Dependable Distributed Systems Master of Science in Engineering in Computer Science

AA 2023/2024

LECTURE 22A - OVERVIEW ON CAPACITY PLANNING

Schema

Part of the material inside is taken from slides of profs Menascé, Almeida, Cardellini, and Ciciani

Recap dependability

Dependability is the ability of a system to deliver a service that can justifiably be trusted,
it is the ability to avoid service failures that are more frequent and more severe than is acceptable

A **service failure** is an event that occurs when the delivered service deviates from correct service

A **correct service** is delivered when the service implements its functional specifications in terms of

- **functionality**
- **performance**

Why do performance affect correct service? **SLA**

It defines how a service should operate within agreed-upon boundaries

It is a formal agreement (contract) between a provider and a consumer of a service

SLAs determine what a user of an application can expect

It is composed by one or more **Service Level Objectives (SLO)** defined over **Service Level Indicator (SLI)** generally in terms of response time, throughput, system availability, reliability, etc.

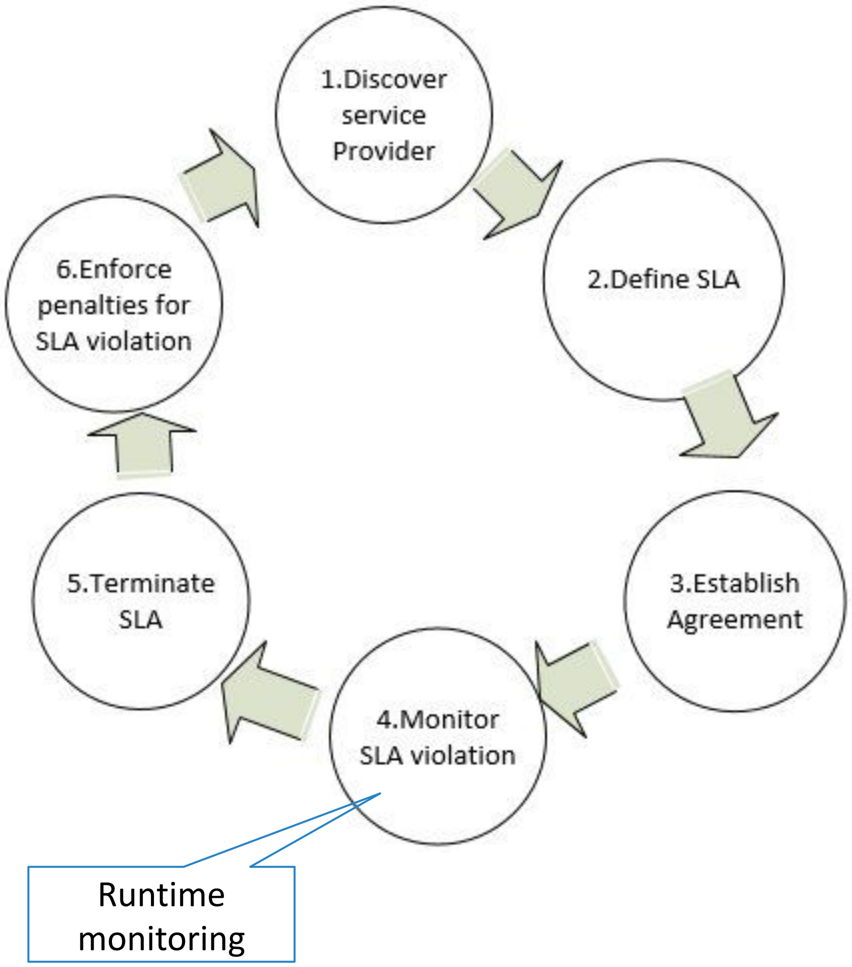
- focus on metrics that users can understand
- set easy-to-measure goals
- strong relationship between IT costs and SLA

SLA composed by SERVICE LEVEL OBJECTIVES (SLO)

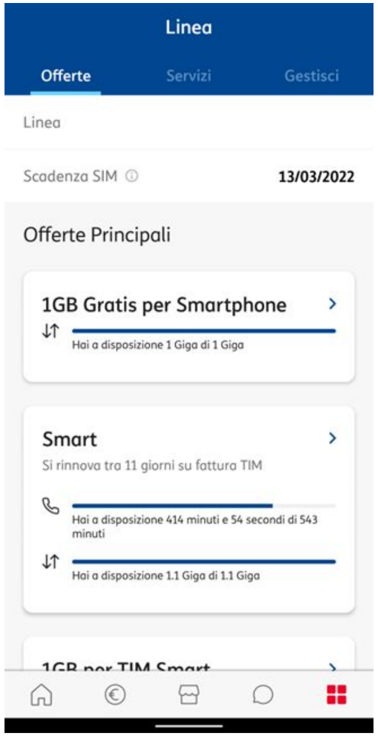
defined over SERVICE LEVEL INDICATOR (SLI)

- response time
- througput
- ...

