An Optimal Model and Solution for Resource Scheduling in Cloud Computing

Qiang Li
College of Computer Science
Sichuan University
Chengdu, China

***Abstract*-Resource scheduling based on SLA (Service Level Agreement) in cloud computing is NP-hard problem. There is no efficient method to solve it. This paper proposes an optimal model for the problem and gives a solution by applying stochastic integer programming technique. Applying Gröbner bases theory for solving the stochastic integer programming problem and the experimental results of the implementation are also presented.**

Keywords-Gröbner Bases, SLA, Stochastic Integer Programming, Cloud Computing

I. INTRODUCTION

Cloud computing is a resource delivery and usage model to get resource (hardware, software, applications) via network ”on-demand” and ”at scale” as services in a multi-tenant environment. The network of providing resource is called Cloud. All resources in the cloud are scalable infinitely and used whenever as utility. In practice of cloud computing, providing an optimal/appropriate resource to user becomes more and more important.

In cloud computing, each application is often designed a business process which includes a set of abstract services. Each abstract service encapsulates the function of an application component using its interface, and a concrete service(s) or resource(s) is selected (bound) at runtime to fulfill the function. A Service Level Agreement (SLA) in cloud computing is defined upon a business process as its end-to-end Quality of Service (QoS) constraints since a business process defines how abstract services interact to accomplish a certain business goal. Since different concrete services may operate at different QoS measures, an appropriate/optimal set of concrete services/resources may be selected so that it guarantees the fulfillment of SLA and cost is minimal. Such problem, the QoS-aware service composition problem, is a combinatorial optimization problem which ensures the optimal mapping between each abstract service and available resources [1], [2]. Since the problem is known as NP-hard[3], it takes a significant amount of time and costs to find optimal solutions (optimal combinations of resources) from a huge number of possible solutions.

This paper proposes an optimization model based on stochastic integer programming which address the SLA-aware resource composition problem. We define a resource composition model based on stochastic integer programming and provide an algorithm to solve stochastic integer programming problem. This paper is organized as follows. Section 2 shows the problem of resource composition and section 3 proposes a model of a resource composition based on stochastic integer programming technique. Section 4 describes the algorithm for solving stochastic integer programming based on Gröbner Bases theory [4], [5]. Section 5 presents simulation results to evaluate the algorithm. Sections 6 and 7 conclude with some discussion on related work.

II. SLA-BASED RESOURCE COMPOSITION PROBLEM

We define the SLA-based resource composition problem with the following assumptions:

(1) A SLA between user and cloud provider: a service agreement on
throughput, latency and cost.

(2) A business process instance: it realizes users request.

(3) A series of abstract services: it executes the business process.

(4) Concrete service/Resource: it implements an abstract service.

(5) QoS for each concrete service/resource: it has three attributes:
throughput, latency and cost, while throughput and latency can
vary at runtime.

The problem is how to select concrete service/resource to realize abstract service in business process instance that satisfies SLA, while the overall cost is minimal. The resource composition problem is also shown as Figure 1.

III. MODEL FOR SLA-BASED RESOURCE SCHEDULE

Applying stochastic integer programming technique, we give a model for SLA-based resource schedule. With this model, we can find the optimal resource schedule to realize a business process and satisfies SLA.

$$\text{Min} \sum_{i=1}^{|\alpha|}\sum_{j=1}^{|\beta_i|} c_{ij}x_{ij} \quad \text{subject to} \tag{1}$$

$$\sum_{i=1}^{|\alpha|}\sum_{j=1}^{|\beta_i|} c_{ij}x_{ij} \le C_{SLA} \tag{2}$$

$$\sum_{j=1}^{|\beta_i|} x_{ij} = 1, i \in \{1,\cdots,|\alpha|\}, \tag{3}$$

Pro

Published by Atlantis Press, Paris, France.

$$\left\{ \begin{array}{c} \min\{\xi_{ij}^{t} x_{ij}; i \in \{1,\cdots,|\alpha|\}, j \in \{1,\cdots,|\beta_i|\}, x_{ij} \neq 0\} \geq T_{SLA} \\ \sum_{i=1}^{|\alpha|} \sum_{j=1}^{|\beta_i|} \xi_{ij}^{l} x_{ij} \leq L_{SLA} \end{array} \right\} \geq \gamma \quad (4)$$

$$x_{ij} \in \{0,1\} \quad (5)$$

Where

α : *a set of Abstract services*

$|\alpha|$: *number of elements in set* α

β_i : *a set of resource available to implement i-th abstract service*

$|\beta_i|$: *number of elements in set* β_i

$$x_{ij} = \begin{cases} 1\text{:selsect jth resource for ith adstract} \\ 0\text{:otherwise} \end{cases}$$

