INTERNATIONAL JOURNAL OF RESEARCH IN COMPUTER APPLICATION & MANAGEMENT

IJRCM

A Monthly Double-Blind Peer Reviewed (Refereed/Juried) Open Access International e-Journal - Included in the International Serial Directories

Indexed & Listed at:

Ulrich's Periodicals Directory ©, ProQuest, U.S.A., EBSCO Publishing, U.S.A., Cabell's Directories of Publishing Opportunities, U.S.A.,

Open J-Gage, India [link of the same is duly available at Inflibnet of University Grants Commission (U.G.C.)],

Index Copernicus Publishers Panel, Polandwith IC Value of 5.09 &number of libraries all around the world.

Circulated all over the world & Google has verified that scholars of more than 2401 Cities in 155 countries/territories are visiting our journal on regular basis.

Ground Floor, Building No. 1041-C-1, Devi Bhawan Bazar, JAGADHRI – 135 003, Yamunanagar, Haryana, INDIA

http://ijrcm.org.in/

CONTENTS

CALL FOR MANUSCRIPTS

Weinvite unpublished novel, original, empirical and high quality research work pertaining to recent developments & practices in the area of Computer, Business, Finance, Marketing, Human Resource Management, General Management, Banking, Education, Insurance, Corporate Governance and emerging paradigms in allied subjects like Accounting Education; Accounting Information Systems; Accounting Theory & Practice; Auditing; Behavioral Accounting; Behavioral Economics; Corporate Finance; Cost Accounting; Econometrics; Economic Development; Economic History; Financial Institutions & Markets; Financial Services; Fiscal Policy; Government & Non Profit Accounting; Industrial Organization; International Economics & Trade; International Finance; Macro Economics; Micro Economics; Monetary Policy; Portfolio & Security Analysis; Public Policy Economics; Real Estate; Regional Economics; Tax Accounting; Advertising & Promotion Management; Business Education; Management Information Systems (MIS); Business Law, Public Responsibility & Ethics; Communication; Direct Marketing; E-Commerce; Global Business; Health Care Administration; Labor Relations & Human Resource Management; Marketing Research; Marketing Theory & Applications; Non-Profit Organizations; Office Administration/Management; Operations Research/Statistics; Organizational Behavior & Theory; Organizational Development; Production/Operations; Public Administration; Purchasing/Materials Management; Retailing; Sales/Selling; Services; Small Business Entrepreneurship; Strategic Management Policy; Technology/Innovation; Tourism, Hospitality & Leisure; Transportation/Physical Distribution; Algorithms; Artificial Intelligence; Compilers & Translation; Computer Aided Design (CAD); Computer Aided Manufacturing; Computer Graphics; Computer Organization & Architecture; Database Structures & Systems; Digital Logic; Discrete Structures; Internet; Management Information Systems; Modeling & Simulation; Multimedia; Neural Systems/Neural Networks; Numerical Analysis/Scientific Computing; Object Oriented Programming; Operating Systems; Programming Languages; Robotics; Symbolic & Formal Logic and Web Design. The above mentioned tracks are only indicative, and not exhaustive.

Anybody can submit the soft copy of his/her manuscript **anytime** in M.S. Word format after preparing the same as per our submission guidelines duly available on our website under the heading guidelines for submission, at the email address: infoijrcm@gmail.com.

GUIDELINES FOR SUBMISSION OF MANUSCRIPT

1. **COVERING LETTER FOR SUBMISSION:**

DATED: _____

THE EDITOR
IJRCM

Subject: **SUBMISSION OF MANUSCRIPT IN THE AREA OF .**

(e.g. Finance/Marketing/HRM/General Management/Economics/Psychology/Law/Computer/IT/Engineering/Mathematics/other, please specify)

DEAR SIR/MADAM

Please find my submission of manuscript entitled '_____' for possible publication in your journals.

I hereby affirm that the contents of this manuscript are original. Furthermore, it has neither been published elsewhere in any language fully or partly, nor is it under review for publication elsewhere.

I affirm that all the author (s) have seen and agreed to the submitted version of the manuscript and their inclusion of name (s) as co-author (s).

Also, if my/our manuscript is accepted, I/We agree to comply with the formalities as given on the website of the journal & you are free to publish our contribution in any of your journals.

