Service Provider Selection of IAAS using Naive Bayes Approach

JAY VALA
Computer Engineering Department
L. D. College of Engineering
Ahmedabad Gujarat, India

KALPESH PATEL
Assistant Professor
Computer Engineering Department
L. D. College of Engineering
Ahmedabad Gujarat, India

`

Abstract

Cloud is a new trend of computing. This type of computing resources are pay-as-you-go plans ,As more and more consumer delegate their task to cloud provider, service level agreement are necessary. A relationship between cloud provider and cloud consumer must be described with SLA. Clouds have dynamic nature of providing services so comparison of SLA template should be continuous checking of quality of services are very important to enforce SLAs.SLA template contain many parameter like(Availability ,Response time etc.).This work addresses to find probability of SLA parameter to find suitable Service Provider for Service Consume6r requirement.

Index Terms: Cloud Computing, SLA, Service provider Capability Model, Service Consumer Requirement Model.

1. Introduction

Cloud computing is a new trend of computing. Here readily available resources like computing are exposed as services. This type of computing resources are pay-as-you-go plans and because of that it has become attractive to cost conscious customer. While cloud computing is a metaphor for the internet. Cloud computing is complex infrastructure of hardware, software, processing, and storage that is available as a service. Cloud Computing resides on personal computer and local servers is comprised essentially of application running remotely. A relationship between cloud provider and cloud consumer must be described with Service Level Agreement. Service level Agreement is a part of the contract between service consumer and service provider, and formally defines the levels of service. It is vital to define those services, how they are delivered and how they are used.

An SLA contains several parameters like a set of properties and resources the service provider will deliver, a specific definition of each resource, the responsibilities of the provider and the consumer, a resource quantity, an auditing mechanism to monitor the service etc [1].

Managing and monitoring the quality levels of services rely heavily on automated tools, at present the actual Cloud SLAs are typically plain-text document. Example of a document with legal obligations is the Amazon S3 Service level Agreement [2], Amazon EC2 Service Level Agreement [3]. Finding correspondences between multiple elements is required in many application scenarios and is often referred to as matching. This work aims to define the process of identifying compatible service provider for a given requirements by finding probability SLA parameters.

SLA parameters of the cloud are defined in a service provider capability model and application's requirements are defined in a service consumer application requirement model. Capability model and application requirement model are defined in XML format and it is a W3C Recommendation. To identify the compatible service provider Naive Bayes algorithm is used [5].

2. Related Work

At present the actual Cloud SLAs are typically plain-text document. Example of a document with legal obligations is the Amazon S3 Service level Agreement [2], Amazon EC2 Service Level Agreement [3]. Consumer needs to manually match application requirements with each and every service provider to identify compatible service provider. Proposed algorithm identifies the compatible service provider. It gives suggestion to a consumer in terms of name of service provider with highest probability among existing service provider.

3. Proposed Approached

The first step is to define cloud model (XML file) that contains cloud resources, properties, resource quantity etc. and requirement model (XML File) that contains application's required resources and its quantity. Let's consider a service consumer model as a 'Consumer Model' and service provider model as a 'Provider Model'.

Second step is to collect cloud provider model and make a data set for prediction of service provider based on service consumer requirement. Consumer Model and Provider Model uses parameter of IAAS which is defined [4].

Third step is to apply Naive Bayes classification algorithm. By using mention algorithm we can find out probability of each and every service provider based on service consumer requirement.

At last step we suggest best suitable service provider among all service provider to service consumer based on highest probability.

4. Algorithm

In this section algorithms are described.

4.1 SLA Parameter Processing

SLA Parameter Processing is the procedure that takes requirement model and capability model as a input and gives a suggestion in terms of result.

4.1 SLA Parameter Processing

INPUT: XML File, Requirement Model, Capability Model
OUTPUT: Result

1: Load XML File
2. Load Requirement Model,
3. **For all** capability Model **do**
4. Load capability Model
5. Call NaiveBayes
6. Add to Result
7. **End For**

4.2 Naive Bayes Algorithm

Naive Bayes Algorithm does process on Requirement Model and at last find probability of each Requirement based on Capability Model.

4.2 NaiveBayes Algorithm

INPUT: Requirement Model, Capability Model, D be a training set of tuple
OUTPUT: Result List

Procedure Naive Bayes (D)

1. X represent a vector attribute, C represent the classes.
2. **if** Class Prior probabilities are not known **then**
3. Classes are equally likely that is P(C1)=P(C2)=P(C3)=.....=P(Cm)
4. **else**
5. Find P(Ci)= $|Ci, D|$ / $|D|$ where $|Ci, D|$ is number of Training tuple of Class Ci in D
6. **end if**
7. Find Class Conditional Independence $P(X|Ci) = \pi P(Xk|Ci)$
8. Find $P(Ci|X) = (P(X|Ci)*P(Ci))/P(X)$
9. **if** $P(Ci|X) > P(Cj|X)$ for $1<=j<=m, j!=i$ **then**
10. Naive Bayesian classifier predicts that tuple x belongs to class Ci
11. **end if**
12. **if** $P(X|Ci)P(Ci) > P(X|Cj)P(Cj)$ for $1<=j<=m, j!=i$ **then**
13. The classifier predicts that the class lable of tuple X is in the class Ci.
14. **end if**

5. Experiments and Results

5.1 Experimental Setup

Proposed Algorithm implemented in .Net. For analysis 10 capabilities Model and 6 requirements Model with SLA parameter listed below were considered: Storage (HD), Boot Time, Memory Size (RAM), and Scale up (VM), Scale down (VM), Availability, Response Time, CPU Capacity, Log, Latency per User, Turnaround Time, Tolerance limit, Uptime Guarantee.

