Proceedings of the IASTED International Conference
Parallel and Distributed Computing and Networks (PDCN 2013)
February 11 - 13, 2013 Innsbruck, Austria

REGULATE MY APPETITE ACCORDING TO DISHES: A DISTRIBUTED ON-LINE OPTIMAL STOPPING WEB SERVICE SELECTION SCHEME

Siping Liu*[12], Yaoxue Zhang†[12], Yuezhi Zhou‡[12], Hao Liu§[12], Di Zhang¶[12], and Wei Hu‖[12]

[1]Tsinghua National Laboratory for Information Science and Technology, Tsinghua University, Beijing 100084, China
[2]Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China

ABSTRACT
This paper demonstrates a distributed on-line service selection(probe/access) scheme: optimal stopping web service selection scheme based on the rate of return problem from optimal stopping theory. There are three differences between our scheme and the conventional schemes. Firstly, it does not need to probe all web services, and only probe a few web services. Secondly, our scheme focuses on maximizing the average QoS(Quality of Service) return per unit of cost over all stages of probe and access for a long period rather than maximizing QoS return per single stage of probe and access in usual schemes. Thirdly, our scheme develops a return function based on three factors: QoS return, user's requirement and probe cost which are seldom considered simultaneously before. Through theory analysis and computation, we demonstrate that compared with the conventional schemes our scheme has additional advantages while achieving same good performances.

KEY WORDS
Web Service Selection, Quality of Service, Probe Cost, User Requirements, Optimal Stopping Theory.

1 Introduction

With the advent of new technologies such as grid computing, cloud computing, internet of things, or even smart hyperspace [1], more and more web services appear in internet to let users access at anytime from anywhere. Many software tools [2] or platforms [3] are developed to support web services exploration, binding and construction. One key issue in the Web Service area is to discover the most relevant services meeting the functional requirements of users. Equally important, users also would like to discover services that best meet their requirements in terms of QoS that reflects user experiences, i.e., performance, throughput, reliability, availability, trust, etc. Thus QoS-based web service selection and ranking mechanisms will play an essential role in service-oriented architectures, especially when the functional requirements matchmaking process returns many services with same functionalities.

In this paper, we present a service selection scheme based on three factors: QoS, user's requirement and probe cost. Our work is based on requirements from a real-world paradigm on cloud computing. This paradigm is named as transparent computing and demonstrated in [4] working in an internet environment. Some IaaS(Infrastructure as a Service) servers where users' software and data, including OSes and applications are stored upon them and some users' clients are connected together through internet. A user can choose OSes and applications upon them on demand through these clients, similar to choose different TV channels in daily life. A selected OS and its applications run on clients, not on the servers. For the purpose of load balance, the IaaS provider delivers many web services on different positions in internet. Clients can select one web service and then fetch software and data on computing demand. Many factors including network condition such as congestion between the web service and client or the web services' load and process ability influence their transmission rate of web services. At the same time, for a single web service, its transmission rate always changes with time. If clients tend to select best web service, one probe cost should be paid for every web service's transmission rate probe. When a web service is selected, data will be transmitted to the client for a period. Considering users' behavior feature, this period will not be infinity. Therefore, a problem arises that how to make a compromise between these three factors: transmit rate(i.e. QoS), user's requirement(i.e. transmit time) and probe cost.

We develop the optimal stopping selection scheme with a purpose of maximizing the long-term average throughput per unit of time rather than maximizing the throughput per stage of probe and transmit. Since that users' two-phase event comprising of probing some web services and then selecting one to transmit data is repeated in an interval of time, we transform the problem above to a rate of return problem in optimal stopping theory [5](The rate of return problem will be introduced in section 3 below).

The major contribution of our work is an optimal stopping web service selection scheme. Our scheme comprises three parts as below:

*email: liusp11@mails.tsinghua.edu.cn
†email: zyx@moe.edu.cn
‡email: zhouyz@mail.tsinghua.edu.cn
§email: liuhao09@mails.tsinghua.edu.cn
¶email: di-zhang10@mails.tsinghua.edu.cn
‖email: huwei09@mails.tsinghua.edu.cn

DOI: 10.2316/P.2013.795-042

1. Return function. It involves three factors: service QoS, probing cost and user requirement;

2. Optimal stopping threshold. According to services' QoS probability distribution, we analyze and compute the optimal stopping threshold beforehand;

3. Optimal probe/access algorithm. This algorithm makes a decision of accepting or rejecting a web service based on comparing result between this web service's QoS and the optimal stopping threshold computed above. Applied in every stage of probe/access repeated, user will achieve a long-term maximum average return of QoS per unit of cost.

