SATYA: A Reputation-based Approach for Service Discovery and Selection in Service Oriented Architectures

Alexandre G. Lages, Luci Pirmez
Federal University of Rio de Janeiro – Brazil
+55 21 2598-3214
{alexandrelages, luci}@nce.ufrj.br

Paulo F. Pires, Flávia C. Delicato
Federal University of Rio Grande do Norte – Brazil
+55 84 3215-3815
{flavia.delicato, paulo.pires}@dimap.ufrn.br

ABSTRACT

We present SATYA, a system that computes a reputation value for Web service providers in order to enhance the service discovery and selection process increasing reliability in SOA transactions. In this work, objective values of service evaluations supplied by monitoring entities are used along with subjective evaluations supplied by service consumers. The objective and subjective values are compared in order to: (i) validate subjective evaluations; (ii) minimize the degree of subjectivity of computed reputation values; and (iii) discover consumers' preferences in terms of QoS metrics. By assigning Web services a trustable reputation value, SATYA enhances the service descriptions provided by registries with additional information to be used during the service discovery phase.

Categories and Subject Descriptors

H.3.4 [**Systems and Software**]: Distributed systems.

General Terms

Management, Reliability.

Keywords

Web services, web service discovery, quality of service.

1. INTRODUCTION

Service Oriented Architecture (SOA) [12] strongly relies on mechanisms for advertising and discovering available services. Current Web standards for SOA advertising and discovering are mainly concerned on describing functional and syntactic features of services. Web Service Description Language (WSDL) [14] describes Web services mainly addressing communication issues and the syntactic description of service interfaces. Besides, current service registries such as UDDI [1] do not enable discovery and selection based on service capabilities and behavior, since they primarily rely on WSDL information. Therefore, many works suggest a need for richer semantics description in order to enhance service discovery and selection in SOAs [9,15]. It is a consensus that QoS requirements are important metadata to be included in service descriptions. In QoS enriched Web environments, besides metadata for describing QoS, it is also necessary a mechanism to assure that the advertised QoS is effectively provided. In SOA, such mechanism is implemented using Service Level Agreements (SLA) [6].

Traditionally, SLAs are bilateral agreements between service providers and their consumers, in which QoS parameters, such as service availability, throughput and response time, are precisely defined. Electronic contracts based on SLAs only work properly if third parties periodically monitor the agreed QoS parameters, in order to check if the QoS effectively provided to the consumers complies with the values previously established in the SLA. To accomplish such task, monitoring entities are included in SOA. Usually, monitoring entities employ probing mechanisms, which consist of sending a service request to providers and recording the delivered QoS value. In order to unburden the Web environment, the probing frequency must be as lower as possible, while guaranteeing the freshness of recorded QoS values. Therefore, there is a tradeoff to be addressed concerning probing frequencies and the freshness of stored QoS values.

However, the SLA approach is not suitable in all Web usage scenarios. For instance, pay per use services such as the Amazon Web services (amazon.com) have as their potential consumers the whole Web. Thus, the cost of establishing formal contracts may not be feasible in these scenarios. Moreover, the Web is an environment characterized by the freedom of choices and the adoption of fixed agreements tends to decrease such freedom. Therefore, for such scenarios the formal and bilateral model may be replaced by a more open model, in which providers publish the QoS parameters of each provided service in the service registry and potential consumers search and select services based on the published QoS. Despite the adopted scheme for QoS usage, the current environments still lack from guarantees that the QoS values (published or formally agreed) will be fulfilled. In the real world, contracts (such as SLAs) can be broken and published QoS are likely to be disrespected. This scenario implies that it is also necessary to address reliability issues, indicating for example, the degree of commitment a provider has regarding its published/agreed QoS values.

This paper presents SATYA system, developed with the purpose of augmenting the reliability of SOA-based systems. Reliability is represented in SATYA through reputation values, which are assigned to each service provider regarding each QoS parameter. SATYA assigns and manages values of service providers reputations, computed from: (i) subjective evaluation values issued by service consumers and (ii) objective evaluation values generated by monitoring entities. The objective and subjective values are compared in order to: (i) validate subjective

WIDM'07, November 9, 2007, Lisbon, Portugal.
Copyright 2007 ACM 978-1-59593-829-9/07/0011...$5.00.

evaluations; (ii) minimize the degree of subjectivity of computed reputation values; and (iii) discover consumer preferences in terms of QoS metrics. By comparing objective and subjective evaluations, SATYA can detect eventual discrepancies between them. Such discrepancies may suggest that either the consumer issued a careless or malicious evaluation (which is discarded by SATYA) or the objective values held by the system is out-of-date. To deal with the second case, SATYA adopts an approach for dynamically adapting the probing frequency in order to more realistically reflect the provider current state. This approach is a unique feature of our proposal in comparison to existing SLA-based systems in which probing frequency is fixed. By adopting a dynamically adjusted probing frequency, SATYA has the advantage of increasing the scalability of the whole system in terms of the number of probing messages needed to keep the freshness of QoS information.

