Response time of BPEL4WS constructors

Serge Haddad
LSV, ENS de Cachan
61 Avenue du Prsident Wilson
94235 CACHAN, France
serge.haddadd@lsv.ens-cachan.fr

Lynda Mokdad
LACL, Université Paris 12
61 Avenue du Général de Gaulle
94010 Créteil, France
lynda.mokdad@univ-paris12.fr

Samir Youcef
Lamsade, Université Paris Dauphine
Place du Mal. de Lattre de Tassigny
75775 cedex 16, France
samir.youcef@lamsade.dauphine.fr

***Abstract*— Response time is an important factor for every software system and it becomes more salient when it is associated with introducing novel technologies, such as Web services. Most performance evaluation of Web services are focused toward composite Web services and their response time.One important limitation of existing work is in the fact that only constant or service exponential time distribution are considered. However, experimental results have shown that the Web services response times is typically heavy-tailed, in particulary, if there are heterogeneous. So, heavy-tailed response times should be considered in the dimensioning Web services. In this study, we propose analytical formulas for mean response times for structured BPEL constructors such as *sequence*, *flow* and *switch* constructors, etc. The difference with previous studies in the literature, is that we consider heterogenous servers, the number of invoked elementary Web services can be variable and the elementary Web services response times are heavy-tailed.**

Keywords: composite Web service, BPEL constructors, response times, heavy-tailed.

I. INTRODUCTION

Service oriented computing utilizes services to support low-cost, flexible software. The underlying services are loosely-coupled, thus allowing rapid change of such systems. Although a framework for defining the functional interfaces of Web services has been established, non-functional properties remain under-development. The Web services architecture is defined by W3C (The World Wide Web Consortium) in order to determinate a common set of concepts and relationships that allow different implementations working together. The Web services architecture consists of three entities, the service provider, the service registry and the service consumer. The service provider creates or simply offers the Web service. The service provider needs to describe the Web service in a standard format WSDL (Web Service Description Language), which is often XML, and publish it in a central service registry UDDI (Universal Description, Discovery and Integration). The service registry contains additional information about the service provider, such as address and contact of the providing company, and technical details about the service. The service consumer retrieves the information from the registry and uses the service description obtained to bind and to invoke the Web service, using the SOAP (Simple Object Access Protocol) protocol.

*This work is supported by French research projects CheckBound ANR-06-SETI-002 and Perso ANR-07-JCJC-0155-01

Elementary Web services, such as described by WSDL, are conceptually limited to relatively simple functionalities modeled through a collection of simple operations. However, for certain types of applications, it is necessary to combine a set of individual Web services to obtain more complex Web services, called composite or aggregated Web services. This last is possible using BPEL4WS (Business Process Execution Language For Web Services) standard, which is the result of the merger of the previous languages such WSFL (Web Services Flow Language) and XLANG (XML Business Process Language). One important issue within Web service composition is related to their Quality Of Service (QoS), which must be guaranteed for an adhesion clients. Web services quality of services is a combination of several properties and may include availability, security, response time, and reliability of Web services. For this, quantitative methods are needed to understand, to analyse and to operate such large infrastructure.

The goal of our research is to propose an extension of a recent study [1], where we have taken into account different statistical characteristics for the services and a random number of invoked services and Web service response time are supposed exponential with different parameters, contrarily to the models presented by Manascé [2] and Sharf [3]. However, most existing work only considers constant or exponential service times. As will be shown in [15][5], measurements in the WWW and in e-commerce systems have observed heavy-tailed server response time distributions. In this study, we take into account the fact that the Web services response time is typically heavy-tailed, like Pareto distribution, which is attributed to the burstiness of arriving requests [15]. More precisely, the objective of this paper is to consider the heavy-tailed response times in the dimensioning of web service platforms.

The rest of the paper is structured as follows. Section II presents the related work. Section III details the different structured BPEL constructors. Section IV presents analytical formulas for response time of these constructors. In section V, we give the response time formula for multi-choice pattern which is a generalization of switch constructor. Numerical results are given in section VI. Finally, section VII concludes and gives some perspectives to this work.

