Abstract—QoS-aware service selection involves finding services to match a client’s functional and non-functional requirements. Current QoS advertisements typically provide a single value to represent the distribution of a non-functional property such as response time. However, the SOA literature implies that non-functional properties such as response time have inherently high variance in their values and thus representing non-functional properties with a single value does not reveal much about their actual distribution. This makes it hard for service clients to choose any selection strategy other than the conventional QoS-aware service selection which is geared for clients choosing “a single service” to serve their needs. In this paper, we propose a new strategy for QoS-aware service selection which takes advantage of the existing variability in QoS data to provide higher quality services with less cost compared to conventional QoS-aware service selection methods. In this method, we replicate each request over multiple independent services to achieve the required QoS. We also present a number of recommendations regarding the QoS advertisements in SOA so as to reveal more information about underlying distributions and thus enabling more sophisticated selection strategies. We will show using various examples how this approach works and enhances conventional QoS-aware service selection methods.

Keywords-SOA; QoS Advertisements; QoS-Aware Service Selection; Request Replication;

I. INTRODUCTION

Service Oriented Architecture (SOA) is an increasing trend in developing business applications. In this architecture, software functionality is represented as a set of services with well defined interfaces which can be reused to build various types of applications. A service is published by a service provider (from now on, we will simply use “provider”) and used by one or more service clients (“client” for short). Research in web services includes many challenging areas such as service composition, quality of service (QoS) aware service selection, etc.

Service selection is the process of choosing a service implementation from a pool of previously published services in a way that the selected service satisfies a client’s functional and non-functional requirements. This process is done in two steps. In Functional based service selection [1], [2], services matching a set of functionality requirements are retrieved. In the Non-functional Based (or QoS-aware) service selection [3], [4], functionally equivalent services, discovered in the previous step, are ranked based on non-functional requirements of the client.

The non-functional requirements are soft constraints on non-functional properties (NFPs), including quality of service (security, reliability, response time, call cost, etc.) and Context (location, intention, client name, provider details, etc.). In the process of QoS-aware service selection, a client submits to the service selection engine a set of non-functional requirements. Furthermore, many service providers advertise service functional and non-functional capabilities at the time of publication. Hence the service selection engine can match requirements of a client against advertised capabilities of service providers.

Non-functional properties of services, such as response time, are stochastic in nature. The dynamics of the environment in which a service is deployed, such as network-related delays and server congestion, can result in high variability in service non-functional properties. This yields two outcomes: On the negative side, the selected service may for a particular service invocation have response time that significantly exceeds the average value advertised. On the positive side, one could take advantage of the inherent variability in non-functional properties of a service to propose alternative service selection strategies.

In this paper, we present a novel alternative strategy, namely Request Replication, to satisfy the QoS requirements of a client in a more cost-efficient way by taking advantage of the existing high variance in NFPs. Unlike conventional QoS-aware service selection, in Request Replication we choose from available services “a set” of independent low-cost and low-quality services in a way that their cooperation provides the required QoS. Throughout this work, we specifically consider response time as a representative of performance related NFPs. We concurrently send a request to a set of services, take the fastest response, and discard the remaining requests.

The term “replication” is also used in other contexts in the literature of QoS-aware service selection. For example, Service Replication is a mechanism providers use to guarantee their quality of service obligations in their service level agreements (SLAs) [5]. Also, the idea of sending a request to multiple functionally equivalent services along with voting
mechanisms are used in the context of fault tolerance [6], [7], [8].

In this paper, we enhance the state of QoS-aware service selection in SOA from the client’s perspective. The contributions of the paper are:

1) We provide an alternative strategy to conventional “QoS-aware service selection” in SOA, which has the potential to allow clients higher quality services with less cost. In this method, a client can build better quality services from low cost, low quality ones.

2) We present a number of recommendations about service advertisements for performance related NFPs, and specifically service response time. We believe that the advertisements should provide enough information about non-functional properties of a service to enable clients to make better decisions with regard to service selection. For this reason, we believe that clients should know the distribution of NFPs, or a sufficient number of parameters to estimate the distribution.

The rest of this paper is organized as follows: Section II discusses related work in the literature of QoS-aware service selection in SOA. Section III presents the proposed “Request Replication” strategy. Section IV provides a number of recommendations for service advertisements. Section V concludes the paper.