NAME OF CORRESPONDING AUTHOR:
Designation:
Affiliation with full address, contact numbers & Pin Code:
Residential address with Pin Code:
Mobile Number (s):
Landline Number (s):
E-mail Address:
Alternate E-mail Address:

NOTES:

a) The whole manuscript is required to be in ***ONE MS WORD FILE*** only (pdf. version is liable to be rejected without any consideration), which will start from the covering letter, inside the manuscript.
b) The sender is required to mentionthe following in the **SUBJECT COLUMN** of the mail:
New Manuscript for Review in the area of (Finance/Marketing/HRM/General Management/Economics/Psychology/Law/Computer/IT/ Engineering/Mathematics/other, please specify)
c) There is no need to give any text in the body of mail, except the cases where the author wishes to give any specific message w.r.t. to the manuscript.
d) The total size of the file containing the manuscript is required to be below **500 KB**.
e) Abstract alone will not be considered for review, and the author is required to submit the complete manuscript in the first instance.
f) The journal gives acknowledgement w.r.t. the receipt of every email and in case of non-receipt of acknowledgment from the journal, w.r.t. the submission of manuscript, within two days of submission, the corresponding author is required to demand for the same by sending separate mail to the journal.

2. **MANUSCRIPT TITLE:** The title of the paper should be in a 12 point Calibri Font. It should be bold typed, centered and fully capitalised.

3. **AUTHOR NAME (S) & AFFILIATIONS:** The author (s) **full name**, **designation**, **affiliation** (s), **address**, **mobile/landline numbers**, and **email/alternate email address** should be in italic & 11-point Calibri Font. It must be centered underneath the title.

4. **ABSTRACT:** Abstract should be in fully italicized text, not exceeding 250 words. The abstract must be informative and explain the background, aims, methods, results & conclusion in a single para. Abbreviations must be mentioned in full.

5. **KEYWORDS:** Abstract must be followed by a list of keywords, subject to the maximum of five. These should be arranged in alphabetic order separated by commas and full stops at the end.

6. **MANUSCRIPT:** Manuscript must be in ***BRITISH ENGLISH*** prepared on a standard A4 size ***PORTRAIT SETTING PAPER***. It must be prepared on a single space and single column with 1" margin set for top, bottom, left and right. It should be typed in 8 point Calibri Font with page numbers at the bottom and centre of every page. It should be free from grammatical, spelling and punctuation errors and must be thoroughly edited.

7. **HEADINGS:** All the headings should be in a 10 point Calibri Font. These must be bold-faced, aligned left and fully capitalised. Leave a blank line before each heading.

8. **SUB-HEADINGS:** All the sub-headings should be in a 8 point Calibri Font. These must be bold-faced, aligned left and fully capitalised.

9. **MAIN TEXT:** The main text should follow the following sequence:

INTRODUCTION

REVIEW OF LITERATURE

NEED/IMPORTANCE OF THE STUDY

STATEMENT OF THE PROBLEM

OBJECTIVES

HYPOTHESES

RESEARCH METHODOLOGY

RESULTS & DISCUSSION

FINDINGS

RECOMMENDATIONS/SUGGESTIONS

CONCLUSIONS

SCOPE FOR FURTHER RESEARCH

ACKNOWLEDGMENTS

REFERENCES

APPENDIX/ANNEXURE

It should be in a 8 point Calibri Font, single spaced and justified. The manuscript should preferably not exceed ***5000 WORDS***.

10. **FIGURES &TABLES:** These should be simple, crystal clear, centered, separately numbered &self explained, and **titles must be above the table/figure**. **Sources of data should be mentioned below the table/figure**. It should be ensured that the tables/figures are referred to from the main text.

11. **EQUATIONS:**These should be consecutively numbered in parentheses, horizontally centered with equation number placed at the right.

12. **REFERENCES:** The list of all references should be alphabetically arranged. The author (s) should mention only the actually utilised references in the preparation of manuscript and they are supposed to follow **Harvard Style of Referencing**. The author (s) are supposed to follow the references as per the following:

- All works cited in the text (including sources for tables and figures) should be listed alphabetically.
- Use (**ed.**) for one editor, and (**ed.s)** for multiple editors.
- When listing two or more works by one author, use --- (20xx), such as after Kohl (1997), use --- (2001), etc, in chronologically ascending order.
- Indicate (opening and closing) page numbers for articles in journals and for chapters in books.
- The title of books and journals should be in italics. Double quotation marks are used for titles of journal articles, book chapters, dissertations, reports, working papers, unpublished material, etc.
- For titles in a language other than English, provide an English translation in parentheses.
- The location of endnotes within the text should be indicated by superscript numbers.

PLEASE USE THE FOLLOWING FOR STYLE AND PUNCTUATION IN REFERENCES:

BOOKS

- Bowersox, Donald J., Closs, David J., (1996), "Logistical Management." Tata McGraw, Hill, New Delhi.
- Hunker, H.L. and A.J. Wright (1963), "Factors of Industrial Location in Ohio" Ohio State University, Nigeria.