SLA life cycle



Why do performance affect correct service?
Users expectation


Linea

Linea

Scadenza SIM 13/03/2022

Offerte Principali

1GB Gratis per Smartphone

Hai a disposizione 1 Giga di 1 Giga

Smart

Si rinnova tra 11 giorni su fattura TIM

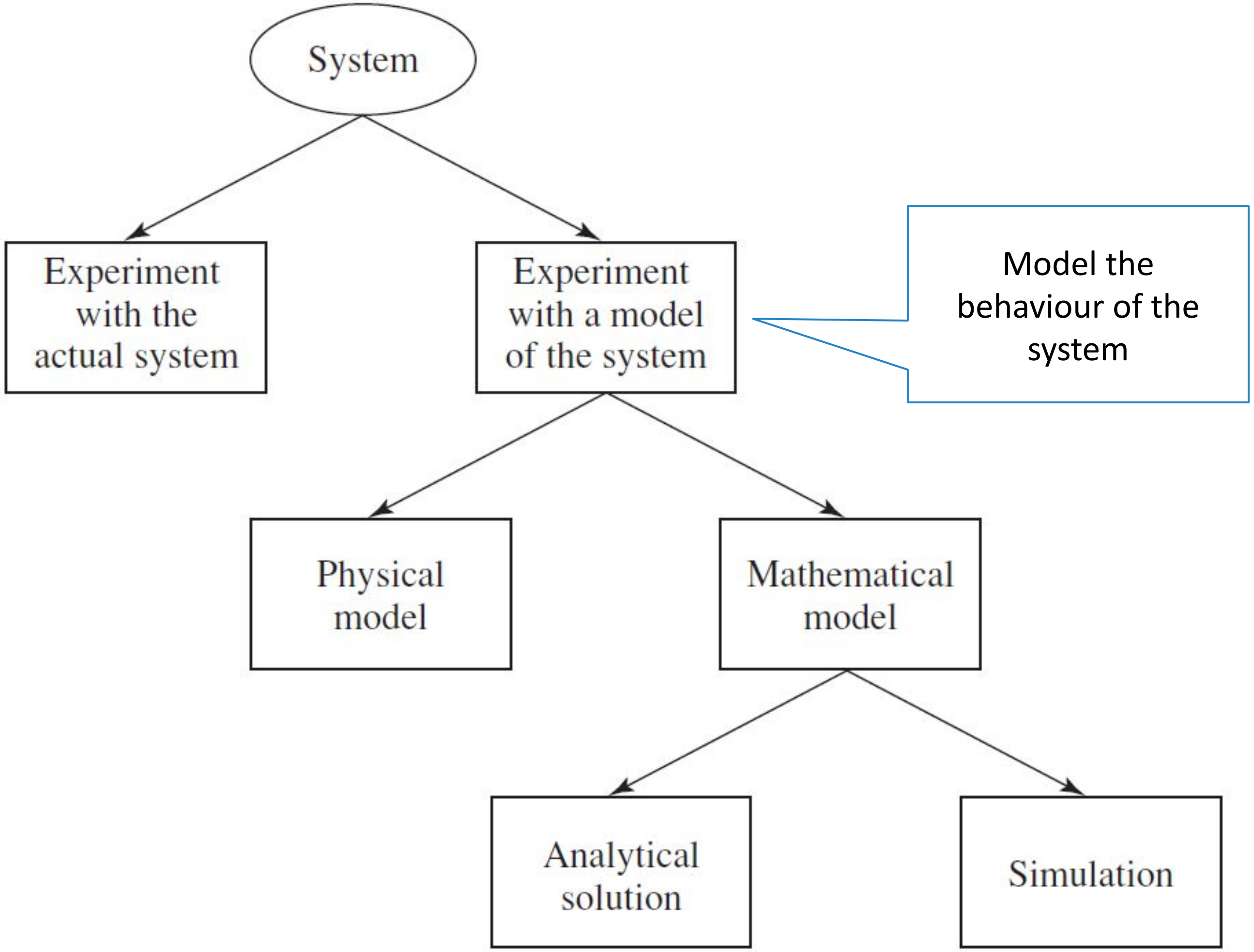
Hai a disposizione 414 minuti e 54 secondi di 543 minuti

Hai a disposizione 1.1 Giga di 1.1 Giga

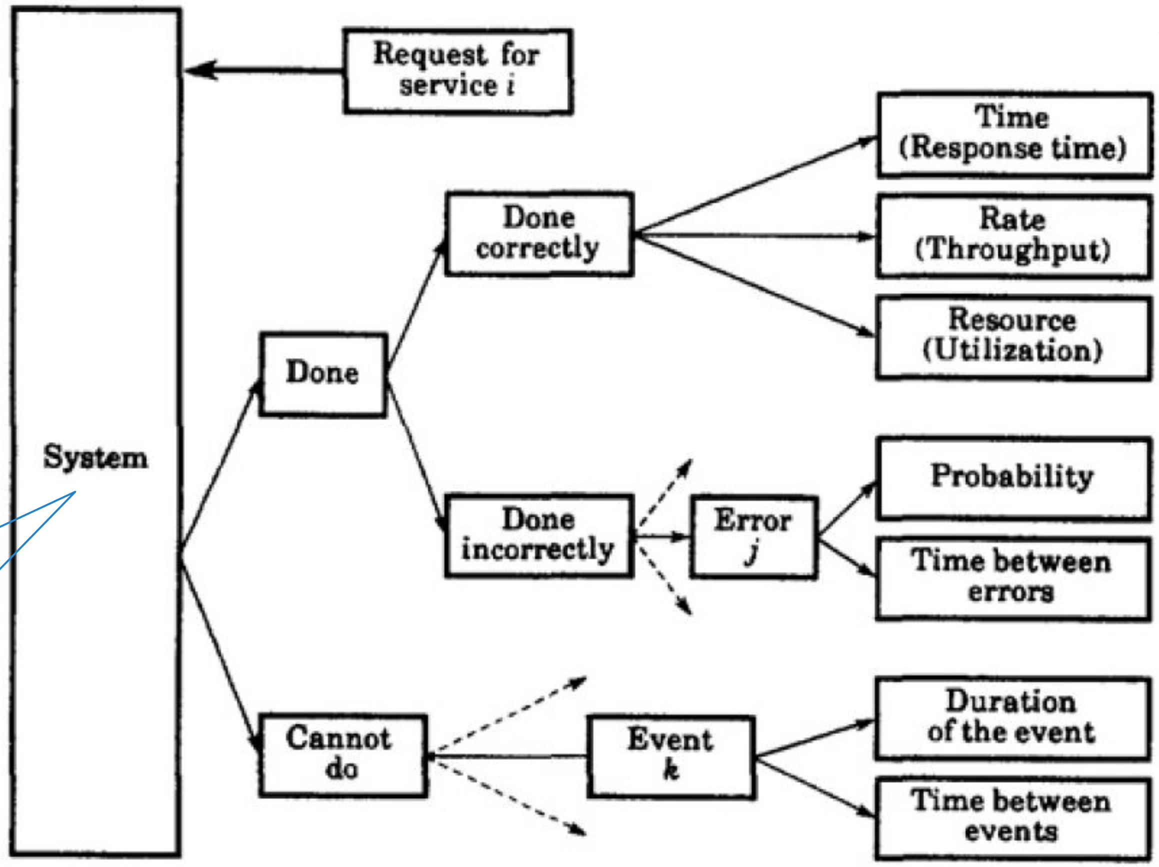
1GB per TIM Smart

Users expectation **varies depending on what type of application** they are using and even what portion of the application they are interacting with