Regarding the computed preference values, SATYA uses them (i) to assist consumers in the process of service discovery, constraining the search to providers belonging to the same preference group of the consumer and (ii) to compute the final reputation value of a service provider. SATYA approach enhances current SOA systems, augmenting the process of service discovery and selection with reliable QoS values. We carried on a set of experiments that proved the effectiveness of the proposed approach in guaranteeing a high level of consumer satisfaction while keeping the system overhead lower than traditional SLA-based systems.

The remainder of this paper is organized as follows. Section 2 describes SATYA system and explains how it enhances SOA-based systems. Section 3 shows experimental results that assess the overall benefits of using SATYA. Section 4 presents related works and Section 5 concludes the paper.

2. SATYA SYSTEM DESCRIPTION

SATYA was conceived to be integrated into SOA-based e-commerce systems [12]. SOA encompasses three components: service providers, service consumers and service agents. Service providers publish Web services descriptions (including QoS parameters) in a repository by using service agents. Service consumers query this repository to discover and select Web services that match their requirements. Upon the service discovery, the next step is the service usage carried on through the direct interaction between the consumer and the selected provider.

In an open environment like the Web, information provided by unreliable service providers are mixed together with those provided by reliable providers and there is no way of putting them apart. SATYA provides a service evaluation mechanism as a solution for this SOA issue. Such evaluation mechanism improves service discovery and selection by augmenting QoS values published by service providers with reliability values. These values are represented in the SATYA context as reputation values and they numerically express how much the published QoS value of a given provider is reliable.

To achieve its goal, SATYA performs a set of tasks: (i) generation of service providers objective evaluation values; (ii) retrieval and validation of subjective evaluations sent by service consumers; (iii) establishment of preference groups for both consumers and providers; (iv) computation of reputation values for each triple of provider-service-QoS parameter and (v) publishing of computed values of reputations and preferences for consumers and providers in service registries.

SATYA takes into account both subjective evaluations issued by Service consumers and objective values generated by Monitoring Entities (ME) [6]. ME obtains such objectives values by probing Service providers and thus retrieving QoS values effectively supplied by them. Such objective values are then used to validate the subjective evaluation values provided by consumers. This validation is important due to three main issues: (i) to avoid the formation of groups of malicious clients that might compromise the use of services; (ii) to minimize or prevents careless evaluations, and (iii) to allow adapting the probing frequency used by ME in a more precise manner.

The idea behind creating preference groups is to aggregate consumers and providers according to similarity criteria. With regard to consumers, a group encompasses consumers who have similar preference of service evaluations concerning to QoS parameters (consumers who wish services with the lowest-possible response time, for example, might compose a client preference group). With respect to providers, a preference group encompasses providers that are inclined to provide services respecting the published QoS in the same metric. After the establishment of preference groups, consumers can preferentially access providers that belong to their own group, that is, providers that tends to respect the published QoS metrics of their group.

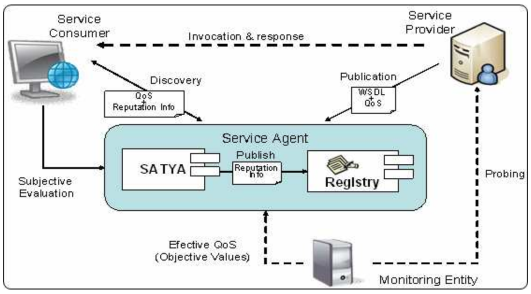

Figure 1. SOA enhanced with SATYA.

Figure 1 shows a SOA-based system enhanced with SATYA. The Service evaluation mechanism extends the Service agent generating reputation information and adding it to the service registry. Such information can then be queried by SOA consumers in the same way QoS values are. This Service evaluation mechanism provided by SATYA is composed of five modules described in the following sections.

2.1 Compliance Computation Module (CCM)

CCM computes the value of compliance for a given provider. In SATYA, the compliance value represents a measure of the provider behavior in respect to its commitment regarding each of its published QoS metrics. To achieve its goal CCM takes as inputs the published QoS and its actual delivered value, for each QoS metric, and generates compliance values for these metrics.

An important issue for computing compliance values is how to obtain the actual QoS delivered by Service providers. In SATYA, the effectively provided QoS data are obtained through the sending of probing messages by the Monitoring entity (ME). Such approach raises another issue, since there is a tradeoff between scalability and data freshness. In order to guarantee the freshness of the stored values of QoS delivery by service providers, it is necessary to have a very high frequency of probing messages. This solution is likely to generate updated values of QoS delivery but, at the same time, it overloads the Service providers, thus affecting their performance and overall system scalability. In order to obtain probing frequencies that reflect the current state of providers and, at the same time, preserving scalability, SATYA employs a strategy based on the comparison between objective values, obtained by probing, and subjective values, obtained from evaluations of Service providers supplied by their consumers. This comparison is accomplished as part of the functionalities of the CPDM module (Section 1.3).

For computing compliance values, CCM employs a mechanism based on an extension of the method presented in [11]. This compliance computation takes the published QoS values and the correspondent ones obtained by probing as inputs and it calculates the compliance value of the service as result. Let A be a QoS metric of a given service, Apub be the published value for A, and Aretreived the actual (probed) QoS value for A. The compliance computation of metric A when the service is invoked the jth time is given by formula (I): $Compliance_j = \frac{Apub,j - Aretrieved,j}{A_{pub,j}}$ (I)

As the range of values may be different for each QoS metric, in the present work the compliance value is normalized in the interval [- 1, 1]. Values greater than 1 or lower than -1 are defined as being “1” and “-1”, respectively. In the jth time of service usage is defined that if Compliancej (“<” or “=” or “>”) zero, for a given metric, the data obtained by probing represents an objective value (“above” or “equal” or “bellow”) the published value.