c_{ij} : : *cost of j-th resource for implementing i-th abstract service*

C_{SLA} : : *cost of SLA*

T_{SLA} : : *throughput of SLA*

L_{SLA} : *latency of SLA*

ξ_{ij}^{t} : *random variable for resource's throughput*

ξ_{ij}^{l} :*random variable for resource's latency*

γ : *probability of fulfillment of agiven SLA*

Formula (1) means that the overall cost is minimal under the solution of variables x_{ij} . Formula (2) means that overall cost is less than or equal to SLA's cost. Formula (3) means that only one resource is selected to implement an abstract service and formula (4) means that probability of fulfillment of SLA's two attributes is great than or equal to γ .

The solution of above model is discussed in next section.

IV. GRÖBNER BASES FOR STOCHASTIC INTEGER PROGRAMMING

In this section we first introduce Gröbner Bases for integer programming (IP) and then extend it to solve stochastic integer programming (SIP).

Consider the following model of IP problems:

$$IP_{A,C}(b) = \min\{Cx : Ax = b, x \in N^n\}$$

where C is an n-vector of real numbers, A is an mxn matrix of integers and b is an m-vector of integers. This model means that we solve variables x under the constraints Ax = b so that the value of Cx is minimal. We use $IP_{A,C}$ to denote a generic IP problem.

The method for solving IP via Gröbner Bases was firstly introduced by Conti et al. in commutative algebra [6] and by Thomas in geometry [11] independently. The key idea is to encode an IP problem into a special ideal associated with the constraint matrix A and the cost (objective) function Cx. An important property of such an encoding is that its Gröbner bases correspond directly to the test sets of the IP problem. Thus, by employing an algebraic package such as MACAULAY or MAPLE, the test sets of the IP problem can be directly computed. Using a proper test set (such as the minimal test set which corresponds directly to the reduced Gröbner basis of the encoded ideal), the optimal value of the cost function can be computed by constructing a monotonic path from the initial non-optimal solution of the problem to the optimal solution.

In practical IP problem, the size of Gröbner basis increases quickly and the computation for it becomes expensive as the number of variables in IP becomes large. To overcome the disadvantage, Qiang et.al proposed a new algorithm for
solving IP called Minimised Geometric Buchberger Algorithm (MGBA)[9], which improves the computation of Gröbner Basis for IP problem by truncating the basis with the fixed right hand b. The experimental results of the implementing BGBA state that MGBA shows significant performance improvement comparing to Conti[6] and Thomas[11]'s algorithms.

Now we consider the class of SIP as the following form.

Min h(x) subject to

Prob $\{T_X \leq \xi\} \geq \gamma$

Ax=b

$x \in N^n$, ξ is a vector of random variables

The above model means that we solve variables x under the probabilistic constraint Prob $\{T_X \leq \xi\} \geq \gamma$ and the constraints Ax = b so that the value of h(x) is minimal.

Based on S.R.Tayur et al.'s idea[10], we apply MGBA to solve SIP problem with the following process.

(1) Divide the model into two parts

One is composed of the probabilistic constraints and some complicated constraints, called membership oracle; and the other is a simple IP after removing the membership oracle from the SIP problem, called Reduced IP (RIP).

(2) Compute the test set of RIP

We compute RIP's test set simply by using MGBA.

(3) Compute the test set of SIP

The test set of RIP provides a set of directions that can be used to trace paths from every nonoptimal solution to the optimal solution of the RIP. So, for any feasible solution of the RIP, we get an optimal solution by searching the set of directions. Simultaneously, we can also walk back from the optimal solution to every other feasible solution of the RIP by simply reversing these paths. By reversing all directions of RIP's test set, we get the test set of SIP, which was proved by S.R.Tayur et al.[10].

(4) Compute the optimal solution of SIP

We compute the optimal solution of the SIP by walking back from the optimal solution of RIP to other feasible

solutions and querying the membership oracle to check whether the reached point is feasible or not. If the reached point is feasible for the membership oracle, it is the optimal solution of SIP. Same as the test set of IP, we can prove that the search (walking back) terminates with either the optimal solution of SIP or all paths are searched, i.e., the SIP is infeasible.

V. Simulation Evaluation

We have implemented MGBA for SIP and have down several simulation by finding optimal resource schedule for business process with different abstract services and the different SLA. The simulation of resource scheduling optimization includes the following study cases:

(1) A user's process includes 2 abstract services. Each abstract service has 3 resources available to be selected, while the probability of fulfillment of SLA is 0.84 or 0.88.

