



# Computing Infrastructures

 POLITECNICO DI MILANO



## Performance modeling

Prof. Danilo Ardagna

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



# The topics of the course

2



## A. HW Infrastructures:

- **System-level:** Computing Infrastructures and Data Center Architectures, Rack/Structure;
- **Node-level:** Server (computation, HW accelerators), Storage (Type, technology), Networking (architecture and technology)
- **Building-level:** Cooling systems, power supply, failure recovery



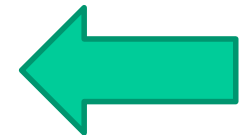
## B. SW Infrastructures:

- **Virtualization:** Process/System VM, Virtualization Mechanisms (Hypervisor, Para/Full virtualization)
- **Computing Architectures:** Cloud Computing (types, characteristics), X-as-a service, Edge/Fog Computing
- **Machine and deep learning-as-a-service**



## C. Methods:

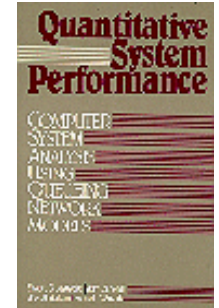
- **Reliability and availability of datacenters** (definition, fundamental laws, RBDs)
- **Disk performance** (Type, Performance, RAID)
- **Scalability and performance of datacenters** (definitions, fundamental laws, queuing network theory)



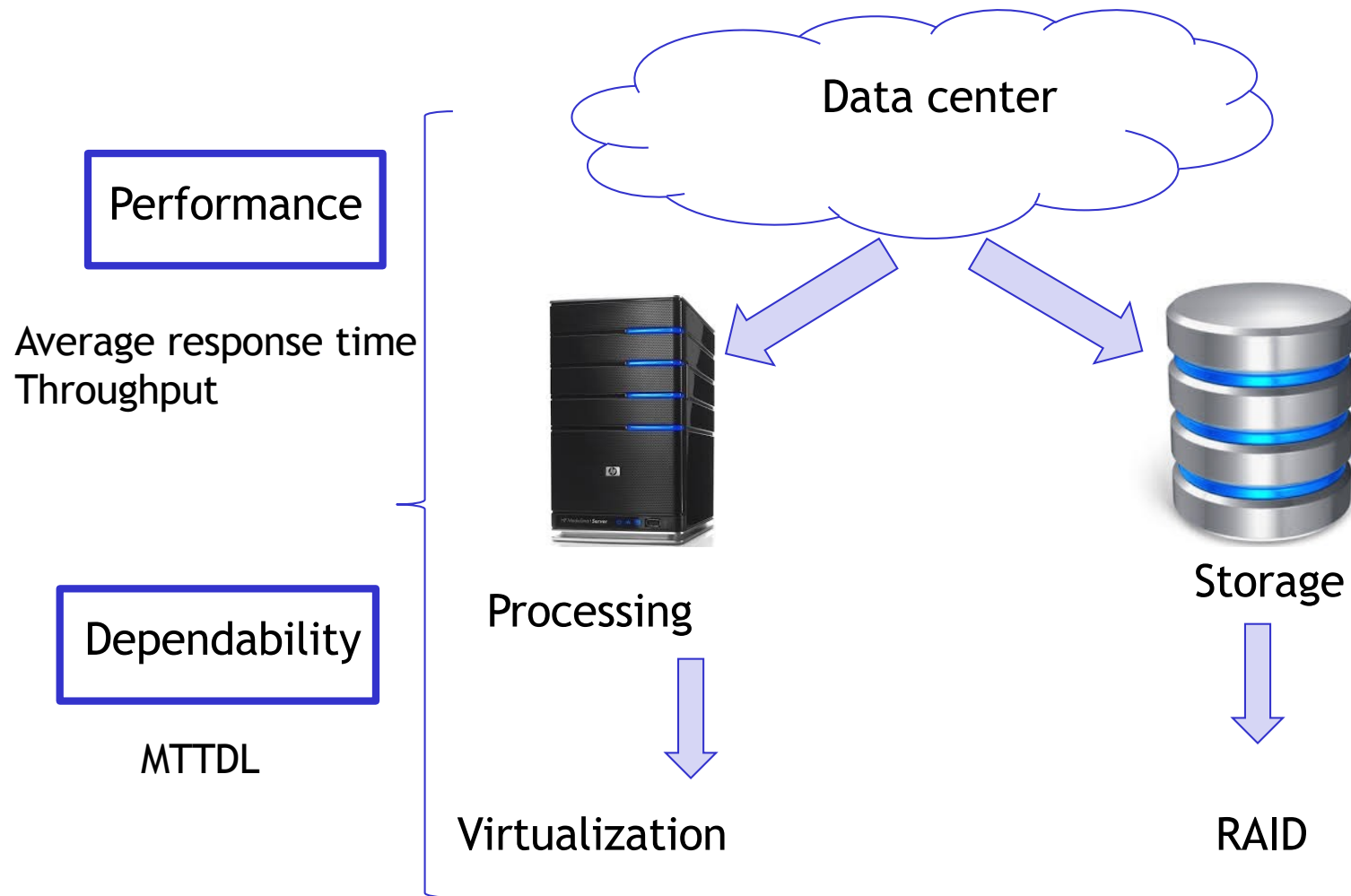


# Quantitative System Performance Computer System Analysis Using Queueing Network Models

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



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





- **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

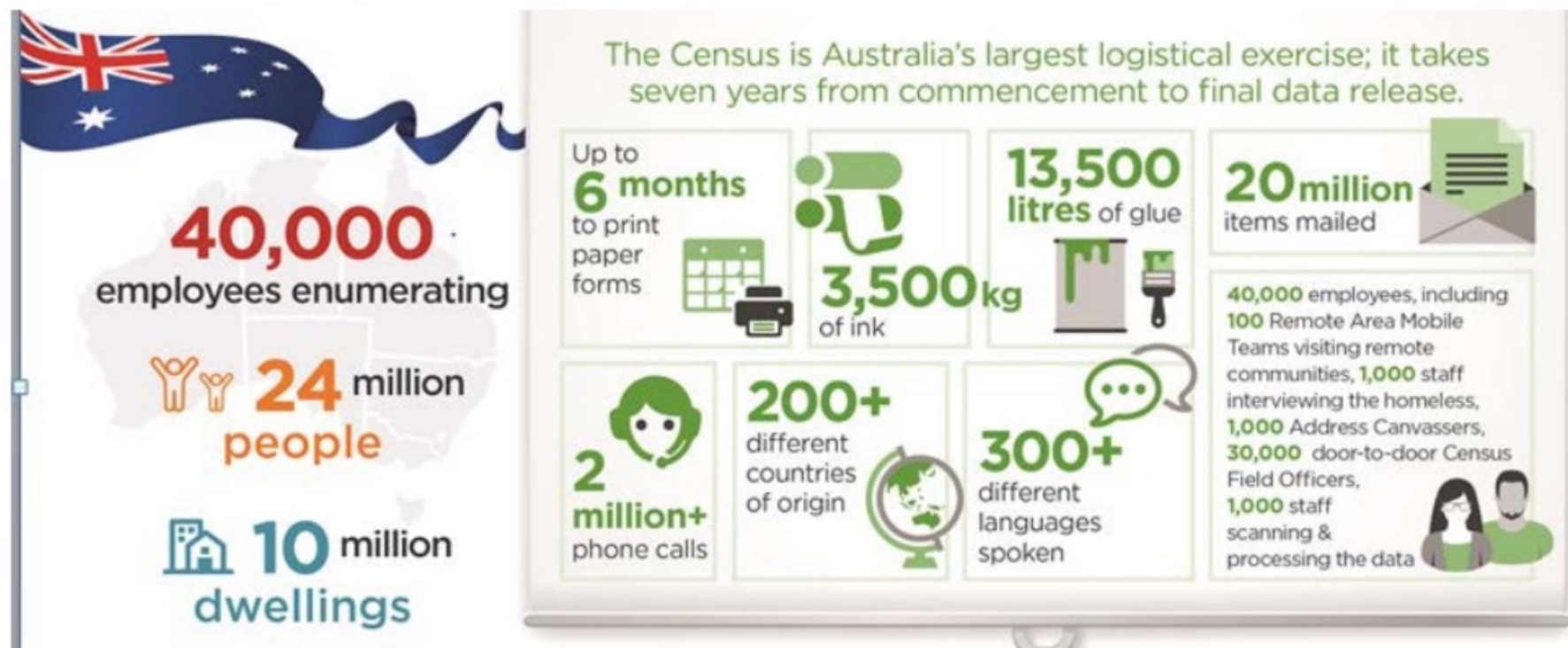


- Common practice: system mostly validated versus “functional” requirements rather than versus quality ones
- Different (and often not available) skills required for quality verification
- Short time to market, i.e. quickly available products and infrastructures “seem” to be more attractive nowadays!
- Little information related to quality is usually available early in the system lifecycle but its understanding is of **great importance from the cost and performance point of view**
  - During design and system sizing
  - But also during system evolution
    - E.g. changes in workloads or number of users

# Head Count

- Every 5 years Australians update their census.

*“Australia’s largest peacetime logistical operation”*





# Trusting Your Partners

8

- The ABS\* through a public tender awarded IBM for \$9.6M a contract to implement a digital census solution by 2016.
- ABS wisely allocated a "Load Test" appropriation (\$469K of which \$325K was spent on software licenses).

CN ID: CN2641301  
Agency: Australian Bureau of Statistics  
Publish Date: 27-Oct-2014  
Category: Software maintenance and support  
Contract Period: 1-Oct-2014 to 31-Oct-2016  
Contract Value (AUD): \$9,606,725.00  
Description: Design, development and implementation of eCensus Solution 2016

Procurement Method: Limited tender  
Confidentiality - Contract: No  
Confidentiality - Outputs: No  
Consultancy: No  
Agency Reference ID: ABS2014.105

## Supplier Details

Name: IBM Australia Ltd  
Postal Address: 8 Brisbane Ave  
Town/City: Barton  
Postcode: 2600  
State/Territory: ACT  
Country: AUSTRALIA  
ABN: 79 000 024 733

<https://www.tenders.gov.au/Cn/Show/1D46611D-EA19-ED83-2C73-D65E88772130>



# A story in three acts



**Malcolm Turnbull** ✓  
@TurnbullMalcolm

We filled in the @ABSCensus tonight online - v easy to do. And so important for planning better Govt services & investment for the future

♡ 327 11:17 AM - Aug 9, 2016

💬 1,475 people are talking about this



<https://twitter.com/TurnbullMalcolm/status/762940763801989121>



**Australian Bureau of Statistics** ✓  
@ABSStats

The ABS & Census websites are currently experiencing an outage. We're working to restore the service. We will keep you updated. Thank you.