II. RELATED WORK

Non-functional based (or QoS-aware) service selection involves finding services that match a set of non-functional requirements. This process is composed of three major steps: 1) Providers advertise non-functional properties of their services and clients specify their non-functional requirements; 2) A matching engine evaluates each non-functional requirement against available advertisements; 3) A selection engine aggregates the evaluation results and chooses a service as the best candidate to satisfy a client’s requirements.

So far there are no established standards for specifying NFP advertisements and requirements in a formal, machine-readable way [9]. However, various XML-based languages such as Web Service Level Agreement (WSLA) [10] and Web Services Offerings Language (WSOL) [11] have been proposed. These languages allow the specification of agreed-upon, non-functional properties of Web services in the form of Service Level Objectives (SLOs). A service level objective expresses a commitment to maintain a particular state of the service in a given period. The state is defined as a logical expression over predicates that refer to NFPs and defines an obligation, that is, what is asserted by the provider to the client. An example of such an obligation is: “it is guaranteed that the average response time of the service is less than 5 seconds”.

Many QoS-aware service selection approaches deal with single-valued descriptions of NFPs [12], [13]. That is, a non-functional property is specified with a single value and a given unit of measurement. For example, “the price of the service is 10 dollars/month” or “the throughput of the service is at least 10MB/sec” or “the average response time of the service is at most 10 msec”. In this case, to evaluate a non-functional requirement, one performs a one-to-one comparison between the advertised and requested values. This approach works well for deterministic NFPs such as cost but not for non-deterministic ones such as response time, throughput, availability, etc.

An extension to single-valued description of NFPs is to consider ranges that indicate boundaries on NFPs. For example, “the service delay is between 5 and 100 msec”. In this case, different approaches could be incorporated to evaluate non-functional requirements. In [14], [15], the boundary values of an NFP’s range are considered for the evaluation of non-functional requirements.

There are also a few approaches which specify NFPs as intervals with certain probability characteristics. Stantchev and Schropfer [16] recently proposed a structure for formalization of service level objectives and technical service capabilities. A service level objective in this structure has the following form: non-functional property + predicate + metric (value, unit) + percentage + if + qualifying conditions (non-functional property + predicate + metric). An example of such a SLO would be “The transaction rate of the service is higher than 90 transactions per second in 98% of the cases if throughput is higher than 500 kB/s.”

There are a few approaches that provide distribution-based description of NFPs. Rosario et al. [17] suggest that hard guarantees (e.g., response time always less than 5 msec) are not realistic and using soft probabilistic descriptions is more appropriate. In this work, NFPs are specified using probability distributions of the form \( P(X \leq x) \), where \( X \) is a random NFP (such as response time) and \( x \) is particular value of interest. Hwang et al. [18] consider an NFP measure as a discrete random variable associated with a probability mass function (PMF) which results in discrete probability distributions. This can be applied to model NFPs such as reliability, fidelity and price. However, it is less intuitive to use a PMF for describing other NFPs, such as response time, whose domain is inherently continuous. Li et al. [19] also assume a distribution-based model for description of NFPs. They propose a novel evaluation mechanism which indicates “the degree of match” between an NFP offer and a requirement, where a requirement is described using a utility function on the range of acceptable values.

Others also consider non-quantitative NFPs and introduce ontologies for semantic based matching, where semantic defines the relationship between QoS-related terms and provides a mechanism to evaluate NFP properties whose values are instances of given domain ontologies [20].

Conventional service selection approaches are geared for choosing a single service to satisfy the client’s requirements by the use of optimization algorithms to choose the
To the best of our knowledge, there are no other approaches similar to Request Replication where a set of services are incorporated to satisfy a single non-functional requirement set.

III. The Request Replication Strategy

In this paper, we are presenting an alternative strategy for QoS-aware service selection. The proposed strategy benefits clients in potentially receiving better services for less cost. In this method, a client uses multiple functionally equivalent services to get the quality they want while minimizing the service costs. We will show how and when using multiple services can increase the quality of service.

A. Motivating Example

Assume services $S_1$ to $S_5$ are different implementations of a calendar service with usage prices of $40, $10, $20, $10, and $10 per month, respectively. Assume that the following service advertisements are provided by corresponding providers of $S_1$ to $S_5$.

