Performance Modeling

Software Design

2DV608

Diego Perez

Department of Computer Science and Media Technology

diego.perez@lnu.se

Credits: Raffaela Mirandola

Assignment

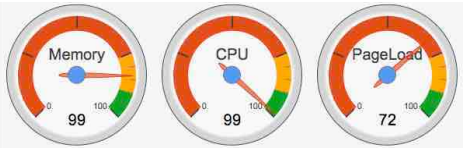
- During Wednesday 4th March
- Submit before 23:59 CET
- Estimated work time (if you have studied the contents before March 4th): 1h
- The assignment will be a practical exercise where you need to apply the contents that you have seen during the lessons.

GOAL of performance modeling and analysis

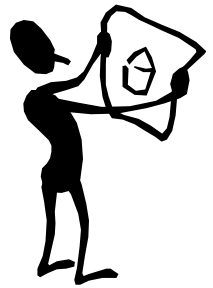
- Systems are expected to be powerful, highly available, scalable, well performing, green, cheap, open and secure at the same time.







- Designing systems that will satisfy these properties is difficult.
  - Even when the system under design is an evolution of an already running system.
  - And even the measurement of these properties in already developed systems is not easy.



GOAL of performance modeling and analysis

To provide **modeling principles** and simple **mathematical laws** to characterize and evaluate the performance of computing systems

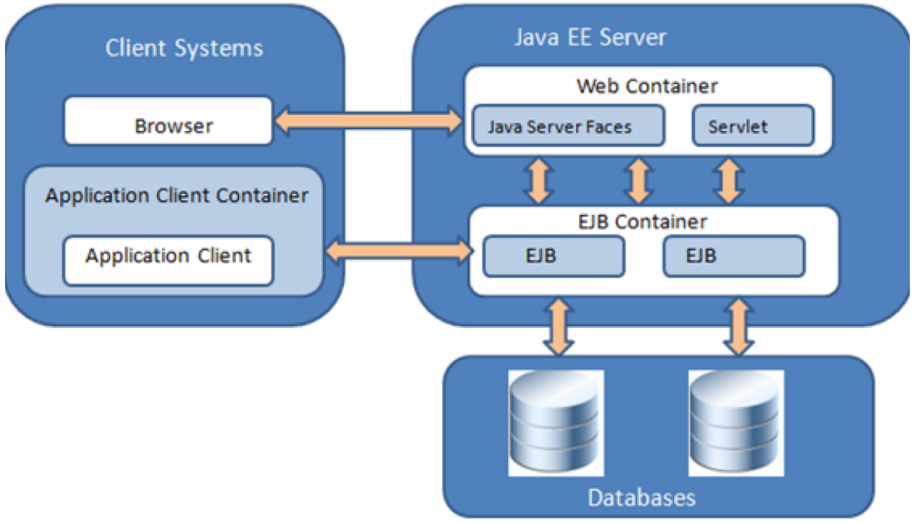
- They will be used for supporting system design and evolution:
  - Sizing of a computing infrastructure
  - Scalability
  - Evaluation of different design alternatives (predict the performance of a given design)
  - Renegotiate/refine requirements
  - Get insights about the limits of a system

High level example for performance

- Client/server system:

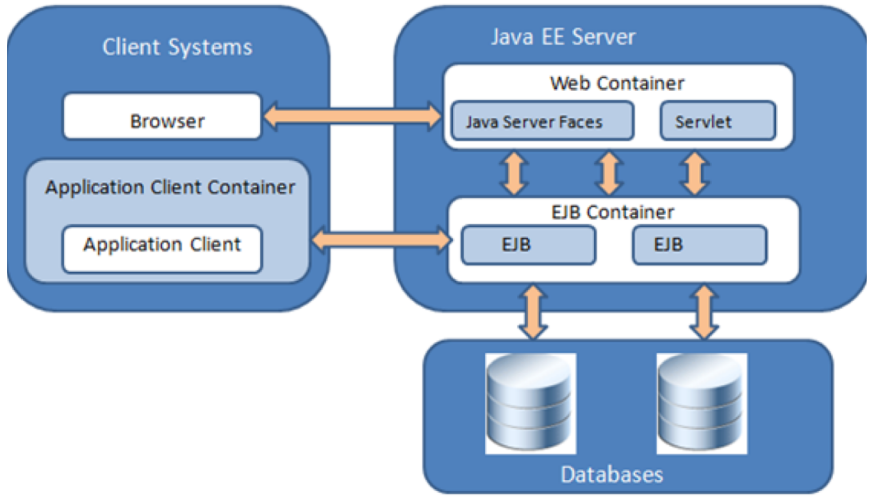
- Application response time?
- Resource utilization?

