Computer Networks Performance Evaluation

Chapter 4
Performance Engineering Methodology

Performance by Design:
Computer Capacity Planning by Example

Daniel A. Menascé, Virgilio A.F. Almeida, Lawrence W. Dowdy
Prentice Hall, 2004

Outline

Central Question

- Here is a central question:
  - how can one plan,
  - design,
  - develop,
  - deploy and
  - operate IT services that
- meet the ever increasing demands for
  - performance,
  - availability,
  - reliability,
  - security, and
  - cost?

Specific Questions

Specific Questions:

1. Is a given IT system properly designed and sized for a given load condition?
2. Can the insurance claims management system meet the performance requirement of sub-second response time?
3. Is the infrastructure of a government agency scalable and can it cope with the new online security policies required for financial transactions?
4. Can the security mechanisms be implemented without sacrificing user-perceived performance?
5. Is the reservations system for cruise lines able to respond to the anticipated peak of consumer inquiries that occurs after a TV advertisement campaign?

System Analysis

- By breaking down the complexity of an IT system, one can analyze
  - the functionality of each component,
  - evaluate service requirements, and design and
  - operate systems that will meet user's expectations.
- Answer to the questions requires a deep understanding of the system architecture and its infrastructure.
- This chapter presents the basic steps of a methodology for performance engineering that are needed to fulfil the major performance needs of IT services.

Outline

Performance Engineering

- It has been a common practice in many systems projects to consider performance requirements only at the final stages of the software development process.
- As a consequence, many systems exhibit performance failures that lead to delays and financial losses to companies and users [25].
- Performance engineering analyzes the expected performance characteristics of a system during the different phases of its lifecycle.

Performance Engineering Task

Performance engineering Tasks:

1) Develops practical strategies that help predict the level of performance a system can achieve.

2) Provides recommendations to realize the optimal performance level.

Activities

- Both tasks rely on the following activities that form the basis of a methodology.
  - Understand the key factors that affect a system's performance.
  - Measure the system and understand its workload.
  - Develop and validate a workload model that captures the key characteristics of the actual workload.
  - Develop and validate an analytical model that accurately predicts the performance of the system.
  - Use the models to predict and optimize the performance of the system.

Collection of Methods

- Performance engineering can be viewed as a collection of methods for the support of the development of performance-oriented systems throughout the entire lifecycle [7].
- The phases of the system lifecycle define
  - the workflow,
  - procedures,
  - actions, and
  - techniques

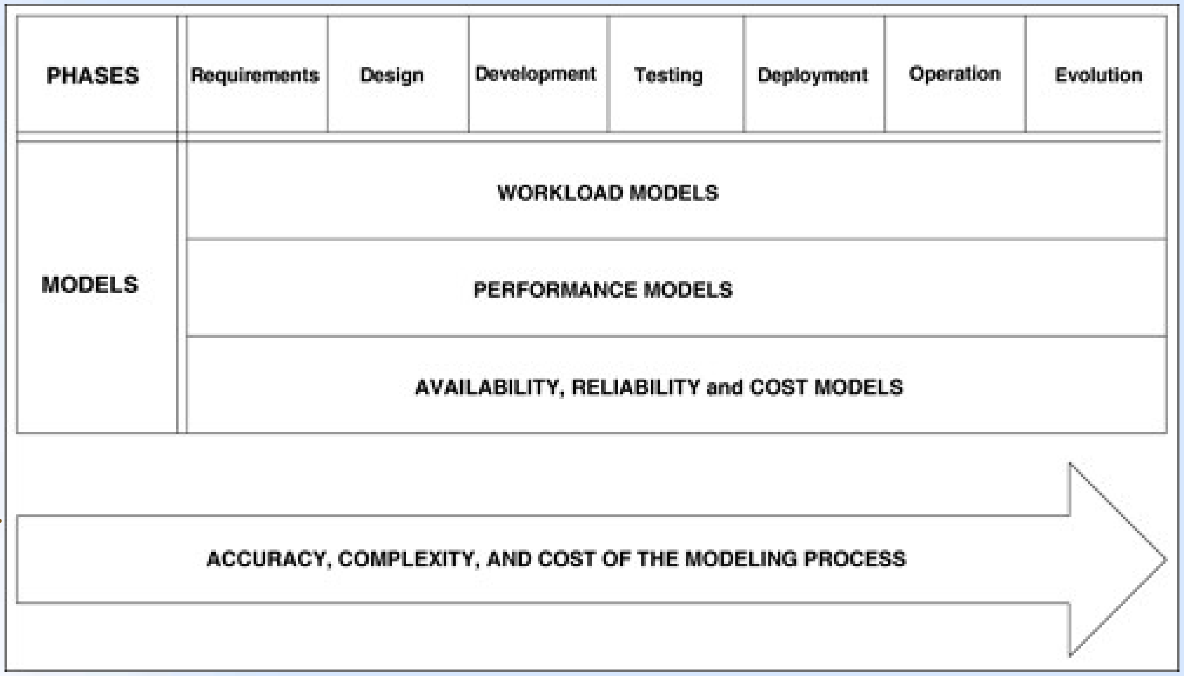
  that are used by analysts to produce and maintain IT systems.
- This methodology is based on models that are used to provide QoS assurances to IT systems.

Performance Engineering Models

- These models are:
  - workload model,
  - performance model,
  - availability model,
  - reliability model, and
  - cost model.