$S_1$: The response time of $S_1$ is less than or equal to 9s in 96% of the cases
$S_2$: The response time of $S_2$ is less than or equal to 10s in 92% of the cases
$S_3$: The response time of $S_3$ is less than or equal to 10s in 92% of the cases
$S_4$: The response time of $S_4$ is less than or equal to 8s in 70% of the cases
$S_5$: The response time of $S_5$ is less than or equal to 8s in 70% of the cases

Assume that a client is looking for a calendar service with the following QoS requirement.

$R$: Response time of the service must be less than or equal to 9s in 96% of the cases

With this information, a conventional service selection mechanism will choose $S_1$, because it is the only service matching the QoS requirement. The price to be paid in this case is $40 per month.

However, one can employ a different strategy than this conventional service selection. One could choose any combination of services, concurrently send a request to all of them, and pick the fastest response. In general, this new strategy could improve the results. Assuming that the service response times for $S_1$ to $S_5$ are mutually independent and exponentially distributed, we can show that choosing $S_2$ and $S_3$ is a better choice. In this case, the resulting response time would be less than 9s in 99% of the cases and the price to be paid would be $30 per month. The details are as follows:

First, we represent the advertisements of $S_2$ and $S_3$ as

$P(R_2 \leq 10s) = 0.92$
$P(R_3 \leq 10s) = 0.92.$

Knowing that the distributions of $R_2$ and $R_3$ are exponential, in this step, we need to find the rate parameters $\lambda_2$ and $\lambda_3$ of the corresponding distributions. We have

$P(R_2 \leq 10s) = 0.92$
$1 - e^{-10\lambda_2} = 0.92 \Rightarrow \lambda_2 = \frac{-\ln0.08}{10} \approx 0.25.$

Similarly,

$\lambda_3 \approx 0.25.$

Since we pick the fastest response, the resulting response time $R_{min}$ will be equal to the minimum of $R_2$ and $R_3$ and thus it is exponentially distributed with parameter $\lambda_2 + \lambda_3$, that is

$P(R_{min} \leq r) = 1 - e^{-(\lambda_2+\lambda_3)r}$

$P(R_{min} \leq 9) = 1 - e^{-(0.50)(9)} \approx 0.99.$

This satisfies the client’s requirement which is represented as

$P(R_{req} \leq 9s) = 0.96.$

Of course it is not at all clear that the exponential distribution is a reasonable choice for the individual response time distributions. However, if we have more information about the distribution of service response times we could estimate the underlying distributions. We will show in Section III-D how adding more information to current advertisements can change the results.

B. General Approach

In this paper, we are dealing with the general problem of QoS-aware service selection, defined as:

Having functionally equivalent services $S_1$ to $S_n$, find the most cost efficient service(s) that match the QoS requirements of a client.

In other words, we need to minimize the cost of the selected services while satisfying client defined constraints on NFPs.

We assume the following format for QoS advertisements.

$P(R_i \leq r_i) \geq p_i$
$E[R_i] = m_i,$

which is read as “the probability that the response time of service $i$ is less than or equal to $r_i$ is greater than or equal to $p_i$, where the mean response time of service $i$ is $m_i$.”

We also assume the following format for a non-functional requirement.

$P(R \leq r_{req}) \geq p_{req},$

which is read as “the probability that the response time of the selected service(s) is less than or equal to $r_{req}$ must be greater than or equal to $p_{req}$.”
C. Request Replication

The proposed strategy for QoS-aware service selection is called Request Replication. In this method, we choose one or more services whose aggregate QoS serves the needs of a client. We concurrently send a request to all of the selected services, pick the fastest response and cancel the other requests. A key motivation comes from the idea that taking advantage of even a small amount of additional choice for a client can lead to significant performance improvements. This idea has been explored by Mitzenmacher [25], amongst others, in another context (queueing problems).

Algorithm 1 presents a pseudo code description of the proposed Request Replication method. Similar to conventional QoS-aware service selection, in the first step we need to find all services with matching functional properties. We call this set the functionally eligible services (FES) set. We can use any functional-based service selection method to find this set. In the next step, we choose one or more services from FES whose aggregate QoS matches the request. This is done in two steps, as follows.