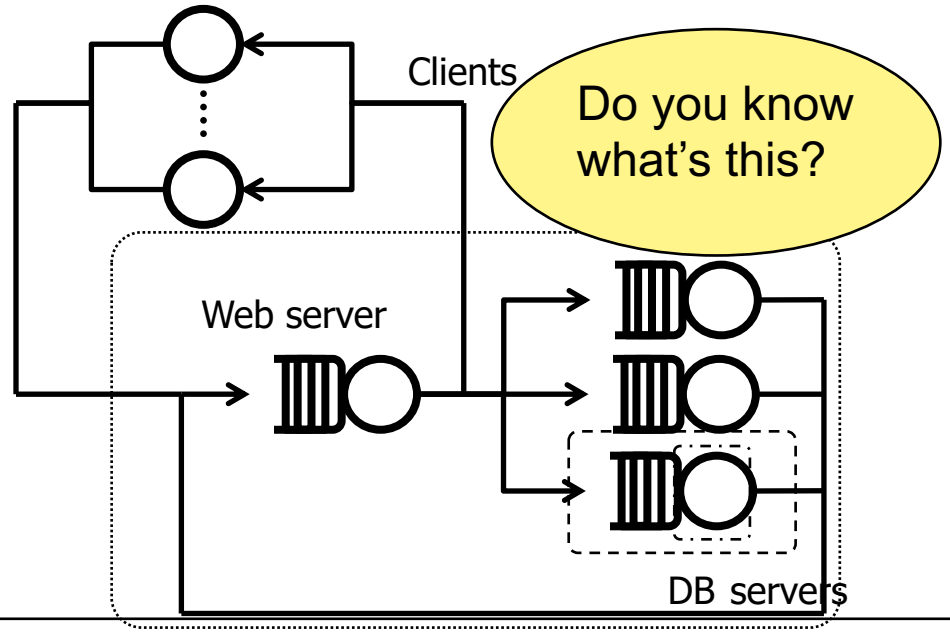
- Single centralized DB?
- Distributed DB?



High level example for performance

- Client/server system:
- Application response time?
- Resource utilization?
- Single centralized DB?
- Distributed DB?
- Performance Model Example:





Definitions

- **Computer performance**: The total effectiveness of a computer system, including throughput, individual response time and availability.

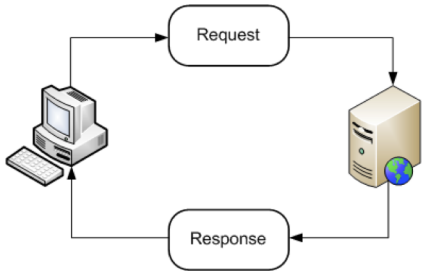
- **Computer performance:** is characterized by the amount of useful work accomplished by a computer system or computer network compared to the time and resources used.

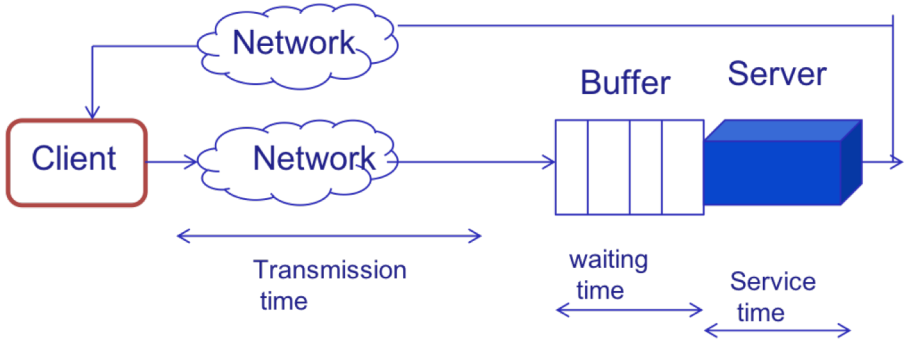
Examples of observed performance indices

- Response time of an application
- Throughput 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

Performance Indices

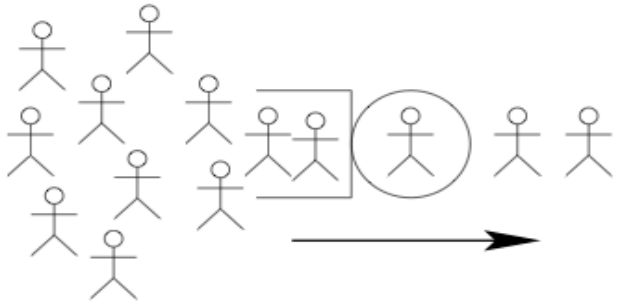
- **Response time**: is the total amount of time the system takes to respond to a request for service. The response time is the sum of:
  - Service time - How long it takes to do the work requested.
  - Wait time - How long the request has to wait for requests queued ahead of it before it gets to run.
  - Transmission time – How long it takes to move the request to the computer doing the work and the response back to the requestor.





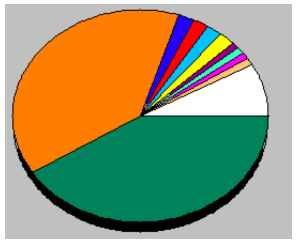
Performance indices

- **Throughput**: is the rate of production or the rate at which something can be processed



