International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Publishers in a Queuing System. Application to Simulated Tele-Reservation for Banks and Money Transfer Agencies.

Ngindu Mulumba Pascal

Institut Supérieur Des Techniques Médicales

Summary

We have observed with the advent of the banking of salaries of state agents and civil servants in our country the Democratic Republic of Congo in general and in the province of Kasaï central in particular, long queues in different banks and money transfer agencies, where customers sometimes spend entire days without being served. This work consisted in the analysis and the evaluation of the performance parameters of a bank or money transfer agency for an automated management of the queues which are created there in order to improve the quality of service.

Introduction

Nowadays computing is growing in power and glory, it has become difficult and why not impossible to do without it and hope for much better results. Since the advent of banking, banks and funds transfer agencies have displayed distressing scenes: State officials form queues that last for entire days and one wonders if they still have time to work in their offices. We sought to understand the factors at the root of these queues which pose a serious problem of time management.

With the development of information systems, it is necessary to study the behavior displayed by civil servants during the payroll period using appropriate tools. Certainly, when we want to evaluate the performances, we speak implicitly of the calculation of the indices of performance of a system which depend directly on said system. Stochastic modeling in operations research is useful whenever the system includes uncertain but quantifiable elements through probability distributions. The modeling then aims either to calculate the performance measures (averages, distributions, etc.) in order to optimize the system parameters, or to find the best system from a performance point of view among several alternatives.

After stabilizing the system, a customer communication server is set up to allow it to remotely enter the queue. It will also allow him to know the status of the waiting center once his reservation has been validated, to know his position in the queue, and to know when he will be served. The customer after having received all the details necessary for the withdrawal, will only have to launch the application on his mobile, through the internet, to contact the server of the bank or agency to know all this. So he can decide to go to the bank to be served, or wander first to his other occupations so as not to waste his time.

1. Stochastic processes and queues

The constant evolution of information systems presents the real need for tools allowing the study of their behavior. The need for assistance in the phases of design, development and use of these (information) systems is essential. Indeed, the evaluation of the performances is interested in the calculation of the dependent parameters (indexes) of performance of a system (of information), which parameters closely depend on the studied system. There are five most important performance parameters which are actually throughput, latency, response time (which is in practice the sum of service time and latency), average number of a given entity and the utilization rate.

The evaluation of the performance of a system can therefore be done upstream or downstream, that is to say before its construction for an a priori validation of the system's performance during the design phase, or afterwards in order to evaluate performance. In other words, it is done at two levels:

- In design: when the system does not exist,

- In operation: to modify the existing system.

1.1. Stochastic processes

Consider a probability space (Ω, a, P), T an arbitrary set of indices, which can refer to time or space, and E the set of all stochastic realizations on Ω. We call stochastic process or random process, a family of random variables defined on the same probability space, depending on the time parameter.

In other words, a stochastic process represents an evolution over time of a random variable. T and E can be discrete or continuous, so we can have 4 combinations: T discrete and E discrete; T discrete and E continuous; T continuous and E continuous: T continuous and E discrete.

1.2. Markovian stochastic processes (PM)

We will say of a process that it is Markovian (No memory process) if and only if, only the present realizations can influence the future realizations. The formula of this process means that the current state of the system depends on the previous state only. Past history is not taken into account. This is called the memory loss property. Or we can say that predicting the future from the present does not require knowledge of the past.

2. Elements of queuing theory

Queuing theory is a mathematical theory in the field of probability, which studies the optimal solutions for managing queues, or queues. A queue is necessary and will create itself if it is not anticipated, in all cases where supply is less than demand, even temporarily. It can be applied to different situations: management of aircraft taking off or landing, waiting for customers and citizens at the counters, or even storing computer programs before processing them.

