POLITECNICO DI MILANO

Computing Infrastructures





Performance modeling

Prof. Danilo Ardagna

Credits: Jane Hilston, Moreno Marzolla, Raffaela Mirandola, Marco Gribaudo, John Zahorian, Ed Lazowska

Reference book

Quantitative System Performance
Computer System Analysis Using Queueing Network Models

Edward D. Lazowska, John Zahorjan,
G. Scott Graham, Kenneth C. Sevcik

Available on-line: http://homes.cs.washington.edu/~lazowska/qsp/

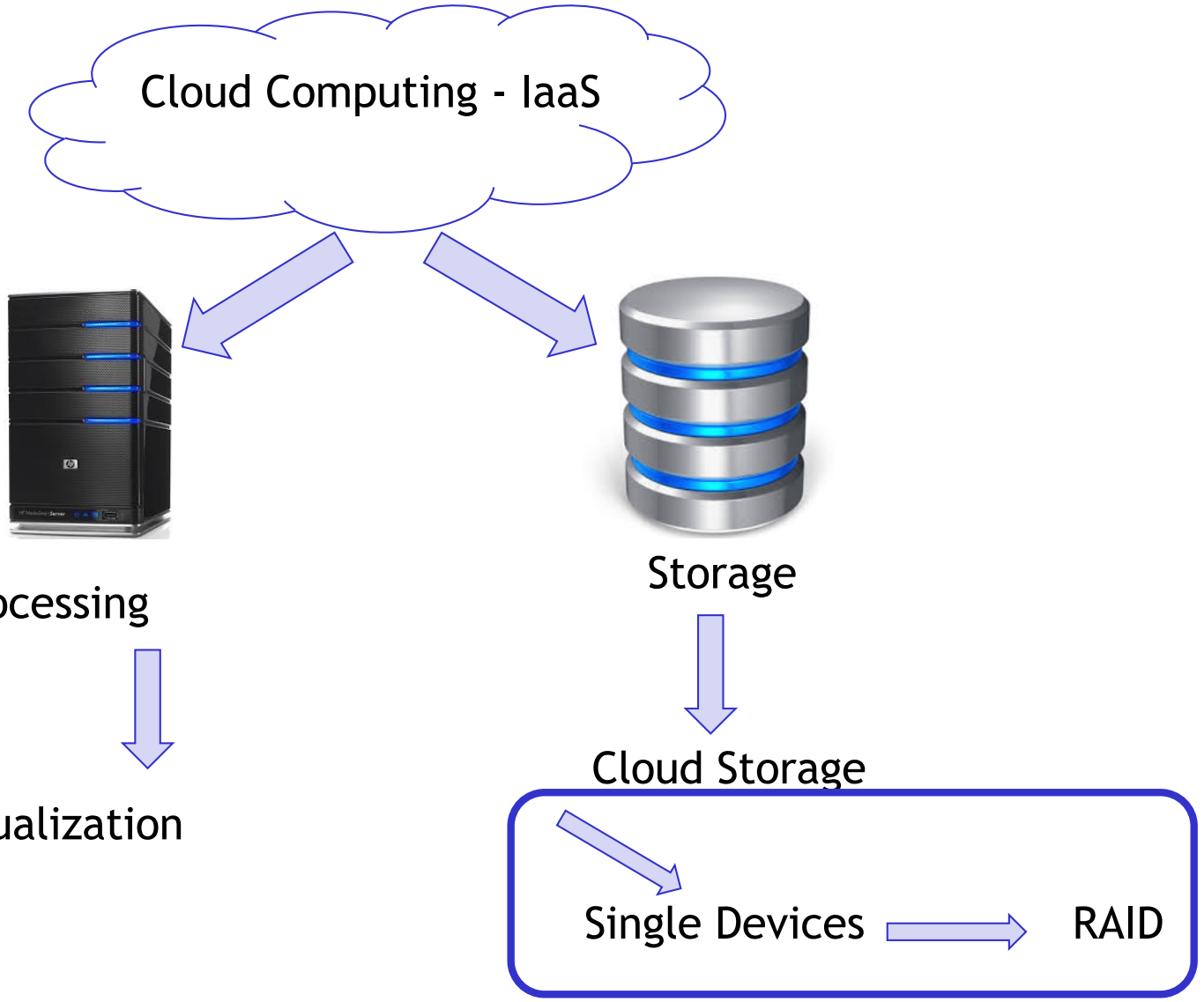
Big Picture

Performance

Average response time
Throughput

Dependability

MTTDL



Definition: Performance

- **Computer performance:**
  - The total effectiveness of a computer system, including throughput, individual response time and availability
  - Can be characterized by the amount of useful work accomplished by a computer system or computer network compared to the time and resources used

Examples of observed performance indices

- Response time of an application

- Service time of a resource
- Resource utilization

- Loss rate of messages on a physical communication channel

- Bandwidth utilization on a wireless channel
- Blocking probability on a wireless channel

System quality: a central issue

- Common practice: system mostly validated versus “functional” requirements rather than versus quality ones
- Different (and often not available) skills required for quality verification
- Short time to market, i.e. quickly available products and infrastructures “seem” to be more attractive nowadays!
- Little information related to quality is usually available early in the system lifecycle because related decisions are taken later

System Quality Engineering

Systematic quantitative approach to constructing systems that meet quality objectives

… based on the methodical assessment of quality (performance and dependability) issues from design to implementation/operation and maintenance

How?

- Use of intuition and trend extrapolation
  - Unfortunately, those who possess these qualities in sufficient quantity are rare
- Pro: rapid and flexible
- Con: accuracy

- Experimental evaluation of alternatives
  - Experimentation is always valuable, often required, and sometimes the approach of choice
  - It also is expensive - often prohibitively so
  - A further drawback: an experiment is likely to yield accurate knowledge of system behavior under one set of assumptions, but not any insight that would allow generalization
- Pro: excellent accuracy
- Con: laborious and inflexible

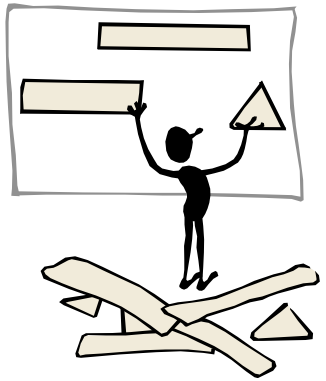
Solution: Model-Based Approach

Systems are complex so…

Abstraction of the systems: Models

"an attempt to distill, from the details of the system, exactly those aspects that are essentials to the system behavior"....
(E. Lazoswka)

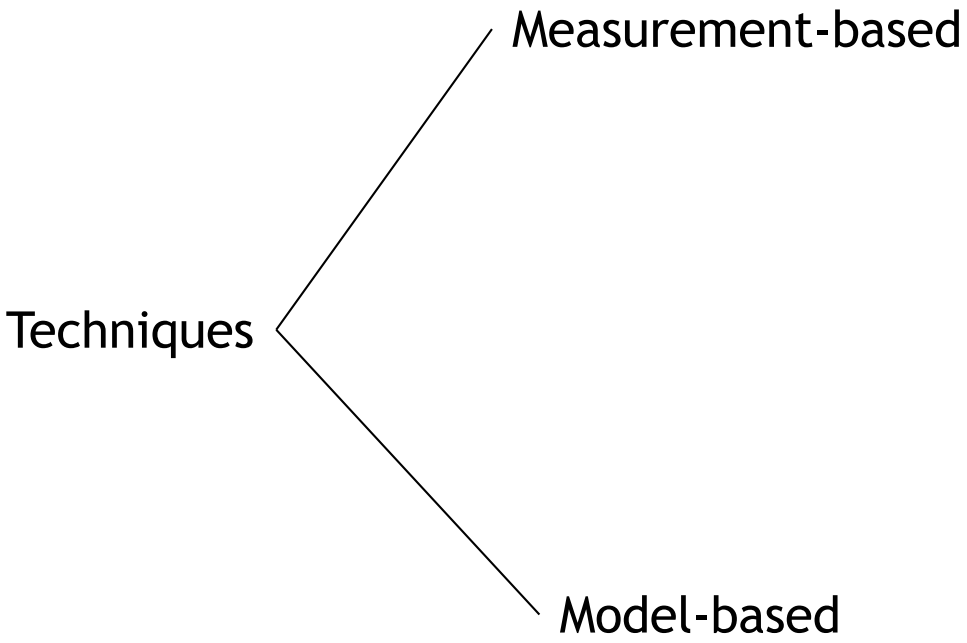
Often models are the only artifact to deal with!
e.g., design phase



Models used to *drive* design decisions

- Which architecture ?
- How many resources to meet some performance/reliability goal?
- ...

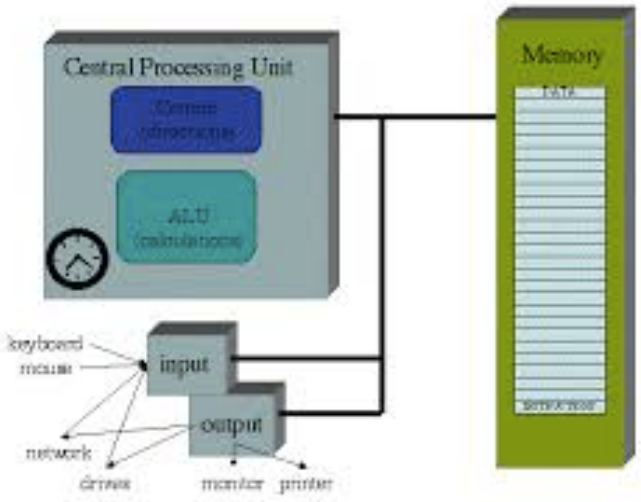
Quality Evaluation techniques

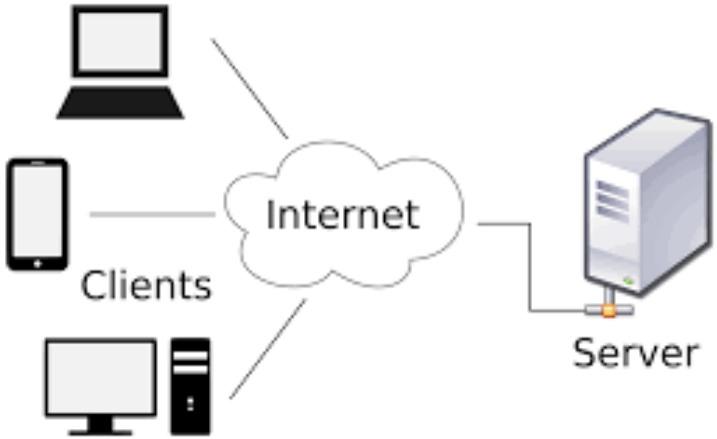


What is a model?

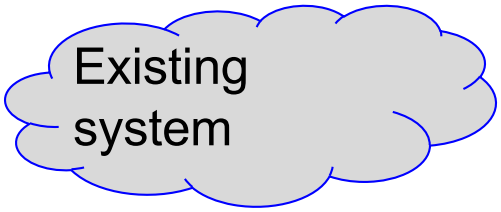
A representation of a system that is

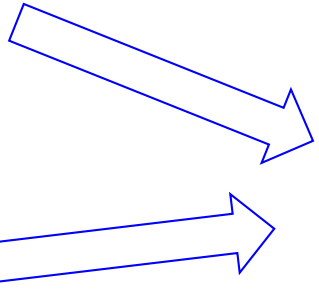
- *simpler* than the actual system
- *captures the essential* characteristics
- *can be evaluated to make* ***predictions***





