International Journal of Computer Engineering and Information Technology

VOL. 9, NO. 5, May 2017, 97–107
Available online at: www.ijceit.org
E-ISSN 2412-8856 (Online)

Mathematical Analysis of the Relative Speedup Dynamics of the Service Layered Utility Maximization Model as Applied to Dynamic Workflow Oriented Webservice Composition

Abiud Mulongo[1], Elisha Opiyo[2], Elisha Abade[3] and William Odongo[4]

[1] InformationUAP-Old Mutual Group Limited, Nairobi, Kenya

[2, 3, 4] School of Computing and Informatics, University of Nairobi, Nairobi, Kenya

[1]aabiudwere@gmail.com, [2]opiyo@uonbi.ac.ke, [3]eabade@uonbi.ac.ke, [4]wokelo@uonbi.ac.ke

ABSTRACT

The Service Layered Utility Maximization model, SLUM is an emerging two phase layered mixed integer programming (MIP) for solving the dynamic webservice composition problem far more efficiently than the state of the art. Recent performance studies have demonstrated that: One, SLUM relative speedup, Ω , with respect to the standard one phase MIP algorithms, dynamically grows larger as the number of service providers per workflow task, n grows larger and vice versa. Two, Ω scales exponentially in the number of sequential worklow tasks, k . However, the effect of varying degrees of service phase transition from layer one to layer two on Ω has not been adequately explored. Secondly, existing studies have emphasized the layering scheme in which the size of layer one q_1 and the size of layer two q_2 are equal. However, in reality, q_1 and q_2 could differ depending on application scenarios. This paper formulates mathematical models that quantify the dynamics of SLUM relative speedup for any SLUM layering scheme of the form (q_1, q_2) taking into account the degree of service phase transition, p . For a known layering scheme (q_1, q_2) , we derive mathematical functions for the minimum possible Ω and for the maximum possible Ω at known values of k , n , q_t and p , where q_t is total number of webservice quality of service parameters. Third, we show that given q_t , there are $(q_t - 1)$ valid SLUM layering schemes. From the set of $(q_t - 1)$ possible schemes, mathematical functions that capture the lower and upper SLUM speedup limits are presented. We pioneer a graphical method of visualizing the speedup dynamics as function of p . At design time, service system architects could use these models to anticipate the runtime performance efficiency benefits of a selected SLUM layering scheme.

Keywords: *Mathematical Analysis, Performance Dynamics, Relative Speedup, Service Layered Utility Maximization, SLUM, Service Composition.*

1. INTRODUCTION

1.1 The Dynamic Work-flow Based Web-service Composition Problem

The problem of dynamic work-flow based web-service composition is a strategic success factor especially in distributed virtual collaborative large scale electronic commerce networks [1]-[2]. This is because it has the potential to afford such organizations the degree of agility required to optimally respond to time varying service demands [3]-[4]. As illustrated in figure 1, in the case of collaborative networks, the problem involves a business process that spans a network of different types of collaborating service providers. The business process is automated via a work-flow of k different types of collaborating service providers. The business process is automated via a work-flow of tasks. Each task represents a service offered by one of the service providers. At design time, each work-flow task is automated through a web-service. Every task in the work-flow could be performed by n alternative service providers and hence executable by alternative web-services. Every web-service has the same set of quality of service (QoS) attributes such as service reliability, service cost, service throughput, service response time. **T**he total number of QoS attributes is q_t . The QoS values may vary within and across web-services from time to time, for example at some point in time, a service that was previously very responsive becomes slower or the access cost of the service is changed by the service owner or a service may become unavailable during the execution of a work-flow, terminating the work-flow execution and hence hindering service delivery. Customers emit complex service requests that require the execution of the k tasks in the logical sequence enforced by the work flow thus requiring composition of k web services. Different customers have varying constraint implications on the set of q_t QoS attributes at a given time instant. At run time, given a customer request, a service composition system is expected to automatically generate an optimal sequence of web-services efficiently from the pool of candidate web services. This is the dynamic work-flow service composition problem. Although potentially useful, the problem is known to be non-deterministic polynomial hard multiple criteria decision making problem due to the curse of dimensionality of the design variables involved [7-10]. This limits the applicability of the technique to small scale problems only [3-5].

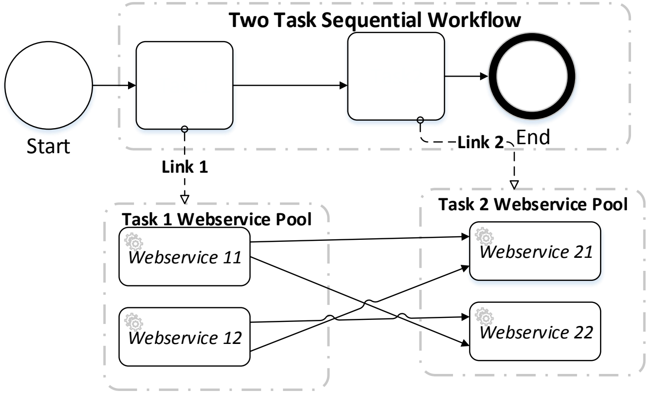

Fig. 1. Dynamic Workflow Webservice Composition

A standard and widely used method of efficiently modeling and solving the dynamic work flow based service composition problem, pioneered by Benatallah et al [6], hereafter, S-MIP, formulates a single problem as a mixed inter program (MIP) over the range of q_t QoS parameters. Practical implementation of S-MIP require that end users specify value constraints and preferences in form of weights over the range of q_t QoS attributes, from which the objective function and constraint inequalities are instantiated. This requirement places a huge and unnecessary burden on the user inhibiting usability of S-MIP [5]. Additionally, although more efficient than naïve approaches, S-MIP is amenable to exponential state space explosion as the complexity of the optimization model grows large enough [3]-[6].