Ways to evaluate a system


System
Experiment with the actual system
Experiment with a model of the system
Model the behaviour of the system
Physical model
Mathematical model
Analytical solution
Simulation

Very Basics for Dependability Evaluation



Note that it is assumed that the system is available for measurements!

Collect measures

How to achieve SLOs? Capacity Planning

↓

OBJECTIVES

IT capacity planning consists in **estimating** the storage, hardware, software and connection infrastructure **resources required over some future period** of time to **correctly support service provisioning**

Alternatively

IT capacity planning is the process of **predicting when the service levels will be violated as a function of the workload evolution**, as well as the **determination of the most cost-effective** way of delaying system saturation.

_> **Adequate capacity**

Properly handle peaks and average behaviour

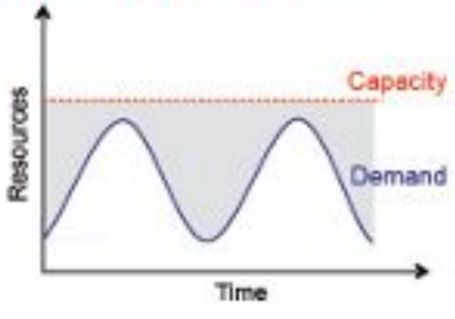
Why Do Capacity Planning?

Naive solution, **over-provisioning** : resource provisioning by taking into account **peak loads**

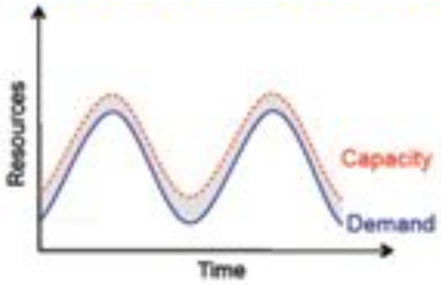
picco

_> **under-utilization**

_> higher costs than required



over-provisioning

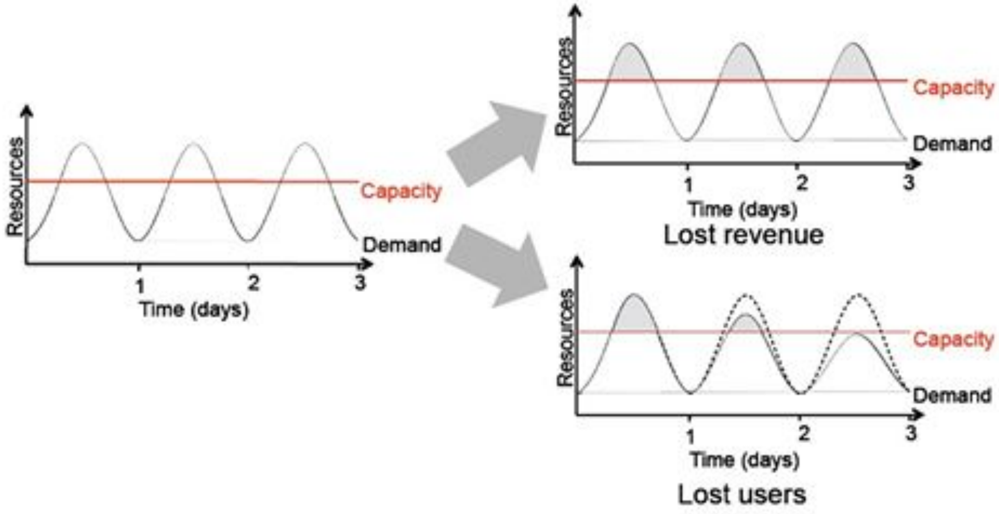


optimum

Why Do Capacity Planning?