  At the early stages of a project, the information and data available to develop the models are approximate at best.

Figure 4.1. Modelling process. spectrum of the models for the different phases, ranked from least to most complex and costly.



PHASES	Requirements	Design	Development	Testing	Deployment	Operation	Evolution
MODELS	WORKLOAD MODELS						
	PERFORMANCE MODELS						
	AVAILABILITY, RELIABILITY and COST MODELS						

ACCURACY, COMPLEXITY, AND COST OF THE MODELING PROCESS

as complexity and cost increase, the refined models becomes more accurate

Models Evolve

- In the requirement analysis stage, it is difficult to construct a detailed workload model, specifying the demands placed by transactions on the system's resources.
- As the project evolves, more system information becomes available, reducing the error margin of the model's assumptions and increasing confidence in the estimated predictions of the future system's performance [23].
- The workload and performance models evolve side by side with the system project throughout the stages of its lifecycle.
- Various types of workload and performance models can be used during the system lifecycle.

Outline

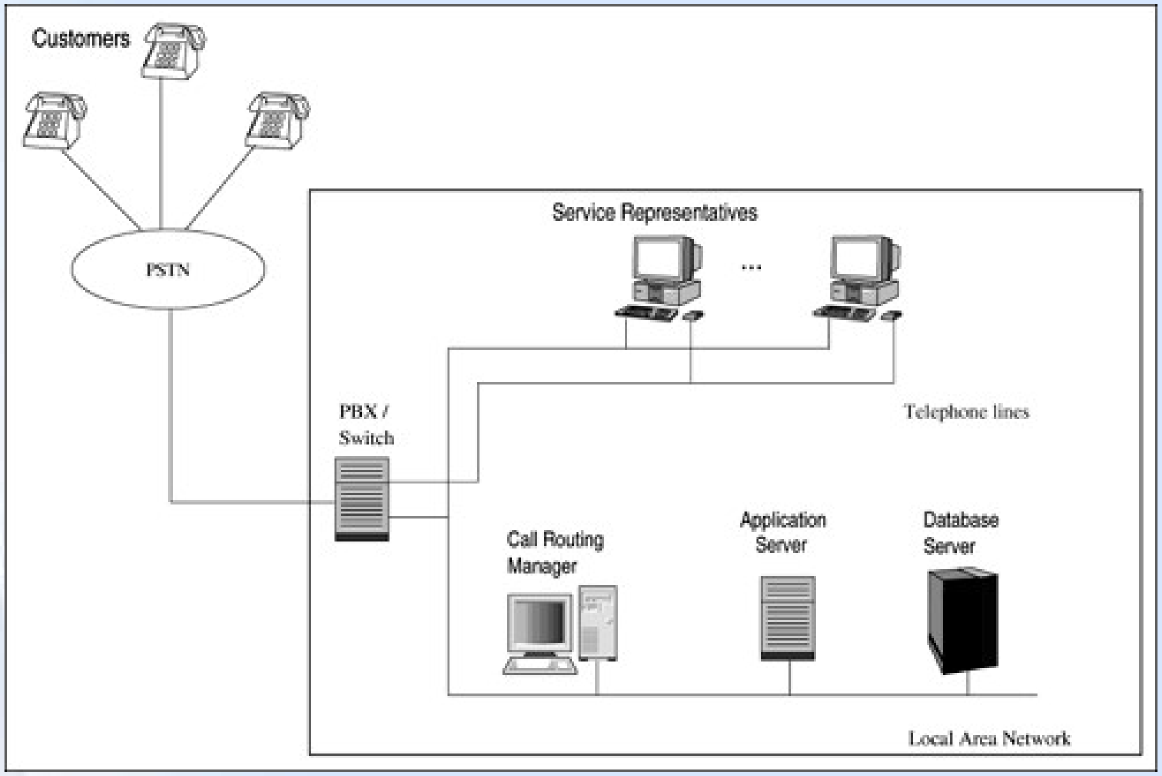
Motivating Example

- Facing sharp competition, consider an automotive parts distribution company that has decided to develop and implement a new call centre.
- The goals of the new call centre system are to:
  1) foster better relationships with customers, creating customer loyalty and ensuring quality service,
  2) improve efficiency and service performance, and
  3) identify and explore new sales opportunities.

Integrated Customer

- The information technology team has decided to create an integrated customer relationship management system.
- The proposed new architecture consists of computer telephony integration software that controls inbound calls and routes these calls to service representatives integrated with the application software.
- Figure 4.2 depicts an overview of the call centre architecture.

Figure 4.2. Architecture of the call centre.



What is a call Center

- A **call centre** (UK) or **call center** (US) is a centralized office of a company that answers incoming telephone calls from customers.
- A call center may be an office that makes outgoing telephone calls to customers (telemarketing).
- Such an office may also responds to letters, faxes, e-mails and similar written correspondence.
  - However the term **contact centre** (UK) or **contact center** (US) is often applied when such multiple functions are blended in one office.

Call Center

- The call centre will be staffed by service representatives who handle customer orders,
  - returns,
  - queries by phone,
  - fax, and
  - e-mail.
- Basically, customers inquire about:

  1) the status of an order,

  2) the location of a shipment, and

  3) problem resolution status.

Basic Function

- Each customer service representative will have immediate access to the following basic functions:
  1) historical tracking of all customer contacts,
  2) a single view of customer information,
  3) records of resolution of past problems, and
  4) help functions.
- Management realizes that customers are less tolerant of long calls.
- Thus, the system should meet sub-second response times on all functions and be operational 24 hours a day, 7 days a week.

