ICEB 2019 Proceedings

International Conference on Electronic Business (ICEB)

Winter 12-8-2019

A Switch on Electronic Commerce Mobile Payment: From Traditional Queuing to Elastic Request as a Payment Service Based on the Edge Computing Model

Hongbin Shen

Bichuan Shen

Penghao Ye

Shen, H. B., Shen, B. C., & Ye, P. H. (2019). A switch on electronic commerce mobile payment: From traditional queuing to elastic request as a payment service based on the edge computing model. In *Proceedings of The 19th International Conference on Electronic Business* (pp. 13-26). ICEB, Newcastle upon Tyne, UK, December 8-12.

A Switch on Electronic Commerce Mobile Payment: From Traditional Queuing to Elastic Request as a Payment Service Based on the Edge Computing Model

(Full Paper)

Hongbin Shen*, Chongqing Technology and Business University, Chongqing, China, 857760697@qq. com
Bichuan Shen, Chongqing University of Technology, Chongqing, China, shenbc@cqut. edu. cn
Penghao YE, Zhongnan University of Economics and Law, Wuhan, China, paulyph@whu. edu. cn

ABSTRACT

The most common consumption payment problems are time delay, efficiency loss and poor experience caused by Queuing service congestion. This paper claims that a secure and highly efficient online mobile payment protocol is necessary. According to the special requirements of massive payment requests, high elasticity, high concurrency, high risk and instant response in mobile e-commerce, this paper propose an appropriate model for the application requirements of the service pattern for edge computing model-computing is service pattern, and the task model, computing model and service model provided by this pattern are abstracted. Finally, through LoadRunner load testing simulate request as a payment service model. Simulation results show that the construction of e-commerce retail payment network queuing model and mobile security payment protocol can solve the conversation of payment queuing efficiency and reduce waste of resources, and customers can enjoy safety payment experience.

Keywords: Queuing theory, mobile payment, edge computing, LoadRunner load testing.

*Corresponding author

INTRODUCTION

Congestion is a common phenomenon in social life. Congestion, also known as congestion, is extremely common in discrete events, such as road traffic, telephone exchange, airport dispatching, medical treatment, railway Spring Festival travel tickets, supermarket payment, water and electricity payment, etc. This is not only a common social life phenomenon, but also an important field of economic research. From an economic perspective, congestion is caused by demand exceeding supply and scarce resources. In queuing, the most important reason for congestion is the currency payment caused by the use of goods and services, and the queuing problem caused by the completion of the transfer of currency ownership by both the supply and the demand. Because of universal existence in traditional retail sales, by increasing the physical counter and teller to reduce and eliminate the payment queue congestion, can cause appear alternately season settlement services in short supply and the off-season of clearing idle and waste of resources, the counter even in elasticity to the counter and teller resources optimization scheduling model cases.

LITERATURE REVIEW

In the study of queuing problem, as early as 1838, French mathematician Simeon D, Poisson (Land, McCall, & Nagin, 1996) studied the discrete probability distribution commonly seen in statistics and probability. Erlang studies the customer arrival density function (Fallah Nezhad & Akhavan Niaki, 2011), which is often used in telephone and communication system queuing. Jones and Peppiatt (1996) analyzed the relationship between actual queuing time and acceptable waiting time of customers in the service industry and verified it through empirical research. Baldinger and Rubinson (1996) established the queuing theory model to verify the effect of dynamic adjustment of service capacity on easing queuing congestion in the service industry. Houston, Bettencourt, and Wenger (1998) studied the relationship between customer queuing time and customer perception of service quality in the service industry. The above scholars all studied the payment settlement queuing problem in physical retail under the background of industrial economy based on the existence of physical settlement counters. Under the background of network economy, physical settlement counters have gradually evolved into network virtual payment (Mills et al., 2016), and mobile payment has become the inevitable direction of future payment development (Fan, Xu, & Wan, 2013). Therefore, the traditional queuing research has turned to network queuing and its solutions. On the basis of computer payment DES and RSA data encryption protocol (Forouzan, 2007), some scholars designed a authentication and payment protocol applicable to mobile environment in the improved Random Early Detection (RED) nonlinear algorithm (Zhang, Ma, Wang, & Chen, 2011). However, due to the large amount of computation of the protocol, the user's waiting time for payment is delayed, which can only be used for small payment. In order to reduce the computational load, some scholars proposed to construct a mobile phone payment system based on signature encryption algorithm (Floyd & Jacobson, 1993). Other scholars proposed to make up for the deficiency of transaction speed and security in the above mobile payment schemes by means of the third-party payment platform and the key distribution mechanism of "one key at a time" (Floyd & Jacobson, 1993). In order to improve the security of payment protocol. Xie (2008) proposed a compound payment protocol that can meet multiple security attributes.

Based on the traditional retail queuing line model, this paper extends it to e-commerce payment queuing and constructs the $M/M/n/m$ payment request queuing circle model. Furthermore, this paper propose an appropriate model for the application requirements of the service pattern for edge computing model, and service model provided by this pattern are abstracted. Finally, through LoadRunner load testing simulate request as a payment service model. Simulation results show that the construction of e-commerce retail payment network queuing model and mobile security payment protocol can solve the conversation of payment queuing efficiency and reduce waste of resources, and customers can enjoy safety payment experience.

Basic concept of physical retail payment queuing

Definition 1 A system consisting of one type of multiple services (desks) to provide certain services for random arrival consultants, and a system in which the customer views the server (desk) as "idle" or "busy" and is served or waited according to its queuing law is called queuing (service) system. Here, customer and waiter are generalized.

Discrete event system uses stochastic process theory and probability statistics to describe the main elements of retail event system: entity, event, activity and process. For this reason, the discrete system is represented as a logical set of external events (X), output events (Y), sequential states (δ), output functions (λ), and the logic set $M=(X,Y,S,\delta,\lambda,t_0)$ of the time advancement functions (t_0).