II. RELATED WORK

Major works in the domain of Web services performance are concentrated towards composite Web services and their response time. Although there have been several studies reported on the workload characetrization of general Web servers, where the response-time distribution is found to be heavy-tailed, which has been attributed to the heavy-tailed nature of request and response file-sizes [15][6]. However, most existing work only considers constant or service exponential time distribution. Only few studies have been taken into account this result on the computation of composite Web services response time. Actually, the execution of a composite service have been studied as a fork-join model in [2], where Web services response time are supposed exponential with the same parameters, excepted one which is slower than others. This model states that a single Internet application invokes many different Web services in parallel and gathers their responses from all these launched services in order to return the results to a client. Sharf [3] studies the response time of a centralized middleware component performing largescale composition of web services. This last work is similar to the first study [2], that analyzes the effects of exponential response times. The work is more oriented towards studying fork-join model in order to understand the merger of results from various servers. More recently, [26] proposed how service providers can optimally allocated to support activities of business process with topologies that can include any combinaison of BPEL constructors. However, authors are content to propose a general formula for a given composite Web service without giving the exact result when the service of elementary Web services are know. The exact response time of fork and join system, under some hypothesis, can be found in [7]. However, these last state that the number of servers is equal to two, the job arrival is Poisson process and the tasks have exponential service time distribution. Nelson and Tantawi [8] proposed an approximation in the case where the number of servers is greater or equal to two and homogeneous exponential servers. Thereafter, a more general case is presented in [9] [10], where arrival and service process are general. An upper and lower bound are obtained by considering respectively $G/G/1$ and $D/G/1$ queuing parallel systems. Klingemann and al. [11] use a continuous Markov chain to estimate the execution response time and the cost of workflow. In [11], authors propose an algorithm which determines the QoS of a Web service composition by aggregating the QoS dimensions of the individual services, based on a collection of workflow patterns defined by Van der Aalst's and al. [12], where Web services response times are supposed constants. These QoS include upper and lower bounds of execution time as well as throughput. In [13], we have studied end-to-end response time for composite Web services representing a factor of Internet overhead in the execution model, using simulation technique. Contrarily to these previous studies, where the servers are not heterogenous, their number is always constant and their response times are supposed exponential, the aim of this paper is to overcome theses limitations. Thus, we propose analytical formulas for mean response time of composite Web services assuming that servers are heterogenous, the number of invoked elementary Web services can be variable.

III. BPEL CONSTRUCTORS

Business Process Execution Language for Web services (BPEL4WS) has been built on IBM's WSFL (Web Services Flow Language) and Microsoft's XLANG (Web services for Business Process Design) and combines accordingly the features of a block structured language inherited from XLANG with those for directed graphs originating from WSFL [14]. The language BPEL is used to model the behavior of both *executable* and *abstract* processes.

- An abstract process is a not an executable process and which is a business protocol, which use process descriptions that specify the mutually visible message exchange behavior of each parts involved in the protocol, without revealing their internal behavior.
- An executable process specifies the execution order between a number of activities constituting the process, the partners involved in the process, the messages exchanged between these partners and the fault and exception handling specifying the behavior in cases of errors and exceptions.

In the BPEL process each element is called an activity which can be a primitive or a structured one. The set { *invoke, receive, reply, wait, assign, throw, terminate, empty* } are primitive activities and the set {*sequence, switch, while, pick, flow, scope*} are structured activities.
In this paper, we are interested on the *sequence*, *flow* and *switch* activities also called constructors. In the following, we give analytical formulas to evaluate the response times to each considered constructor.

IV. RESPONSE TIMES OF STRUCTURED BPEL CONSTRUCTORS

In this section, we give analytical formulas for mean response times for structured BPEL constructors and we consider the case that the execution time of each elementary Web service s_i, of a composite Web service S, is heavy-tailed and we consider also that the number of invoked elementary services are variable. The Pareto function distribution is given by the following equation :

$$F(t) = \begin{cases} 0 & \text{t} \leq k \\ 1 - (\frac{k}{t})^{\alpha} & \text{t} > k \end{cases} \quad (1)$$

which has an infinite variance for $\alpha < 2$ and is then heavy-tailed.
Thus, we consider in the following the control patterns supported by BPEL standard. More specifically, the control patterns considered are: sequence, parallel split (flow), exclusive choice (switch), multi-choice. This last pattern is not directly supported by BPEL, but we can implement it using control links inherited from WSFL.