(b) Throughput

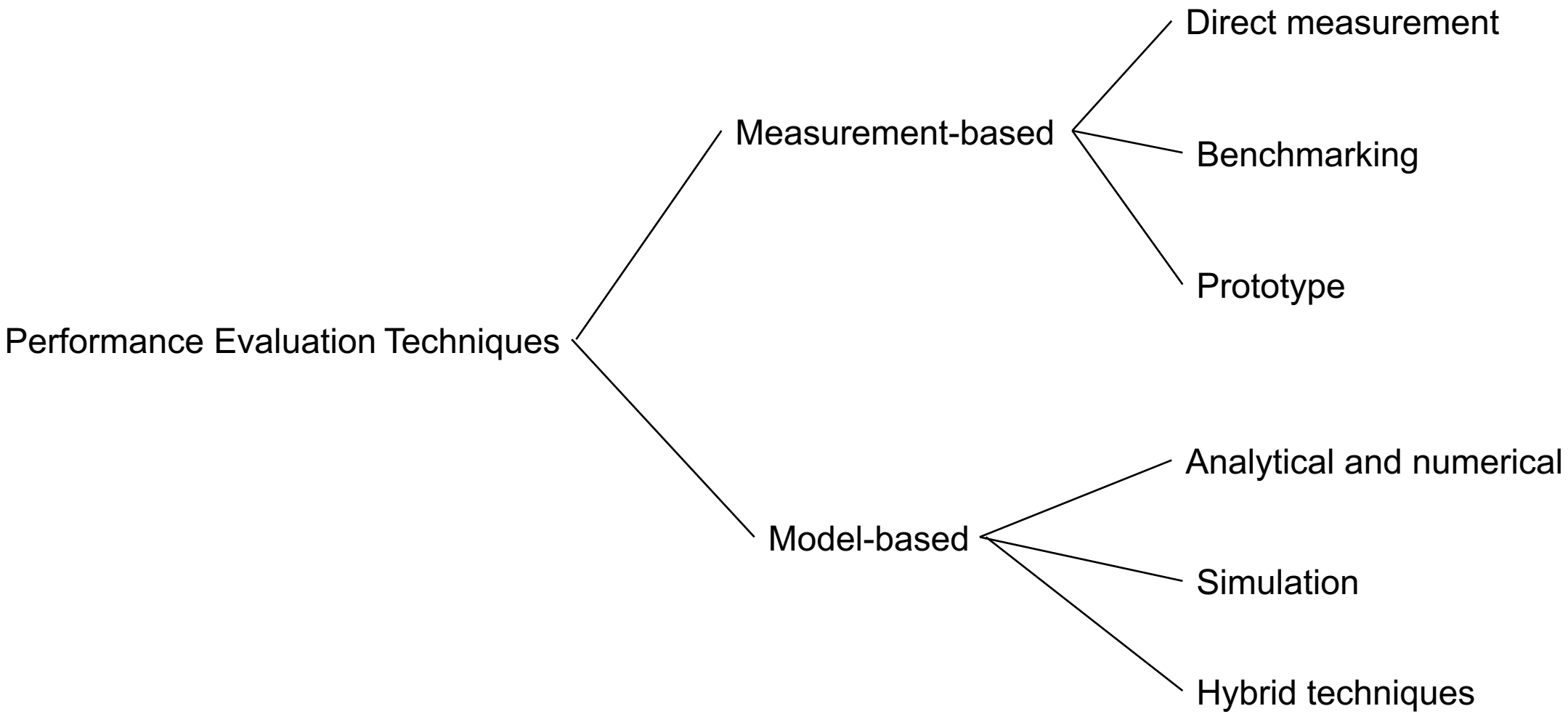
- **Resource Utilization**: Proportion of time in which the resource is used



Performance interests

- There are often conflicting interests at play:
  - Users typically want to optimize external measurements of the dynamics such as response time (as small as possible), throughput (as high as possible) or blocking probability (preferably zero);
  - In contrast, system managers may also seek to optimize internal measurements of the dynamics such as utilization (reasonably high, but not too high), idle time or failure rates (as low as possible).

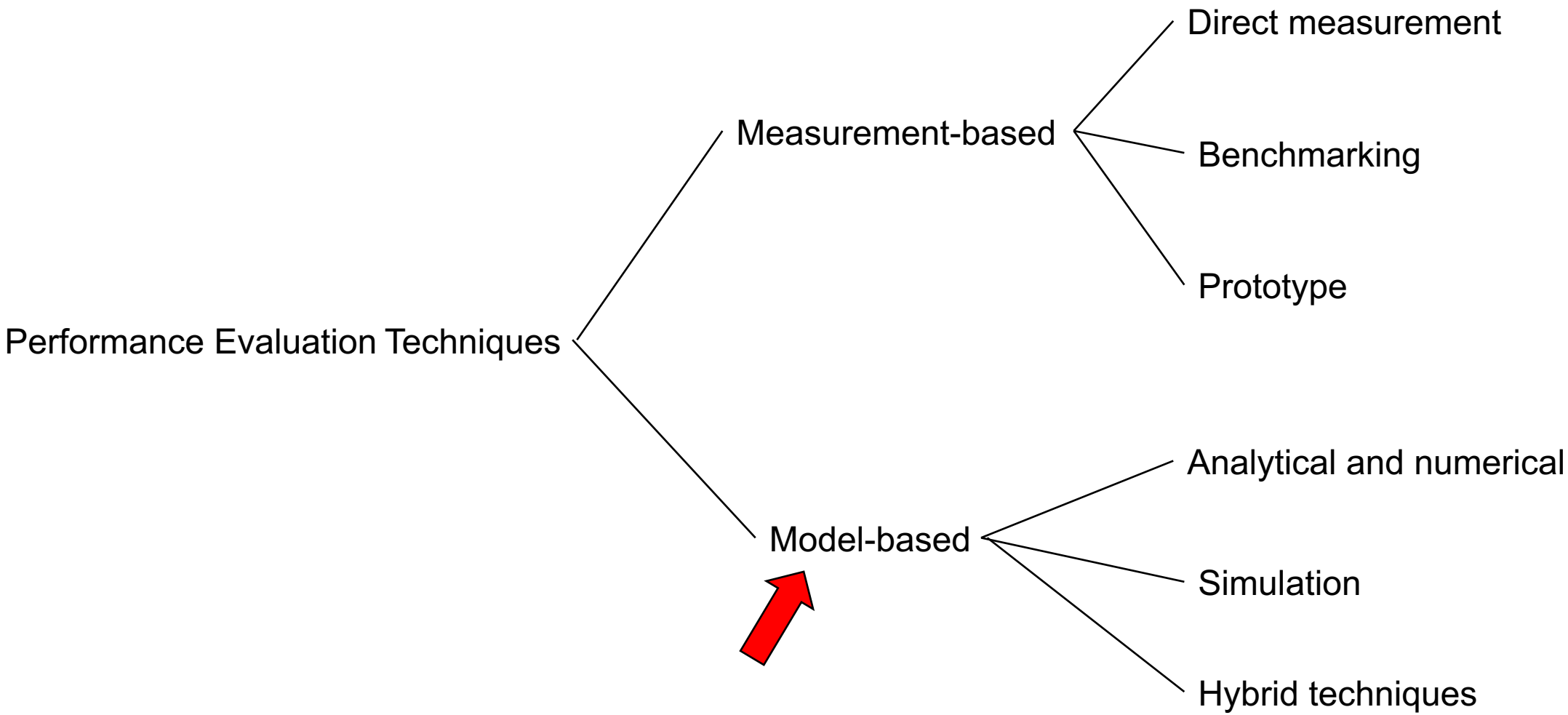
How to evaluate the performance of a software system?



Measurement-based performance evaluation

- Measurement through experimental evaluation
  - Experimentation is always valuable, often required, and sometimes the approach of choice.
  - It also is expensive - often prohibitively so.
  - A further drawback is that 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, maybe disruptive or dangerous.

How to evaluate the performance of a software system?



Model-based techniques

- **Analytical and numerical** techniques are both based on the application of mathematical techniques.
- The model is converted into a set of equations, and performance indices are computed by solving the equations.
- 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. It obtains precise results efficiently.
- 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 utilization of a service ”s” is equal to the product of the throughput of the system and the average demand for that service

Utilization Law: $U_r = XD_r$

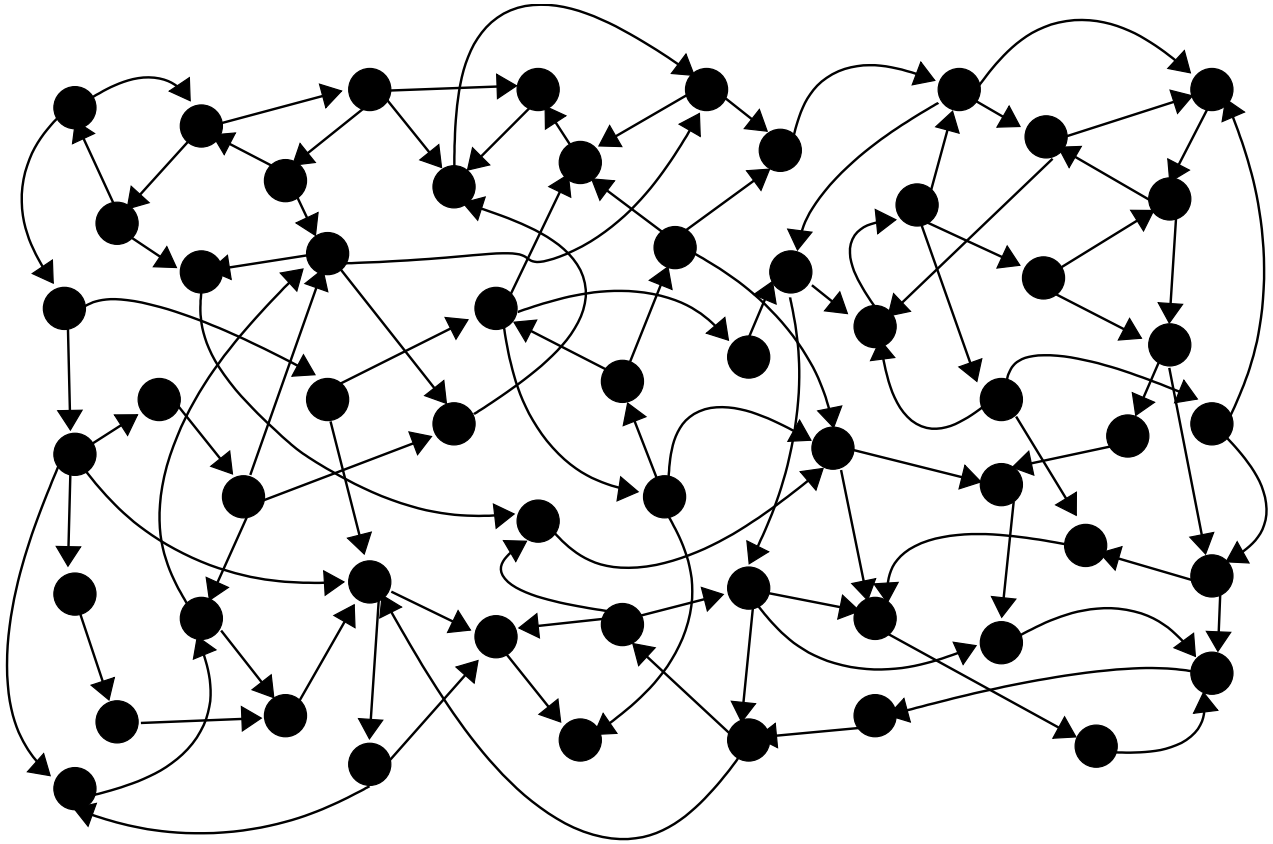
Case: A computer executes a certain task every 2 seconds.
That task needs 0.5 seconds of CPU.
What is the utilization of the CPU?

- Amdahl’s law

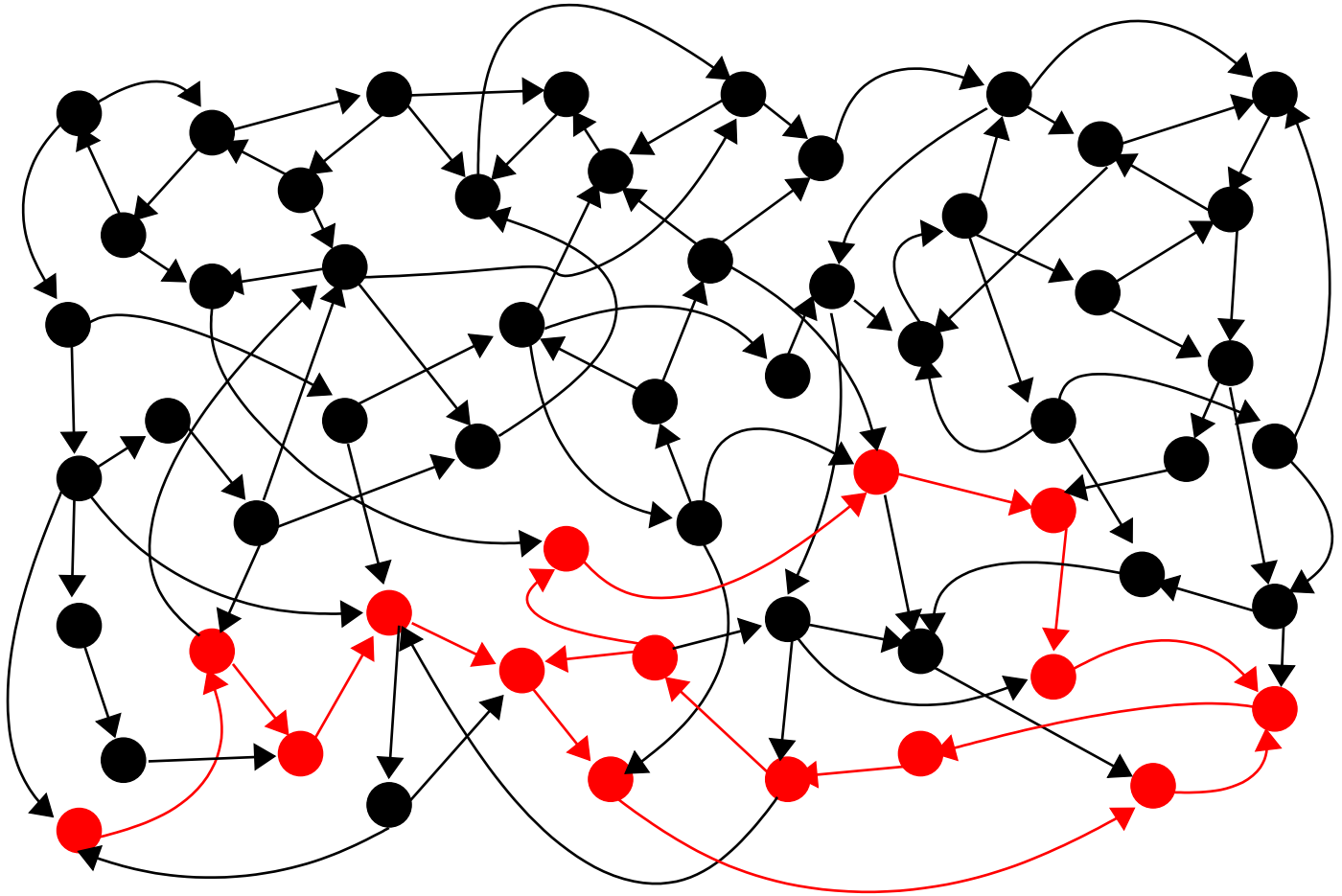
Simulation

- 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.

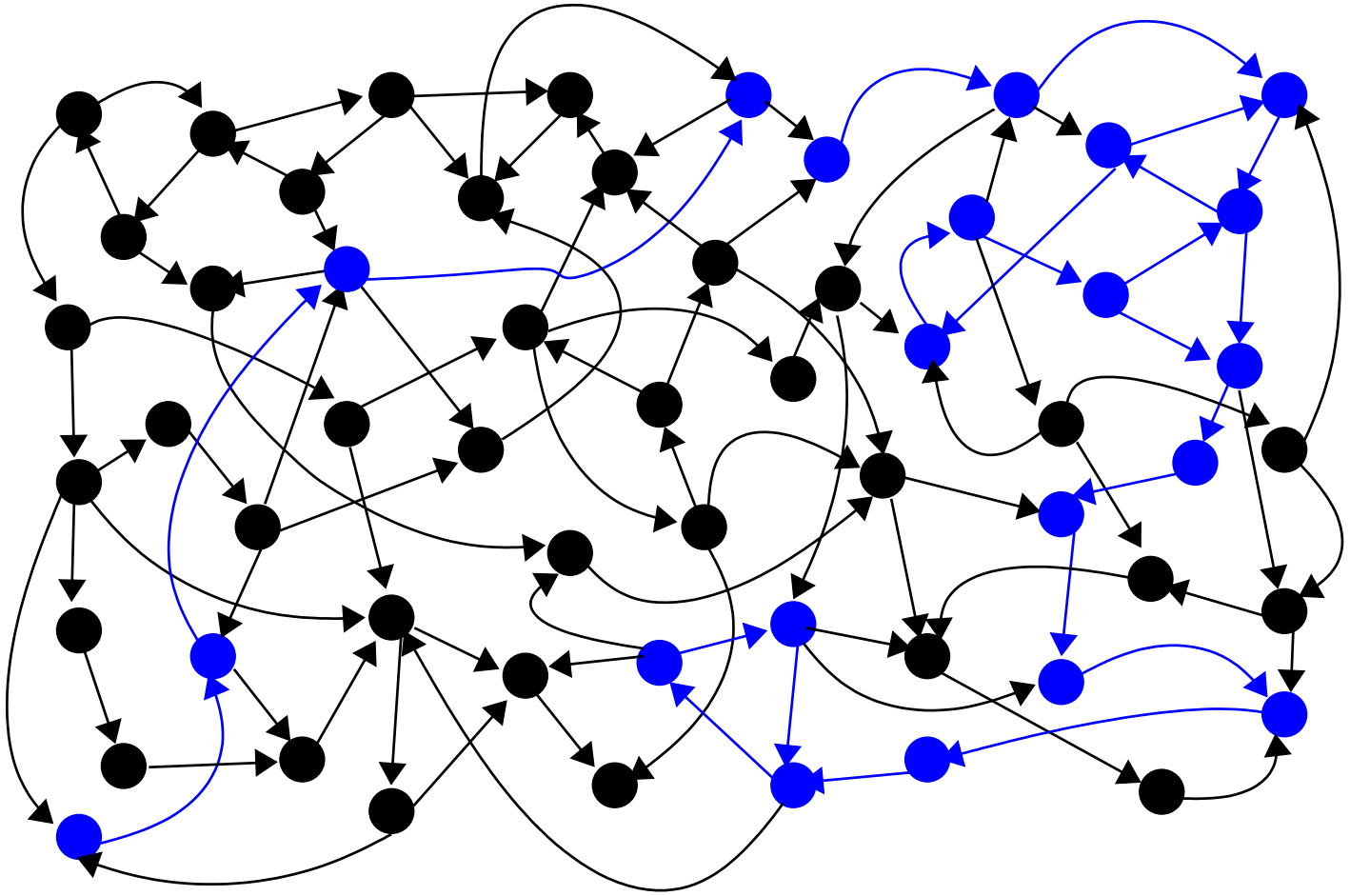
Sample traces



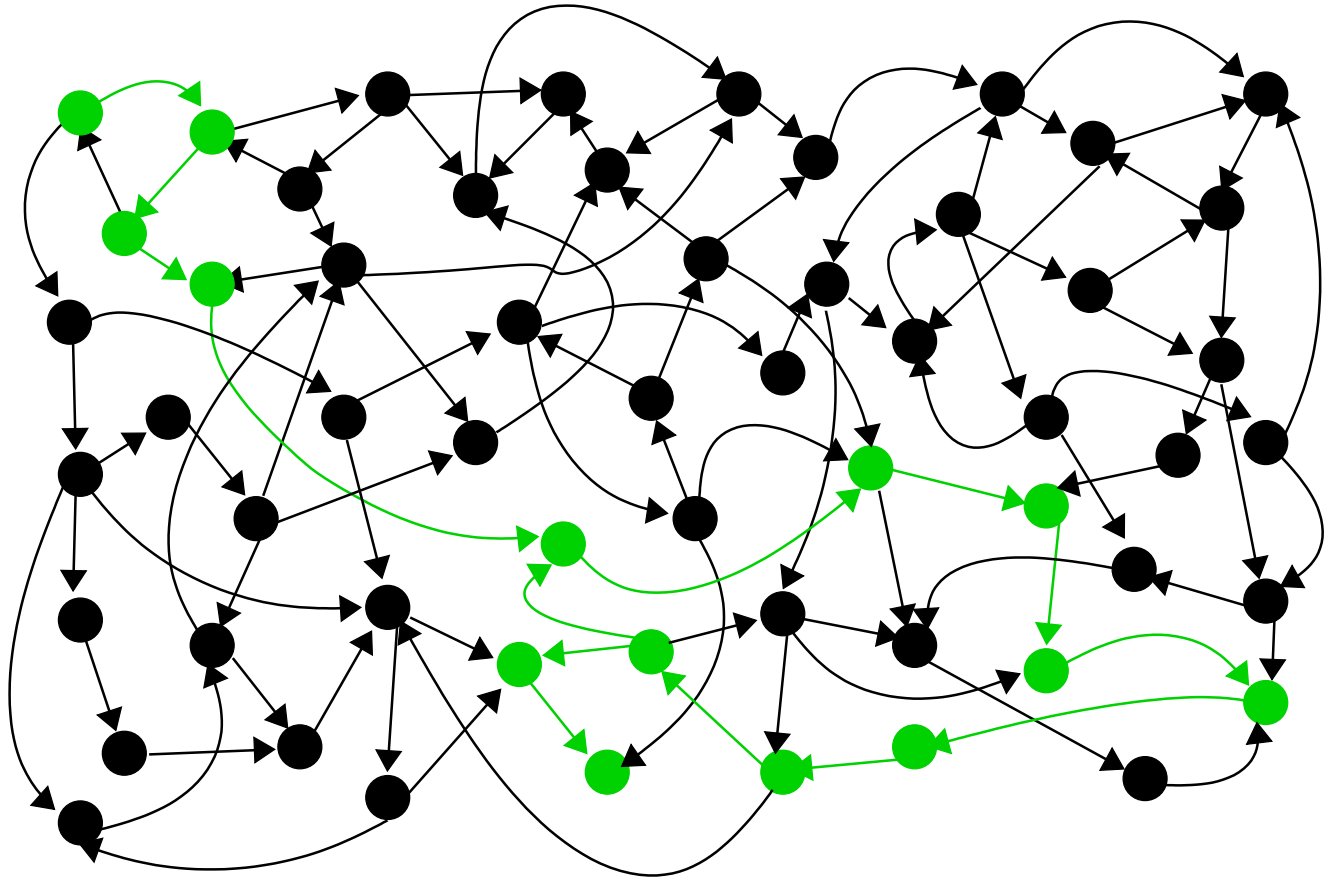
Sample traces



Sample traces



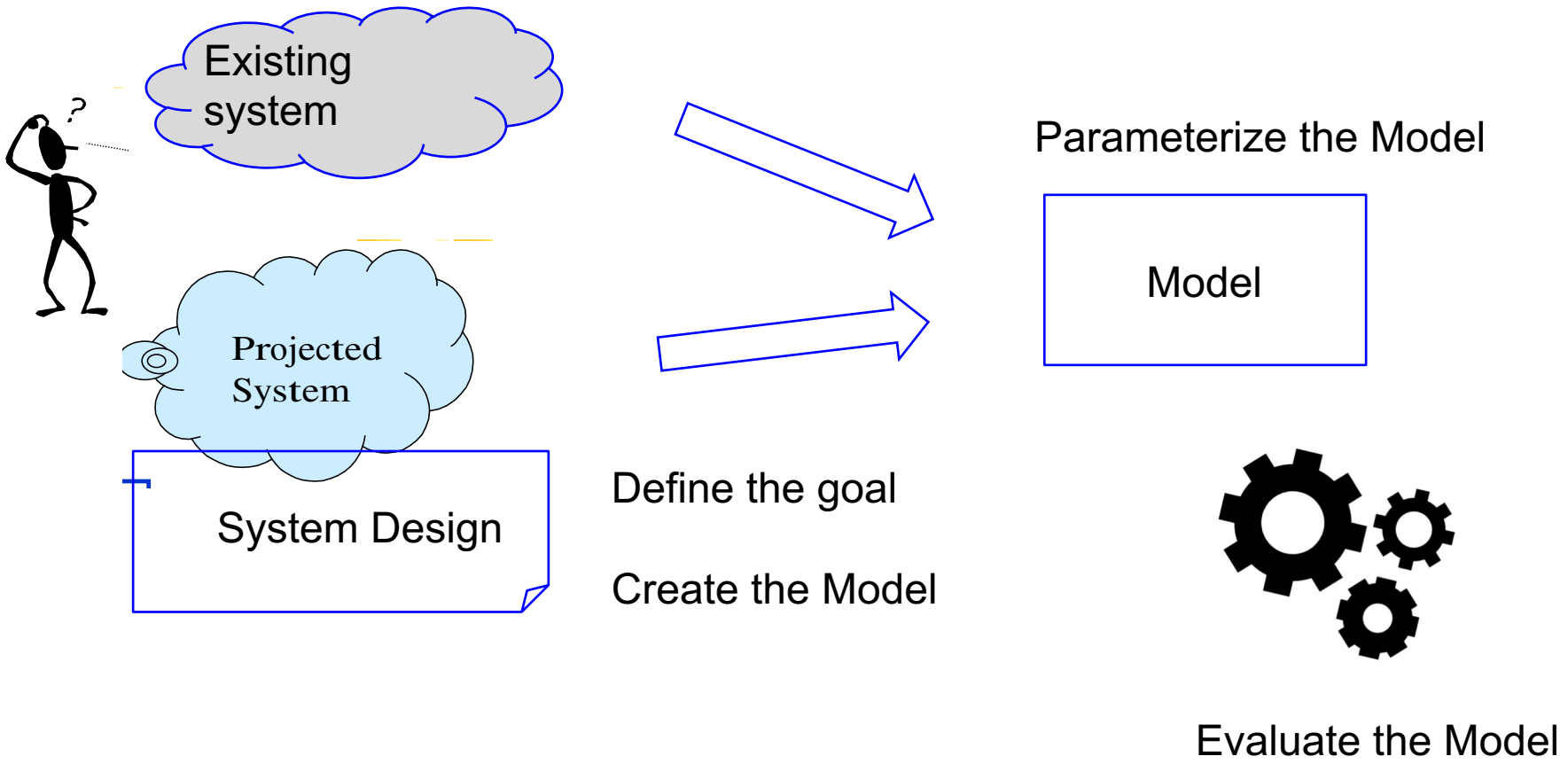
Sample traces



Summary of model-based analytical and simulation

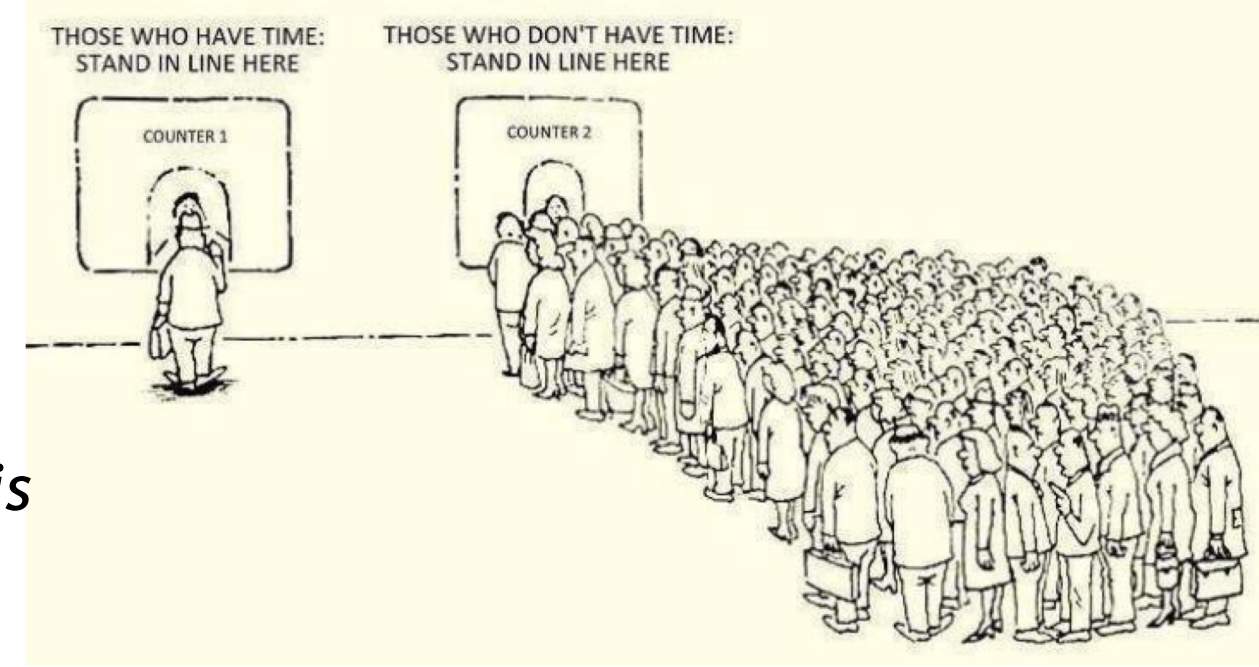
- Using an analytic approach we characterize all possible sample traces by solving the overall model.
    - It is the most precise and efficient method
- Using simulation we study the sample traces directly.