On the other hand, in case of **under-provisioning**

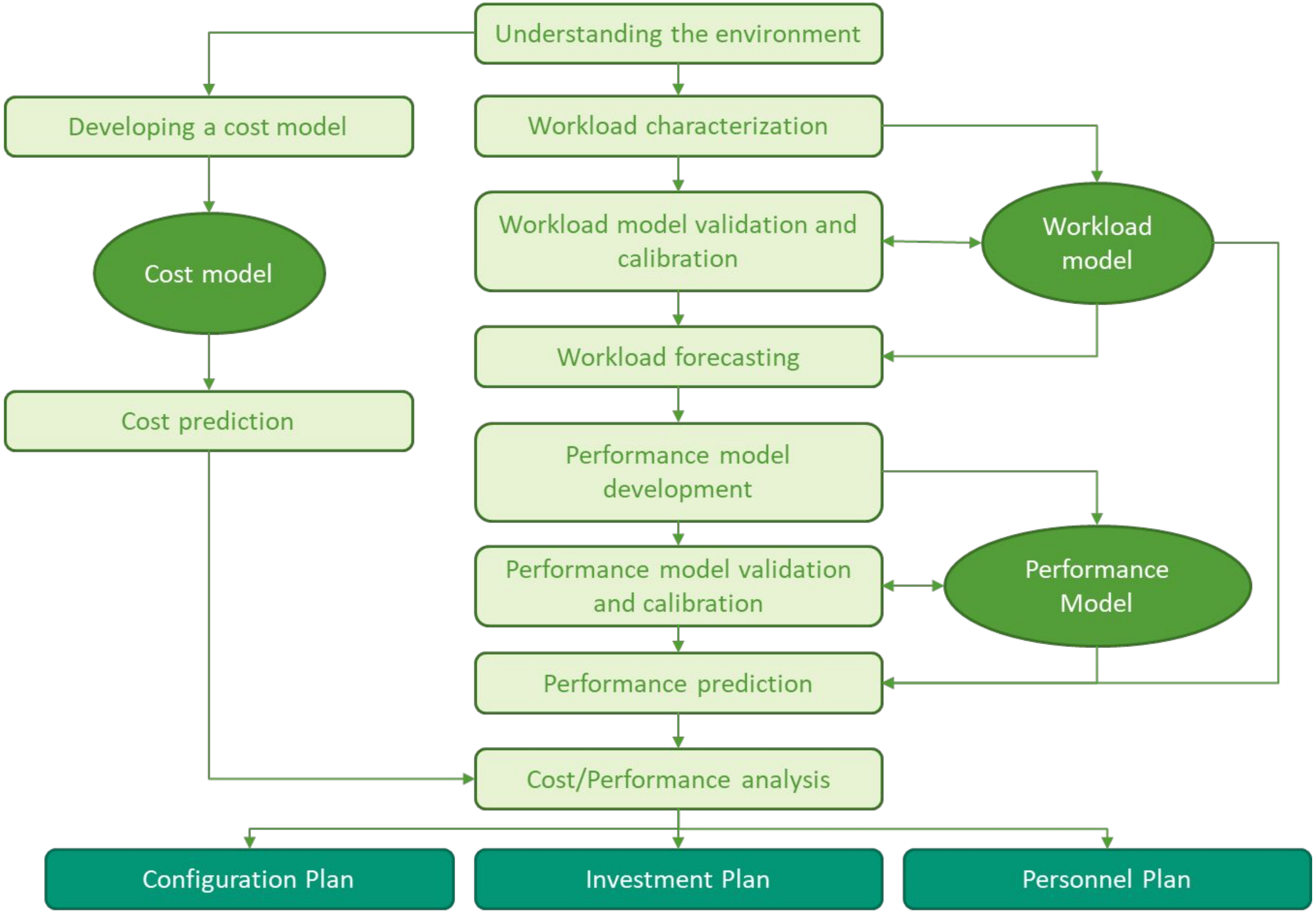
- **Overload**
- **Poor performance**
- **Losses** (users and SLA violations)



Why Do Capacity Planning?

- Avoid financial losses
- Ensure customer satisfaction
- Preserve company's external image
- It cheaper compared to deploying and testing
- Often you are in the design phase of a system
- **Capacity planning problem cannot be solved instantaneously**

A methodology for Capacity Planning



Understanding the environment

Developing a cost model

Workload characterization

Cost model

Workload model validation and calibration

Workload model

Workload forecasting

Cost prediction

Performance model development

Performance model validation and calibration

Performance Model

Performance prediction

Cost/Performance analysis

Configuration Plan

Investment Plan

Personnel Plan

Understanding the environment

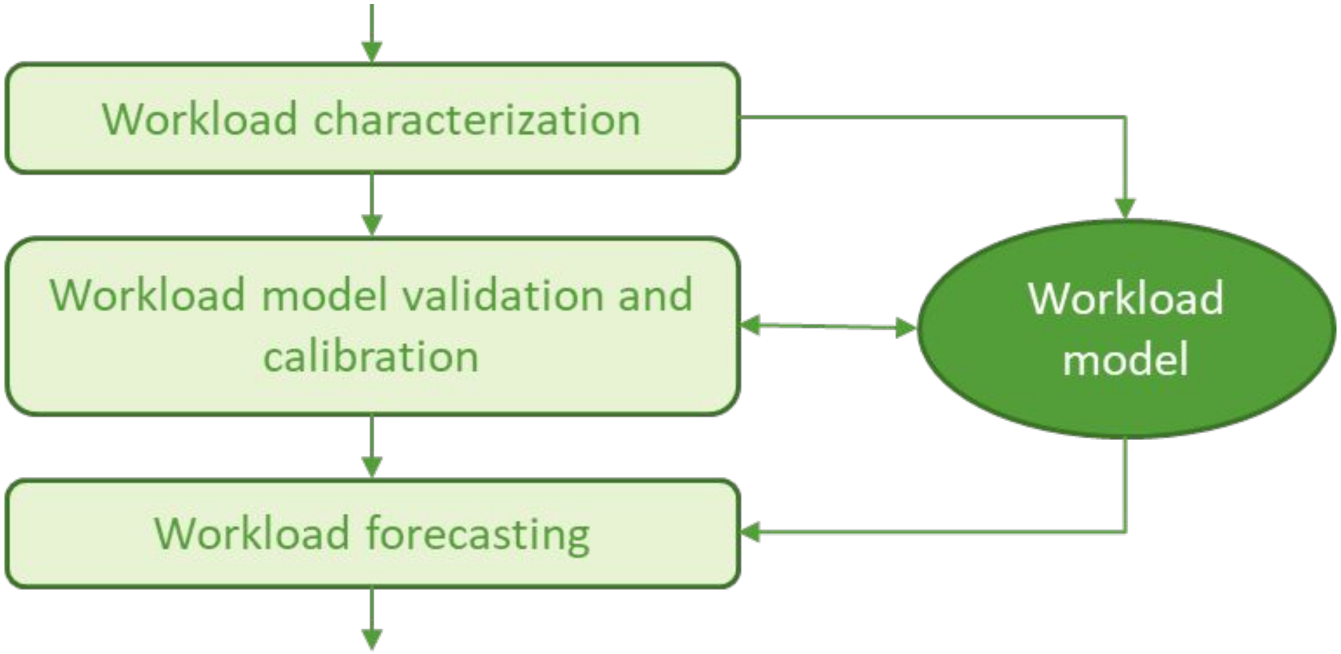
The goal is to learn what kind of

- **hardware (clients and servers)**
- **software** (OS, middleware, applications)
- **network** connectivity and protocols
- **SLA**
- … (whatever may have an impact on the considered performance metrics)