CONTRIBUTIONS TO BOOKS

- Sharma T., Kwatra, G. (2008) Effectiveness of Social Advertising: A Study of Selected Campaigns, Corporate Social Responsibility, Edited by David Crowther & Nicholas Capaldi, Ashgate Research Companion to Corporate Social Responsibility, Chapter 15, pp 287-303.

JOURNAL AND OTHER ARTICLES

- Schemenner, R.W., Huber, J.C. and Cook, R.L. (1987), "Geographic Differences and the Location of New Manufacturing Facilities," Journal of Urban Economics, Vol. 21, No. 1, pp. 83-104.

CONFERENCE PAPERS

- Garg, Sambhav (2011): "Business Ethics" Paper presented at the Annual International Conference for the All India Management Association, New Delhi, India, 19–22 June.

UNPUBLISHED DISSERTATIONS AND THESES

- Kumar S. (2011): "Customer Value: A Comparative Study of Rural and Urban Customers," Thesis, KurukshetraUniversity, Kurukshetra.

ONLINE RESOURCES

- Always indicate the date that the source was accessed, as online resources are frequently updated or removed.

WEBSITES

- Garg, Bhavet (2011): Towards a New Natural Gas Policy, Political Weekly, Viewed on January 01, 2012 http://epw.in/user/viewabstract.jsp

A GOSSIP PROTOCOL FOR DYNAMIC LOAD BALANCING IN CLOUDS

V.VIMALA DHEEKSHANYA
STUDENT
DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
DEPARTMENT OF INFORMATION TECHNOLOGY
SRINIVASAN ENGINEERING COLLEGE
PERAMBALUR

A.RAMACHANDRAN
ASST. PROFESSOR
DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
DEPARTMENT OF INFORMATION TECHNOLOGY
SRINIVASAN ENGINEERING COLLEGE
PERAMBALUR

ABSTRACT

The components of the middleware layer run on every processing node of the cloud environment in a decentralized design. To achieve scalability, it envisions that all key tasks of the middleware layer, including estimating global states, placing site modules and computing policies for request forwarding are based on distributed algorithms. Further, it relies on a global directory for routing requests from users on the Internet to access points to particular sites inside the cloud. A gossip protocol P, executes in a middleware platform and meets the design goals. It provides an optimal solution for a simplified version of the resource allocation problem and an efficient heuristic for the hard problem. The protocol proposed continuously executes, while it's input and consequently its output dynamically changes. Hence to reduce the demand, a time and cost based slot mechanism have been implemented to convert the application into a business oriented application for cloud providers which will be efficient for cloud providers and consumers to minimize the cost of accessing the cloud applications. It will reduce the waiting time of the consumer for accessing the resource in cloud at traffic less environment with efficient cost.*

KEYWORDS
Middleware platform, Heuristic solution, Resource allocation, gossip protocol.

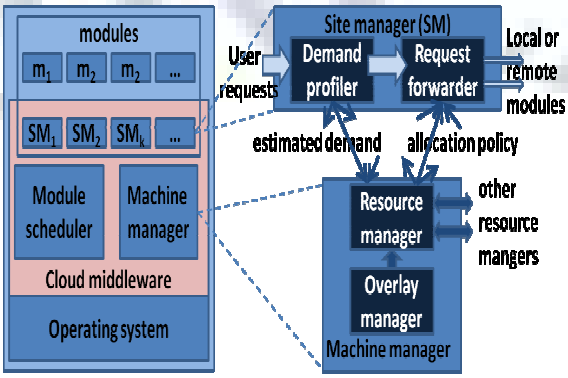
I.INTRODUCTION

Cloud computing is a popular trend in current computing which attempts to provide cheap and easy access to make the computational resources. Compared to previous paradigms, cloud computing focuses on treating computational resources as measurable and billable utilities. From the clients point of view, cloud computing provides an abstraction of the underlying hardware architecture. This abstraction saves them the costs of design, setup and maintenance of a data center to host their Application Environments (AE). This economy of scale provides benefits to both parties, but leaves the providers in a position where they must have an efficient and cost effective data center. This approach centers around a decentralized design whereby the components of the middleware layer run on every processing node of the cloud environment. To achieve scalability, it is envisioned that all key tasks of the middleware layer, including estimating global states, placing site modules and computing policies for request forwarding are based on distributed algorithms.

