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.

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

Registered & Listed at: Index Copernicus Publishers Panel, Poland & number of libraries all around the world.

Circulated all over the world & Google has verified that scholars of more than 1667 Cities in 145 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

www.ijrcm.org.in

CONTENTS

CALL FOR MANUSCRIPTS

We invite unpublished novel, original, empirical and high quality research work pertaining to recent developments & practices in the area of Computer, Business, Finance, Marketing,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, 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 mention the 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

AN ANALYSIS OF ANEKA (CLOUD COMPUTING TOOL)

AANHA GOYAL
M. TECH. STUDENT
COMPUTER SCIENCE
JAYOTI VIDHYAPEETH WOMENS UNIVERSITY
JAIPUR

ANSHIKA BANSAL
B. TECH. STUDENT
JAYOTI VIDHYAPEETH WOMENS UNIVERSITY
JAIPUR

ABSTRACT

Aneka is a platform for deploying Clouds developing applications on top of it. It provides a runtime environment and a set of APIs that allow developers to build .NET applications that leverage their computation on either public or private clouds. Aneka is the ability of supporting multiple programming models that are ways of expressing the execution logic of applications by using specific abstractions. Aneka is an Application Platform-as-a-Service (PaaS) for Cloud Computing. which offers a runtime environment and a set of APIs that enable developers to build customized applications by using multiple programming models such as Task Programming, Thread Programming and Map Reduce Programming, Aneka provides a number of services that allow users to control, auto-scale, reserve, monitor and bill users for the resources used by their applications. Aneka is its support for provisioning resources on different public Cloud providers such as Amazon EC2, Windows Azure and Go Grid.

KEYWORDS

Aneka, Paas (Platform-as-a-service);.Net Frame work; API(Application program interface); Cloud, Anatomy.

I. INTRODUCTION

Aneka is a platform for deploying Clouds developing applications on top of it. It provides a runtime environment and a set of APIs(Application Program Interface) that allow developers to build .NET applications that leverage their computation on either public or private clouds. ***Aneka*** allows servers and desktop PCs to be linked together to form a very powerful computing infrastructure. Aneka is a workload distribution and management platform that accelerates applications in Microsoft .NET framework environments. It simplifies the development of distributed applications by providing: a collection of different ways for expressing the logic of distributed applications, a solid infrastructure that takes care of the distributed execution of applications, and a set of advanced features such as the ability to reserve and price computation nodes and to integrate with existing cloud infrastructures such as Amazon EC2. Some of the key advantages of Aneka over other GRID or Cluster based workload distribution solutions include:

- rapid deployment tools and framework,
- ability to harness multiple virtual and/or physical machines for accelerating application
- result provisioning based on QoS/SLA
- support of multiple programming and application environments
- simultaneous support of multiple run-time environments
- built on-top of Microsoft .NET framework

A. PUBLIC CLOUDS

A public cloud encompasses the traditional concept of cloud computing, having the opportunity to use computing resources from anywhere in the world. The clouds can be used in a so-called pay-per-use manner.

B. PRIVATE CLOUDS

Private clouds are normally data centers that are used in a private network and can therefore restrict the unwanted public to access the data that is used by the company.

II. ARCHITECTURE

Aneka is a platform and a framework for developing distributed applications on the Cloud. It harnesses the spare CPU cycles of a heterogeneous network of desktop PCs and servers or data centers on demand. Aneka provides developers with a rich set of APIs for transparently exploiting such resources and expressing the business logic of applications by using the preferred programming abstractions. System administrators can leverage on a collection of tools to monitor and control the deployed infrastructure. This can be a public cloud available to anyone through the Internet, or a private cloud constituted by a set of nodes with restricted access. Aneka is a .NET-based application development Platform-as– a-Service (PaaS), which offers a runtime environment and a set of APIs that enable developers to build customized applications by using multiple programming models