are present in the environment

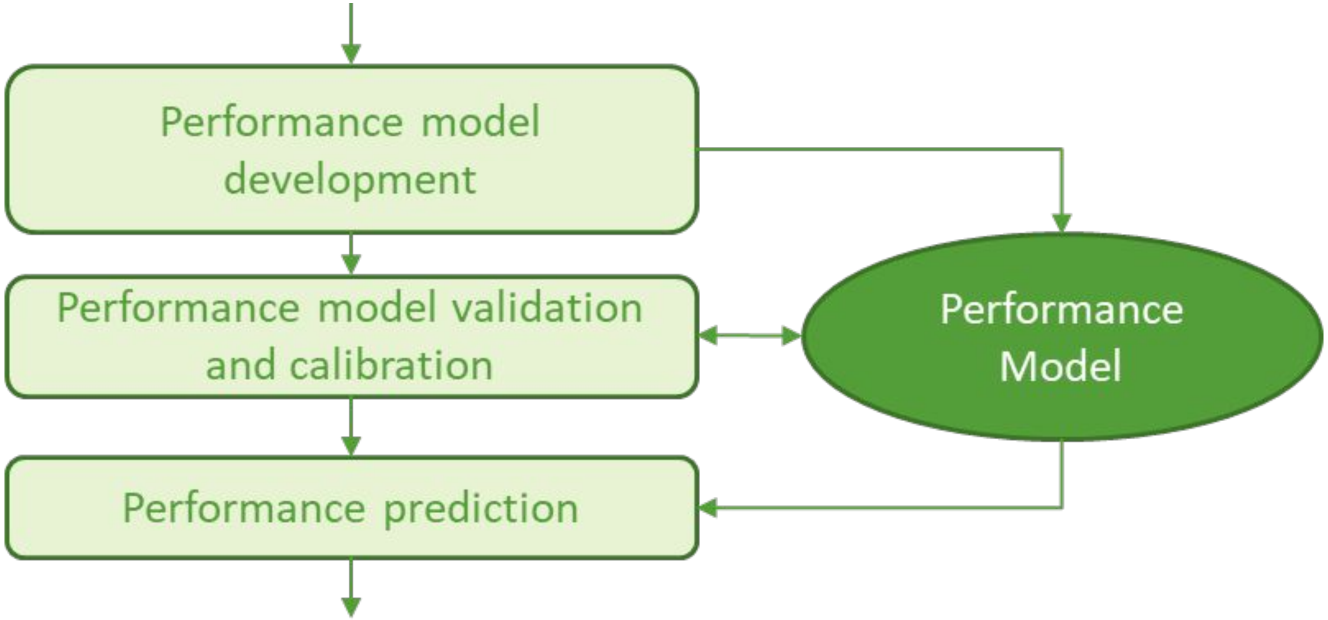
Workload model

The **workload** of a system is the **set of all inputs that the system receives from its environment** in a given period of time.


Workload characterization
Workload model validation and calibration
Workload forecasting
Workload model

Performance model

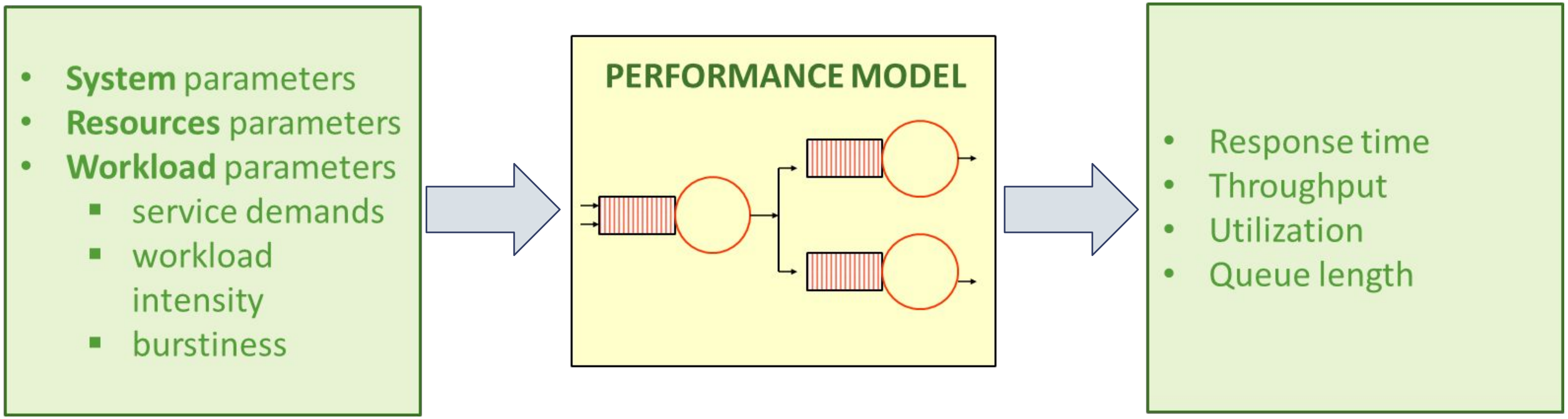
Predicts performance metrics based on **system description** and **workload parameters**


Performance model development
Performance model validation and calibration
Performance prediction
Performance Model

Estimates performance measures of a computer system for a given set of parameters

Outputs: response times, throughputs, system resources utilizations, queue lengths, etc.