1.2 The Service Layered Utility Maximization Model.

The service layered utility maximization model, SLUM, is a pioneering optimization model recently designed by Mulongo et al [3] to operate adaptive service oriented architecture (SOA) work-flow systems. SLUM specifically targets dynamic service composition in highly volatile e-commerce trading environments with tight time varying constraints on service response times and stringent constraints on the optimality of complex packages of virtual goods and services as dictated by different service consumers from time to time. SLUM extends the S-MIP formulated by Benatallah et al [6]. Unlike all the other MIP strategies based on S-MIP, inspired by the ideas of layering as optimization decomposition in [11]-[14], SLUM restructures a dynamic web-service composition problem into two partially layered mixed integer programs. The first subproblem known as the Service Consumer Utility Maximization, SCUM problem is formulated on a subset of web-service quality of service parameters of size q_1 that are deemed a direct concern of the user, for example, service access cost and service response time [3]. Only feasible solutions at the end of SCUM flow to the second layer known as the Service Provider Utility Maximization, SPUM layer for further optimization [3]. The SPUM mixed integer program is formulated over a second subset of parameters of size q_2 that are too technical and too low level in nature to be discernible by and to be a direct concern of an average user but which contribute to the overall optimality of the composite service [3]. Quality parameters at the SPUM layer include service throughput and service reliability [3].Thus, SLUM optimization design approach, unlike all the rest, alleviates the service consumer from the tedium of having to specify QoS constraints and weight preferences on a huge range of quality parameters, some which should otherwise be only optimized transparently [3]. In addition, our previous performance studies on SLUM in [4]-[5] demonstrate that SLUM is multiple times more efficient that S-MIP while simultaneously delivers acceptable service optimality averaging 93% in accuracy.

The structural design of SLUM connotes the following broad scenarios. In scenario one, all composite services at SCUM might be feasible and flow to the SPUM layer. The relative speedup of SLUM ,with respect to S-MIP under 100% transition to the SPUM layer, known as the SLUM super-linear speedup has been studied in [5] and shown to be $2^{(k-1)}$ under the assumption that $q_1 = q_2$. The second scenario represents a zero percent flow of composite services from the SCUM layer to the SPUM layer. Since a zero percent flow implies infeasibility, empirically, it's difficult to measure performance under this circumstance. The empirical alternative is to tune optimization constraint inequalities such that the resulting percentage flow is reasonably close to zero. Although the percentage flow of services is not explicitly reported, the study in [4] demonstrates that on a two task work-flow, SLUM is about 3.6 times faster than S-MIP on average at $q_1 = q_2$. A re-examination of the empirical results in [4] reveals that the average case results hold at a percentage flow of about 2%, which is closer enough to zero percent. Moreover, the study in [4] shows that the average speedup could not be more than 2^k . The first two scenarios are extreme cases which are only highly probable in situations where either there is little differentiation on web services QoS attribute values at a point in time or the variance in consumer service request QoS constraints is negligible. The third scenario is that in which the percentage of services that flow from SCUM to SPUM layer could be anything between 0% and 100%.

There is no existing study that investigates the performance dynamics of SLUM under the third scenario where the percentage flow of services could be any value between 0% and 100%. Given the highly dynamic service environments that SLUM targets, the percentage flow of services from SCUM to SPUM layers is likely to be highly varied from time to time. A systematic understanding of how the varying percentage of service flows in SLUM impacts the run time efficiency of SLUM and hence its relative speedup dynamics is therefore imperative in order to optimally benefit from the relative efficiency benefits of SLUM.

A second gap in the literature is that the previous performance studies have emphasized a scenario where the SLUM layering scheme is such that $q_1 = q_2$. However, given a total of q_t service quality attributes associated with each and every work-flow task in a service composition, a system architect could derive several SLUM alternative layered structures (schemes) having different permutations of q_1 and q_2 values. The first question that arises is, how many alternative SLUM layering schemes could be formulated given? Secondly, how does the performance of SLUM change with different layering schemes defined by the structure of the form? No previous study comprehensively addresses these questions as well.

The purpose of this study is to formulate generalized mathematical models that quantity the relative speedup dynamics of any valid SLUM layering scheme taking into account the varying degrees of the percentage of the services that flow from the SCUM to SPUM layers.

1.3 Problem Statement

Consider the dynamic work-flow web-service composition problem described in section *A.* Let $T_B(n, k, q_t)$ be the time taken by the SLUM algorithm to solve the problem with parameters n, k, q_t . Let $T_A(n, k, q_t)$ be the time taken by the S-MIP one shot algorithm to solve the same problem. The ratio $(T_A)/(T_B)$ is the SLUM relative speedup and is denoted by Ω [5]. Analyzing Ω is a significant problem as it aids in quantification of the relative efficiency benefits of adopting a SLUM over S-MIP for dynamic web service composition. The result is that a virtual enterprise broker could determine when to apply which of the schemes, under what conditions and the magnitude of relative efficiency gains [4], [15]. However, several dynamics make the analysis of Ω a nontrivial problem. This is because of three main factors which are highlighted in the subsections 1, II and III.

1.3.1 Effect of Service Elimination at the Service Consumer Utility Maximization Layer

At fixed values of n, k, q_t for a given SLUM layering scheme, the magnitude of Ω when no web-service is eliminated is bound to differ from the magnitude Ω of when some web services are eliminated. Quantifying the variation of Ω given a certain magnitude of service elimination at the SCUM layer is thus critical. The solution this problem is unknown in the literature.