Scenarios

- Various scenarios when designing its call centre.
- The IT team is planning to
  - design,
  - build,
  - test, and
  - deploy the new call centre applications in 12 months.

Questions

- Before rollout, management wants assurances that the performance requirements are satisfied.
- Questions that project management asks include:
  - Is the system design able to meet the sub-second response time for all functions?
  - What will be the impact of doubling the number of system representatives in the next year?

Questions.

- Can acceptable performance levels be maintained after integrating the system with the mainframe-based inventory application?
- Is the system capacity adequate to handle up to 1,000 calls in the busiest hour and yet preserve the sub-second response time goal?
- How do failures in the database server affect the 24x7 availability goal?
- What is the impact of starting to offer Web-based self-service to customers?

Initial Idea

- During the requirement analysis stage, the analysts specify the type of system architecture and the resources that will support the QoS goals for the system.
- The analysts want an initial idea about the models that will be used in the performance engineering analysis phase.

Workload Model

- Consider a first attempt for a workload model.
- Initially, one needs to define the workload to be characterized.
- There are multiple and different workloads in a system depending on the point of view from which one looks at the workload.
- The workload presented to the call centre system consists of the rate of telephone calls into the call centre.

Workload Model.

- From the IT application standpoint, the workload consists of all functions it receives from the representatives during an observation period.
- A database server may receive queries from the application server, which in turn receives function execution requests from the representatives.
- The load of the local network is usually described in terms of its traffic characteristics, including the
  - packet size distribution and
  - the inter-packet arrival time.

Workload Model..

- A second issue that must be about the workload model is the level of detail its description.
- A high-level description specifies the workload from the user's point of view.
- For instance, one could specify the load in terms of functions submitted by the users.
- On the other hand, a low-level characterization describing the user's requests in resource-oriented terms

Performance Model

- Initially, the performance model is a simple black box.
- The desired output of the model includes
  - throughput,
  - response time, and
  - availability.

Performance Model.

- Consider the call center system described in the motivating example.
- During the system design stage, the analyst needs to answer the following question:
  - what should be the system throughput to meet the sub-second response time requirement?
- At this stage, the analyst makes assumptions such as:
  1) the number of service representatives will be 200, and
  2) during the busiest hour, 80% of the representatives are working simultaneously answering calls.

Interaction Model

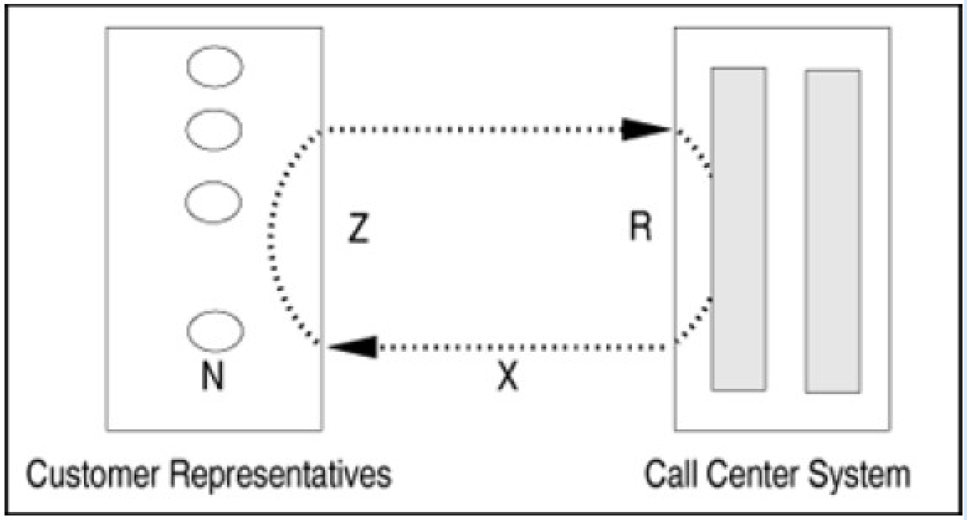
- The conversation of a representative with a customer can be represented by a simple interaction model.
- The representative
  - listens to the customer,
  - selects a function,
  - submits it to the system,
  - waits for the system to respond,
  - watches while the results appear,
  - talks to the customer,
  - terminates the call, and
  - waits for another customer call.
- The think time corresponds to the period of time between a reply to a function execution request and the following submission.

Example 4.2

- At this stage of the project, the analyst's view of the call centre application is a black box, modelled by Fig. 4.3
- The analyst estimates an average think time of 30 sec, denoted by Z.
- The number of active representatives in the system, denoted by N, is equal to 200 x 0.8.
- The system throughput is denoted by X_0.
- If the response time, R, is not to exceed 1 second, it follows from the Interactive Response Time Law (see Chapter 3) that
- Then, $R = N/X_0 - Z \leq 1 \text{sec.}$

$$X_0 \geq \frac{N}{(Z+R)} = \frac{N}{(Z+1)} = 5.16 \, functions \; / \sec.$$

Figure 4.3. Model of the call center system.



Database Throughput

- During the system development phase, the various components of the system are implemented.
- Suppose that during this phase, the analysts determine that each function submitted by a representative demands 2.2 queries from the database server.
- Before developing the database component of the system, they need to know the required capacity of the database server that is needed to meet the performance objectives.
- The Forced Flow Law, discussed in Chapter 3, applied to the database server establishes a relationship between the database server throughput and the system throughput.

Database Throughput.

- That is, $X_{DB} = V_{DB} \times X_0$ (4.3.1)
- where X_{DB} is the database server throughput, X_0 denotes the total system throughput, and V_{DB} is the average number of visits per function to the database server.
- The minimum database server throughput, in queries/sec, is obtained in Example 4.2 as $X_0 \geq 5.16$ functions/sec.
- Considering that the minimum system throughput is 5.16 functions/sec and each function accesses the database server 2.2 times,
- the minimum database server throughput, measured in queries per second is

$$X \geq 2.2 \times 5.16 = 11.32\, transactio\, ns / \sec$$

Measurement

- Measuring the performance of the call center applications is a key issue in the process of guaranteeing the quality of service objectives.
- It is also an essential step for performance engineering, because it collects data for performance analysis and modelling.
- These measurement data are used to calculate the model input parameters, which describe
  - the system configuration,
  - the software environment, and
  - the workload of the call center system.

Time Decomposition

- Let the average response time be denoted by R_{call}.
- The response time can be decomposed into three main components:

$$R_{call} = R_{APPL} + R_{LAN} + R_{DB}$$

- where $R_{APPL}, R_{LAN}, and\ R_{DB}$ represent the response time at
  - the application server,
  - the Local Area Network, and
  - the database server,

respectively.

Performance Verification

- The description of the motivating example provides a performance requirement that $R_{call} < 1$.
- In the operation stage, the system is constantly monitored to verify that performance objectives are being met.
- During the peak hour, the system shows signs of saturation, with average response times longer than one second.
- Management is considering upgrading the database server and wants to know what component of the server is most responsible for slowing down its response time.

Questions

- The relevant questions include:
  - What is the average response time per query?
  - What is the average throughput of the DB server?
  - What are the utilizations of the CPU and disks?
- For the sake of simplicity, consider that the database server services only one type of transaction: query.
  - The DB server has one CPU and two disks.
- During the peak period, the DB server receives requests from the application server at a rate of 57,600 queries per hour.

Parameters

- Based on measurements collected during the operation of the call center application, the analyst obtains the following data:
- on average, each query needs 50 msec of CPU and performs four I/Os on disk 1 and two I/Os on disk 2.
- Each I/O takes an average of 8 msec.
- The throughput, X_0, is equal to the average arrival rate λ, given by:
  - 57,600/3,600 = 16 queries/sec.
- The service demand at the CPU, D_{CPU}, is 0.050 sec.

Service Demands

- The service demand at the CPU, D_{CPU}, is 0.050 sec.
- The service demands at disks 1 and 2 are: From the Service Demand Law (see Chapter 3),

$$D_{disk1} = V_1 \times S_{disk} = 4 \times 0.008 = 0.032 \sec.$$

$$D_{disk2} = V_2 \times S_{disk} = 2 \times 0.008 = 0.016 \sec.$$

- Therefore, the utilization of the CPU and the disks are given by: $U_i = D_i \times X_0$

$$U_{cpu} = D_{cpu} \times X_0 = 0.05 \times 16 = 80\%$$

$$U_{disk1} = D_{disk1} \times X_0 = 0.032 \times 16 = 51.2\%$$

respectively.

$$U_{disk2} = D_{disk2} \times X_0 = 0.016 \times 16 = 25.6\%$$

Residence Time

- Using the residence time equation for open queuing networks of Chapter 13, the residence times at the CPU and disks are:

$$R'_{cpu} = \frac{D_{cpu}}{1 - U_{cpu}} = \frac{0.050}{1 - 0.8} = 0.250 \text{ sec}$$

$$R'_{disk\ 1} = \frac{D_{disk\ 1}}{1 - U_{disk\ 1}} = \frac{0.032}{1 - 0.512} = 0.066 \text{ sec}$$

$$R'_{disk\ 2} = \frac{D_{disk\ 2}}{1 - U_{disk\ 2}} = \frac{0.016}{1 - 0.256} = 0.022 \text{ sec}$$

- The total response time is the sum of all residence times.

Result

- Thus,
  The CPU is the bottleneck since it has the largest service demand.

$$R_{DB} = R'_{CPU} + R'_{disk1} + R'_{disk2}$$
$$= 0.250 + 0.066 + 0.022 = 0.338 \sec$$

- As the load increases, the CPU utilization will reach 100% before any other device and will limit the throughput of the database server.

Call Center Evolution

- Now consider the evolution stage of the call center system.
- The company is considering to develop Web applications to allow customers access to the information they need without assistance from a customer representative.
- Web self-services reduce transaction costs and enhance the customer experience.
- To let users directly access the call center database, management wants to make sure that security requirements will be met.
- Security depends on the context in which security issues are raised [4].

New Requirement

- In the business context, the security emphasis is on the protection of assets.
- Management wants to minimize the risk of unauthorized modification of the information in the call center database.
- Thus, software engineers involved need to design a new type of transaction that satisfies the security requirements.
- The security requirement has implications on two areas of the system [4]:
  - interface and
  - internal architecture.

New Interface, Audit

- The new interface should include authentication services that perform the mapping between the user's identity and the person using the system.
- New auditing features should be included to aid administrators discover the history of unauthorized access in the event of a security breach.
- In terms of the internal architecture, the designers plan to modify the access control mechanisms to the database information.
- Before embarking on the developing and implementation of the Web application, management wants to assess the likely impact of the new application on the call center service.

Parameters

- From the software model [25], the analyst is able to estimate the service demands for the new class of transactions, submitted via the Web.
- A new performance model for the database server is developed.
- The database server model is subject to two types of transactions:
  - local queries and
  - Web queries.
  - the arrival rate of local queries transactions is 16 tps and
  - the arrival rate of Web queries transactions is 1 tps.
- The service demands for the
  - CPU,
  - disks 1, and
  - disk 2, are given in Table 4.1.

Table 4.1. Service Demands and Arrival Rates of Database Transactions for Ex. 4.5

	Local Queries	Web Queries
Arrival Rate (tps)	16	1
Service Demands (sec)		
CPU	0.050	0.150
Disk 1	0.032	0.200
Disk 2	0.016	0.100

Open Queuing Network

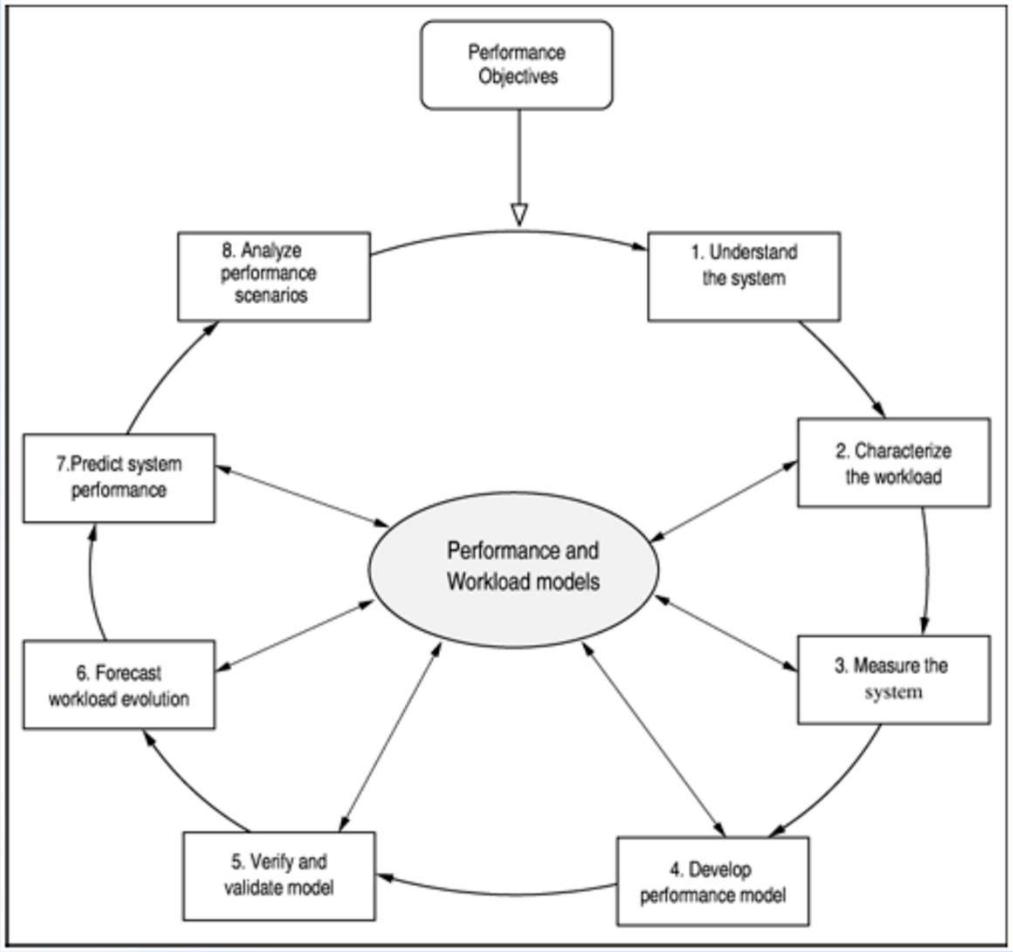
- Using the equations for the multiple-class case of open queuing networks (see Chapter 13) and the OpenQN.XLS MS Excel workbook the model can be solved and the results show management the impact of Web transactions on call center applications.
- The response times for local queries and Web queries at the database server are 1.14 sec and 3.85 sec, respectively.
- These numbers provide management with a concrete sense of the impact of implementing Web applications on the current infrastructure of the call center system.

Outline

A Model-based Methodology

- This section describes the basic steps used to design and analyze computer systems with performance in mind.
- The methodology builds on workload and performance models and can be used throughout the phases of the system lifecycle.
- Performance engineering provides a series of steps to be followed in a systematic way [2].
- Figure 4.4 gives an overview of the main steps of the quantitative approach to analyze the performance of a system.

Figure 4.4. A model-based performance engineering methodology.


Performance Objectives
1. Understand the system
2. Characterize the workload
3. Measure the system
4. Develop performance model
5. Verify and validate model
6. Forecast workload evolution
7.Predict system performance
8. Analyze performance scenarios
Performance and Workload models

Performance Objective

- The starting point of the methodology is to specify system performance objectives.
- These objectives should be quantified as part of the system requirements.
  - Performance objectives are used to establish service level goals.
  - Service level goals are defined,
  - Business metrics are established, and
  - Performance goals of the system are documented.
- Once the system and its quantitative objectives have been determined, one is able to go through the quantitative analysis cycle.

1-Understand the System,

- Understand the system.
  - Obtain an in-depth understanding of the system architecture and
  - Conduct an architecture-level review with emphasis on performance.
    - This means answering questions such as:
      - What are the system requirements of the business model?
      - What type of software (i.e., operating system, transaction monitor, DBMS, application software) is going to be used in the system?

1-Understand the System.

- This step yields a systematic description of the system architecture, its components, and goals.
- It is an opportunity to review the performance issues of the proposed architecture.

2-Characterized the Workload

- Characterize the workload.
  - In this step, the basic components that compose the workload are identified.
    - The choice of components depends both on the nature of the system and the purpose of the characterization.
  - The product of this step is a statement such as "The workload under study consists of e-business transactions, e-mail messages, and data-mining requests."

2-Characterized the Workload

- The performance of a system with many clients, servers, and networks depends heavily on the characteristics of its load.
- Thus, it is vital in any performance engineering effort to understand clearly and characterize accurately the workload [8, 13].
- The workload of a system can be defined as the set of all inputs that the system receives from its environment during any given period of time.
- For instance, if the system under study is a database server, then its workload consists of all transactions (e.g., query, update) processed by the server during an observation interval.

3-Measure the System,

- Measure the system and obtain workload parameters.
  - The third step involves measuring the performance of the system and obtaining values for the parameters of the workload model.
  - Measurement is a key step to all tasks in performance engineering.
  - It allows one to understand the parameters of the system and to establish a link between a system and its model.
  - Performance measurements are collected from different reference points, carefully chosen to observe and monitor the environment under study.

3-Measure the System,

- For example, consider that a database server is observed during 10 minutes and 100,000 transactions are completed.
- The workload of the database during that 10-minute period is the set of 100,000 transactions.
- The workload characteristics are represented by a set of information e.g.,
  - arrival and completion time,
  - CPU time, and
  - number of I/O operations) for each of the 100,000 database transactions.

4-Develp Performance Model

- Develop performance models.
  - In the fourth step, quantitative techniques and analytical (or simulation or prototype) models are used to develop performance models of systems.
  - Performance models are used to understand the behaviour of complex systems.
  - Models are used to predict performance when any aspect of the workload or the system architecture is changed.
  - Simple models based on operational analysis discussed in Chapter 3 are accessible to software engineering practitioners.
  - They offer insight into how software architectural decisions impact performance [11].

5-Verify and Validate,

- Verify and validate the models.
  - The fifth step aims at verifying the model specifications and validating the model's results.
  - This step applies to both, performance and workload models.
  - A performance model is said to be validated if the performance metrics (e.g., response time, resource utilizations, throughputs) calculated by the model match the measurements of the actual system within a certain acceptable margin of error.

5-Verify and Validate.

- As a rule of thumb, resource
  - utilizations within 10%, system
  - throughput within 10%, and
  - response time within 20% are considered acceptable [16].
- A model is said to be verified if its results are an accurate reflection of the system performance [25].
- Details of performance model calibration techniques are available [16].
- In summary, this step answers questions such as:
  - Is the right model for the system being considered?
  - Does the model capture the behaviour of the critical components of the system?

6-Forecast Workload Evolution,

- Forecast workload evolution.
- Most systems suffer modifications and evolutions throughout their lifetime.
- As a system's demands change, so do workloads.
  - Demands grow or shrink, depending on many factors, such as
    - the functionalities offered to users,
    - number of users,
    - hardware upgrades, or
    - changes to software components.

6-Forecast Workload Evolution.

- The sixth step forecasts the expected workload for the system.
- Techniques and strategies for forecasting [12] workload changes should provide answers to questions such as:
  - What will be the average size of e-mails by the end of next year?
  - What will be the number of simultaneous users for the online banking system six months from now?

7-Predict System Performance,

- Predict system performance.
  - Performance guidance is needed at each stage of the system lifecycle, since every architectural decision can potentially create barriers in achieving the system performance goals.
  - Thus, performance prediction is key to performance engineering work, because one needs to be able to determine how a system will react when changes in load levels and user behaviour occur or when new software components are integrated into the system.
  - This determination requires predictive models.

7-Predict System Performance.

- Experimentation, is not usually viable because fixing performance defects may require structural changes that are expensive.
- In the seventh step, performance models are used to predict the performance of a system under many different scenarios.

8-Analyze Performance Scenarios,

- Analyze performance scenarios.
  - Validated performance and workload models are used to predict the performance of a system under several different scenarios, such as
  - upgraded servers,
  - faster networks,
  - changes in the system workload,
  - changes to the user behaviour, and
  - changes to the software system.

8-Analyze Performance Scenarios.

- To help find the most cost-effective system architecture, different scenarios are analyzed in this step.
- Basically, each scenario consists of a future system feature and/or a workload forecast.
- Because every forecast item carries a certain degree of uncertainty, several possible future scenarios are considered.
- Different possible system architectures are analyzed.
- A selection of alternatives is generated so that system engineers may choose the most appropriate option, in terms of cost/benefit.

Central Components

- Two models are central components of the methodology:
  - the workload model and
  - the system performance model.
- Workload models are studied in detail in this chapter.
- Performance models were introduced in Chapter 3.
- Techniques for constructing models of different types of systems are developed in Chapters 10 through 15.

Outline

Workload Model

- A workload model is a representation that mimics the real workload under study.
- It can be a set of programs written and implemented with the goal of artificially testing a system in a controlled environment.
- A workload model may also serve as input data for an analytical model of a system.
- It is not practical to have a model composed of thousands of basic components to mimic the real workload.
- Workload models should be compact and representative of the actual workload [14].

Workload Categories

- Workload models can be classified into two main categories:
  - Natural models
  - *Artificial models*
- Natural models are constructed
  - either using basic components of the real workload as building blocks or
  - using execution traces of the real workload.

Natural Model-Real Workload

- A natural benchmark consists of programs extracted from the real workload of a system.
- These programs are selected so that the benchmark represents the overall system load in given periods of time.

Benchmark

- Originally: a mark on a workbench used to compare the lengths of pieces so as to determine whether one was longer or shorter than desired.
- For computers: a "benchmark" is a test, or set of tests, designed to compare the performance of one computer system against the performance of others.
- Note: a benchmark is not necessarily a capacity planning tool.
  - That is, benchmarks may not be useful in attempting to guess the correct size of a system required for a particular use.

Natural Models-Trace

- Another natural model often used in performance studies is a workload trace.
- It consists of a chronological sequence of data representing specific events that occurred in the system during a measurement session.
- For example, in the case of a Web server, the log access contains one entry per HTTP request processed by the server.
- Among other information, each log entry specifies the name of the host making the request, the timestamp, and the name of the file requested.
- This type of log characterizes the real workload during a given period of time.

Trace Drawbacks

- Although traces exhibit reproducibility and representativeness features, they do have drawbacks.
- Usually, traces consist of huge amounts of data, which increases the complexity of using them in modelling exercises.
- It is usually difficult to modify a trace to represent different workload scenarios.
- Moreover, traces are suitable only for
  - simulation or
  - prototype models.

Artificial Models

- Artificial models do not make use of any basic component of the real workload.
- Instead, these models are constructed out of special-purpose programs and descriptive parameters.
- Artificial models are partitioned into two classes:
  - executable and
  - non-executable models.
- Executable artificial models consist of a suite of programs especially written to experiment with particular aspects of a computer system.

Artificial Model-Executable,

- The class of executable models include workloads such as
  - instruction mixes,
    - Instruction mixes are hardware demonstration programs intended to test the speed of a computer on simple computational and I/O operations.
  - kernels,
    - Program kernels are pieces of code selected from computationally intense parts of a real program.
    - In general, kernels concentrate on measuring the performance of processors without considering the I/O system.

Artificial Model-Executable.

- synthetic programs,
  - Synthetic programs are specially devised codes that place demands on different resources of a computing system.
  - Unlike benchmarks, synthetic programs do not resemble the real workload.
  - Benchmarks, synthetic programs, and other forms of executable models are not adequate inputs for performance models.
- artificial benchmarks, and
- drivers.

Artificial Model-Non Executable,

- When the performance of a system is analyzed through the use of analytic or simulation models, non executable representations for the workload are required.
- Because the approach to performance engineering relies on the use of analytic models for performance prediction, this book focuses on workload representations suitable for these kinds of models.

Artificial Model-Non Executable.

- Non-executable workload models are described by a set of average parameter values that reproduce the same resource usage as the real workload.
- Each parameter denotes an aspect of the execution behaviour of the basic component on the system under study.
- The basic inputs to analytical models are parameters that describe the service centers
  - (i.e., hardware and software resources) and
  - the customers (e.g., transactions and requests).

Parameter of Non Executable Model

- Typical parameters are:
  - Component (e.g., transaction and request) inter-arrival times,
  - Service demands,
  - Component sizes, and
  - Execution mixes (classes of components and their corresponding levels of multiprogramming).
- Each type of system may be characterized by a different set of parameters.
- As an example, consider a parametric characterization of a workload of a distributed system file server [5].

Example: Parameter of a File Server,

- Several factors have a direct influence on the performance of a file server:
  - the system load,
  - the device capability, and
  - the locality of file references.
- From these factors, the following parameters are defined:
  - Frequency distribution of the requests: describes the participation of each request (e.g., read, write, create, rename) on the total workload.

Example: Parameter of a File Server.

- Request inter-arrival time distribution: indicates the time between successive requests. It also indicates the intensity of the system load.
- File referencing behaviour: describes the percentage of accesses made to each file in the disk subsystem.
- Size of reads and writes: indicates the I/O load. This parameter has a strong influence on the time needed to service a request.
- The above parameters specify the workload model and are capable of driving synthetic programs that accurately represent real workloads.

Example: Parameter of Storage Device,

- The workload model for a storage device (i.e., queues, caches, controllers, disks) is specified by three classes of parameters:
  - access time attributes,
  - access type (i.e., fraction of reads and writes), and
  - spatial locality attributes [1].
    - temporal locality refers to the property that an item frequently accessed in the past is likely to be accessed in the future.
    - spatial locality refers to the property that items neighbouring an item frequently accessed in the past are likely to be accessed in the future.

Example: Parameter of Storage Device,

- Access time attributes capture the time access pattern, which includes the
  - arrival process (i.e., deterministic, Poisson, bursty),
  - arrival rate,
  - burst rate,
  - burst count, and
  - burst fraction

Example: Parameter of Storage Device.

- Spatial locality attributes specify the relationship between consecutive requests, such as sequential and random accesses.
- Due to "seek" delays, workloads with a larger fraction of sequential accesses typically exhibit better performance than those with random accesses.
- Thus, for a detailed model of an I/O device, it is important to include the spatial locality attributes of the workload.

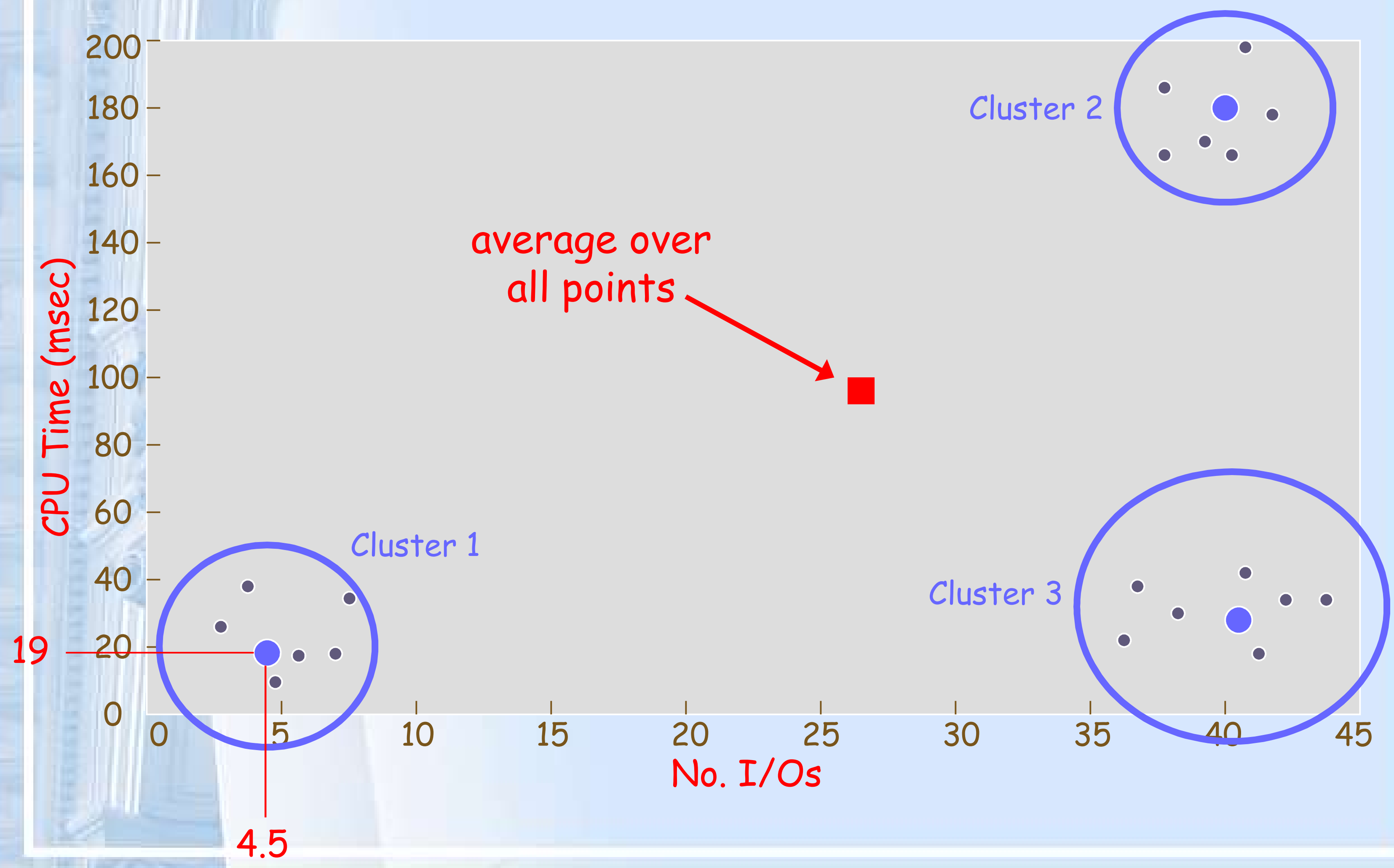
Clusters

- Consider a workload that consists of transactions that exhibit a large variability in terms of their CPU and I/O demands.
- Averaging the demands of all transactions would likely produce a workload model that is not representative of most of the transactions.
- Therefore, the transactions in the workload have to be grouped, or clustered, so that the variability within each group is relatively small when compared to the variability in the entire data set. These clusters correspond to different classes of a multiclass performance model.

Example

- Example: Suppose 20,000 transactions executed in peak period of 1 hour, CPU time and number of I/Os are measured for each transaction.
  - Fig. 5.5 shows a graph depicting points of type (number of I/Os, CPU time) for all transactions.
  - The figure shows 3 natural groups (clusters) with centroids at
    - (4.5, 19),
    - (38, 171), and
    - (39, 22).
  - The 20,000 transactions is given by the 3 centroids.
  - Transactions of calss1 perform, an the average, 4.5 I/Os and spend 19 msec of CPU time during their execution.
  - Overall average is (28, 68), representing the workload by this points leads to less accuracy if the model.

Figure 5.5


Cluster 2
average over
all points
Cluster 1
Cluster 3
19
4.5
CPU Time (msec)
No. I/Os
200
180
160
140
120
100
80
60
40
20
0
0
5
10
15
20
25
30
35
40
45