To provide a mechanism that adapts to the dynamism of the Web environment, our work extends the Compliance computation along with historical compliance computation. In other words, our work uses the historical QoS values together with instantaneous ones aiming to achieve more significant evaluation values. While the computation of instantaneous compliance considers only the most recent effectively QoS value obtained by probing, the historical compliance computation considers an average value based on the last effectively delivered QoS. The historical compliance value is used by SATYA to prevent spurious oscillations of QoS delivery influence the computation of the final reputation of a given provider. Consider a scenario in which the published QoS metrics for Response Time (RT) is 200 ms, the instantaneous RT value is equals to 800 ms and the historical RT value is 190 ms. In this case, the instantaneous compliance value is “-1” while the historical compliance value is “0.05”. Since the historical compliance is very low (approaches 0), the instantaneous TR value (800 ms), although well above the published QoS value, does not configure a problem; rather it means the occurrence of an anomaly, since historical compliance is low. Thus, the compliance computation based on historical values of the QoS metrics mitigate the effect of spurious behaviors in SATYA, due to, for example, short-term overload fluctuations.

2.2 Bias Computation Module (BCM)

The concept of biased evaluation in SATYA refers to an evaluation of a service according to a given bias in respect to a particular QoS metric. The BCM takes as inputs both the instantaneous and historical compliance values for each QoS metric calculated by the CCM and combine them using fuzzy inference rules to get two biased final evaluations for the service in that invocation. The first evaluation is calculated based on the instantaneous compliance values while the second is calculated based on the historical compliance values. BCM employs a set of inference rules to compute service biased evaluations. The first is the unbiased set, which considers all QoS metrics as having the same relevance for computing the evaluation. The remaining sets consider (each one) a different QoS metric as being more relevant for computing the service evaluation. Then, employing different sets of rules produces different estimated evaluations.

SATYA instantiates a BCM for each set of inference rules. Thus, considering a scenario with two QoS metrics, for instance, response time and availability, there will be three BCMs. The BCM assigned to the response time metric takes as input the compliance values of both metrics and it returns evaluation values (instantaneous and historical) biased to response time, meaning values that represent an evaluation that tends to consider such metric as been more relevant than the others. The same procedure is applied to the availability metric. The third BCM is in charge of computing unbiased values, representing the evaluation of a service without prioritizing any particular QoS metric. The BCM outputs are used as inputs for the discovery of consumer and provider preferences, as we describe in the next sub-Section.

2.3 Provider’s and Consumer’s Preference Discovery Modules (PPDM and CPDM)

To create and update providers’ preference groups, the PPDM uses a simple rating system that takes as inputs the historical biased evaluation values returned by all BCMs. Such evaluation values are used to update a data structure, named Providers Preferences Table (PPT), which stores information for defining providers bias to deliver QoS according to a given particular metric. PPT has one slot for each QoS metric plus one slot for unbiased behavior. Whenever the BCM produces a new set of historical evaluation values, for each slot in the PPT, the current value of the slot is incremented with the evaluation value referring to its corresponding QoS metric (or unbiased). Thus, the preference of a given provider can be discovered by searching PPT for the slot with the higher value.

Differently from PPDM, CPDM determines consumers’ preferences by combining the values of instantaneous biased evaluations generated by BCM along with the subjective evaluation value issued by consumers regarding their usage of a service. The subjective evaluation implicitly carries the rationale behind the evaluation given by the consumers, meaning the QoS metric that she judges as the most relevant when evaluating a service. One of the goals of our work is to make explicit such tacit knowledge (the consumer preference) and to exploit it both to increase the reliability and the system performance.

To compute the consumer preference group, CPDM uses a rating system based on a set of ranges. Each range has its lower and upper bounds defined as a percentage of each set of more recent instantaneous biased evaluations generated by BCM. Similarly to the PPDM, CPDM uses a data structure, named Consumer Preference Table (CPT), which contains one column for each instantaneous biased evaluation. When receiving a consumer subjective evaluation, CPDM scans the ranges of instantaneous biased evaluations and determines if the value of the evaluation falls within an existent range. If it does, the CPT column of the corresponding biased evaluations is set to “1” while the others are set to “0” (zero). The consumer’s preference is calculated by summing each of the CPT columns and determining the higher value. This means that, in the context of this work, there is only one consumers’ preference for all services. Future CPDM extensions are expected to address different consumers’ preferences for different category of services.

During the process of computing consumer’s preference, a situation might occur where the subjective value falls within no range. There are two hypotheses that may explain this situation. First, registry may be holding out-of-date objective values of QoS. Second, the consumer may be presenting a deceiving or malicious behavior. In this case, the subjective evaluation must be invalidated and discarded. In order to check the first hypothesis, SATYA assigns a freshness value to each objective evaluation stored in the registry. This freshness value is then compared with a predetermined threshold. If the freshness value is above this threshold, the objective evaluations are considered updated and the second hypothesis is assumed as true. On the other hand, if the objective evaluations are below the threshold, theses evaluations are considered outdated or invalid and, as a consequence, the ME probing frequency must be adjusted to collect freshener data. Since service consumers consist of application guided by specific requirements of their human users, consumers preferences may fluctuate in an ad hoc manner along the time, according to intangible factors. To address this behavior, the computation for determining consumers preference is done by only accounting the sum of the most recent elements in a CPT column (currently, we consider the last 8 elements).