(2) A user's process includes 4 abstract services. Each abstract service has 3 resources available to be selected, while the probability of fulfillment of SLA is 0.5.

(3) A user's process includes 6 abstract services. Each abstract service has 3 resources available to be selected, while the probability of fulfillment of SLA is 0.62.

(4) A user's process includes 7 abstract services. One abstract service has 5 resources available to be selected, while the others have 6 resources available to be selected. The probability of fulfillment of SLA is 0.47.

The experiment result is shown in Table 1. From the result, we can see that the computation of optimal resource schedule finished in a reasonably short time. But the time is growing exponentially with the increasing of the numbers of the services and resources. The reason is that the computation of Gröbner basis is still suffer from the fact that the size of Gröbner basis grows exponentially with the increasing of the

VI. Related Works

In [13], H.Wada et al. develop a multi-objective optimization model to tackle SLA-aware service composition problem. They consider multiple SLAs simultaneously and provided a set of solutions of equivalent quality. But their model is based on the heuristic genetic algorithm in which the performance cannot be expected.

In [14], S.Chaisiri et al. apply stochastic integer programming for resource provision optimization problem. The algorithm minimizes the total cost of resource provision in a cloud computing environment. The optimal solution is obtained by formulating and solving stochastic integer programming with two-stage recourse. However, they do not consider the notion of SLA, which is one of the most important business notion in cloud computing.

VII. Conclusion

In this paper, we have proposed a model for optimization of SLA-based resource schedule in cloud computing based on stochastic integer programming technique. By applying Gröbner bases theory, we extend MGBA to solve the stochastic integer programming with two-stage recourse, furthermore introduce a method to solve the optimal model of SLA-based resource schedule problem. The performance evaluation has been performed by numerical studies and simulation. The experimental result shows that the optimal solution is obtained
in a reasonable short time.

We plan to have several extensions as future work. The resource scheduling optimization model will be extended to tackle multi-objective optimization problems. The MGBA for stochastic integer programming will be extended to solve multi-objective stochastic integer programming problems. The overall performance of the algorithm will be improved by improving the computation of Gröbner Basis/test set for integer programming.

References

[1] J. Anselmi, D. Ardagna, and P. Cremonesi, A QoS-based Selection Approach of Autonomic Grid Services. In ACM High Performance Distributed Computing, Workshop on Service-Oriented Computing Performance, June 2007.

[2] T. Yu and K. J. Lin,Service Selection Algorithms for Composing Complex Services with Multiple QoS Constraints. In Int'l Conf. on Service-Oriented Computing. Addison-Wesley Professional, Dec 2005.

[3] G. Canfora, M. D. Penta, R. Esposito, and M. L. Villani, An Approach for QoS-aware Service Composition based on Genetic Algorithms. In Genetic and Evolutionary Computation Conference, June 2005.

[4] W. W. Adams and P. Loustaunau, An Introduction to Gr¨obner bases. American Mathematical Society, volume 3, 1994.

[5] B. Buchberger and F. Winkler (ed.), Gröbner bases and applications. Cambridge University Press, 1998.

[6] P. Conti and C.Traverso, Buchberger algorithm and integer programming. Proceedings AAECC-9, New Orleans, pp. 130–139, LNCS 539 Springer-Verlag.

[7] S. Hosten and B. Sturmfels, Grin: An implementation of Gröbner bases for integer programming. In Balas, E., Clausen, J., editors, Integer Programming and Combinatorial Optimization, pp. 207–276, LNCS 920 Springer-Verlag.

[8] P. Kall and S. W. Wallace, Stochastic Programming. John Wiley & Sons Ltd, Chichester, 1994.

[9] Q. Li, Y. K. Guo and T. Ida, Minimised Geometric Buchberger Algorithm for Integer Programming. Annals of Operations Research, Vol. 108, pp. 87-109, 2001.

[10] S.R. Tayur, R.R. Thomas, and N.R. Natraj, An algebraic geometry algorithm for scheduling in the presence of setups and correlated demands. Mathematical Programming, 69(3):369-401, 1995.

[11] R. R. Thomas, A geometric Buchberger algorithm for integer programming. Mathematics of Operations Research 20:864–884, 1995.

[12] R. R. Thomas and R. Weismantel, Truncated Gröbner bases for integer programming. Applicable Algebra in Engineering, Communication and Computing 8:241–257, 1997.

[13] H.Wada and K.Oba,Multi-objective Optimization of SLA-aware Service Composition. Proc. of IEEE Workshop on Methodologies for Nonfunctional Properties in Services Computing, Honolulu, 2008.