The core contribution is a gossip protocol P*, which executes in a middleware platform and meets the design goals outlined above. The protocol has two innovative characteristics. First, while gossip protocols for load balancing in distributed systems have been studied before, has no results are available for cases that consider memory constraints and the cost of reconfiguration, which makes the resource allocation problem hard to solve (memory constraints alone make it NP-hard). An optimal solution is provided for a simplified version of the resource allocation problem and an efficient heuristic for the hard problems. Second, the protocol proposed is continuously executes, while its input and consequently its output dynamically changes. Most gossip protocols that have been proposed to date are used in a different way. They assume static input and produce a single output value. The benefit of a single, continuous execution vs. a sequence of executions with restarts is that in which global synchronization can be avoided and that the system can continuously adapt to changes in local input. On the other hand, its drawback is that the behavior of a protocol with dynamic input is more difficult to analyze. Also, the cost of the system to react to a high rate of change in local output can potentially be higher than implementing a set of changes after each synchronized run. Based on the work thus far, it is believed that, for a gossip protocol running in large-scale dynamic environments, the advantages of continuous execution with dynamic input outweigh its potential drawbacks.

II. SYSTEM MODEL

FIG. 1: SYSTEM DESIGN



A cloud environment spans several datacenters interconnected by an internet. Each of these datacenters contains a large number of machines that are connected by a high-speed network. Users access sites hosted by the cloud environment through the public Internet. A site is typically accessed through a URL that is translated to a network address through a global directory service, such as DNS. A request to a site is routed through the Internet to a machine inside a

datacenter that either processes the request or forwards it. In this paper, we restrict ourselves to a cloud that spans a datacenter containing a single cluster of machines and leave for further work the extension of our contribution to an environment including multiple datacenters.

Each site manager handles user requests to a particular site. It has two important components: a demand profiler and request forwarder. The demand profiler estimates the resource demand of each module of the site based on the request statistics, QoS targets, etc This estimate is forwarded to all machine managers that run instances of modules belonging to this site. Similarly, the request forwarder sends user requests for processing to instances of modules belonging to this site. Request forwarding decisions take into account the resource allocation policy and constraints such as session affinity. Figure shows the components of a site manager and how they relate to machine managers. The remainder of this paper focuses on the functionality of the resource manager component.

III. FORMALIZING THE PROBLEM OF RESOURCE ALLOCATION BY THE CLOUD MIDDLEWARE

The specific problem addressed is that of placing modules (more precisely: identical instances of modules) on machines and allocating cloud resources to these modules, such that a cloud utility is maximized under constraints. As cloud utility we choose the minimum utility generated by any site, which we define as the minimum utility of its module instances. We formulate the resource allocation problem as that of maximizing the cloud utility under CPU and memory constraints. The solution to this problem is a configuration matrix that controls the module scheduler and request forwarder components. At discrete points in time, events occur, such as load changes, addition and removal of site or machines, etc. In response to such an event, the opti- mization problem is solved again, in order to keep the cloud utility maximized. We add a secondary objective to the optimization problem, which states that the cost of change from the current configuration to the new configuration must be minimized.

A.THE MODEL

We model the cloud as a system with a set of sitesS and a set of machines N that run the sites. Each site s ∈ S is composed of a set of modules denoted by Ms.We consider a system that may run more than one instance of a module m, each on a different machine, in which case its CPU demand is divided among its instances. The demand ωn,m (t) of an instance of m run- ning on machine n is given by ωn,m (t) = αn,m (t)ωm (t), where

n∈N αn,m (t) = 1 and αn,m (t) ≥ 0.

It is called that the matrix A with elements αn,m (t) the configuration(matrix) of the system. A is a non-negative matrix with 1T A = 1T

A machine n ∈ N in the cloud has aCPU capacity Ωn and a memory capacity Γn . We use Ω and Γ to denote the vectors of CPU and memory capacities of allthe machines in the system. An instance of module m running on machine n demands ωn,m (t) CPU resource and γm memory resource from n. Machine n allocates to module m the CPU capacity ωˆn,m (t) (which may be different from ωn,m (t)) and the memory capacity γm . We define the utility un,m (t) generated by an instance of module m on machine n as the ratio of the allocated CPU capacity to the demand of the instance on that particular machine. Wefurther define the utility of a module m as

um (t) =minn∈N {un,m (t)}

and that of a site as the minimum of utility of its modules. Finally, the utility of the cloud U c is the minimum of the utilities of the sites it hosts. As a consequence, the utility of the cloud becomes the minimum utility of any module instance in the system.