2.4 Reputation Computation Module (RCM)

An important component of SATYA is the RCM, which has a fuzzy engine used for computing the final reputation value for a service provider. The provided reputation value is used as an additional measure in the process of selecting a given service besides the traditional service descriptors like its interface and QoS parameters. In SATYA, each published QoS metrics for each service has its associated reputation. Thus, given a specific provider which implements a set “s” of Web services, and considering a set of “n” published QoS metrics, “n*s” reputation values are assigned to such provider, one for each service and QoS metrics. In the current version of RCM, we used a component developed by our research team as part of a reputation system proposed in [7]. In such system, the computed reputation value is the result of a fuzzy process that uses subjective values reported by different consumers, weighted by their respective degree of relationship with the consumer requesting a reputation value about a service. To compute the final reputation of a provider in a particular QoS metric, RCM takes as input the output of both CPDM and PPDM along with the consumer subjective evaluation. Thus, the output of the fuzzy reputation machine is a set of reputation values according to different biases.

3. EXPERIMENTAL RESULTS

The main purpose of the experiments was to validate and assess SATYA mechanisms as well as to point out the potential benefits of their use in the context of service discovery. SATYA implementation was done in a simulation environment. We adopted MATLAB for developing the fuzzy engines used by SATYA and the networks simulator NS-2 [133] to carry out the experiments. Simulations were performed in four phases, according to the several goals to be achieved. The next sub-sections describe the experiments and present the results and analyses. Each set of such experiments considered 30 simulation rounds and a confidence interval of 95%.

3.1 Assessing the evaluation mechanism

The goal of this phase of experiments is to demonstrate the SATYA capability for rating service providers, in a scalable and efficient way. In other words, simulations were conducted to prove that the use of SATYA effectively provides an evaluation value of a particular provider that mirrors its current state while, at the same time, it decreases the need of high probing frequencies. It is worthwhile mentioning that high probing frequencies can contribute to degrade the provider performance itself.

The simulation scenario was set as follows: from a set of 60 nodes, one node was configured as a service provider, one as both monitoring entity and service registry and the remainders 58 nodes were configured as service consumers. QoS metrics for the simulated services include response time, availability and throughput. In the experiments, each service provider has different QoS performance for each QoS metric and each service consumer has different QoS requirements (moreover, each one prioritize a different QoS attribute). A model for service requests generation was adopted where the request rate was a configuration parameter ranging according to the goal of the different simulations. In this set of experiments, the service request rate was constant and configured as 1.5s. A model for QoS fault generation following a Bernoulli distribution was adopted, in which a provider had 98% of chances for delivering a service meeting the value of published QoS. For achieving the goal of this simulation phase, the same scenario was run twice: the first time without using SATYA mechanisms and the second time activating SATYA.

Figure 2 shows the results of simulation performed without using the SATYA mechanisms. These values will be used in the next step of simulation as a comparison basis. Results were obtained as follows. Initially, the consumer sends a service request to the service registry describing the desired service along QoS requirements. Then, the registry returns the stored value representing an unbiased objective evaluation value for the requested service (remember in this experiment there is only one provider). After, the consumer uses the provided service and an effective QoS value is obtained as result of the service utilization. It is worthwhile mentioning that such effective value reflects the current state of the provider and it is known in simulation time, but it is not available for the SATYA mechanisms when running the system in the “real world”.

At this point, a new unbiased objective evaluation value is computed, based on the effective QoS value provided when using the service. After that, the difference between this value and the unbiased objective evaluation value initially stored in the registry is computed. The freshness of data stored in the registry is directly proportional to the frequency of probing adopted by ME. In order to represent different thresholds for the freshness of the stored objective evaluation values, three ranges were adopted (2%, 4% e 10%), represented by the three curves shown in Figure 2. For instance, for a 2% range, the value held in the registry is considered updated only if the difference between such value and the effective one is lower or equal to 2% (of the effective value). Figure 2 plots the success rate as a function of the probing interval, for different values of ranges. Success rate is a measure that denotes in percentage terms the degree of freshness of the registry information, meaning how much the stored values are close to the values effectively obtained when using the services. A value of 1 (or 100%) means that for all service requests the stored values correspond to the effectively provided QoS value. We simulated five values for the time interval of sending probing messages (5, 10, 20, 30 and 60 seconds).

