Journal of Computational Science 4 (2013) 232–241

Contents lists available at SciVerse ScienceDirect

Journal of Computational Science

journal homepage: www.elsevier.com/locate/jocs

Bounding models families for performance evaluation in composite Web services

Serge Haddad [a], Lynda Mokdad [b,*], Samir Youcef [c]

[a] LSV, ENS de Cachan, Cachan, France
[b] LACL, University of Paris-Est, Créteil, France
[c] LORIA-INRIA-UMR 7503, Nancy, France

ARTICLE INFO

Article history:
Received 31 July 2011
Received in revised form 14 October 2011
Accepted 24 November 2011
Available online 10 January 2012

Keywords:
Composite Web services
Continuous time Markov chains
Stochastic coupling

ABSTRACT

One challenge of composite Web service architectures is the guarantee of the Quality of Service (QoS). Performance evaluation of these architectures is essential but complex due to synchronizations inside the orchestration of services. We propose methods to automatically derive from the original model a family of bounding models for the composite Web response time. These models allow to find the appropriate trade-off between accuracy of the bounds and the computational complexity. The numerical results show the interest of our approach w.r.t. complexity and accuracy of the response time bounds.

1. Introduction

A Web service essentially denotes an application (or program) accessible via Internet standard protocols [3]. The basic protocol used to access Web services is SOAP (*Simple Object Access Protocol*), an XML (*eXtensible Markup Language*) based protocol that allows a service customer to invoke services [6]. The Web services, called elementary Web services, such as described by WSDL (*Web Service Description Language*), are conceptually limited to relatively simple functionalities modeled through a collection of simple operations without control flow. For certain types of applications, it is necessary to combine a set of elementary Web services to obtain more complex one, called aggregated or composite Web services, in order to meet customer requirements [4,1]. This aggregation is possible using for example BPEL (*Business Process Execution Language For Web Services*) standard which is the result of merging previous composition languages such like WSFL (*Web Service Flow Language*) and XLANG (*XML Business Process Language*) [5].

The Web services are executable on more platforms than the previous technologies such like CORBA (*Common Object Request Broker Architecture*) and RMI (*Remote Method Invocation*), however they raise new requirements like the dynamic adaptability and the insurance of QoS (*Quality of Service*). The latter criterium is defined as a combination of several attributes which can be qualitative (e.g. security) and quantitative (e.g. response time [9,10]). Their increasing complexity requires the development of methods and tools in order to monitor and evaluate their QoS. In fact, the QoS degradation can lead to serious consequences including a significant economic impact.

In this paper, we focus on the composite Web service (CWS) response time computation, where the requests are decomposed into sub-queries to different elementary Web services and then merged into a final result. In our previous study [11], we have considered the BPEL constructors directly supported by this standard. The control patterns considered here are not directly supported by BPEL:

- parallel invocation of a constant number of elementary Web services merged by a federation component (see Fig. 1),
- parallel invocation of a variable number of elementary Web services merged by a federation component.

Under the assumption of Markovian elementary service and merging times, the modeling of a composite Web service yields a Markov chain with $O(n^2)$ states, where n is the number of invoked elementary Web services. The particular structure of this Markov chain allows us to establish recurrence equations which lead to a computational complexity time of order $O(n^2)$. It is because of the structure of the obtained Markov chain (see Section 4). In the open systems such as peer to peer environment, the number of invoked services can lie between 10^3 and 10^6. So, their exact analysis becomes difficult and often intractable. Using the stochastic comparison [7,8] and more precisely the coupling process technique, we propose a generic transformation of the studied Markov chain which guarantees that the response time of the new Markov chain is an upper bound of the initial Markov chain response time. We

* Corresponding author.
E-mail address: lynda.mokdad@univ-paris12.fr (L. Mokdad).

1877-7503/$ – see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.jocs.2011.11.003

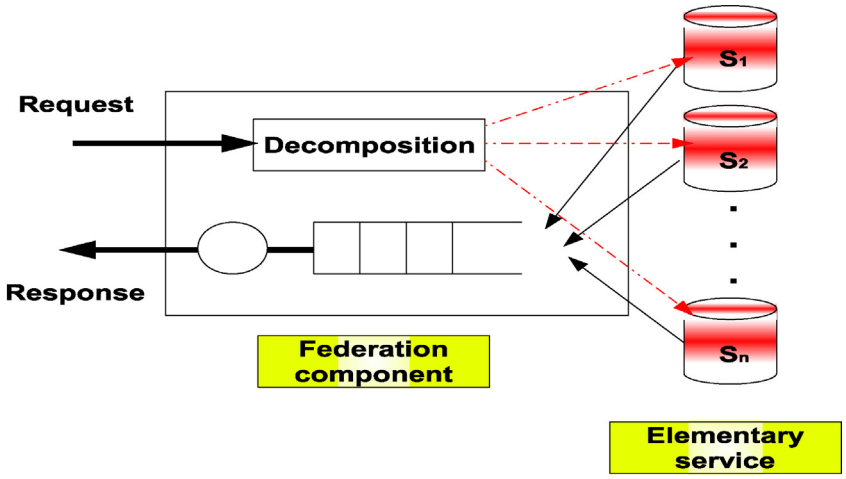

Fig. 1. Data processing in a composite Web service platform.

instantiate this transformation in three ways, where each obtained new Markov chain is parameterized by a "quantitative" parameter. Otherwise stated, we propose three families of the bounding models. By an appropriate choice of the parameter, the recurrence equation systems can be resolved with an algorithm with $O(n)$ and $O(n\sqrt{n})$ respectively space complexity and time complexity.

Moreover, we show by empirical studies that depending on the numerical values of the original Markov chain, the bound provided by any of the three bounding families can be better than the two other ones. We also characterize the three cases w.r.t. these numerical values.

We generalize our work as follows. Assume that an elementary Web service can be invoked with a constant probability. Thus the response time of a given composite Web service can be computed as a weighted sum of the elementary Web services response times where the computational cost is of order $O(n^3)$. To handle this case, we combine the upper bounds proposed for the first pattern and the Chernoff bounds in order to limit the study only to two cases: the case where all services are invoked and a probabilistic "worst case" i.e. a constant number of invoked services with a very small probability to exceed this threshold. This approach allows us to keep the same computational cost as in the first case (i.e. when the number of elementary Web services is constant). Note that Chernoff bounds give bounds on the tail distributions of the sums of independent random variables (more details are given in Section 3).

The rest of the paper is organized as follows. Section 2 presents some related works. Section 3 recalls some definitions and results related to the process coupling technique and the Chernoff bounds. In Section 4, we study the control pattern parallel invocation where the number of elementary Web services is constant. Section 5 summarizes the obtained numerical results in this case. In Section 6, we study the control pattern parallel invocation where the number of elementary Web services is variable. Section 7 summarizes the obtained numerical results in this case. Section 8 summarizes the contributions of this paper and gives some perspectives to this work.

2. Related work

In the framework of Web services performance evaluation, two approaches are generally used: benchmarking and modeling methods. In the following, we present some studies using the two approaches.

As far as performance measurement of Web services is concerned, XML specification and SOAP protocol have been studied in [21–23] by testing and measuring of SOAP-based Web services response time. A comparative study on response time and throughput with existing protocols, like RMI, RMI/IIOP or CORBA/IIOP, is presented in [21]. A critical study of XML-based protocols for Web services is presented and binary encoded protocol has been proposed instead of text XML-based ones in [22]. In [24], information about past workflow executions is collected in a log. Starting from this log a continuous Markov chain is derived, in order to compute the execution response time and the cost of this workflow.