1.3.2 Effect of the Structure of the Layering Scheme

Holding the rest of the factors constant, different SLUM layering schemes could yield different Ω values. Suppose you have three SLUM layering schemes of the form $(q_{1,} q_2)$ such $LS_1 = (a, b), LS_2 = (c, d), LS_3 = (b, a)$. Despite that in all the three layering schemes, $a + b = c + d = b + a = q_t$ the performance efficiency and hence the Ω values of the three schemes might not necessarily be the same. Even more complicated, $\Omega(LS_1)$ and $\Omega(LS_3)$ might not necessarily be equal in spite of the two schemes being identical except that their structures are the inverse of each other. There are two reasons for this hypothesis. The first reason is that the run time performance of SLUM has been shown to be nonlinear, precisely exponential in k in [4]-[5]. In [5], it's been shown that Ω is proportional to $(q_1^k + q_2^k)$. Thus, the fact that $(q_1 + q_2)^k \neq (q_1^k + q_2^k)$ explains why

$\Omega(LS1) \neq \Omega(LS2)$. The second reason lends itself to three coupled factors -the nonlinear growth, the strict ordering of layers and the effect of service elimination. Because in SLUM the optimization process strictly starts at the SCUM layer before proceeding to the SPUM layer [3], service elimination at the end of SCUM layer has an effect on the computational complexity of the optimization process at the SPUM while the converse is not true. Coupling this with the non-linearity of the complexity, means that $\Omega(a,b)$ might not necessarily be equal to $\Omega(b,a)$. These issues lead us to the refinement of the problem in section 1.5.1 leading to the question: For a given SLUM layering scheme defined by the ordered set (q_1, q_2), at fixed n, k, q_t, how does service elimination affect Ω? This refined problem has not been tackled previously. Questions related to this are, what is the range of Ω for a given layering scheme in the presence of service elimination at the SCUM layer?

1.3.4 A large Layering Scheme Design Space

Given the parameter q_t, it's easy to show that theoretically there are $q_t - 1$ possible layering schemes of the form (q_1, q_2) that satisfy the following two conditions. One, $q_1 + q_2 = q_t$. And because SLUM must have two layers each with at least one web service QoS attribute, an obvious condition is that $q_1 \geq 1, q_2 \geq 1$. As an example, consider a dynamic web service composition problem in which each web service in which $q_t = 7$, the following are valid SLUM layering schemes: $(1,6),(6,1),(2,5),(5,2),(3,4),(4,3)$. We refer to the set of all possible SLUM layering schemes of SLUM layering scheme design space and denote by D. Hence, formally, D is the set $((1,q_t-1),(q_t-1,1),(1,q_t-2),(q_t-2,2),\ldots,(q_t-m,m),(m,q_t-m))$ of size $q_t - 1$ where $1 \leq m \leq q_t - 1$. Confronted with a D of size $q_t - 1$, both the researcher and practitioner could be interested in answering the following questions. What is the worst possible and best possible performance of SLUM? Answering this question is central to understanding the performance limits and hence the relative speedup limits of SLUM given the initial parameters n, k, q_t.

1.4 Research Objectives

1. Analytically, using mathematical models , quantify the dynamics of the SLUM relative speedup with variations in percentage of services eliminated at the SCUM layer for different layering schemes
2. Establish the theoretical relative speedup limits of a given SLUM layering scheme
3. Establish the theoretical global relative speedup limits of a given SLUM.

1.5 Research Questions

On the basis of the foregoing problems stated in section 1.5 and the research objectives in 1.6, the study sought to answer the following specific research questions:

RQ1**:** For a given SLUM layering scheme, how does the percentage of services eliminated at the SCUM layer quantitatively affect the overall relative speedup of SLUM?

RQ2**:** For given SLUM layering scheme, what is the minimum possible relative speedup?

RQ3: For a given SLUM layering scheme what is the maximum possible relative speedup?

RQ4: Given a SLUM layering scheme design space, what is the minimum possible relative speedup that could ever be achieved by SLUM?

RQ5: Given a SLUM layering scheme design space, what is the maximum possible relative speedup that could ever be achieved by SLUM?

1.6 Organization of the Rest of the Paper

The rest of the paper is organized as follows. The entire of section 2 is structured to answer all the seven research questions. Subsection 2.1 introduces a generalized SLUM run time performance efficiency function $T_B(n, k, q_t)$, considering service elimination at the SCUM layer. In subsection 2.2, a generalized relative speedup model Ω is derived from the generalized run time performance function. In 2.3, we introduce and define the concept of composite service phase Transition Ratio denoted by p and show that p is a normalized parameter on the continuous interval $[0,1]$ for quantifying the magnitude of service elimination at the SCUM layer. In section 2.4, we recast the generalized relative speedup function Ω obtained in 2.2 as a function of p. By the end of subsection 2.4, it's expected that the reader shall have obtained a comprehensive quantitative response to research question $RQ1$ .Subsection 2.5 is dedicated to addressing research questions $RQ1, RQ2, RQ3, RQ4, RQ5$ in which we derive the lower and upper bounds of the relative speedup functions of a given layering scheme, and the lower and upper limits of the relative speedup of SLUM over the layering scheme design space D. A special layering scheme called the SLUM balanced layering scheme $SBLS$ is introduced in 2.6, its special relative speedup function, minimum and maximum functions are as well are derived building on the results obtained in the preceding subsections. We relate this work with past studies in section 3. In section 4, we summarize our main contributions, derive conclusions and reflect on ongoing and future work.