B.THE OPTIMIZATION PROBLEM

For the above model, we consider a cloud with CPU capacity Ω, memory capacity Γ, and demand vectors ω, γ . We first discuss a simplified version of the problem. It consists of finding a configuration A that maximizes the cloud utility U c.

Maximize U c(A,ώ) OP(1)

subject to A ≥ 0, 1T A = 1T (a)

Ωˆ (A, ω)1 Ω (b)

Our concept of utility is max-min fairness and our goal is to achieve fairness among sites. This means that we want to maximize the minimum utility of all sites, which we achieve by maximizing the minimum utility of all module instances.

Constraint (a) of OP(1) relates to dividing into shares the CPU demand of each module into the demand of its instances. The constraint expresses that all shares are non-negative and add up to 1 for each module.

maximize U c (A(t + 1), ω(t + 1))

minimize c∗(A(t), A(t + 1))

subject to

A(t + 1) ≥ 0, 1T A(t + 1) = 1T

Ωˆ (A(t + 1), ω(t + 1))1 Ω

sign(A(t + 1))γ Γ.

(OP(2))

This optimization problem has prioritized objectives in the sense that, among all configurations A that maximize the cloud utility, we select one that minimizes the cost function c. While this paper considers only events in form of changes in demand, OP(2) allows us to express (and solve) the problem of finding a new allocation after other events, including adding or removing sites or machines.

IV. THE PROTOCOL FOR DISTRIBUTIVE RESOURCE ALLOCATION

In this section, we present a protocol P, which is a heuristic algorithm for solving OP(2) and which represents our proposed protocol for resource allocation in a cloud environment. P is a gossip protocol and has the structure of a round- based distributed algorithm (whereby round-based does not imply that the protocol is synchronous). When exe- cutting a round-based gossip protocol, each node selects a subset of other nodes to interact with, whereby the selection function is often probabilistic. Nodes interact via 'small' messages, which are processed and trigger local state changes. In this work, node interaction follows the so-called push-pull paradigm, whereby two nodes exchange state information, process this information and update their local states during a round.

P runs on all machines of the cloud. It is invoked at discrete points in time, in response to a load change. The output of the protocol, the configuration matrix A, is distributed across the machines of the system. A controls the start and stop of module instances and determines the control policies for module schedulers and request forwarders. The protocol executes in the resource manager components of the middleware architecture. A set of candidate machines to interact with is maintained by the overlay manager component of the machine manager. We assume that the time it takes for P to compute a new configuration A is small compared to the time between events that trigger consecutive runs of the protocols. At the time of initialization, P reads as input a feasible configuration of the system (see below). At later invocations, the protocol reads as input the configuration matrix produced during the previous run.

A. Functionalities the protocol P Uses

a) Random selection of machines: P relies on the ability of a machine to select another machine of the cloud uniformly at random. In this work, we approximate this ability by using CYCLON, an overlay protocol that produces a time-varying network graph with properties of a random network [3].

b) Resource allocation and module scheduling policy.

c) Computing a feasible configuration: P requires a feasible configuration as input during its initialization phase. A simple greedy algorithm can be used for this purpose, which we present in [4] due to space limitation.

B. Protocol P': An Optimal Solution to OP(1)

We developed the protocol P', which is a distributed solution to OP(1). P' is a gossip protocol that produces a sequence of configuration matrices

A(r), r = 1, 2, . . .,

such that the series of cloud utilities U c (A(r), ω) con- verges exponentially fast to the optimal utility. Due to space limitation, P' is described and its properties proved in [4]. We would encourage the reader to look up this protocol, as it is quite simple and enables a better understanding of P, which can be

seen as an extension of P'. During each round of P', two machines perform an equalization step whereby CPU demand is moved from one machine to another machine in such a way that their relative demands are equalized.

C. PROTOCOL P: A HEURISTIC SOLUTION TO OP (2)

OP(2) differs from OP(1) in that memory constraints of individual machines are considered and a secondary objective is added for the purpose of minimizing the cost of adapting the system from the current to a new configuration that maximizes the utility for the new demand. Introducing local memory constraints to the optimization problem turns OP(1), which we showed can be efficiently solved for many practical cases [4], into an NP-hard problem [2].

P employs the same basic mechanism as P' as it attempts to equalize the relative demands of pairs of machines during a protocol round. Due to the local memory constraints, such a step does not always succeed.