Algorithm 1 Request Replication

find all functionally eligible services and represent them as a set \( FES \).
fit appropriate distributions to the response time data of all functionally eligible services.
\( \text{cost} = \infty \)
for all \( fes \subseteq FES, fes \neq \emptyset \) do
if \( \text{cost}_{\text{sum}} = \sum_{s \in fes} s \cdot \text{cost} \leq \text{cost} \) then
compute cumulative distribution function (CDF) for the minimum response time distribution of the services in the subset \( fes \) at point \( r_{req} \), that is
\( CDF_{\text{min}}(r_{req}) = 1 - \prod_{s \in fes} (1 - CDF_s(r_{req})) \),
where \( CDF_s(r_{req}) \) is the cumulative distribution function for service \( s \) at point \( r_{req} \).
if \( CDF_{\text{min}}(r_{req}) \geq p_{req} \) then
\( \text{cost} = \text{cost}_{\text{sum}} \)
selectedServicesPool.replace(fes)
end if
end if
end for

STEP 1 - Fit appropriate distributions to the response time data of all functionally eligible services.

Current service advertisements do not indicate the actual distribution of NFPs. Therefore, we need to find estimates of the actual distributions in a way that they best describe available service advertisements. The first question here is "what sort of distribution better fits the available service response time data in SOA?".

The choice of what distribution to fit and the method that we use to make the fit do not affect our approach—the insights provided in the rest of this section will still hold for other methods of fitting a distribution. Gorbunova et al. [26] provide an approach to measure the performance and dependability of Web services from the client’s perspective and suggest that the Gamma distribution best describes response time data in SOA. Therefore, we represent response time data using Gamma distributions, if the actual distribution is unknown. As an illustration, we show how a Gamma distribution can be fit to a service advertisement of the form

\[
P(R_i \leq r_i) \geq p_i, \quad E[R_i] = m_i
\]

The Gamma distribution is a two-parameter family of continuous probability distributions. It has a scale parameter \( \theta \) and a shape parameter \( k \). A random variable \( X \) that is Gamma-distributed with scale \( \theta \) and shape \( k \) is denoted \( X \sim \text{Gamma}(k, \theta) \). The mean and cumulative distribution function of the Gamma distribution can be expressed in terms of the Gamma function parameterized in terms of the shape parameter \( k \) and scale parameter \( \theta \). Both \( k \) and \( \theta \) will be positive values. The mean of a Gamma-distributed random variable \( X \) is \( k\theta \) and the cumulative distribution function of \( X \) is

\[
CDF(x) = P(X \leq x) = \frac{\gamma(k, x/\theta)}{\Gamma(k)},
\]

where \( \gamma(k, x/\theta) \) is the lower incomplete Gamma function, defined as

\[
\gamma(s, x) = \int_0^x t^{s-1} e^{-t} \, dt,
\]

and \( \Gamma(k) \) is the Gamma function, defined as

\[
\Gamma(k) = \int_0^\infty t^{k-1} e^{-t} \, dt.
\]

There is no fixed way to fit a Gamma distribution to an available advertisement. In this paper, we incorporate the bisection search method where we search for the root of the following function

\[
g(k) = \frac{\gamma(k, r_i/k, m_i)}{\Gamma(k)} - p_i
\]

which is derived from the cumulative distribution function of the Gamma distribution (CDF) at \( x = r_i \) where \( \theta \) is replaced by \( m_i/k \).

The bisection method searches for the root of \( g(k) \) in an initial interval \([a, b]\) such that \( g(a) \) and \( g(b) \) have opposite signs. Then, it iteratively divides the interval in half in each step until it finds a sufficiently small interval that encloses the root. The method is guaranteed to converge to a root of \( g \) if \( g \) is a continuous function on the interval \([a, b]\) and \( g(a) \) and \( g(b) \) have opposite signs. It is not difficult to see that \( g \) is a continuous function of \( k \).

To find the initial interval \([a, b]\), we start by setting \( k \) to integer values and computing \( g(k) \). Knowing that \( k \) is a positive value, we compute \( g(k = iv) \) for \( iv = 1, 2, \ldots \) until one of the following is true:
• \( g(iv) = 0 \): in this case the root has been found and is equal to \( iv \).
• \( g(iv) \cdot g(iv+1) < 0 \): in this case \( a = iv \) and \( b = iv+1 \).