2. SLUM RELATIVE SPEEDUP DYNAMICS

Consider that the number of candidate composite web-services available at the beginning of phase one optimization is n^k for a sequential business work-flow with k tasks [5]. the computational effort performed by SLUM at layer one is proportional to $\left[\left[\frac{q_1}{q_1+q_2}\right] * n\right]^k$ [5]. In the presence of service elimination at phase one, for each ith task $(n- \in_i)$ transition from layer one to layer two. Following an analysis similar to that one in [5], the computational effort at layer two is therefore $\prod_{(i=1)}^{k} n - \epsilon_i \left[\left[\frac{q_2}{q_1+q_2}\right] * n\right]^k$ .Consequently, the generalized run-time efficiency model and the generalized relative speedup model **of** SLUM are according to (1 -2) and (3-4) respectively.

2.1 Generalized SLUM Run time Efficiency Model

$$T_B(n,k,q_t) = \left(\frac{(q_1)^k}{(q_1+q_2)^k}\right) n^k + \left(\prod_{(i=1)}^{k} n - \epsilon_i\right)\left(\frac{(q_2)^k}{(q_1+q_2)^k}\right) \quad (1)$$

$$T_B(n,k,q_t) = \frac{\left((q_1)^k n^k + (q_2)^k \left(\prod_{(i=1)}^{k} n - \epsilon_i\right)\right)}{(q_1+q_2)^k} \quad (2)$$

2.2 Generalized SLUM Relative Speedup Model

$$\Omega = \frac{n^k}{\frac{\left((q_1)^k n^k + (q_2)^k \left(\prod_{(i=1)}^{k} n - \epsilon_i\right)\right)}{(q_1+q_2)^k}} \quad (3)$$

$$\Omega = \frac{\left(n^k (q_1+q_2)^k\right)}{\left((q_1)^k n^k + (q_2)^k \left(\prod_{(i=1)}^{k} n - \epsilon_i\right)\right)} \quad (2)$$

This study defines the function in (4) as the generalized SLUM relative speedup function or simply the Ω *function*.

2.3 Composite Service Phase Transition Ratio and Mean Service Transition Rate

We define the composite service phase Transition Ratio p, as the ratio of the number of alternative composite services that transition from the SCUM layer (phase one) to the SPUM layer (phase two) for further optimization, to the initial number of alternative composite web-services that are available at the start of the SCUM optimization process. It follows that $\rho = \frac{\left(\prod_{(i=1)}^{k} n - \epsilon_i\right)}{(n^k)}$. Given that the maximum value of $\prod_{(i=1)}^{k} n - \epsilon_i$ is n^k and the lowest value of $\prod_{(i=1)}^{k} n - \epsilon_i$ is zero, it follows that $0 \le p \le 1$. Thus when $p = 1$, it means that all candidate web services satisfied the SCUM composite service selection problem specification and therefore there was 100% transition of the services to the SPUM phase. On the other hand when $p = 0$, it signifies two things. One, no composite service constituted a feasible solution to the SCUM problem specification. Hence, secondly, none of the services transitioned to the SPUM layer for further optimization.

The parameter p plays a significant role in the analysis of the SLUM relative speedup. To understand the significance, consider the equation (4) above. To report the value of Ω, one needs to report a series of ϵ_i values and the value of n for which Ω is valid. Alternatively, one would report the absolute value of the function $\prod_{(i=1)}^{k} n - \epsilon_i$ as well as the value of n for which Ω holds. For different combinations of values of n and $\prod_{(i=1)}^{k} n - \epsilon_i$, it's not quickly intuitive to correlate the magnitude of Ω against the absolute magnitudes of n and that of $\prod_{(i=1)}^{k} n - \epsilon_i$. Yet, being able to easily correlate the effect of the magnitude of service elimination to the speedup is key. The composite web-service Transition Ratio p overcomes the challenge. Since p lies on the closed interval $[0,1]$ regardless of the magnitude of n and $\prod_{(i=1)}^{k} n - \epsilon_i$, it presents a normalized tool for correlating the relative percentage of web services eliminated at the SCUM phase with the resultant Ω .

Having explained the significance of p , let's take a step back. We have seen that $p=0$ implies infeasibility of the optimization problem at SCUM layer. Further according to SCUM model as presented in [3], SPUM optimization can only follow after satisfying the needs of the service consumers at the SCUM layer. In turn $p=0$ also implies overall infeasibility of the SLUM problem. What follows is the question, if $p=0$ means infeasibility, should we care about the relative speedup of SLUM at $p=0$? Yes we should. By determining Ω at $p=0$, in practice ones starts to understand how Ω behaves as $p \rightarrow 0$, in which case, all other factors held constant, the value of Ω at $p=0$ is the limiting maximum value for Ω .

While p is a measure of the proportion of those composite services that flow from SCUM layer to the SPUM, at any one time, it might of be interest to estimate the average number of web-services per task and hence the average number of service providers per work flow task that account for the a given Ω . We define the average number of services per work-flow task as the mean service transition rate denoted by m . Given p and n , it follows that

$$m^k = \prod_{(i=1)}^{k} n - \epsilon_i \rightarrow \rho = \frac{m^k}{(n^k)}$$

Therefore $m = n \times \sqrt[k]{\rho}$.