Definition 2 In the retail trade customer and the waiter currency payment settlement queue as a queuing system, its basic structure consists of four parts including the input process (customer traffic), service time (business processing time, non-business processing time customer stop line, stop also provide payment services), service (service window settings) and queuing rules.

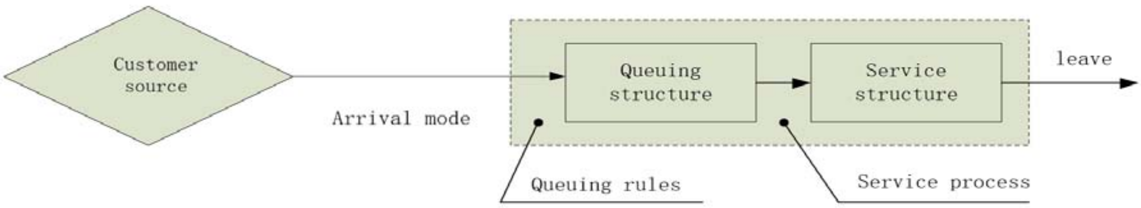

Figure 1: Composition of retail queuing service system.

Queued arrival mode

Describe the distribution pattern in which customers randomly come to the settlement counter in a retail store and request payment and settlement services. Arrival mode, also known as system input process, is represented by random arrival time. Common arrival modes include: constant length distribution, Poisson distribution, Erlang distribution, exponential distribution and super exponential distribution.

(1) Fixed-arrival distribution: the arrival of the entity (customer) in the equidistant time interval is the fixed-length arrival distribution, and the mathematical expression is

$$A_0(t)=p\{T\le t\}=\begin{cases}0, & t\ge a\\ 1, & t\le a\end{cases} \tag{1}$$

(2) Poisson arrival distribution: an entity (customer) has an arrival probability within a given length of time. This is the typical arrival distribution in the payment queuing service system of the retail industry

$$p(n)=\frac{(\lambda t)^n e^{-\lambda t}}{n!}, \qquad n=0,1,\cdots \tag{2}$$

(3) Erlang distribution: often used in telephone and communication switching systems whose mathematical expression is

$$A_0(t)=e^{-k\lambda t}\sum_{n=0}^{k-1}\frac{(k\lambda t)^n}{n^!} \tag{3}$$

k is a positive integer greater than zero.

(4) Exponential arrival distribution: is a universal arrival distribution, whose cumulative function is

$$F(t) = y = 1 - e^{-\lambda t} \tag{4}$$

If the random number y is uniformly distributed, it can be simplified as

$$t = \frac{-1n(y)}{\lambda} = T_a 1n(y) \tag{5}$$

Where T_a is the mean mutual reaching time.

(5) Hyper exponential arrival distribution: this is a type of distribution where the variance is greater than the mean. It's main characteristic is that the data appears low value and high value more frequently. Generally, when the coefficient C_v of variation is greater than 1, the data can be fitted with a super exponential distribution. Distribution function is

$$A_0(t) = Se^{-2s\lambda t} + (1-S)e^{-2(1-s)\lambda t}, \qquad 0 \le S \le \frac{1}{2} \tag{6}$$

$S = KT^2$ is the variance; $C_v = K^{\frac{1}{2}}$ is the coefficient of variation, where $K = \frac{(1-2S-2S^2)}{2S(1-S)}$

Queuing rules
A queuing rule is a rule that selects the next customer from the queue when the counter clerk completes the billing service for the current customer. Due to the randomness of customer arrival and service time, there are many queuing rules. The queuing rule is called the queuing law. Retail stores usually take First Come First Service (FCFS) law, or priority service such as VIP service law. If the captain exceeds N, the customers will leave automatically when they arrive later. When the customer's waiting time is longer, the customer will leave automatically.

Service organizations
The configuration of a certain number of service tellers and counter machines that provide settlement services to customers. Such as several service counters in parallel, serial, or other connection mode. Retail stores generally adopt single queue, single service counter system.

Service process
The service process that takes time to complete the settlement service for each customer. It describes the statistical characteristics of service processes and also has some distribution characteristics. It usually has the following distribution:

(1) Service time of fixed length distribution. All customer service hours are constant, denoted by D

(2) Negative exponential distribution of service time. Each customer service time $\frac{1}{\mu}$ is independent of each other and has the same negative exponential distribution. The M is the average service time.

(3) Erlang distributes service hours. If the service of customers must go through a series of K service counters, the service time T_i of each service counter is independent of each other and subject to a negative exponential distribution with a mean value of E_k .

(4) Service hours are generally randomly distributed. For example, service tellers may have different service time distribution, and the service time distribution depends on the team leader.

Physical retail payment queuing service system modeling
The basic mathematical model of payment, settlement and queuing in physical retail stores can be expressed as a $M|M|C$ model. The $M|M|C$ model represents a queuing system model in which the input process (customer arrival) is Poisson's input, service time obeys negative exponential distribution. The main quantitative indicators of the model can be expressed as:

Ls : Represents the number of customers in the system, including all customers waiting in line and being served (also known as the average captain).

Lq : Represents the number of customers waiting in line in the system (called the average queue length).

Tq : Represents the average waiting time of customers in the system (i. e. the average queuing time).

Ts : Represents the average customer stay time in the system (including waiting time and service time).

λ : Represents the average customer arrival rate (known as customer arrival rate).

μ : Represents the average service rate of the system (i. e. the average service rate of the service desk).

ρ : Represents service intensity, whose value is the ratio of effective average arrival rate λ to average service rate μ, i. eρ=λ/ρ.

The less the waiting time, the better the system performance. Obviously, this is something that customers and physical retail stores are very concerned about. Customers want to queue as short as possible. Similarly, for cashiers, the service intensity is as small as possible.