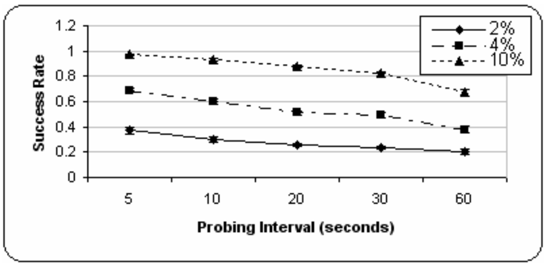

Figure 2. Success Rate – Without SATYA

As Figure 2 shows, the success rate for all ranges decreased proportionally to the increasing of the probing interval. So, as it was expected, the higher is the frequency of probing (lower probing interval), the freshest the values stored at the registry are. Considering the range of 10% and a probing interval of 5 seconds, the success rate is about 97%, meaning that in 97% of cases the value of effectively delivered QoS and the objective evaluation value stored in the registry matched (thus the registry values were updated). At the range of 2%, the success rate drops to 37%.

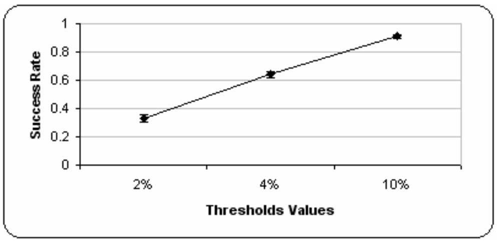

Figure 3. Success Rate – With SATYA

Figure 3 presents the values for the success rate obtained when activating SATYA, considering the thresholds (2%, 4% and 10%). Figure 3 does not plot the probing interval, since this value varies along the time when SATYA is activated. To analyze the system behavior, we compared values presented in Figures 2 and 3, considering a probing interval of 5 seconds. Such interval was chosen since it represents the worst case for SATYA (when QoS values stored in the monitor are the freshest). Results show that, for a 2% range, the success rate is about 37% without SATYA while with SATYA this value is about 33%. For ranges of 4% and 10%, the success rates are respectively 69% and 97% without SATYA, and 64% and 91% with SATYA. Therefore, in the worst case for SATYA, that is, with the monitor adopting a high probing frequency, the success rates obtained when using SATYA are very close to those obtained without using it. This result highlights the fact that SATYA allows registries to hold fresh (and therefore reliable) evaluation values of service providers without incurring in the overhead of MEs keeping a high probing frequency. To ratify the point, the final step of this simulation phase is to verify the actual reduction in the number of monitoring messages when SATYA is activated.

Figure 4 plots a value in % that represents the ratio between the number of probing messages sent with and without SATYA mechanisms, for the different values of ranges and with the probing interval set as 5 seconds. We observe that, for a 2% range, about 38.6% of messages are sent with SATYA activated when comparing with the total number of messages sent without SATYA. Considering the other ranges (4% and 10%) this ratio is still more favorable for SATYA (only 21% and 12% of sent messages in comparison to simulations without SATYA). Results show that SATYA is able to provide a highly scalable solution for QoS monitoring in the context of Web services, without degrading the accuracy of the provided evaluation values.

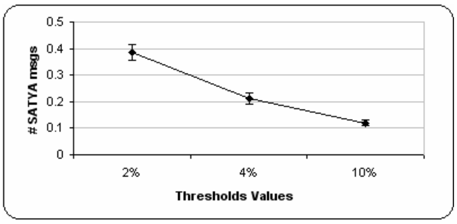

Figure 4. % of sent messages with SATYA

3.2 Assessing the Use of Preference Groups

A second phase of simulations was accomplished with the goal of evaluating the benefits of using SATYA for creating preference groups and using such groups during the process of service selection. To achieve this goal, we performed a set of comparisons among service evaluations provided by consumers when accessing services delivered by providers of their own group or by providers of different (random) groups. For these experiments, from the 60 nodes, 10 were set as providers and the remainders as consumers. To analyze the effect of using preference groups, two graphs were built, one representing the choice of service providers accomplished in a random way (that is, without considering preference groups) and the other one representing the choice of providers belonging to the same group as the consumer.

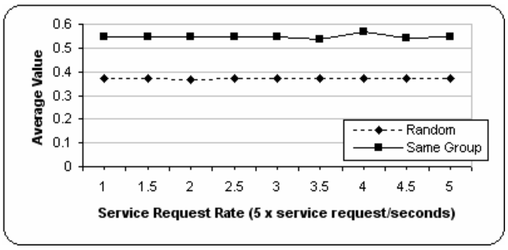

Figure 5. Average Subjective Evaluation of delivered services

Figure 5 depicts, for different service request rates, the average of the subjective evaluations provided by consumers about services that were delivered by providers belonging to: (i) the same preference groups as the respective requesting consumers ("Same Group"); and (ii) different preference groups ("Random"). We observe that the average of the evaluation values is higher when providers of the same group were accessed, for all values of service request rates. This result points out that the use of services supplied by providers of the same group as the consumer supplies a higher level of user satisfaction.

However, the use of preference groups when selecting providers may have the drawback of resulting a higher number of QoS violations. These violations occur since the search space of providers is reduced due to the use of preference groups. To minimize this problem, SATYA employs load balancing when selecting service providers in a given group.

3.3 Assessing the Benefits of Load Balance

This third phase of simulations had the goal of evaluating the benefits of using load balancing when selecting service providers of the same preference group of the client. In this phase, the same scenario described in Section 4.2 was adopted, and the service request rates varied aiming to compare the scalability of the system usage. Three different rate values, denoted by high, intermediate and low, were adopted in the experiments. Rates denoted as High were about 8 req/sec; Intermediate were about 4 req/sec and Low were close to 1.5 req/sec.