Estimating performance measures



- **System** parameters
- **Resources** parameters
- **Workload** parameters
  - service demands
  - workload intensity
  - burstiness

PERFORMANCE MODEL

- Response time
- Throughput
- Utilization
- Queue length

Parameters affecting performance metrics

System **parameters examples**:

- load-balancing disciplines
- network protocols
- max. num of connections supported
- …

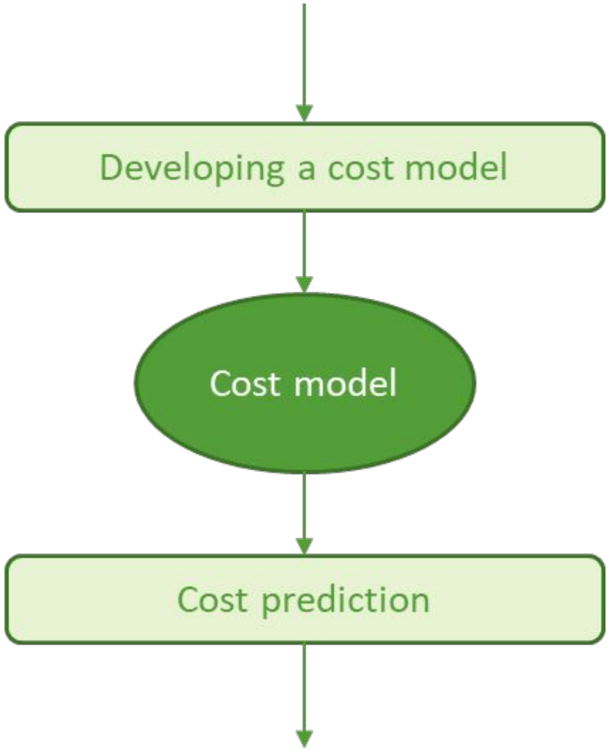
Resource **parameters examples**:

- disk latency, transfer rate
- network bandwidth;
- CPU speed
- …

Workload **parameters examples**

- WL intensity parameters:
  - num. of requests
  - num. of clients running an application
  - Burstiness
  - …
- WL service demand parameters:
  - CPU time per request
  - Disk usage per request
  - …

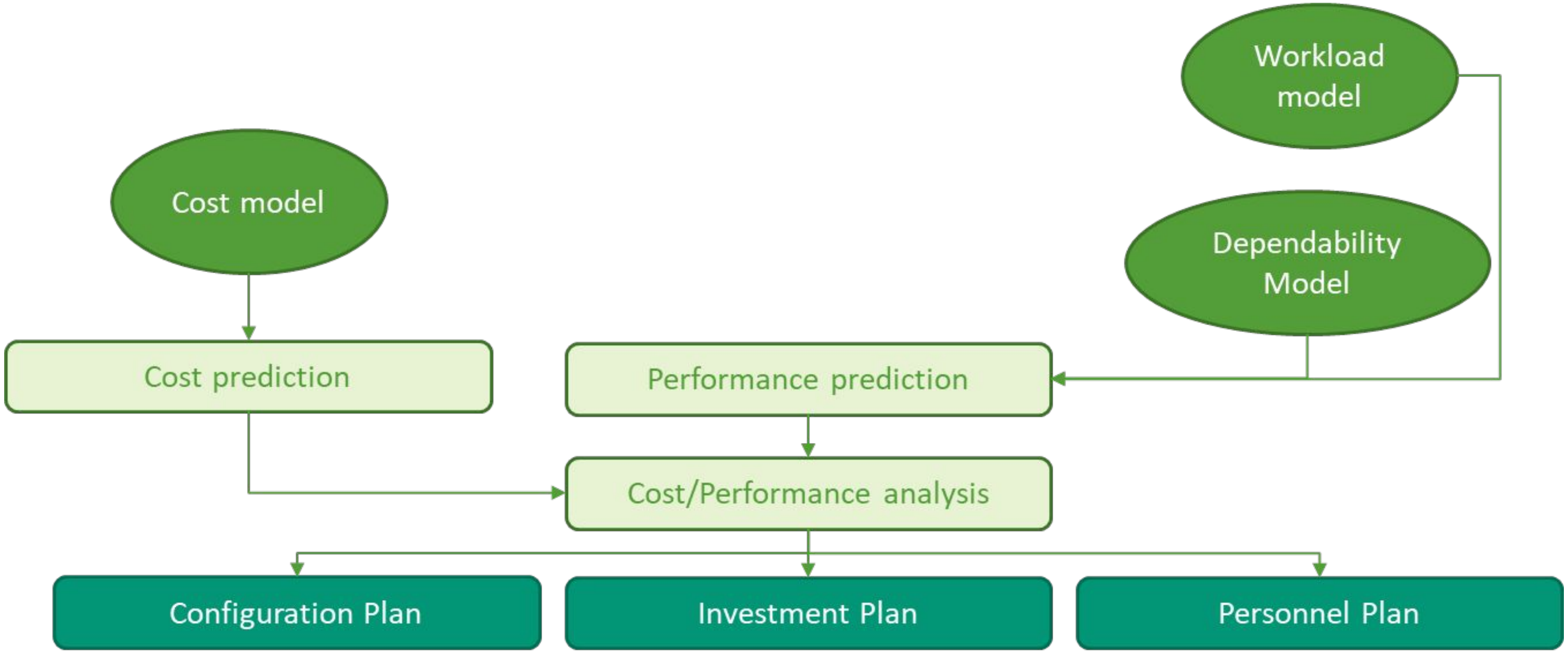
Cost model



Categories:

- Hardware cost: machines, disks, routers, etc.
- Software cost: operating systems, middleware, etc.
- Telecommunication cost
- …