P uses the following approach to achieve its objectives. First, pairs of machines that execute an equalization step are often chosen in such a way that they run instances of common modules. To support this concept, we maintain on each machine n the set Nn of machines in the cloud that run module instances common with n. To avoid the possibility of the cloud being partitioned into disjoint sets of interacting machines, n is occasionally paired with a machine outside of the set Nn to execute an equalization step. This dual approach keeps low the need for starting new module instances and thus keeps the cost low. Second, during an equalization step, P attempts to reduce the difference in relative demand between two machines, in case it cannot equalize the demand. Further, P attempts to execute an equalization step in such a way that the demand for a specific module is shifted to one machine only. This concept aims at increasing the probability that an equalization step succeeds in equalizing the relative demands, thus increasing the cloud utility.

The pseudo code of P is given in Algorithm 1. To keep the presentation simple, we omit thread synchronization primitives which prevent concurrent machine to machine interactions. Note that setting αn,m = 0 implies stopping module m on machine n.

During the initialization of machine n, the algorithm reads the CPU demand vector, the CPU and memory capacity vectors, and the row of the configuration matrix for nThen, it starts two threads: an active thread, in which the machine periodically executes a round, and a passive thread that waits for another machine to start an interaction.

Algorithm 1 Protocol P computes a heuristic solution for OP(2) and returns a configuration matrix A.

The active thread executes rmax rounds. In each round, n chooses a machine n0 uniformly at random from the set Nn with probability p and from the set N − Nn with probability 1 − p. Then n sends its state (i.e., rown (A)) to n0 , receives n0's state as a response, and calls the procedure equalizeWith(), which performs the equalization step.

```
Initialization
1: read ω, Ω, Γ, rown (A), Nn ;
2: start the passive and active threads
active thread
3: for r = 1 to rmax do
4:         if rand(0..1) < p then
5:         choose n0 at random from Nn ;
6:         else
7:         choose n0 at random from N − Nn ;
8:         send(n0 , rown (A));rown0 (A) =receive(n0 );
9:         equalizeWith(n0 , rown0 (A));
10:        sleep(roundDuration);
11: write rown (A);
passive thread
12: while true do
13:        rown0 (A)=receive(n0 );send(n0 , rown (A));
14:        equalizeWith(n0 , rown0 (A));
```

```
proc equalizeWith(j, rowj (A))
1: l = arg max{vn , vj }; l0 = arg min{vn , vj };
2: if j ∈ Nn then
3:         moveDemand1(l, rowl (A), l0 , rowl0 (A));
4: else
5:         moveDemand2(l, rowl (A), l0 , rowl0 (A));
```

```
proc moveDemand1(l, rowl (A), l0 , rowl0 (A))
1: compute Δω such that
1/Ωl ( P m ωl,m − Δω) = 1/Ωl0 ( P m ωl0 ,m + Δω)
2: let mod be an array of all modulesthat run on both l and l0 , sorted by increasing ωl,m
3: for i = 1 to |mod| do
4:         m = mod[i]; δω = min(Δω, ωl,m );
5:         Δω − = δω; δα = αl,m δω /ωl,mαl0 ,m + = δα; αl,m − = δα;
```

```
proc moveDemand2(l,rowl(A),l0,rowl0(A))

1: computeΔωsuchthat
   1/Ωl ( P m ωl,m−Δω)= 1/Ωl0 ( P m ωl0,m +Δω)
2:letmod be an arrayofall modules
   that runon l,sortedby decreasing
   ωl,m
   γm ;

3: forl=1 to|mod|do

4:    m=mod[i];δω=min(Δω,ωl,m);
```

Then, from the set of modules that run on both machines, taking an instance with the smallest demand on l, it proceeds to shift the demand from l to l0, until a total of Δω demand is shifted, or it has exhausted the set of modules. moveDemand2() equalizes (or reduces the difference) of the relative demands of the two machines, by moving demand from the machine with larger relative demand.

V. PRICE AND TIME SLOT NEGOTIATION

The *PTN* mechanism consists of the following: 1) an aggregated utility function; 2) negotiation strategies; and 3) a negotiation protocol.

A. UTILITY FUNCTIONS

A utility function $U(x)$ represents an agent's level of satisfaction for a negotiation outcome x. Since each Cloud participant has different preferences for different prices and time slots, a price utility function, a time-slot utility function, and an aggregated utility function are used to model the preference ordering of each proposal and each negotiation outcome.