A. *Computation for the* **sequence** *constructor*

The *sequence* constructor correspond to a sequential execution of s_1 to s_n elementary Web services. The analytical formulas of mean response time $E(T^{sequence})$ is given by the following proposition:

Proposition 1: When elementary Web services $s_i, i = \{1..n\}$ are exponentially distributed, the mean response time of composite Web service S is given by:

$$E(T^{sequence}) = \sum_{i=1}^{n} E(T_i) \quad (2)$$

Proof: The execution time of composite Web service S composed by n elementary Web services is given by: $T^{sequence} = \sum_{i=1}^{n} T_i$ which is easier to derive from equation (2). ■

Case of homogeneous servers. In the case where $T_i, i \in \{1, ..., n\}$ are random variables with Pareto distributions with parameters (α, k) for each T_i, the mean response time of composite Web service S is trivial and is given by:

$$E(T_{par}^{sequence}) = n\frac{k\alpha}{\alpha - 1}$$

Case of heterogenous servers. As we notice before, we overcome the limitation of other studies by considering that the servers are heterogeneous. Thus, we consider that the execution time of k elementary services s_i follow a Pareto distribution with rate (α_1, k_1) and the execution time of $n-k$ services follow a Pareto distribution with rate (α_2, k_2). Thus, the response time for a composite Web service S is given by:

$$E(T_{par}^{sequence}) = \frac{k_1\alpha_1}{\alpha_1 - 1}k + \frac{k_2\alpha_2}{\alpha_2 - 1}(n-k)$$

B. *Computation for the* **flow** *constructor*

One the most important benefits of the component approach is the interoperability. This inherent interoperability that comes with using vendor, platform, and language independent XML technologies and the ubiquitous HTTP as a transport mean that any application can communicate with any other application using Web services. Thus, the client only requires the WSDL definition to exchange message with the service. However, in the WSDL language, the elementary Web services are conceptually limited to relatively simple operations. In fact, for certains types of applications it is necessary to combine a set of elementary Web services into composite Web services. These services are generally invoked in parallel, using the flow constructor. Thus, in this section, we are focused on the mean response time of a composite Web service S which is composed by n elementary services invoked in parallel. In [2], the author give an analytical formula for the response time of flow constructor but he supposes that n is fixed and elementary Web services are exponential service time distribution. Our contribution is to consider that n is random and Web services are heterogenous.

In the following, we give an analytical expression for the mean response time:

$$E(T^{flow}) = \sum_{i=1}^{n} \int_0^{\infty} t f_i(t) \prod_{j\neq i}^{n} F_j(t) dt \quad (3)$$

where:

$$T^{flow} = Max\{T_i, i = \overline{1, n}\}$$

As we assume that the random variables T_i are independents, the cumulative function of random variable T^{flow} is given by:

$$F(T^{flow}) = P(T^{flow} \leq t) = \prod_{i=1}^{n} F_i(t)$$

Thus the probability density of T^{flow} is:

$$f_{T^{flow}}(t) = \sum_{i=1}^{n} f_i(t) \prod_{j\neq i}^{n} F_j(t) \quad (4)$$

Thus $E(T^{flow})$ can be easily derived.

Case of Pareto distributions. We give in the following the mean response time analytical formula where the random variables T_i, $i \in \{1, ..., n\}$ are Pareto distributed with parameters $(\alpha_i, k_i), i \in \{1, ..., n\}$.

$$E(T_{par}^{flow}) = \sum_{i=1}^{n} \alpha_i k_i^{\alpha_i} \sum_{X\in\mathcal{P}(E_n\setminus\{i\})} (-1)^X \frac{\beta^{-(\sum_{j\in X}\alpha_j + \alpha_i - 1)}}{\sum_{j\in X}\alpha_j + \alpha_i - 1} \prod_{j\in X} \alpha_j k_j \quad (5)$$

Where:

$$\beta = \max(k_i, i \in \{1...n\}) \quad and \quad E_n = \{1, ..., n\}$$

$$and\ \mathcal{P}(E_n \setminus \{i\})\ the\ sub-set\ of\ E_n\ without\ \{i\}.$$

Proof: From equation 4, the probability density of random variable T_{par}^{flow} is given by:

$$f_{T_{par}^{flow}}(t) = \begin{cases} 0 & if\ t \leq max\{k_i, i = 1...n\} \\ \sum_{i=1}^{n} \frac{\alpha_i k_i^{\alpha_i}}{t^{\alpha_i+1}} \prod_{j\neq i}^{n} (1 - (\frac{k_i}{t})^{\alpha_i}) & else., \end{cases};$$