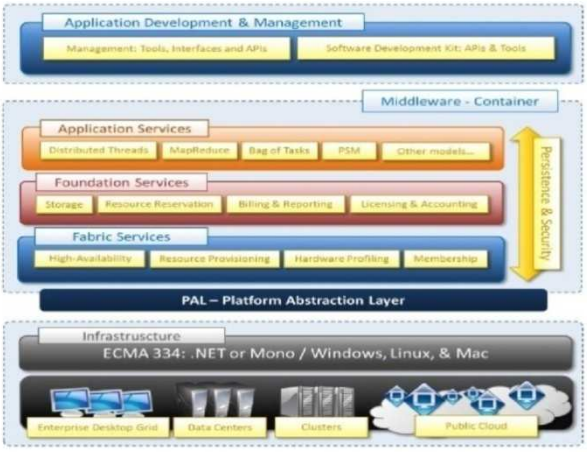
A. OVERVIEW

The Aneka based computing cloud is a collection of physical and virtualized resources connected through a network, which could be the Internet or a private intranet. Each of these resources hosts an instance of the Aneka Container representing the runtime environment in which the distributed applications are executed. The container provides the basic management features of the single node and leverages all the other operations on the services that it is hosting. In particular we can identify fabric, foundation, and execution services.

Additional services such as persistence and security are transversal to the entire stack of services that are hosted by the Container. At the application level, a set of different components and tools are provided to:

- Simplify the development of applications (SDK)
- Porting existing applications to the Cloud;
- and Monitoring and managing the Aneka Cloud.

FIG.1: OVERVIEW OF ANEKA

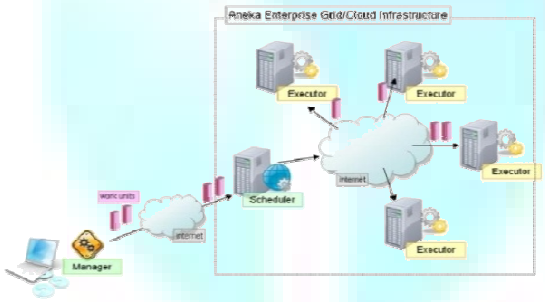


An Aneka based Cloud is constituted by a set of interconnected resources that are dynamically modified according to the user needs by using resource virtualization or by harnessing the spare CPU cycles of desktop machines. If the deployment identifies a private Cloud all the resources are in house, for example within the enterprise. This deployment is extended by adding publicly available resources on demand or by interacting with other Aneka public clouds providing computing resources connected over the Internet. The heart of this infrastructure is the Aneka Container which represents the basic deployment unit of Aneka based clouds. Some of the most characteristic features of the Cloud Computing model are:

- flexibility,
- elasticity (scaling up or down on demand)
- Pay as per usage.

The architecture and the implementation of the Container play a key role in supporting these three features: the Aneka cloud is flexible because the collection of services available on the container can be customized and deployed according to the specific needs of the application. It is also elastic because it is possible to increase on demand the number of nodes that are part of the Aneka Cloud according to the user needs. The integration of virtual resources into the Aneka Cloud does not introduce specific challenges: once the virtual resource is acquired by Aneka it is only necessary to have an administrative account and a network access to it and deploy the Container on it as it happens for any other physical node. Moreover, because of the Container being the interface to hosting node it is easy to monitor, meter, and charge any distributed application that runs on the Aneka Cloud.

FIG.2: ARCHITECTURE OF ANEKA



B. ANATOMY OF THE ANEKA CONTAINER

The Container represents the basic deployment unit of Aneka based Clouds. The network of containers defining the middleware of Aneka constitutes the runtime environment hosting the execution of distributed applications. Aneka strongly relies on a Service Oriented Architecture and the Container is a lightweight component providing basic node management features. All the other operations that are required by the middleware are implemented as services. Such as

- Fabric Services
- Foundation Services
- Execution Services
- Transversal Services

C. FABRIC SERVICES

It defines the lowest level of the software stack representing the Aneka Container. They provide access to the resource provisioning subsystem and to the hardware of the hosting machine. Resource provisioning services are in charge of dynamically providing new nodes on demand by relying on virtualization technologies, while hardware profile services provide a platform independent interface for collecting performance information and querying the properties of the host operating system and hardware. These services rely on the Platform Abstraction Layer (PAL) that allows the Container to be completely independent from the hosting machine and the operating system and the whole framework to be portable over different platforms

D. FOUNDATION SERVICES

It represents the core of the Aneka middleware on top of which Container customization takes place. Foundation services constitute the pillars of the Aneka middleware and are mostly concerned with providing runtime support for execution services and applications. The core of Aneka addresses different issues:

- Directory and Membership;
- Resource reservation;
- Storage management;
- Licensing, accounting, and pricing;

D.1. Directory and Membership

It is responsible for setting up and maintaining the information about the nodes and the services constituting the Aneka Cloud. These services include Membership Catalogue, Heartbeat Service, and Discovery Service.