Model solution

Firstly, we discuss the situation $C=1$, the model becomes $M|M|1$, that is, the model represents the situation that there is only one settlement counter, so $\rho=\frac{\lambda}{\mu}$, when $\rho<1$. In other words, when the average number of customers arriving per unit time is less than the average number of customers served, the queue length can avoid infinite growth and reach equilibrium. The probability of having a customer in the system at any given time t there are n customers is $p_n(t)$. When the system reaches a stable state, the probability $p_n(t)$ tends to a stable state p_n, here p_n is not related to t, the system is said to be in statistical equilibrium, also p_n is the steady-state probability under the quilibrium state. It represents the probability that the system has n customers under the stable state, where $P_n=(1-\rho)\rho^n$, specifically $P_o=1-\rho(\rho<1)$.

The following expression can be obtained by using probability statistics:

(1) The average number of customers (average queue length) in the system Ls is:

$$Ls=\sum_{n=1}^{\infty}nP_n=\frac{\rho}{1-\rho}=\frac{\lambda}{\mu-\lambda} \tag{7}$$

(2) The average number of customers waiting in a queue (average queue length) Lq is:

$$Lq=\sum_{n=1}^{\infty}(n-1)P_n=Ls-\rho=\rho Ls \quad \left(Ls=\frac{\rho}{1-\rho}\right) \tag{8}$$

(3) The average customer stay time Ts in the system and the average wait time Tq in the queue are respectively:

$$Ts=\frac{1}{\mu-\lambda}=\frac{Ls}{\lambda} \qquad Tq=Ts-\frac{1}{\mu}=\frac{Lq}{\lambda} \tag{9}$$

Secondly, we suppose that the average service rate of each billing counter is the same μ, at this time, the average service rate of the whole system is $C\mu$, so service intensity $\rho=\lambda/C\mu$, when $\rho<1$, There is an equilibrium state in the system, and the probability of the number of customers n in the steady-state system at any time is $P_n=P\{N=n\}$; especially when $n=0$, P_n is P_0, P_0 represents the probability that all settlement counters in the steady-state system are empty (because the number of customers in the system is 0). According to queuing theory and probability statistics:

$$P_0=\left[\sum_{k=0}^{C-1}\frac{1}{k!}\left(\frac{\lambda}{\mu}\right)^k+\frac{1}{C!}\left(\frac{\lambda}{\mu}\right)^C\frac{C\mu}{C\mu-\lambda}\right]^{-1} \tag{10}$$

At this point, the performance indicators of the model are as follows:

(1) Average queue length Lq is:

$$Lq = \sum_{n=C+1}^{\infty}(n-C)P_n = \frac{(C\rho)^C}{C!(1-\rho)^2}\rho P_0 \tag{11}$$

(2) Average queue length Ls is:

$$Ls = Lq + C\rho \tag{12}$$

(3) The average customer stay in the system Ts is:

$$Ts = \frac{Ts}{\lambda} \tag{13}$$

(4) The average time a customer waits in a queue Tq is:

$$Tq = \frac{Lq}{\lambda} \tag{14}$$

According to the above expression, as long as know the average arrival rate λ of customers in the system and the average service rate μ, the system can calculate the average customers waiting time to improve the quality of service, as shown in figure 2.

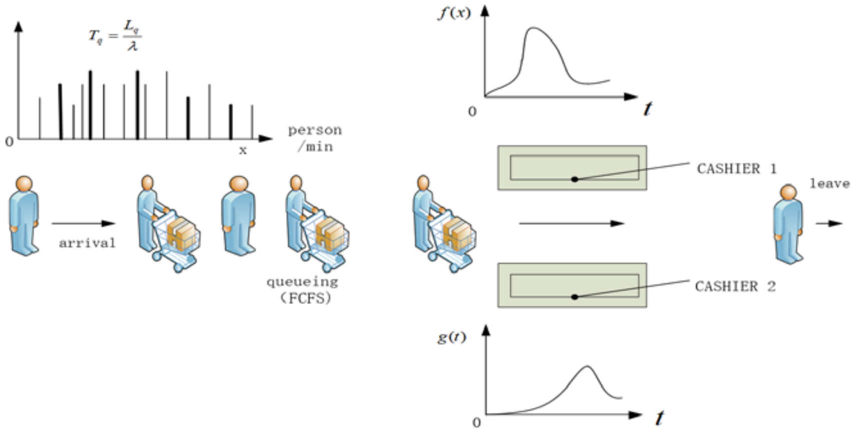

Figure 2: Linear system model of payment queuing in physical retail stores.

HYPOTHESES DEVELOPMENT

Modeling of e-commerce queuing payment

In e-commerce payment, the e-commerce platform does not need to provide customers with the necessary physical settlement counters and tellers in physical retail payment. Customers only need to send a payment request to the payment link page, and the payment system provides corresponding services according to the customer's request. The customer and the payment service provider can be regarded as queuing according to the circumference regardless of the distance between the spatial entity and the virtual space of the network. The customer and the payment service provider have the same space distance. The queuing rule is Along With the Service (AWS). The customers complete the whole payment process by himself. Its queuing model can be constructed as figure3.

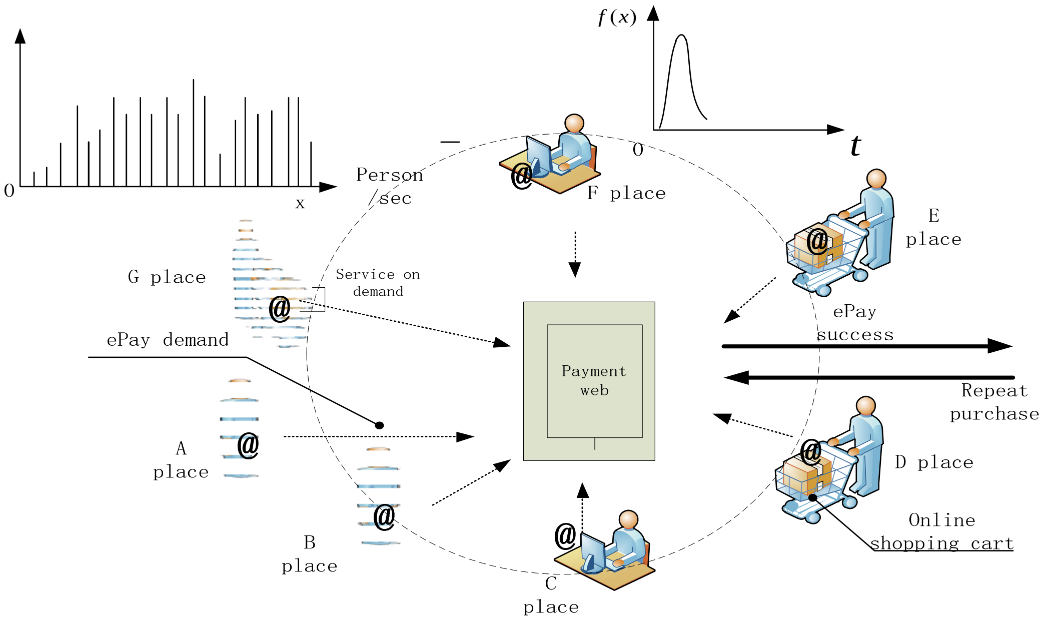

Figure 3: e-commerce payment queue circumference system model.

E-commerce payment queuing model solution

It is assumed that there are new service windows which can provide the required file access in the file list managed by the network payment node, and the payment requests made by users or connected to the payment node reach the system according to Poisson distribution, with the strength of λ. We suppose that each service time that can provide payment request obeys negative exponential distribution, and the intensity is μ. The characteristic of this model is that the payment page node can accommodate up to one user's payment request m, namely customer ($m > n > 1$). If the payment page node is full (that is, there are already m payment requests), the new payment page node request will immediately leave and find another payment page node to provide services. Obviously, there must be a place in the system for customers $m-n$ to wait in line.

So assume $n=8$, $m=12$, $\lambda=10$, $\mu=4$, $\rho_1=\frac{\lambda}{\mu}=2.5$, $\rho=\frac{\lambda}{n\mu}=0.3125$

Then

$$p_0=\begin{cases}\left[\sum_{k=0}^{n-1}\frac{(n\rho)^k}{k!}+\frac{(n\rho)^{m-n+1}}{1-\rho}\right]^{-1} & \rho\neq 1\\ \left[\sum_{k=0}^{n-1}\frac{n^k}{k!}+\frac{n^n}{n!}(m-n+1)\right]^{-1} & \rho=1\end{cases} \tag{15}$$

$$p_k=\begin{cases}\frac{\rho_1^k}{k!}p_0=\frac{n^k\rho^k}{}p_0 & 0\le k\le n\\ \frac{\rho_1^k p_0}{n!n^{k-n}}=\frac{n^n\rho^k p_0}{n!} & n\le k\le m\end{cases} \tag{16}$$

(1) The probability of losing the payment request at the payment node is:

$$p_0=\left[\sum_{k=0}^{7}\frac{(2\ \ 5)^k}{k!}+\frac{(2\ \ 5)^8}{8!}\square\frac{1-(0.3125)^5}{1-0.3125}\right]^{-1}=0.08263801$$

(2) The relative passage capacity of the payment node:

$$Q = 1 - p_m = \frac{n^m \rho^m}{n!} p_0 = \frac{8^8 \cdot (0.3125)^{12}}{8!} p_0$$

(3) The average number of requests per unit time to the payment node:

$$\lambda_e = \lambda(1 - p_m) = \lambda Q = 9.99970383$$

(4) The average queue length of a payment request queued at the payment node:

$$L_q = \sum_{k=n}^{m} (k-n) pk = \sum_{k=8}^{12} (k-8) pk = 0.001979834$$

(5) The average queue length of the payment request waiting at the payment node:

$$L_s = L_q + L_{\text{服}} = L_q + \frac{\lambda_e}{\mu} = 2\ 501905792$$

(6) According to little formula, the average stay time and average queuing time of the payment request at the payment node can be obtained:

$$W_s = \frac{L_s}{\lambda_e} = W_q + \frac{1}{\mu} = 0.250197989$$

$$W_q = \frac{L_q}{\lambda_e} = 0.000197989$$

E-commerce queuing mobile payment initial assumption

(1) Merchants and Banks have their own digital certificates.

(2) The customer and the merchant have registered with the bank, and the customer and the merchant have their own accounts ID_X .

(3) Customers and merchants trust the bank.

(4) A series of safe and efficient pseudo-random numbers can be generated between the customer and the bank through a common initial seed, which has been deposited in the mobile phone when the customer registers in the bank. Registered in the bank according to the assumptions, customers via secure off-line methods set up efficient pseudo-random number generator cyclic shift and hash function operation, can produce a series of pseudo-random number PRN_i . Each pseudo-random number is based on a pseudo-random number.

Definition:

C , M , B : Represent customers, businesses and Banks respectively.

ID_X : Identity includes all information of X .

DC_X : Digital certificate X .

PK_X : Public key X .

SK_X : Private key X .

PSW : Client's banking login password.

$E(msg)K$: Symmetric encryption computation on msg with a confidential key K .

$H(msg)$: Hash the message msg.

Signx (msg) : Sign the message msg using the key K.

TID : Transaction characteristics including time and date.

OI : Order information = $OI = \{TID, OD, \Pr ice, Description\}$, here OD means the type and quantity of the order, $Description$ represents the description of the digital product ordered.

PAY : Payment information, including the account number of the transfer and the payer's own account information.

$PAY = \{TID, \Pr ice, Account_C, Account_M\}$.

PRN_i : The initial seed produces the i^{th} in a series of pseudo-random numbers.

$Y_{M,C}$: Shared key between merchant and customer.

$\otimes$ Exclusive or operation.

Payment protocol

When the price of goods or services has been negotiated between the two parties to the transaction, the customer initiates a payment request. If one party does not receive a reply within the specified time, the agreement shall be terminated. The payment procedure is as follows:

Step1 $C \rightarrow M$

$$ID_C, ID_M, OI, i, HASH\left(ID_C, ID_M, OI, i\right), PRN_i \otimes PRN_{i+1} \otimes PSW \qquad (17)$$

The customer initiates a payment request to the merchant, and the request message includes the identity information of the customer and the merchant, the order information and the corresponding integrity authentication information. Finally, the authentication information of the customer to the bank is attached, which generates two pseudo-random numbers and the difference or result between the bank login password for the customer offline. The main purpose of using two pseudo-random numbers for exotic operation is to prevent the user password from being guessed correctly and revealing the pseudo-random number information.

Step2 M→B

$$E\left\{\text{Sign}_{SK_M}\left\{ID_C, ID_M, i, PRN_i \otimes PRN_{i+1} \otimes PSW, PAY, Y_{M,C}, E\left\{ID_M, OI\right\}_{Y_{M,C}}\right\}, DC_M\right\}_{PK_B} \qquad (18)$$

After receiving the payment request from the customer, system will check the order information and verify its integrity. After confirmation, it will send the transfer request message to the bank. The content of the message includes the identity information of the customer and the merchant, the digital certificate of the merchant, the account information, the authentication information forwarded by the customer to the bank, the Shared key generated by the merchant for the transaction with the customer and the identity and billing information encrypted with the shared key. The transfer request message is signed with the private key of the merchant and then encrypted with the public key of the bank to ensure the security and non-repudiation of the message.

Step3 B→C

$$PRN_{i+2} \otimes PRN_{i+3} \otimes E\left\{ID_M, OI\right\}_{Y_{C,M}}, PRN_{i+4} \otimes PRN_{i+5} \otimes PAY, PRN_{i+6} \otimes PRN_{i+7} \otimes Y_{C,M} \qquad (19)$$

After the bank receives the customer authentication information forwarded by the merchant, it compares whether the value of I is consistent with its pseudo-random sequence. If consistent, the corresponding pseudo-random number PRN is generated. The authentication message is exotic calculated with PRN_{i+1} to obtain the login password PSW. If the calculated PSW is inconsistent with the original password, the protocol terminates and a short message notifies the client to proofread. If the pseudo-random number, customer identity information and login password correspond, then the authentication. After confirming the customer information, use two pseudo-random numbers for the account information forwarded by the merchant, the shared key, and the order information encrypted with the shared key PRN_{i+2} to PRN_{i+7}, perform exotic encryption. The bank will send the above information to the customer and wait for the customer's final confirmation.

Step4 C→B

$$PRN_{i+8} \otimes PRN_{i+9} \otimes E\left\{ID_{C,} OI\right\}_{Y_{C,M}} , PRN_{i+10} \otimes PRN_{i+11} \otimes PAY \tag{20}$$

After the customer receives the confirmation information sent by the bank, the corresponding pseudo-random number is generated from PRN_{i+2} to PRN_{i+7} and perform exotic operation to obtain the account information and the shared key information with the merchant, and then use the shared key to unlock the encrypted order information. After the confirmation of the customer, two pseudo-random numbers from PRN_{i+8} to PRN_{i+11}, express customers' confirmation and agree to transfer. The transaction is terminated if the customer receives an objection to the information.

Step5 B→M

$$E\left\{Sign_{SK_B}\left\{E\left\{ID_C, OI\right\}_{Y_{C,M}} , PAY\right\}, DC_B\right\}_{PK_M} \tag{21}$$

After the bank receives the confirmation information sent by the customer, the corresponding pseudo-random number is generated from PRN_{i+8} to PRN_{i+11}. Notify the merchant that the transfer is successful. If the bank fails to receive the confirmation of the goods within the stipulated time, the transaction will be terminated.

METHODOLOGY

Elastic request as a payment service model

In recent years, the computing workload of e-commerce mobile payment task has been migrating, first from the local data center to the cloud, and now increasingly from the cloud data center to a location closer to the data edge. The purpose of doing this is to solve the problem faced by cloud computing: the linearly growing centralized cloud computing ability cannot adapt to the rapid growth of massive edge data, especially the massive centralized mobile payment requests and sudden growth, which makes the network edge devices transmit a large amount of data to the cloud computing center. At the same time, limited power network edge devices will consume a lot of power when they transmit data to the cloud computing center. Moreover, the personal privacy problem of network edge data is particularly prominent, which makes the core payment data in e-commerce have the risk of being eroded.

The key elements in the task model include time delay, bandwidth utilization, context awareness, generality and elastic computing power. Although the development of accurate models is very complex for tasks, for edge computing models, mathematical optimization allows simple tasks to meet the requirements of zero delay, zero error and zero leakage. The following are two commonly used calculation task models.

Task model of binary transfer

Highly integrated or relatively simple tasks cannot be partitioned, and must be performed as a whole in the edge device or transferred to the edge computing center server for processing, which is called binary transfer. Such task can be represented as (L, TD, x) by three symbol field, which contains the following information: task input data size L (in bit), completion period TD, calculation workload x (in CPU clock period). The use of these parameters can not only capture the basic properties of the edge device processing tasks, but also facilitate the simple evaluation of execution delay and the estimation of energy consumption performance.

Task (L, TD, x) needs to be completed before the hard deadline TD. This model can also be extended to deal with the soft deadline requirements, allowing tasks with large amount of data to be completed later. In this case, the computational effort required to perform the input data of the 1 bit task is modeled as the random variable x.

In practice, many edge tasks are composed of many subprocesses, which makes it possible to transfer each subprocess to local edge devices for execution, so as to achieve fine-grained task transfer. Specifically, the task can be divided into two parts: one is the task processed by the edge device itself, the other is the task performed by the edge device or the edge computing center which is closer to the device, or it can be coordinated by the edge computing center.

The simplest task model for local transfer is the data partition model, in which task input is bit independent. This task model can be arbitrarily divided into different groups, which can be implemented in different entities under the edge computing system. However, there are obvious dependencies between different processes or components. How to deal with these dependencies will significantly affect the efficiency of execution. There are two key problems:

(1) The execution sequence of functions or routines, because the output of some processes or components is the input of other processes or components, the control of execution sequence is the guarantee to realize the correctness of local transfer.

(2) Due to the limitation of the software and hardware of the edge devices, at present, we can only achieve isomorphic transfer, and cannot transfer the task to the heterogeneous devices. The abstraction of the software and hardware of the edge devices is the guarantee to achieve the maximum utilization of the local transfer.

CPU performance is represented by CPU cycle frequency f_m (CPU clock speed). The architecture of edge computing equipment adopts Dynamic Frequency and Voltage Scaling (DVFS). It can adjust the energy consumption by changing the cycle frequency of CPU. In practice, f_m is limited by the maximum $f_{CPU}^{\max}$ (the maximum computing power that edge devices can provide). In combination with above mentioned task model A (L, T_d , x) introduced in the previous section, the delay of the model can be evaluated with the following formula:

$$t_m = \frac{LX}{f_m} \qquad (21)$$

Among them, $f_m \le f_{CPU}^{\max}$

Equation (21) reflects the delay TM affected by the computing performance. FM represents the computing power possessed by the edge device. When the content of LX processing is certain, the delay of the system can be adjusted by adjusting FM. Increasing FM can reduce the delay, but decreasing FM will increase the delay. According to the CPU energy consumption, FM reflects the energy consumption. Therefore, edge devices need to allocate energy consumption and delay reasonably. When low latency is needed to ensure the real-time performance of tasks, FM can be appropriately increased; otherwise, when low power consumption is needed to ensure the energy-saving performance of tasks, FM can be appropriately reduced. DVFS is the key technology to provide this capability, which is an elastic request as a payment service model.

According to the service characteristics of edge computing devices, two widely used service models are abstracted according to the requirements of low latency and low energy consumption.

Sequential service model

As shown in Figure 1, the model is mainly for services with clear sequence requirements. Each node is represented by $e(i, j)$, and the subscript i represents tasks in different stages. The subscript j only represents the node number that jointly completes j tasks at the same time as the subscript i. the node is characterized by isomorphic architecture. The advantages of this model: it can provide different services according to different users, making the service more user-friendly; the same tasks with subscript i can provide low latency services according to binary transfer or local transfer task model; it can increase or reduce the number of j by DVFS technology according to the requirements of service delay, with flexible structure and maximum utilization of energy consumption. The following presents the delay t_{total} (minimum delay time) and energy consumption E_s (minimum energy consumption) model of the structure:

$$t_{total} = \min_{1 \le i \le n, 1 \le j \le m} \sum_{i=1}^{n} \frac{L_i X_i}{\sum_{j=1}^{m} f_j} \qquad (22)$$

Therein, $n \in Z^+, m \in N^+, f_j \le f_{CPU_j}^{\max}$

$$\mathrm{E}_s = \min_{1\le j\le m} \sum_{j=1}^{m} f_j \qquad (23)$$

Therein, $m = Z^+, f_j \le f_{CPU_j}^{\max}$

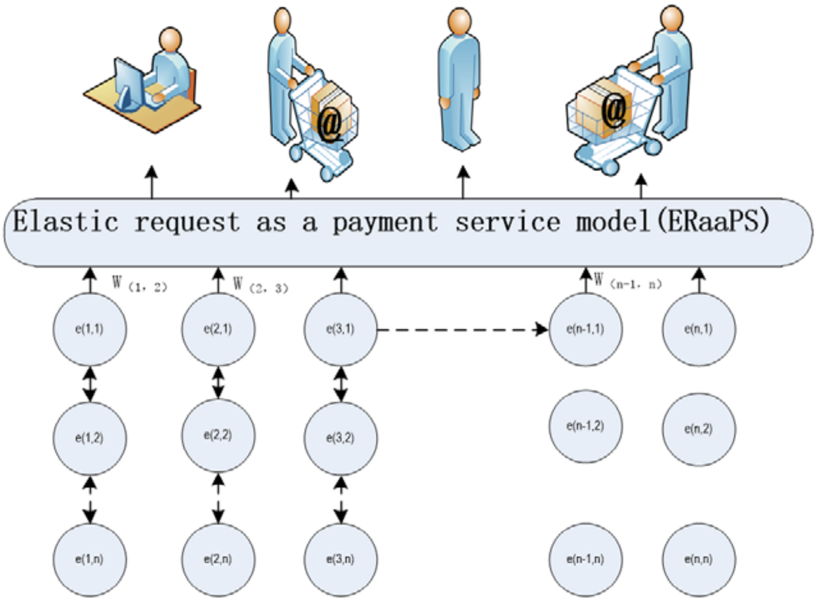

Figure 4: Sequential Service Topology after Row Arrival Model.

