AN SLA-AWARE CLOUD COALITION FORMATION APPROACH FOR VIRTUALIZED NETWORKS

BOODIDA BHANUTEJA[1], S. NAGARAJU[2], SHAIK SUBAHAN[3]

ABSTRACT: The authors proposed the in-tegration of SDN network programmability mechanisms into cloud environments, in order to provide SLA guar-antees for users of virtualized networks. The aim of the proposed approach is to enforce user-defined SLA re-quirements within single and federated cloud domains, while offering an SLA negotiation mechanism between clients and cloud providers. The solution proposed con-sists of a three-tier middleware, dubbed "ProgNET" that consists of a front end, an SLA manager, and a cloud man-ager. The authors envision their middleware as a comple-ment to cloud federation approaches, as means to custom-ize the behavior of the formed networks. The proposed solution consists of a high level architecture that was not implemented nor evaluated. Furthermore, it does not ad-dress the issue of cloud federation formation, focusing solely on adding mechanisms for negotiating and enforc-ing SLA requirements in cloud environments. Contrail [28] is another solution proposed to provide SLA manage-ment support for federated clouds. In the Contrail plat-form, an SLA management layer is added to each cloud provider to manage the clients SLA requirements. This approach differs from our approach in that SLA require-ments are not considered during the federation for-mation, while SLA management is added as additional overhead function to each cloud domain. By considering SLA requirements as a constraint in the cloud federation formation process, we ensure that the selected CPs have the necessary resources and capacity to meet the client re-quirements without delays or degraded performance. On the other hand, not considering SLA in the federation pro-cess and adding it during the operation phase may still lead to resources' shortages, delays and performance deg-radation, while providers try to meet variable clients' re-quirements.

INTRODUCTION & OBJECTIVE

A. PROBLEMS EXISTING SYSTEM

Existing cloud federation approaches fail to consider clients' SLA requirements during the coalition formation process or provide a self-healing mechanism to deal with unexpected resources' shortage during operation. Furthermore, the state of the art approaches suffer from performance issues, such as high execution times, unstable performance, and lack of convergence to a solution in complex scenarios (e.g. requests with mixed, independent types of VMs). This paper proposes a novel social gaming based approach for coalition formation in the cloud that finds the best coalition of cloud providers to answer requests, while satisfying the clients' SLA requirements. The proposed algorithm, dubbed SLA Aware Cloud Coalition Formation algorithm (S-ACCF), leverages Irving's roommate algorithm to form a stable coalition of cloud providers, with a rapid execution time. The S-ACCF algorithm is designed to maximize the coalition's profit, while minimizing the number of participants in the coalition as well as the penalty incurred by providers who fail to offer all or some of the promised resources using a self-healing process.

B. SOLUTION OF THESE PROBLEMS

The authors proposed the in-tegration of SDN network programmability mechanisms into cloud environments, in order to provide SLA guar-antees for users of virtualized networks. The aim of the proposed approach is to enforce user-defined SLA re-quirements within single and federated cloud domains, while offering an SLA negotiation mechanism between clients and cloud providers. The solution proposed con-sists of a three-tier middleware, dubbed "ProgNET" that consists of a front end, an SLA manager, and a cloud man-ager. The authors envision their middleware as a comple-ment to cloud federation approaches, as means to custom-ize the behavior of the formed networks. The proposed solution consists of a high level architecture that was not implemented nor evaluated. Furthermore, it does not ad-dress the issue of cloud federation formation, focusing solely on adding mechanisms for negotiating and enforc-ing SLA requirements in cloud environments.

SOFTWARE REQUIRMENT SPECIFICATIONS

The software, Site Explorer is designed for management of web sites from a remote location.

INTRODUCTION

Purpose: The main purpose for preparing this document is to give a general insight into the analysis and requirements of the existing system or situation and for determining the operating characteristics of the system.

Scope: Currently, there are only little information is available on the adoption of cloud computing in the Institutions in India providing Technical Education. A better understanding of the state of art Technology in this area would help in the modernization of technical

educational organizations in India and the development of more appropriate cloud products and new business models for this role. For this purpose, a review about cloud computing and IT models was carried out. Also an empirical study using an online questionnaire at various Universities, Institutions and Colleges carried out to find the current state of cloud and future scope of cloud computing at Indian organization softechnicaleducation.