As we have:

$$\prod_{j\neq i}^{n} (1 - (\frac{k_i}{t})^{\alpha_i}) = \sum_{X\in\mathcal{P}(E_n\setminus\{i\})} (-1)^{|X|} \prod_{j\in X} \left(\frac{k_j}{t}\right)^{\alpha_j}$$

Thus, the average response time is:

$$E(T_{par}^{flow}) = \sum_{i=1}^{n} \alpha_i k_i^{\alpha_i} \sum_{X\in\mathcal{P}(E_n\setminus\{i\})} (-1)^{|X|} \int_{\beta}^{\infty} t^{-\sum_{j\in X}\alpha_j - \alpha_i} \prod_{j\in X} \alpha_j k_j\ dt$$

As we have:

$$\int_{\beta}^{\infty} t^{-\sum_{j\in X}\alpha_j - \alpha_i}\ dt = \frac{\beta^{-(\sum_{j\in X}\alpha_j + \alpha_i - 1)}}{\sum_{j\in X}\alpha_j + \alpha_i - 1}$$

Thus we obtain that the mean response time for a composite Web service S is given by the following formula:

$$E(T_{par}^{flow}) = \sum_{i=1}^{n} \alpha_i k_i^{\alpha_i} \sum_{X\in\mathcal{P}(E_n\setminus\{i\})} (-1)^X \frac{\beta^{-(\sum_{j\in X}\alpha_j+\alpha_i-1)}}{\sum_{j\in X}\alpha_j+\alpha_i-1} \prod_{j\in X}\alpha_j k_j$$

■

Case of homogeneous servers. In the case of all elementary service times are Pareto distributed with same rates $(\alpha_i, k_i) = (\alpha, k)$ (i.e $\forall i \in \{1, ..., n\}$), $\alpha_i = \alpha$, $k_i = k$). In this case the response time for S is given by:

$$E(T_{par}^{flow}) = n\alpha k^{\alpha} \sum_{m=0}^{n-1} (-1)^m \frac{k^{(-(m+1)\alpha-1)}(k^{\alpha})^m}{(m+1)\alpha-1} C_{n-1}^{m} \tag{6}$$

Where:

$$C_{n-1}^{m} = \frac{(n-1)!}{m!(n-1-m)!}$$

Case of heterogeneous servers. In the case where $n-k$ elementary service times follow a Pareto distribution with parameters α_1, k_1 and k elementary service times follow a Pareto distribution with rates α_2, k_2. Let factor g which is the slowdown factor such that $\frac{k_2\alpha_2}{1-\alpha_2} = (\frac{k_1\alpha_1}{1-\alpha_1})g$. With these assumptions, the response time of S is as follows:

$$E(T_{par}^{flow}) = R_1 + R_2 \tag{7}$$

$$\begin{cases} R_1 = (n-k)\alpha_1 k_1^{\alpha_1} \sum_{m=0}^{n-1}\sum_{j=0}^{m} \frac{(-1)^m k_1^{-((j+1)\alpha_1+(m-j)\alpha_2-1)}}{((j+1)\alpha_1+(m-j)\alpha_2-1)} \\ \\ R_2 = k\alpha_2 k_2^{\alpha_2} \sum_{m=0}^{n-1}\sum_{j=0}^{m} \frac{(-1)^m k_2^{-(j\alpha_1+(m-j+1)\alpha_2-1)}}{(j\alpha_1+(m-j+1)\alpha_2-1)} \end{cases}$$

This equation (7) is easily derived by the equation (5) by considering that $(\alpha_i, k_i) = (\alpha_1, k_1)$, $\forall i \in \{1, ..., n-k\}$ and $(\alpha_i, k_i) = (\alpha_2, k_2)$, $\forall i \in \{n-k+1, ..., n\}$.

C. Computation for the **switch** *constructor*

In this case, we consider that we have one choice of n elementary Web services. Let $P(Y = i)$ the invocation probability of elementary Web service i, with $\sum_{i=1}^{n} P(Y = i) = 1$. The response time of *switch* constructor is then given by the following analytic formula:

$$E(T^{switch}) = \sum_{i=1}^{n} P(Y = i)E(T_i) \tag{8}$$

with $E(T_i)$ the mean response time of service i.

Proof: First we calculate the probability density of the random variable T^{switch}. The cumulative distribution function of the variable T^{switch} is defined as: $F_{T^{switch}}(t) = P(T^{switch} \leq t)$. According to the total probability theorem, we have:

$$F_{T^{switch}}(t) = \sum_{i=1}^{n} P(T^{switch} \leq t \mid Y = i)P(Y = i)$$

Thus, probability density function of random variable T^{switch} is given by:

$$f_{T^{switch}}(t) = \sum_{i=1}^{n} f_{T_i}(t)P(Y = i)$$

The definition of the average of T^{switch} allow to deduce the result given in equation (8). ■

Case of Pareto distribution. As in this paper, we consider the case of exponential distribution time for each elementary service time, thus the formula for mean response time is given by:

$$E(T_{par}^{switch}) = \sum_{i=1}^{n} \frac{\alpha_i k_i}{\alpha_i - 1} P(Y = i) \tag{9}$$

Case of heterogeneous servers. As well as in the case of the previous presented constructor, we give in the following the response time for the case that the execution times of elementary services are not the same:

$$E(T_{par}^{switch}) = \sum_{i=1}^{n-k} P(Y = i)\frac{\alpha_1 k_1}{\alpha_1 - 1} + \sum_{i=n-k+1}^{n} P(Y = i)\frac{\alpha_2 k_2}{\alpha_2 - 1} \tag{10}$$

In the next section, we are interested to multi-choice pattern which is not supported directly by BPEL, but it can be implemented using the links controls inherited from WSFL.

V. Computation for the **multi-choice** pattern

The difference with the previous pattern where only one Web service is chosen, the multi-choice pattern allows the invocation of a subset of elementary services among the n possible. Take for example the case of a booking flights operated as follows: Web services invoked depend on two criteria namely the city of departure and destination. Next, according to these cities, agencies providing this trip are invoked on parallel. The number of services, and relied on is random. Let N the random variable for the number of invoked services and $P(N = i)$ the probability that the number of invoked service is equal to i, with n maximum number of the invoked services. In this case, the response time of composite web service S is given by the following formula:

$$E(T^{multichoice}) = \sum_{i=1}^{n} [P(N = i)E(T_{S^i})] \tag{11}$$

Where $E(T_{S^i})$ is the mean response time for composite Web service S when i elementary services are invoked.

Proof: First, we give the cumulative function $F_{T^{multichoice}}(t)$ of random variable $T^{multichoice}$. $F_{T^{multichoice}}(t) = P(T^{multichoice} \leq t)$. From totaly probability theorem, we can obtain:

$$F_{T^{multichoice}}(t) = P(\bigcup_{i=1}^{n} \{P(T^{multichoice} \leq t) \wedge N = i\})$$

The events ($N = i,\ i \in \{1, ..., n\}$) are incompatible, so:

$$F_{T^{multichoice}}(t) = \sum_{i=1}^{n} P(T^{multichoice} \le t \wedge N = i)$$

thus,

$$F_{T^{multichoice}}(t) = \sum_{i=1}^{n} P(T^{multichoice} \le t \mid N = i) P(N = i)$$

So:

$$F_{T^{multichoice}}(t) = \sum_{i=1}^{n} P(T_{S^i} \le t) P(N = i)$$

The cumulative function of $T^{multichoice}$ is:

$$F_{T^{multichoice}}(t) = \sum_{i=1}^{n} F_{T_{S^i}}(t) P(N = i)$$

We can derive the probability density $f_{T^{multichoice}}$ of $T^{multichoice}$ and we obtain:

$$f_{T^{multichoice}}(t) = \sum_{i=1}^{n} f_{T^{S^i}} P(N = i)$$

■

Case of homogenous servers. As, we consider the case that the elementary service execution times are Pareto distributed with (α, k) parameters and the invocation probability of elementary service s_i is p, thus the mean response time for composite Web service S can be easily derived from equation (11) and is given as follows:

$$E(T_{par}^{multichoice}) = \frac{n}{\lambda} \sum_{i=1}^{n} C_n^i p^i (1-p)^{n-i} \gamma(i) \qquad (12)$$

Where:

$$\gamma(i) = i\alpha k^{\alpha} \sum_{m=0}^{i-1} (-1)^m \frac{k^{(-(m+1)\alpha - 1)} (k^{\alpha})^m}{(m+1)\alpha - 1} C_{i-1}^m$$

Case of heterogeneous servers. We give also the analytical formula for composite Web service response time where we consider two classes of elementary services. The execution time in each class is the same. N^1 (resp. N^2) is the random variable which defined the number of elementary services in class 1 (resp. class 2). The mean response time formula is also derived from equation (11) and is given by:

$$E(T_{par}^{multichoice}) =$$

$$\sum_{i=1}^{n} P(N^1 = i) \sum_{j=0}^{k} E(T^{multichoise}(i, j)) P(N^2 = j \mid N^1 = i) \qquad (13)$$

VI. EXPERIMENTAL RESULTS AND DISCUSSIONS

In this section, we present some numerical computation and results that we have obtained. When two class of services are considered, let first define a heterogenous coefficient noted g, such as $\frac{k_2\alpha_2}{1-\alpha_2} = g\frac{k_1\alpha_1}{1-\alpha_1}$ (the mean response time of elementary Web services belong respectively to class one and two). It is clear that if $g = 1$, then all of elementary Web services belong to the same class (i.e. the elementary Web services are homogeneous). However, if $g > 1$ means that Web services belong to the second class are slower than services belong to the first class. For simplicity, we assume that the probability of elementary Web services invocation is p for all services.
The synchronization time, when $g = 1$, is the same for any value for the number of elementary Web services belong to the second class denoted N^2. In figure 1, we give the response times by varying the slowdown factor g and where we consider different values of the number of elementary services for second class which takes these values $N^2 = 20$, $N^2 = 60$, $N^2 = 80$ and $N^2 = 100$. In figure 2, we give the response times by varying the the number of elementary services for second class and we consider the case of $g = 2$, $g = 3$, $g = 4$ and $g = 5$. From figure 1, we can conclude two things. First, for any value of N^2, the synchronization response time increases linearly with the heterogeneous coefficient g. Second, when $g = 1$ the response time of the composite Web service is the same for any value of the elementary Web services belong to the second class. From figure 2, we

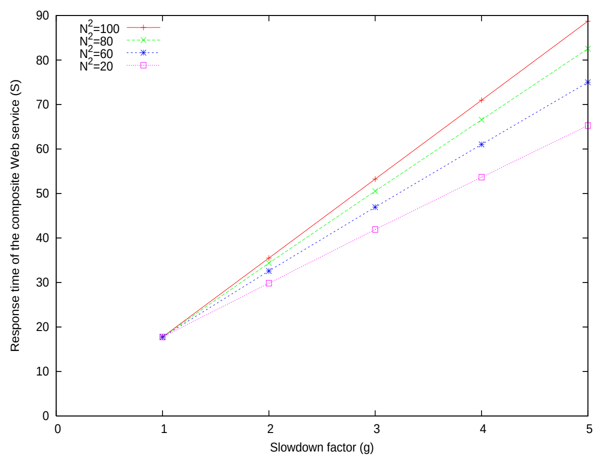

Fig. 1. Response times for composite Web service versus slowdown factor g

can notice that the waiting time increase logarithmicaly with invocation probability p. It is clear that the response time increases logarithmicaly with the number of invoked Web services (see figure 3). So, we can conclude that the choice of elementary Web services must be made on their physical characteristics and not on their number.

In the figure 4, we shown the evolution of $\frac{T_{exp}}{T_{par}}$, where T_{exp} and T_{par} is the response time of a composite Web services when respectively the response time of elementary Web services is exponential and heavy-tailed. The results

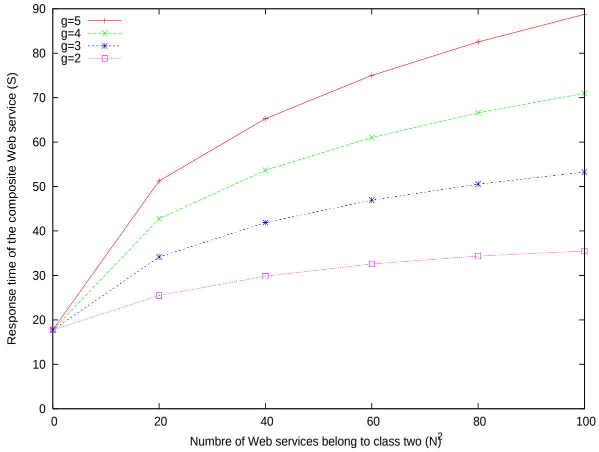

Fig. 2. Response time for composite Web service versus slowdowns Web services

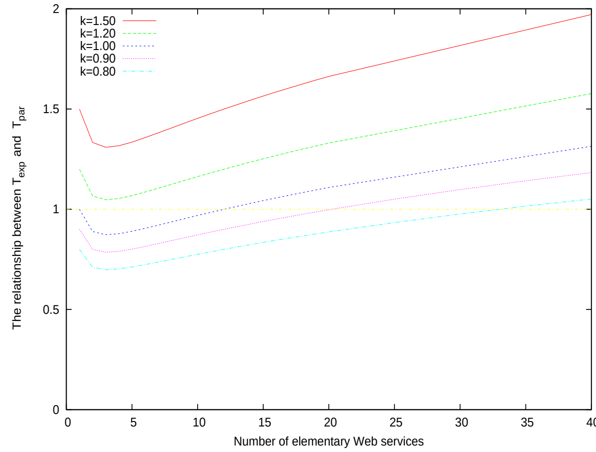

Fig. 4. Application slowdown factor $\frac{T_{exp}}{T_{par}}$ versus Web services number

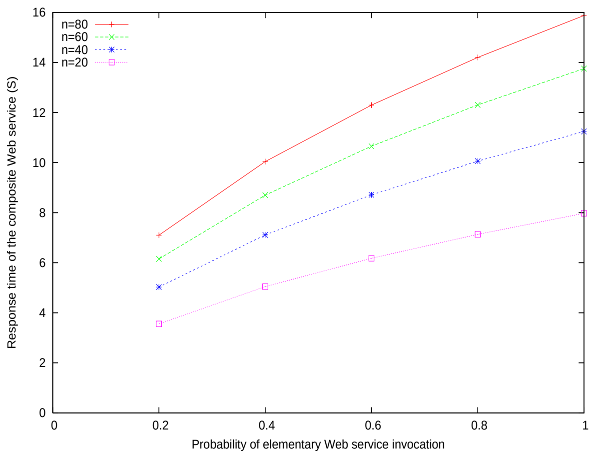

Fig. 3. Response time for composite Web service versus probability invocation

shown in this figure, for different values of elementary Web services response time, reveals that the choice conditions of elementary Web services must be more restrictive in the case of exponential, when their number is great.

VII. Conclusion

Web Services are based on a set of standards and protocols, that allow us to make processing requests to remote systems by exchanging with a common language, and using common transport protocols. Once deployed, Web services provided can be combined (or inter-connected) in order to implement business collaborations, leading to composite web services. With the proliferation of Web Services as a business solution to enterprise application integration, the quality of service offered by Web Services is becoming the utmost priority for service provider and their partners. The QoS is defined as a combination of the different attributes of the Web services such as availability, response time, throughput, etc. In this paper, we have focused in the response time of composite Web services. We have proposed analytical formulas for the mean response of the different control patterns supported by BPEL standards. In this paper, we have studied the Pareto distribution. It is justified by the fact that experimental studies shown that Web services response time is typically heavy-tailed. However, the methodology can be applied to other service response time distributions.

We plan to consider the dynamic composition of Web services and we will give the analytical formulas for BPEL constructors as a perspective study.

References

[1] S. Haddad, L. Mokdad, S. Youcef , ”Response-time analysis of composite Web services”, In Proceedings , 2008. CSNDSP 2008, Communication Systems, Networks and Digital Signal Processing, IEEE Computer Society, 23-25 July 2008, Graz University of Technology, Austria

[2] D.A. Menascé, ”Response-time analysis of composite Web services”, IEEE Internet computing, vol. 8, No. 1, pp. 90-92, 2004

[3] M. Sharf, ”On the response time of the large-scale composite Web services”, Proceedings of the 19th International Teletraffic Congress (ITC 19), Beijing, 2005

[4] M. Crovella and A. Bestavros, ”Self-similarity in world-wide traffic : Evidence and possible causes”, on Networking, vol. 5, pp. 835846, December 1997

[5] M. E. Crovella and A. Bestavros, Self-similarity in World Wide Web traffic: Evidence and possible causes, IEEE/ACM Trans. Netw., vol. 5, no. 6, pp. 835846, 1997