Central service model

As shown in Figure 4, the model is mainly for services without clear sequence requirements. Each node is represented by e_1, and the subscript i represents the node number under the structure. The model has two central nodes, such as e_1 distribution center node and e_2 collection center node. According to the current status of e_i ($i \ne 1$, $i \ne n$), the distribution center node will be assigned to different nodes for execution according to the task characteristics; the collection center node will provide the execution results of e_i ($i \ne 1$, $i \ne n$) to users after sorting. The advantages of the model: the distribution center node coordinates tasks, can handle multiple tasks in parallel, and meet the real-time requirements; the aggregation center node provides services, unified external interface, and provides more friendly services; each node has a clear division of labor, and can maximize the use of limited computing resources of edge devices. The following presents the delay total (minimum delay time) and energy consumption E_s (minimum energy consumption) model of the structure:

$$t_{total} = \underset{1\le i\le n}{MAX} \frac{L_i X_i}{f_i} \qquad (24)$$

Therein, $n \in Z^+, f_i = f_{CPU}^{\max}$

$$E_s = \underset{1\le i\le n}{MAX} \sum_{i=1}^{m} f_i \qquad (25)$$

Therein, $n \in Z^+, f_i = f_{CPU}^{\max}$

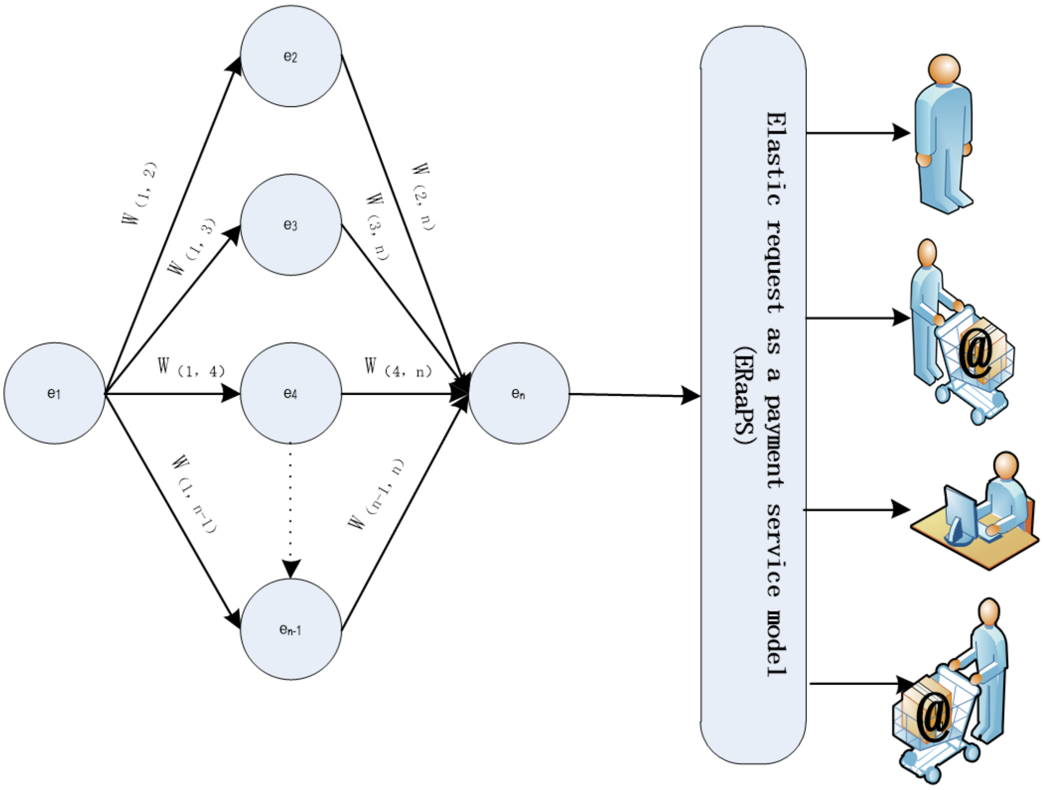

Figure 5: Service model of central topology.

DATA & RESULTS

E-commerce payment system LoadRunner load test target

LoadRunner is a kind of suitable for all kinds of architecture of the automatic load test tool, which implements real-time performance monitoring for the system under test at the same time, and automatically generate test results. Its purpose is to put the e-commerce payment in the simulation environment to test the performance of the system: predict the load of the system under pressure, analyze and evaluate the bottleneck of the system, and optimize the system performance before the release of the system. The test content includes: (1) average response time of access; (2) the maximum response time of access; (3) average number of access per second processing; (4) the success rate of access users; (5) the above record value under the condition of different concurrent users.

Load test steps for e-commerce payment system LoadRunner

Run the Contorller of LoadRunner, select the recorded script, set the number of virtual users and run it. When running, the number of virtual users set (the initial number of users is 100, 100 users are added each time) will be the number of concurrency, and the website will be accessed concurrently. It is preliminarily estimated that the maximum concurrency of the payment webpage will be 500 users. Then, all kinds of important index data of system performance were recorded, and the resource occupancy of the system was concerned. Then, test results were analyzed and experimental conclusions were drawn.

(1) Average Transaction Response Time

Table 1 Average Transaction Response Time

Color	Scale	Measurement	Min.	Ave.	Max.	SD
Red	1	Action_Transaction	238. 022	238. 084	238. 166	0. 03
Yellow	1	Actionl_Transaction	9. 019	9. 06	9. 123	0. 021
Blue	1	vuser_emi_Transaction	0. 0	0. 0	0. 0	0. 0
Green	1	vuser_init_Transaction	0. 0	0. 0	0. 0	0. 0

Results: table 1 shows the minimum average transaction response time: 9. 019 average 9. 06

Maximum: 9. 123 average transaction error: 0. 334.

Analysis: transaction response time <2s is in an excellent state.

2s ~ l0s good condition.

It fails when it's greater than l0s.

From the results, it can be analyzed that the average response time of things is greater than 2s and less than 10s in a good state.

