

Introduction to Simulation

Queueing Systems

Graham Horton

Motivation & Goals

Many real systems can be modelled as networks of queues

Queueing systems have been well studied

They are often used in discrete simulations

Goals:

- Introduce basic queueing terminology
- Introduce some basic queueing strategies

Queueing Systems

Queueing systems are a class of conceptual models

They have been studied intensively

There are many theoretical results available

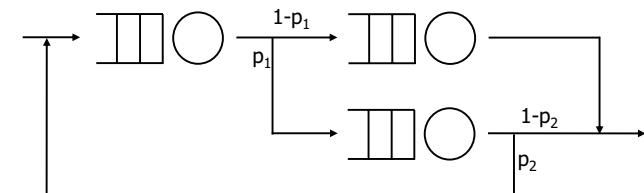
They are used widely for modelling, especially in

- Communications systems
- Networks
- Manufacturing
- Logistics

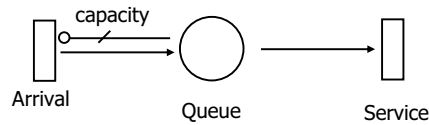
Graphical Notation

Standard graphical notation for queues:

- Servers as circles
- Queues as open rectangles with "slots"
- Paths as arrows
- Probabilities as annotations



We can represent a simple queue by an SPN:



BUT...

- Petri nets do not distinguish between tokens
- Therefore queueing strategies cannot be represented

A standard notation for queueing systems

$A / B / c / k / m - Z$

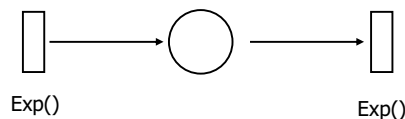
Where...

- A: Type of arrival process (M, G, D, ...)
- B: Type of service process (M, G, D, ...)
- c: number of servers
- k: capacity of queue + server(s) (default ∞)
- m: total job population (default ∞)
- Z: scheduling discipline/strategy (default FIFO)

Example:

M/M/1:

- Markovian (i.e. exponential) arrivals
- Markovian (i.e. exponential) service times
- One server
- Infinite capacity, infinite population



Example:

G/D/2/10 - SJF :

- Generally distributed arrivals
- Deterministic service times
- Two servers
- Queue capacity of 10 jobs, infinite population
- Shortest job first (SJF) queueing strategy

The M/M/1 Queue

The M/M/1 queue is the simplest of all

- Several analytical results are available
- (Queueing Theory is a branch of Computer Science)

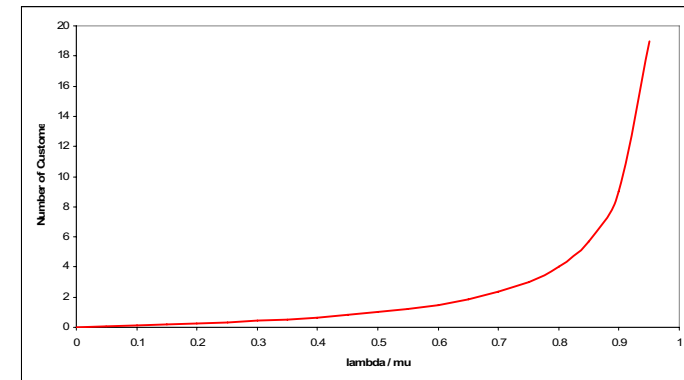
Define

- λ : Parameter of the (exponential) arrival intervals
- μ : Parameter of the (exponential) service intervals

Then:

- Mean # customers in the queue = $\lambda / (\mu - \lambda)$
- Total waiting time = $1 / (\mu - \lambda)$

The M/M/1 Queue



Little's Theorem

The most important theoretical result for queues

Little's Theorem:

- Given
 - Average arrival rate λ ,
 - Average time spent in queue T,
 - Average number of jobs in queue N

Then

$$N = \lambda \cdot T$$

Queueing Strategies

A queueing strategy is a rule for ordering jobs in a queue

There are two main categories of strategies:

- Static or dynamic priorities
- Pre-emptive or non-pre-emptive

		Priorities	
		Static	Dynamic
Pre-emptive?	No	Dentist	Moderated discussion
	Yes	Super-market	Office work

Goals of a queueing strategy:

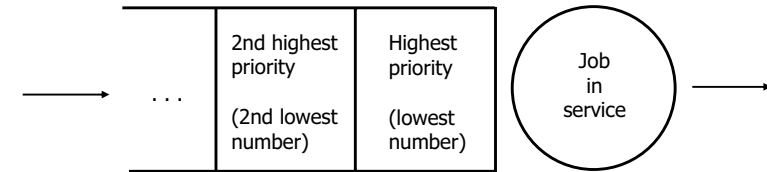
- Fairness
- Maximise throughput
- Minimise overhead
- Minimise waiting time
- Avoid infinite postponement
- Graceful degradation
- Enforce priorities

Different situations call for different strategies!