Model Definition

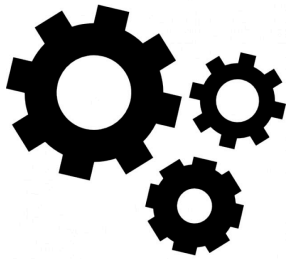




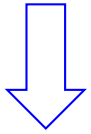
Parametrize the Model

Define the goal

Create the Model

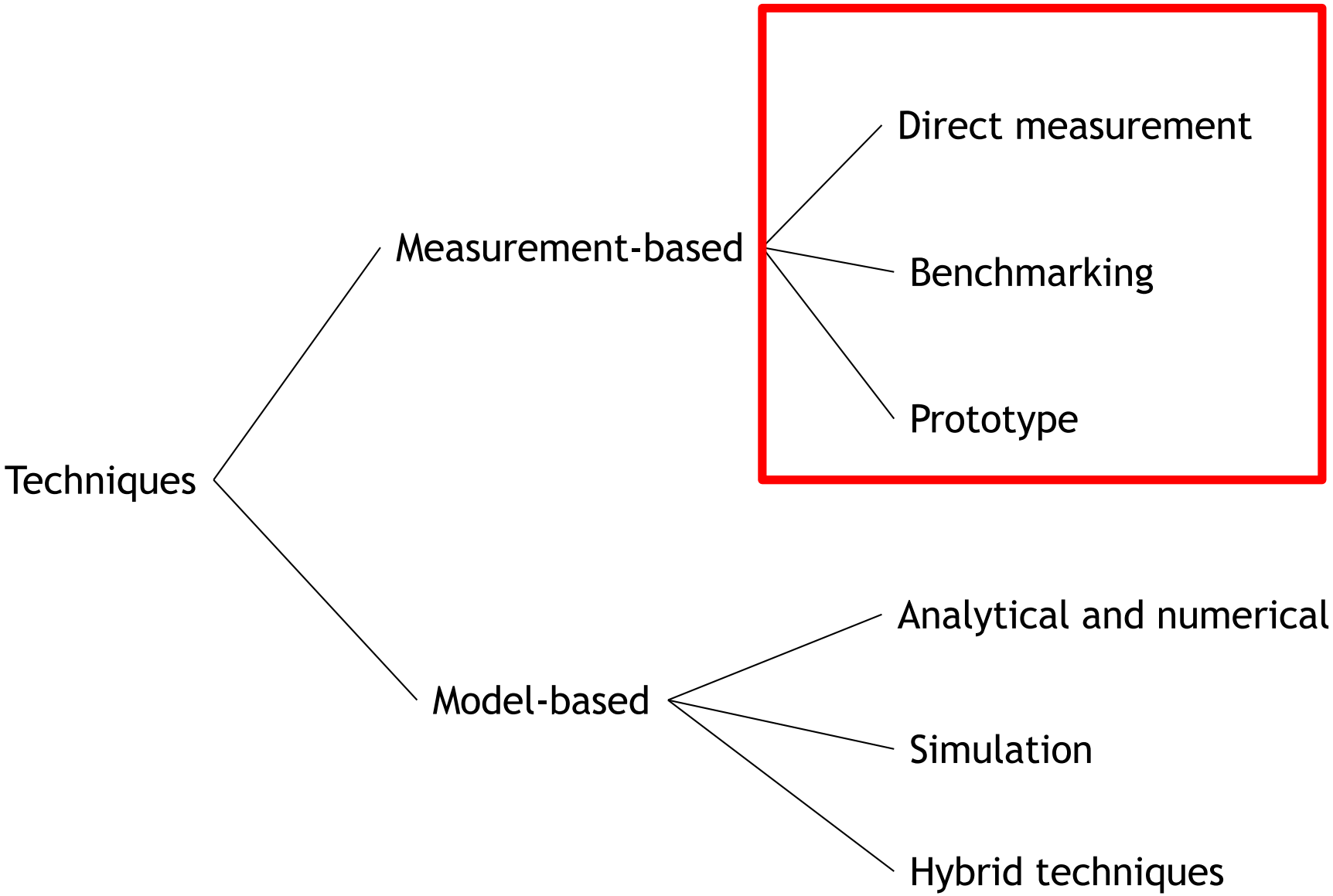


Evaluate the Model



Performance indices

Quality Evaluation techniques

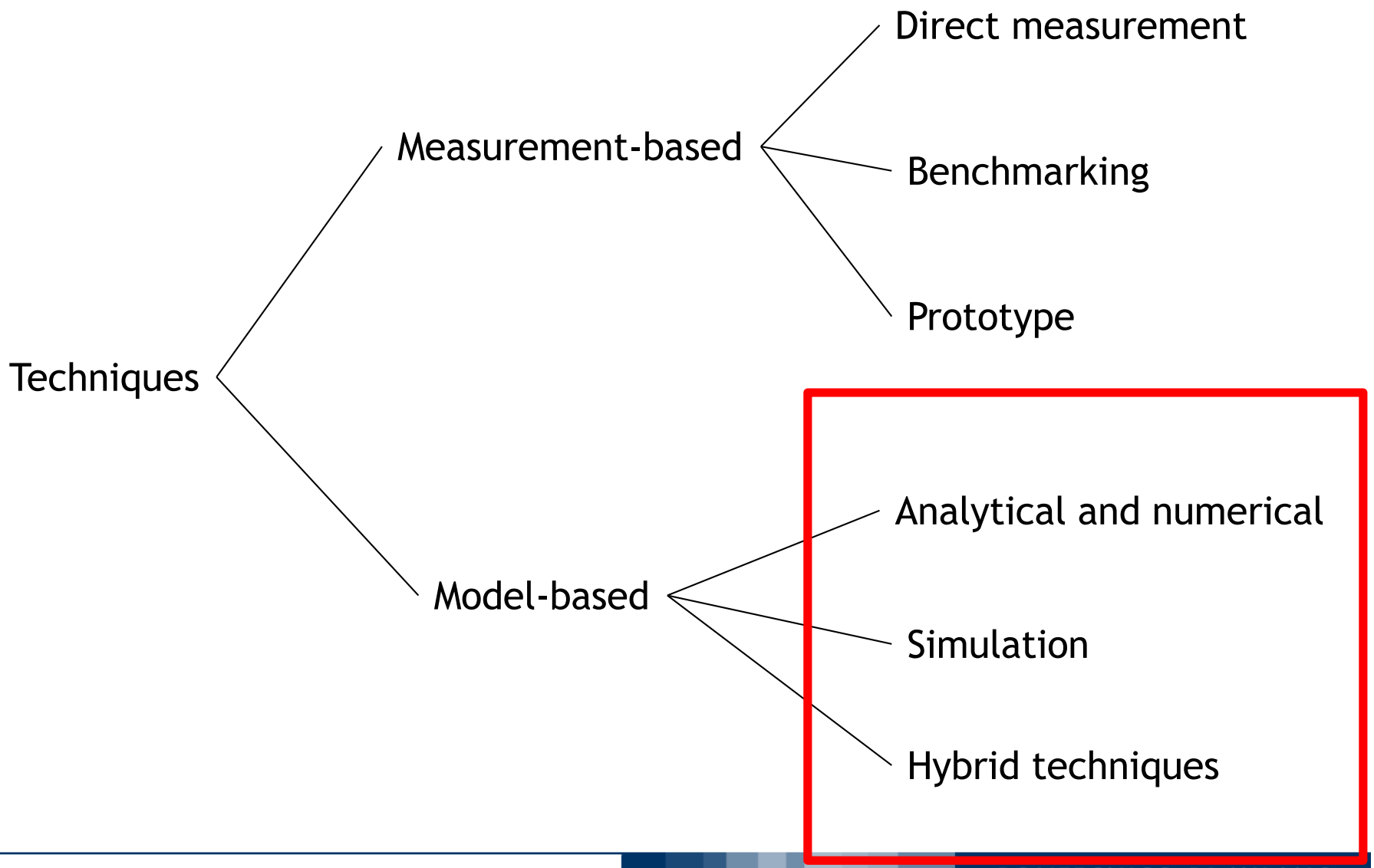


Measurement techniques

- Direct measurement
  - system behavior is measured using its real workload using appropriate instrumentation and techniques
- Benchmarking (measurement with artificial workload)
  - system behavior is measured using an artificial workload or **benchmark** (experiments can be repeated, measures can be compared,…)
- Prototyping (system does not exist)
  - main disadvantage: potentially high cost

Low level of flexibility for what-if-analysis

Quality Evaluation techniques



Model solution techniques

- **Analytical and numerical** techniques are both based on the application of mathematical techniques, which usually exploit results coming from the theory of probability and stochastic process

- In both cases, the model is converted into a set of equations, and performance indices are computed starting from their solution
  - If the equations can be solved in closed form (i.e., a formula or an exact algorithm to compute them can be derived), the techniques are said to be analytical
  - If the solutions of the equations can only be approximated by suitable numerical procedures, the techniques are said to be numerical

Analytical techniques: examples

The average number of jobs in a system is equal to the product of the throughput of the system and the average time spent in that system by a job

Little Law:

$$N=X*R$$

Utilization Law:

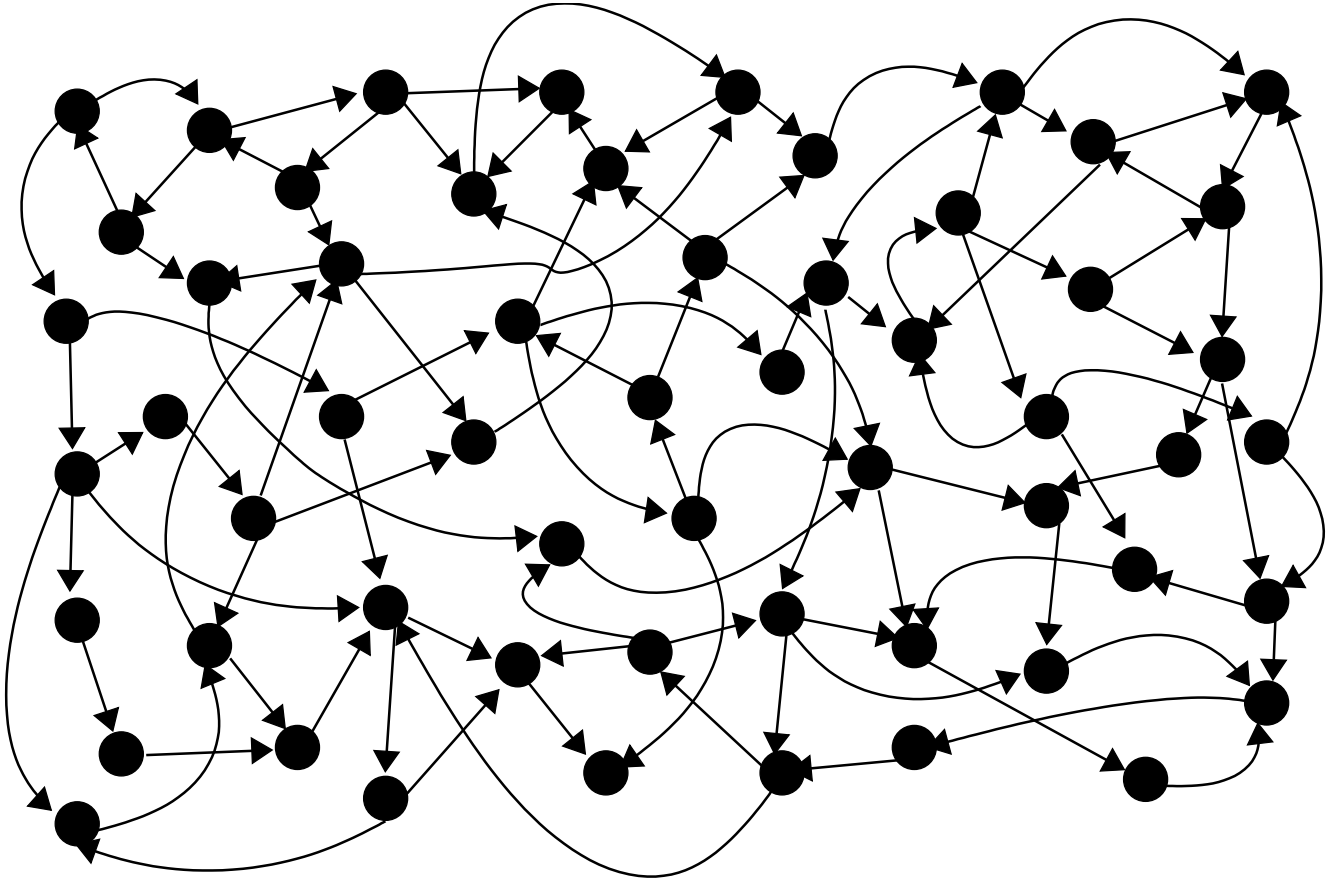
$$U_k=X*D_k$$

The average utilization of a service center is equal to the product of the throughput of the system and the average service demand at the center