Numerical and simulation results have shown that the newly proposed optimal stopping web service selection scheme yields very good performances under various web service QoS probability distributions. This is mainly due to a fact according the principle of optimality [5]. If the QoS probability distributions is known in advance, we can sequentially in a random order probe a web service, make a decision whether accept or reject this web service, we can always stop and access a web service that its QoS is better than follow-up web services' QoS expectation. Therefore, the long-term average QoS return expectation is optimal if this stage is repeated again and again. This is particularly important since our scheme can be implemented as a distributed on-line decision making algorithm. Firstly, our scheme consists of a simple process, it can sequentially in a random order probe a web service, then make a decision whether accept and access this web service or reject and go on to probe next web service. Secondly, this scheme does not need to probe all web services, and only probe a few web services while achieving a maximum average QoS per unit of cost.

The rest of the paper is organized as follows. First, we briefly mention the related work in Section 2. Section 3 introduces optimal stopping theory. The optimal stopping web service selection scheme is described in detail in Section 4. Section 5 analyzes and computes the optimal stopping threshold. We discuss various numerical and simulations results in section 6 and conclude the paper in section 7.

2 Related Work

QoS is a yardstick for measuring web services' various nonfunctional characteristics such as response time, throughput, availability, and reliability. Therefore, there are many QoS-based web service selection mechanisms focusing on developing the QoS return function based on all kinds of QoSes that best reflect users' experience and services' true states. For example, [6] [7] define web service selection function and forecasting model based on QoS measures.

The service selection has the cost performance problem since service probes impose costs for the service users and consume resources of the service providers. The constraints define acceptable values of QoS cost ratio, typically upper (or lower) bound. It is necessary, and has been investigated [8] [9] [10] [11] [12], to explore near-optimal solutions based on some kinds of heuristics which probe each service.

From the space dimension who executes information retrieving and selection decision, schemes can be classified to centralized or distributed. Centralized schemes provide a specialized server in charge of retrieving and making selection decision for end users. In distributed schemes, end users take charge of these things. From the dimension when to retrieve information and make decision, schemes can be classified to on-line or off-line. On-line means making selection decision based on real-time QoS information when user requests to access a service. Off-line means making service selection decision based on historic information. [13] is a centralized off-line scheme. [14] [15] define distributed schemes.

In some paradigms, probe cost is very critical. [16] makes a debut to apply optimal stopping theory in service selection field for cutting down the number of times to probe. It uses a fundamental optimal stopping algorithm: classical secretary problem. There is one secretarial position available. The applicants are interviewed sequentially in a random order. As each applicant is being interviewed, you must either accept the applicant for the position and stop the decision problem, or reject the applicant and interview the next one if any. The decision to accept or reject an applicant must be based only on the relative ranks of the applicants interviewed so far. An applicant once rejected cannot later be recalled. Your objective is to select the best of the applicants; that is, you win one if you select the best, and zero otherwise. In [16], an analogy between applicants and services is made, achieves the objective of choosing the best web service at lowest probe cost.

Nevertheless, in some paradigms, users prefer not to select the best web service especially if the probe cost is so large with respect to users' requirement. Therefore, we think it is essential to consider the three factors: QoS, user's requirement and probe cost concurrently. Moreover, at the same time, this algorithm should be a lightweight distributed on-line service selection algorithm. [16] hints us that optimal stopping theory is a good method to solve this problem. Since classical secretary problem does not consider both users' requirement and accumulated cost in service selection process, we derived our optimal stopping web selection scheme on the rate of return in optimal stopping theory. To the best of our knowledge, our scheme is the first one to solve the service selection in this paradigm.

3 A Preliminary On Optimal Stopping Theory

Optimal stopping theory is a mathematical method focusing on choosing a right time to stop, to maximize the ex-

pected return in a stochastic and sequential process. Stopping rule problems are defined by two objects:

- A sequence of random values X1,X2, . . ., whose joint distribution is assumed to be known in priory;
- A sequence of real-valued return functions, y_0,$y_1(x_1)$, $y_2(x_1, x_2)$, . . ., $y_\infty(x_1, x_2, ...)$.

The sequence of X_1, X_2, ... can be observed as long as possible. For each n=1, 2, ..., after observing X_1=x_1, X_2=x_2, ..., X_n=x_n, the decision is either to stop and receive the known return $y_n(x_1, ..., x_n)$, or to continue and observe X_n+1 for further decision. If the decision is not to take any observation, the received return is a constant value $y_0 0$. If the observation never stops, the received return is $y_\infty(x_1, x_2, \cdots)$. The theory of optimal stopping concentrates on determining the stopping rule N to maximize the expected return E[Y_N].

In stopping rule problems that are repeated in an interval of time, it is often appropriate to maximize the average return per unit of time. This leads to the problem of choosing a stopping rule N to maximize the ratio EY_N/EN. The reason we wish to maximize this ratio rather than the true expected average per stage, $E(Y_N/N)$, is that if the problem is repeated independently n times with a fixed stopping rule leading to i.i.d. stopping times $N_1, ..., N_n$ and i.i.d. returns $Y_{N_1}, \cdots, Y_{N_n}$, the total return is $Y_{N_1}+\cdots+Y_{N_n}$ and the total time is $N_1+\cdots+N_n$, so that the average return per unit time is the ratio $(Y_{N_1}+\cdots+Y_{N_n})/(N_1+\cdots+N_n)$. If both sides of this ratio are divided by n and if the corresponding expectations exist, then this ratio converges to EY_N/E_N by the law of large numbers. We call this ratio the rate of return. We wish to maximize the rate of return.