Some queueing policies use priorities

- Each job is assigned a priority
- The jobs are sorted in the queue by their priority

Low values represent high priorities



Priorities can be dynamic, i.e. change their values

Priorities are a measure of urgency or importance

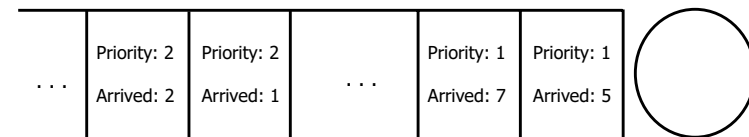
Examples:

- Jobs for important customers (Two-class health treatment)
- Jobs that earn more money (Luxury versus standard products)
- Jobs with close deadlines (Passenger whose plane is about to leave)
- Jobs that have a special function (System processes in a computer)

Usually, there is only a small number of priorities
Secondary criterion is used to sort jobs of equal priority

Example:

- Order according to priority, then FIFO



Pre-emption:

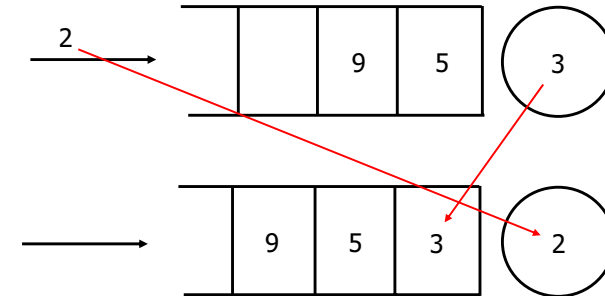
- A job being served can be interrupted by another

Examples:

- High-priority jobs
- Emergencies
- Telephone calls
- Supermarket (sometimes!)

In non-pre-emptive queues, a job that has started service will remain in service until complete

A new job with a high priority (2) pre-empts the job currently being processed (3)



Some well-known strategies are:

- First come, first served (First in first out, FIFO)
- Last In, First Out (LIFO)
- Shortest Job First (SJF)
- Round Robin
- Upper Time Limit (deadline)
- Shortest remaining time

Strategies can also be combined

Jobs are ordered according to their arrival times

Examples:

- Most systems where people or vehicles queue

Non-preemptive strategy without priorities

Advantages:

- Fairness, low overhead, no infinite postponement

Disadvantage:

- Long jobs make short jobs wait

Simple, but often not good enough

- A hospital organised by FIFO would not be good!

Shortest Job First

Jobs are ordered by smallest estimated processing time

Non-pre-emptive strategy without priorities

Waiting time is more unpredictable than FIFO

Problem:

- Requires a priori knowledge of processing time

Upside:

- Short jobs are finished quickly

Downside:

- Large jobs suffer at the expense of shorter jobs
- Danger of infinite postponement

Round Robin

Idea:

- Each job is given a "time-slice"
- When the time-slice has expired, the job is pre-empted and is put back in the queue

Examples:

- Processes in a computer CPU
- Patients at the dentist

Disadvantage:

- Overhead for changing jobs

Advantage:

- Fairness

Not a good idea at a supermarket checkout!

Deadline Scheduling

Assumptions:

- Each job must be completed by a specific time
- Processing time must be known in advance

Idea:

- Run the job whose deadline is closest

Advantage:

- Ensures that "emergency" jobs are finished on time

Disadvantage:

- Unfair to jobs with long-term deadlines

Not a good idea at a supermarket checkout!

Mixed Strategies

Queueing strategies may be combined

Example: Dentist

- Three priority classes:
"Privat", "Kassenpatient" and "emergency"
- Each priority class is FIFO
- Round Robin treatment of three patients
- Emergencies may also pre-empt

Queueing strategies may be combined

Example: Computer system

- All processes have static priorities
- Important system processes (swap) have high priorities
- User jobs have medium priorities
- Non-critical system processes (backup) have low priorities
- Each priority class is treated in a Round Robin fashion
- Interrupts can pre-empt

Queueing strategies may be combined

Example: Secretary

- Different paperwork priorities:
"Boss", "Other people", "E-Mail"
- Different urgency priorities:
"Today", "This week", "Sometime"
- Sort according to urgency first
- Telephone pre-empts all other jobs