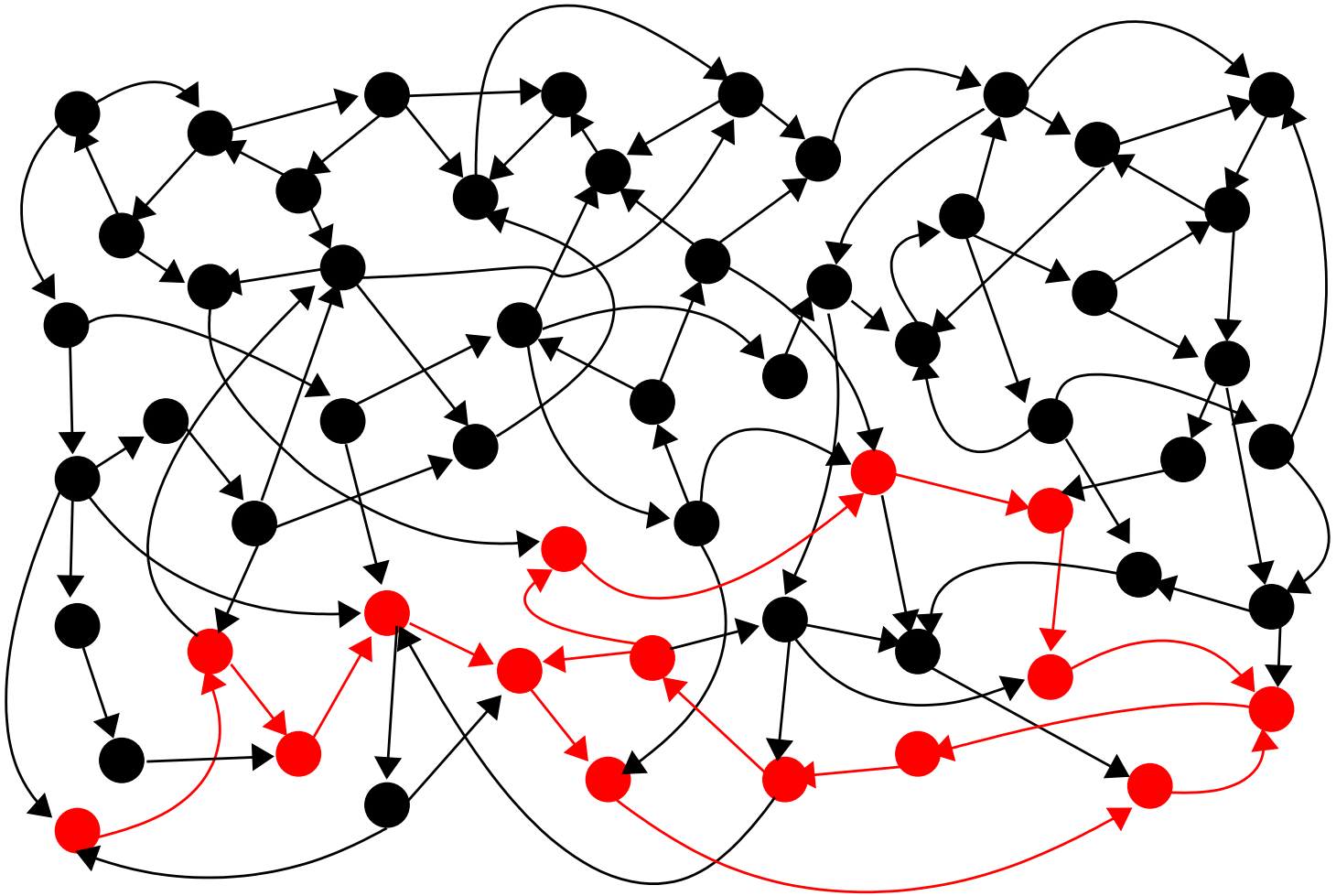
Simulation

- Simulation is based instead on the reproduction of traces of the model
- A trace is a possible sequence of events that can characterize one possible evolution of the model
- A simulation computes a large number of traces
- It then determines the performance indices by performing suitable statistics on the results computed during the traces
- The accuracy of the results depends on the number of traces that have been collected: the larger the number, the more precise the results will be, but also the longer will take to obtain them

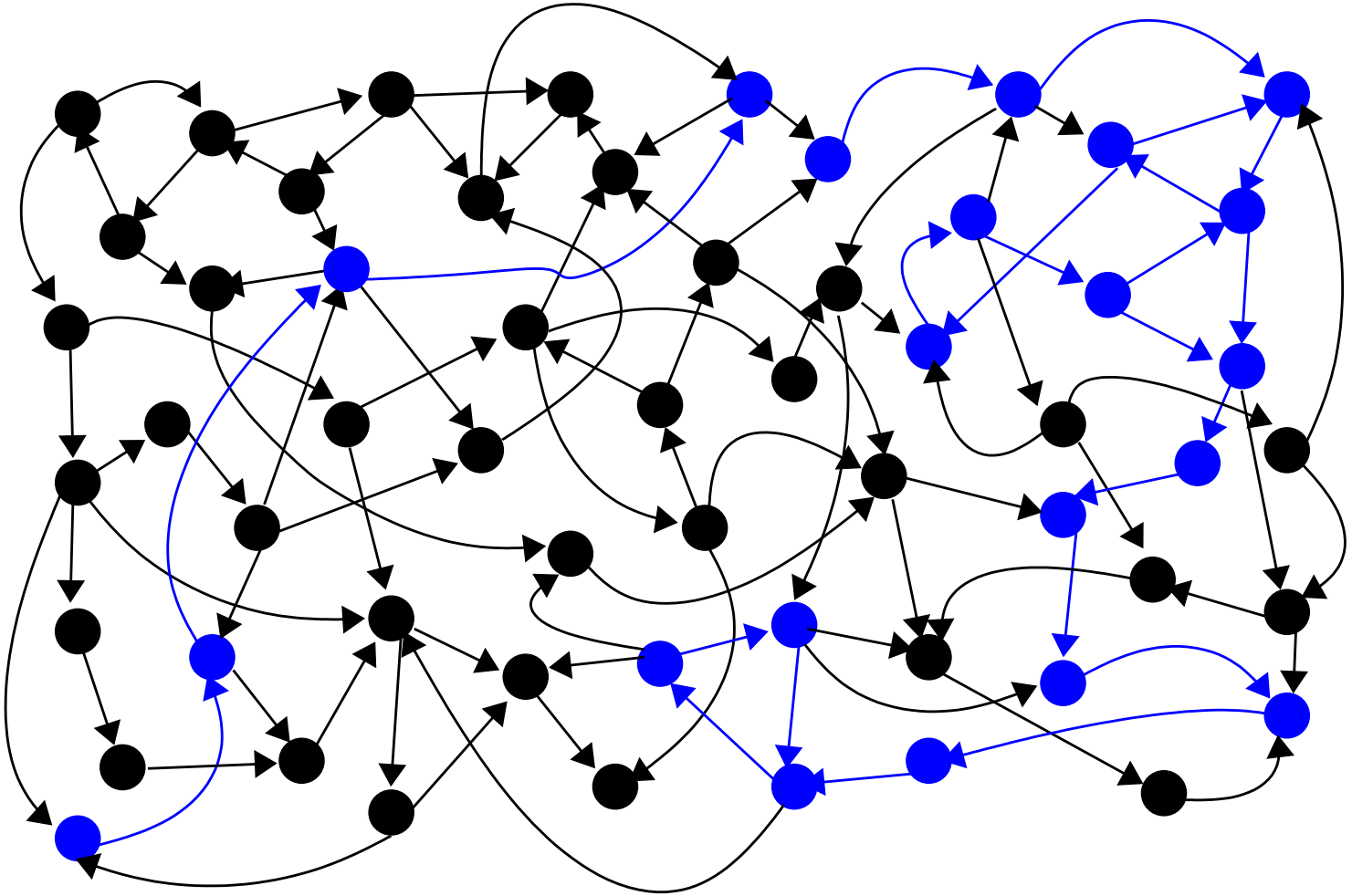
Sample traces



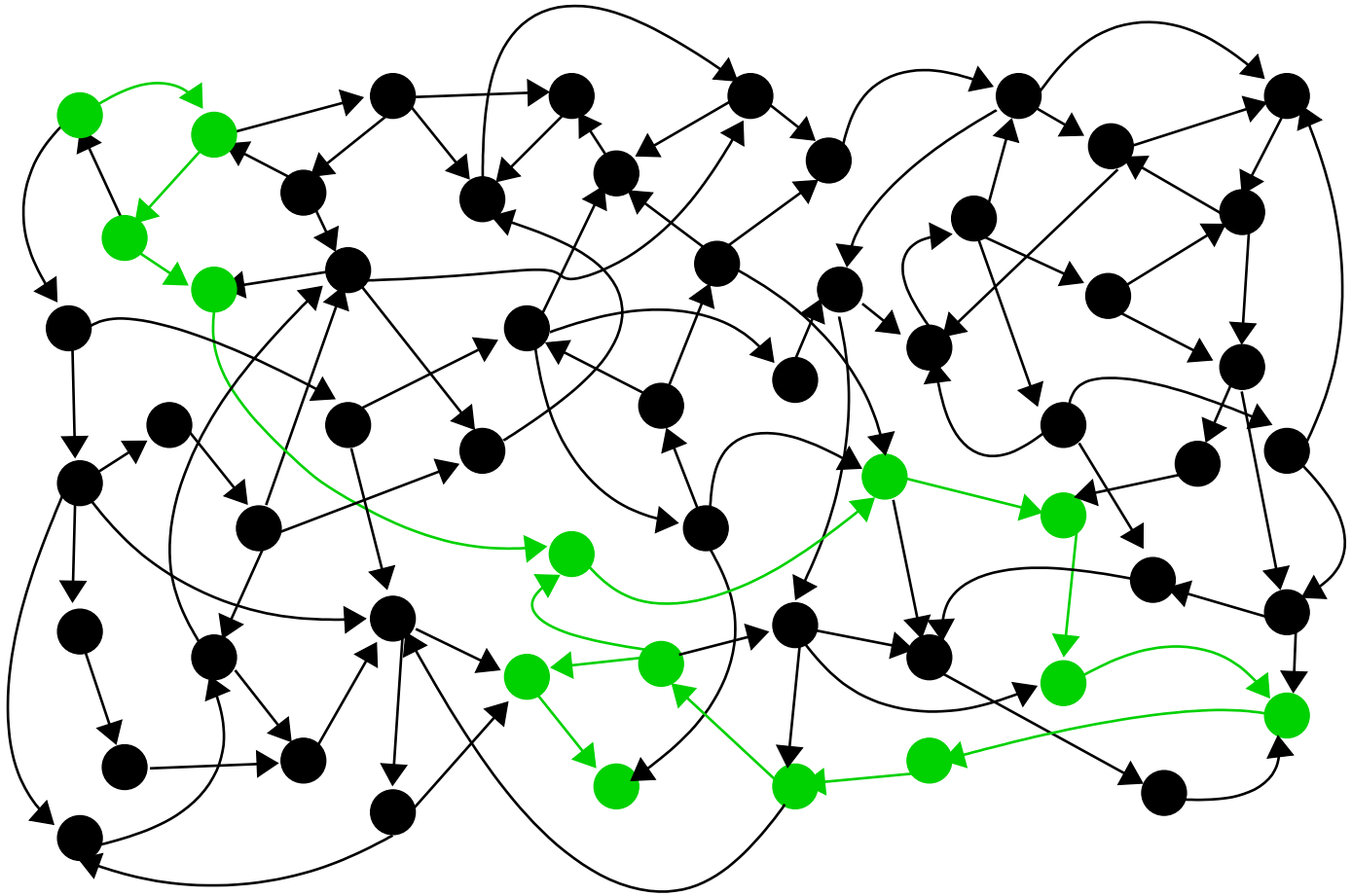
Sample traces



Sample traces



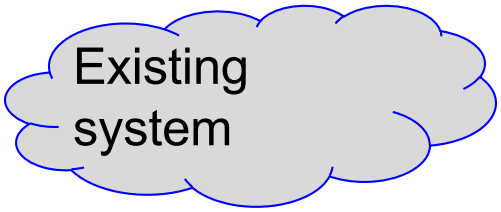
Sample traces

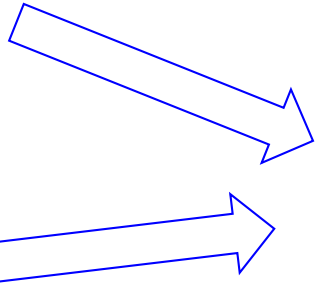


Sample traces and runs

- Using an analytic approach we characterise all possible sample traces by solving the overall model
- Using simulation we investigate the sample traces directly
- We allow the model to trace out a sample path over the state space
- Each run of the simulation model will generate another, usually distinct, sample path

Model Definition

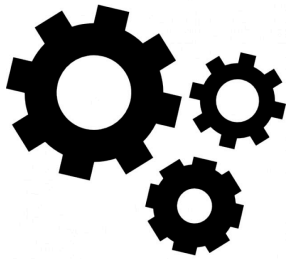




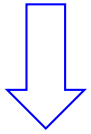
Parametrize the Model

Define the goal

Create the Model



Evaluate the Model



Performance indices

Evaluation techniques: summary

- Analytical techniques are the most efficient and the most precise, but are available only in very limited cases
- Numerical techniques can be made arbitrarily accurate using specific computation algorithms, but usually require a lot of time to produce the solution, which in many cases can make the approach unfeasible
- Simulation techniques are the most general, but might also be the less accurate, especially when considering cases in which rare events can occur. The solution time can also become really large when high accuracy is desired
- Hybrid techniques combine analytical/numerical methods with simulation