♡ 1,079 12:38 PM - Aug 9, 2016

💬 1,956 people are talking about this

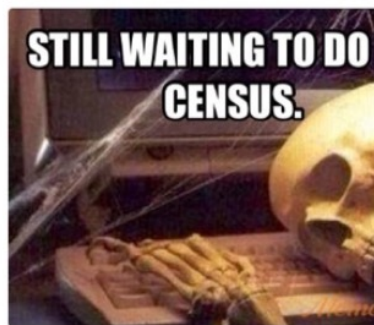


<https://twitter.com/ABSStats/status/762961251764805633>



**Narelle #FreePress**  
@narelleford

@b\_spectabilis @TurnbullMalcolm @ABSCensus  
♡ 20 2:11 PM - Aug 9, 2016



🔍 See Narelle #FreePress's other Tweets

<https://twitter.com/narelleford/status/762984702915465216>



**Colby Swandale**  
@oceanicpanda

Breaking news. We have an image of #census2016 data from moments ago. #censusfail  
♡ 565 12:15 PM - Aug 9, 2016

💬 255 people are talking about this

<https://twitter.com/oceanicpanda/status/762955516096094208>

Given that millions of Australians can play Pokemon Go at once and it doesn't crash is a good reason to outsource the census to Nintendo — Tim Beshara (@Tim\_Beshara) August 9, 2016



Credits: Prof- G. Moltò, UPV

# Official vs No Official

- The official conclusion (13/10/2016) of the Office of Cybersecurity was:
  - [...] *although the site withstood an initial DDoS attack and was coping with over 7,000 census forms a minute, a second and third attack took it down*
- Critical: It is believed that the system was developed on IBM WebSphere and ran on IBM Softlayer (on-premises Cloud) instead of a public Cloud.



# An Unexpected Turn of Events <sup>11</sup>

- A couple of students, with no previous AWS experience, developed a serverless system in one weekend supporting 4 times the load used to test the IBM system for ~~\$500~~ \$30

TECH

How two Uni Students built a better Census site in just 54 hours for \$500



By TREVON LONG



Posted on August 16, 2016



<https://eftm.com/2016/08/how-two-uni-students-built-a-better-census-site-in-just-54-hours-for-500-30752>

Credits: Prof- G. Moltò, UPV



- 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 is also 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



Systems are complex so...

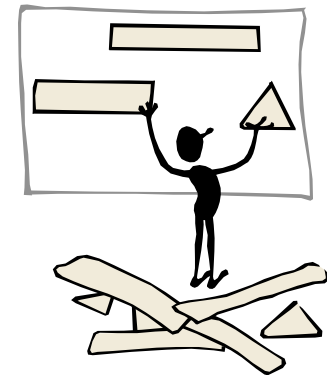
Abstraction of the systems: Models

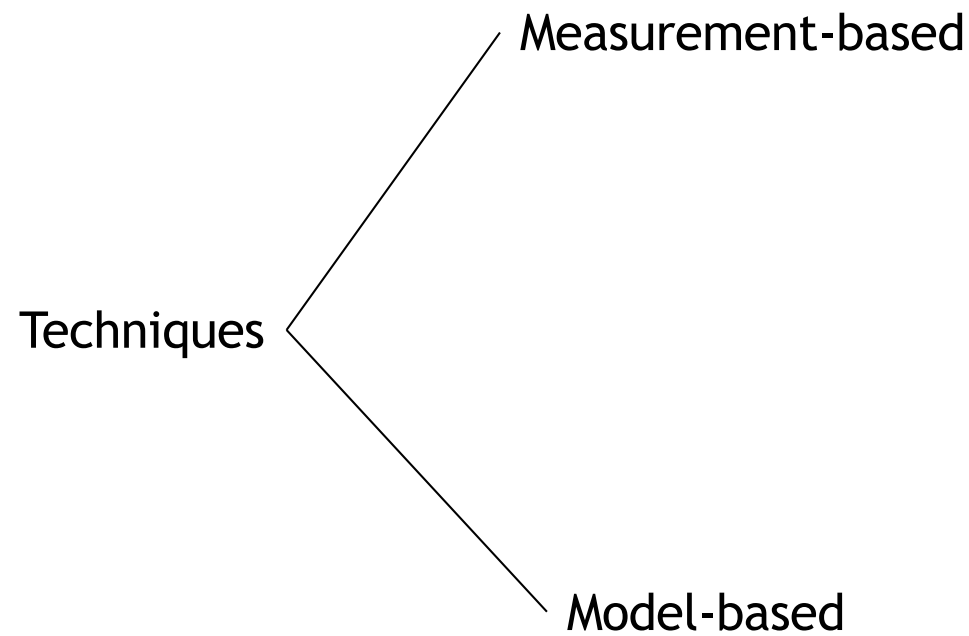
Often models are the only artifact to deal with!  
e.g., design phase

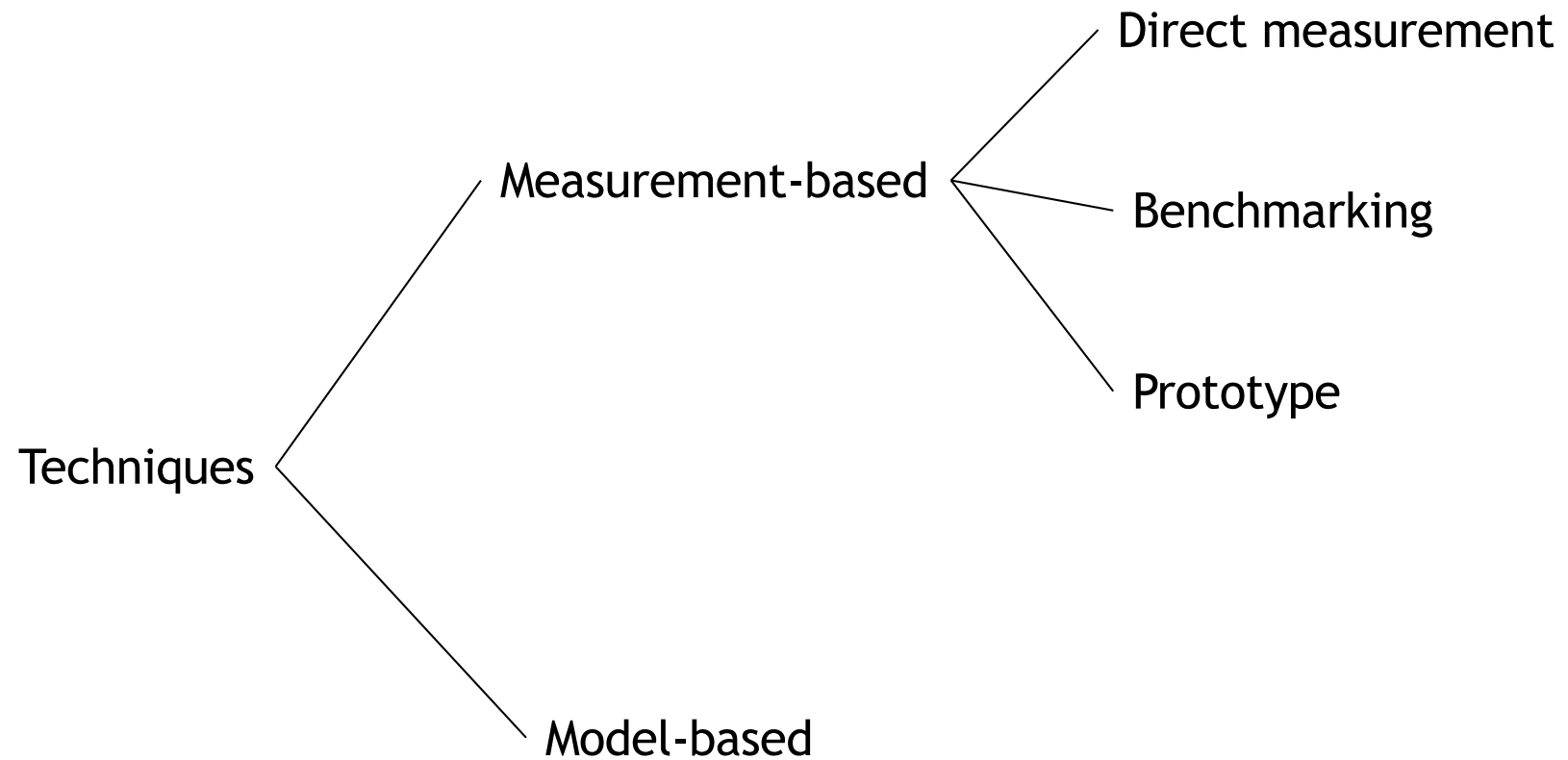
Models used to *drive* design decisions

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

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









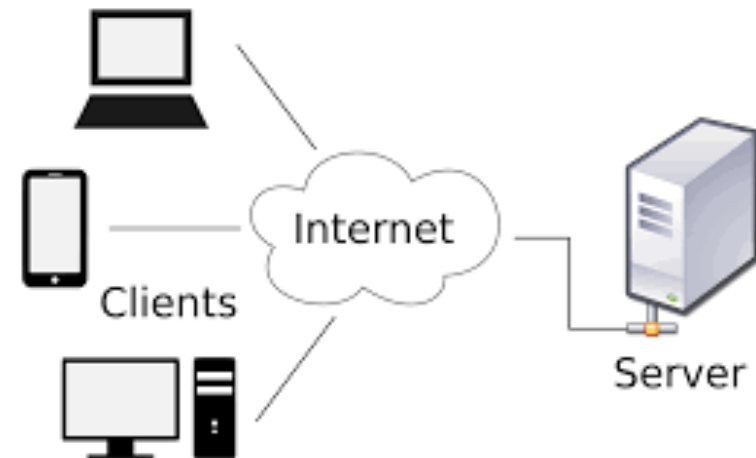
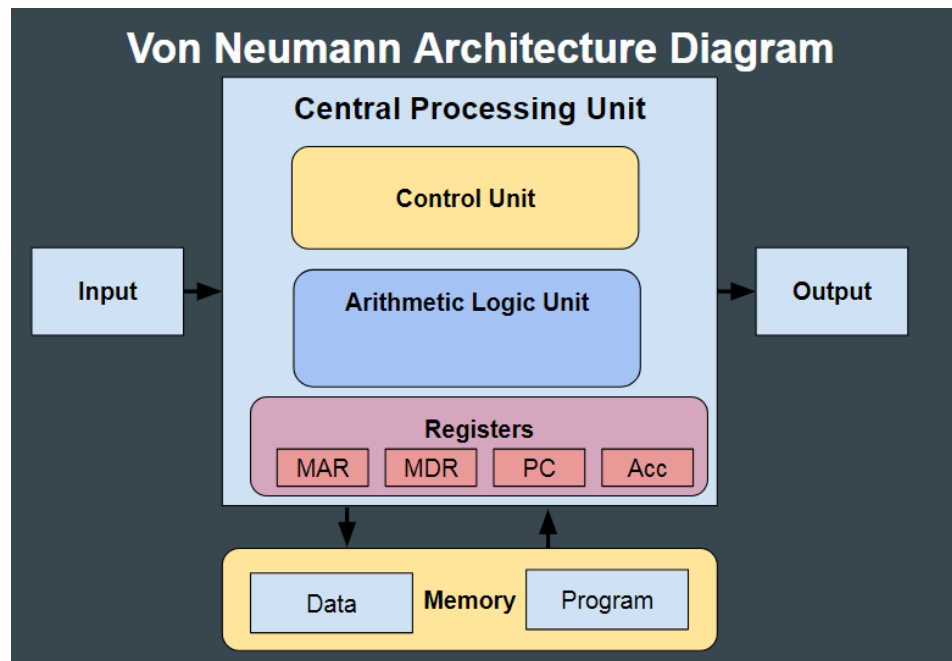