In [10], the composite Web service response time is considered as a response time of fork and join model. This model states that a single Internet application can invoke in parallel a set of elementary Web services and gather their responses from all these launched services in order to return the results to a client. In this considered study, authors analyze the effects of exponential response times based on earlier work in [12]. An exact analysis of fork and join system is possible when the system is significantly simplified. This is the case for example when the job arrival process in the system follows a Poisson distribution with execution task having exponential distribution and the number of queues is equal to two. The exact computation response time of a such system can be found in [13–15]. An approximation technique has been proposed in the case where the number of servers is greater than two and the servers are homogeneous [15]. This last study is extended in [16]. General arrival process and services times are considered in [17]. The most general case is considered in [18]. In this work, upper and lower bounds are proposed by assuming that the response times in each queue are mutually independent. Two approximation techniques are presented: one is based on a decomposition approach and the other is based on an iterative solution method.

In order to overcome the limitations of these studies and particularly the one presented in [10], we have proposed a general model taking into account the fact that elementary Web services are heterogenous and the number of invoked services can be variable (this is the case when we use for example the BPEL multi-choice constructor) [11]. More recently, the problem of computing the distribution of the throughput time in workflow nets has been studied in [20]. In this paper, authors consider workflow with transition execution time having exponential distributions and formulas have been proposed for each refinement rule (sequence, parallel, synchronization and loop execution pattern). Response time of a Web service middleware is considered in [19], which follows a fork and join model of execution. The author proposes that while performing

a join operation, servers with slow response times can be eliminated to maximize the performance. The work is more oriented toward studying fork and join model in order to understand how to optimally merge the results from various servers.

3. Coupling of Markov chains and Chernoff bounds

A continuous time Markov chain (CTMC) $\mathcal{M}$ is defined by:

- a finite space state S,
- a real matrix $Q : S \times S \longrightarrow \mathbb{R}$ called infinitesimal generator such that:
  1. $\forall s \neq s' \in S, \quad Q[s, s'] \geq 0$,
  2. $\forall s \in S, \quad \sum_{s' \in S} Q[s, s'] = 0$
- and an initial distribution denoted $\mathcal{M}(0)$.

A coupling of two Markov chains is a "product" chain where the set of states is a subset of the product of the initial sets and such that by only observing the behavior of a component of the state, one obtains the corresponding chain.

Definition 1. Let $\mathcal{M} = (S, Q, \mathcal{M}(0))$ and $\mathcal{M}' = (S', Q', \mathcal{M}(0))$ be two CTMCs. Then a coupling of $\mathcal{M}$ and $\mathcal{M}'$ is a CTMC $\mathcal{M}^* = (S^*, Q^*, \mathcal{M}^*(0))$ such that:

- $S^* \subseteq S \times S'$
- $\forall s \in S, \quad \mathcal{M}(0)[s] = \sum_{(s,s') \in S^*} \mathcal{M}^*(0)[s, s']$
- $\forall s' \in S', \quad \mathcal{M}'(0)[s'] = \sum_{(s,s') \in S^*} \mathcal{M}^*(0)[s, s']$
- $\forall s \neq s_1 \in S, \forall (s, s') \in S^*, Q[s, s_1] = \sum_{(s,s'),(s_1,s'_1) \in S^*} Q^*[(s, s'), (s_1, s'_1)]$
- $\forall s' \neq s'_1 \in S', \forall (s, s') \in S^*, \quad Q'[s', s'_1] = \sum_{(s,s'),(s_1,s'_1) \in S^*} Q^*[(s, s'), (s_1, s'_1)]$

The coupling technique is related to the properties of the state space S^*. The following proposition whose proof is a direct consequence of Definition 1, is sufficient for our purposes.

Proposition 1. *Let $\mathcal{M}^*$ a coupling of $\mathcal{M}$ and $\mathcal{M}'$.*
Let $s_f \in S$ and $s'_f \in S'$ such that:

$$\forall (s, s') \in S^*, \quad s' = s'_f \Rightarrow s = s_f$$

We denote $T_{\mathcal{M}}(s_f)$ (resp. $T_{\mathcal{M}'}(s'_f)$) the expected first passage time between an initial state s_i and a final state s_f (resp. s'_f) in $\mathcal{M}$ (resp. $\mathcal{M}'$). Then:

$$T_{\mathcal{M}}(s_f) \leq T_{\mathcal{M}'}(s'_f)$$

The Chernoff bounds provide an accurate bound of the deviation probability of a random variable, defined as a sum of independent random variables and taking their values in $\{0, 1\}$.

Proposition 2 (Chernoff bounds). *Let $X_1, \ldots, X_n$ be independent random variables with values in $\{0, 1\}$ such that $P(X_i = 1) = p_i$, $X \equiv \sum_{i \leq n} X_i$, $\mu \equiv E(X)$ and $0 \leq \delta < 1$, then:*

$$P(X \geq (1 + \delta)\mu) \leq e^{-\mu\delta^2/3} \quad \text{and} \quad P(X \leq (1 - \delta)\mu) \leq e^{-\mu\delta^2/2}$$

4. Study of the parallel invocation pattern and constant number of elementary Web services

4.1. Specification of the problem

We consider a distributed application where the data is stored in databases and can be accessed by XML-based protocols. When a composite Web service is started, it is decomposed into elementary Web services allocated to servers $s_1, s_2, \ldots, s_n$ while *in parallel* with the still working servers, the partial responses are integrated into a single result (merge) which is the response to the client. The system under study in this paper is given in Fig. 1.

We assume, in the rest of this paper, that the response time of the servers $s_i (i = 1, \ldots, n)$ are independent and identically distributed following an exponential distribution with mean $1/\lambda$. The merging time is also exponential random variable with mean $1/\mu$. n denotes the number of elementary Web services that are invoked.

For completeness, we distinguish two cases of a composites Web services:

- **Simple case.** The federation component starts the merging after all of the elementary Web service results have been received.
- **Main case.** The federation component proceeds (sequentially) to the merging on receipt of elementary Web service results.

Observe that the response time of the simple case is an upper bound to the response time of the main case.

In the simple case, the average response time of such composite Web service is given by:

$$E(T^n_{fix}) = E(T^n_{syn}) + n\mu \tag{1}$$

where the average response time $E(T^n_{syn})$ of the local servers is given by Lemma 1.

Lemma 1. *If the elementary Web services are homogenous and their service times are exponentially distributed with rate λ, the overall local servers response time is given by:*

$$E(T^n_{syn}) = \frac{1}{\lambda} \sum_{i=1}^{n} \frac{1}{i}$$

Proof. The average time for a first elementary Web service to handle its request is $1/n\lambda$, the average time for the two first elementary Web services to complete their requests is $(1/n\lambda) + (1/(n-1)\lambda)$, and so forth. By iterating, we obtain:

$$E(T^n_{syn}) = \frac{1}{n\lambda} + \frac{1}{(n-1)\lambda} + \cdots + \frac{1}{2\lambda} + \frac{1}{\lambda} = \frac{1}{\lambda} \sum_{i=1}^{n} \frac{1}{i}$$

□

In the main case, the considered model can be described by a continuous Markov chain $\mathcal{M}(t)$. To describe this chain, we define the state $s_{i,j}$ where i indicates the responses that are queued in the federation component and j the elementary Web services that are still to respond. The transition graph of the Markov chain for $n = 6$ is given in Fig. 2.