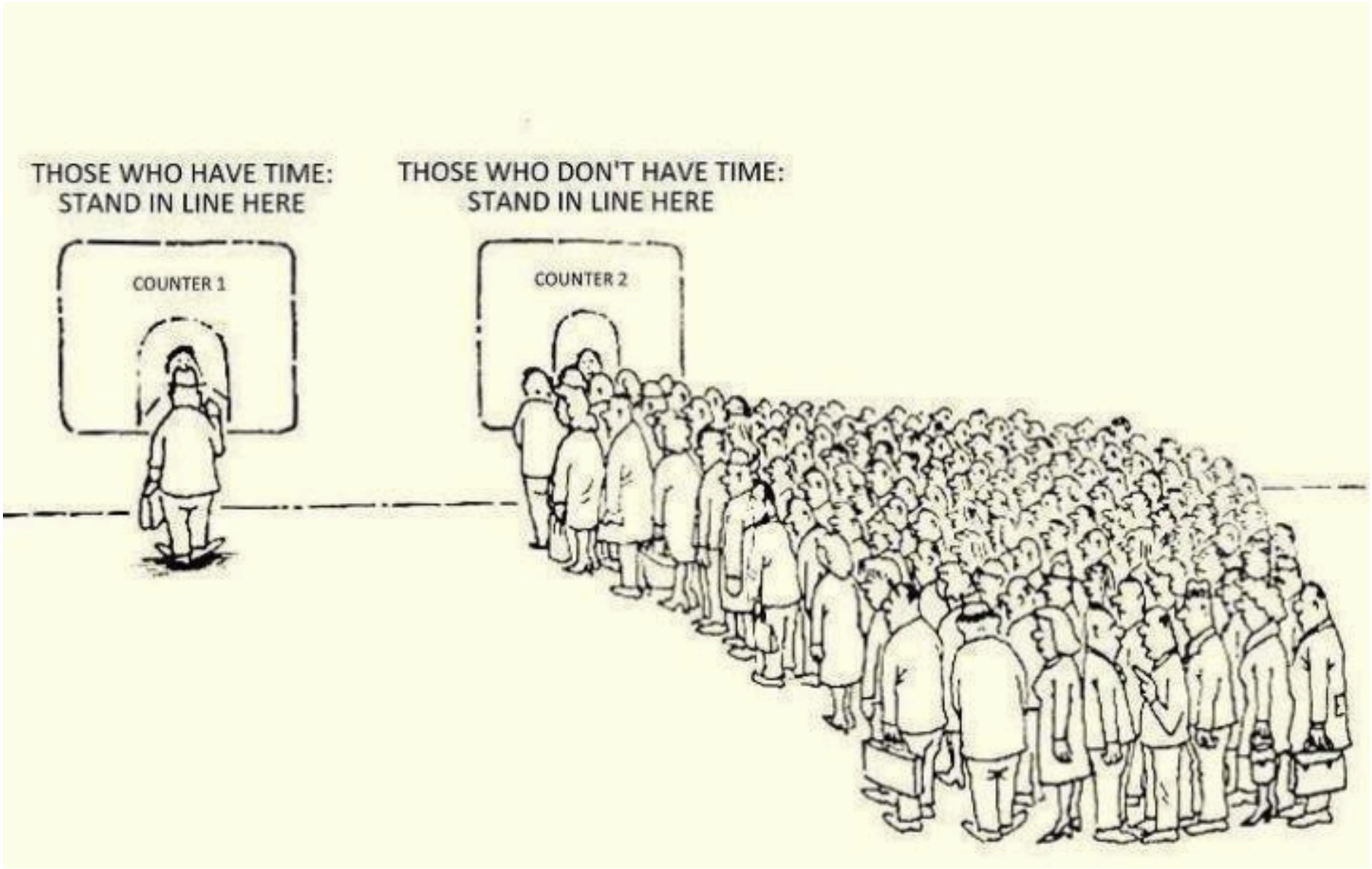
Reference Models of this course: Queueing Networks

Queueing theory is the theory behind what happens when you have a lot of jobs, scarce resources, and so long queue and delays.

Queueing network modelling is a particular approach to computer system modelling in which the computer system is represented as a *network of queues*

A network of queues is a collection of *service centers*, which represent system **resources**, and *customers*, which represent **users** or **transactions**

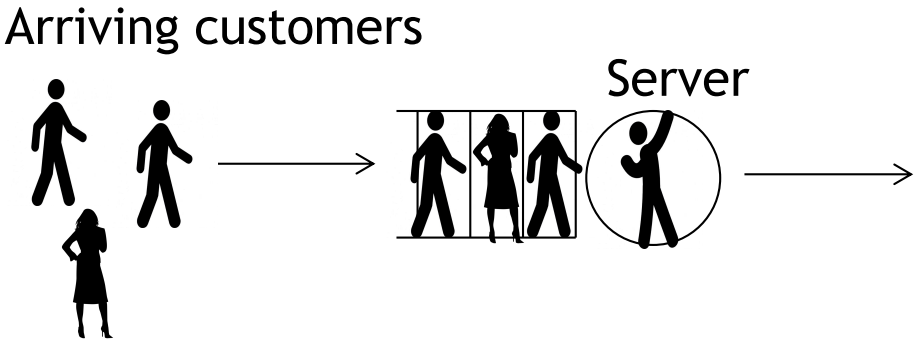
Queueing Networks



Queueing Networks

- Queueing theory applies whenever queues come up
- Queues in computer systems:
  - CPU uses a time-sharing scheduler
  - Disk serves a queue of requests waiting to read or write blocks
  - A router in a network serves a queue of packets waiting to be routed
  - Databases have lock queues, where transactions wait to acquire the lock on a record
- Predicting performance e.g. for capacity planning purposes
- Queueing theory is built on an area of mathematics called stochastic modelling and analysis

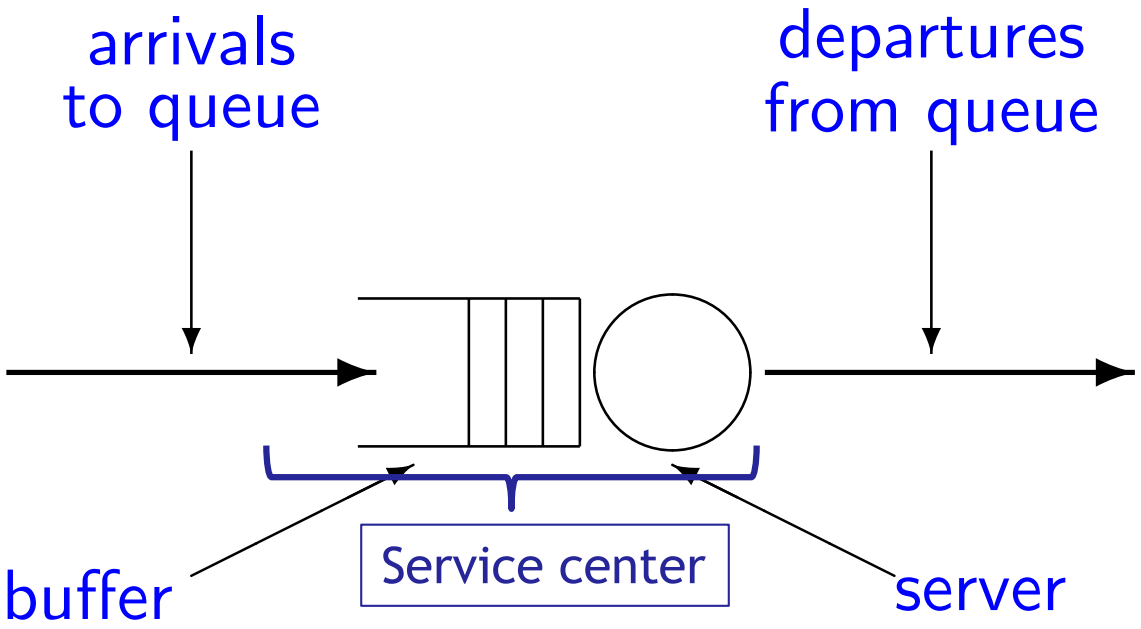
Success of queueing network: low-level details of a system are largely irrelevant to its high-level performance characteristics



Single queue

- The basic scenario for a single queue is that customers, who belong to some population arrive at the service facility
- The service facility has one or more servers which can perform the service required by customers
- If a customer cannot gain access to a server it must join a queue, in a buffer, until a server is available
- When service is complete the customer departs, and the server selects the next customer from the buffer according to the service discipline (queueing policy)

QN: graphical representation



Service center

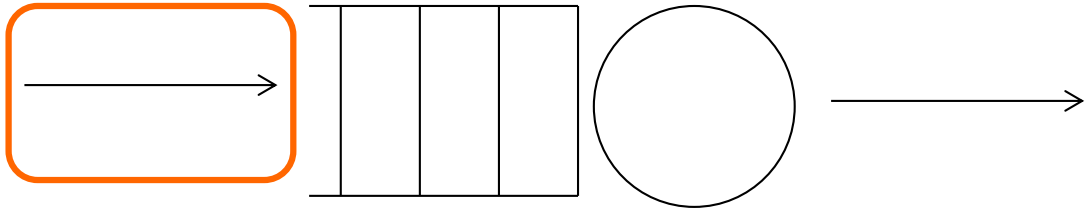
Different aspects characterize queuing models:

- Arrival
- Service
- Queue
- Population

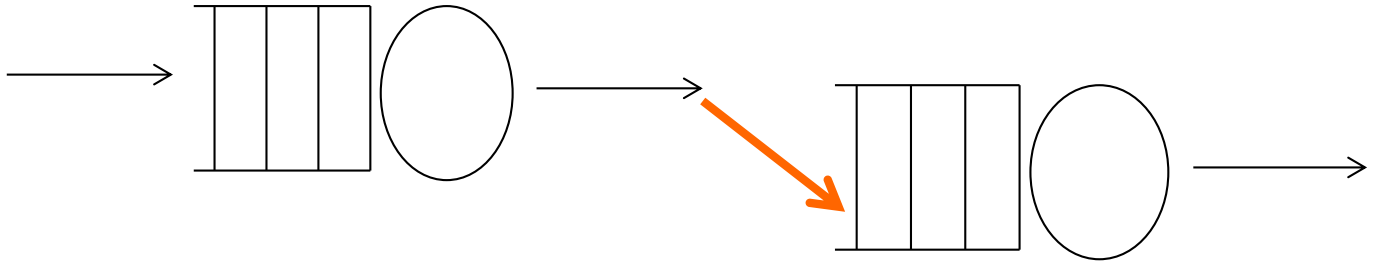
Arrival of customers

Arrivals represent jobs entering the system: they specify how fast, how often and which types of jobs does the station service.

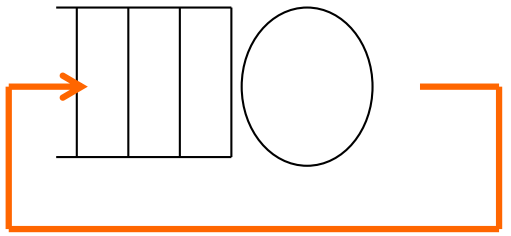
Arrival can come from an external source:



Arrival can come from another queue:



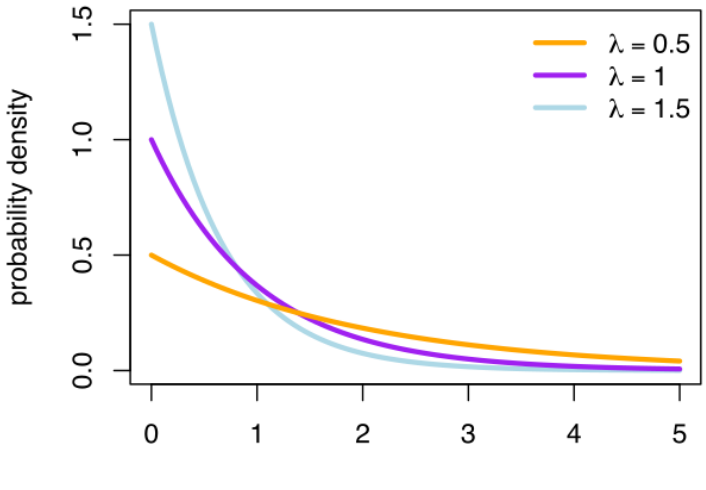
or even from the same queue, through a loop-back arc



Arrival patterns of customers

- The ability of the service facility to provide service for an arriving stream of customers depends not only on the mean arrival rate λ, but also on the pattern in which they arrive, i.e. the distribution function of the inter-arrival times

- We will primarily be considering queues in which the times between arrivals are assumed to be exponentially distributed



$$f(x;\lambda) = \begin{cases} \lambda e^{-\lambda x} & x \geq 0, \\ 0 & x < 0. \end{cases}$$

- However you should be aware that many other possibilities exist and are studied, such as bulk or batch arrivals

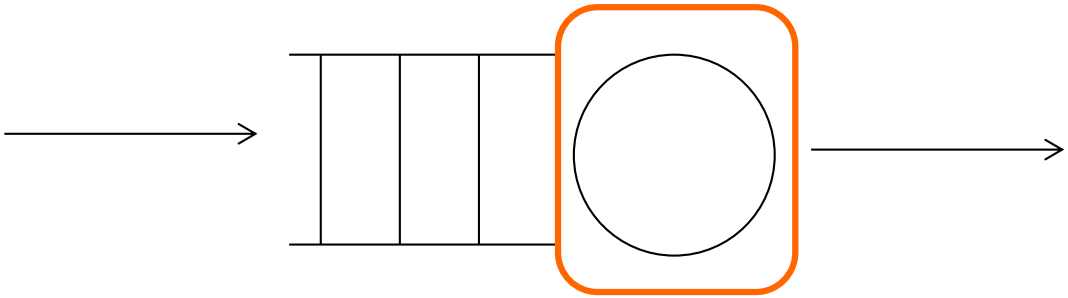
Service center

Different aspects characterize queuing models:

- Arrival
- Service
- Queue
- Population

Service

The service part represents the time a job spends being serviced.