2.4 The SLUM Relative Speedup as a Function of the Composite Service Phase Transition Ratio

The Ω function in (4) can be easily re-written as a function of p . Algebraically factorizing the numerator and denominator of the R.H.S of (4) by n^k yields the equation (5).

$$\Omega = \frac{(q_1 + q_2)^k}{(q_1^k + \rho q_2^k)} \qquad (5)$$

2.5 Maximum and Minimum Ω Functions within the SLUM Design Space

In this section we explore solutions to research questions $RQ2, RQ3, RQ4, RQ5$. The solution to these questions is a two-step process. First, we determine the Transition Ratio p at which an arbitrary layering scheme LS drawn from a layering design space D of size $q_t - 1$ that yields the minimum or maximum relative speedup value. The result is a multivariate function in k , q_1 and q_2 . We will hence forth refer to the resultant functions respectively as SLUM generalized minimum relative speedup function Q_{min} , and SLUM generalized maximum relative speedup function Q_{max} . Thus if q_1 and q_2 are known, the specific minimum and maximum values of the LS can be computed using Q_{min} and Q_{max} . Each of these functions can thus yield D different minimum and maximum values one for each layering scheme. Secondly, from the set of $q_t - 1$ minimum Ω functions and $q_t - 1$ maximum functions, we solve for the global minimum and global maximum functions. For the minimization problem, this entails finding a layering scheme (q_1, q_2) that would yield the least possible Ω value. We will call the resultant solution as the SLUM generalized global minimum relative speedup function, Q_{gmin} .For the maximization problem, it involves determining a layering scheme (q_1, q_2) that would yield the maximum possible Ω . We refer to the resultant solution as SLUM generalized global maximum relative speedup function, Q_{gmax} .

We can obtain Q_{min} by substituting the largest possible p value in equation (5). This p value is 1. This leads to (6). On the other hand Q_{max} can be determined by substituting the smallest possible $p=0$ value in (5).

$$\Omega_{min} = \frac{(q_1 + q_2)^k}{(q_1^k + q_2^k)} = \frac{q_t^k}{(q_1^k + q_2^k)} \qquad (6)$$

$$\Omega_{max} = \frac{(q_1 + q_2)^k}{q_1^k} = \frac{q_t^k}{q_1^k} \qquad (7)$$

Next, to compute Q_{gmin} , we need to locate a layering scheme (q_1, q_2) from D such that the values of q_1 and q_2 minimize the function Q_{min} . To tackle this problem, we need to recall the fundamental properties of a SLUM network. Firstly, observe that since $q_1 + q_2 = q_t$, the value of the function q_t^k is constant and independent of the individual values of q_1 and q_2 hence independent of the layering scheme chosen. We then conclude that to in order to determine Q_{gmin} , all we need is to maximize the expression in the numerator in (6) which is the expression $(q_1^k + q_2^k)$. Again from $q_1 + q_2 = q_t$, we see that q_1 and q_2 are complements of one another. Given that $q_1 \geq 1$ and $q_2 \geq 1$, the maximum value of $(q_1^k + q_2^k)$ happens when either q_1 or q_2 has the largest possible value. This value is (q_{t-1}) . Thus given q_t , either of the two layering schemes $(1, q_t - 1)$ and $(q_t - 1,1)$ would yield the least possible relative speedup among all the possible layering schemes. Substituting $(1, q_t - 1)$ or $(q_t - 1,1)$ in (6) yields (8).

$$\Omega_{gmin} = \frac{(q_t^k)}{(1 + (q_t - 1)^k)} = \frac{(q_1 + q_2)^k}{(1 + (q_1 + q_2 - 1)^k)} \qquad (8)$$

To determine Q_{gmax} , we need to find a layering scheme $(q_{1,} q_2)$ that minimizes the expression q_1^k in equation (5).Since Q_{max} is independent of q_2 , what we need is the least possible value of q_1 , which is 1. When $q_1 = 1$, $q_2 = q_t - 1$. Therefore the only SLUM layering scheme that would yield the maximum possible relative speedup Q_{gmax} is the scheme $(1, q_t - 1)$. Substituting $q_1 = 1$ in (7) gives (9).

$$\Omega_{gmax} = (q_t)^k = (q_1 + q_2)^k \qquad (9)$$

2.6 Functions of SLUM Balanced Layering Schemes

A SLUM balanced layering scheme, *SBLS* is such that $q_1 = q_2$. Here, we show one interesting performance characteristic unique to a SBLS; the relative speedup of a SBLS is independent of the parameters q_1 , q_2 and q_t . To confirm this, we substitute q_2 with q_1 in (5) to have equation (10). We will denote the relative speedup at $q_1 = q_2$ as Q_b , the minimum possible value of Q_b as Q_{bmin} and the maximum possible value of Q_b as Q_{bmax} . Q_{bmin} and Q_{bmax} are given according to equation (11) and (12) respectively.

$$\Omega_b = \frac{(q_1 + q_1)^k}{((q_1)^k + (\rho q_1^k))} = \frac{(2q_1)^k}{(q_1^k(1+\rho))} = \frac{2^k}{(1+\rho)} \qquad (10)$$

$$\Omega_{bmin} = 2^{(k-1)} \qquad (11)$$

2.7 SLUM Relative Speedup- Composite Service Phase Transition,-Decay Curves

For a given layering scheme, it would be desirable to graphically visualize how the SLUM relative speedup grows as a function of the composite service Transition Ratio. We refer to such a graph as a SLUM Ω - p decay curve. The decay phrase emphasizes the fact that Ω declines with an increase in p , starting at a peak of $\Omega_{max} = \frac{(q_1 + q_2)^k}{q_1^k}$ at $p = 0$ and dropping to the lowest point $\Omega = \frac{q_t^k}{(q_1^k + q_2^k)}$ at $p = 1$. Therefore, the Ω - p decay curve has a Y intercept at $\Omega_{max} = \frac{(q_1 + q_2)^k}{q_1^k}$. In case of a SLUM balanced layering scheme, the value of Y intercept would be 2^k . We will illustrate the power of Ω - p as a visualization tool through an example.

Example 1:

Given the layering scheme $(q_{1,} q_2) = (4,3)$, plot the Ω - p curves at $k = 2$ and at $k = 3$. To solve this, notice that in theory there are infinite values of p on the continuous interval $[0,1]$. However, in practice, we could generate monotonically increasing p values at small intervals on the interval $[0,1]$ then compute the corresponding Ω values for each p using equation (6).

