambda greater than mu1 and mu2