[14] S.Chaisiri, B.S. Lee, and D. Niyato,Optimal Virtual Machine Placement across Multiple Cloud Providers. IEEE APSCC 2009, Singapore Dec, 2009

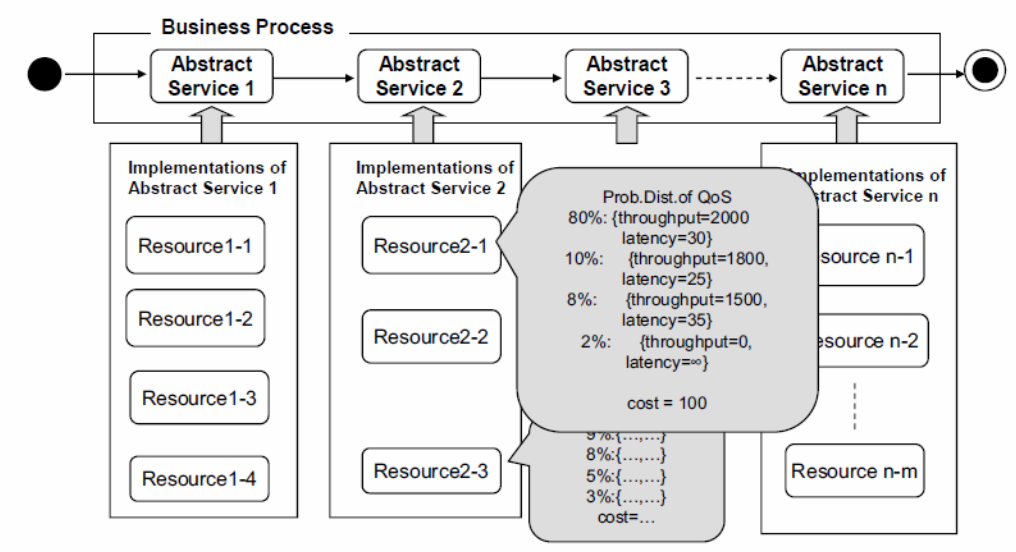

Figure 1. resource composition

TABLE I. EXPERIMENTAL RESULTS

Problems	Y	Optimal resource schedule	Time(sec)
$\|\alpha\| = 2$ $\|\beta_i\| = 3$ i=1,2	0.84	$x_{11}=0, x_{12}=0, x_{13}=0, x_{21}=0, x_{22}=0, x_{23}=0$ $x_{11}=0, x_{12}=1, x_{13}=0, x_{21}=0, x_{22}=0, x_{23}=0$ Overall $Qos: C=44, T=100, L=10$	2.36
	0.88	$x_{11}=1, x_{12}=0, x_{13}=0, x_{21}=0, x_{22}=0, x_{23}=1$ Overall $Qos: C=97, T=130, L=8$	182.98
$\|\alpha\| = 4$ $\|\beta_i\| = 3$, i=1,…,4	0.5	$x_{11}=0, x_{12}=1, x_{13}=0, x_{21}=0, x_{22}=0, x_{23}=1$ $x_{31}=1, x_{32}=0, x_{33}=1, x_{41}=1, x_{42}=0, x_{43}=1$ Overall $Qos: C=93, T=250, L=30$	235.59
$\|\alpha\| = 6$ $\|\beta_i\| = 3$, i=1,…,6	0.62	$x_{11}=0, x_{12}=0, x_{13}=1, x_{21}=0, x_{22}=0, x_{23}=0$ $x_{31}=1, x_{32}=0, x_{33}=0, x_{41}=1, x_{42}=0, x_{43}=0$ $x_{51}=0, x_{52}=1, x_{53}=0, x_{61}=0, x_{62}=1, x_{63}=0$ Overall $Qos: C=122, T=450, L=40$	433.64
$\|\alpha\| = 7$ $\|\beta_i\| = 3$, i=1,…,6 $\|\beta_7\| = 5$	0.47	$x_{11}=1, x_{12}=0, x_{13}=0, x_{21}=0, x_{22}=0, x_{23}=1$ $x_{31}=0, x_{32}=1, x_{33}=0, x_{41}=0, x_{42}=0, x_{43}=1$ $x_{51}=0, x_{52}=1, x_{53}=0, x_{61}=0, x_{62}=1, x_{63}=0$ $x_{71}=1, x_{72}=0, x_{73}=0, x_{74}=0, x_{75}=0$ Overall $Qos: C=212, T=650, L=80$	1219.18