In this example, we choose an interval of 0.05 which yields the sequence

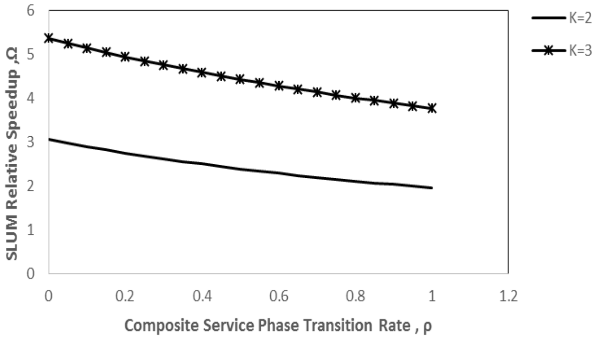

Fig. 2. Sample $\Omega - p$ *Curves at* $k = 2$ *and* $k = 3$

$\rho = 0, 0.05, 0.1, 0.15, \ldots, 0.95, 1.$ of twenty p values. In practice, the p values are automatically computed based on the actual number of web services per task that got promoted to the second layer. Figure 2 shows the resultant $\Omega - p$ decay curves. From the graphs, it's clear that Ω declines with increase in p . Secondly, Ω increases exponentially in k . Thus theoretically, businesses operating larger web service driven dynamic work flows are likely to enjoy even greater efficiency benefits from SLUM.

3. RELATED WORK

Here are two previous studies related to this study. Our work in [4] empirically showed that SLUM is initially 1.3 times worse in performance than S-MIP, 3.6 times faster than S-MIP on average on a two task sequential business work flow for a large enough n . The service composition problem involved web services differentiated on eight different quality of service attributes equally distributed. The SLUM network was balanced with four QoS attributes per layer. Although not reported, the performance results obtained in [4] reveal that the composite service phase Transition Ratio p was 0.0296. Substituting p in "(10)" gives $\Omega = 3.885$. This theoretical value is not significantly different from the empirical value of 3.6 reported in [4]. In any case, it's not possible to achieve the theoretical speedup - the sequential overheads that SLUM has to overcome as data is sequentially moved from layer one to layer two limits this potential [4]. Hence the empirical study in [4] confirms the validity of the theoretical model in"(10)" for a balanced SLUM network derived in this study, in as far as $p \to 0$ .Thus these two studies are complementary. However, while the study in [4] focused on the empirical performance of SLUM at a single p value and for a balanced SLUM network, the focus of this study has been on deriving generalized Ω functions that can be applied for any layering scheme and for any p between 0 and 1. This study can be seen as an extension of our work in [4] and the work in [5] could been seen a special form of the ana; analysis in this paper. The study in [5] establishes a generalized function for super-linear relative speedup which is as $\Omega = \frac{(q_1 + q_2)^k}{((q_1)^k + (q_2^k))}$ the authors in [5] also go ahead to derive the SLUM super-linear relative speedup of a balance SLUM layering scheme as $2^{(k-1)}$. This study formally shows that the super-linear relative speedup up of the function $\Omega = \frac{(q_1 + q_2)^k}{((q_1)^k + (q_2^k))}$ is valid at $p = 0$. Further this study complements the study in [5] by showing that for a balanced SLUM layering scheme, Ω lies on the interval $[2^{(k-1)}, 2^k]$. In addition, this study differs from those in [4]-[5] in following ways. First, in this paper, we determine the performance limits for a defined SLUM layering scheme and the overall performance limits of SLUM given the set of all possible SLUM layering schemes. This paper is the first one to introduce the concept of the composite service phase Transition Ratio p and to formulate mathematical functions expressing the relationship between Ω and p . In conclusion, unlike all the previous SLUM performance studies that focus on a single point, this study is the first to comprehensive quantify the dynamics of the SLUM relative speedup mathematically.

4. CONCLUSIONS

4.1 Research Contributions

In this research, we have derived a series of mathematical models that comprehensively describe and predict the relative performance dynamics of SLUM as applied to the dynamic work-flow based web-service composition, taking into account all the relevant factors which are: the number of web service quality of service parameters, q_t , the number of web-services per work-flow task, n , the number of work-flow tasks, k , the layering scheme design space D , the choice of a layering scheme whose structure is of the form $(q_{1,} q_2)$, and the number of web-services eliminated per task at the SCUM layer at run time, ϵ_i . From these, we have made the contributions highlighted in the subsections that follow.

4.1.1 Establishment of the Range of SLUM layering Schemes

We have demonstrated that given the design time parameter q_t , there are $(q_t - 1)$ different ways of structuring a SLUM network. We called the set of these layering schemes as the layering scheme design space D. Hence, at design time, a virtual e-commerce brokerage organization's system architect initially has a range of $(q_t - 1)$ design options to choose from D . By exploring the performance efficiency of each scheme under varying conditions, system architects can choose the best fit layering scheme for the dynamic work flow based service composition based on domain specific business scenarios. This study empowers the system architect with mathematical tools to analyze the performance of different layering structures of SLUM under different configurations.

4.1.2 Generalized SULUM Running time Model

Our second main contribution is in deriving a generalized relative speedup mathematical model for any layering scheme taking into account all the foregoing six variables. Given any layering scheme drawn from the space D, we first derive a generalized SLUM run time absolute performance model described by the function

$$T_B(n, k, q_t, q_{1,} q_{2,} \epsilon_i) = \frac{((q_1)^k n^k + (q_2)^k (\prod_{(i=1)}^{k} n - \epsilon_i))}{(q_t)^k}$$
.