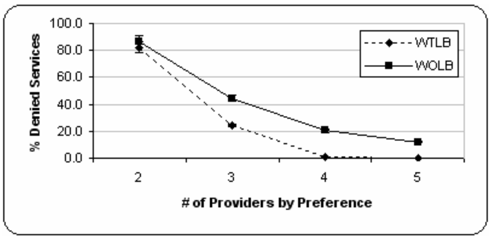

Figure 6. Percentage of Denied Services – High Request Rate

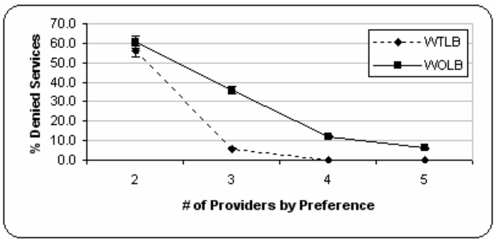

Figure 7. Percentage of Denied Services – Intermediate Request Rate

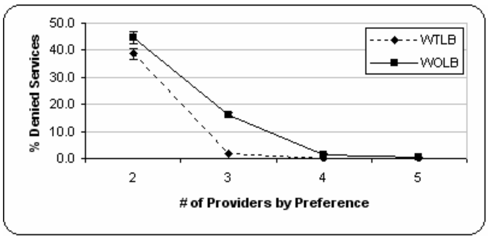

Figure 8. Percentage of Denied Services – Low Request Rate

The mechanism for load balancing when choosing service providers was simulated as follows. Whenever a client requests a service (whose provider is selected among providers in the same preference group) SATYA returns the provider with the lowest request rate for time interval. The load balancing is always accomplished within a given preference group, that is, the selected provider will be the one with the lowest request rate for time interval in the client preference group. Thus, the experiments carried out to evaluate the load balancing mechanism were based on comparing the amount of requested services that are denied with and without the use of the mechanism. This denial of services occurs whenever a selected provider lacks of resources for delivering the published QoS. The numbers of providers and clients were variable parameters in the simulations. Three QoS metrics were used: Response Time, Availability and Performance. Each node configured as a client was assigned a preference group. Considering the three adopted QoS metrics, three preferences groups were established (one for each metric). The number of providers belonging to each preference group range from 2 to 5 (meaning a total of 6 to 15 providers for attending client requests).

Figures 6, 7 and 8 present the % of denied services for each different service request rate adopted in the experiments (High, Intermediate and Low). A significant reduction is achieved when the load balancing (LD) mechanism was used. The benefits achieved with the use of this mechanisms result from adding in the Service Register the information of which provider has the lower workload in a given time and using such information for selecting the provider to attend a request. For example, in scenarios running with three providers in each preference group and a high rate of service requests, the number of denial of services was about 22.3% lower with the load balancing mechanism (WTLB) than without it (WOLB). In experiments with four providers in each preference group this difference decreases to 19.4% (Figure 8). The obtained results demonstrated that the load balancing is a useful and efficient mechanism for distributing the total of service requests among the available providers (inside a preference group) thus increasing the amount of requests able to be met, achieving a best overall usage of the Web resources.

3.4 SATYA in the Service Discovery Process

Experiments performed in this simulation phase had the goal of evaluating how the behavior of a service provider influences its final computed reputation value. Such reputation value can be used as an indicator of how frequently a provider actually meets its published QoS. In a service discovery process enhanced with SATYA, the choice of a provider may take into account: (i) QoS values published by providers; (ii) reputation values computed by SATYA; (iii) both of them. A provider reputation is a value that denotes how much this provider effectively meets the published QoS. Therefore, using SATYA in the process of service discovery, a consumer will be able to search in a service register for providers that have (i) the best QoS value for a given metric (for example, the provider that has the lowest Response time); (ii) the highest reputation value for a given metric; (iii) the best balance between reputation and QoS values.

In this phase, the same scenario with 60 nodes was run and the same three values of service request rates (High, Intermediate and Low) were adopted. The number of providers and consumers were set as 15 and 45, respectively. The simulated mechanism for a consumer choosing a provider comprises the following steps: (i) selection of a service from a set of available services; (ii) selection of a set of QoS requirements from the available requirements (for example, selection of Response Time); (iii) assigning of values for QoS requirements (for example, providers with response time lower than 1000 ms); (iv) running the operation ProvidersReturns which returns a list of providers that fulfill the established QoS requirement (in this case, all providers with a published response time lower than 1000 ms); (v) once receiving the selected list of providers, running the operation ProvidersSelects, that determines which is currently the best provider to be used given the consumer criteria of choice (best published QoS, for example). Once a provider is selected, the consumer is able to perform the requested service by running the operation PerformService. The PerformService operation simulates a provider performing a service request according to its current execution context, which dynamically varies depending on a set of factors and directly influences its capability of meeting the previously published QoS. For modeling the behavior of a provider, a Bernoulli process was adopted. According to the adopted model, a provider had 90%, 80%, 70%, 60% and 50% of chances for delivering a service according to the published QoS (meaning its QoS performance). Figure 9 shows the obtained results. Axis X denotes the variable value of QoS Performance of a provider and axis Y represents the Final Average Reputation of the simulated providers. According to the graph of Figure 9, the final average reputation of the providers presents a linear drop proportional to the QoS performance of the provider. This drop occurs since lower reputation values are assigned along the time to providers that disrespect the published QoS. The values in Figure 9 show that reputation values really reflect how reliable the QoS values published by a provider are.

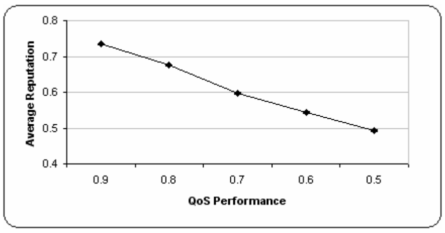

Figure 9. Variation of Reputation according to current provider QoS performance

In a final step, we evaluate the overall benefits of using SATYA in comparison to a traditional SOA system, which is based on a service registry providing values of published QoS for each registered provider without any reputation information. Therefore, in traditional SOA systems, only after a consumer uses a service provider it will find out if it is or not able to supply the service meeting the previously published QoS. On the other hand, in SATYA enhanced systems provide to consumers reputation values, which reflect providers' real capacity of service provision.

In this simulation phase we compared SATYA enhanced systems with traditional SOA in terms of the degree of consumer satisfaction, given by the percentage of values of published QoS that are not fulfilled. To reach the simulation goal, the same scenario of Section 4.2 was run, first without using SATYA and after with SATYA activated. For these experiments, the following environment was simulated: from the 60 nodes, 10 were set as service providers and 50 as consumers. All providers were set as supplying the same type of service and publishing its QoS value concerning the same metric, so that whenever a customer requests a service all providers have the same chance of being selected. A service request rate of 5 requests per second was set.

In traditional SOA environments, all providers (those that meet and those that do not meet published QoS) have the same probability of being chosen, since they publish the same QoS value. The negative impact of the presence of unreliable providers will be reflected in the final degree of customers' satisfaction. On the other hand, in a SATYA enhanced environment, the process of choosing a service provider uses both the value of published QoS and the value of reputation. Thus, providers that do not meet the published QoS will receive a low reputation value so that the probability of these providers been chosen will progressively decrease along the time.

In the simulations, consumers are individually assigned a random criterion for choosing services. The possible criteria are defined as described in Section 4.4. Figure 10 shows the % of occurrences of QoS violations with and without the use SATYA. Axis X denotes the amount of providers that meet the published QoS (therefore, the reliable providers) in each simulation round and axis Y denotes the percentage of QoS violations. In simulations where all providers are reliable, there is no occurrence of QoS violation.

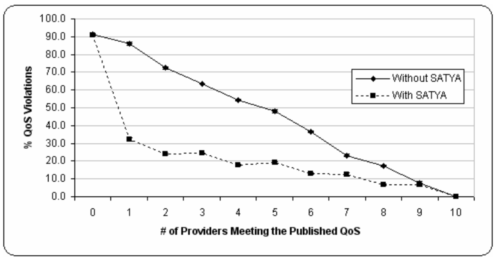

Figure 10. Percentage of QoS violations

According to Figure 10, the lack of reliable providers leads to a large number of QoS violations (approximately 91.3% without SATYA and 91.0% using SATYA), generating a low degree of customer satisfaction. When inserting only one reliable provider in the system, the amount of QoS violations decreases to about 32.4% when using SATYA and to 86.1% without its use. Such results reflect the use of reputation values in the service discovery process, since unreliable providers tends to be not selected by customers most of the time. However, the presence of only one reliable provider among a set of available providers can generate a high amount of service requests for such provider. Consequently, this unique reliable provider may get overloaded and, in a near future, becoming unable to meet its published QoS unless it updates the published value in order to reflect its current state. The addition of new reliable providers generates a linear drop in the percentage of QoS violations with and without SATYA.

4. RELATED WORK

Reputation-based systems have been used as efficient tools for services discovery and selection [12], mainly in environments where traditional agreements (SLAs) are unfeasible or undesirable. Commercial sites like eBay [2] employ a method known as Qualification Process for evaluating consumers after the purchase or sale of products. This method suffers from two main drawbacks. First, it does not take into account the reputation of the consumer who evaluates the service. Second, consumers are not rated according to their past transactions. Therefore, evaluations of newcomers have the same weigh as evaluations of older ones, possibly introducing distortions in the computed reputations since new users tend to be less reliable than older ones. The academia has presented evaluation systems that are more sophisticated. In [4,8] models for selecting services based on both QoS parameters and consumers evaluations are introduced. Several proposals [1,3] use the consumer evaluation of services for establishing the prices of e- transactions. Service prices are calculated by taking into account the provider reputation.