[6] Z. Sahinoglu and S. Tekinay, ”On multimedia networks: Self-similar traffic and network performance”, IEEE Communications Magazine, January 1999

[7] S. Hahn and L. Fatto, ”Two parallel queues created by arrivals with two demands”, Applied Mathematics, Vo. 44, pp. 1041-1053, October, 1984

[8] R. Nelson and A.N. Tantawi, ”Approximate Analysis of Fork/Join Synchronization in Parallel Queues”, IEEE Transaction Computer, vol. 37, No. 6, pp. 739-743, 1998

[9] F. Bacelli, A.M. Makowski and A. Shwartz, ”The fork and join Related Systems with Synchropnization constraints: Stochastic Ordering and Computable Bounds”, Applied Probability, pp. 629-660, July, 1983

[10] F. Bacelli and A.M. Makowski, ”Simple computable bounds for the fork-join qeue”, Proceeding Information Scinece, pp. 436-441, March, 1985

[11] G. Kiczales, ”Aspect-oriented programming”, ACM Comput. Surv., pp. 154, http://doi.acm.org/10.1145/242224.242420, 1996

[12] W.M.P. van der Aalst, A.H.M. ter Hofstede, B. Kiepuszewski, and A.P. Barros, ”Workow patterns”, Technical report FIT-TR-2002-2, Faculty of IT, Queensland University of Technology, July 2002. Accessed from http://www.tm.tue.nl/it/

[13] M.U.Bhatti, S.Youcef, L.Mokdad and V.Menfort, ”Simulation-based Response Time Analysis of composite Web Services”, In Proceedings 10th IEEE international Multitopic conference, INMIC, pp. 1-7, juin 2006

[14] F. Curbera, ”Business Process Execution Language for Web Services”, Version 1.0, IBM, 2002

[15] U. Vallamsetty, K. Kant, and P. Mohapatra, Characterization of e-commerce traffic, Electronic Commerce Research, vol. 3, no. 1-2, pp. 167192, 2003

[16] D. Davis and M. P. Parashar, ”Latency Performance of SOAP Implementations”, IEEE Computer Society, pp. 407-415, 2002

[17] S. Paul, P.G. Santiago, K. Kobuske, H. Marc and P. Eduardo, ”Fast Web Services, Web site: http://wwww.java.sun.com/developper/technicalArticles/WebServices/fastWS/”, 2006

[18] C. Kohlhoff and R. Steele, ”Evaluating SOAP for High Performance Business Applications: Real-Time Trading Systems”, In Proceedings of WWW’03, Budapest, Hungry, 2003

[19] F. Curbera and M. Duftler, R. Khalaf, W. Nagy, N. Mukhi and Sanjiva Weerawarana, ”Unraveling the Web Services Web: An Introduction to SOAP, WSDL, and UDDI”, IEEE Internet Computing, vol. 6, No. 2, pp. 86-93, 2006

[20] ”Web Services Description Language, WSDL”, Web site at: http://www.w3.org/TR/wsdl, http://www.w3.org/TR/wsdl, 2007

[21] ”The SOAP specification, which is published (and endorsed) by the World Wide Web Consortium, is available on-line at http://www.w3.org/TR/SOAP/”, http://www.w3.org/TR/wsdl, 2007

[22] Heiko Ludwig, ”Web Services QoS: External SLAs and Internal Policies - Or: How do we deliver what we promise”, citeseer.ist.psu.edu/699287.html

[23] Daniel A. Menasce, ”QoS Issues in Web Services, IEEE Internet Computing, Vol. 6, No. 6, 2002, issn. 1089-7801, pp. 72-75, http://dx.doi.org/10.1109/MIC.2002.1067740, IEEE Educational Activities Department, Piscataway, NJ, USA

[24] Anbazhagan Mani and Arun Nagarajan,”Understanding quality of service for Web services”, http://www-106.ibm.com/developerworks/library/ws-quality.html, 2002

[25] D. A. Menasce and V. Almeida, ”Capacity Planning for Web Services: metrics, models, and methods”, 2001

[26] Optimal Service Selection Heuristics in Service Oriented Architectures, (with E. Casalicchio, V. Dubey, and L. Silvestri), The 3rd International Workshop on Advanced Architectures and Algorithms for Internet DElivery and Applications (AAA-IDEA 2009), Las Palmas de Gran Canaria, The Canary Islands, Spain, November 26, 2009.