This field of research, born in 1917, from the work of the Danish engineer Erlang on the management of the telephone networks in Copenhagen between 1909 and 1920, studies in particular the arrival systems in a queue, the different priorities of each newcomer, as well as than statistical modeling of execution times. It is thanks to the contributions of the mathematicians Khintchine, Palm, Kendall, Pollaczek and Kolmogorov that the theory really developed. A queue is created for each case where customers wish to receive a service from a producer. In computing, we will speak of client and server, elsewhere, we will prefer the terms of units and station.

2.1. General description of a queue problem

When multiple customers simultaneously attempt to obtain service, some must wait and wait in queues, or buffers. The server is then the sending automaton, and the service time is the sending time. Another type of customer is circuit seizure requests in a telephone system. In this case, the servers form a group: these are the circuits serving the requested direction.

Clients arrive at random intervals in a system comprising several servers to which they will address a request. After being served (which supposes a stop at one or more servers as the case may be), customers leave the system. The main purpose of this analysis is to characterize the degree of performance of the system by answering questions such as:

- On average, how long does a customer wait before being served?

- What is the average number of customers in the system?

- What is the average utilization rate of the servers?

In a bank branch, the servers are the counters of the branch. Typically all counters offer the same service, and each customer will therefore only have to visit one counter. Thus, to put some order in the midst of chaos, theoreticians have been led to classify the systems encountered by specifying some of their characteristics. In particular we distinguish:

- Systems with parallel servers, where each client only requests the service of a single server and all the servers are capable of providing this service.

- Serial server systems, where each client must visit several successive servers in a fixed order to receive satisfaction.

Customer arrivals can be grouped or individual; the same applies to the service by each server. Customers form one or more queues, possibly characterized by different priorities (for example, in a workshop, urgent orders or not). Within each queue, the next customer to serve is selected based on a predetermined rule, called service discipline.

The most common service disciplines are:

- First come, first served ;

- Shortest service time first (used in production workshops);

- Last come, first served;

- Random selection, etc.

This is where the notion of a priori impatience appears where the customer who comes does not even enter the system as soon as he thinks it is saturated; and the notion of impatience a posteriori where the customer, after having been part of the system, leaves it before being served. For our study, we exclude the case of impatience a posteriori, i.e. a customer can only leave the system after having been served.

2.2. Performance measures (long-term behavior)

We call state of a system at time t, the number n(t) of customers present in the system at this time (a customer is present in the system if it is in queue or in service) the quantities with which the analysis is interested in the context of the queuing models are the state probabilities, which we define as follows: for n= 0.1, 2, … and t≥0; Pn(t)= probability of state n at time t = probability that n customers are present in the system at time t = Pr[n(t)=n].

We then interpret Pn as the probability that, at any time in the long term, exactly n customers are present in the system. Under rather weak assumptions, it is shown that Pn also represents the proportion of the time (in the long term) where the system contains n customers. The existence of limits (ii) therefore implies that in the long term, the system reaches a state of stable equilibrium, independent of its initial state.

When this is the case, we will often content ourselves with the long-term description of the system provided by (ii) rather than calculating the probability distributions (i) which characterize the system in its non-stabilized phase.

Our theoretical study aims to know in terms of probability, the state of the waiting system at any time. so we can assert that the state of the expectation system is a stochastic process {X(t), t≥0 }. Here X(t) is the state of the waiting system (number of customers) at time t in the waiting system. If it is a Markovian waiting system, this will cause many problems because the latter has no memory.

2.3. General notations of a waiting system

Following Kendall's suggestion, any queuing system must satisfy the following notation which describes its essential characteristics: A|B|X|Y|Z, where:

- The first letter A indicates the nature of the inter-arrivals of the customers (the law followed by the arrivals of the customers-independent random variables-)

- The second letter B indicates the nature of the service durations (independent random variables),

- The third letter X indicates the number of servers or counters in parallel,

- The fourth letter Y indicates the capacity of the waiting system (without or with truncation),

- The fifth letter Z indicates the waiting discipline of the system

3. Client-server model

The client-server appears today as the point of convergence of the needs that companies have been expressing for years: that of integrating new technologies without calling into question existing systems, of providing users with speed, flexibility and comfort in the daily use of IT tools, and more generally, to have an IT system adapted to the high demands of responsiveness imposed by the current economic context. The client-server model makes it possible to take full advantage of this trend in the same way, the economic context encourages organizations using IT to question their operating scheme.