The method now divides the interval \((a, b)\) in two by computing the midpoint \(c = (a + b)/2\) of the interval. Unless \( c \) is itself a root, there are now two possibilities: either \( g(a) \cdot g(c) < 0 \) in which case we select the interval \((a, c)\), or \( g(c) \cdot g(b) < 0 \) in which case we select the interval \((c, b)\) to continue.

As an example, assume that in the advertisement above, \( r_i = 100, p_i = 0.81 \) and \( m_i = 66 \). In the initial step, we find out that \( k = 2 \) results in \( CDF(100) - 0.81 = -0.00466777 \) and \( k = 3 \) results in \( CDF(100) - 0.81 = 0.02147040 \). In the next steps we continue breaking this interval in half and testing \( CDF(100) - 0.81 \) for \( k = 2.5, 2.25, ... \). We find that \( k = 2.17116 \) is a very close fit. In fact, with \( 10^{-7} \) precision, the result is:

\[
\frac{\gamma(2.17116,217.116/66)}{\Gamma(2.17116)} - 0.81 = 0.00000003
\]

**STEP 2 - Check if any single service or combination of services satisfies the QoS requirements.**

In this step, we need to find one or more services where the distribution of the minimum of their response times matches the requirements. We also need to select from available candidates, the set of services with minimum cost.

For this purpose, we first choose a subset of functionally equivalent services FES where the cumulative cost of services is less than the current cost, initially set to infinity. Then we compute the distribution of the minimum response times for services in the selected subset (In Request Replication we choose the fastest response and thus the distribution of response times for the super service is equal to the distribution of the minimum response time of its underlying services). The cumulative distribution function for the minimum response time for a set of services \( fes \) is computed as

\[
CDF_{\text{min}}(R_{\text{min}} = r) = P(R_{\text{min}} \leq r) = P(\text{Min}_{s \in fes}(R_s) \leq r) = 1 - P(\land_{s \in fes}(R_s > r)).
\]

It is in general difficult to calculate this value unless the response times of services in \( fes \) are mutually independent. In this case:

\[
1 - P(\land_{s \in fes}(R_s > r)) = 1 - \prod_{s \in fes} P(R_s > r) = 1 - \prod_{s \in fes} (1 - CDF_s(R_s = r)).
\]

At this point we compute \( CDF_{\text{min}}(r_{req}) \) for all eligible subsets and update the pool of selected services, if a subset satisfies the requirement (i.e., \( CDF_{\text{min}}(r_{req}) \geq p_{req} \)).

**D. Motivating Example Revisited**

Adding to the previous example in Section III-A, assume that we also know the means of the distributions:

\( m_1 = 8, m_2 = 8, m_3 = 8, m_4 = 7.2, m_5 = 7.2 \).

Following Algorithm 1, in the first step we find matching Gamma distributions for all service advertisements. In this case:

\[
k_1 \approx 0.01, \, \theta_1 \approx 800
\]
\[
k_2 \approx 34, \, \theta_2 \approx 0.24
\]
\[
k_3 \approx 34, \, \theta_3 \approx 0.24
\]
\[
k_4 \approx 18, \, \theta_4 \approx 0.4
\]
\[
k_5 \approx 18, \, \theta_5 \approx 0.4
\]

In the next step we try different subsets of services, starting from single services, the results are:

\( S_1 \) is still a match.

\( S_2 \) or \( S_3 \) does not match \( CDF_2(9) \approx 0.78 \) and \( CDF_3(9) \approx 0.78 \).

\( S_2+3 \) (replication over \( S_2 \) and \( S_3 \)) does not match \( CDF_{min}(9) \approx 0.95 \).

\( S_4 \) or \( S_5 \) does not match \( CDF_4(9) \approx 0.86 \) and \( CDF_5(9) \approx 0.86 \). 

\( S_4+5 \) (replication over \( S_4 \) and \( S_5 \)) matches \( CDF_{min}(9) \approx 0.98 \). 