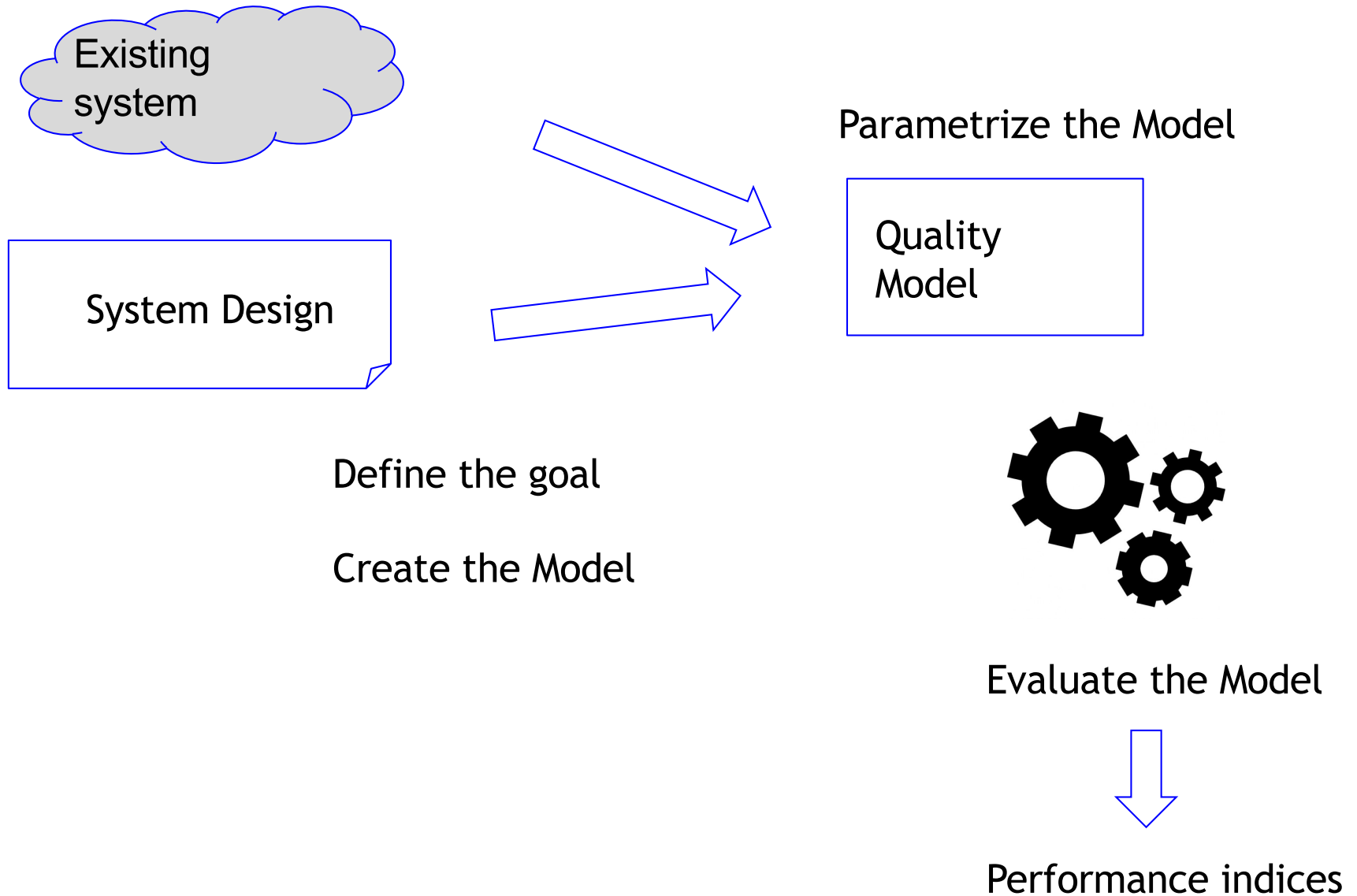
# What is a model?

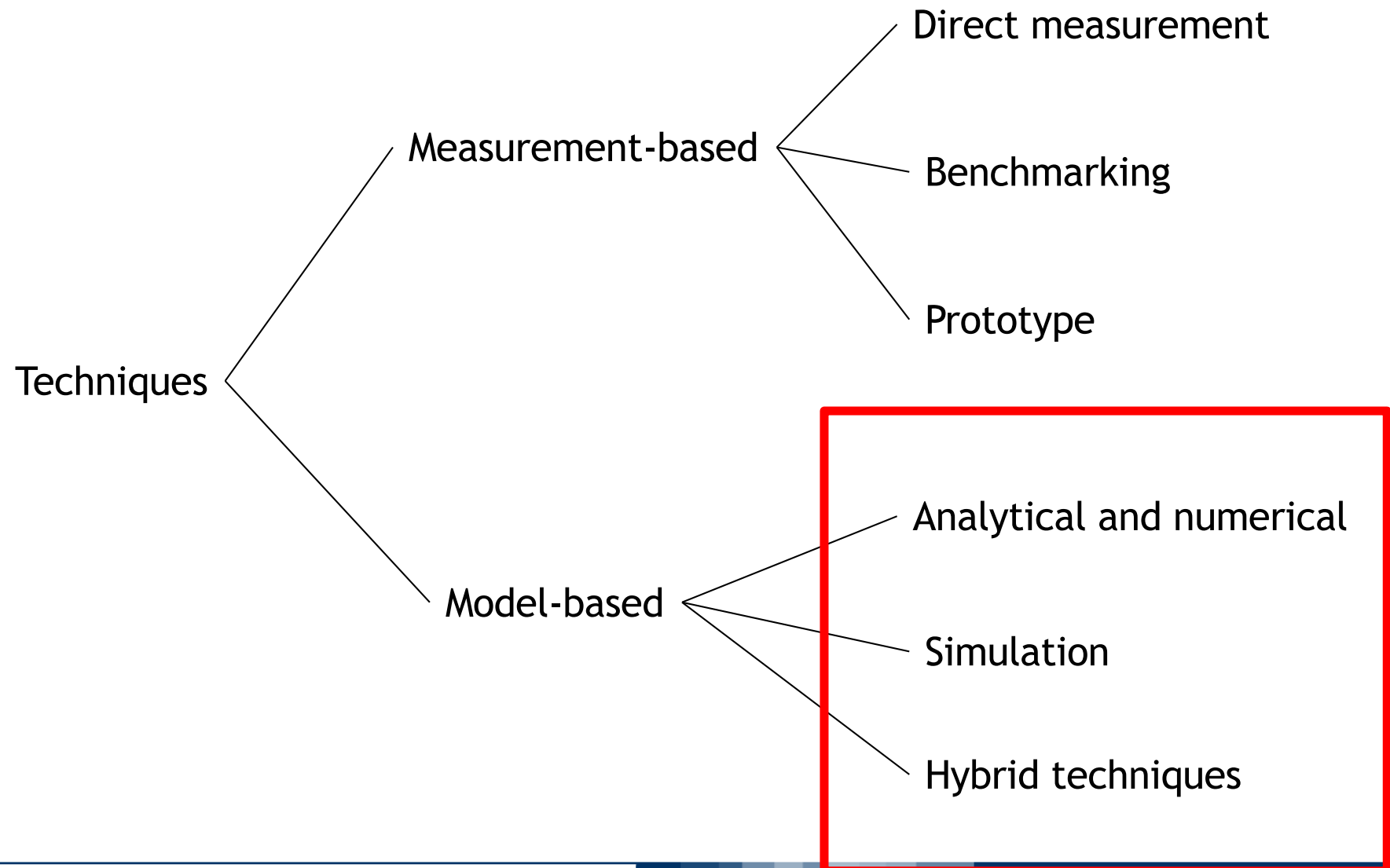
17

A representation of a system that is

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









- **Analytical and Numerical techniques** are based on the application of mathematical techniques, which usually exploit results coming from the **theory of probability and stochastic process**
  - They are the most efficient and the most precise, but are available only in very limited cases
- **Simulation techniques** are based on the reproduction of traces of the model
  - They 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



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*

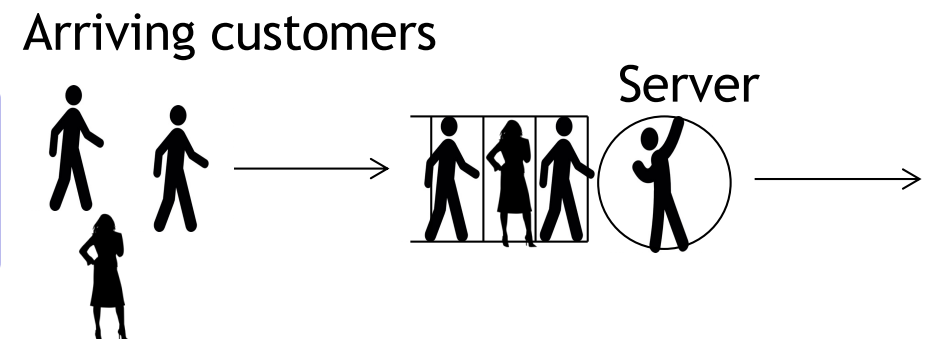
A network of queues is a collection of *service centers*, which represent system resources, and *customers*, which represent users or transactions





- Queueing 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

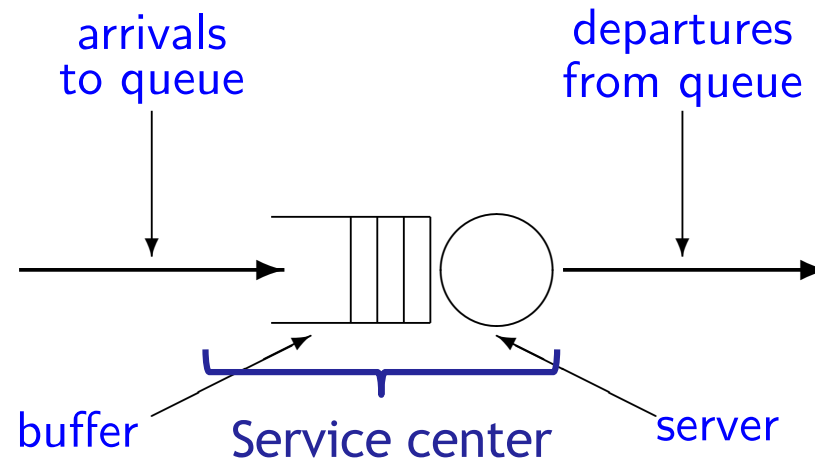
Success of queueing network: low-level details of a system are largely irrelevant to its high-level performance characteristics





## Single queue

23



- 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)





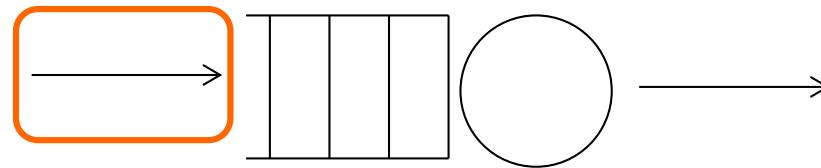
Different aspects characterize queuing models:

- Arrival
- Service
- Queue
- Population

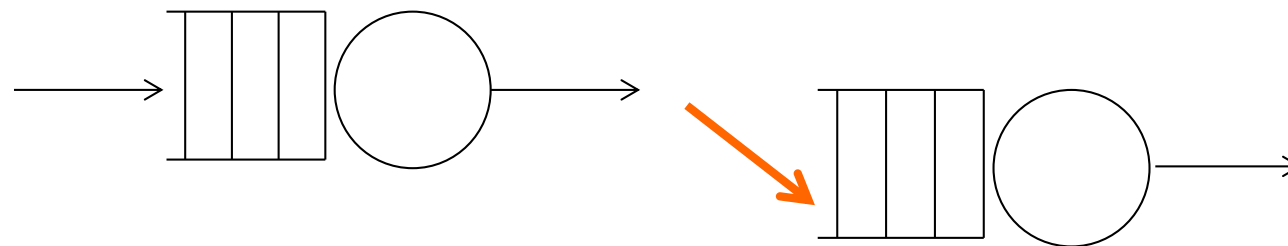


- Arrivals represent jobs entering the system: they specify how fast, how often and which types of jobs does the station service
- We are interested in the average arrival rate  $\lambda$  (req/s)

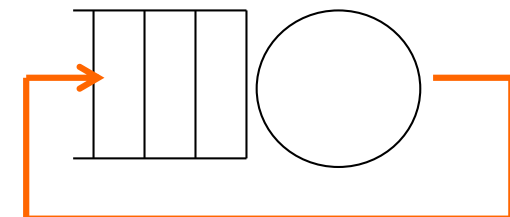
Arrival can come from an external source:



arrival can come from another queue:



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



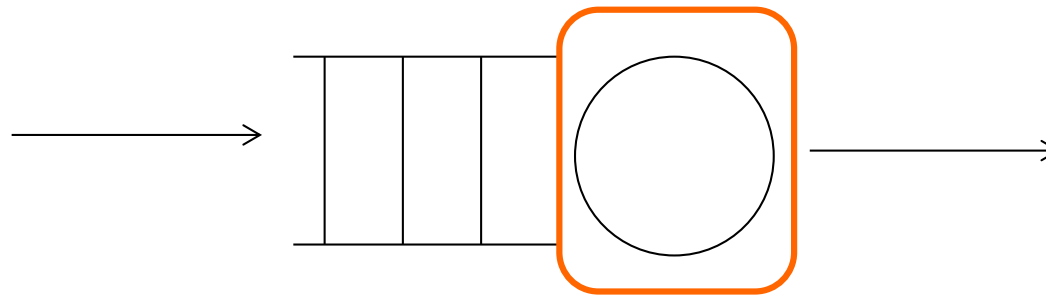


Different aspects characterize queuing models:

- Arrival
- **Service**
- Queue
- Population



*The service part represents the time a job spends being served*

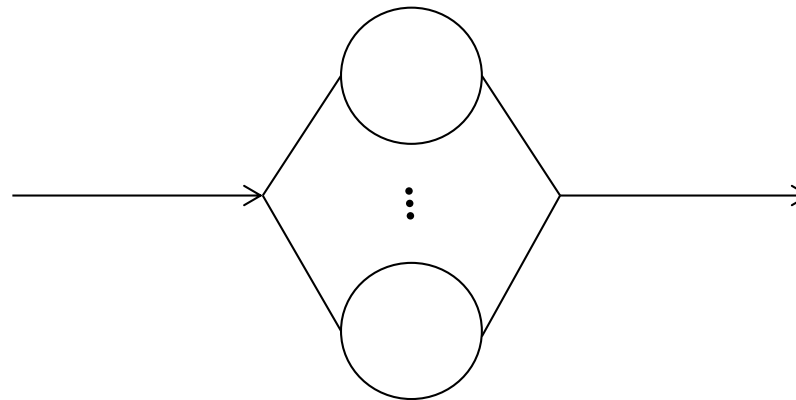


- The **service time** is the time which a server spends satisfying a customer
- As with the inter-arrival time, the important characteristics of this time will be its **average duration** (advanced: **the distribution function**)
- If the average duration of a service interaction between a server and a customer is  $1/\mu$  then  $\mu$  is the **maximum service rate**



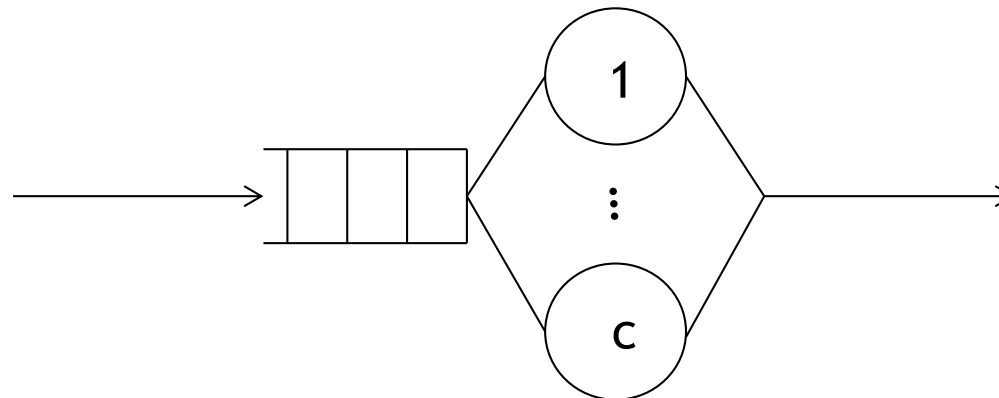
Possible situations:

- **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





- 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**





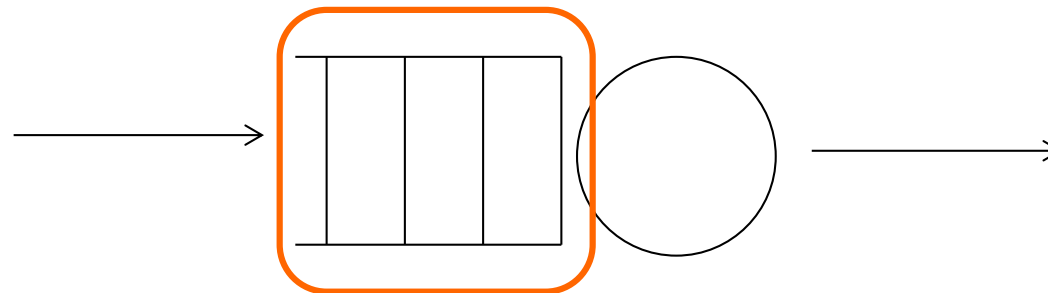
Different aspects characterize queuing models:

- Arrival
- Service
- Queue
- Population





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

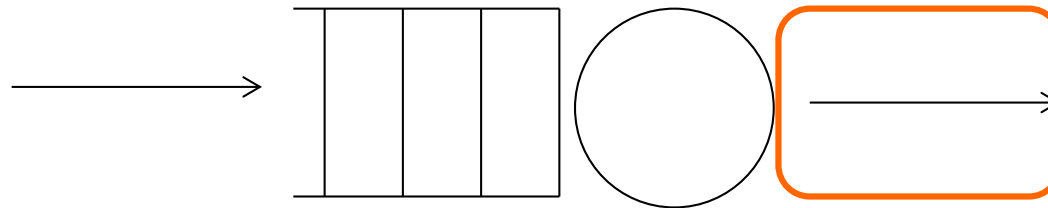




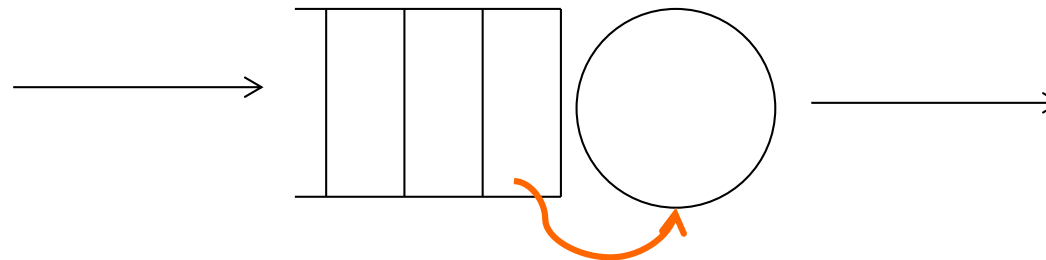
- Customers who cannot receive service immediately must wait in the **buffer** until a server becomes available
- If the buffer **has finite capacity** there are two alternatives for when the buffer becomes **full**:
  - the fact that the facility is full is passed back to the arrival process and **arrivals 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**



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





- 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



Different aspects characterize queuing models:

- Arrival
- Service
- Queue
- 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**



- 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





- 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

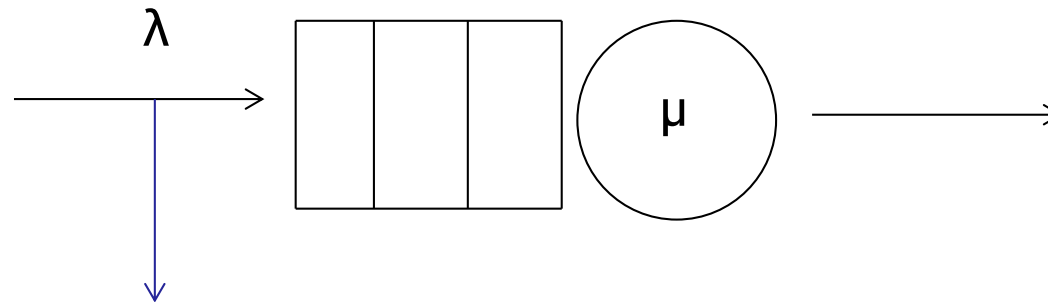


## Example

39

A single queue center with:

- Finite queue capacity=13
- FCFS service discipline
- Arrival rate  $\lambda = 125$  req/s
- Service rate  $\mu = 1/(2\text{ms}) = 500$  req/s