Observe that the average response time of the composite Web service is the average absorption time by the state $s_{0,0}$ starting from the state $s_{0,n}$, noted $T(s_{0,n})$. It is given by:

$$T(s_{i,j}) = Soj(s_{i,j}) + \sum_{s_{i,j} \longrightarrow s_{i',j'}} P[s_{i,j}, s_{i',j'}] T(s_{i',j'})$$

where $Soj(s_{i,j})$ is the sojourn time in the state $s_{i,j}$ and $P[s_{i,j}, s_{i',j'}]$ the probability transition from the state $s_{i,j}$ to the state $s_{i',j'}$, deduced from the infinitesimal generator Q.

Thus, the response time of such composite Web services can be computed with a $O(n^2)$ time complexity (i.e. the number of equations) and with a $O(n)$ complexity space (because the terms $T(s_{.,j})$ are enough to compute the terms $T(s_{.,j+1})$).

4.2. General bounding models

In this section, we propose bounding models and prove that the response time in these models provides an upper bound of the

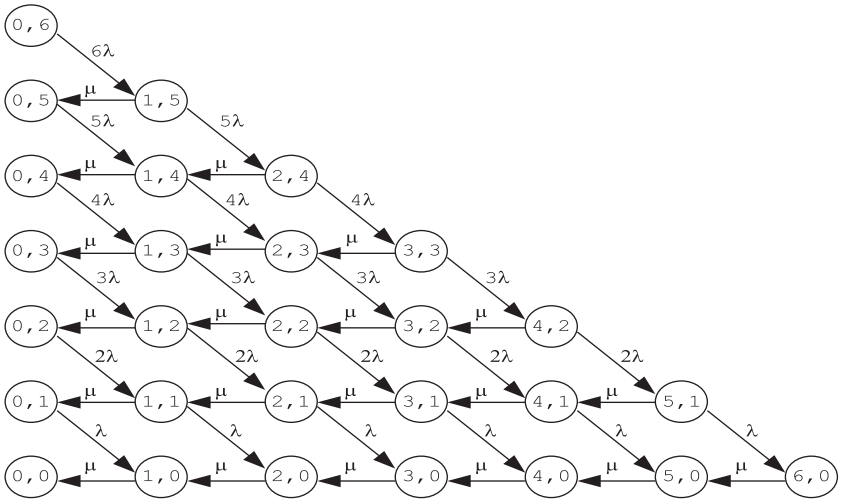

Fig. 2. Markov chain for the case $n = 6$.

original model response time. These models are obtained by redirection of some arcs in the original Markov chain. Let $\mathcal{M}$ be the Markov chain corresponding to the original model. We redirect $\mathcal{A}$, a set of arcs of $\mathcal{M}$. $\mathcal{M}_{\mathcal{A}}$ denotes the new Markovian process obtained by this redirection. We distinguish two kinds of redirections:

- redirection of an arc $(s_{i,j}, s_{i+1,j-1})$ (with rate $j\lambda$) which becomes a loop $(s_{i,j}, s_{i,j})$,
- redirection of an arc $(s_{i,j}, s_{i-1,j})$ (with rate μ) which becomes a loop $(s_{i,j}, s_{i,j})$.

We formalize these redirections as follows.

Definition 2. Let $\mathcal{A}$ be a set of arcs. Then function $suiv_\lambda : S \longrightarrow S$ is defined by:

$$\forall s_{i,j} \in S,\ suiv_\lambda(s_{i,j}) = \begin{cases} s_{i,j} \text{ if } (s_{i,j}, s_{i+1_{\{j>0\}},j-1_{\{j>0\}}}) \in \mathcal{A}, \\ s_{i+1_{\{j>0\}},j-1_{\{j>0\}}}, \text{ otherwise.} \end{cases}$$

Function $suiv_\mu : S \longrightarrow S$ is defined by:

$$\forall s_{i,j} \in S,\ suiv_\mu(s_{i,j}) = \begin{cases} (s_i, s_j) \text{ if } (s_{i,j}, s_{i-1_{\{i>0\}},j}) \in \mathcal{A}, \\ s_{i-1_{\{i>0\}},j}, \text{ otherwise.} \end{cases}$$

The following theorem is the basis for the particular family of bounding models described in Section 4.3.

Theorem 1. *Let $\mathcal{M}$ be the Markov chain associated with the initial model and $\mathcal{M}'$ be the Markov chain obtained by redirection of the set of arcs $\mathcal{A}$. Thus, there exists a coupling $\mathcal{M}^*$ of $\mathcal{M}$ and $\mathcal{M}'$ such that the set of states S^* is defined by:*

$$S^* = \{(s_{i,j}, s_{i',j'}) \mid j \le j' \wedge i \le i' + (j' - j)\} \quad (2)$$

Proof. We define the output transitions from $(s_{i,j}, s_{i',j'}) \in S^*$, by distinguishing four cases:

1. $i = i'$ and $j = j'$. The outgoing transitions from state $(s_{i,j}, s_{i,j})$ are:
   (a) if $i > 0$ a transition with rate μ toward state $(s_{i-1,j}, suiv_\mu(s_{i,j}))$
   (b) if $j > 0$ a transition with rate $j\lambda$ toward state $(s_{i+1,j-1}, suiv_\lambda(s_{i,j}))$
   (c) a transition with rate $(n-j)\lambda + 1_{\{i=0\}}\mu$ toward state $(s_{i,j}, s_{i,j})$
2. $j = j'$ and $i < i'$. Thus $i' > 0$. The outgoing transitions from state $(s_{i,j}, s_{i',j'})$ are:
   (a) a transition with rate μ toward state $(s_{i-1,j}, suiv_\mu(s_{i',j}))$ if $i > 0$, toward state $(s_{i,j}, suiv_\mu(s_{i',j}))$ otherwise
   (b) If $j > 0$ a transition with rate $j\lambda$ toward state $(s_{i+1,j-1}, suiv_\lambda(s_{i',j}))$
   (c) a transition with rate $(n-j)\lambda + 1_{\{i=0\}}\mu$ toward state $(s_{i,j}, s_{i',j})$
3. $j < j'$ and $i + j = i' + j'$. Thus $i > 0$ and $j' > 0$. The outgoing transitions from state $(s_{i,j}, s_{i',j'})$ are:
   (a) a transition with rate μ toward state $(s_{i-1,j}, suiv_\mu(s_{i',j'}))$.
   (b) If $j = 0$ a transition with rate $j'\lambda$ toward state $(s_{i,j}, suiv_\lambda(s_{i',j'}))$
   (c) If $j > 0$
      - a transition with rate $j\lambda$ toward state $(s_{i+1,j-1}, suiv_\mu(s_{i',j'}))$
      - and a transition with rate $(j' - j)\lambda$ toward state $(s_{i,j}, suiv_\mu(s_{i',j'}))$
   (d) a transition with rate $(n-j)\lambda + 1_{\{i=0\}}\mu$ toward state $(s_{i,j}, s_{i',j'})$
4. $j < j'$ and $i + j < i' + j'$. Thus $i' > 0$ and $j' > 0$. The outgoing transitions from state $(s_{i,j}, s_{i',j'})$ are:
   (a) a transition with rate μ toward state $(s_{i-1,j}, suiv_\mu(s_{i',j'}))$ if $i > 0$, toward state $(s_{i,j}, suiv_\mu(s_{i',j'}))$ otherwise
   (b) If $j = 0$ a transition with rate $j'\lambda$ toward state $(s_{i,j}, suiv_\lambda(s_{i',j'}))$
   (c) If $j > 0$
      - a transition with rate $j\lambda$ toward state $(s_{i+1,j-1}, suiv_\mu(s_{i',j'}))$
      - and a transition with rate $(j' - j)\lambda$ toward state $(s_{i,j}, suiv_\mu(s_{i',j'}))$
   (d) a transition with rate $(n-j)\lambda + 1_{\{i=0\}}\mu$ toward state $(s_{i,j}, s_{i',j'})$

One checks that $\mathcal{M}^*$ is indeed a coupling by comparing the two marginal rates with the ones of $\mathcal{M}$ and $\mathcal{M}'$. □

The following corollary is an immediate consequence of Theorem 1.