D.2. Resource Reservation

Resource reservation identifies the ability of reserving a set of nodes and using them for executing a specific application. The ability of reserving compute resources two different components has been implemented: Reservation Service and Allocation Manager.

D.3. Storage management

The availability of disk space, or more generally storage, is a fundamental feature for any distributed system implementation. Applications normally require files to perform their tasks, whether they are data files, configuration files, or simply executable files.

D.4. Licensing, Accounting, and Pricing

The Licensing Service provides the very basic resource controlling feature that protects the system from misuse. It restricts the number of resources that can be used for a certain deployment. The Accounting and Pricing Services, available in the next release of Aneka, are more directly related with billing the user for using the Cloud.

2.5. Execution Services

Execution services identify the set of services that are directly involved in the execution of distributed applications in the Aneka Cloud.

III. APPLICATION DEVELOPMENT

Aneka is a platform for developing applications that leverage Clouds for their execution. It then provides a runtime infrastructure for creating public and private Clouds and a set of abstractions and APIs through which developers can design and implement their applications. More specifically Aneka provides developers with a set of APIs for representing the Cloud application and controlling their execution, and a set of Programming Models that are used to define the logic of the distributed application itself. These components are part of the Aneka Software Development Kit.

A. THE ANEKA SDK

The Aneka Software Development Kit contains the base class libraries that allow developers to program applications for Aneka Clouds. the SDK contains a collection of class libraries constituting the Aneka Application Model, and their specific implementations for the supported programming models. The Aneka Application Model defines the properties and the requirements for distributed applications that are hosted in Aneka Clouds. Differently from other middleware implementations Aneka does not support single task execution, but any unit of user code is executed within the context of a distributed application. An application in Aneka is constituted by a collection of execution units whose nature depends on the specific programming model used. An application is the unit of deployment in Aneka and configuration and security operates at application level. Execution units constitute the logic of the applications. The way in which units are scheduled and executed is specific to the programming model they belong to. By using this generic model, the framework provides a set of services that work across all programming model supported: storage, persistence, file management, monitoring, accounting, and security.

B. PROGRAMMING MODELS

A programming model represents a way for expressing a distributed application within Aneka. It defines the abstractions used by the user to model their application and the execution logic of these applications as a whole in the Aneka Cloud. Every application that is executed in the Aneka Cloud is expressed in terms of a specific programming model. The current release of Aneka includes three different programming models ready to use for developing applications. These are: Task Programming Model, Thread Programming Model, and Map Reduce Programming Model.

B.1. TASK PROGRAMMING MODEL

The Task Programming Model provides developers with the ability of expressing bag of tasks applications. By using the Task Model the user can create a distributed application and submit a collection of tasks to Aneka. The submission can be either static or dynamic. The scheduling and execution services will manage the execution of these tasks according to the available resources in the Aneka network. Developers can use predefined tasks that cover the basic functionalities available from the OS shell or define new tasks by programming their logic. With tasks being independent from each other, this programming model does not enforce any execution order or sequencing but these operations have to be completely managed by the developer on the client application if needed. The task programming model is the most straightforward programming model available with Aneka and can be used as a base on top of which other models can be implemented. For example the parameter sweeping APIs used by the Design Explorer rely on the Task Model APIs to create and submit the tasks that are generated for each of the combinations of parameters that need to be explored. More complex models such as workflows can take advantage of this simple and thin implementation for distributing the execution of tasks.

