VIETNAM NATIONAL UNIVERSITY, HO CHI MINH CITY
UNIVERSITY OF TECHNOLOGY
FACULTY OF COMPUTER SCIENCE AND ENGINEERING

PERFORMANCE EVALUATION

Report for Assignment - HK201

Queueing System

$M/M/1/B = \infty/K = \infty/SD = SJF$

Advisor: Tran Van Hoai
Students: Tran Minh Tam - 1813926
Tran Long Vi - 1814804

HO CHI MINH CITY, DECEMBER 2020

Contents

1 Introduction topic

This report includes all the main steps (there are ten in total) required in a performance evaluation project. The following ten sections, each will dive into details of one of the ten steps. The content of each section, subsection consists of one of two parts (or both): general theoretical aspect of the corresponding step and our system's information in specific of that step. The last section is about references referred while doing the project and writing the report.
The queuing system my team chooses is the one that describes a CPU's queuing system. A stream of processes will come to the system to request the CPU, each process has a number of certain IO requests between it's CPU bursts. Each process has 3 states: waiting in the CPU queue, holding the CPU, doing IO.

2 State goals and define the system

2.1 State goals

- Simulating a queuing system by *SimPy*.
  SimPy is a process-based discrete-event simulation framework based on standard *Python*.

- Processes in *SimPy* are defined by *Python* generator functions and may, for example, be used to model active components like customers, vehicles or agents. *SimPy* also provides various types of shared resources to model limited capacity congestion points (like servers, checkout counters and tunnels).

2.2 Define the system by Kerdall notation

2.2.1 Queuing system theoretical basis

Queueing is a process where people, materials or information need to wait at a certain time to get a service. Basically queuing means congestion or crowd. Congestion happens when customers must queue up and await their turn to be served. We can generalize the definition of a queuing system based on the purpose as a transfer of information, materials, people or things between customers and servers.
We can view queuing as a system, which purpose is to transfer people, materials or information. A system consists of many components. Queueing system consists of customers and servers and the facilities where the customers can queue. The customers may arrive at a certain time distribution and certain queuing discipline such as certain prioritization, or first come first serve. The customers may cone from certain limited sources. The servers that provide services to the customers may have certain service time distribution or certain configuration such as serial or parallel servers. The queuing facilities may involve certain designs with limited capacity to reject the customers who arrive after the capacity is reached.

2.2.2 Kerdall notation

A notation used to classify a wide variety of different waiting line.
The Kendall notation system: $A/S/m/B/K/SD$

- A: Denote time probability distribution of arrival processes.

- S: denotes probability distribution of service time. Since time spent servicing a request is random, we use a probability distribution to describe the behavior of how long servers service a process.

- m: number of server in system.

- B: Number of buffers (or system capacity), which is the maximum number of processes can be in a system. Include the ones waiting in queues and the ones serviced at servers.

- K: Population size. Population is the source from which the requests of a system come.

- SD: service discipline is the way a system uses to service a request. there are two categories non-preemptive and preemptive. In non-preemptive, a process will be served until the server finishes servicing the request. In this case, SD is basically how the system picks the next process waiting in the queue to service it.

Our system: $M/M/1/\infty/\infty/SJF$

- $t_1, t_2, \ldots, t_n$ are points of time processes arrive at the system, $\tau_i = t_i - t_{i-1} (2 \leqslant I \leqslant n)$ are interarrival time between 2 successive processes. τ_i form a sequence of independent and identically distributed (IID) random variables. In our system, all τ_i are exponentially distributed and have a common rate parameter λ.

- Our system has 1 server.

- Period times the server spends servicing requests are independent, identically distributed random variables. Those random variables are exponentially distributed with the same rate parameter μ.

- The system has infinite capacity (I.e the system can hold unlimited number of processes).

- The number of requests come to the system is unlimited.

- The system's service discipline is non-preemptive shortest job first. (I.e no interrupt while the server is doing a service for a process, and the next process to be served is the process with the smallest service time. If there are more than one request have the same smallest service time value, the process comes first will be serve first.

2.3 Boundaries

3 List Services and Outcomes

For each performance study, a set of performance criteria or metrics must be chosen. One way to prepare this set is to list the services offered by the system. For each service request made to the system, there are several possible outcomes. Generally, these outcomes can be classified into three categories. The system may perform the service correctly, incorrectly, or refuse to perform the service.
If the system performs the service correctly, its performance is measured by the time taken to perform the service, the rate at which the service is performed, and the resources consumed while performing the service. These three metrics related to time-rate-resource for successful performance are also called responsiveness, productivity, and utilization metrics, respectively.
If the system performs the service incorrectly, an error is said to have occurred. It is helpful to classify errors and to determine the probabilities of each class of errors.

If the system does not perform the service, it is said to be down, failed, or unavailable. Once again, it is helpful to classify the failure modes and to determine the probabilities of each class. A list of services and possible outcomes is useful later in selecting the right metrics and workloads. Our queuing system has one service (not specific, just a service in general). Processes come in a *poison* distributed fashion. If the server is available when a request comes, it will service the request. There is 1 **outcome**: The server performs the service correctly.

4 Select Metrics

- **Average throughput** *(processes / time unit)* (HB: higher is better) is defined as the rate (requests per unit of time) at which the requests are serviced by the system. For batch streams, the throughput is measured in jobs per second.
- **Average response time** *(time units / process)* (LB: lower is better) is the interval between a user's submission and the corresponding completion.
- **Average reaction time** *(time units / process)* (LB) is the time between submission of a request and the beginning of its execution by the system.
- **Average waiting time** *(time units / process)* (LB) is the average total time a process spends waiting in the CPU queue.
- **Probability of CPU being idle** (NB) is the ratio of total idle time and total elapsed time.
- **Average queue's length** (LB) is the average length of the CPU queue.

5 List parameter

5.1 System parameter

System parameters include both hardware and software parameters, which generally do not vary among various installations of the system. system parameters can be adjusted to improve the performance.
Our system parameters: *CPU Service rate μ (process / time unit)*: how fast the CPU servicing requests.

5.2 Workload parameter

The measured quantities, service requests, or resource demands, which are used to model or characterize the workload, are called workload parameters or workload features.

Examples of workload parameters are transaction types, instructions, packet sizes, source destinations of a packet, and page reference pattern. In choosing the parameters to characterize the workload, it is preferable to use those parameters that depend on the workload rather than on the system. For example, the elapsed time (response time) for a transaction is not appropriate as a workload parameter, since it depends highly on the system on which the transaction is executed. This is one reason why the number of service requests rather than the amount of resource demanded is preferable as a workload parameter. For example, it is better to characterize a network mail session by the size of the message or the number of recipients rather than by the CPU time and the number of network messages, which will vary from one system to the next.

Our workload parameters:

- Process arrival rate λ.
- IO requests distributions and it's parameters.
- IO bursts distributions and it's parameters.

6 List factor to study

A factor is a parameter to be varied. Among the workload parameters listed above only the number of users may be chosen as a factor; other parameters may be fixed at their typical values. Not all parameters have an equal effect on the performance. It is important to identify those parameters, which, if varied, will make a significant impact on the performance. Unless there is reason to believe otherwise, these parameters should be used as factors in the performance study.

Factors to study:

- *CPU service rate μ (process / time unit)* with two levels: 4 and 40.
- IO requests distributions of arrival processes are *Poisson* distributions. The rate parameter has two levels: 1 and 10.

7 Select evaluation technique

In this assignment, my team chooses *Simulation technique* to evaluate the performance of the system. Then we'll use *Little's Law* to validate the simulation in a simple manner.

8 Select workload

The workload in our simulation will have some fixed (or depend on other factors as functions) parameters:

- *Arrival rate = 4 (processes / time unit).*
- IO bursts distributions between process's CPU bursts are normal distributions.
- The two parameters of the normal distributions above are dependent functions of service rate: $\mu = 5 \cdot (1 \ / \ service\ rate)$ and $\sigma = 1 \ / \ service\ rate$.

References

[1] *Art of Computer Systems Performance Analysis Techniques For Experimental Design Measurements Simulation And Modeling*, by Raj Jain, Wiley Computer Publishing, John Wiley & Sons, Inc.

[2] Slides of Performance Evaluation course on elearning

[3] Simpy material on website: https://simpy.readthedocs.io/