Corollary 1. *Let $(s_{i,j}, s_{i',j'}) \in S^*$, then:*

$$T(s_{i,j}) \le T'(s_{i',j'})$$

And in particular, $T(s_{0,n}) \le T'(s_{0,n})$.

Proof. Observe that for every $(s, s') \in S^*$ $s' = s_{0,0}$ implies $s = s_{0,0}$ and apply Proposition 1. □

Let be $\mathcal{A}_1 \subseteq \mathcal{A}_2$ be two sets of redirected arcs and $\mathcal{M}_{\mathcal{A}_1}$, $\mathcal{M}_{\mathcal{A}_2}$ the two corresponding Markov chains. Theorem 2, whose proof is similar to the one of the Theorem 1 shows that absorbing time obtained by redirection is "increasing" w.r.t. to the (partial) order defined by set inclusion.

Theorem 2. *Let be $\mathcal{M}$ the original Markov chain and $\mathcal{M}_{\mathcal{A}_1}$ (resp. $\mathcal{M}_{\mathcal{A}_2}$) a Markov chain obtained by the redirection of the set of arcs $\mathcal{A}_1$ (resp. $\mathcal{A}_2$) where $\mathcal{A}_1 \subseteq \mathcal{A}_2$. Then, there exists a coupling $\mathcal{M}^*$ of $\mathcal{M}_{\mathcal{A}_1}$ and $\mathcal{M}_{\mathcal{A}_2}$ such that the set of states S^* is defined by:*

$$S^* = \{(s_{i,j}, s_{i',j'}) \mid j \leq j' \wedge i \leq i' + (j' - j)\}$$

4.3. Particular bounding models

We propose, in this section, three particular families of bounding models obtained by a specific choice of the redirected set of arcs. These families correspond to redirections called in the sequel λ-aggregation, initial μ-aggregation and final μ-aggregation. It may seem surprising that we obtain a reduction of the computational complexity, although the space states of the bounded model are the same as that of the initial model. The key point of this reduction lies in the fact that the bounding Markov chains allows us to "merge" many equations into a single one, according to the parameter of aggregation for each particular family of the bounding models.

4.3.1. Upper bound by λ-aggregation

These bounding models are parameterized by an integer m, with $0 \leq m \leq n-2$, called aggregation level. Intuitively, it consists to "stop" the elementary Web services execution when there is at least one request in the federation component until either there is no request or $m+1$ requests are processed. We present below two bounding models obtained by λ-aggregation for $n = 6$ and $m \in \{1, 2\}$ (see Fig. 3).

4.3.2. Upper bound by initial μ-aggregation

These bounding models are parameterized by an integer m, where $0 \leq m \leq n-2$ called aggregation level. It consists to "stop" the component processing while it remains at least m running elementary Web services. We present below two bounding models obtained by λ-aggregation for $n = 6$ and $m \in \{1, 2\}$ cases (see Fig. 4).

4.3.3. Upper bound by final μ-aggregation

These bounding models are parameterized by an integer h called the aggregation level. It consists to "stop" the federation processing while there are more than h running elementary Web services.

5. Numerical results for the "fork merge" pattern

We present now a summary of the numerical results obtained for the bounding models presented above. It is clear that if $\mu \ll \lambda$ (resp. $\mu \gg n\lambda$) the upper bounds obtained by λ-aggregation are better than those obtained by μ-aggregation and vice versa. So, the most interesting case, studied in the following, is the one where $\lambda \leq \mu \leq n\lambda$.

The quantitative behavior of the studied Markov chain is determined by the ratio between μ and $n\lambda$, thus without loss of generality, we fix the parameter λ at 0.1 and we let the federation rate μ vary.

5.1. Comparison between λ-aggregation and initial μ-aggregation

We present the upper bounds for composite Web services where the number of elementary Web services n is equal to 5000, 10,000, 15,000 and 20,000. The aggregation level $m = O(n - \sqrt{n})$. So the number of recurrence equations required to compute the absorption by the state $s_{0,0}$ from the state $s_{0,n}$ is $O(n\sqrt{n})$. Figs. 5 and 6 show the absorption time evolution and the upper bounds obtained by λ-aggregation and initial μ-aggregation. Thus, according to these results, we conclude that:

Table 1
Computational cost vs. accuracy.

Aggregation level (h)	Ratio	Upper bound	Relative error
0	0.02%	390.47	36.84%
100	1.01%	338.62	18.52%
500	4.95%	322.58	12.90%
1000	9.76%	315.66	10.49%
1200	11.65%	313.83	9.85%
1400	13.52%	312.29	9.30%
1600	15.38%	310.96	8.84%
1800	17.21%	309.78	8.43%

- the λ-aggregation models are better than the models with μ-aggregation,
- the best upper bounds are obtained by λ-aggregation models when the federation rate μ is not very large compared to the rate server λ.

5.2. Comparison between λ-aggregation and final μ-aggregation

We present the upper bounds obtained by λ-aggregation and final μ-aggregation for composites Web services where the number of elementary Web services number n is equal to 5000, 10,000, 15,000 and 20,000. Thus, according to the obtained results (see Figs. 7 and 8), we observe that the upper bounds obtained by final μ-aggregation are better that the ones obtained by λ-aggregation when the difference between the federation rate μ and the server rate λ is important.

5.3. Tradeoff between the upper bound quality and the computational cost

We now analyze the tradeoff between computational cost and the accuracy of the upper bounds, in the case where the final μ-aggregation is better than the λ-aggregation. Table 1 shows the tradeoff between the computational cost and the accuracy of the upper bounds with *a relative error* and *a ratio*. The relative error is defined as the absolute error divided by the exact response time value. The ratio is defined as the ratio between the number of recurrence equations when we consider the Markov chain $\mathcal{M}$ and the number of recurrence equations when we consider the Markov chain $\mathcal{M}'$. The value of the parameters are: $n = 20{,}000$, $\lambda = 0.1$ and $\mu = 70$. The exact response time of this composite Web service is 258.70 and the number of recurrence equations is around 2×10^8. We emphasize two facts. On the one hand, the accuracy increases with the level aggregation h. On the other hand, the number of recurrence equations required to compute absorption time increases with the level aggregation h.

6. Study of the parallel invocation pattern and random number of elementary Web services

We consider in this section the case where the invocation of elementary Web services is i.i.d. binary random variables with parameter p. This assumption is reasonable in the framework of Web services with more or less equally loaded servers. As in the previous section, we study the simple case (the merging follows all the answers) and the main case (the merging is performed in parallel with the local servers).