Service time distribution

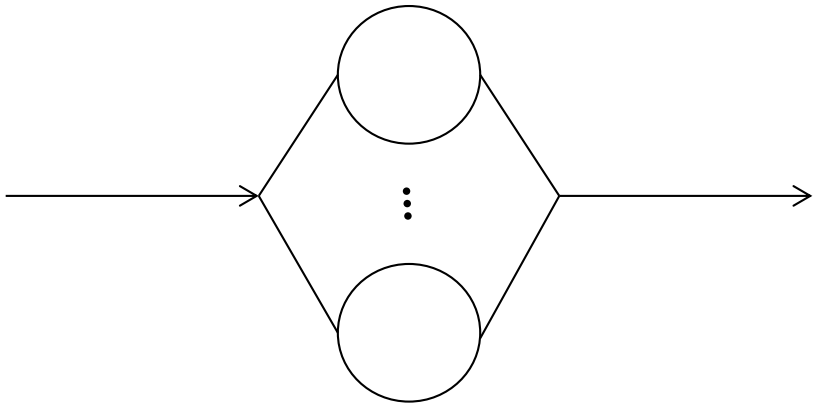
- The service time is the time which a server spends satisfying a customer
- As with the inter-arrival time, the important characteristics of this time will be its average duration and the distribution function
- We will mostly consider service times which obey an exponential distribution
- If the average duration of a service interaction between a server and a customer is 1/μ then μ is the **maximum** service rate

Number of servers

Possible situations:

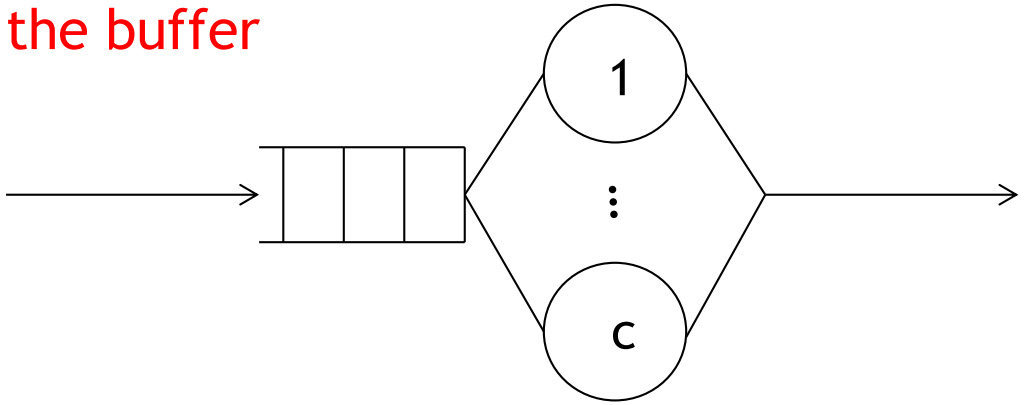
- a single server: the service facility only has the capability to serve one customer at a time; waiting customers will stay in the buffer until chosen for service; how the next customer is chosen will depend on the service discipline

- an infinite server: there are always at least as many servers as there are customers, so that each customer can have a dedicated server as soon as it arrives in the facility. There is no queueing, (and no buffer) in such facilities



Multiple servers

- Between these two extremes there are multiple server facilities
- These have a fixed number of c servers, each of which can service a customer at any time
- If the number of customers in the facility is less than or equal to c there will no queueing—each customer will have direct access to a server
- If there are more than c customers, the additional customers will have to wait in the buffer



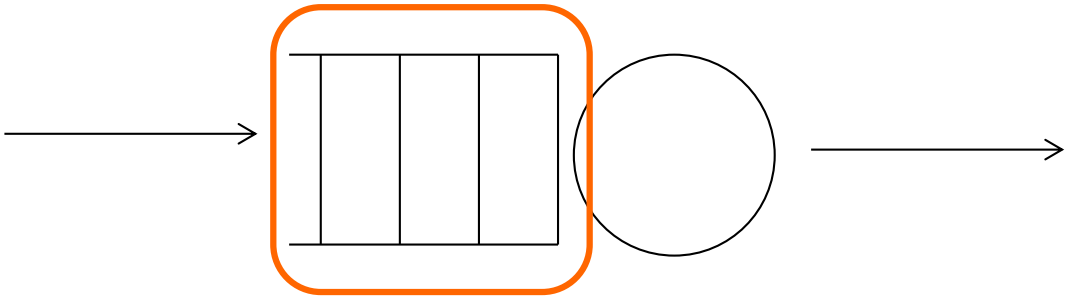
Service center

Different aspects characterize queuing models:

- Arrival
- Service
- Queue
- Population

Queue

If jobs exceed the capacity of parallel processing of the system, they are forced to wait *queueing* in a *buffer*



Buffer capacity

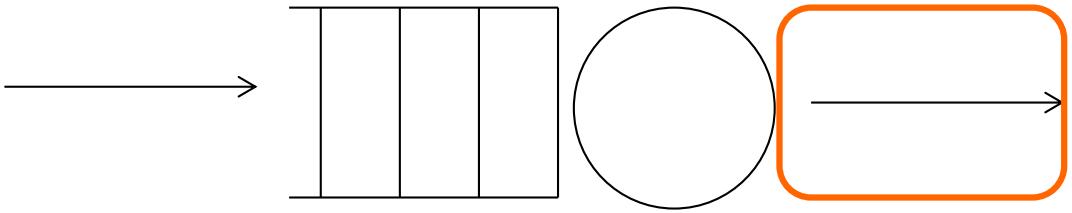
- Customers who cannot receive service immediately must wait in the buffer until a server becomes available

- If the buffer has finite capacity there are two alternatives for when the buffer becomes full:
  - either, the fact that the facility is full is passed back to the arrival process and arrivals are suspended until the facility has spare capacity, i.e. a customer leaves;
  - or, arrivals continue and arriving customers are lost (turned away) until the facility has spare capacity again

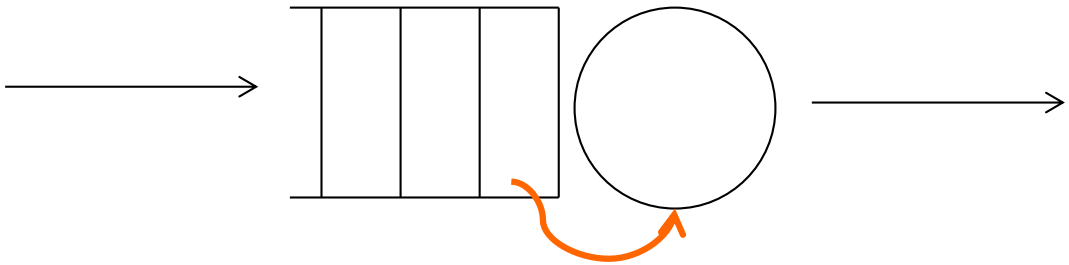
- If the buffer capacity is so large that it never affects the behaviour of the customers it is assumed to be infinite

Queue

When the (one of the) job(s) currently in service leaves the system, one of the job in the queue can enter the now free service center



Service discipline/queuing policy determines which of the job in the queue will be selected to start its service



Service discipline

- When more than one customer is waiting for service, we need a rule for selecting which of the waiting customers will be the next one to gain access to a server

- The commonly used service disciplines are
  - FCFS first-come-first-serve (or FIFO first-in-first-out)
  - LCFS last-come-first-serve (or LIFO last-in-first-out)
  - RSS random-selection-for-service
  - PRI priority, the assignment of different priorities to elements of a population is one way in which classes are formed

- We will only consider systems with the FCFS (FIFO) service discipline, at least initially

Service center

Different aspects characterize queuing models:

- Arrival
- Service
- Queue
- Population

Population

- The characteristic of the population which we are interested in is usually the size
- Clearly, if the size of the population is fixed, at some value N, no more than N customers will ever be requiring service at any time
- When the population is finite, the arrival rate of customers will be affected by the number who are already in the service facility (e.g. zero arrivals when all N are all already in the facility)
- When the size of the population is so large that there is no perceptible impact on the arrival process, we assume that the population is infinite

Population

- Ideally, members of the population are indistinguishable from each other
- When this is not the case we divide the population into classes whose members all exhibit the same behaviour
- Different classes differ in one or more characteristics, for example, arrival rate, service demand
- Identifying different classes is a workload characterisation task

Example

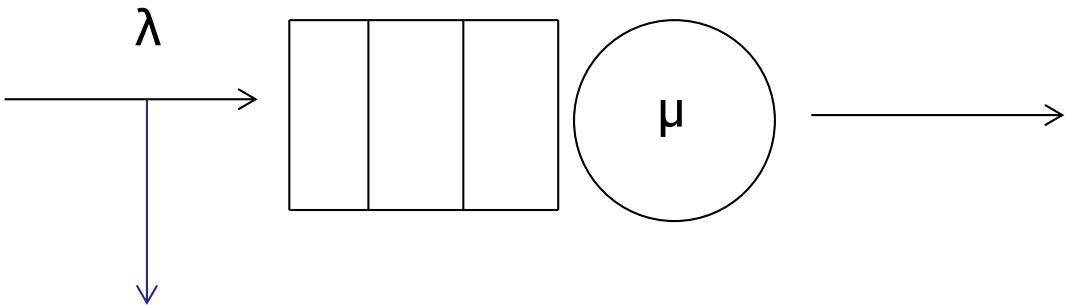
- Consider a wireless access gateway:
- Measurements have shown that packets arrive at a mean rate of 125 packets per second, and are buffered
- The gateway takes 2 milliseconds on average to transmit a packet
- The buffer currently has 13 places, including the place occupied by the packet being transmitted and packets that arrive when the buffer is full are lost
- **Goal of the modelling and analysis:**
  - We wish to find out if the buffer capacity which is sufficient to ensure that less than one packet per million gets lost

Example

Making exponential assumptions about the arrival rate and the service rate we would model the gateway as:

A single queue center with:

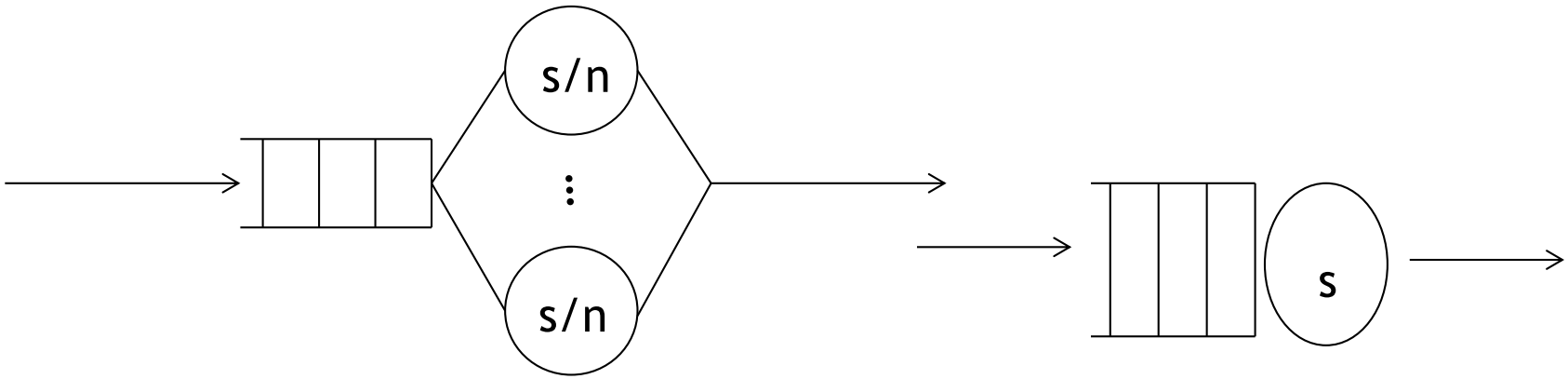
- Finite queue capacity=13
- FCFS service discipline
- Exponential arrival distribution with rate λ = 125 req/s
- Exponential service distribution with rate and μ = 1/(2ms) = 500 req/s