B.2. THREAD PROGRAMMING MODEL

The Thread Programming Model allows quickly porting multi-threaded applications into a distributed environment. Developers familiar with threading API exposed by the .NET framework or Java can easily take advantage of the set of compute resources available with Aneka in order to improve the performance of their applications. The Thread Model provides as fundamental component for building distributed applications the concept of distributed thread. A distributed thread exposes the same APIs of a thread in the .NET framework but is executed remotely. Developers familiar with the multi-threaded applications can create, start, join, and stop threads in the same way in which these operations are performed on local threads. Aneka will take care of distributing and coordinating the execution of these threads.

B.3. MAP REDUCE PROGRAMMING MODEL

The Map Reduce Programming Model is an implementation of Map Reduce as proposed by Google, for .NET and integrated with Aneka. Map Reduce is originated by two functions from the functional language: map and reduce. The map function processes a key/value pair to generate a set of intermediate key/value pairs, and the reduce function merges all intermediate values associated with the same intermediate key. This model is particular useful for data intensive applications. The Map Reduce Programming Model provides a set of client APIs that allow developers to specify their map and reduce functions, to locate the input data, and whether to collect the results if required. Map Reduce is good example for the flexibility of the architecture of Aneka in supporting different programming abstractions supported by the infrastructure.

IV. CLOUD MAINTENANCE AND MONITORING

Aneka provides a platform on top of which it is possible to develop applications for the Cloud. The Software Development Kit addresses all the needs from a development point of view but it is just a part of the feature set required by a Cloud Computing platform. Essential in this case is the support for monitoring, managing, maintaining, and setting up computing clouds. These operations are exposed by the management API and the Platform Abstraction Layer on top of which all the management tools and interfaces have been designed. Of a particular interest are the Management Studio and the web management interfaces. The Management Studio is an important tool for system administrators. It is a comprehensive environment that allows them to manage every aspect of Aneka Clouds from an easy to use graphical user interface. Since Clouds are constituted of hundreds and even thousands of machines both physical and virtual, it is not possible to reach and setup each single machine by hand.Having a tool that allows remote and global management is then a basic requirement. Briefly, the set of operations that can be performed through the Management Studio are the: -Quick setup of computing clouds;-Remote installation and configuration of nodes;-Remote control of containers; -System load monitoring and tuning. Besides the remote control features, which dramatically simplify the management of the Cloud, it is important to notice the support for viewing the aggregate dynamic statistics of Aneka Clouds. This helps administrators to tune the overall performance of the Cloud. It is also possible to probe each single node and collect the single performance statistics: the CPU and memory load information is collected from each container and by inspecting the container configuration it is possible to identify bottlenecks in the Cloud. As the entire framework, the Management Studio has been designed to be extensible: it is possible to add new features and new services by implementing management plug ins that are loaded into the environment and get access to the Cloud.

V. CONCLUSIONS AND FUTURE DIRECTIONS

Aneka is an implementation of the Platform as a Service approach, which focuses on providing a set of APIs that can be used to design and implement applications for the Cloud. The framework is based on an extensible and service oriented architecture that simplifies the deployment of clouds and their maintenance and provides a customizable environment that supports different design patterns for distributed applications. Aneka Container which is the minimum unit of deployment for Aneka Clouds and also the runtime environment for distributed applications. The container hosts a collection of services that perform all the operations required to create an execution environment for applications. A programming model is a specific way of expressing the execution logic of distributed applications. It provides some familiar abstractions that developers can use to define the execution flow of applications and its component. The development team is now working on providing full support for the elastic scaling of Aneka Clouds by relying on virtualized resources

VI. REFERENCES

1. C. Jin, R. Buyya, MapReduce programming model for .NET-based distributed computing, Proc. 15th European Conference on Parallel Processing (Euro-Par 2009), (2009)
2. J.S. Miller, S. Ragsdale, The Common Language Infrastructure Annotated Standard, Addison Wesley, 2004.
3. L. Vaquero, L. Rodero-Marino, J. Caceres, M. Lindner, A break in the clouds: towards a cloud definition, SIGCOMM Computer Communication Review, 39 (2009) 137–150.
4. R. Buyya, S. Venugopal, The Grid bus Toolkit for service oriented grid and Utility Computing: An overview and status report, Proc. of the First IEEE International Workshop on Grid Economics and Business Models (GECON 2004), (2004), 19–36.

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 to supply 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 you have 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