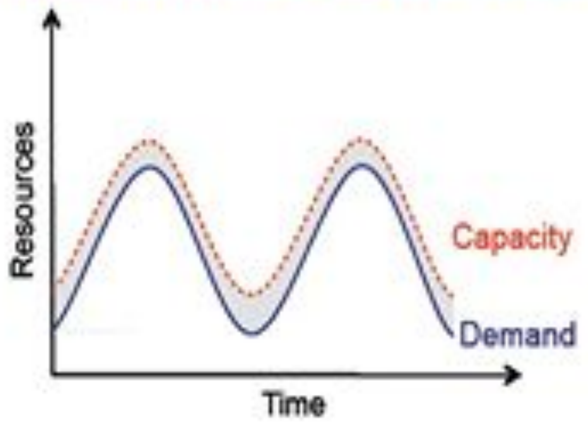
Cost/performance analysis



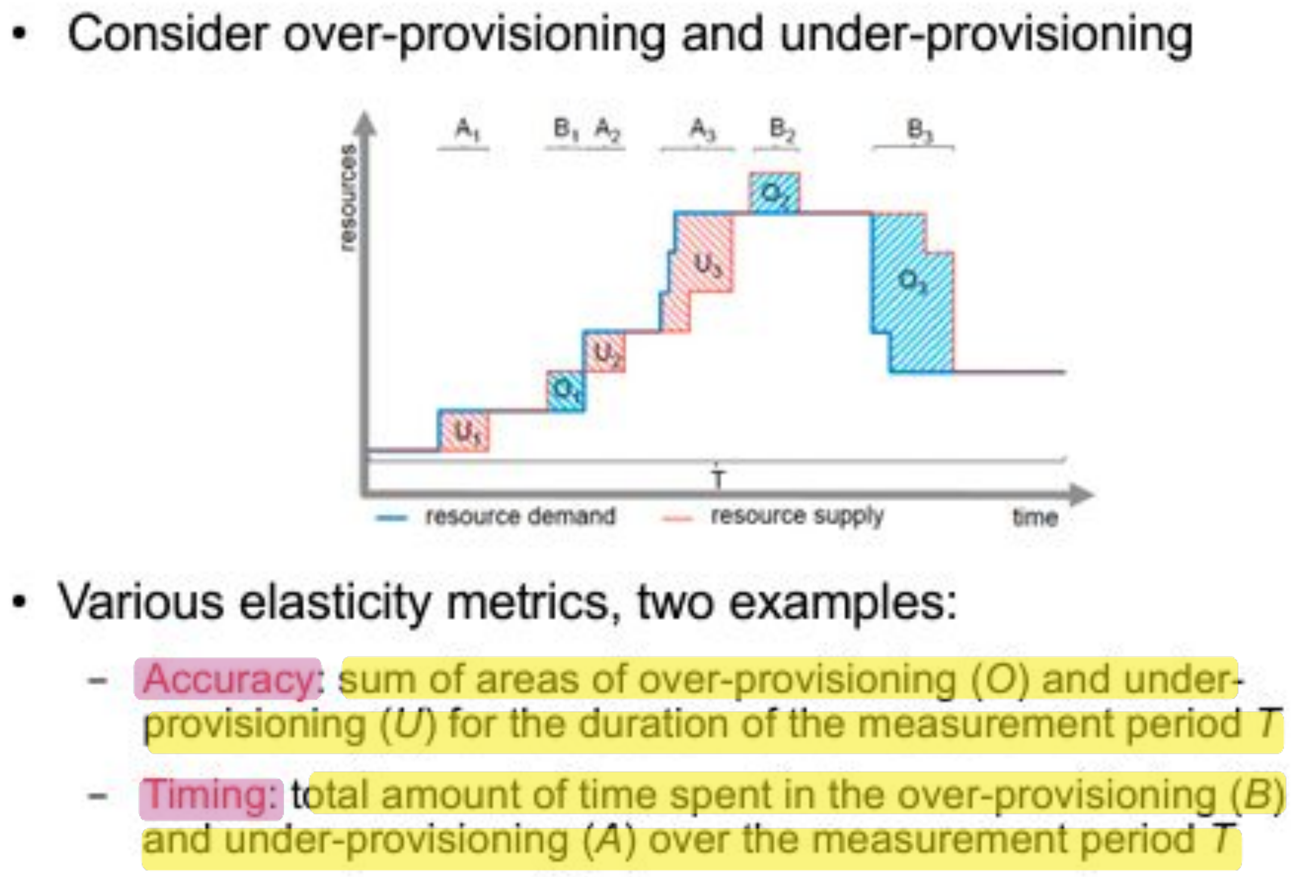
- → Assess possible **scenarios**
- → For each scenario, **predicts performance metrics and costs**
- → Comparing scenarios, **get** configuration, investment and personnel **plans**
- → Assess **payback**: ROI (return of investment), company’s image, etc.

Elasticity

Elasticity is the **degree** to which a system is **able to adapt to workload changes** by provisioning and de-provisioning resources **in an autonomic manner**, such that at each point in time the available resources match the current demand as closely as possible.



Elasticity Metrics



- Consider over-provisioning and under-provisioning

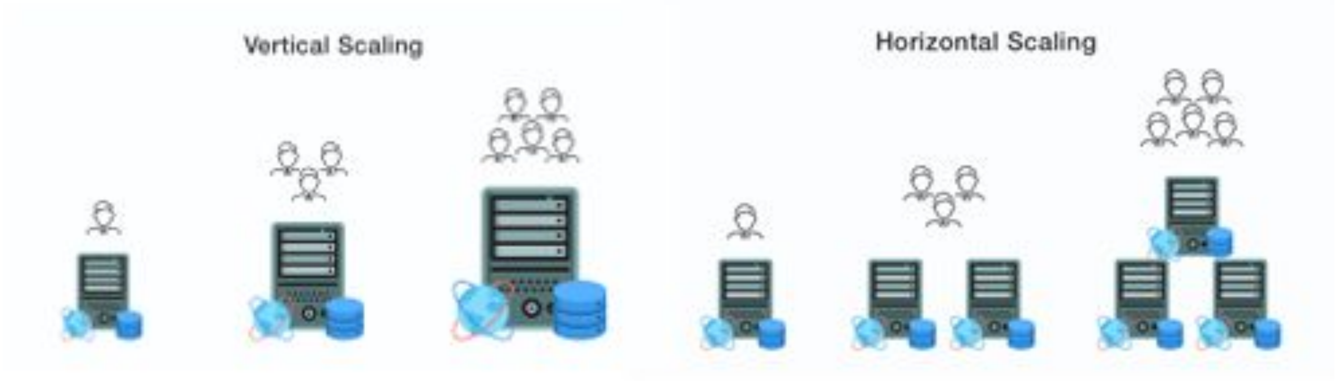
A$_1$
B$_1$
A$_2$
A$_3$
B$_2$
B$_3$
resources
O$_2$
U$_3$
O$_3$
U$_2$
O$_1$
U$_1$
T
resource demand
resource supply
time