In the simple case, the composite Web service average response time is given by the following equation:

$$E(T_{var}^{n,p}) = \sum_{i=1}^{n} p^i (1-p)^{n-i} \binom{n}{i} \sum_{j=1}^{i} \frac{\lambda}{i} \tag{3}$$

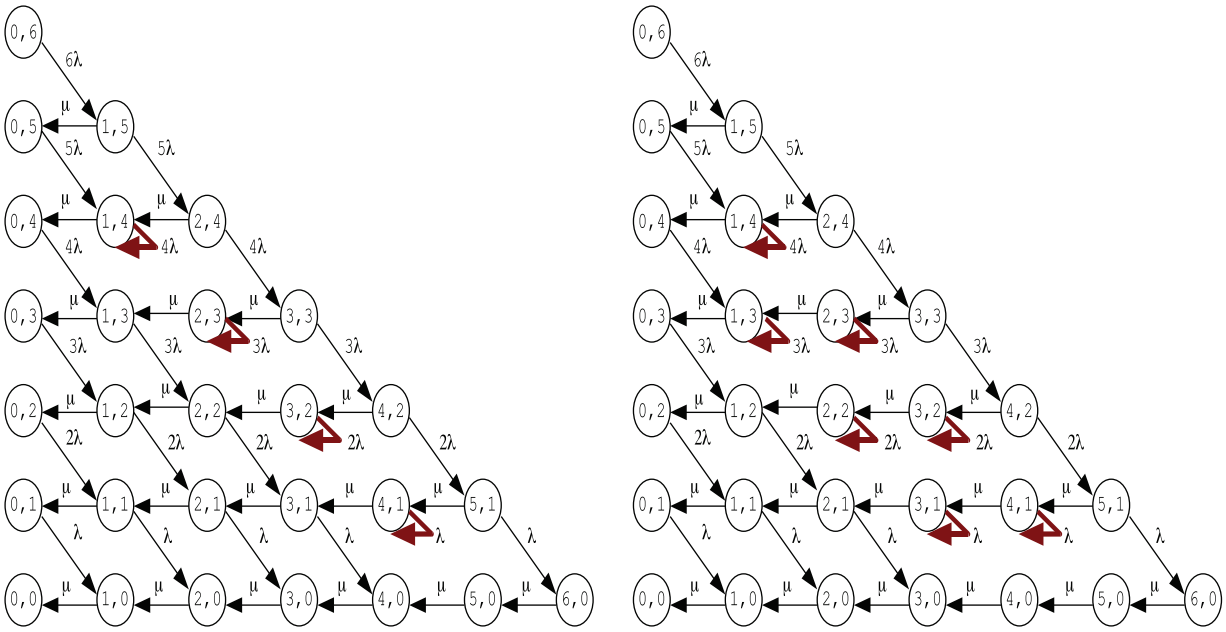

Fig. 3. λ-aggregation: (a) $m = 1$; (b) $m = 2$.

In the main case, we generalize the case where the elementary Web services invoked number is constant using the Chernoff bounds.

6.1. Bounding models

We note $p < 1$, the elementary Web service probability invocation. First, we observe that the composite Web service response time increase with the number of elementary Web services who participate to its composition. This intuitive observation can be easily proved using Corollary 1, with $\mathcal{M}' = \mathcal{M}$ where $\mathcal{M}$ is the Markov chain corresponding to the composite Web service with the bigger elementary Web services. Indeed, the initial state of the Markov chain corresponding to the smaller number of elementary Web services is paired with the initial state of $\mathcal{M}$ in the coupling. So, the following proposition establishes the reachable upper bound.

Proposition 3. *Let X be the binomial random variable corresponding to elementary Web services with parameters (n, p). Let n' be such that $np < n' \leq n$ and $\delta = (n' - np)/np$. Thus:*

$$E(T_{var}^{n,p}) \leq E(T_{fix}^{n'}) + e^{-\delta^2 np/3} E(T_{fix}^{n}) \tag{4}$$

Proof. The total probability theorem allows us to write:

$$E(T_{var}^{n,p}) = E(T_{var}^{n,p} \mid X \leq n')P(X \leq n') + E(T_{var}^{n,p} \mid X > n')P(X > n')$$

Applying twice the bound obtained in the case of fixed elementary Web services, bounding $P(X \leq n')$ by 1 and applying the Chernoff bound, we obtain:

$$E(T_{var}^{n,p}) \leq E(T_{fix}^{n'}) + e^{-\delta^2 np/3} E(T_{fix}^{n})$$

□

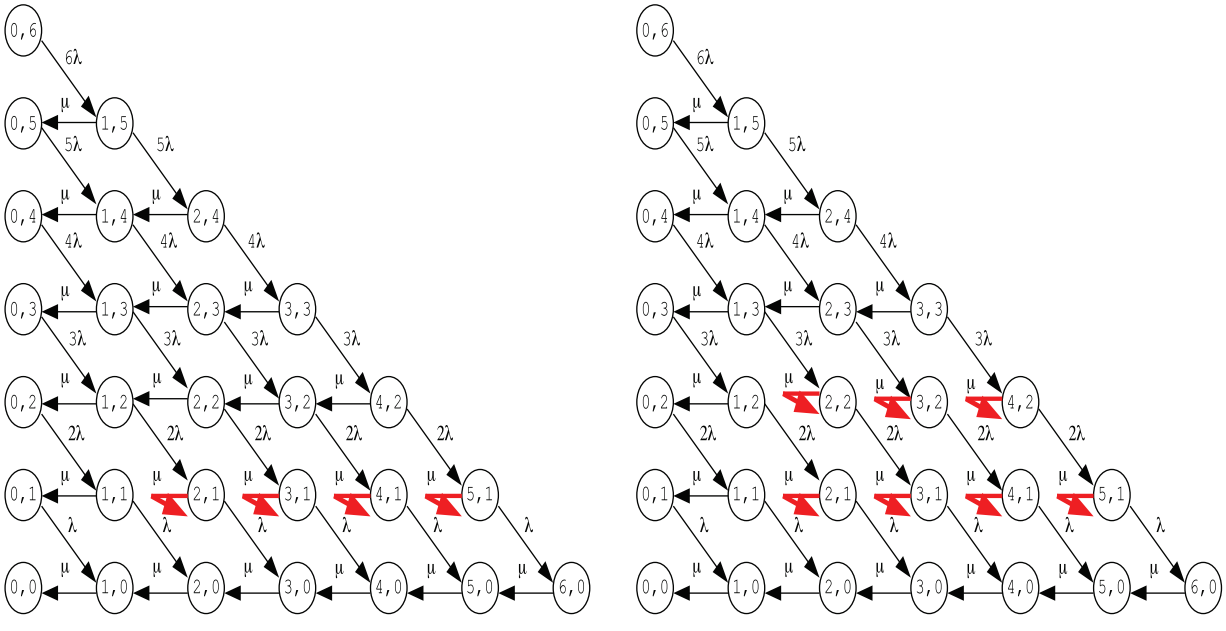

Fig. 4. μ-aggregation: (a) $m = 1$; (b) $m = 2$.

