AENSI Journals

Australian Journal of Basic and Applied Sciences

ISSN:1991-8178

Journal home page: www.ajbasweb.com

Strategies for Solving the NP-Hard Workflow Scheduling Problems in Cloud Computing Environments

[1]S.Balamurugan, [2]Dr.P.Visalakshi, [3]V.M.Prabhakaran, [4]S.Charanyaa, [5]S.Sankaranarayanan

[1]*Anna University, IT Department, Kalaignar Karunanidhi Institute of Technology, Coimbatore ,Tamilnadu, India*
[2]*Anna University, ECE Department, PSG College of Technology, Coimbatore, Tamilnadu, India.*
[3] *Anna University, CSE Department, Kalaignar Karunanidhi Institute of Technology, Coimbatore ,Tamilnadu, India*
[4]*Anna University, IT Department, SNS College of Technology, Coimbatore, Tamilnadu, India.*
[5]*Dell International Services, SVCITSR Associate II, Noida, India.*

ARTICLE INFO

Article history:
Received 8 August 2014
Received in revised form
12 September 2014
Accepted 25 September 2014
Available online 20 November 2014

Keywords:
Cloud Computing, Task Scheduling, Grid Computing, Workflow Management Systems, Aneka, CloudSim

ABSTRACT

Introduction: Cluster and cloud computing technologies have shown a steep rise during the recent few years. Cluster comprises of logically connected independent systems working in cohesion with each others. Hence cluster is termed to be a variant of parallel or distributed processing system. The most important problem area, that need to be focused in is devising algorithm to split the given application into tasks, and allocating the tasks to different processors, coordinating the communication between the tasks and so on. **Objective:** The objective of this paper is to provide a detailed investigation on solving np-hard problems like scheduling and task allocation. The paper also surveys various Workflow Management Systems available as exiting projects. **Results:** To simulate the user own programs the cloud must have minimum configurations of a basic system with a CloudSim package and the java toolkit 1.6 or any other enhanced version. Creation of scalable cloud is illustrated with a simulation **Conclusion:**The paper concludes providing a literary review on migrating to clouds to solve the NP-hard workflow scheduling problem using Aneka and CloudSim.

To Cite This Article: S.Balamurugan, Dr.P.Visalakshi, V.M.Prabhakaran, S.Charanyaa, S.Sankaranarayanan., Strategies for Solving the NP-Hard Workflow Scheduling Problems in Cloud Computing Environments. ***Aust. J. Basic & Appl. Sci.,*** *8(16): 345-355, 2014*

INTRODUCTION

Cloud computing is new paradigm for distributed computing which delivers Infrastructure AS A Service (IaaS), Platform as a Service (PaaS), and Software as a Service(SaaS) and are made available as subscription-based services in a pay-as-you-go model to consumers computing on to the cloud (2003). It aids customers to dynamically provision computing resources around the globe, and also a. Among the various challenges faced in emerging cloud as a successful technology, the major challenge identified is scheduling tasks on to resources. Tasks identified and placed in the cloud may be static or dynamic in nature.In order to efficiently and cost effectively schedule the tasks and data of applications onto these cloud computing environments, application schedulers have different policies that vary according to the objective function: minimize total execution time (1993) (1994), minimize total cost to execute, balance the load on resources used while meeting the deadline constraints of the application, and so forth.

The most important problem area, that need to be focused in is devising algorithm to split the given application into tasks, and allocating the tasks to different processors, coordinating the communication between the tasks and so on. Allocating the tasks to the processors, should be carried out with an objective to utilize the CPU resources to its maximum and to maximize the throughput and also to reduce or maximize the turnaround time, waiting time and response time of the processes while devising a novel scheduling algorithm an important factor that is to be taken into consideration ,is the nature of the scheduling problem such as task dependency communication and data dependency process synchronization are known in prior, the scheduling is termed to be static(1995)(1990). Since the pre requisites are known in prior, to the scheduling decision, the parallel programming construct can be represented by a Directed Acyclic Graph (DAG). In certain cases, we need to make a few assumptions about the parallel processing and the scheduling ideas has to be executed instantly (1994). Such type of algorithms is dynamic in nature and does not meet the objective of minimizing the execution time of an application. Developing a static scheduling algorithm, that is capable of managing and

Corresponding Author: S.Balamurugan, Anna University, IT Department, Kalaignar Karunanidhi Institute of Technology, Coimbatore-641004, India.
Tel: +91 9952521935; E-mail: sbnbala@gmail.com.

mapping tasks of a work flow on resources will greatly aid in meeting the user specified objective function (2008). Such static scheduling algorithms, falls into two categories namely BNP(Bounded Number of Processors) scheduling algorithm, which exhibits the capability of processing and scheduling cluster of systems where the number of processing elements is limited and the computers are connected via a high speed networks such as fast Ethernet, High-bit-rate Digital Subscriber Line(HDSL), Symmetric Digital Subscriber Line(VDSL) (2006) and another category called ADN (Arbitrary Processor Network) which is capable of scheduling messages incompliance with computational task (1990). When looking at implementation perspective APN algorithms are more difficult to implement since, it exhibits the property of simultaneous scheduling of both task and messages.

This paper presents a detailed investigation on solving tasks scheduling problems typically in cloud computing environments and various existing Workflow Management Systems. The paper is organized as follows: Section 2 describes the Np-completeness problem mentioned by J.D.Ullman (1975). Section 3 gives a broad overview of techniques for solving the np-hard problem of task scheduling in Grid Environment. Section 4 gives a survey on existing tool support for solving scheduling problems in Grid Environment. This includes Grid Ant, V-LAM, a Grid Based Virtual Laboratory and Grid flow.Workflow Management Systems for Grid Computing. Section 5 describes the methodology to migrate the Workflow Scheduling problems onto clouds. Section 6 provides the overview of necessary tool support for solving the np-hard problem of scheduling on clouds which includes Aneka: a Software platform for .NET based cloud computing and CloudSim, a novel framework for modeling and simulation of cloud computing infrastructure and service. Section 7 concludes the paper giving the future research direction.

2. The NP-Completeness Problem:

Workflow applications are commonly represented as a directed acyclic graph. The mapping of jobs to the compute-resources is an NP-complete problem as mentioned by J.D.Ullman (1975) . The author proved that the problem of finding an optimal schedule for a set of jobs is NP-complete. The mapping of tasks to compute-resources is an NP-complete problem in the general form. The problem is NP-complete even in two simple cases: (in the negative a conjecture of R. L. Graham, Proc. SJCC, 1972, pp. 205-218).

(1) Scheduling tasks with uniform weights to an arbitrary number of processors and

(2) Scheduling tasks with weights equal to one or two units to two processors

As a consequence, the general preemptive scheduling problem is also NP-complete.

3. Solving The NP-Hard Problem Of Task Scheduling In Grid Environment:

L. Zhang, Y. Chen, R. Sun, S. Jing, and B. Yang(2008) has come out with an approach based on Particle Swarm Optimization(PSO) algorithm to solve task scheduling problem in grid environment such that it reduces the completion time of the associated tasks. In task scheduling mainly Genetic algorithm, Simulated Annealing and Tabusearch are used. PSO is one of the latest optimization techniques that is proposed in this paper to solve scheduling problems. In grid scheduling issues, when the number of tasks is less than the total number of resources, the resources can be allocated to each task based on the first come first serve priority but the scheduling problem arises when the number of resources is less than the number of tasks. To handle such type of scheduling problems, this paper comes up with PSO algorithm. In this algorithm each particle corresponds to an evolutionary algorithm. When a swarm of particles are used it leads to a new possible solution for the problem. To enable users to access remote resource transparently over a secure, shared scalable worldwide network, grid computing is essential.Based on utility computing models, grid computing defines a new way of service providing. Workflow management system (2005) manages and executes workflow .Workflow engine manages the execution by utilizing grid middleware.

The three major components in a work flow engine is

1. Workflow scheduling
2. Data movement
3. Fault management

The factors to be considered for scheduling workflow application

1. Resources are shared on grids
2. Resources are not under the control of scheduler
3. Resources should be heterogeneous

It is an abstract workflow model which performs without the specification of location of resources . According to J. Yu, R. Buyya, and K. Ramamohanarao (2005) there are two types of workflow model. One is the deterministic model where the dependencies of tasks and I/O known in advance and the other is Non deterministic model where in the dependencies of tasks and I/O known at run time.

The two major types of workflow scheduling are

(1) Best–Effort based scheduling

(2)QoS constraint based scheduling

A. Best-Effort based Scheduling Algorithms:

Best effort algorithms work towards the objective ofminimizing the execution time. Here the resource is shared by community model (2008). It tries to attempt to complete execution time at the earliest time or to minimize the make span of the workflow application. Authors in (2008) derive from heuristics or meta-heuristics based approach.

a) Heuristics:

There are four types of heuristics algorithms, are proposed in the scheduling literature. They are:

a. Individual task scheduling:
It is simple and based on only on one individual task. Here myopic (13) algorithm is used.
b. List scheduling:
Based on the task priority and resource selection phase, the tasks are ranked and shortlisted.

It can categorize as

i. Batch mode - for parallel dependent task. Batch mode algorithms were proposed by Maheswaran (2006).
ii. Dependency mode-for independent tasks on distributed configuration .Heterogeneous Earliest –Finish – Time (HEFT) algorithm was proposed by Topcuoglu (2002).
iii. Dependency-batch mode- it combines both dependency and batch mode. It was proposed by Sakellariou and Zhao (2004).

c. Cluster based and duplication based scheduling:
To avoid the transmission time and results between data inter dependent tasks and to reduce the overall execution time, this algorithm is employed. It was proposed by Bajaj and Agarwal (2004).

b) Meta-heuristics:

To solve computational problem, meta-heuristics algorithm were employed by several researcheres in the literature, which provides both a general structure and stratergy guidelines.

a. Greedy Randomized Adaptive Search Procedure(GRASP):
It was proposed by Feo and Resende (1995). It is a randomized iterative search technique.
b. Genetic algorithm:
It allows deriving a high quality solution from a large search space by applying the principle of evolution.

Wang et.al., (1997) have developed this algorithm to map and schedule task graph on heterogeneous environment.

The construction of genetic algorithm can be divided into four parts.

1) The choice of representation.
2) Deterministic of fitness function.
3) The design of genetic operation.
4) Determination of probabilities.

Fitness value indicates how good the individual tasks are when compared to other group of tasks. For selecting the fitness individuals, a technique called Route Wheel selection, rank selection and elitism methods are used. Compared with the heuristic based scheduling, the meta-heuristics algorithm produces an optimized scheduling solution based on the entire work flow.

B. QoS constraint based workflow scheduling:

Quality of service is an important factor to be considered .While scheduling such workflow execution can be completely maps to suitable resources. QoS not only depend upon the global scheduling decision but also depend on the local resources allocation. Most QoS constraints are based on either Time known as dead line or Costknown as budget. Main aim of Deadline constrained scheduling is to deliver the results before deadline. The need of monetary cost makes it unique from best effort algorithms. To minimize the cost two algorithms are used. One was proposed by Menasce and Casalicchio (2004) is backtracking algorithm. In backtracking algorithm, if the execution time exceeds the limit of time, it backtracks the previous step and removes the least expensive and reassigns tasks with reduced set. Other one was proposed by Yu *et al*. (2005) (2005) is Budget constrained scheduling, which distributes the deadline over the task partitions in workflow and optimizes execution cost. The scheduling time taken by the time distribution is lower than backtracking (3).

4. EXISTING Tool Support For Scheduling Problem In Grid Environment:

A. Grid Ant: A Client-Controllable Grid Workflow System

Since most of the services are directed toward the service aggregation other than the distributed process management, K. Amin, G. von Laszewski, M. Hategan, N. J. Zaluzec, S. Hampton, and A. Rossi (2004) comes with an idea in providing the grid user with an extensible client side work flow management system called gridant. Since the software available is market are commercially motivated, their research work has come up in providing a client-controllable workflow system on Grid. Grid computing mainly deals with the resource

sharing providing a flexible mechanism for dynamic scheduling. The Gridant is built using workflow engine, runtime environment, workflow vocabulary, workflow monitoring. Workflow engine deals with the direction of the flow of data and control during gridant process. It handles the failures, recovering, analyzing, task dependencies and synchronization. Ant is an existing methodology is used to manage tasks, allows incorporate grid functionality, capability of being embedded and it is a reliable foundation for workflow system. But on the other hand Ant does not allow simple executions and would require additional functionality does not support workflow composition and it does not allow output of one activity to be input for the other activity. At runtime environment white board style model communication to overcome the deficiencies in Ant. The workflow is given by grid setup created based on the clients requirements, grid authenticate which makes the initialization based on the hosting environment, grid execute which executes the grid, grid copy which copies the files between different grid resources, grid cancel to cancel the execution, grid query to check availability and grid status to find the execution status, grid ps to resolve all job execution. In work flow monitoring, XML and graphical visualization tool is provided to program workflow and to accommodate large specification. To study the application position resolved diffraction technique is used with diffraction technique and passes the results between multiple grid environment. It uses data acquisition, back up, data analysis and result display. In workflow services, proxy services are used to allow user to submit their workflow and get disconnected from the system. It is not necessary for the user to be always connected. The proxy service ubiquitously maintains a connection for the user with the system.

The service is of two modules, one is gridant core to accept XML specification, communicate between task, workflow control, information between event mapper. The other one is the event mapper which logs all events generated and also used by the client to remotely monitor the work flow programs. The future work proposed by the authors is to extend the GridAnt system into a service-oriented architecture in order to act as a proxy on behalf of the Gridusers. An advanced information exchange mechanism is under research by authors to incorporate dynamic publish/subscribe style of communication between independent tasks. The currentimplementation of GridAnt supports the Globus Toolkit versions 2 and 3. In future research directions may be focussed to make support to other Grid environments such as Unicore, Legion, and Condor.

B. VLAM-G: A Grid-Based Virtual Laboratory

H. Afsarmanesh, R. G. Belleman,A. S.Z.Belloum, A. Benabdelkader, J. F. J. van den Brand, G. B. Eijkel, A. Frenkel, C. Garita, D. L. Groep, R. M. A. Heeren, Z. W. Hendrikse, L. O. Hertzberger, J. A. Kaandorp, E. C. Kaletas, V. Korkhov, C. T. A. M. de Laat, P. M. A. Sloot, D. Vasunin, A. Visser, and H. H. Yakali (2002) proposed a Grid Based Virtual Laboratory AMsterdam (VLAM-G),the authors have proposed a model for remote cloud computing, distributive analysis and interaction between the fundamental services and the application programs. The VLAM-G is built using Globus Toolkit which is well suited for grid development. Since this is capable of supporting only low level middle tier, additional layers are needed for the usability of the scientists. Hence VLAM-G has developed a scientific community to make use of preexisting geographically dispersed resources. These are derived from chemo-physics, from bio-informatics, and from medical and systems engineering which shares common technology for the case study. The VLAM-G toolkit helps the scientists in carrying out their experiments using workflow templates. It provides GUI, runtime system, information management to experiment hardware and software. The toolkit also helps in building the distributed network through the availability of sufficient network resources helping the scientists to carry out their experiment through a distributed environment. To support collaboration experiments, per experiment manager is employed for credential brokering and forwarding. The study proposed by authors helps in collaboration application and provide distributed resources to scientists with less detailed knowledge over middleware. The study explains with a step by step work flow process in solving a problem. This is implemented using Process Flow Template (PFT) instantiating a PFT into a Process Flow Instantiation (PFI). The PFT comprises data description, automated execution on distributed infrastructures, combining the generic function and features from case study using VLAM-G lead to an Experimentation Environment database model. The modules that contain generic software and Data Flow Graph (DFG) are used. The VLAM-G architecture consists of user interface, session manager with RTS to provide staged interface with grid service, collaboration to support audio and video availability, modular repository and federation information management for accessing RTS and PFT database. The RTS in VLAM-G directly controls the resources involved and data transfer between them. The modules are sent in the form of parameters in always active state. It supports the connection through uni-directional ports between different processing element, module in different machine and device bounded to specific machines. The RTS consists of module launcher to initiate the execution, module connector to implement the third party arbitration through Grid FTP data channels and a module controller to monitor the module status after instantiated with the help of SITE command. In practical application the data is retrieved, then it is transformed and executed with FFT, visualized using VR environment and final data is stored in an external storage using the Grid FTP server.

C. GRIDFLOW :A Workflow Management System for Grid Computing:

For large scale distributed resources sharing and system integration, Grid computing is used as one of the main technology. Grid computing on workflow management is referred as Gridflow J. Cao, S. A. Jarvis, S. Saini, and G. R. Nudd (2003). Services of the grid include Information services, Resource management, Data transfer and Security. Gridflow describes, frame work development and supporting algorithms A two-tier service framework is implemented with both global grid workflow management and local grid sub-workflow scheduling. Grid workflow provides, Workflow construction, Simulation, Scheduling and Execution, monitoring and conflict solving. For global grid, an agent based methodology is developed. For local grid, a system implementation, ARMS and titan is implemented. Titan uses the heuristics algorithm to dynamically minimize the make span and idle time of particular grid resources without destroying user contracts. The aim of their work is to monitor grid application to flow of multiple tasks.

To implement and define the grid, it requires following new challenges:

I. Cross domain: The lack of central ownership and organization results in incomplete information.

II. Dynamism: The resource allotment to the environment can vary dynamically. Hence performance of application becomes difficult.

Globus is used to provide grid resource information and indexing services.For modeling and prediction capabilities for parallel program, PACE toolkit are used. The implementation is carried out at multiple layers.

1) Grid Resource: These can be multiprocessors or clusters of workstations or PCs with maximum storage space. Titan is used as a grid resource manager that defines the execution of multiple tasks.
2) Local Grid: A local grid consists of multiple grid resources that belong to one organization.
3) Global Grid: The global grid includes all grid resources belongs to different organization. Authors used ARMS as management system.

i. PACE:

To provide a system of application performance modeling and evaluation, PACE toolkit is used for both local schedulers and grid agents.

Pace evaluation engine is used to combine application and resource models at run time to produce performance data. PACE performs using ASCI high performance computing application.

ii. TITAN:

Titan is a grid resource manager. By coupling application performance with heuristic algorithms, the system is able to dynamically balance the process of minimizing makespan of multiple tasks and idle time of multiprocessors. At first, resource is passed to the task management ,where they form queue. Resource monitor provides gathering of static concerning on which tasks may execute .Using heuristic algorithm, the scheduling process search for near optimal schedules for the current task queue. It also acts as the grid resource information provider in Globus MDS implementation.

iii. ARMS:

ARMS are composed of agents. Agents are defined as representation of a local grid at a global level of grid resource management. Using globus MDS, agent store the local grid resource information and those advertised from other agents. Agents are also used to find available resources for the task. To optimize agent performance, different strategies are used, which is controlled using performance monitor and advisor. Grid workflow user portal provides GUIs and additional grid services. The implementation of grid workflow management is carried out at several layers.

a) Task: Tasks are the smallest element in a grid. Tasks are MPI and PVM jobs running on multiple processors. Using titan, task scheduling can be implemented.
b) Sub-workflow: Sub workflow is the flow of task that is to be executed.

i. Gridflow user portal:

It enables user to construct a grid workflow and access grid service. The portal provides direct user interfaces to the information and performance services.

ii. Global grid workflow:

It receives request from the grid flow portal.

It provides three main functionalities.

- Simulation
- Execution
- Monitoring

GGWM algorithm is proposed to simulate the sub-workflow and LGSS algorithm is implemented for local grid sub-workflow scheduling.

5. Migrating The Workflow Scheduling Problem To Clouds:

RajkumarBuyya, SurajPandey, and Christian Vecchiola (2005) defined architecture for creating market oriented clouds and computing atmosphere. Everyone access the service based on their requirements without regard to where the service are hosted. This analogy is termed as cloud computing. Cloud computing provides services, available as subscription based services. The services may be Infrastructure as a service (IaaS), Platform as a service (PaaS) and Software as a service (SaaS). Cloud provides datahouse that users can access and deploy application from anywhere in the world at competitive cost depend on the quality of service (QoS). Cloud computing is defined as "Gartner's IT Hype Cycle". Hype cycle is used to represent the emergence,adoption,maturity and impact on applications of specific technologies. Service oriented arthitectural framework consisting of client's broker and coordinating services that enchances utility-driven management of clouds. To bring services producers and consumers together, cloud exchange is used. The architectural design is concerned with Security, Privacy and Trust. Besides security, there are legal and regular issues to be take care of. Virtualization technologies provides features of isolation,QoS. Sandboxing (2005) and managed execution are considered by programming level virtualization. User can access the resource by using application directly or through meta-broker as indirectly. A cloud broker client on the client side, that will interact with the mete-broker by requesting the desired QoS. The meta-broker will look up and match the best one among several cloud providers.mCloud federation is the arrangement model that is used to custom service among cloud providers to increase their value of services.mThecloudbus toolkit is used to design and develop the components for market oriented cloud computing.

C. Vecchiola, M. Kirley, and R. Buyya(2009) comes out with an EMO (Evolutionary Multiobjective Optimization) algorithm which is a distributed implementation of network based multi object evolutionary. The network between large populations is more efficient for the algorithm to be implemented controlling the computational time. The multi object evolutionary algorithm(MOEA) is established for getting solutions on problem parameters conflicting objects and constrains. Hence a complex network based MOEA called EMO is derived to inherit the challenges faced by MOEA mainly problems with large number of objectives. Distributed implementation is done using Offspring, a framework that is developed for execution and dissipating the large computation load generated. This is achieved by just asking the user the required field in distributed execution. The computational power to solve a problem in a reasonable time is provided by Enterprise cloud.

The main objective of the authors is to provide a friendly and simple environment with the user for optimization. The distributed version is created just using simple strategies such as combining serial executions, applying smart migration at the end of each iteration algorithm. The background of this idea is based on the influences of various tools that have made the distributed problem solving technique to be simple. The distributed evolutionary algorithm is fully based on the step by step process in analyzing the optimal solution for the problem. The term population is used which denotes a group of individuals. Each individual holds an optimal solution and these are combined together to form a new optimal solution. The population model are of four types, they are single master slave population which deals with the evolutionary phase, multiple population in which the whole algorithm is executed, cellular and hierarchical combination. The evolutionary algorithms are carried out in two levels, the first level uses multiple populations and second level uses the master slave model. In complex network model based on diffusion based evolution algorithm, each individual is taken as nodes. The individual combines with other defined by a topology in the network and the evolution Prento optimal front is stored in an external achieve. The drawback of this model is the variation in connectivity and selection individual in a network. The network architecture also shows variation when considered with ZDT multi objective problem. In order to handle these issues a scalable hierarchical version of complex network model is used in employing multiple isolated population. In this method each individual is mapped within a separate topology and evaluated. The required settings can be applied on each topology by the user. After the evaluation all the individuals are mounted to a control coordination node which combines all the nodes, analysis statically and applies migration over the individuals belonging to different population.

The Architecture is developed in such a way that the user can quickly implement the algorithm with a less knowledge about enterprise clouds and grid, with simplicity, step by step rapid development, distributed execution, and result analysis. The system view is developed based on the environment for running, monitoring and control distributed applications, and the middleware takes care of interactions with the enterprise cloud and a reference model for the implementation. The distributive applications are created by offspring in two models by developing a plugin to control the interaction with the cloud, and simply defining the logic that provide the environment to be iterated and executed. Many cloud computing capabilities were introduced in Aneka with SLA oriented resource allocation. This platform helps in selecting the node and it also provides a high programming interface with cloud computing. The offspring is loaded over Task Model supported by Aneka which helps in writing simple distributed application with no knowledge on middleware. At the implementation phase, EMO algorithms are used in creating reasonable individuals over network model and to apply the migration strategies at different topologies. To implement EMO++ authors define EMO strategy, EMO task and EMO datapipe. For remote mode executions, an EMO algorithm is started with proper input values,

configuration parameters and results. The implementation is done using strategy controller on Aneka (a Cloud computing tool) which prefers execution and analysis of the data . Depending upon the computing, storaging and communication resources, the cost of running an application would vary. Several research projects are used to compute cycles on the national and international cyber infrastructure resources such as those of the tera grid or EGEE project. Such solution requires different levels of cost and implementation and level of service. Cloud computing, should be low cost, operating maintenance and upgrading a local computing infrastructure. To achievethis, an application called montage is used.

Montage, which delivers science grade mosaics of the sky to the community, composed of both professional and amateur astronomers.

Amazon (11) supports to handle montage on the cloud by

1. Handling sporadic overloads of mosaics requests.
2. Provide resources for all its computations.
3. