

Computer Engineering

Performance Evaluation Computer Systems and Networks

Multi-Programmed Server

*STUDENTS:*

*Fernando De Nitto*

*Nicola Ferrante*

*Simone Pampaloni*

*Academic Year 2019/2020*

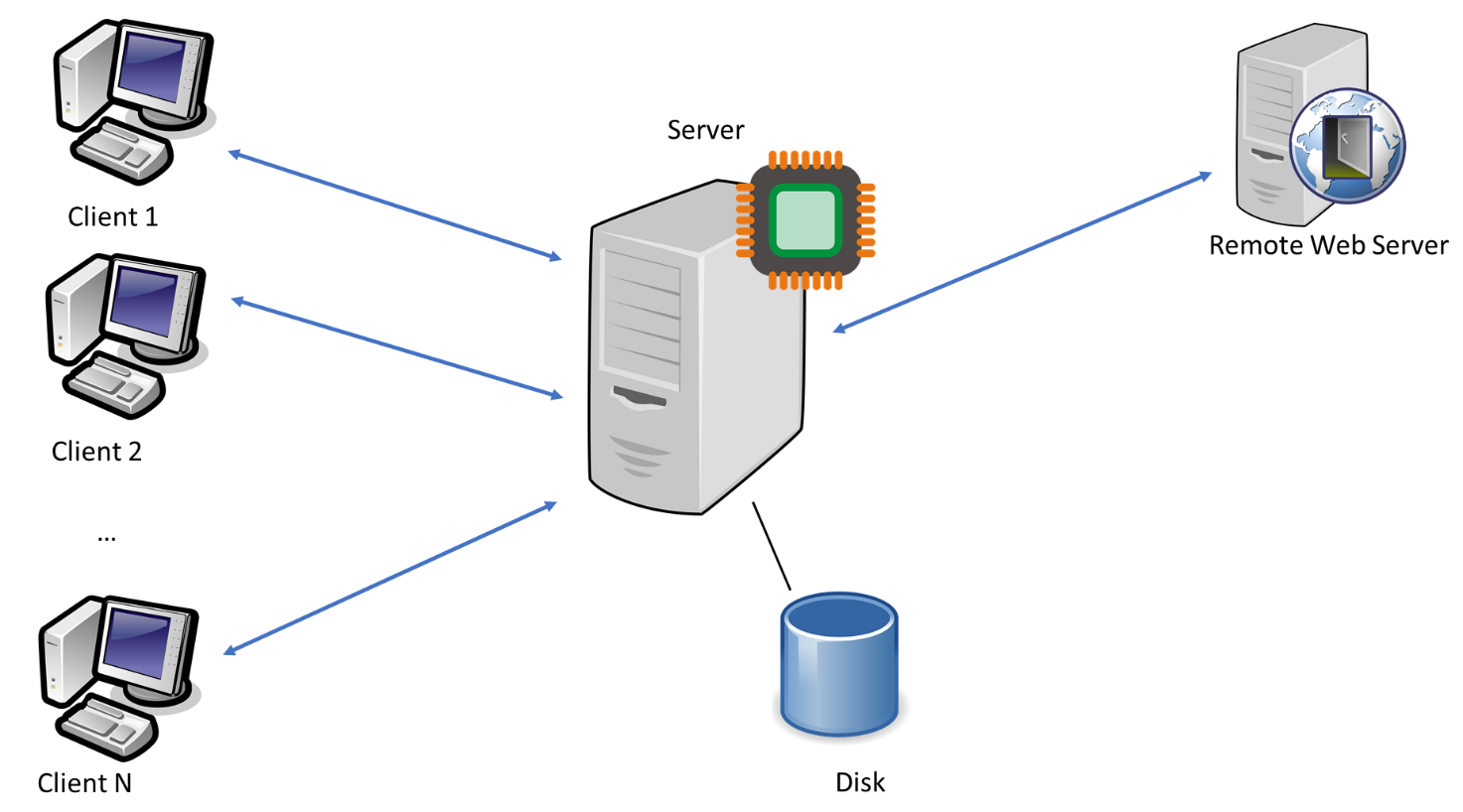
Overview

A multi-programmed server provides service to different concurrent clients. The server has access to a disk and can communicate with a remote web server. Clients send requests to the server and each request is processed one at a time by the server following a FIFO order. After an initial pre-processing phase, each request can be handled in the following ways:

* The request has finished processing and a response is sent to the client.
* The request requires a disk access and then a new processing is required.
* The request is sent to a remote web server and after its response a new processing is required.

Even disk and the remote web server handle one request at a time in a FIFO order.

A client that receives a reply immediately issues another request.



The aim of this study is to evaluate the performance of the system described above with particular emphasis on throughput. This study must be done considering various scenarios by varying the level of multiprogramming and making sure that the service demands at the

three components (server processor, disk, and remote web server) have a considerably different mean.

Performance Indexes

Since the objective of this study is to evaluate the performance of the system described above, the following performance indexes have been defined:

* Throughput: the number of served requests divided by the operating time.
* Mean response time: the mean time it takes between the departure of a request and the arrival of the response to the client
* Utilization: How long each system component (server processor, disk, and remote web server) processes requests during operating time.

Model

the following model represents an abstraction of the system to be studied:

Immagine che contiene testo

Descrizione generata automaticamente

Transactions may involve processing in the main server, access to the disk, and remote queries to a remote web server. As mentioned above, a new transaction always requires some processing time as a first step. After the processing has occurred:

* With a probability ***p1*** the transaction is terminated, and a reply is sent to the client

that originated it.

* With a probability ***p2***a disk access is required. After that, a new processing is required.
* With a probability ***p3*** = 1-p1-p2 a query to remote web server is required. After that, a new processing is required.

The server processor, the disk, and the remote web server handle one request at a time in a FIFO order. Processing, disk access and query to remote server service demands (with rates ***µ1***, ***µ2*** and ***µ3***) are exponential IID RVs, and they are different from one iteration to another, even for the same transaction.

As can be seen from the diagram, the system is designed as a Closed Jackson’s Network

since clients send instantly a new request whenever they receive a response, so there is always the same number of service requests in the system.