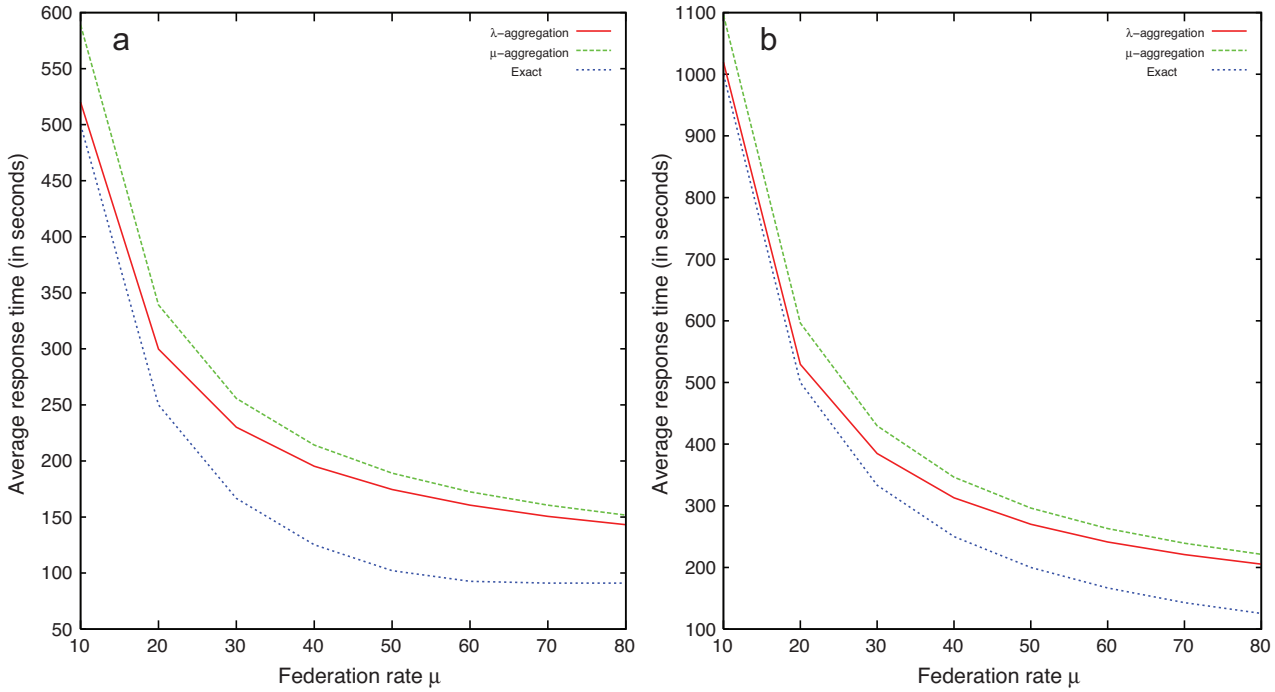

Fig. 5. Services number: (a) $n = 5000$; (b) $n = 10{,}000$.

7. Numerical results for the "fork merge" pattern and a variable number of invocations

We present, in this section, the numerical results obtained in the case where the invoked elementary Web services are modeled by a random variable. The curves in Fig. 9 show the δ optimal value evolution as a function of the elementary Web service invocation probability p. To obtain the δ optimal value, we have computed for a given probability invocation p the best value of upper bound and then we have deduced the best value of upper bound by varying δ. According to the obtained results, we observe that:

- when the invocation probability is very small, δ optimal tends to 1,
- the optimal value δ decreases as a function of the invoked elementary Web services number.

A summary giving an indication about the upper bounds quality and the significant gain on the execution cost, is given in Tables 2 and 3 (note that the upper bound computation is instantaneous) in the case respectively of the probability invocation equal to 0.2 and 0.8. The rate values are $\lambda = 0.1$ and $\mu = 0.8$. Thus the upper bound values are given by the initial μ-aggregation are better than

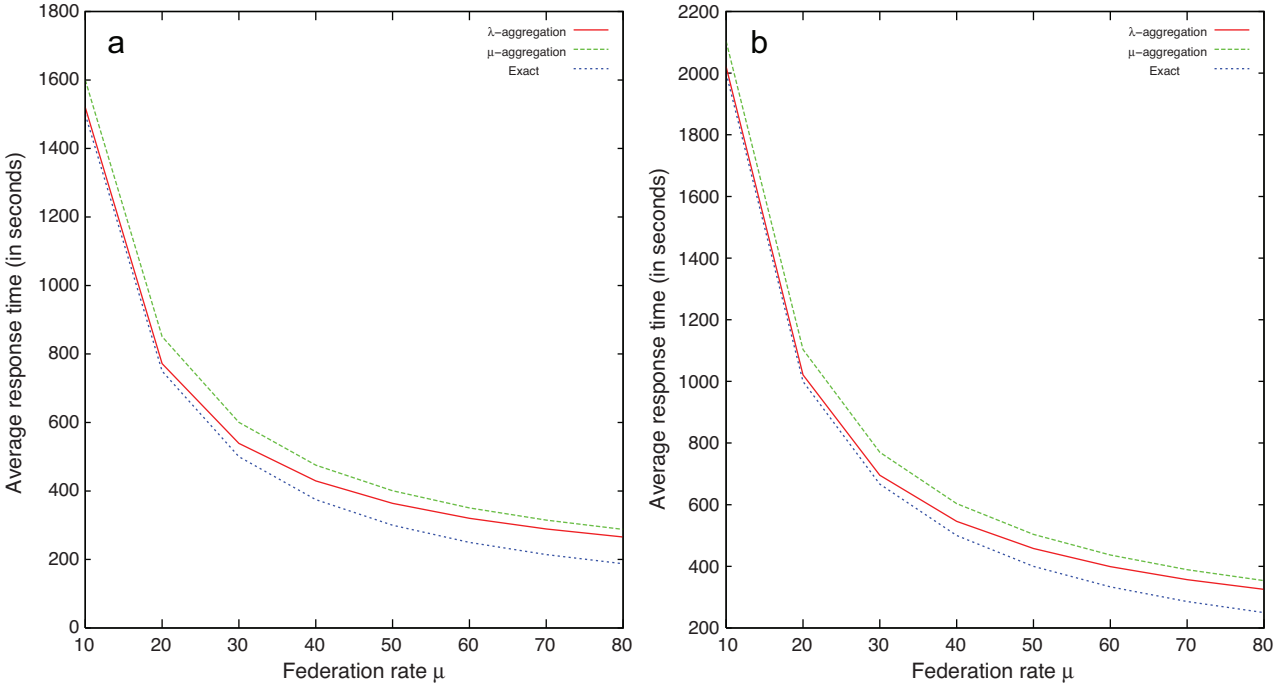

Fig. 6. Services number: (a) $n = 15{,}000$; (b) $n = 20{,}000$.

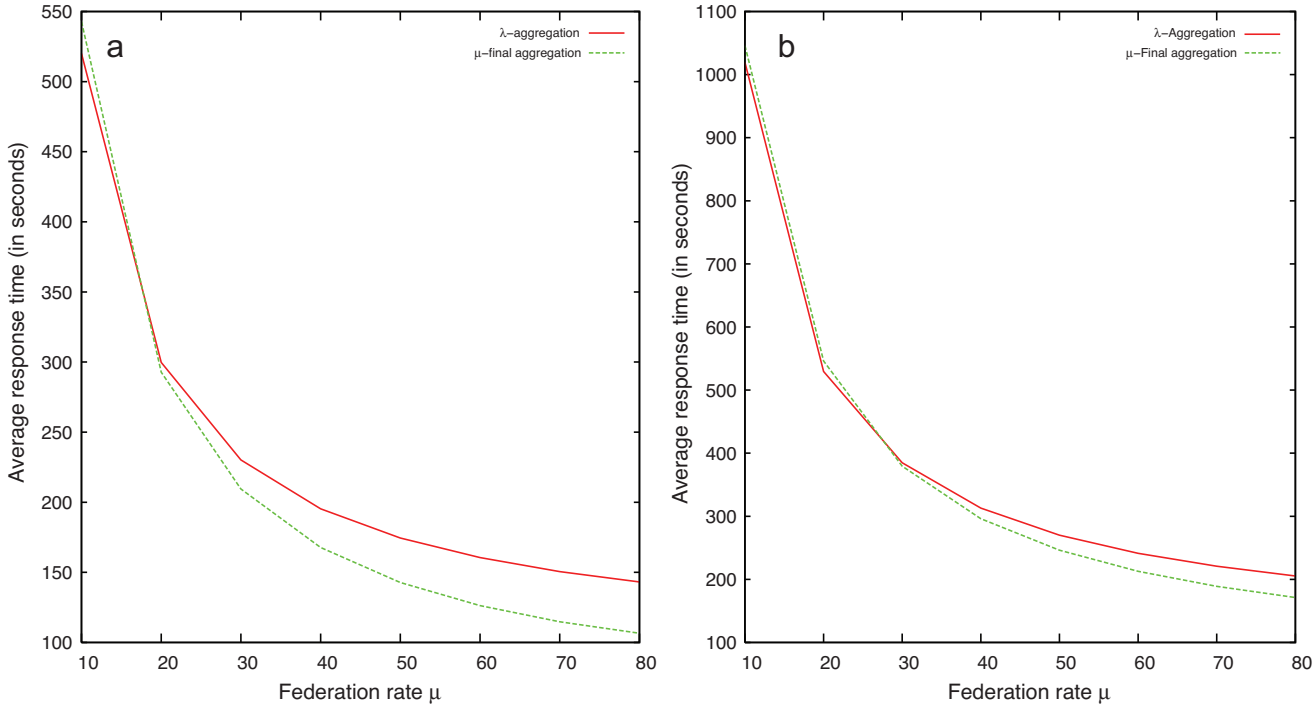

Fig. 7. Services number: (a) $n = 5000$; (b) $n = 10{,}000$.