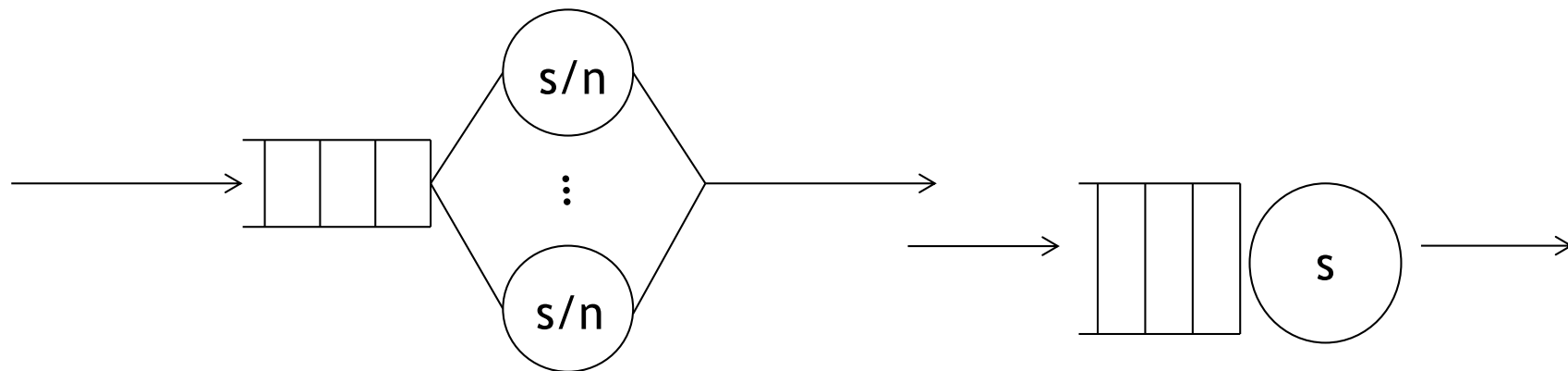




## Example

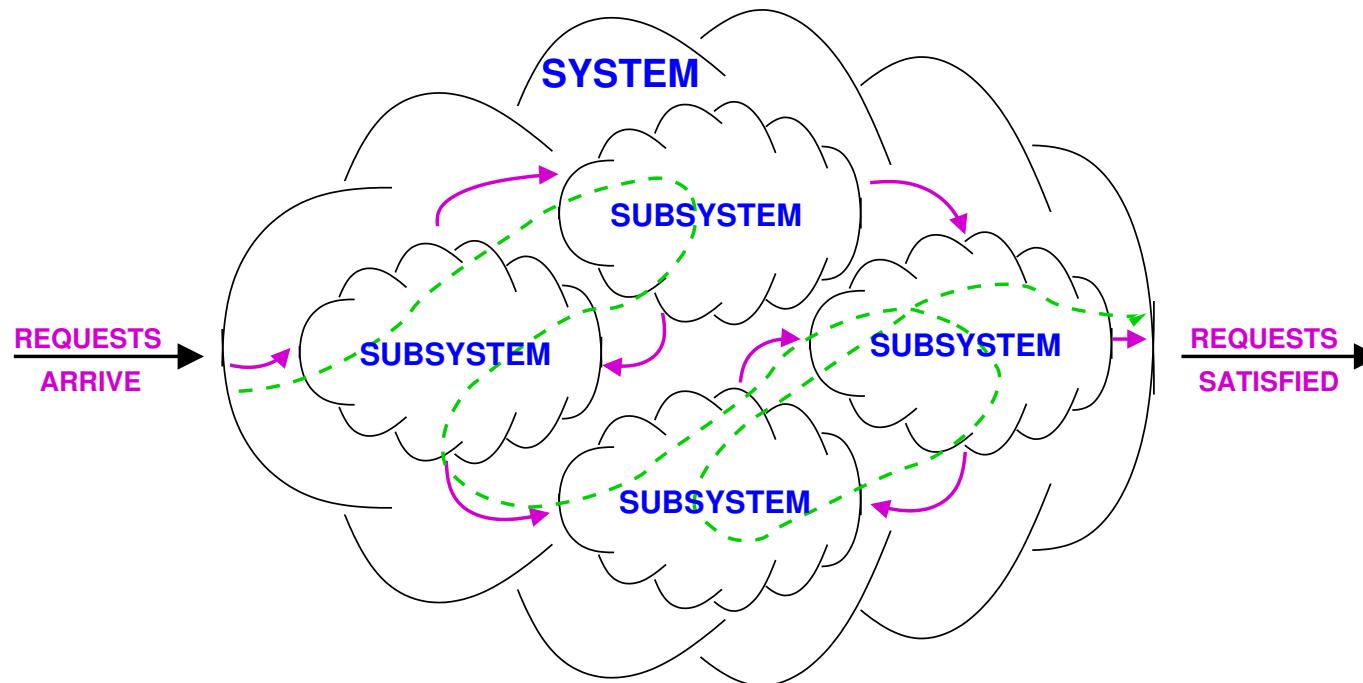
40

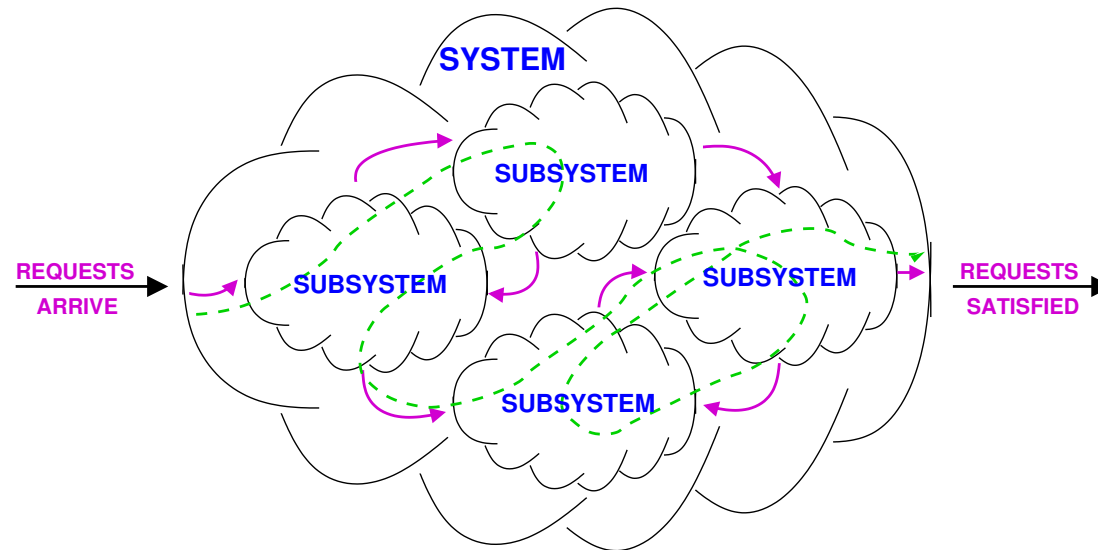
- You are given a choice between **one** fast CPU of speed  **$s$**  (**maximum service rate  $\mu$** ) or  **$n$**  slow CPU each of speed  **$s/n$**  (**maximum service rate  $\mu/n$** ). Your goal is to minimize mean response time
- Question: Which is the better choice? (Assume FCFS)
  - Arrival rate? Job type?
  - Pre-emption?





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



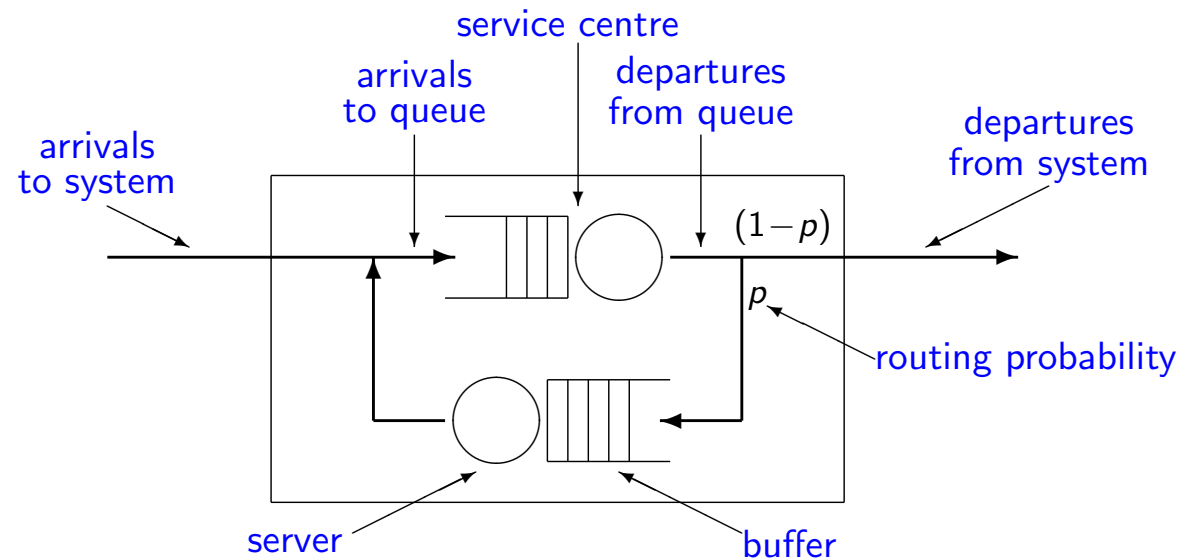


- We can associate a service center with each resource in the system and then route customers among the service centres
- After service at one service centre a customer may progress to other service centres, following some previously defined pattern of behaviour, corresponding to the customer's requirement



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



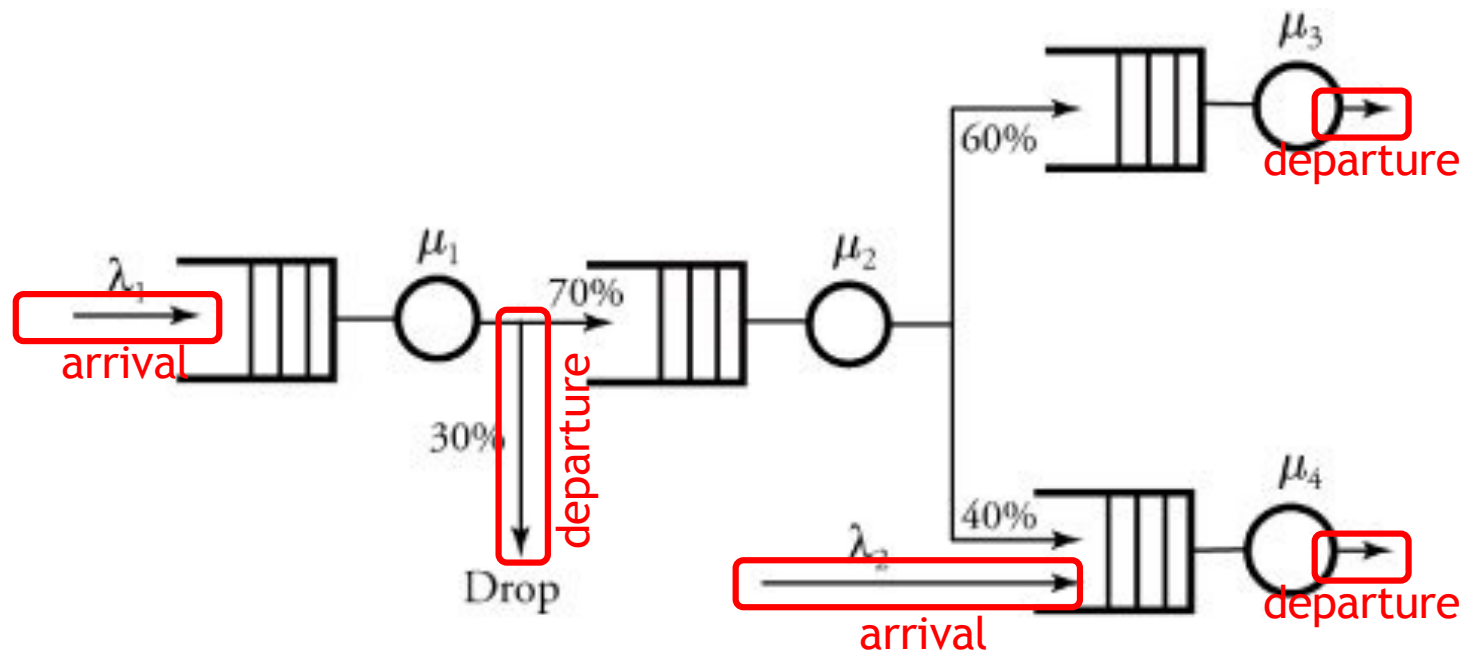


A network may be:

- **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



*Open Models* are characterized by *arrivals* and *departures* from the system

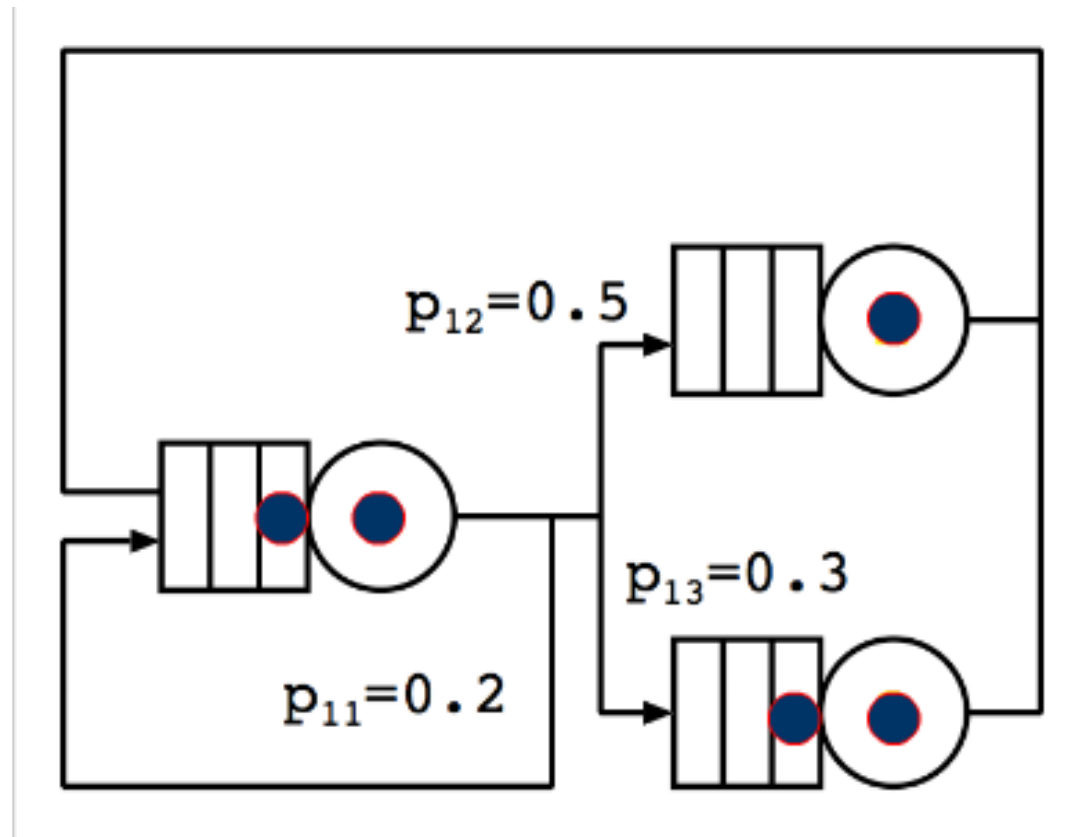






In closed models we have a parameter  $N$  that accounts for the fixed population of jobs that continuously circulate inside the system

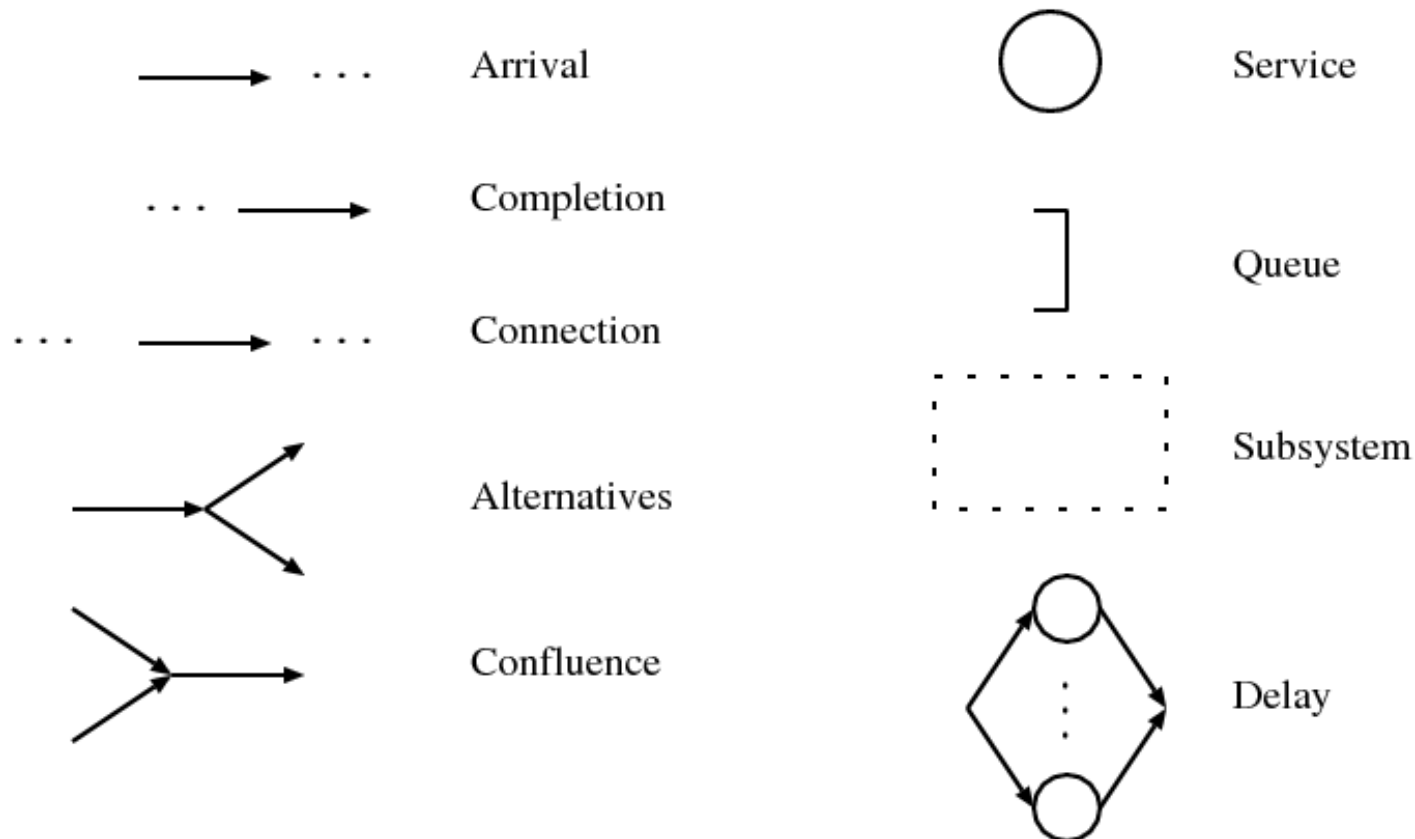
$N=5$





## Graphical notation

47



- Graphical notation is not unique, but it usually corresponds to a graph where edges denotes the flow of customers in the network

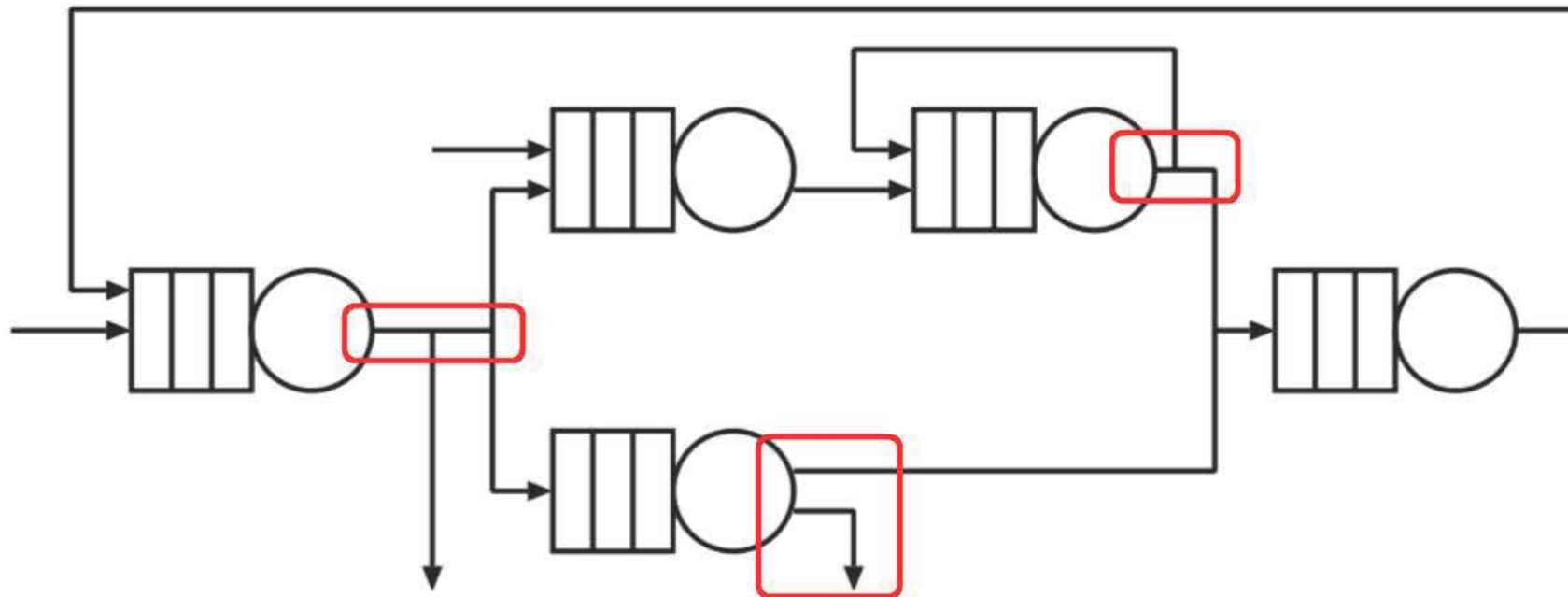


Different aspects characterize queuing models:

- Arrival
- Service
- Queue
- Population
- 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

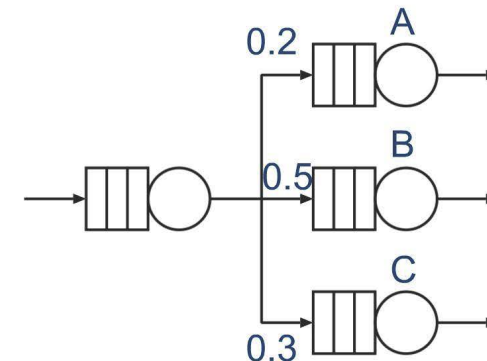




The main routing algorithms that we will consider are:

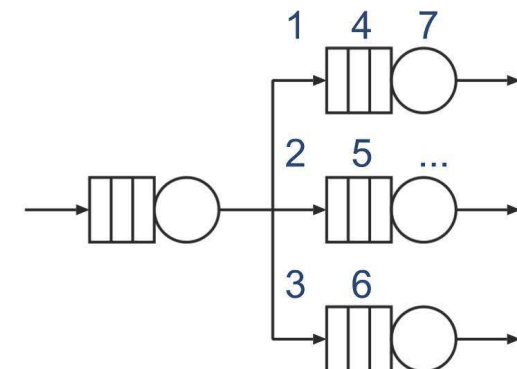
## Probabilistic

- each path has assigned a probability of being chosen by the job that left the considered station