4.1.3 Generalized SLUM Relative Speedup Function

Based on the run-time model, we derived the SLUM relative speed function

$$\Omega = \frac{(q_t)^k}{((q_1)^k + (q_2^k \frac{(\prod_{(i=1)}^{k} n - \epsilon_i)}{(n^k)}))}$$

as . This model shows that for any layering scheme, generally Ω grows exponentially in k as visualized in figure 2, generally inversely proportional to q_1 as shown in figure 2 . Because the condition $q_1 + q_2 = q_t$ must hold for any layering scheme, we conclude that if Ω increases as q_1 reduces, then Ω generally increases with an increase in q_2 . The conclusions about the relationships between Ω and q_1 and Ω and q_2 are also numerically illustrated in the Ω matrix in table 3. In general, a SLUM layering scheme in which $q_1 < q_2$ is on average more efficient than one in which $q_1 > q_2$

4.1.4 Definition of the Concept of Composite Service Phase Transition Ratio and its Significance

Because the parameter ϵ_i is variable across work flow tasks and from time to time, this study makes a fundamental contribution, which is the definition of a normalized parameter, p known as the composite service phase Transition Ratio. The study then showed that Ω is inversely proportional to p as implied in the model

$$\Omega = \frac{(q_t)^k}{(q_1^k + \rho q_2^k)}$$

and also illustrated graphically in figure 2.The parameter p provides a normalized method of accounting for and quantifying the overall effect of service elimination on Ω . Given that p lies on the interval $[0,1]$ regardless of the variability in the number of web-services eliminated, at design time, systems architects, systems engineers and any interested stakeholder can easily predict the dynamics of Ω as p is varied between $[0,1]$. Similarly, at run-time, having observed a given Ω value, one can more easily associate the Ω value with systems p . This methodology is more intuitive than having to recall a series of k ϵ_i values that account for Ω .

4.1.5 Alternative Method for Quantifying Magnitude of Service Elimination

Our fourth contribution is an alternative method of quantifying the effect of service elimination at the SCUM layer on Ω - the mean number of successful service providers per work-flow task m . By definition of p , the study has shown that m is given by $n \times \sqrt[k]{\rho}$ .Thus $\prod_{(i=1)}^{k} n - \epsilon_i$. Analyzing the effect of service elimination on Ω in terms of m could be useful in understanding how many service providers do meet and how many do satisfy service consumer requirements on average at any given time and how does that contribute the efficiency of the SLUM model when operating work-flows. Beyond efficiency analysis, this could be useful in other business intelligence and analytics use cases.

4.1.6 Derivation of a Family of Mathematical Curves showing SLUM Relative Speedup Elasticity Limits

The study sought to determine the relative speedup limits of SLUM in general and the relative speedup range of a given SLUM scheme. Towards these goals, the study established that the relative speedup function for a known SLUM layering scheme $(q_{1,}q_2)$ is generally given by the function $\Omega=\frac{(q_t)^k}{(q_1^k+\rho q_2^k)}$. From we conclude that as p tends to 0, the relative speedup elastically grows larger hitting maximum possible Ω limit of $\Omega_{max}=\frac{q_t^k}{q_1^k}$ at $p=0$, whereas as p moves closer to 1, Ω grows smaller hitting a minimum possible Ω limit of $\Omega_{min}=\frac{q_t^k}{(q_1^k+q_2^k)}$ at $p=1$. These conclusions mean for example, that given eight web-service QoS attributes considered as key, if the scheme $1{,}7$ is selected, then service consumers would experience system execution speeds that are nearly 64 times better than when the service composition was operated by the S-MIP model, if those service requests led to a p that is approximately zero. Under the same scheme, service consumers would experience a relative speedup of only 1.28 times if the service consumer quality of service constraints were such that all or nearly all the service providers met all the SCUM requirements leading to a p value equal to or approximately equal to 1. However, given that $p=0$ signifies infeasibility at the SCUM layer and therefore overall infeasibility, attaining exactly the maximum speed up of $\Omega_{max}=\frac{q_t^k}{q_1^k}$ is therefore impossible practically. Nevertheless, ρ values sufficiently closer to 0 would yield a Ω that is approximately $\Omega_{max}=\frac{q_t^k}{q_1^k}$.

The study went ahead to explore the relative speedup elasticity characteristics of a special kind of layering scheme in which the scheme $(q_{1,}q_2)$ is such that $q_1=q_2$ - the balanced SLUM layering scheme. We established that under this scheme, Ω is independent of the parameters $q_{1,}q_{2,}q_t$. That is we showed that under a balanced layering scheme $\Omega=\frac{2^k}{(1+\rho)}$. We conclude that when $q_1=q_2$, the size of q_t is inconsequential on Ω. The equation $\Omega=\frac{2^k}{(1+\rho)}$ connotes a relative speedup on the range $[2^{(k-1)},2^k]$ at $q_1=q_2$. The lower bound $2^{(k-1)}$ in $2^{(k-1)}[2^{(k-1)},2^k]$ was the result of our previous work in [5].The contribution in this paper is deriving a generalized model $q_1=q_2$ in the presence of service elimination and consequently showing the upper bound as 2^k.

Another problem that this paper investigated was the global maximum and global minimum relative speedup of SLUM across the layering design space. From $\Omega_{max}=\frac{q_t^k}{q_1^k}$, we showed that the global maximum Ω is attainable at $q_1=1$, yielding $\Omega=(q_t)^k$ as the maximum possible. Thus the largest possible speedup is attainable under the layering scheme $(1,(q_t-1))$. On the other hand, we showed that either the scheme $(1,(q_t-1))$ or the scheme $((q_t-1),1)$ would yield the lowest possible Ω value of $\frac{(q_t^k)}{(1+(q_t-1)^k)}$.