Content	Description
OS	Windows XP with SP2 or Windows 8 Above
Database	MS-SQL server 2014
Technologies	ASP.NET with C#.NET
IDE	Ms-Visual Studio .Net 2015
Browser	IE

SYSTEM DESIGN

INTRODUCTION

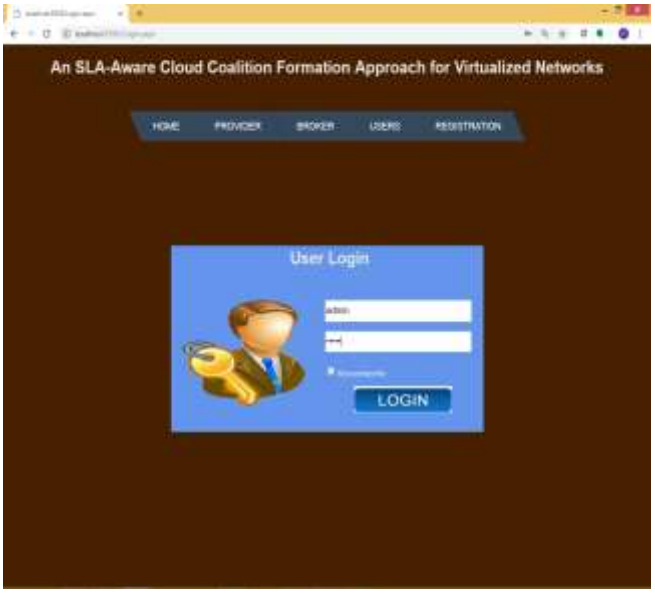
Software design sits at the technical kernel of the software engineering process and is applied regardless of the development paradigm and area of application. Design is the first step in the development phase for any engineered product or system. The designer's goal is to produce a model or representation of an entity that will later be built. Beginning, once system requirement have been specified and analyzed, system design is the first of the three technical activities -design, code and test that is required to build and verify software.

The importance can be stated with a single word "Quality". Design is the place where quality is fostered in software development. Design provides us with representations of software that can assess for quality. Design is the only way that we can accurately translate a customer's view into a finished software product or system. Software design serves as a foundation for all the software engineering steps that follow. Without a strong design we risk building an unstable system – one that will be difficult to test, one whose quality cannot be assessed until the last stage.

During design, progressive refinement of data structure, program structure, and procedural details are developed reviewed and documented. System design can be viewed from either technical or project management perspective. From the technical point of view, design is comprised of four activities – architectural design, data structure design, interface design and procedural design.

OUTPUT SCREENS

Screens:




An SLA-Aware Cloud Coalition Formation Approach for Virtualized Networks
HOME
ACCEPTREQUEST
VMINSTANCES
AVAILABLERESOURCE
PREFERENCELISTS
LOGOUT
GRAPHS
Registered Employees
UserId
FirstName
CurrentLocation
FuctionalArea
1
raju
Ammerpet
ammerpet
Personal Accept


An SLA-Aware Cloud Coalition Formation Approach for Virtualized Networks
HOME
ACCEPTREQUEST
VMINSTANCES
AVAILABLERESOURCE
PREFERENCELISTS
LOGOUT
GRAPHS


An SLA-Aware Cloud Coalition Formation Approach for Virtualized Networks
HOME
ACCEPTREQUEST
VMINSTANCES
AVAILABLERESOURCE
PREFERENCELISTS
LOGOUT
GRAPHS


An SLA-Aware Cloud Coalition Formation Approach for Virtualized Networks
HOME
PROVIDER
BROKER
USERS
REGISTRATION
User Login
LOGIN


An SLA-Aware Cloud Coalition Formation Approach for Virtualized Networks
HOME
ACCEPTREQUEST
VMINSTANCES
AVAILABLERESOURCE
PREFERENCELISTS
LOGOUT
GRAPHS
available resources
provider VM costs


An SLA-Aware Cloud Coalition Formation Approach for Virtualized Networks
HOME
ACCEPTREQUEST
PREFERENCELISTS
SENDMESSAGE
SELECTINFO
LOGOUT
CHARTS


An SLA-Aware Cloud Coalition Formation Approach for Virtualized Networks
HOME
ACCEPTREQUEST
PREFERENCELISTS
SENDMESSAGE
SELECTINFO
LOGOUT
CHARTS
Preferences Lists


An SLA-Aware Cloud Coalition Formation Approach for Virtualized Networks
HOME
ACCEPTREQUEST
PREFERENCELISTS
SENDMESSAGE
SELECTINFO
LOGOUT
CHARTS


An SLA-Aware Cloud Coalition Formation Approach for Virtualized Networks
HOME
ACCEPTREQUEST
PREFERENCELISTS
SENDMESSAGE
SELECTINFO
LOGOUT
CHARTS
ADD
SEND
Search


An SLA-Aware Cloud Coalition Formation Approach for Virtualized Networks
HOME
PROVIDER
BROKER
USERS
REGISTRATION
User Login
LOGIN


An SLA-Aware Cloud Coalition Formation Approach for Virtualized Networks
HOME
ACCEPTREQUEST
PREFERENCELISTS
SENDMESSAGE
SELECTINFO
LOGOUT
CHARTS


An SLA-Aware Cloud Coalition Formation Approach for Virtualized Networks
HOME
USERREQUEST
RESPONSESTATUS
USERS
LOGOUT
Cloud user request
Submit

SYSTEM TESTING IMPLEMENTATION

IMPLEMENTATION

Implementation is the process of converting a new or revised system design into operational one. There are three types of Implementation:

- Implementation of a computer system to replace a manual system. The problems encountered are converting files, training users, and verifying printouts for integrity.
- Implementation of a new computer system to replace an existing one. This is usually a difficult conversion. If not properly planned there can be many problems.
- Implementation of a modified application to replace an existing one using the same computer. This type of conversion is relatively easy to handle, provided there are no major changes in the files.