Table1 Instance count of SLA Parameter

Capability Model of Service Provider													
Provider Name	Storage (HD)	Boot time	Memory Size (RAM)	Scale Up (VM)	Scale Down (VM)	Availability	Response Time	CPU Capacity	Turnaround time	Tolerance limit	Uptime Guarantee	Log	Latency per million users
Unit	TB	mili second	GB	Machine	Machine	Percentage	Speed	GHz	Second	Percentage	Percentage	Yes/no	second
Cloud switch	3	2	1	2	1	3	2	2	2	1	1	0	1
Google	1	2	1	2	1	1	2	1	1	1	2	0	1
Amazon S3	2	1	2	1	1	2	2	1	3	3	2	1	2
IBM	2	2	2	3	1	1	1	1	1	1	1	0	1
Rack space	1	2	1	2	1	1	1	1	2	1	1	0	1
Amazon EC2	3	2	1	2	1	2	1	1	2	1	1	0	1
3Tera	1	2	1	2	1	1	1	2	1	2	1	1	1
AT&T	3	2	2	3	1	1	1	1	2	1	1	0	1
Azure Storage	3	2	1	2	1	2	1	1	2	1	2	1	2
Cloud scale	1	1	1	3	1	1	1	1	2	1	1	0	1

Table2 Instance count of Application SLA Parameter

Requirement Model of Service consumer													
Consumer Requirement	Storage (HD)	Boot time	Memory Size (RAM)	Scale up(VM)	Scale Down (VM)	Availability	Response Time	CPU Capacity	Turnaround time	Tolerance limit	Uptime Guarantee	Log	Latency per million users
Unit	TB	mili second	GB	Machine	Machine	Percentage	Speed	GHz	Second	Percentage	Percentage	Yes/no	second
Requirement1	3	1	1	2	1	2	2	2	2	1	1	0	1
Requirement2	1	2	1	2	1	1	2	1	1	2	2	0	1
Requirement3	2	1	1	1	1	2	1	1	2	3	2	1	2
Requirement4	2	2	3	2	1	1	1	2	1	1	1	0	1
Requirement5	1	2	1	2	1	1	1	1	2	2	1	0	1
Requirement6	3	2	1	2	1	2	3	1	2	3	1	0	1

.

Table 1 shows the instance count of SLA parameter that is declared in capability Model. Table2 shows the instance count of SLA parameter that is declared in Requirement Model.

5.2 Results

SLA Parameter matching with Naive Bayes Classifier gives probabilistic solution. For all requirements Model of service consumer first find probability for each capability of service provider. Among all, which service provider have highest probability will be the suggested service provider. Figure 1 shows results of suggested service provider based on requirement of service consumer. Its value range is between 0 and 1.

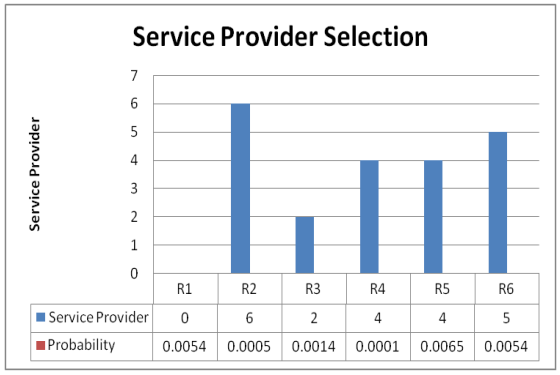

Figure 1: **Result of Requirement of service Consumer**

6. Conclusion

A simple algorithm is proposed here which is based on probabilistic and useful for finding probability for SLA parameter to identify compatible service provider. It gives suggestion to consumer in terms of highest probability among all service providers. This algorithm can be used as an independent tool or it can also be used as a plug in to any middleware.

7. References

[1] Mohammed Alhamad, Tharam Dillon, Elizabeth Chang, Conceptual SLA Framework for Cloud Computing
[2]Amazon simple storage service (s3) service level agreement. Amazon.[Online]. Available: http://aws.amazon.com/s3-sla/
[3]Amazon elastic compute cloud (ec2) service level agreement.amazon.[Online].Available: http://aws.amazon.com/ec2-sla/
[4]Intangible SLA skeleton for cloud computing, Anirrudh Dodiya, Jay Vala, Kalpesh Patel.
[5]Service Level Agreement Parameter matching in Cloud Computing, Tejas Chauhan, Sanjay Chaudhary, Vikas Kumar and Minal Bhise.