4 Optimal Stopping Web Service Selection Scheme

4.1 Model

For the purpose of state simplicity, we will describe and analyze our scheme in transparent computing paradigm from here on. QoS of web services is represented by its transmission rate, which is replied by its provider while users probed it. User requirement is transmission time over the service which user transmits data for a period if this service is accepted. Probe cost is represented as probe time for every service from the time probe packet is sent to the time user receives a reply packet. Table.1 shows all notations which are used in our algorithm and their meaning respectively.

In internet, because of the network environment's complexity, for example: many routes exist from end user to probed service, service load difference, so the probe time of services(τ_p) is different. In our service scheme model, we classified the services according different levels of probe time. Just like shown in Figure.1, the services are classified by groups. In our model, we fix the minimal probe time level is $\tau_p = 10ms$, max probe time level is 960ms, and probe time level's increment is 50ms. So all of services are divided to 20 groups. We presume that in every group service transmission rate(R_i) joint probability distribution is same. According to our algorithm of select a web service in some group, the difference is only that probe time level is different. From our instinct, we maybe prefer to select a web service applying our scheme in service group with minimum probe time level. From simulation result, we will verify this instinct is right.

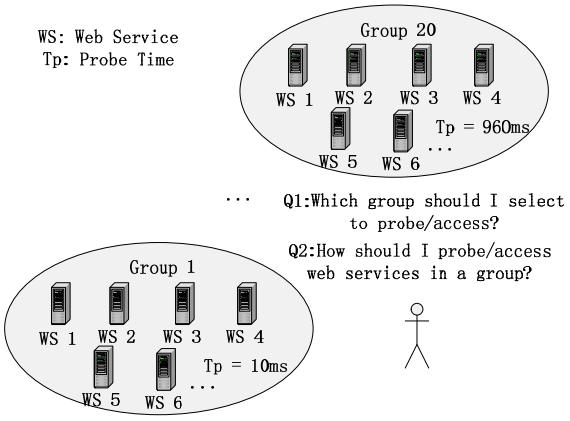

Figure 1: Web Service Exploration Model

4.2 Return Function and Optimal Stopping Selection Algorithm

Every stage consists of a few probe and one transmission. As we said above, our scheme maximizes the average throughput per unit of time over all stages rather than over a single stage. Probe time, τ_p , from the time probe packet is sent to the time user receive a replay packet. It represents probe cost. τ_d is transmission time over which user transmits after some service is accepted. It means users' requirement. R_i transmission rate that can be provided by the service in the position of service exploration sequence i. Service transmission rate: $\{R_1, R_2, ...\}$ satisfies joint probability distribution: π_k. π_k is known in advance. If we accept R_i service, we assume in period of τ_d, user always have data to transmit. Apparently, time consumed in one stage is the sum of a few τ_p and one τ_d.

In the probe sequence position i, if the service with transmission rate R_i is accepted, the observed sequence is :$\{R_1, R_2, \cdots, R_i\}$, the throughput of this stage can be calculated by Formula1 shown below:

$$\Phi(i) = \frac{B(i)}{T(i)} = \frac{\tau_d R_i}{i\tau_p + \tau_d} \quad (1)$$

$B(i)$ and $T(i)$ represent throughput and time cost in this stage stopped at position of i.

Apparently, i can be any value from set: $\Psi = \{i : i \in [0, \infty)\}$. $\Phi(i)$ is a r.v.. If stopped at position i, then

Notation	unit	Description
i		service exploration sequence number
R_i	Mbit/s	transmission rate which can be provided by the service in position of service exploration sequence i
τ_p	s	probe time, from the time probe packet is sent to the time user receive a replay packet. It represents probe cost.
τ_d	s	transmission time over which user transmit after some service is accepted. It means users' requirement.
$B(i)$	M	In the position of probing sequence i, if this service is accepted, data volume that users can transmit over this service in τ_d time.
$T(i)$	s	From the beginning to the position of probing sequence i on which the service is accepted and transmission is finished, the summary time that this exploration process cost.
$\Phi(i)$	Mbit/s	In the position of probing sequence i, if this service is accepted, throughput that the user can achieve.
$E[\Phi(i)]$	Mbit/s	throughput expectation if the exploration process is repeated in an interval of time. It is average return per unit of time.
q_k		In the case of discrete transmission rate, the probability of rate k in some service group.
π		transmission rate joint distribution in a web service group.

Table 1: Summary of Notations

the user can achieve the throughput:$\Phi(i)$. Since Our object is to maximize the throughput expectation while the probe/access stage is repeated over a long period of time, we define return function for this purpose that stops at some position i^* at every stage. Expectation of $\Phi(i)$ is maximized, e.g. maximized throughput expectation shown below:

$$\max_{i\in\Psi} E\left[\Phi\left(i\right)\right] = \max_{i\in\Psi} E\left[\frac{B(i)}{T(i)}\right] \tag{2}$$

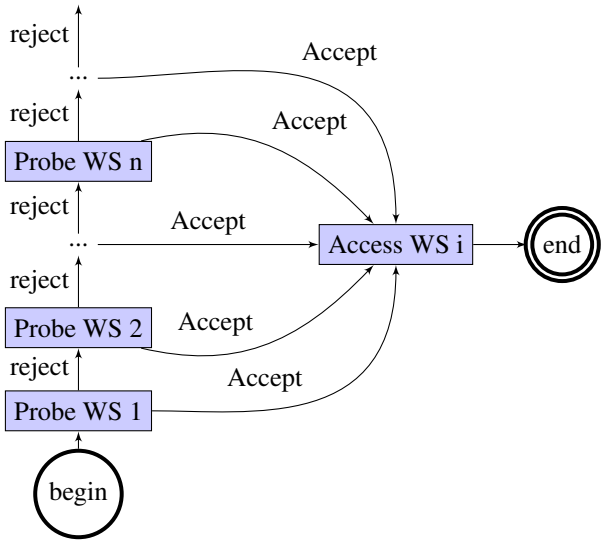

Figure 2: Optimal Stopping Probe/Access Algorithm

In section 4, we will derive the optimal stopping threshold that is a transmission rate. In every stage, firstly, user selects one group to proceed to execute the following probes and transmission. User sequentially probes the service that is selected randomly from services that has not been probed before in this stage. According to probe result, user makes a decision on line. If the transmission rate the service can provide satisfies the optimal stopping threshold, then probe is stopped and transmit over this service. If service transmission rate cannot satisfy the threshold, the service is rejected and probe goes on. A service once rejected cannot later be recalled. This process is shown in Figure.2.

5 Analysis

Since service transmission rate joint distribution in the same group is independent of every stage of probe and access, so one stage is independent of other stages. So maximizing $E\left[B(\psi)/T(\psi)\right]$ is the same with maximizing $E\left[B(\psi)\right]/E\left[T(\psi)\right]$ [5]. So in every stage, our object is to stop in the optimal position of $\psi^* \in \Psi$, e.g.:

$$\psi^* = \arg\max_{\psi\in\Psi} \frac{E\left[B(\psi)\right]}{E\left[T(\psi)\right]} \tag{3}$$

Lemma 1 *If* $\sup_{\psi\in\Psi} E\left[B(\psi) - \lambda T(\psi)\right] = 0$, *then* $\sup_{\psi\in\Psi} E\left[B(\psi)\right]/E\left[T(\psi)\right] = \lambda$. *Moreover, if* $\sup_{\psi\in\Psi} E\left[B(\psi) - \lambda T(\psi)\right] = 0$ *is attained at* $\psi^* \in \Psi$, *then* ψ^* *is optimal for maximizing* $\sup_{\psi\in\Psi} E\left[B(\psi)\right]/E\left[T(\psi)\right]$.

Strict proof of Lemma 1 can be found in [5]. Based on this lemma, we can transform Equation (1) to:

$$\begin{aligned} V =& B(i) - \lambda T(i) \\ =& \tau_d R_i - \lambda(i\tau_p + \tau_d) \end{aligned} \tag{4}$$

The objective is to find appropriate value of λ to maximize expectation of Equation (4),e.g.:

$$V^* = \max_{i\in\Psi} E\left[B(i) - \lambda T(i)\right] \tag{5}$$

According to Lemma.1 in Equation (5), if λ^* is found to make $V^* = 0$, and the corresponding i^* makes Equation (2) achieves optimal value, it is also the maximum throughput:λ^*. Based on this, we can verify the existence of optimal value i^* of Equation (2) and corresponding computation method.

Theorem 1 *The optimal value i^* Equation (2) exists. Based on this optimal value, the maximum throughput λ^* is the solution of:*
$E\left[\max(\tau_d R_i - \lambda(i\tau_p + \tau_d), 0)\right] = \lambda^* \tau_p$. *The optimal stopping position:* $i^* = \min\{i \geq 1 : R_i \geq \lambda^*\}$

Proof: 1). Existence.

We need to prove that for any λ, an optimal stopping rule exists for the transformed Equation (4). It follows from Theorem 1 in Chapter 3 of [5], that the optimal stopping rule exists if the following two conditions are satisfied:

1. $E\left[\sup_i V\right] < \infty$
2. $\lim_{i\to\infty} \sup_i V = -\infty$

By examining Equation (4), condition 2 is clearly satisfied. Condition 1 can be verified by applying Theorem 1 in Chapter 4 of [5], Specifically, it is easy to see that the random variable

$$V = X_i - \lambda\tau_p \tag{6}$$

among which:

$$X_i = \tau_d R_i - \lambda\tau_d \tag{7}$$

So now we only need to observe X_i, then we can verify V satisfies condition 1. Firstly, since R_i is i.i.d., other parameters in X_i are all constants, so X_i is i.i.d., so the value of X_i must also be finite. e.g.:$E[\max(X_i, 0)] < \infty$, and $E[\max(X_i, 0)^2] < \infty$. According to Theorem 1 in Chapter 4 of [5], $E\left[\sup_i V\right] = E\left[\sup_i (X_i - i\lambda\tau_p)\right] < \infty$.So condition 1 is satisfied.