Table 2
Response time computational cost, upper bound value computational cost and bound accuracy in the case where $p = 0.2$.

Services number	ET	EV	Bound	RE	δ-Optimal
5000	1087.69	74.86	76.08	1.6%	0.12
8000	4444.00	79.56	80.57	1.3%	0.10
10,000	8709.96	81.79	82.80	1.2%	0.09
15,000	29,291.6	85.89	88.653	3.2%	0.07

Table 3
Response time computational cost, upper bound value computational cost and bound accuracy in the case where $p = 0.8$.

Services number	ET	EV	Bound	RE	δ-Optimal
5000	1087.69	88.73	97.66	10%	0.06
8000	4444.00	93.84	127.74	36.12%	0.04
10,000	8709.96	103.35	149.16	44.32%	0.04
15,000	29,291.6	150.06	201.89	34.54%	0.03

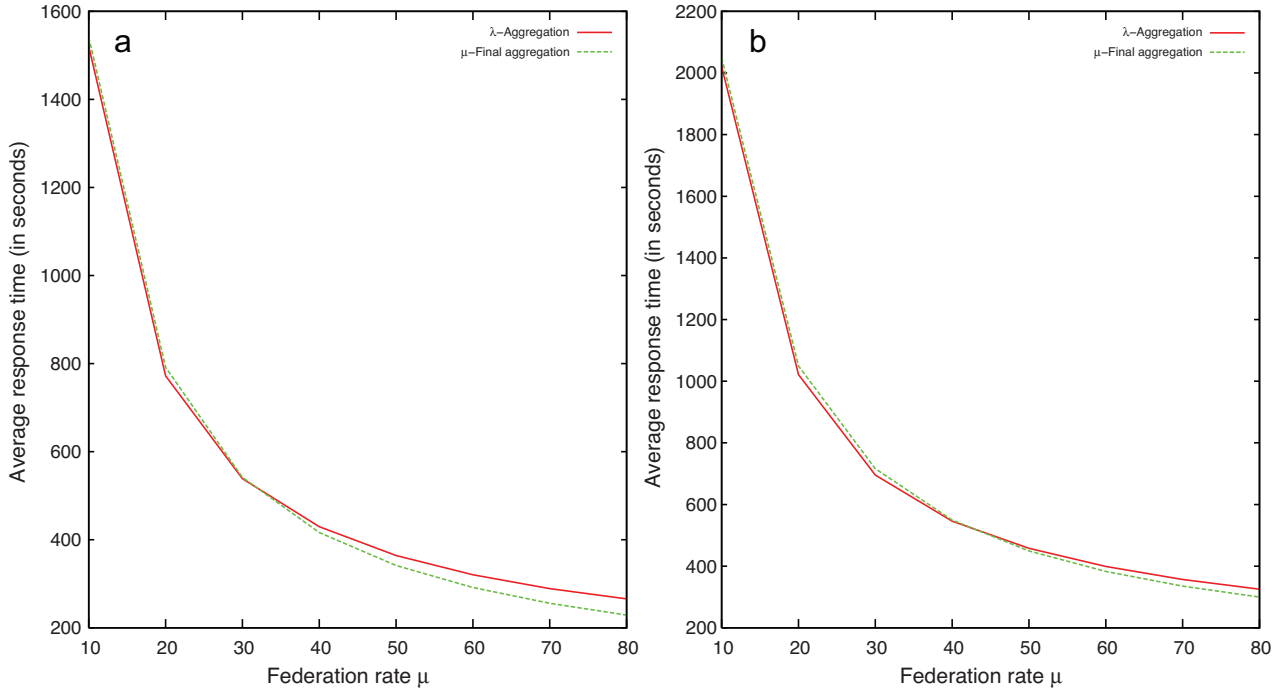

Fig. 8. Services number: (a) $n = 15{,}000$; (b) $n = 20{,}000$.

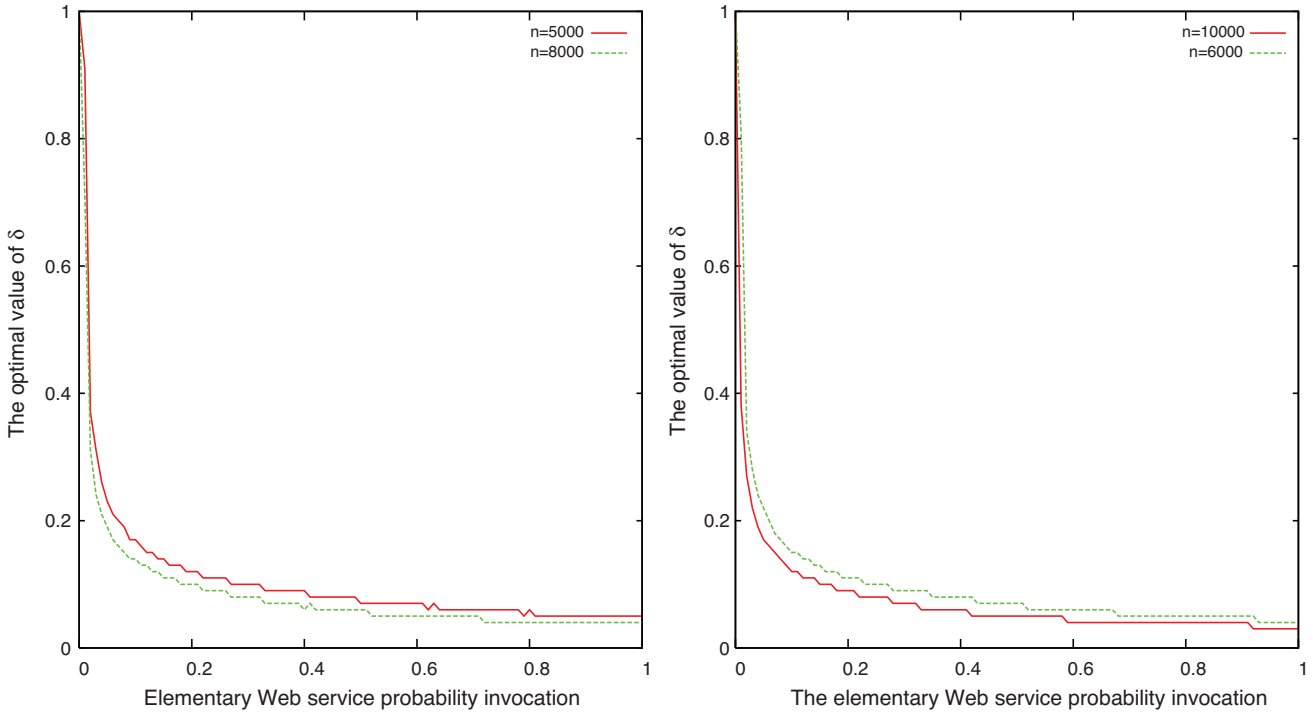

Fig. 9. The δ-optimal value evolution as a function of the invocation probability: (a) $n = 5000$, $n = 8000$; (b) $n = 6000$, $n = 1000$.

the upper bound values obtained by the λ-aggregation. In these tables, *ET* represents the execution time expressed in seconds for the composite Web service computation exact average response time, *EV* represents the exact value of the composite Web service, *RE* is the relative error between the exact value of the composite Web service response time and the upper bound value and *δ-optimal* is the optimal value of the δ-parameter. The optimal value of δ parameter is obtained as follows: for a given invocation probability p, we let the δ-parameter vary by steps 0.001 and select the best upper bound value obtained (i.e. the smallest value).

8. Conclusion

We have proposed, in this paper, an approach based on the stochastic process coupling where the objective is to compute composite Web services response time upper bounds. The interest of our approach comes from the computation time reduction. Moreover as our approach is parametrized, it allows to obtain a tradeoff between the numerical computation cost and the accuracy of bounds by selecting an appropriate value of the parameter. More precisely, we have proposed three families of bounding models for the composite Web service called λ-aggregation, initial μ-aggregation and final μ-aggregation. We have also proposed the combination of these models with the Chernoff bounds in order to take into account the fact that the number of invoked elementary Web services can be variable. Using our approach, in this case, we are reducing the analysis to only two fixed number of invocations.

We will consider two extensions of the work presented here. First, we plan to generalize the study by taking into account more complex patterns (e.g. hierarchical composite Web services). Secondly, we plan to handle heterogenous elementary Web services invoked in the composition are heterogenous. In this last case, our bounding models could decrease exponentially the computational complexity.

References

[1] F. Curbera, I. Silva-Lepe, S. Weerawarana, On the integration of heterogeneous Web service partners, in: OOPSLA Workshop, Tampa, Florida, October 15, 2001, pp. 1–5.
[3] C. Bussler, D. Fensel, A. Maedche, A conceptual architecture for Semantic Web enabled Web services, SIGMOD Record ACM Special Interest Group on Management of Data 31 (4) (2002) 24–29.
[4] J. Yang, M.P. Papazoglou, Service components for managing the life-cycle of service compositions, Information Systems 29 (2) (2004) 97–125.
[5] X. Fu, T. Bultan, J. Su, Analysis of interacting BPEL web services, in: WWW'04: Proceedings of the 13th International Conference on World Wide Web, ACM Press, New York, NY, USA, 2004, pp. 621–630.
[6] T. Bray, J. Paoli, C.M. Sperberg-McQueen, E. Maler, F. Yergeau, J. Cowan (Eds.), Extensible Markup Language (XML) 1.1, 2nd ed., 2006, http://www.w3.org/TR/2006/REC-xml11-20060816/, W3C Recommendation, 16 August 2006, edited in place 29 September 2006.
[7] D. Stoyan, Comparison Methods for Queue and Other Stochastics Models, J-Wiley and Son, 1976.
[8] W.A. Massy, Stochastic ordering for Markov processes on partially ordered space, Mathematics of Operation Research 12 (1986) 350–367.
[9] D.A. Menascé, QoS issues in Web services, IEEE Internet Computing 6 (6) (2002) 72–74.
[10] D.A. Menascé, et al., Response time analysis of composite Web services, IEEE Internet Computing 8 (1) (2004, January/February) 90–92.
[11] S. Haddad, L. Mokdad, S. Youcef, Response Time Analysis of Composite Web Services, Communication Systems, Networks and Digital Signal Processing (CSNDSP), IEEE Computer Society, Graz University of Technology, 2008, July, pp. 42–49.
[12] D.A. Menascé, et al., Static and dynamic processor scheduling disciplines in heterogeneous parallel architectures, Journal of Parallel and Distributed Computing 28 (1) (1995) 1–18.
[13] S. Hahn, L. Fatto, Two parallel queues created by arrivals with two demands, Applied Mathematics 44 (1984, October) 1041–1053.
[14] L. Fatto, Two parallel queues created by arrivals with two demands II, Applied Mathematics 45 (1985, October) 861–878.
[15] R. Nelson, A.N. Tantawi, Approximate analysis of fork/join synchronization in parallel queues, IEEE Transaction Computer 37 (6) (1998) 739–743.
[16] A. Makawski, S. Verma, Interpolation approximations for symmetric fork–join queues, Perform. Evaluation Journal 20 (1–3) (1994) 245–265.
[17] F. Bacelli, A.M. Makowski, Simple computable bounds for the fork–join queue, Proceeding Information Science (1985, March) 436–441.
[18] P. Heidlberg, K.S. Trivedi, Analytic queuing models for programs with internal concurrency, IEEE Transaction Computer C-32 (1993, November) 73–82.

[19] M. Sharf, On the response time of the large-scale composite Web services, in: Proceedings of the 19th International Teletraffic Congress (ITC 19), Beijing, 2005, pp. 1807–1816.
[20] C.W. Piotr, F. Pawel, W. Grzegorz, Time distribution in structural workflow nets, Fundamental Information 85 (1–4) (2008) 67–87.
[21] D. Davis, M.P. Parashar, Latency performance of soap implementations, in: CCGRID'02: Proceedings of the 2nd IEEE/ACM International Symposium on Cluster Computing and the Grid, Washington, USA, IEEE Computer Society, 2002, pp. 407–415.
[22] C. Kohlhoff, R. Steele, Evaluating soap for high performance business applications: real-time trading systems, in: Proceedings of WWW, 2003, pp. 1–8.
[23] P. Sandoz, S. Pericas-Geertsen, K. Kawaguchi, M. Hadley, E. Pelegri-Llopart, Fast Web Services, 2009, http://java.sun.com/developer/technicalarticles/webservices/fastws/.
[24] J. Klingemann, J. Wäsch, K. Aberer, Deriving service models in cross-organizational workflows, in: Proceedings of RIDE -Information Technology for Virtual Enterprises, Sydney, Australia, 1999, pp. 100–107.

Serge Haddad is a former student at the Ecole Normale Supérieure de Cachan. He receivedthe MSc degree in mathematics in 1977 from the University of Orsay andthe MSc and PhDdegrees in computer science in 1983 and 1987, respectively, from theUniversity of Paris 6.He is currently a full professor at the Ecole Normale Supérieure deCachan. His researchinterests include quantitative verification withemphasis on timed and stochastic systems andapplications to software engineering.

Lynda Mokdad is a Professor in Computer Science in the University of Paris-Est, Créteil. Her research interests are performance evaluation techniques (Exact, approximate, stochastic methods), Applications in broadband wired, wireless and mobile networks and QoS in Web services architectures. She has published numerous research articles in peer-reviewed conferences and journals. She obtained her PhD from University of Versailles in 1997 on "techniques and Tools for networks performance evaluation", and her "Habilitation á diriger des recherches" from University of Paris-Dauphine in 2008 on "Contributions to performance evaluation techniques and applications to IP networks and software technologies." She has been member of several program committees and program chairs of workshops and conferences and she was guest editor for Concurrency and Computation: Practice and Experience journal, Wiley, and for Cluster computing journal, Springer.

Samir Youcef received PhD degrees in computer science in 2009 from the university Paris-Dauphine. He is currently an assistant professor at the Nancy 1 university. He did his research at the LORIA (Laboratoire Lorrain de Recherche en Informatiques et ses Appliquations) Laboratory. His main research interest is about performance evaluation of Web services.