## Round robin

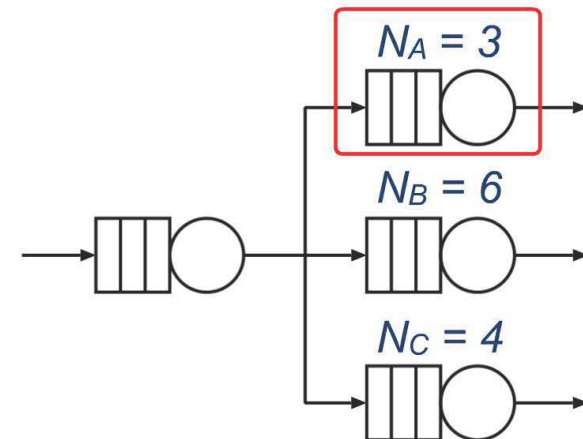
- the destination chosen by the job rotates among all the possible exits

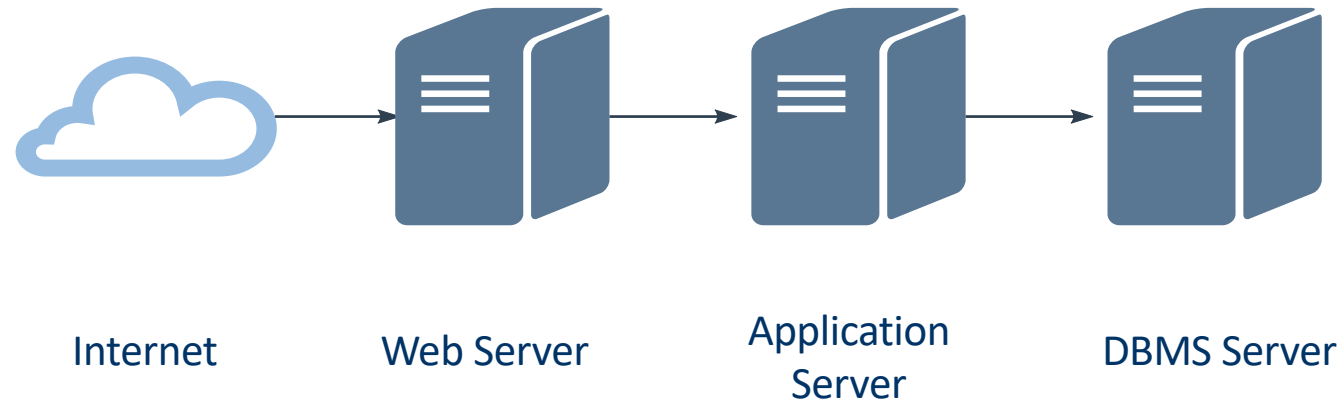




## Join the shortest queue

- jobs can query the queue length of the possible destinations, and chose to move to the one with the smallest number of jobs waiting to be served



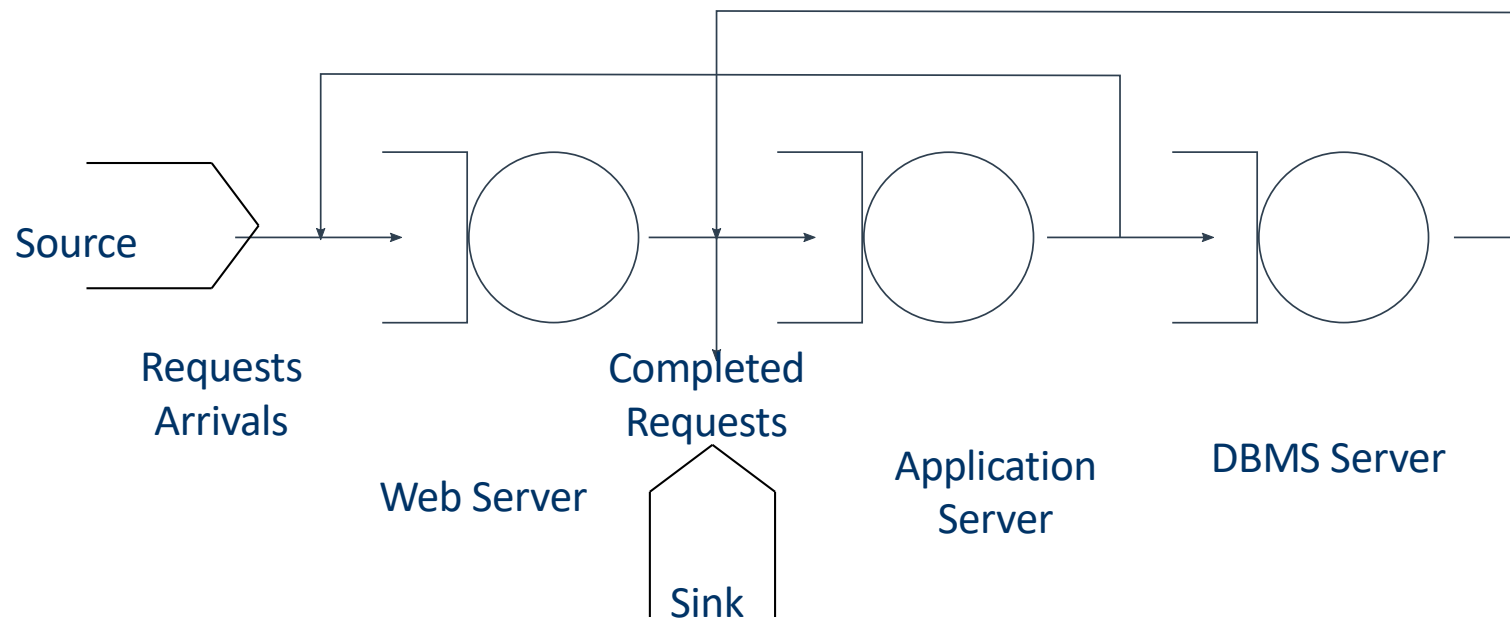
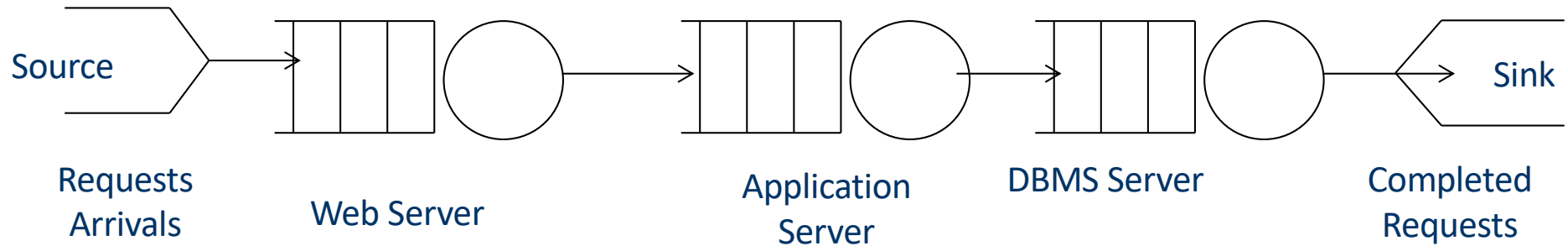


A client server system, dealing with external arrivals, which is architected with three tiers: the first one includes one web server, the second tier includes one application server and the third one includes a database server

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 server** 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**



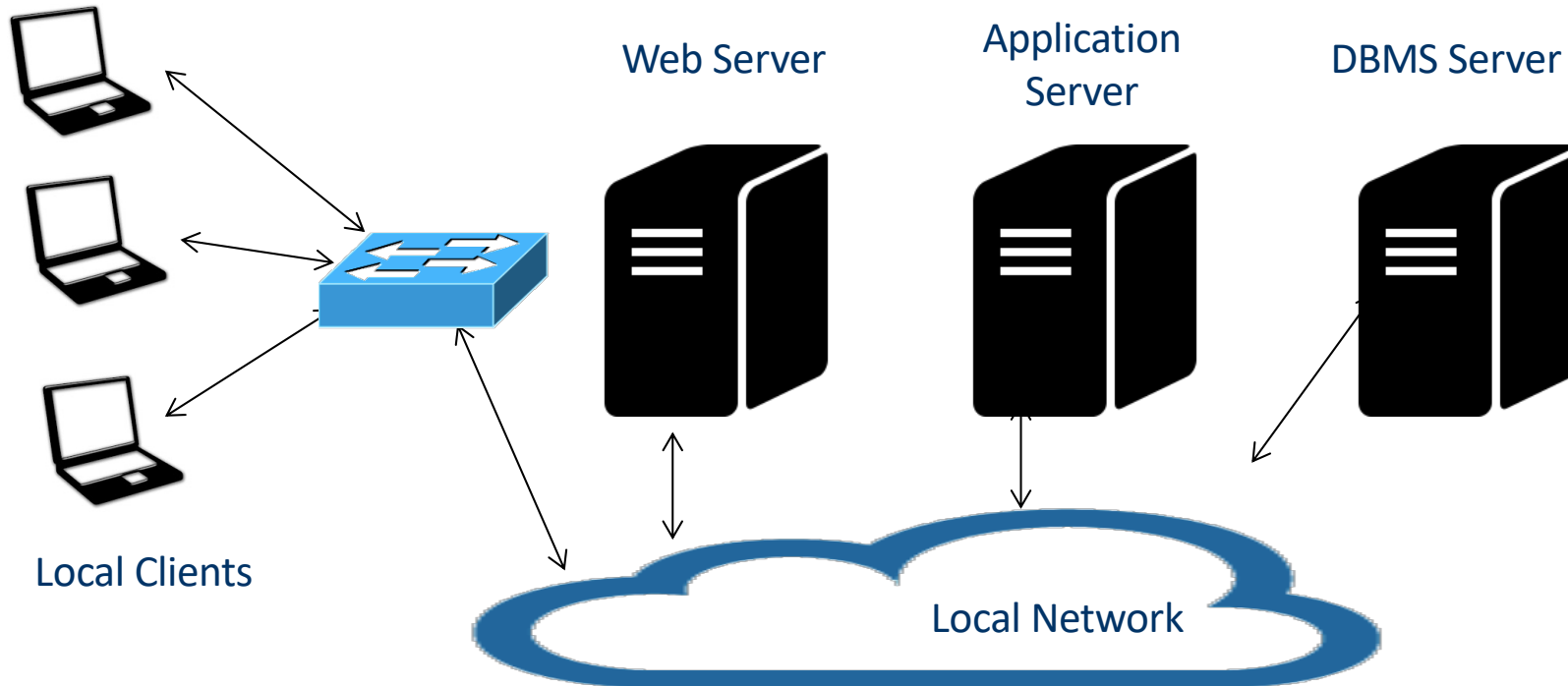




## Scenario 1: Tandem networks

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)





