



UNIVERSITÀ DI PISA

Master's degree in Computer Engineering

592II Performance evaluation of computer systems  
and networks

**Multiprogrammed server**

Designers:

**Tommaso Califano**

**Nicola Ramacciotti**

**Gabriele Suma**

# Contents

<b>1</b>	<b>Overview</b>	<b>2</b>
1.1	Objectives . . . . .	2
1.2	Performance Indexes . . . . .	3
<b>2</b>	<b>Model</b>	<b>4</b>
2.1	General assumption and preliminary validation . . . . .	4
2.2	Factors . . . . .	5
<b>3</b>	<b>Implementation</b>	<b>6</b>
3.1	Modules . . . . .	6
3.2	Verification . . . . .	7

# Chapter 1

## Overview

### 1.1 Objectives

A multi-programmed server provides service to **N concurrent clients** that request the server to perform transactions. **Local CPU** computations, interactions with the **local disk** and remote queries to a **distant web server** are all processes that may be involved in transactions. An interaction between clients and server can be defined as follows:

1. A new transaction always requires some processing time as a first step.
2. Transactions can follow different flows based on probability.
3. A reply is sent to the client that originated the request.
4. A user that receives a reply immediately issues another request.

Utilizing the FIFO policy, each module within the system (Local CPU, local disk and the remote web server) is capable of processing a single request at any given time. Considering the previous assumptions, it becomes crucial to evaluate the system's performance with a particular emphasis on **throughput, mean response time and utilization**. With the aid of the Omnet++ simulation software, we can gain insights into the system's behavior under various conditions. Furthermore, utilizing MS Excel, we can collect the data obtained from simulations for data analysis and graph representations.

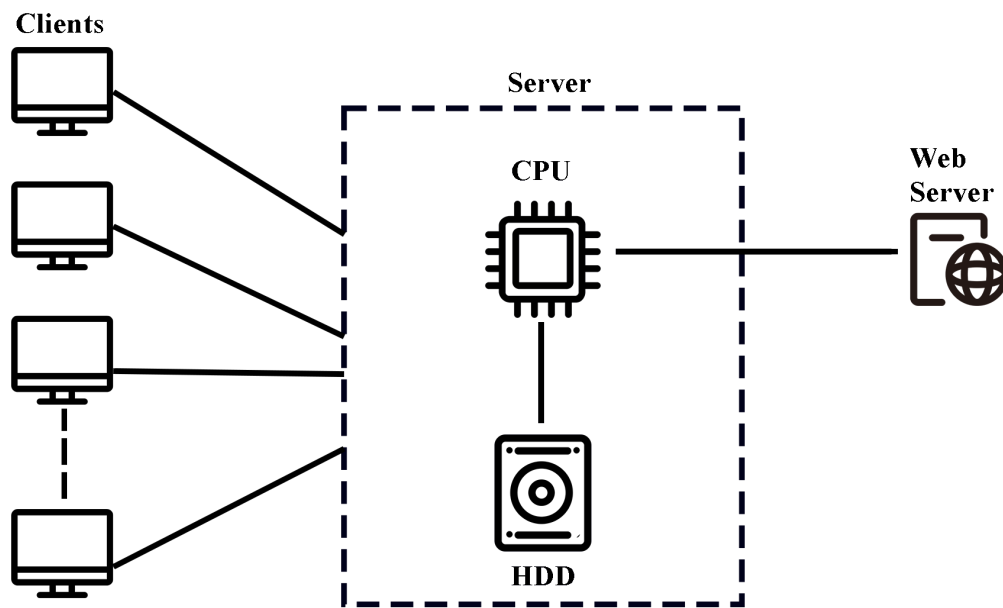


Figure 1.1: Simplified and High-level scheme

## 1.2 Performance Indexes

To evaluate the performances of the system, the following indexes were defined:

- **Throughput:** the number of completed transactions per unit of time.
- **Mean response time:** the mean duration required for a particular client request to obtain the corresponding response.
- **Utilization:** the time percentage during which each node is busy.

# Chapter 2

## Model

### 2.1 General assumption and preliminary validation

- **Clients:** the number of clients is finite and it corresponds with the number of jobs within the system. Given that, a client instantly sends a new request upon receiving a response from the server, implementing a queue or modeling clients as service centers would be unnecessary.
- **Queues:** the queue size is not dimensionally limited. Essentially, this has been an assumption in the simulation. Indeed, having  $N$  clients implies that the queue could reach a size of  $N$  jobs in the worst-case scenario.
- **Service centers:** CPU, HDD and Web server query SCs were needed to represent the actual system. Each SC has an exponential distributed service time with a different rate  $\mu_{CPU}$ ,  $\mu_{HDD}$ ,  $\mu_Q$ . Each request is processed individually in a FIFO order.
- **Handling Transactions:**
  - A new transaction always requires some processing time as a first step.
  - Then:
    - \* with probability **p1** the transaction is terminated.
    - \* with probability **p2** an access to the local disk is required, and then a new CPU processing is required.
    - \* with probability **1-p1-p2** a remote query is required, and then a new CPU processing is required.
  - A reply is sent to the client that originated the request.

Considering these assumptions, it's clear that the system can be represented as a Close Jackson's Network. It follows a potential representation:

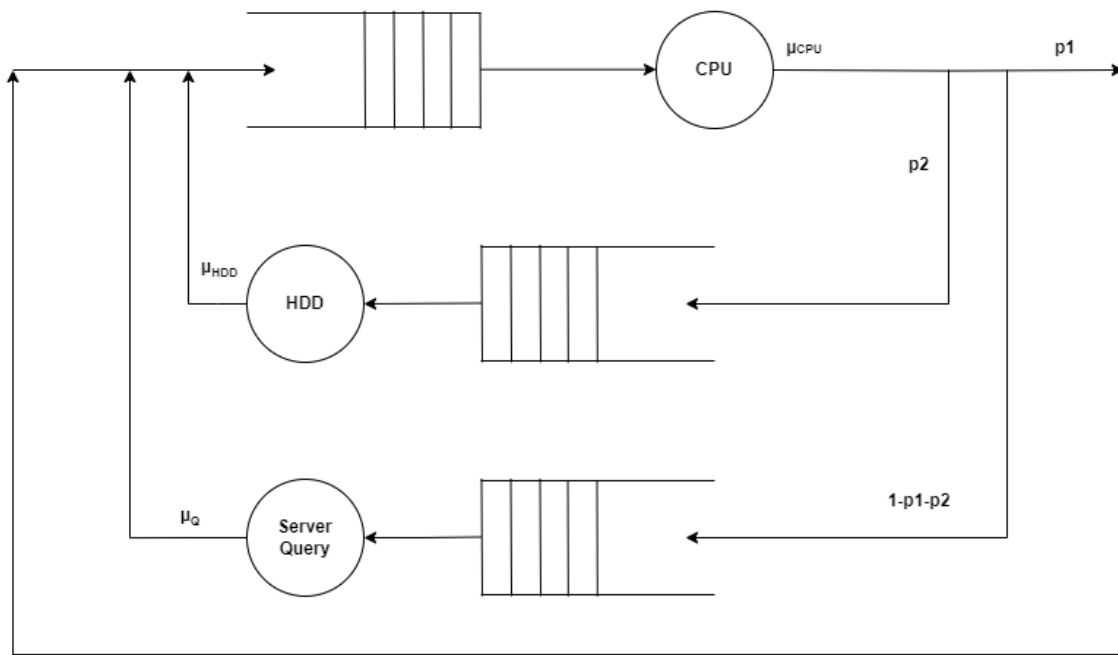


Figure 2.1: Close Jackson's Network modeling representation

## 2.2 Factors

The following factors may affect the performances of the system:

- **N**: the number of clients.
- **p1**: the probability that a transaction is terminated.
- **p2**: the probability that an access to the local disk is required and then a new CPU processing is required.
- **μ<sub>CPU</sub>**: CPU's service rate.
- **μ<sub>HDD</sub>**: HDD's service rate.
- **μ<sub>Q</sub>**: Web query server's service rate.

# Chapter 3

## Implementation

### 3.1 Modules

The following modules have been defined:

- **Client:** a simple module which represents N Clients.
- **Server:** a compound module representing the local server, composed by [Figure 3.1]:
  - **CPU:** a simple module which represents the local processor.
  - **HDD:** a simple module which represents the local hard disk.

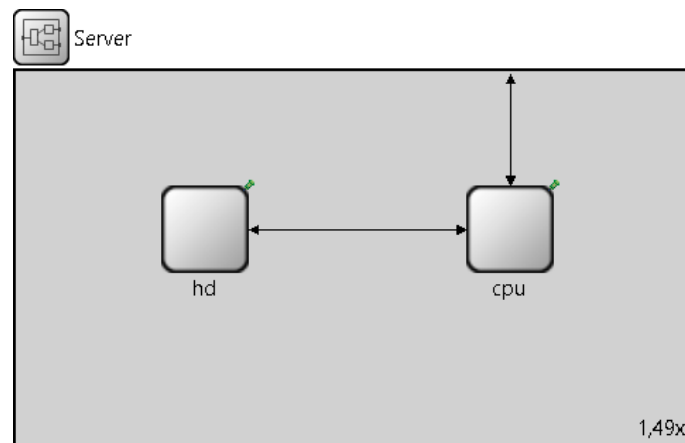


Figure 3.1: Server.ned file

- **Web Server:** a simple module which represents a distant web query server.

The following Figure 3.2 shows the network based on the previous assumptions.

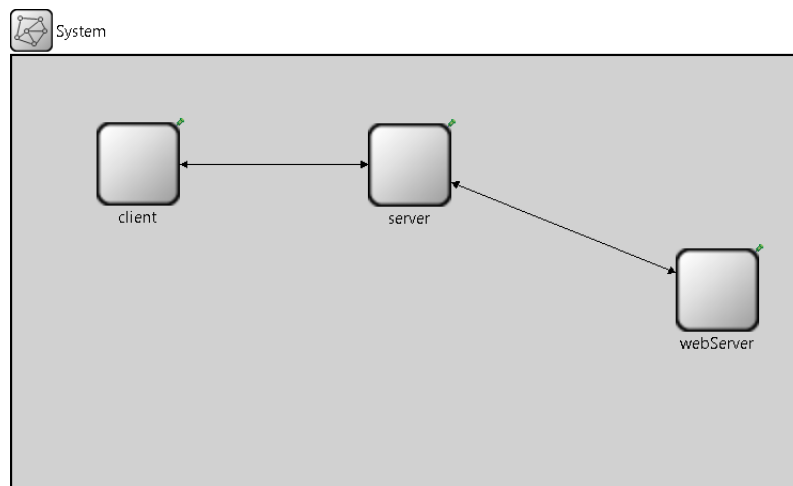


Figure 3.2: System.ned file

## 3.2 Verification