Example

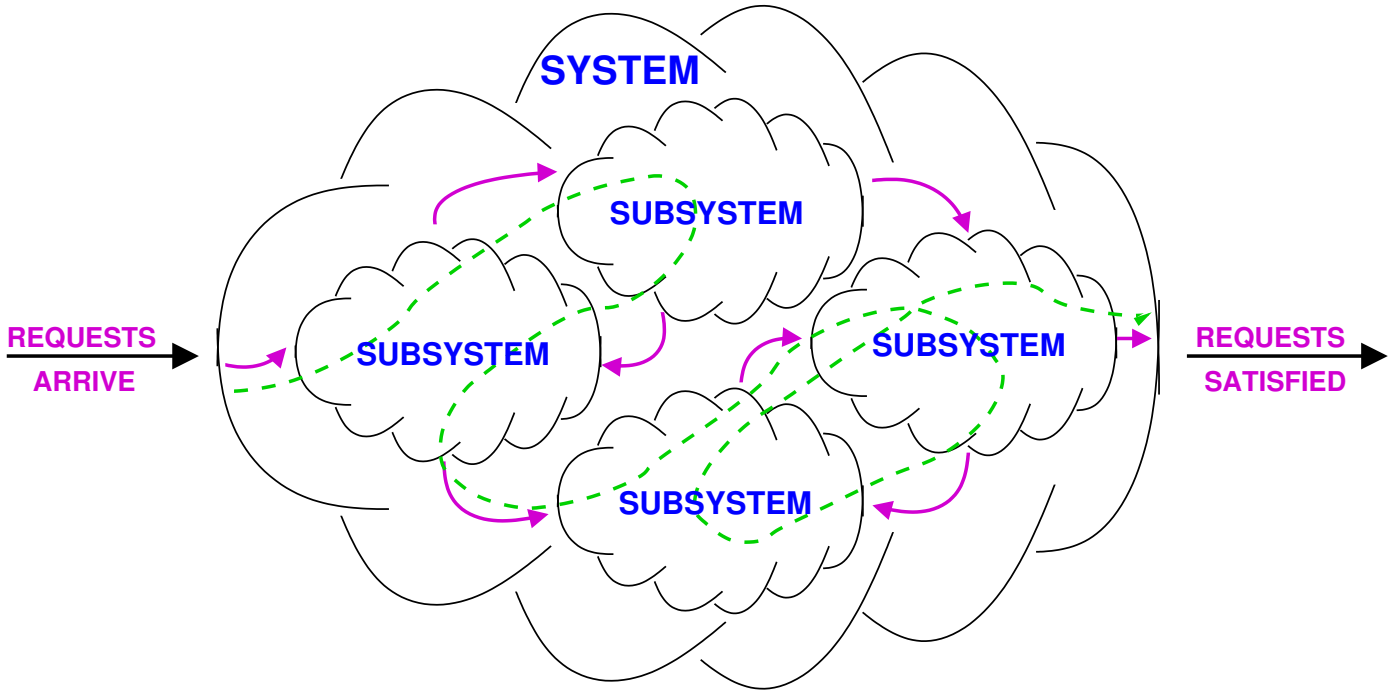
- You are given a choice between one fast CPU of speed s or n slow CPU each of speed s/n. Your goal is to minimize mean response time

- Question: Which is the better choice?
  - Arrival rate? Job type?
  - Pre-emption?

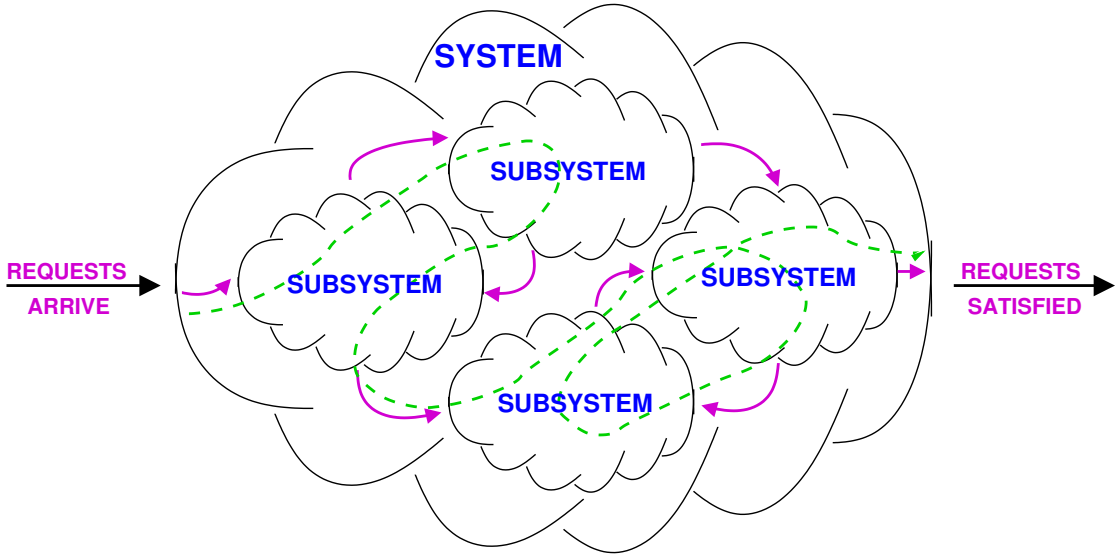


Queueing Networks

For many systems we can adopt a view of the system as a collection of resources and devices with customers or jobs circulating between them



Queueing Networks



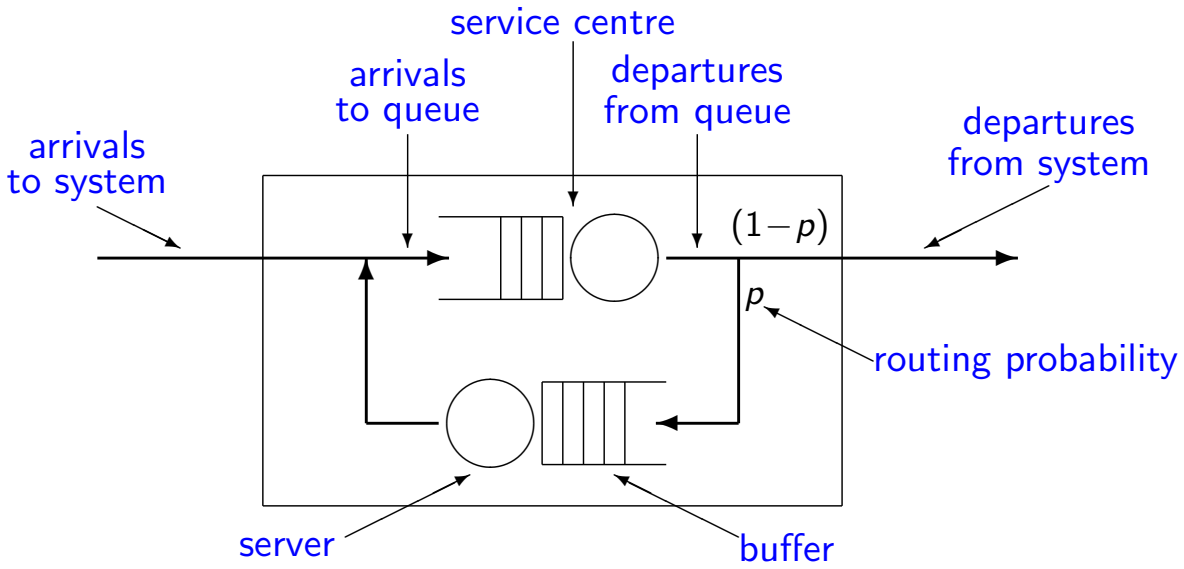
- We can associate a service center with each resource in the system and then route customers among the service centres

- After service at one service centre a customer may progress to other service centres, following some previously defined pattern of behaviour, corresponding to the customer's requirement

Queueing Networks

A queueing network can be represented as a graph where nodes represent the service centers k and arcs the possible transitions of users from one service center to another

Nodes and arcs together define the network topology



Queueing Networks

A network may be:

- Open: customers may arrive from, or depart to, some external environment
- Closed: a fixed population of customers remain within the system
- Mixed: there are classes of customers within the system exhibiting open and closed patterns of behaviour respectively

Queueing networks

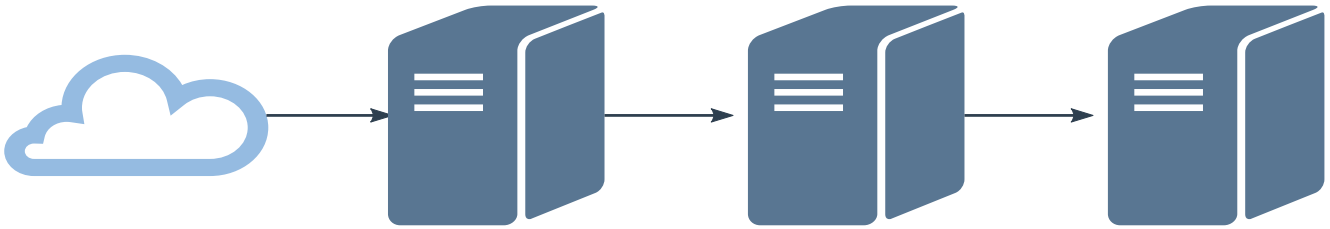
Different aspects characterize queuing models:

- Arrival
- Service
- Queue
- Population
- Routing

Routing

- Whenever a job, after finishing service at a station has several possible alternative routes, an appropriate selection policy must be defined
- The policy that describes how the next destination is selected is called routing
- Routing specification is required only in all the points where jobs exiting a station can have more than one destination

Open Networks: Examples



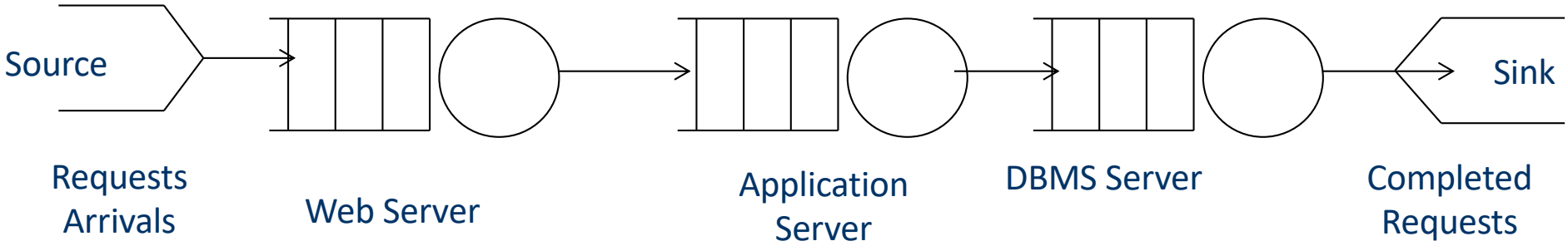
A client server system, dealing with external arrivals, which is architected with three tiers: the first one includes one web server, the second tier includes one application server and the third one includes a database server

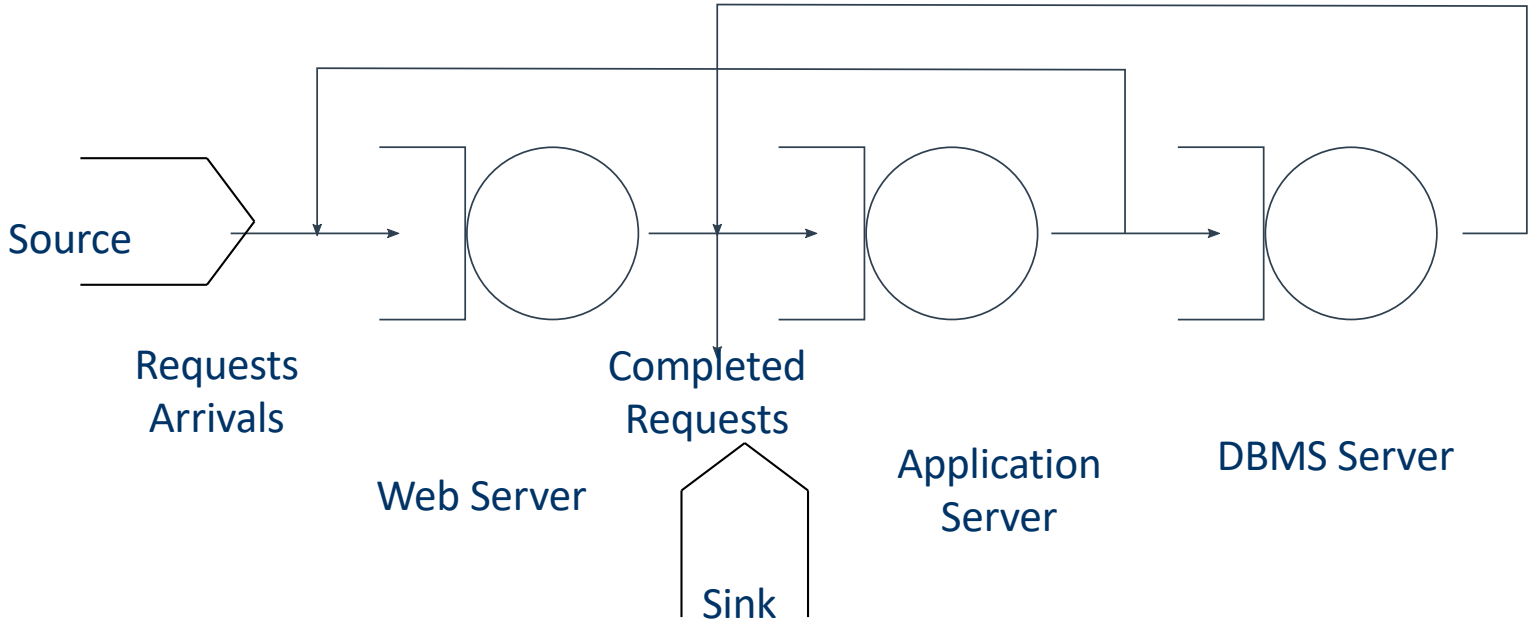
Provide a QN model of the system and evaluate the overall throughput considering that the network delay is negligible with respect to the other devices and two different cases:

1) The only thing we know is that each server should be visited by the application

2) In the second case we know that the application **after visiting the web ser**ver requires some operations at **the application server** and then **can go back** to **the web server** and **leave** the system **or** can require service at the **DBMS** and then **go back** to the **application server**

Open Networks

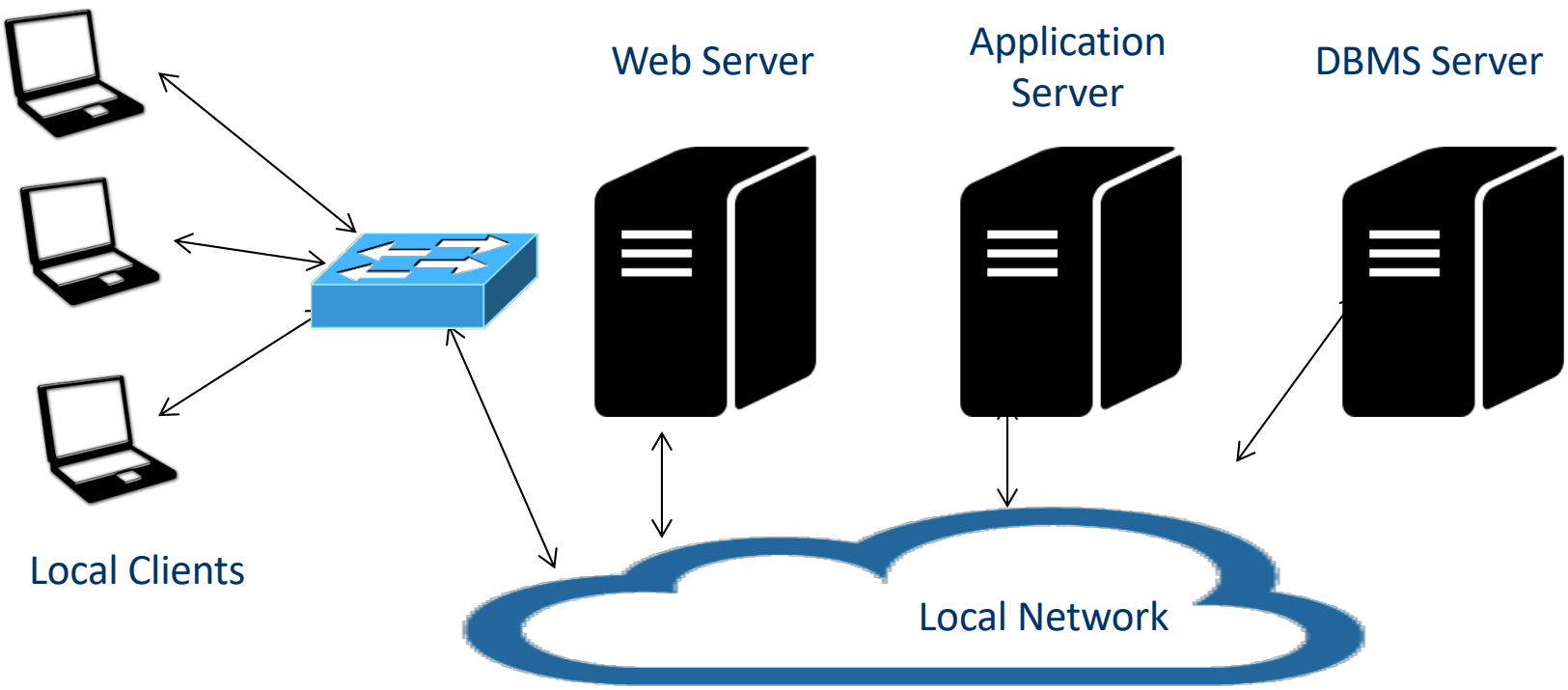




Scenario 1: Tandem networks

Tandem queuing networks are used for example to model production lines, where raw parts enter the systems, and after a set of stages, the final product is completed (and leaves)

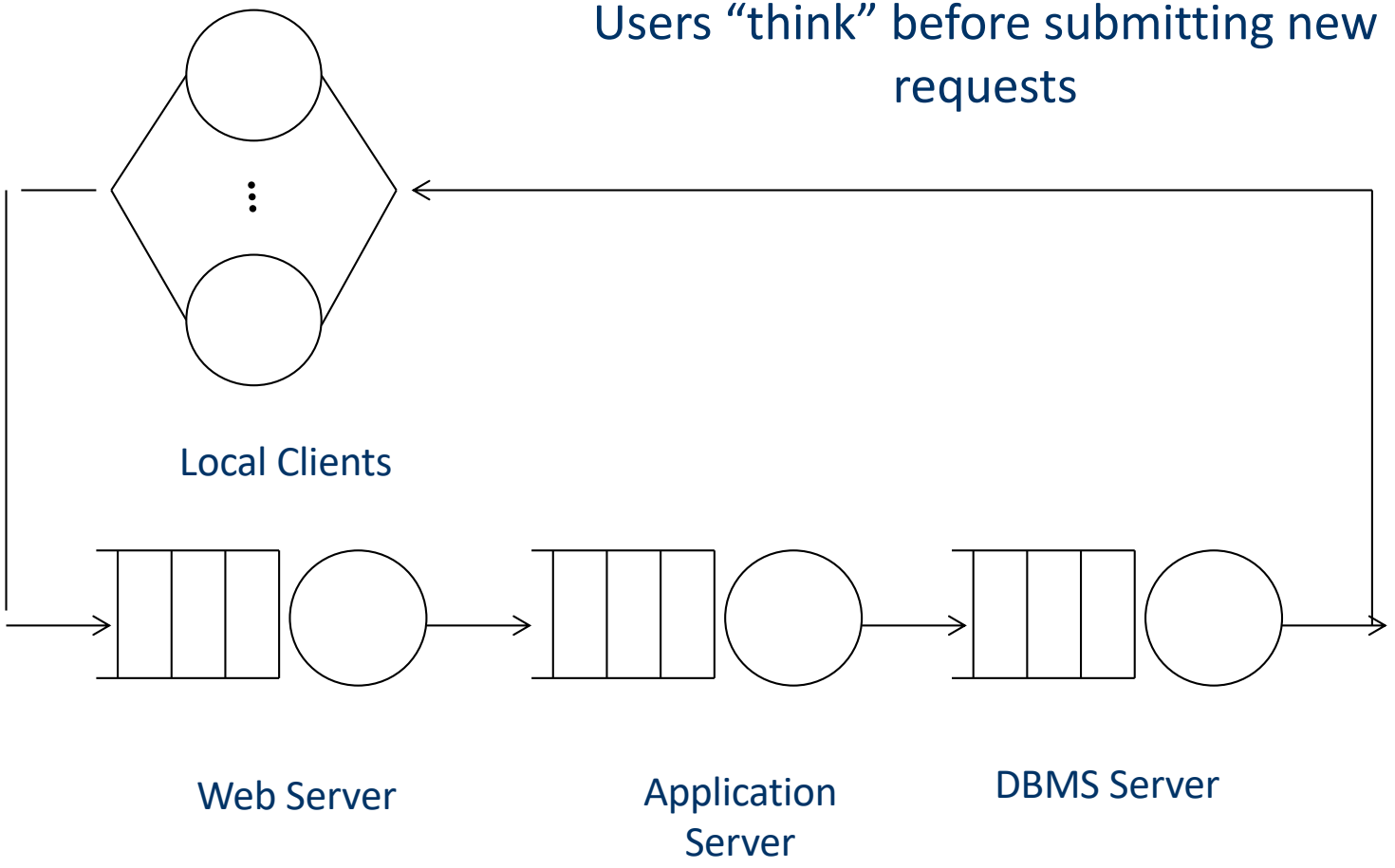
Closed Networks: Examples



A client server system, **with a finite number of customers**, which is architected with three tiers: the first one includes one web server, the second tier includes one application server and the third one includes a database server.

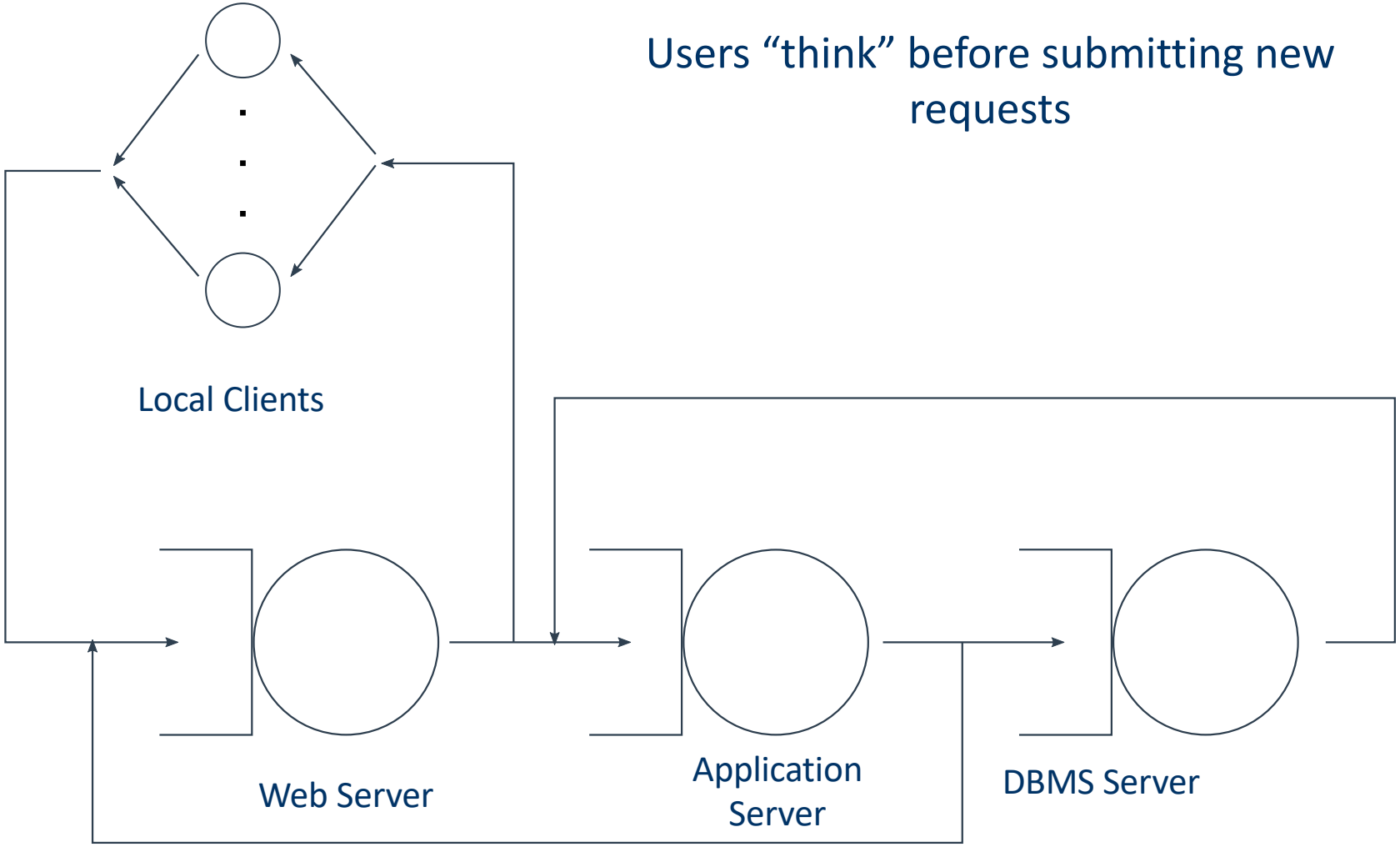
Provide a QN model of the system and evaluate the system throughput considering that Network delay is negligible with respect to the other devices. Model the two different cases previously described.