- Each run of the simulation model will generate another sample trace.
- Simulation can be used in more situations, where there does not exist mathematical theory to solve the problem analytically.

Example of utilization of measurements and model in a system upgrade

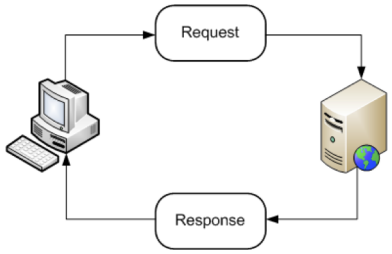


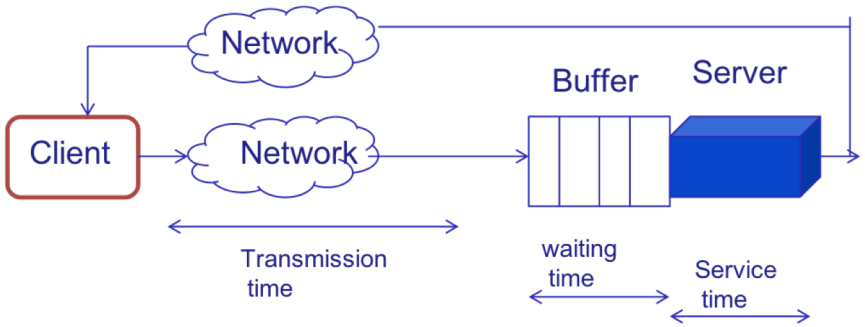
Create the Model

- Modeling Language: **Queuing Network Models**
- Queueing theory is the theory behind what happens when you have a lot of jobs, scarce resources, and so long queue and delays.
- Queueing theory applies whenever queues come up
- It is built on an area of mathematics called *stochastic modelling and analysis*



Queueing Networks





- Queues in computer systems:
  - A queue of web requests in the webserver that wait to be served
  - A router in a network that serves from an incoming queue of packets waiting to be routed
  - Databases have lock queues, where transactions wait to acquire the lock and execute