- Various elasticity metrics, two examples:
  - Accuracy: sum of areas of over-provisioning (*O*) and under-provisioning (*U*) for the duration of the measurement period *T*
  - Timing: total amount of time spent in the over-provisioning (*B*) and under-provisioning (*A*) over the measurement period *T*

Scalability

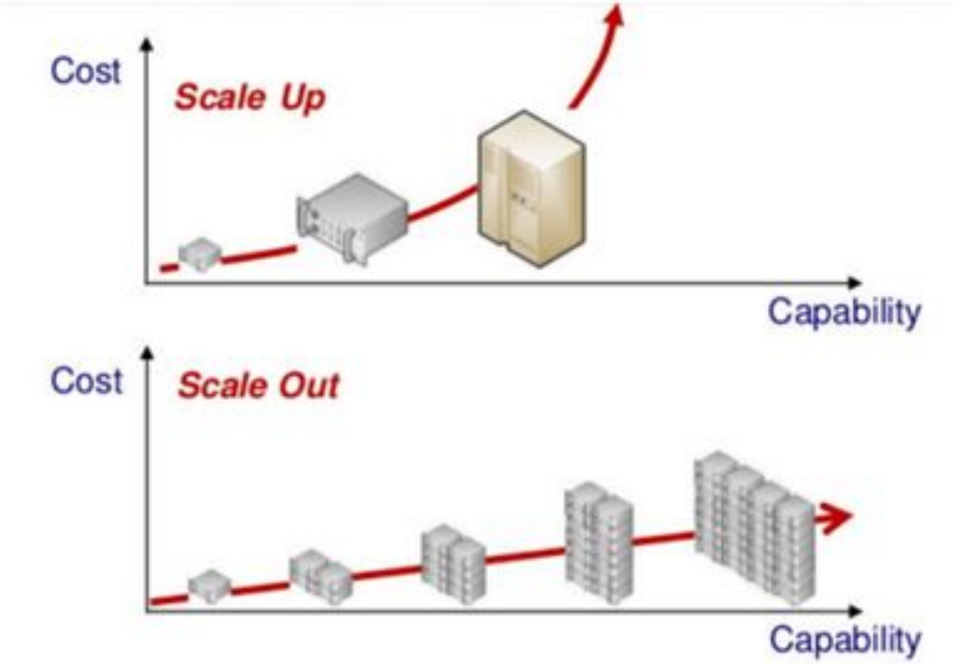
Scalability is the property of a system to handle a growing amount of work

- Two directions for size scalability
  - **Vertical (scale-up)**: more powerful resources
  - **Horizontal (scale-out)**: more resources with same capacity



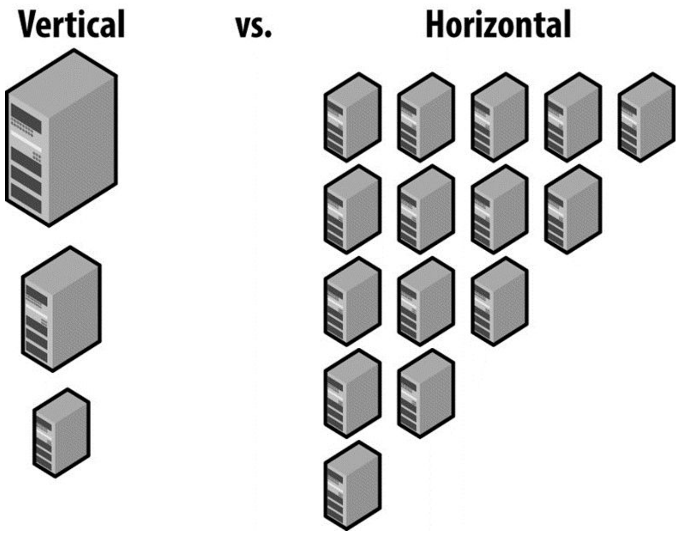
Scalability

Scalability is the property of a system to handle a growing amount of work



Scalability: Vertical VS Horizontal

- Less complex
- Upgrade limitations
- Single point of failure



- Increased complexity
- No limit to the number of processes
- Increased resilience and fault tolerance
- Horizontal scale ⇏ Increase in performance metrics

Reference

- D. A. Menascé, V. A. F. Almeida: *Capacity Planning for Web Services: metrics, models and methods*. Chapter 5 (Available in the library inside Dipartimento di Ingegneria informatica, automatica e gestionale Antonio Ruberti)

Dependability → delivery a trusted service

↓

avoid service failures → deviation from correct service

↙ functionality ↘ performance

SLA composed by SERVICE LEVEL OBJECTIVES (SLO)

defined over SERVICE LEVEL INDICATOR (SLI) → througput

↓ response time

How to achieve SLOs?

↓

Capacity planning → why do CP?

CASE 1 over provisioning

CASE 2 under provisioning

GOAL: what kind of hardware, software, SLA, network, ... are presented in environment.

Metodology CP → ③ COST model → HW, SW, telecommunication cost

↓

① WORKLOAD model

is the set of all inputs that the system receives from its environment during a given period of time

② PERFORMANCE model

↓

parameters:

system: net protocols, ...

workload: WL intensity (# of req, ...)
WL service demand

resource: CPU speed, ...

↓ when we have all of this

cost / perf. analysis

assess various scenario

In DS → elasticity

metrics : accuracy
timing

↓ scalability

size of scal : vertical SCALE-UP
horizontal SCALE-OUT