B. .NEGOTIATION STRATEGY

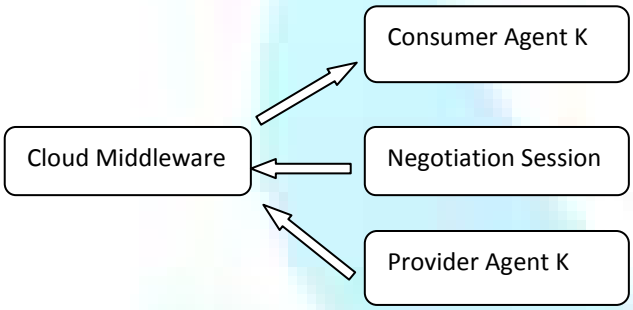
This work considers bilateral negotiations between a consumer and a provider, where both agents are sensitive to time and adopt a time-dependent concession-making strategy for *PTNs*. Since both agents negotiate on both price and time slot, generating a counterproposal can be making either a concession or a tradeoff between price and time slot. Hence, an agent's strategy for multi-issue negotiation is implemented using both the following: 1) a tradeoff algorithm and 2) a concession making algorithm.

C. NEGOTIATION PROTOCOL

The negotiation protocol of the PTN mechanism follows:

Rubinstein's alternating offers protocol in which agents make counteroffers to their opponents in alternate rounds. Both agents generate counteroffers and evaluate their opponent's offers until either an agreement is made or one of the agents' deadline is reached. If a counterproposal is accepted, both agents found a mutually acceptable price and time slot. If one of the agents' deadline expires before the y reach an agreement, the negotiation fails.

FIG. 2: AGENT BASED CLOUD TESTBED



VI. RELATED WORK

The problem of application placement in the context of resource management for datacenters has been studied before (e.g., [2], [7]), and solutions are already available in middleware products [8]. While these product solutions allow for a fair resource allocation in a similar way as our scheme does, they rely on centralized architectures, which do not at all scale to system sizes we consider in this paper.

Distributed load balancing algorithm have been extensively studied for homogeneous as well as hetero- geneous systems, for both divisible and indivisible demands. These algorithms typically fall into two classes: diffusion algorithms (e.g., [11]) and dimension exchange algorithms (e.g., [12]). Convergence results for different network topologies and different norms (that measure the distance between the system state and the optimal state) have been reported, and it seems to us that the problem is well understood today. The key difference to the problem addressed in this paper is that these algorithms do not take into account memory constraints. Considering memory constraints makes the problem NP- hard and does require a new approach.

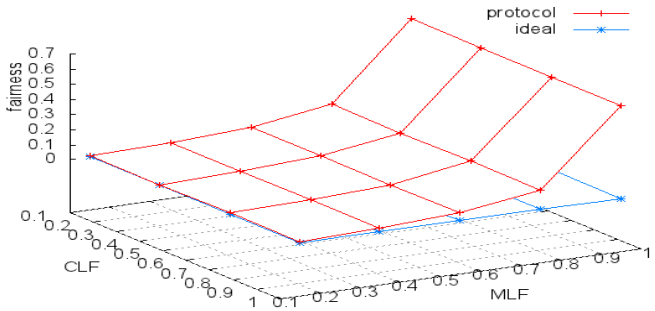
VII. CONCLUSION

With this paper, we make a significant contribution towards engineering a resource management middleware for a site hosting cloud environment. We identify a key component of such a middleware and present a protocol that can be used to meet our design goals for resource management: fairness of resource allocation with respect to sites, efficient adaptation to load changes and scalability of the middleware layer in terms of both the number of machines in the cloud as well as the number of hosted sites.

We presented a gossip protocol, that computes the heuristic solution to the resource allocation problem and evaluated its performance. In all the scenarios we investigated, we observe that the protocol qualitatively behaves as expected based on its design. Regarding fairness, the gossip protocol performs close to an ideal system for scenarios where the ratio of the total memory capacity to the total memory demand is large. The simulations suggest that the protocol is scalable in the sense that all inves- tigated metrics do not change when the system size (i.e., the number of machines) increases proportional to the external load (i.e., the number of sites). Note that if we would solve the resource allocation problem expressed in OP(2) by P in a centralized system, then the CPU and memory demand for that resource allocation system would increase linearly with the system size. Another novelty of this work is formulating a novel time-slot utility function that characterizes preferences for different time slots. These ideas are implemented in an agent based Cloud testbed. This strongly suggests to us that a centralized solution for the problem we address in this paper will not be feasible.

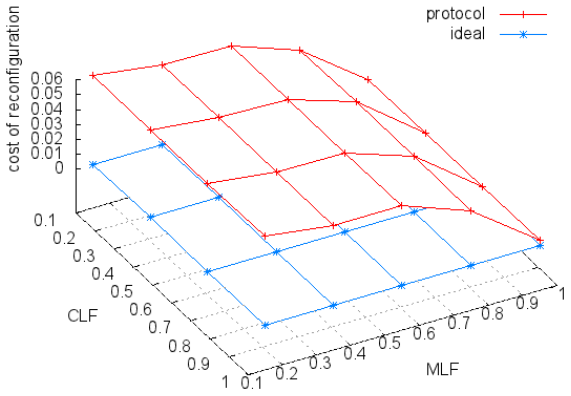
(a) Fairness among sites (0 means optimal fairness).

FIG. 3



(b) Cost of change in configuration over all machines.

FIG. 4



The results reported in this paper are building blocks towards engineering a resource management solution for large-scale clouds. Pursuing this goal, we plan to address the following issues in future work: (1) Develop a distributed mechanism that efficiently places new sites. (2) Extend the middleware design to become robust to machine failures. (3) Extend the middleware design to span several clusters and several datacenters, while keeping module instances of the same site "close to each other", in order to minimize response times and communication overhead.

REFRENCES

1. Adam and R. Stadler, *'Service middleware for self-managing large scale systems,'* IEEE Trans. Network and Service Management, vol. 4, no. 3, pp. 50–64, Apr. 2008.
2. Carrera, M. Steinder, I. Whalley, J. Torres, and E. Ayguade, *'Utility based placement of dynamic web applications with fairness goals,"* in 2008 IEEE Network Operations and Management Symposium.
3. F. Wuhib, M. Dam, and R. Stadler, *'A gossiping protocol for detecting global threshold crossings,'* IEEE Trans. Network and Service Management, vol. 7, no. 1, pp. 42–57, Mar. 2010.
4. Fetahi Wuhib, Rolf Stadler, and Mike Spreitzer, *'A Gossip Protocol for Dynamic Resource Management in Large Cloud Environments,'* IEEE transactions on network and service management, vol. 9, no. 2, June 2012
5. Loureiro, P. Nixon, and S. Dobson, *'Decentralized utility maximization for adaptive management of shared resource pools,'* in 2009 International Conference on Intelligent Networking and Collaborative Systems.
6. M. Jelasity, A. Montresor, and O. Babaoglu, *'Gossip-based aggregation in large dynamic networks,'* ACM Trans. Computer Syst., vol. 23, no. 3, pp. 219–252, 2005.
7. Mark Jelasity, Ozalp Babaoglu, *'T-Man: Fast Gossip-based Construction of Large Scale Overlay Topologies,'* Technical Report UBLCS-2004-7
8. R. L. Graham, *'Bounds on multiprocessing timing anomalies,'* SIAM J. Applied Mathematics, vol. 17, no. 2, pp. pp. 416–429, 1969.
9. S. Voulgaris, D. Gavidia, and M. van Steen, *'CYCLON: inexpensivemembership management for unstructured p2p overlays,'* J. Network and Systems Management, vol. 13, no. 2, pp. 197–217, 2005.
10. Tang, M. Steinder, M. Spreitzer, and G. Pacifici, *'A scalable application placement controller for enterprise data centers,'* in 2007. International Conference on World Wide Web.
11. Wuhib, M. Dam, R. Stadler, and A. Clem, *'Robust monitoring of network-wide aggregates through gossiping,'* IEEE Trans. Network and Service Management, vol. 6, no. 2, pp. 95–109, June 2009.

REQUEST FOR FEEDBACK

Dear Readers

At the very outset, International Journal of Research in Computer Application and Management (IJRCM) acknowledges & appreciates your efforts in showing interest in our present issue under your kind perusal.

I would like to request you tosupply your critical comments and suggestions about the material published in this issue as well as on the journal as a whole, on our E-mail**infoijrcm@gmail.com** for further improvements in the interest of research.

If youhave any queries please feel free to contact us on our E-mail **infoijrcm@gmail.com**.

I am sure that your feedback and deliberations would make future issues better – a result of our joint effort.

Looking forward an appropriate consideration.

With sincere regards

Thanking you profoundly

Academically yours

Sd/-

Co-ordinator

ABOUT THE JOURNAL

In this age of Commerce, Economics, Computer, I.T. & Management and cut throat competition, a group of intellectuals felt the need to have some platform, where young and budding managers and academicians could express their views and discuss the problems among their peers. This journal was conceived with this noble intention in view. This journal has been introduced to give an opportunity for expressing refined and innovative ideas in this field. It is our humble endeavour to provide a springboard to the upcoming specialists and give a chance to know about the latest in the sphere of research and knowledge. We have taken a small step and we hope that with the active co-operation of like-minded scholars, we shall be able to serve the society with our humble efforts.

Our Other Journals