2). Optimal solution.

From Equation (6) and Equation (7), we can view X_i as return of every stage of probe. And then $\lambda\tau_p$ represent probe cost per time unit in this stage. So Equation(4) is a time invariant problem. According to principle of optimality in Chapter 2 of [5], for time invariant problem, its optimal rule can be achieved through this way below:

$$i^* = \min\{i > 1 : X_i \geq V^*\} \tag{8}$$

where V^* denotes the expected return from an optimal stopping rule. It satisfies the following optimality equation:

$$V^* = E[\max(X_i, V*)] - \lambda\tau_p \tag{9}$$

Equivalently, the above equation can be written as:

$$E[\max(X_i - V*, 0)] = \lambda\tau_p \tag{10}$$

Recalling the connection between the original problem and its transformed version, the value of λ^* that gives $V^* = 0$, is simply the solution of Equation (10). With $V^* = 0$, we have the following equation:

$$E[\max(\tau_d R_i - \lambda^*\tau_d, 0)] = \lambda^*\tau_p \tag{11}$$

According Equation (11), we can get value of λ^*, and so, this value is the maximum throughput of Equation (2).

Then, make $V^* = 0$ in Equation (8), we can achieve the optimal stopping position of every stage:

$$i^* = \min\{i > 1 : R_i \geq \lambda^*\} \tag{12}$$

Theorem 2 *Optimal stopping position of Equation (2) is unique.*

Proof: We only need to verify that λ^* value of Equation (11) is unique. The main idea is to show that the LHS of Equation (11) is a mono-decreasing function in λ^*, whereas the RHS is mono-increasing in λ^*. The two functions must intersect at one and only one point.

$$\begin{aligned} L(\lambda^*) =& E[\max(\tau_d R_i - \lambda^*\tau_d, 0)] \\ =& \int_{\lambda^*}^{\infty} \tau_s R_i q(R_i) dR_i - \int_{\lambda^*}^{\infty} \lambda^*\tau_d q(R_i) dR_i \end{aligned} \tag{13}$$

$$\begin{aligned} \frac{dL(\lambda^*)}{d(\lambda^*)} =& \lambda^*\tau_d q(\lambda^*) - [\tau_d \int_{\lambda^*}^{\infty} q(R_i) dR_i - \lambda^*\tau_d q(\lambda^*)] \\ =& -2\lambda^*\tau_d q(\lambda^*) - \tau_d \int_{\lambda^*}^{\infty} q(R_i) dR_i \end{aligned} \tag{14}$$

Apparently, Equation (14) always is negative, and LHS of Equation (13) is a mono-decreasing function in λ^*. Meanwhile, $L(0) = \tau_d E[R_i] < \infty$, $L(\infty) = 0$. RHS of Equation (11) is mono-increasing in λ^*. When λ^* is set 0 and ∞ respectively, the value is 0 and ∞ respectively. so λ^* value of Equation (11) is unique. So, Optimal stopping position: i^* of Equation (2) is unique.

Lemma 2 *For discrete transmission rate case, where $R_i \in R_0, R_1, ..., R_K$, and $R_0 = 0 < R_1 < ... < R_K$, the maximum throughput λ^* for optimal stopping position i^* of Equation (2) can be achieved from this way: define $1 \leq k^* \leq K$, and $R_(k^* - 1) < \lambda^* \leq R_{k^*}^*$, satisfies:*

$$\lambda^* = \frac{\tau_d \sum_{k=k^*}^{K} R_k q_k}{\tau_p + \tau_d \sum_{k=k^*}^{K} q_k} \tag{15}$$

Proof: According the condition of Theorem.2, we can transform Equation (13) to:

$$\sum_{k=k^*}^{K} (\tau_d R_k - \lambda^*\tau_d) q_k = \lambda^*\tau_p \tag{16}$$

So Equation (15) is verified.

Lemma 3 *Transmission delay $D(R_i*)$ is defined as the time period from the beginning of a exploration and transmission round to the optimal stopping position i^* that service is accepted. At position i^*, the corresponding transmission rate is :k^*, so transmission delay expectation is: $E(D(R_i*)) = \frac{\tau_p}{1 - \sum_{k=0}^{K^*-1} q_k}$. Probe numbers is probe numbers expectation, can be achieved by: $\frac{1}{1 - \sum_{k=0}^{K^*-1} q_k}$*

Proof:

$$\begin{aligned} E(D(R_i*)) =& \tau_p \sum_{\ell=1}^{\infty} \{\ell[P(\ell < i^*)]^{(\ell-1)} P(\ell \geq i^*)\} \\ =& \tau_p \frac{1}{1 - P(\ell < i^*)} \\ =& \frac{\tau_p}{1 - \sum_{k=0}^{K^*-1} q_k} \end{aligned} \tag{17}$$

Probe numbers expectation is $\frac{E(D(R_i*))}{\tau_p} = \frac{1}{1-\sum_{k=0}^{K*-1} q_k}$. So Lemma 3 is verified.