Closed Networks (first model)

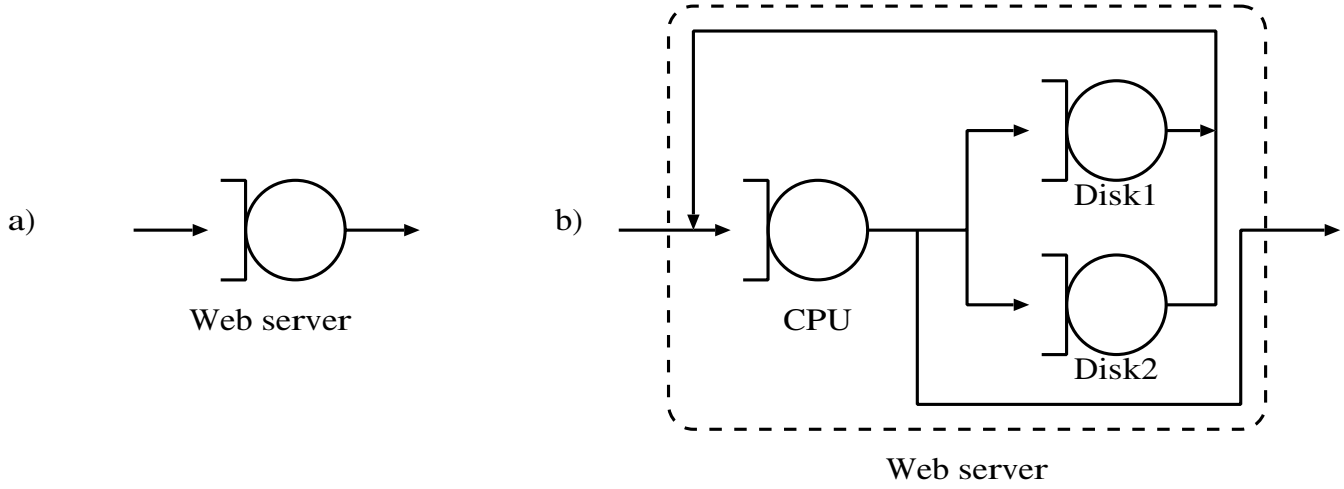


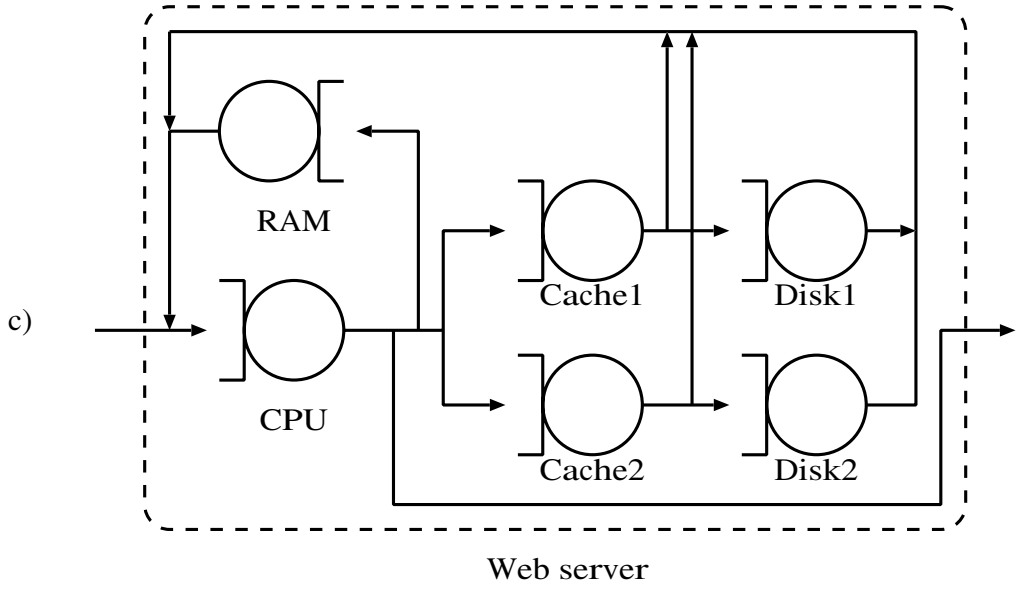
Closed Networks (second model)

Users “think” before submitting new requests

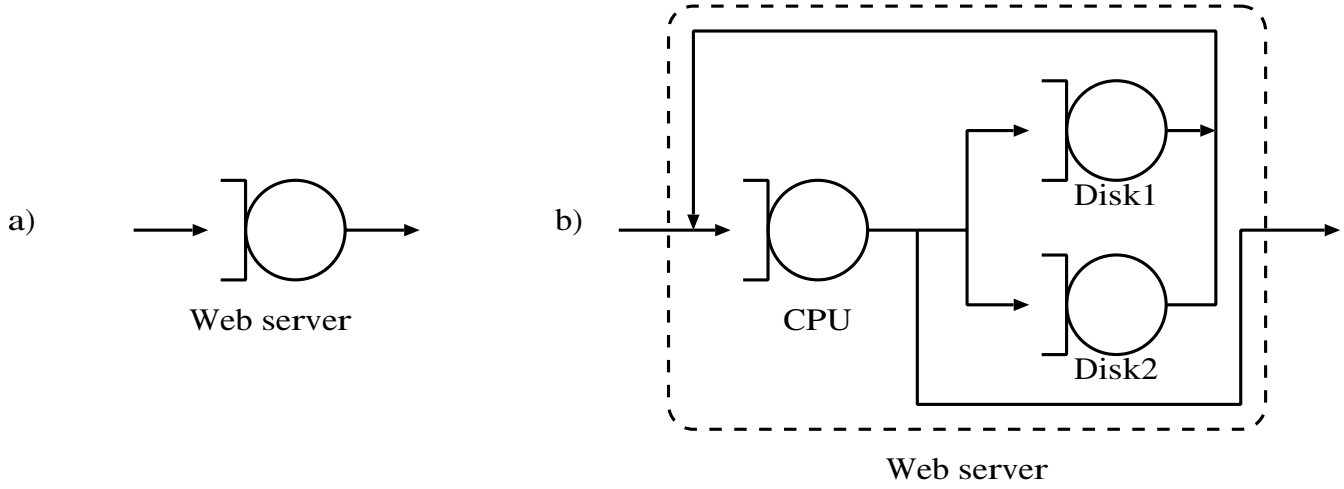


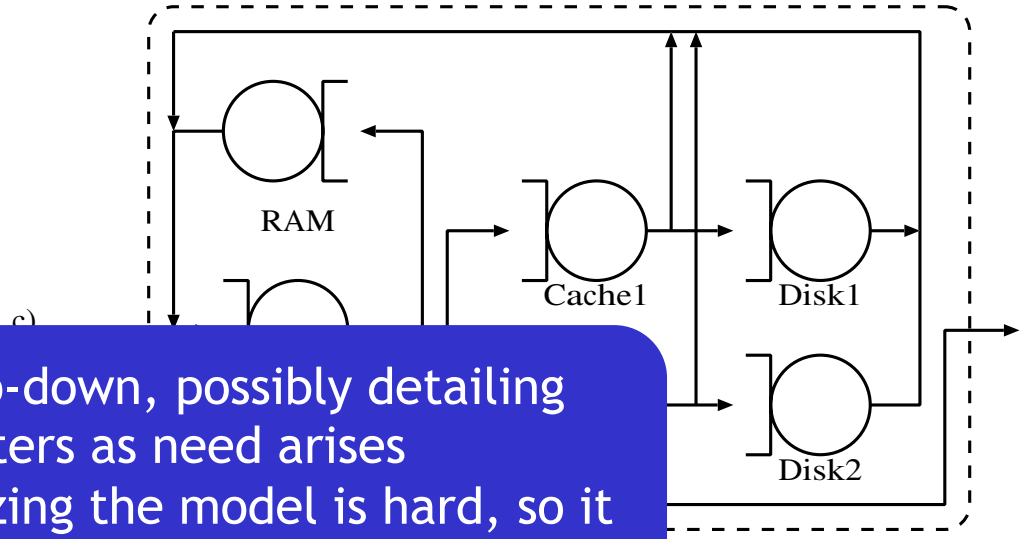
Level of Detail





Level of Detail





- Proceed top-down, possibly detailing service centers as need arises
- Parameterizing the model is hard, so it is better to keep the complexity low

Expressiveness and Extended Queueing Networks

- There are some features of contemporary computer and communication systems which are not easily represented by traditional queueing networks
  - Simultaneous resource possession: In a computer system a job may continue to compute while its I/O requests are progressing in parallel. Thus, the job may be simultaneously using two or more resources of the system
  - Fork and Join primitives used in computer systems to create and synchronise subprocesses, cause the number of jobs in the system to change and invalidate the assumption of independence among jobs

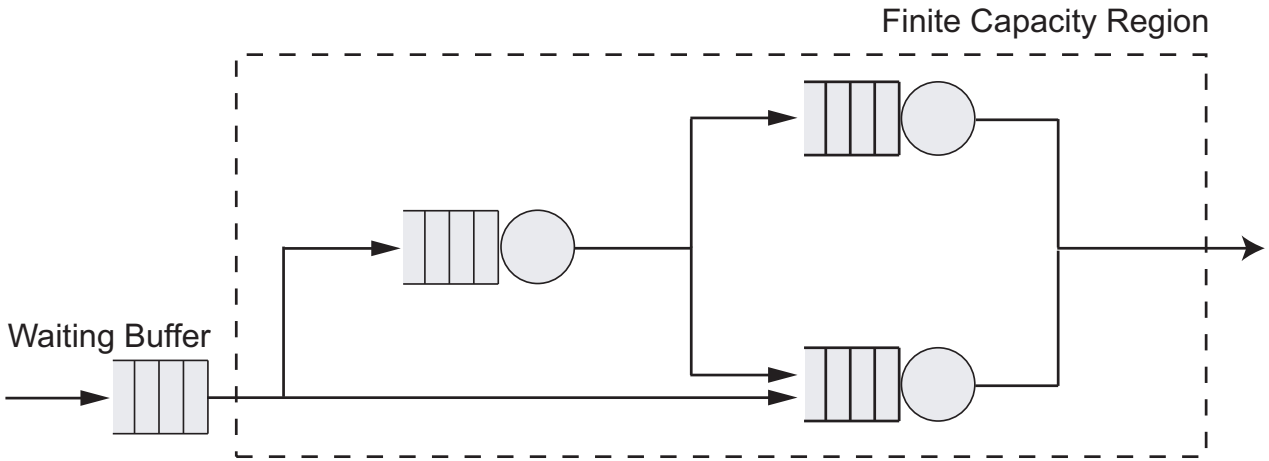
- In such scenarios, you need to rely on simulation

Finite capacity region

Models of simultaneous resource possession due to memory or software constraints often require finite capacity regions

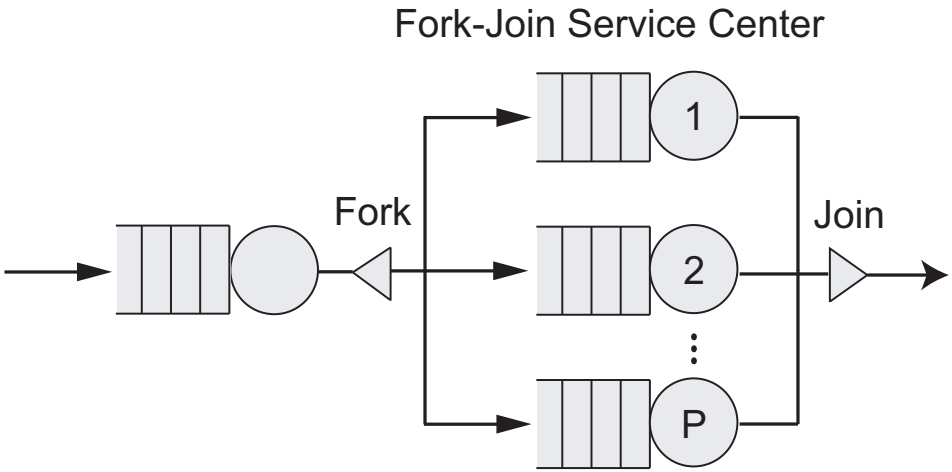
These are subnetworks where the number of circulating jobs is constrained

Shared constraints impose an upper bound on the allowed number of jobs in the region regardless of their service class. *Dedicated* constraints, instead, limit the number of cycling jobs for a specific class. Jobs arriving to a full region enqueue in a waiting buffer outside the region



Fork and Join

- Fork-join service centers are employed to represent resources that can serve jobs in parallel (e.g., storage, parallel, grid and big data systems)
- Fork-join service centers are composed by P > 1 queues in parallel
- Each time a job arrives to a fork-join service centers, it is split by a fork node into P sibling tasks. Each of them is assigned to a different parallel queue. After receiving service, jobs synchronize and merge at a join node before leaving the service center



Expressiveness and Extended Queueing Networks

- Bulk and train arrivals: In communication networks the arrival of packets may occur in batches (bulk arrivals) or in quick succession (train arrivals). This breaks assumptions of independence between customers, and exponential inter-arrival times

- Load dependent arrivals: Computer networks and distributed systems have intelligent load-balancing policies that cause new jobs or packets to go to one of a number of devices depending upon the load observed in the recent past

- Defections from the queue: Routers often set a maximum limit on the time that a packet or request is able to stay in a queue. Thus, packets may be dropped from the queue on the assumption that they may have already been retransmitted by higher protocol layers

- Evaluation: simulation or ad-hoc studies