What do we mean by client-server? It is above all a mode of dialogue between two processes: the first called client requests the execution of a service from the second called server. The server performs the services and sends back responses.

In general, a server is able to process requests from several clients. A server therefore makes it possible to share resources between several clients who address it by requests sent in the form of a message. The client-server environment designates a mode of communication through a network between several programs or software: one qualified as a client, sending requests, the other or the others qualified as servers waiting for requests from clients and responding to them. By extension, the client also designates the computer on which the client software is executed, and the server, the computer on which the server software is executed.

4. Web services and the mobile environment

4.1. Definition

A web service is a software system designed to support computer-computer interaction on the network. It has an interface described in a computer-processable format (WSDL is one example). Other systems interact with the web service in a way prescribed by its description using SOAP messages, typically transmitted with the http protocol and XML serialization, in conjunction with other web-related standards.

Web services are modular, loosely coupled, self-describing applications that provide a standards-based model for programming and deploying applications that run through the web infrastructure.

For Berdet (2003), a web service is an application component made available on a network and having methods that can be invoked remotely via the use of standard protocols. Web services have the advantage of being loosely coupled, platform independent and reusable.

4.2. Ad-hoc mobile environment

Mobility (or nomadism) and the new mode of communication used, generate new characteristics specific to the mobile environment: a frequency of disconnection, a flow of communication and modest resources and limited energy sources. Mobile environments offer great flexibility of use. In particular, they allow the networking of sites whose cabling would be more expensive to complete in their entirety, or even impossible (in the presence of a mobile component).

Mobile wireless networks can be classified into two classes: networks with infrastructures which generally use the cell communication model and networks without infrastructures or ad hoc networks. Several systems already use the cellular model and are currently undergoing very strong expansion (GSM networks for example) but require a significant logistical infrastructure and fixed equipment.

5. Design, modeling and deployment

The publishing server can be a vocal and interactive server that will communicate with the client and this is the aspect that most promotes the independence of clients, or a web server through which a mobile application will contact the data server of the branch or bank; this is the aspect whose implementation is easy for us from the point of view of financial and material resources but limits the independence of the client. Thus in the context of this article we will exploit the second aspect, which is very limited in mobility. But before its implementation, let's take a look at our fictitious company “the Pasbak Transfer group”.

The Pasbak Transfer group, like any other transfer agency, has two types of transactions: sending and withdrawing. And it has for each type of transaction two servers. The number of customers showing up per hour for dispatch is 10 and the time required for the service is 20 minutes per customer. The number of customers presenting themselves per hour for withdrawal is 30 and the time required for the service is 10 minutes per customer. We have assumed that arrivals constitute a Poisson process with parameter λ and that services are of negative exponential type with parameter μ. The System allows a total truncation N = 72, due to N1=24 for the sending and N2=48 for the withdrawal. After performance parameter analysis, we concluded that the Pasbak group is not stable because for the send transaction we have:

$\lambda=10$; $\mu=3$; $C=2$; and $N=24$.

The traffic intensity $\rho = \lambda/C\mu = 10/2.3 = 5/(3) > 1$ or a stable system must have $\rho < 1$.

For the withdrawal transaction, we have: $\lambda=30$; $\mu=6$; $C=2$; and $N=48$.

The traffic intensity $\rho = \lambda/C\mu = 30/2.6 = 5/(2) > 1$ or a stable system must have $\rho < 1$

To stabilize it we have increased the number of servers for each transaction, two servers for sending which will make a total of 4, and 4 servers for withdrawal which will make a total of 6. In this way, the traffic intensity $\rho = 5/(6) < 1$ for sending and $p = 5/(6) < 1$ for withdrawing. Once the system has stabilized, we can move on to step two.

5.1. Setting up the publishing server

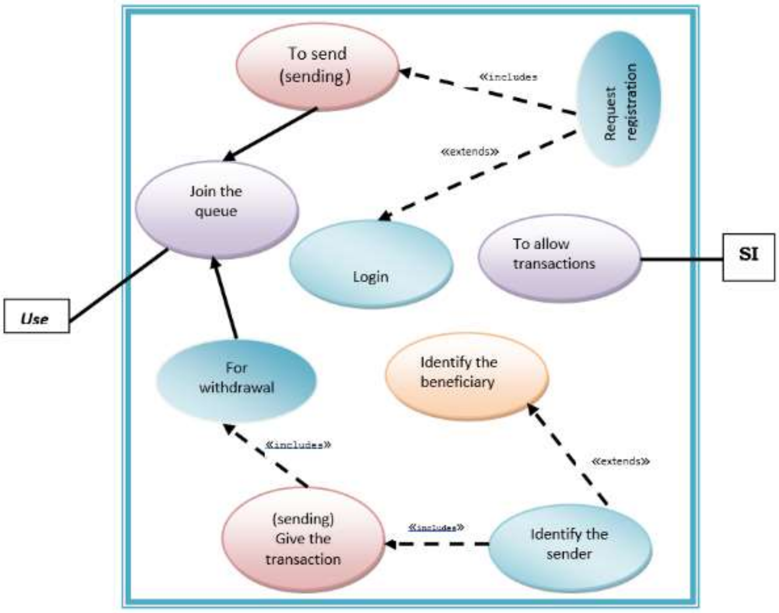
System architecture (tele-reservation)

The architectural constraints vary mainly according to the volumetry of incoming requests but also to the level of interactivity of the Web server with vases of remote or local data.

Mobile app

This is the tool that will be in the hands of the customer wishing to register remotely in the agreement queue of his company. When the customer launches this application on his mobile, a home page paired first to welcome him and lead him to all the other pages or not depending on whether the requirements and conditions will be met or not. This is an application that authenticates the customer by prior verification of the identity of the sender of his prize, that of the beneficiary before any validation of registration in line.

5.2. Use case diagram

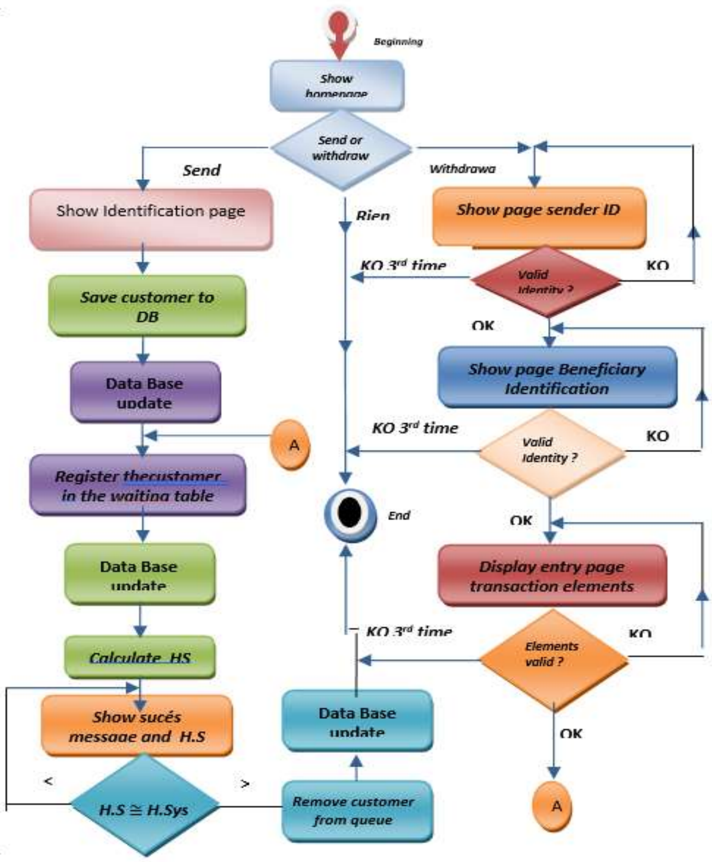


Explanation of the box

The user (customer) launches the application for the sole purpose of registering remotely in the Group Pasbank Transfer queue. There are two variants: Withdrawal and shipping. The shipment does not pose any authentication problem but that of identification. Launch the application and choose "shipping" according to the number of customers awaiting shipment, depending on the service time, an arrival time will be calculated and communicated to them.