Table 2 Payment client system resource calls

Color	Scale	Measurement	Min.	Ave	Max.	SD
Red	0. 01	%Disk Time (Physical Disk_Total) : localhost	0. 15	50. 367	4145. 203	295151
Yellow	0. 0l	File Data Operations/sec (System) : localhost	12	114. 145	6211. 403	475. 231
Blue	0. 0l	Page Faults/sec (Memory) : localhost	1028	1795 227	12324 25	1107. 642
Green	0. 1	Pages/se, e (Memory) : localhost	0. 0	7. 054	278. 008	36. 35
Purple	1E-6	Pool Nonpaged Bytes (Memory) : localhost	34750464	35035412. 757	35323904	162161. 285
Cyan	1E-7	Private Bytes (Process Total) : localhost	745607168	756772956. 252	794722304	4346259. 482
Black	10	Processor Queue Length (System) : localhost	0. 0	1. 072	8	1. 191
White	0. 1	Threads (Ohjects) : localhost	885	905. 833	928	11. 192

(2) CPU utilization Processor / %Processor Time

Results: according to table 2, the minimum CPU utilization value was 0. 260%, the average value was 2. 266%, the maximum value was 10. 156%, and the error was 0. 8660/.

Analysis: the average CPU utilization tested by this system is in an excellent state around 2. 26%.

(3) CPU time consumed by database operation: Processor / %User Time.

Results: as can be seen from table 2, the minimum CPU time consumed by database operation is 1. 367%, the maximum is 5. 280%, and the error is 0. 591%.

Analysis: more than 90% of the CPU consumed by database operation is in a relatively large state, so the optimization of the database system can be considered.

According to the results, the average CPU consumption of database operation is about 1. 376%, and no optimization is needed.

Results: as can be seen from table 2, the minimum, average, 1. 153%, maximum, 2. 344%, and error: 0. 8440/.

Analysis: the value of this parameter has been high, indicating that I/O is problematic. According to the results, the average CPU utilization is in a normal state.

(4) Remaining available Memory/Available Mbytes

Results: as can be seen from table 2, the minimum value of remaining available memory is 3064. 000, the average value is 3067. 678, the maximum value is 3070. 000, and the error is 1. 735.

Analysis: if the remaining available memory is less than 10%, the system memory may leak. As the physical memory of the machine is 4193Mb, the remaining available memory is 3067/4193> and 100% meets the standard.

(5) Download pages per second Memory/pages/ SEC

Results: according to table 2, the minimum number of pages downloaded per second is 0. 0, the average is 43. 071, the maximum is 351. 990, and the error is 125. 789.

Analysis: CPU per second 0 page exchange for good, 10 bad, more than 10 is very poor.

The results show that the number of pages downloaded per second is poor.

Conclusion: in the above design of mobile payment system, mobile terminals only need to carry out active communication twice, which not only reduces the communication burden of mobile terminals, but also accelerates the processing speed of mobile payment. Therefore, it is more suitable for mobile payment business.

CONCLUSIONS

In this paper, based on the traditional retail queuing linear model, the research scope is extended to e-commerce payment queuing, and the payment request queuing circumference model $M/M/n/m$ is constructed. The traditional retail payment queuing mode and cloud computing model of e-commerce payment service mode have been unable to adapt to the demand of elastic response, zero delay, zero error and low power consumption. The edge computing mode is mainly reflected in the limited resources of edge equipment, the real-time service and the intelligent decision-making. In this study, firstly, from the traditional queuing model to the circle model, then to the edge computing model, the task model, computing model and service model of elastic demand payment service are abstractly modeled. According to whether the tasks are sequential or not, this paper presents two common elastic demand service models (sequential service model and central service model) of payment service, and gives the corresponding evaluation formulas of delay and energy consumption. Then, it uses LoadRunner which is suitable for various architectures to simulate massive end-users' simultaneous request for payment service load test evaluation and detection its response time, transaction rate and system stability prevent the payment system from queuing congestion or even collapse. The simulation results show that the service model of elastic demand payment service can effectively solve the problems of elastic demand, uneven server load, delayed payment time, and good customer payment experience.

ACKNOWLEDGEMENT

This research was supported by the funding from the Institute of Digital Economy of Chongqing Technology and Business University and the School of Electrical and Electronic Engineering, Chongqing University of Technology.

REFERENCES

Baldinger, A. L., & Rubinson, J. R. (1996). Brand loyalty: The link between attitude and behavior. *Journal of Advertising Research, 36* (6), 22-34.

Fallah Nezhad, M. S., & Akhavan Niaki, S. T. (2011). A multi-stage two-machines replacement strategy using mixture models, bayesian inference, and stochastic dynamic programming. *Communications in Statistics-Theory and Methods, 40* (4), 702-725.

Fan, Z., Xu, J., & Wan, X. (2013). Research of mobile payments business model in china based on the evolutionary game theory. *Information Technology Journal, 12* (20), 5466-5471.

Floyd, S., & Jacobson, V. (1993). Random early detection gateways for congestion avoidance. *IEEE/ACM Transactions on networking, 1* (4), 397-413.

Forouzan, B. A. (2007). *Cryptography & network security*: McGraw-Hill, Inc.

Houston, M. B., Bettencourt, L. A., & Wenger, S. (1998). The relationship between waiting in a service queue and evaluations of service quality: A field theory perspective. *Psychology & Marketing, 15* (8), 735-753.

Jones, P., & Peppiatt, E. (1996). Managing perceptions of waiting times in service queues. *International Journal of Service Industry Management, 7* (5), 47-61.

Land, K. C., McCall, P. L., & Nagin, D. S. (1996). A comparison of Poisson, negative binomial, and semiparametric mixed Poisson regression models: With empirical applications to criminal careers data. *Sociological Methods & Research, 24* (4), 387-442.

Mills, D. C., Wang, K., Malone, B., Ravi, A., Marquardt, J., Badev, A. I., Brezinski, T., Fahy, L., Liao, K., & Kargenian, V. (2016). *Distributed ledger technology in payments, clearing, and settlement.* (2016-095). Finance and Economics Discussion Series, Washington: Board of Governors of the Federal Reserve System.

Xie, X. (2008). *Computer Network* (5th ed.). Beijing: Publishing House of Electronics Industry.

Zhang, Y., Ma, J., Wang, Y., & Chen, X. (2011). Improved nonlinear random early detection algorithm. *Journal of Computer Applications, 4* (31), 890-917.