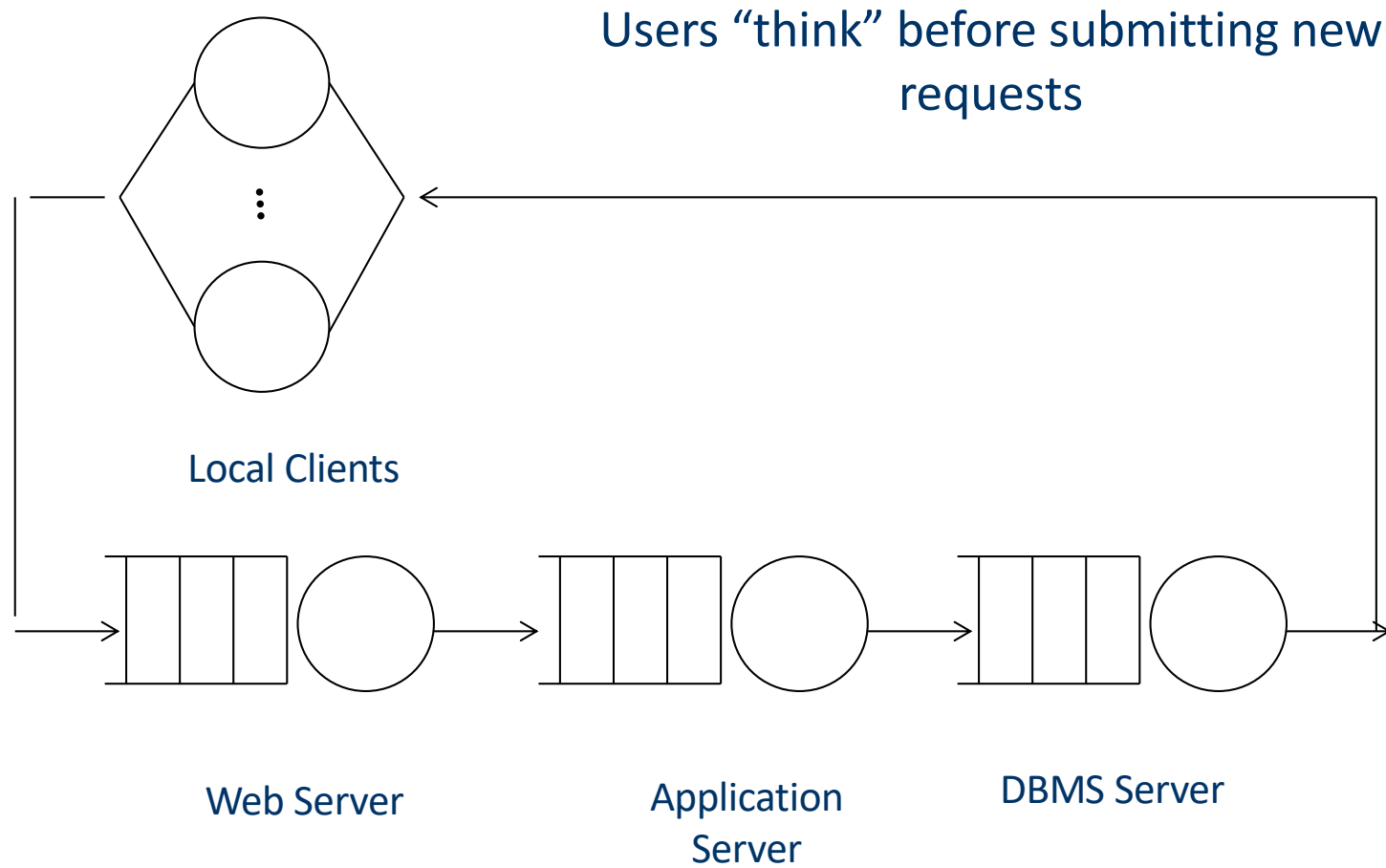
A client server system, **with a finite number of customers**, which is architected with three tiers: the first one includes one web server, the second tier includes one application server and the third one includes a database server.

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)

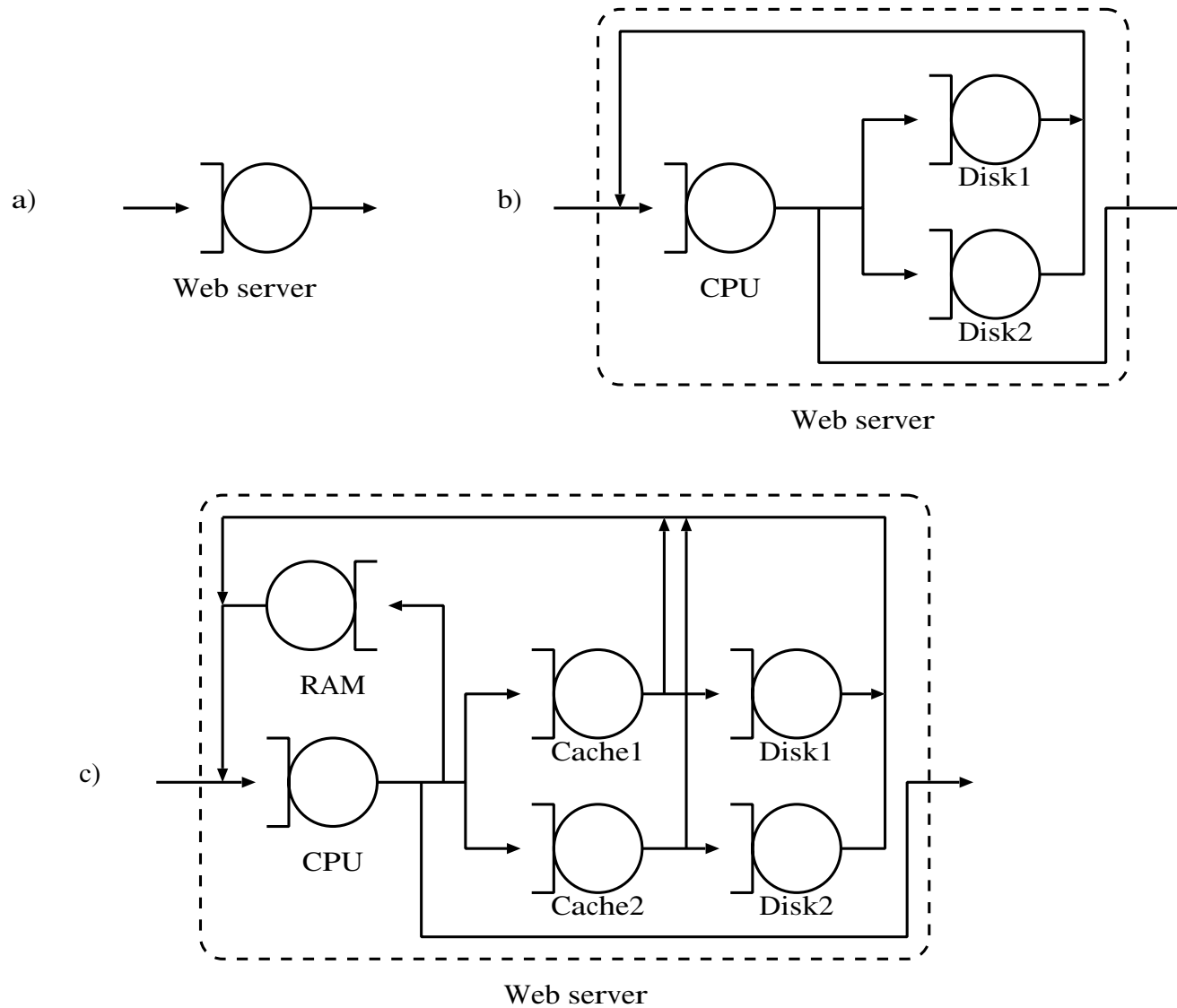
57





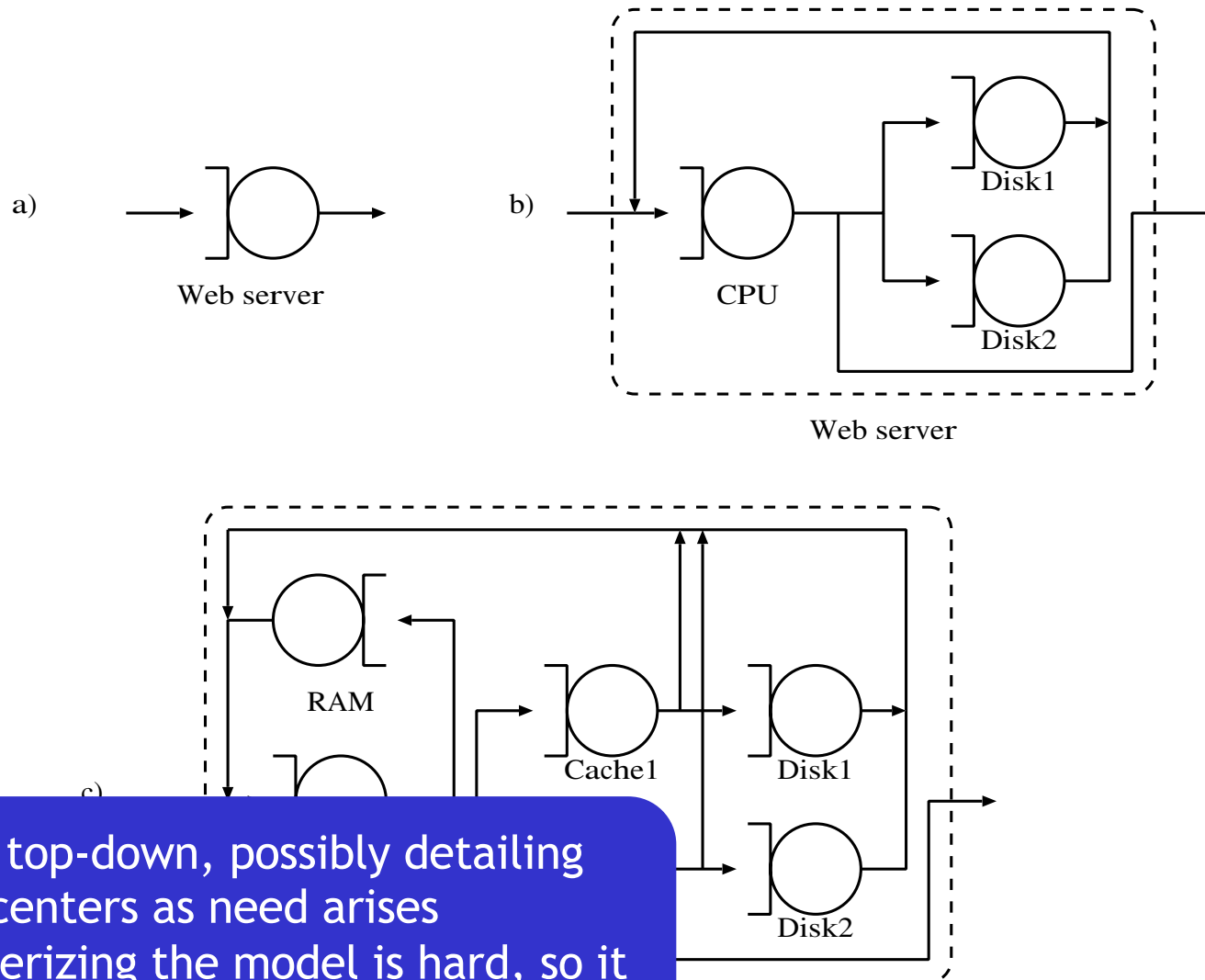


# Level of Detail





# 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