As for the withdrawal, the beneficiary customer only has to launch the application and choose the withdrawal option in the main menu. Indeed, the sender must have communicated to him all the elements necessary for the withdrawal. Should it be authenticated by identifying the sender, beneficiary and giving all the constituent elements of the transaction such as the sender, the data. Otherwise, registration will not be supported.

Activity diagram



Explanation of activity

When the application is launched, a welcome page is displayed, welcoming the user and helping the user to use the tool better. He must then choose in the main menu, the type of transaction for which he wants to register in the queue: sending or withdrawal; otherwise after a while the application stops.

If he has chosen to send, an identification page is displayed, allowing him to already give his complete identity. After validation, it is registered in the company's database and it is then updated. The system then registers him in the queue, calculates the service time, based on the average number of customers who precede him and the agency's average service time, and takes his position in the queue. The system will, after that, display the service time (at which it will be served) and its position in the queue through a success message.

At that point, a test begins to check if their system is equal to the service time communicated to the customer. As soon as there are a few minutes left to reach the appointment time, a reminder message is sent to him, with an appendix of his position at this moment in the queue and the time remaining for him to be called. As soon as the hour has struck, whether he is there or not, he is called to be served and directly removed from the queue. This is where he will give the identity of the beneficiary and all the elements of the transaction are produced: amount and codes (not to mention the dates).

If he chose the withdrawal, he must be authenticated as being the real beneficiary of this gain. For this, he must be able to give the identity of the sender, his own identity and the transaction elements in accordance with what is recorded in the database. Indeed, his sender must have communicated this to him as faithfully as possible. Thus, a sender identification page is displayed, allowing him to give the identity of the sender; it will not advance if the data is not accepted. If three times it gives false information, the application stops.

After validation, he is taken to the next page where he will state his own identity so that it corresponds to that given by the sender; it will not advance if the data is not accepted. If three times it gives false information, the application stops. After validation, he is taken to the next page where he will give the elements that constitute the withdrawal transaction, i.e. the time sent, the withdrawal code, the city of origin and the date of sending so that it corresponds to that given by sender it will not advance if the data are not accepted. If three times it gives false information, the application stops.

After validation, the system then places it in the queue and updates the database. It calculates the service time, based on the average number of customers preceding it and the agency's average service time, and takes its position in the queue. The system will, after that, display the service time (at which will be served) and its position in the queue through a success message.

At that time a test begins to check if their system is equal to the service time communicated from service to the customer. As soon as there are a few minutes left to reach the appointment time, a reminder message is sent to him, with his position in the queue at that moment and the time remaining for him to be called. As soon as the hour strikes, whether he is there or not, he is called to be served and directly removed from the queue.

The database server

The PASBANK group needs a database where to store all information related to all customer transactions. It will consist of four main tables:

- Sender table;

- Beneficiary table;

- Shipper reservation table;

- Beneficiary reservation table.

Depending on the cardinalities or types of relationships between these tables, an intermediate table is created between the sender and beneficiary, which will contain the information of the transactions between sender and beneficiary.

Our database server called BD Transfer will complete a total of five tables:

- Tab_exp: the table containing the sender identities,

- Tab_ben: the table containing the identities of the beneficiaries,

- Tab_exp att: the queue of senders,

- Tab_ben att: the queue of beneficiaries;

- Tab_transact: the table containing all the elements relating to a transaction

Queue Implementation

A queue is defined by the following operations:

- Access ;

- Value function (val F: object queue): object;

- Empty queue function (val F: object queue): boolean;

- Amendment;

- Function create File (ref F: object file): empty;

- Put function (ref F: daughter of object; val v: object): empty;

- 163 Scroll function (ref F: object queue): empty;

- File destroy function (ref F: object file): empty.

Conclusion

The microprocessor and also the operating system are introduced in particular in mobiles, the market of which is growing much higher than that of desktop computers. Smart mobile devices improve worker productivity in many areas.

It is therefore hardly surprising that we see a version of Windows appear on a wide variety of mobile devices. The phenomenon is not new, but it has increased since the arrival of Pocket PCs, real pocket computers, as well as Smartphones, these new generation mobile phones operating, for some, under Windows control. These mobile devices are a reduced version of desktop computers with, of course, many differences.

We have therefore just shown how with these devices, we can set up a system for tele-reservation after having improved the performance parameters: The customer launches the mobile application, connects to the company's server via the server Apache web and a purposely implemented web service. The calculation of the time parameters is carried out by the server and the results returned to the client on his mobile…

The mobile is the client, it constitutes the web server (publication server) and the "BD transfer" database constitutes the data server. Let's say, however, that the codes and interfaces are the result of a local execution, thanks to the DotNet simulators. If we want to save customers from the fact that they must all have a Smartphone to use this system, we will then call on a telephone operator for a connection with our server. This way, with a simple phone and without an internet connection, they can use the system.

In this article, it was a question of making a study and an analysis of the performances of the existing system to see if it is stable or not, in order to propose the solution (manual) which consisted in balancing the system by the use of tools appropriate (stochastic process) and elements of the theory of queues in case the stability is not proven and The solution (computing) which consists in setting up a publication server which will be in communication with the clients of the Agency.

Bibliography

1. Bardet White Paper: web services. Paris: Dunod. Of France 2003.

2. J.-M. PHP/MySQL, : with dreamweaver. Paris: Eyrolles 2005.

3. Discala R. Programming in net Frame work with the C# version 2.0 language. Algiers: Editions Berti, 2004

4. Guignard Damien et al : Android programming: from design to deployment with the Google android 2 SDK. Paris: Editions Eyrolles, 2005.

5. Kadima H. : Web services. Paris: Editions Dumod. 2003.

6. Leblanc G. : C# and .Net version 2. Paris: Editions Eyrolles. 2005

7. Libero Maesano et Al. Web services with J2EE and .Net: design and implementations. Paris: Edit, Eyrolles. 2005

8. Maise S. Installation and configuration of a web server. 2nd Edition. 2005

9.Robert Faure: Summary of operational research 7th edition, method and application exercise, dinod 2014.

10. Ouakil L et al. Telephony over IP. Paris: Editions Eyrolles, 2005.

11. Pujolle G. et al Networks. Paris: Editions Eyrolles. 2005.

12. Ricardo de la Rosa-Rosero : Discovery and selection of web services for a melusine application. Geneva, 2004.

13. Rigaux P. : Practice of MySQL and PHP, design and creation of dynamic websites. Paris: Editions Dunod, 2009.

14. Soutou C. : Learn SQL with MySQL with 40 corrected exercises. Paris: Editions Eyrolles. 2005.

15. Sulla Foutou : Study and design of an interactive vocal server, application to the session of the notes of the computer section, 2006.

16. Valois F. : Network performance modeling and evaluation. University of Lyons. 2003.

17. Frédéric Meunier : Introduction to operational research, ERMICS, Paris, 77455

18. Jacques teghem operational research volume 1, random model production management. Edit. Ellipses, 2013.

19. Bruno Baynat Queuing theory. Eyrolles bookstore, 1970.

20. Vladimir A. and Nikolaos L.: Queuing Theory. ISBN, 2021.

21. Mitewu kabamba Alex : Banque commercial et Gestion des files d'attente, broché 2019.