From the analysis of all the foregoing, we also conclude that for a given pair of layering schemes (q_1,q_2) and (q_2,q_1) provided $p=1$,then $\Omega(q_{1,}q_2)=\Omega(q_{2,}q_1)\,,q_1\neq q_2$. The significance of this dynamics is that if for a given business application we could predict with certainty at design time that p is going to be nearly 1 most of the time, if not all the time, then, swapping of the sizes of the two layers has no relative advantage in terms of performance efficiency.

The other conclusion the study makes is that for an even numbered q_t, provided $p=1$, given a D, the balanced SLUM layering scheme gives the maximum possible Ω. For example, from table 1, the scheme $(4{,}4)$ gives the maximum possible Ω of 2. In other words, the maximum super-linear speedup is $2^{(k-1)}$. This particular observation would be interesting in scenarios where there is insignificant differentiation on the quality of service attributes across web services and hence across the different service providers. When there is little differentiation and nearly all the service providers could satisfy the SCUM constraints, then $p \to 0$ would be almost guaranteed. Under these scenario, the system architect should consider a balanced SLUM layering scheme. Given that the lowest ever achievable SLUM speedup has been shown to be $\frac{(q_t^k)}{(1+(q_t - 1)^k)}$ achievable at $p = 1$, and the maximum possible super-linear linear speedup is $2^{(k-1)}$, another major contribution of this paper is in showing that the super-linear linear speedup across the space D stretches between $\frac{(q_t^k)}{(1+(q_t - 1)^k)}$ and $2^{(k-1)}$.

4.2 Ongoing and Future Work

At design time, given a D containing $(q_{t-}1)$ possible SLUM layering schemes, selecting an optimal scheme from D is a significant problem. This problem is relevant to a system architect interested in determining the scheme from D with the most optimal efficiency benefits. The question is not a straight forward problem since for any chosen layering scheme, at design time, it's difficult to anticipate the exact magnitude of service elimination at the SCUM phase. The magnitude of service elimination is a function of the constraint inequalities of the optimization problem specification at the SCUM layer whose R.H.S values can only be instantiated at run time as dictated by the service consumer. Moreover, the constraint inequalities could be instantiated with different R.H.S values varying from time to time across service consumers. The different instances of the inequalities could lead to different permutations of the number of web-services eliminated per work flow task from time to time. Therefore the number of web services eliminated is a random variable. Hence finding an optimal layering scheme question is in itself a significant problem. We are currently working a mathematical formulation of the problem as well as exploring possible solutions to the problem.

It would be desirable in the near future to empirically validate some of the theoretical models formulated in this study.

REFERENCES

[1] Rabelo R., Gusmeroli S. The ECOLEAD collaborative business infrastructure for networked organizations networked organizations. Pervasive collaborative networks PRO-VE 2008. Springer, New York, 2008.

[2] Rabelo J. Ricardo et al (2007). An Evolving Plug and Play Business Infrastructure for Networked Organizations. International Journal of on Information Technology and Management, 2007.

[3] Abiud W. Mulongo, Elisha T. Omulo and William O. Odongo (2015). A Hierarchical Multilayer Service Composition Model for Global Virtual Organizations, Computer Science and Information Technology 3(4):91-104, 2015.

[4] biud Wakhanu Mulongo, Elisha T.O Opiyo, Elisha Abade, William Okello Odongo and Bernard Manderick (2016): SLUM: Service Layered Utility Maximization Model to Guide Dynamic Composite Webservice Selection in Virtual Organizations, Computer Science and Information Technology Vol. 4, No. 2, 2016.

[5] Abiud W. Mulongo et al (2016): Superlinear Relative Speedup of the Service Layered Utility Maximization Model for DynamicWebservice Composition in Virtual Organizations, International Journal of Computer Applications & Information Technology, Vol. 5, No. 4 , July, 2016.

[6] Benatallah B. et al (2005). QoS-Aware Middleware for Web Services Composition. IEEE Transactions on Software Engineering, Vol. 30, No. 5, May 2005.

[7] Mahboobeh M. and Joseph G.D (2011). Service Selection in Web service Composition. A comparative Review of Existing Approaches, Springer- Verlag Berlin, Heidelberg, 2011.

[8] Bin Xu et al (2011). Towards Efficiency of QoS driven semantic web service composition for large scale service oriented systems, Sringer, 211, DOI 10.1007/s11761-011-0085-8.

[9] Singh K.A (2012). Global Optimization and Integer Programming Networks. International Journal of Information and Communication Technology Research.

[10] Peter Bartalos, M´ariaBielikova (2011). Automatic Dynamic Web Service Composition: A Survey and Problem Formalization, Computing and Informatics Journal, Vol. 30, 2011, 793–827.

[11] Chiang Mung. (2006). Layering as Optimization Decomposition, Electrical Engineering Department, Princeton University. Also available online at http://www.ece.rice.edu/ctw2006/talks/ctw06-chiang.pdf.

[12] Chiang M. et al. (2006). Layering as Optimization Decomposition. Current Status and Open Issues, Electrical Engineering Department, Princeton University.

[13] Chiang M. et al. Layering as Optimization Decomposition. Ten Questions and Answers, available athttp://web.stanford.edu/class/ee360/previous/suppRead/read1/layer_1.pdf.

[14] Steve Low (2013). Scalable Distributed Control of Networks of DER, Computing & Math Sciences and Electrical Engineering, Cal-tech University.

[15] Abiud W. Mulongo (2016). A Two Layer Mixed Integer Programming Model for Dynamic Composite Web-service Selection in Virtual Organizations Inspired by Layering as Optimization Decomposition, PhD Thesis, University of Nairobi, 2016.