In the previously described works, reputation is based on the consumer perception concerning the usage of a given service and it is not suitable to determine how consistent is a service regarding the effectively provided QoS (meaning the provider behavior along the time). Another issue that is not addressed in these works concerns the reliability of the evaluation provided by the service consumer [4,5,11]. With the aim of tackling these issues, in [5] is proposed a reputation model where the reputation is a function of both the consumer evaluation – called subjective dimension of reputation – and a set of historical QoS values effectively supplied by providers – called objective dimension. An extension of this proposal [11] adopts a fuzzy-based approach for inferring the rationale behind a particular subjective evaluation given by a consumer. The inferred rationale is used to (i) detect the formation of collusion, (ii) discover the consumer preferences and (iii) provide recommendations to other consumers. This work shares a set of common features with the proposed system, SATYA, but they differ mainly regarding the adopted methods for achieving objective values of QoS and for exploiting subjective evaluation values. Differently from [11], in SATYA the consumer evaluation is used for computing the reputation value but also for dynamic adjusting the probing frequency of a monitoring entity. A major advantage of our approach is the increase of scalability resulting of the reduction of the number of probing message used in the reputation system. This enhancement is a consequence of SATYA requires probing only when a discrepancy between the QoS values stored in the registry and those perceived by the users is detected.

The work in [9] proposes a framework to allow reputation-based service selection in Semantic Grids environments. The proposed framework provides an adaptive reputation-aware service discovery algorithm and a service-oriented distributed reputation assessment algorithm. According to the authors, current service discovery techniques like UDDI are unable to allow automatically locate services based on both provider capabilities (given by QoS metrics) and behavior (such as the provider trustworthy). The main difference between this work and our proposal is that SATYA mechanisms adopt the joint use of objective and subjective evaluations thus validating the consumer's evaluations. Without such validation, the system becomes more vulnerable to malicious consumers or careless evaluations. A second advantage of our approach is that SATYA relies on probing to obtain effective QoS values while in [9] only published provider capabilities are considered, meaning that effective QoS values are not taking into account in computing reputation values.

5. CONCLUSIONS

We presented SATYA, a system for increasing the reliability in the SOA processes of service discovery and selection without adding significant overhead overall the Web system (consumers, providers and the underlying network infrastructure). For achieving this goal, SATYA model relies upon the joint action of monitoring entities and consumer opinion about provided services. It adopts a fuzzy logic-based approach for calculating reputation values for service providers. Reputation values represent a provider degree of reliability denoting a measure of how frequently such provider respects its published values of QoS parameters. Mechanisms had been proposed for (i) decreasing the level of subjectivity in service evaluations issued by consumers; (ii) providing values for service consumers reputation to denote a degree of commitment a provider has regarding its published QoS; (iii) creating preferences groups that aggregate consumers that present a similar behavior when evaluating services, and aggregate providers that lean to provide services with higher values of QoS in the same metric.

Since SATYA mechanisms are able to validate consumers evaluations and at the same time to rate providers according to their current and historical behavior, it augments the mutual confidence among customers and providers, leveraging the potential of Web usage for whatever transaction that requires some level of trust. Moreover, preference groups can be used as an incentive mechanism to increase the consumer participation in the distributed process of service evaluation.

6. REFERENCES

[1] Bellwood, T., et al. UDDI V. 3.0 Specification [Online]. Available at: http://uddi.org/pubs/uddi-v3.00-published-20020719.htm.

[2] eBay. Available at: http://www.ebay.com, last access December 2006.

[3] Jurca, R., Faltings. B. Reputation-based Pricing of P2P Services. In Proc. of ACM SIGCOMM'05 Workshops, Philadelphia, USA, pp: 144-149, 2005.

[4] Kalepu, S., Krishnaswamy, S., Loke, S., Verity; A QoS Metric for Selecting Web services and Providers. In Proc. of WISEW, pp: 131- 139, 2003.

[5] Kalepu, S., Krishnaswamy, S., Loke, S.W. Reputation = f(user ranking, compliance, verity). In Proc. of the IEEE ICWS, pp. 200- 207, 2004.

[6] Keller, A. Ludwig, H. The WSLA Framework Specifying and Monitoring Service Level Agreements for Web services. JNSM, 11(1), pp. 57-81, Kluwer Academic Publishers, 2003.

[7] Lages, A., Delicato, F.C., Pirmez, L. Sistema de Reputação Orientado a Serviços baseado em Lógica Nebulosa. In Proc. of VI SBSEG, Brasil, 2006 (in Portuguese).

[8] Liu, Y., Ngu, A.H., Zeng, L.Z., QoS Computation and Policing in Dynamic Web Service Selection. In Proc. of the 13th WWW conference, pp: 66 -73, 2004.

[9] Majithia, S., Shaikh, A., Rana, O., Walker, D.. Reputation-based Semantic Service Discovery. In Proc. of the 13th IEEE WET ICE, 2004.

[10] Sherchan, W., Loke, S. W., and Krishnaswamy, S. A fuzzy model for reasoning about reputation in Web services. In Proc. of the 2006 ACM SAC, pp. 1886-1892, 2006.

[11] Singh, M.P., Huhns, M.N., Service-Oriented Computing: Semantics, Processes, Agents. Wiley, 2005.

[12] The Network Simulator – ns2. Available at: http://nsnam. isi.edu /nsnam/i ndex.php/ User_Information, last access: December 2006.

[13] W3C Working Draft [Online]. "Web services description language (WSDL) Version 2.0 Part 1: Core Language". Available at: http://www.w3.org/TR/wsdl12/.

[14] Wishart, R., et al. SuperstringRep: Reputation-enhanced Service Discovery. In Proc. of the 28th Australasian Computer Science Conference, Australia, 2005.