In this case, replication over \( S_4 \) and \( S_5 \) is preferred since it provides the required QoS with a lower price of $20. Note that the previous choice of \( S_2 \) and \( S_3 \) for replication is no longer an option. The extra “mean” information results in the construction of more accurate distributions for \( S_2 \) and \( S_3 \) which consequently indicates that replication over \( S_2 \) and \( S_3 \) does not actually satisfy the requirement.

**E. Discussion**

In this section we discuss a number of issues related to the Request Replication approach.

1) **QoS Heterogeneity:** One of the advantages of our algorithm is that, no matter what format the advertisements have and how one estimates a distribution for an advertisement, one is still able to use the Request Replication algorithm and the underlying mathematics are valid. In other words, our proposed algorithm can deal with a mix of advertisement formats that providers may use in a SOA-based environment. For example, assume service \( S_1 \) is advertised with having average response time of 10 msec and service \( S_2 \) with mean of 9 msec and variance of 12 msec. In this case, we can fit an exponential distribution to \( S_1 \) and a Gamma distribution to \( S_2 \) and rest of calculations in Algorithm 1 are still valid. The proposed Request Replication method also works for other distribution-based NFPs such as availability, reliability, etc.

Note that the minimum distribution here must be changed to the appropriate metric. As future work, we intend to investigate how Request Replication can be enhanced with optimization techniques to handle requests with multiple QoS constraints.
2) Service Interdependencies: The underlying mathematics works well if service response times are mutually independent. In other words, for any two services \( S_i \) and \( S_j \),
\[
P(\text{Min}(R_i, R_j) > r) = P(R_i > r)P(R_j > r) \text{iff } R_i \text{ and } R_j \text{ are independent.}
\]
If response times are dependent, the calculation is not as straightforward. In this case, we would need information about each probability as well as the joint probability, which may be difficult to calculate. Note that our approach should also work well if there is approximate independence, i.e. if the relation above approximately holds. As a future work, we are aiming to extend the current approach by modelling services’ interdependencies.

3) High Variance: The Request Replication method takes advantage of high variability in the response time data. If the response times were deterministic, the result of choosing more than one service would be the same as the result of choosing the service with minimum response time (which is what conventional QoS-aware service selection does). High variability in response times increases the chance that multiple services may complement each other with respect to performance and thus achieving better response times via Request Replication is more likely. Therefore, the provided method improves the results over conventional service selection as long as the response time data are highly variable. The precise form of the distribution is not important.

4) Replication Overheads: Similar to other domains such as fault tolerance [6], [7], [8], replication in this work also introduces a number of overheads.
- Using Request Replication, one should think of a mediator which replicates a service call to selected services, returns the first response to the client and cancels the remaining calls.
- Replication increases incoming network traffic to servers. However, we incorporate request cancellation to avoid lengthy responses and reduce congestion as much as possible.

As suggested by [6] the replication overhead is small and thus acceptable.

5) Request Cancellation Model: We use request cancellation as a companion to our request replication strategy. From the client’s point of view, cancelling a synchronous (request-response) SOAP call is the same as for any other HTTP call. The client just disconnects and stops listening for the response. A well written server will check whether the client is still connected before proceeding with lengthy operations (e.g. in .NET the server would check IsClientConnected) and should cancel the operation if not. One-way calls in asynchronous calls cannot be cancelled in this manner however, because the client has already sent the payload and disconnected. Cancellation of one-way calls would require an explicit call to some sort of cancellation method on the SOAP service (such as Abort), which it would have to explicitly support. Note that our strategy provides the best results when we have the “plan” pricing model for Web service usage as opposed to the “pay per usage” model. For the latter we need to take into account the cancellation fees when computing \( \text{cost_sum} \) in Algorithm 1.

6) Stateful Services: Our approach handles stateless services. Compared to stateful services, stateless services offer higher scalability and availability, are faster and easier to test, and can deal better with failover situations. Therefore, stateless services are preferred when developing Web services. Also, a stateful service may be implemented as a stateless service by sending all relevant information with every request.

IV. REVISITING SERVICE ADVERTISEMENTS

Looking at the literature, current QoS advertisements typically provide a single value (e.g., the average) to represent the response time of a service. For example, “Average response time of operation X from service Y is less than or equal to 0.5 seconds” or “Response time of operation X from service Y is less than or equal to 0.5 seconds in at least 95% of the cases”. This makes it difficult for clients to take a different strategy other than the conventional service selection where the client is limited to choosing “a particular service” as the “best available choice” with regard to their needs.

Conventional service selection would work well if the non-functional properties of services were actually deterministic. However, the literature [26], [27] suggests that non-functional properties of services and in particular, service response time, are non-deterministic and highly variable due to dynamics of the environment in which services are deployed. For example, factors such as network traffic and server congestion greatly influence the response time of services and current advertisements do not reflect this high variance.

Knowing more about the actual distribution of NFPs provides more opportunities for clients, including:

- **Adjusting the requirements:** non-functional requirements are usually soft constraints. Clients may be willing to change their non-functional requirements based on the availabilities and the offered prices, as in the case of QoS negotiation. Providing more information about the actual distribution of NFPs makes it easier for the clients to make better decisions about their requirements. As an example, assume the following situation. Services \( S_1 \) and \( S_2 \) are advertised with average response times of 0.95 and 1.05 seconds, respectively. A potential client needs a service with an average response time of less than 1 second. With this information, the client would choose \( S_1 \) as the better choice. On the other hand, assume that we know...
variances of the response times for services $S_1$ and $S_2$. In this case, $S_1$ has high variance ($\sigma^2 = 1$) and $S_2$ has low variance ($\sigma^2 = 0$). Knowing this information, the client may be willing to revise their request and choose $S_2$ as the better choice as it is more reliable with regard to timing constraints.

- **Changing the selection strategy**: as mentioned before, having a better understanding of the distribution of non-functional values, makes it possible for clients to choose from a variety of strategies for QoS-aware service selection. Request Replication is one example of such strategies which benefits clients by providing them with better quality services with less cost. There may also be other possibilities.

For these reasons we recommend that providers advertise the response time of their services using more than one representative values. This makes it possible to estimate a distribution for the response time of a service. For example, with two pieces of information, one can estimate a Gamma distribution, which is shown to be a good fit for describing the response time distribution in SOA [26]. Providers can use any two pieces of information about the response time. In this work we are considering advertisements of the form

$$P(R \leq \text{limit}) \geq \text{percentage}$$

$$E[R] = m,$$

as this is a lightweight change to the advertisements currently available. However, one could think of other pieces of information, such as mean $m$ and variance $\sigma^2$, for an advertisement.

No matter how the services are advertised, the Request Replication strategy can still be applied. As long as we estimate a distribution to the available advertisement we can use Algorithm 1 to find appropriate services for replication. If an advertisement provides one piece of information about the response time, we could fit an exponential distribution (a simple form of Gamma distribution, where $k = 1$), and if two pieces of information are provided a Gamma fit could be performed. Our algorithm, thus, is able to deal with a mix of advertisement formats which is very likely in a heterogeneous SOA environment.

### V. Conclusion

In this paper, we presented a novel alternative strategy to satisfy the QoS requirements of a client in a more cost efficient manner by taking advantage of the existing high variance in the values of non-functional properties. In this method, we use multiple independent low-cost, low-quality services to satisfy the QoS requirements of a single request.

There is a possibility to combine the Request Replication idea with the QoS negotiation process already existing in the SOA literature. In this approach one can negotiate with the provider of a service to acquire a reasonably low price for a service (and of course loose QoS as a result) and then incorporate multiple loose QoS services of this sort to acquire the required QoS. This idea is currently being investigated.

Also, the underlying assumption of independence should be considered more comprehensively. If service response times are correlated as a result of services sharing resources (e.g., shared services, network access, server, etc), the calculation of the minimum response time distribution becomes more difficult. As future work we are investigating how we can exploit knowledge about structure of the correlations to perform the required calculations.

Moreover, one can think of more complex QoS scenarios where the requirements involve multiple non-functional constraints that must be satisfied at the same time. Many researchers have used optimization approaches such as linear programming [28] and constraint satisfaction [29] techniques to deal with such scenarios in the traditional QoS service selection literature. Request Replication simply suggests more options (the use of multiple services in addition to single ones) when selecting services and can be an appropriate replacement for traditional QoS-aware selection in the optimization-based approaches. This idea is currently being investigated.

### References