Implementation in Generic tool project is done in all modules. In the first module User level identification is done. In this module every user is identified whether they are genuine one or not to access the database and also generates the session for the user. Illegal use of any form is strictly avoided.

In the Table creation module, the tables are created with user specified fields and user can create many table at a time. They may specify conditions, constraints and calculations in creation of tables. The Generic code maintain the user requirements through out the project.

In Updating module user can update or delete or Insert the new record into the database. This is very important module in Generic code project. User has to specify the filed value in the form then the Generic tool automatically gives whole filed values for that particular record.

In Reporting module user can get the reports from the database in 2Dimentional or 3Dimensional view. User has to select the table and specify the condition then the report will be generated for the user.

Conclusion:

This project proposed a new approach for coalition formation in the cloud, to provide needed resources for given user requests. The coalition formation problem was formulated as a hedonic coalitional game with transferable utility. The main objectives considered are the maximization of the coalition profit, the minimization of the number of the coalition members, and the minimization of penalty when a provider fails to offer all or part of the promised resources. The new approach was evaluated using different use case scenarios and the results compared to the optimal and split-and-merge based approaches. Results analysis showed that the new approach outperforms (or gives the same output as) the other two in terms of the execution time, the number of providers per coalition, the number of VMs per provider, and the individual payoff per provider. The executed scenarios cover cases where a user is requesting one type or different types of VMs, the cases where the providers offer the same or different types of VMs, as well as the cases where the VMs are scal-ing linearly or not.

BENEFITS:

The project is identified by the merits of the system offered to the user. The merits of this project are as follows: -

- It's a web-enabled project.
- This project offers user to enter the data through simple and interactive forms. This is very helpful for the client to enter the desired information through so much simplicity.
- The user is mainly more concerned about the validity of the data, whatever he is entering. There are checks on every stages of any new creation, data entry or updation so that the user cannot enter the invalid data, which can create problems at later date.
- Sometimes the user finds in the later stages of using project that he needs to update some of the information that he entered earlier. There are options for him by which he can update the records. Moreover there is restriction for his that he cannot change the primary data field. This keeps the validity of the data to longer extent.

LIMITATIONS:

Our technique is limited to the host data, and guest OS specific information is not accessible with our method. For example, our technique could not detect a container in a VM, but it could show it as a separate process using the GTA algorithm. In contrast, other trace-based methods ([22] and [21]) provide more useful insights about running processes and their interaction with the guest kernel.

Future Scope:

Currently, there are only little information is available on the adoption of cloud computing in the Institutions in India providing Technical Education. A better understanding of the state of art Technology in this area would help in the

modernization of technical educational organizations in India and the development of more appropriate cloud products and new business models for this role. For this purpose, a literature review about cloud computing and IT models was carried out. Also an empirical study using an online questionnaire at various Universities, Institutions and Colleges carried out to find the current state of cloud and future scope of cloud computing at Indian organizations of technical education.

REFERENCES

[1] I. Hosni and N. Hamdi, "Distributed cooperative spectrum sensing with wireless sensor network cluster architecture for smart grid communications," International Journal of Sensor Networks, vol. 24, no. 2, pp. 118–124, 2017.

[2] M. Faisal and A. A. Cardenas, "Incomplete clustering of electricity consumption: an empirical analysis with industrial and residential datasets," Cyber-Physical Systems, vol. 3, no. 1-4, pp. 42–65, 2017.

[3] S. Ma, Y. Yang, Y. Qian, H. Sharif, and M. Alahmad, "Energy harvesting for wireless sensor networks: applications and challenges in smart grid," International Journal of Sensor Networks, vol. 21, no. 4, pp. 226–241, 2016.

[4] G. Xu, W. Yu, D. Griffith, N. Golmie, and P. Moulema, "Toward integrating distributed energy resources and storage devices in smart grid," IEEE Internet of Things Journal, vol. 4, no. 1, pp. 192–204, Feb 2017.

AUTHORS:

BOODIDA BHANUTEJA,

M.TECH scholar, SRI SHIRDI SAI ENGINEERING COLLEGE, ANANTAPUR.

S. NAGARAJU,

Assistant Professor, SRI SHIRDI SAI ENGINEERING COLLEGE, ANANTAPUR.

SHAIK SUBAHAN