6 Numerical And Simulation Results

In this section, we use simulations to evaluate the performance of our proposed optimal service probe/select scheme. Our simulations are developed using MATLAB-based Discrete-Transmission-Rate Web-Services-Selection simulation system.

In our simulation system, we assume all services only have 5 discrete transmission rates, that is $R = [0, 1, 2, 3, 4, 5]$, unit is Mbit/s. And we assume transmission time $\tau_d = 2$, unit is s. And probe time is discrete numbers, $\tau_p = 10 + 50 \cdot L, L \in [1, ..., 19]$, unit is also ms.

As we mentioned above, transmission rates probability distribution π of all these services is a known condition. We defined two kinds of π. $\pi_1 = q_k \in [0.05, 0.05, 0.1, 0.1, 0.35, 0.35]$,and $\pi_2 = q_k \in [0.25, 0.25, 0.15, 0.15, 0.1, 0.1]$. We can see, the probability distribution in π_1 tend to the right in the X coordinate axis of probability distribution function. And it is vice versa for π_2. The difference between π_1 and π_2 represents that in π_1 there are more numbers of services that have better transmission rates than in π_2. We want to evaluate our scheme's performance in both conditions of the contrary probability distributions. We make a convention that we refer to π_1 or π_2 conditions below, we mean other conditions mentioned in last paragraph are all same except the difference between their probability distribution mentioned here.

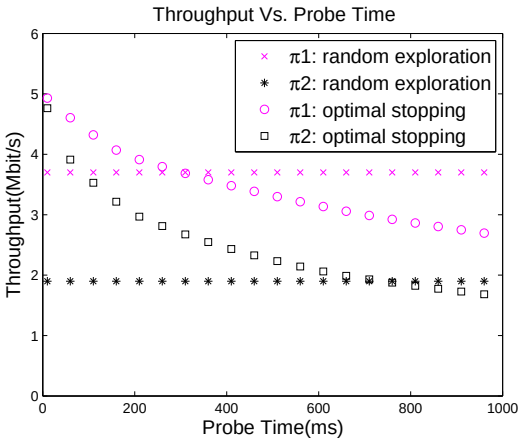

Figure 3: Throughput Vs. Probe Time

For the purpose of showing our scheme's performances over other service probe/access algorithms, we compare our results with ergodic and random probe/access schemes. Ergodic scheme means user will send probe packets concurrently to all services, then access the best service to transmit. This scheme's advantage is that it always transmit over the best service. Its shortage lies in that it has to send many probe packets. Random scheme does

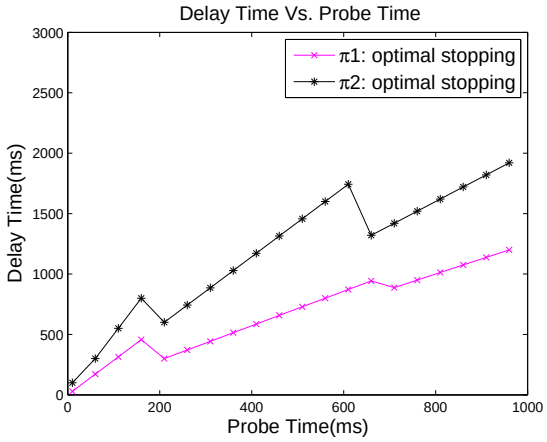

Figure 4: Probe Delay Vs. Probe Time

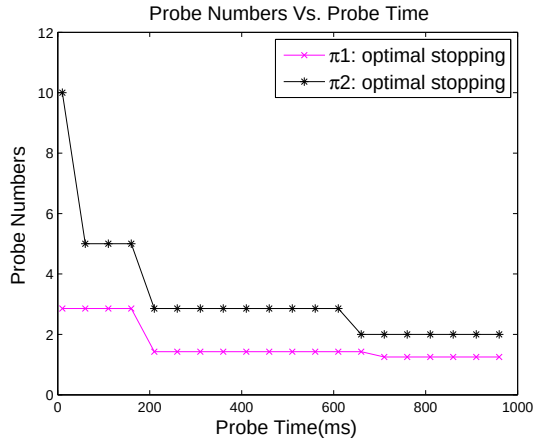

Figure 5: Probe Numbers Vs. Probe Time

not send any probe packets and it just randomly select a service to transmit. The shortage is it only achieves the average throughput expectation and the advantage is there is no probe cost at all.

Apparently, according to our simulation condition: π_1 or π_2, ergodic scheme always achieves the best transmission rate: 5Mbit/s. It has no business with the services' transmission rate probability distribution. If we use minimum probe time $\tau_p = 10ms$, ergodic scheme achieves throughput: 4.98Mbit/s. Random scheme need no probe cost, so its throughput always is same with transmission rate expectation. In condition of π_1, the throughput expectation is 3.7Mbit/s. And in condition of π_2, throughput expectation is 1.9Mbit/s. It shows clearly that random scheme's throughput depends on the internet transmission rate probability distribution.

After our optimal stopping scheme is applied in the conditions of π_1 and π_2 respectively, the simulation results are shown in Table.2. We observed in both conditions of π_1 and π_2 that as probe time τ_p increases, the maximum throughput expectation, probe delay and probe numbers change accordingly.

From Figure.3, we observed that in both condition-

Group Index	1	2	3	4	5	6	7	8	9	10
Probe Time(ms)	10	60	110	160	210	260	310	360	410	460
Throughput in π_1(Mbit/s)	4.9296	4.6053	4.3210	4.0698	3.9130	3.7952	3.6842	3.5795	3.4807	3.3871
Throughput in π_2(Mbit/s)	4.7619	3.9130	3.5294	3.2143	2.9670	2.8125	2.6733	2.5472	2.4324	2.3276
Delay Time in π_1(ms)	28.6	171.4	314.3	457.1	300.0	371.4	442.9	514.3	585.7	657.1
Delay Time in π_2(ms)	100.0	300.0	550.0	800.0	600.0	742.9	885.7	1028.6	1171.4	1314.3
Probe Numbers in π_1	2.8571	2.8571	2.8571	2.8571	1.4286	1.4286	1.4286	1.4286	1.4286	1.4286
Probe Numbers in π_2	10.0000	5.0000	5.0000	5.0000	2.8571	2.8571	2.8571	2.8571	2.8571	2.8571
Group Index	11	12	13	14	15	16	17	18	19	20
Probe Time(ms)	510	560	610	660	710	760	810	860	910	960
Throughput in π_1(Mbit/s)	3.2984	3.2143	3.1343	3.0583	2.9870	2.9237	2.8631	2.8049	2.7490	2.6953
Throughput in π_2(Mbit/s)	2.2314	2.1429	2.0611	1.9880	1.9298	1.8750	1.8232	1.7742	1.7277	1.6837
Delay Time in π_1(ms)	728.6	800.0	871.4	942.9	887.5	950.0	1012.5	1075.0	1137.5	1200.0
Delay Time in π_2(ms)	1457.1	1600.0	1742.9	1320.0	1420.0	1520.0	1620.0	1720.0	1820.0	1920.0
Probe Numbers in π_1	1.4286	1.4286	1.4286	1.4286	1.2500	1.2500	1.2500	1.2500	1.2500	1.2500
Probe Numbers in π_2	2.8571	2.8571	2.8571	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000

Table 2: Expectation Of Throughput, Probe Delay And Probe Numbers Vs. Probe Time

s of π_1 and π_2, as probe time τ_p increases, the throughput expectation decreases. And in some position of τ_p, the throughput expectation will be worse than the random scheme. When the τ_p is minimum, in condition of π_1, the throughput expectation will be 4.93Mbit/s. This throughput expectation is 133% of random scheme's 3.7Mbit/s. Comparing with ergodic scheme, it is 99% of ergodic scheme's throughput expectation. So our scheme's advantage over ergodic scheme is that it needs only a few probe numbers to access an appropriate service guaranteeing the long-term maximum average throughput. The value of probe numbers is only 2.86 if τ_p is minimum(10ms) in condition of π_1. In condition of π_2, the throughput expectation will be 4.76Mbit/s. It is 255% of random scheme's 1.9Mbit/s. So we can conclude in condition of π_2 our algorithm has much more advantages. In condition of π_1 and π_2, if probe times are 300ms and 660ms respectively, our scheme will achieve worse throughput expectation than random algorithm.

From Figure.4, we observed that in both condition of π_1 and π_2, the delay time increases as probe time τ_p increases(but it does not mono-increase). So our scheme is suitable for non-realtime applications.

From Figure.5, we observed in both condition of π_1 and π_2, probe numbers decreases as probe time τ_p increases. This is because as probe time τ_p increases, the probe cost will increase, and probe cost per unit increases accordingly and this will restrain our scheme probing many more services, and results probe numbers decreases. And then at the same time, the optimal stopping threshold decreases, this is the reason why the delay time does not mono-increase in Figure.4.

7 Conclusion

From the theory analysis and computation simulation results above, we can draws some conclusions as below:

1. For every return function of our scheme, there is always a solution. This solution is a single numeric value that can be the stopping threshold applied in our optimal probe/access scheme. If some service satisfies this threshold value, we stop probing and transmit over this service. Applying this scheme, we always gain optimal long-term average return per unit of time.

2. The number of probed services before accepting a service is only related to the factors of return function including the internet services' QoS probability distribution. It has no relation with internet service numbers. Of course, the sum of internet services should let our probe/access scheme observe services as long as possible.

3. Compared with random scheme, our algorithm can achieve a drastically lifted throughput especially in the condition of π_2 that many internet services have bad transmission rate. When users' requirement(represented by transmission time every probe/access process in this paper) is fixed, the long-term average throughput achieved by our algorithm will decrease as the increase of probe time τ_p. In some position of τ_p, the throughput will be worse than the random scheme.

4. Our scheme brings about delay time, so it is suitable for non-realtime services. According simulation re-

sults, the throughput will increase as probe time τ_p increases.

5. Compared with that the ergodic scheme need probe all services, our algorithm just probes a few probe services. But our algorithm can gain approximate performance compared with the ergodic scheme.

Compared with the conventional schemes, our scheme has additional advantages while achieving same good performances. So our scheme is suitable to be implemented as a distributed on-line web service selection algorithm in real-world cases whose purpose is maximizing the long-term average QoS return per unit of cost such as transparent computing paradigm.

Acknowledgments

The authors appreciate the reviewers for their extensive and informative comments for the improvement of this manuscript. The work is partially supported by the National Key Technology R&D Program (Grant No.2012BAH13F04), and the research fund of Tsinghua ¨CTencent Joint Laboratory for Internet Innovation Technology.

References

[1] J. Ma, L. Yang, B. Apduhan, R. Huang, L. Barolli, and M. Takizawa, "Towards a smart world and ubiquitous intelligence: a walkthrough from smart things to smart hyperspaces and ubickids," *International Journal of Pervasive Computing and Communications*, vol. 1, no. 1, pp. 53–68, 2005.

[2] T. Fahringer, A. Jugravu, S. Pllana, R. Prodan, C. Seragiotto, and H. Truong, "Askalon: a tool set for cluster and grid computing," *Concurrency and Computation: Practice and Experience*, vol. 17, no. 2-4, pp. 143–169, 2005.

[3] P. Cappello, B. Christiansen, M. Ionescu, M. Neary, K. Schauser, and D. Wu, "Javelin: Internet-based parallel computing using java," in *ACM Workshop on Java for Science and Engineering Computation*. Citeseer, 1997.

[4] Y. Zhang and Y. Zhou, "Transparent computing: a new paradigm for pervasive computing," *Ubiquitous Intelligence and Computing*, pp. 1–11, 2006.

[5] T. Ferguson, "Optimal stopping and applications." [Online]. Available: http://www.math.ucla.edu/~tom/Stopping/

[6] S. Galizia, A. Gugliotta, and J. Domingue, "A trust based methodology for web service selection," in *Semantic Computing, 2007. ICSC 2007. International Conference on*. IEEE, 2007, pp. 193–200.

[7] A. Amin, A. Colman, and L. Grunske, "An approach to forecasting qos attributes of web services based on arima and garch models," in *Web Services (ICWS), 2012 IEEE 19th International Conference on*. IEEE, 2012, pp. 74–81.

[8] T. Yu, Y. Zhang, and K. Lin, "Efficient algorithms for web services selection with end-to-end qos constraints," *ACM Transactions on the Web (TWEB)*, vol. 1, no. 1, p. 6, 2007.

[9] M. Alrifai and T. Risse, "Combining global optimization with local selection for efficient qos-aware service composition," in *Proceedings of the 18th international conference on World wide web*. ACM, 2009, pp. 881–890.

[10] F. Lecue and N. Mehandjiev, "Towards scalability of quality driven semantic web service composition," in *Web Services, 2009. ICWS 2009. IEEE International Conference on*. IEEE, 2009, pp. 469–476.

[11] K. Kritikos and D. Plexousakis, "Towards optimal and scalable non-functional service matchmaking techniques," in *Web Services (ICWS), 2012 IEEE 19th International Conference on*. IEEE, 2012, pp. 327–335.

[12] L. Barakat, S. Miles, and M. Luck, "Efficient correlation-aware service selection," in *Web Services (ICWS), 2012 IEEE 19th International Conference on*. IEEE, 2012, pp. 1–8.

[13] Z. Zheng, H. Ma, M. Lyu, and I. King, "Qos-aware web service recommendation by collaborative filtering," *Services Computing, IEEE Transactions on*, vol. 4, no. 2, pp. 140–152, 2011.

[14] Z. Zheng and M. Lyu, "A distributed replication strategy evaluation and selection framework for fault tolerant web services," in *Web Services, 2008. ICWS'08. IEEE International Conference on*. IEEE, 2008, pp. 145–152.

[15] A. Watanabe, F. Ishikawa, Y. Fukazawa, and S. Honiden, "Web service selection algorithm using vickrey auction," in *Web Services (ICWS), 2012 IEEE 19th International Conference on*. IEEE, 2012, pp. 336–342.

[16] O. Skroch, "Multi-criteria service selection with optimal stopping in dynamic service-oriented systems," *Distributed Computing and Internet Technology*, pp. 110–121, 2010.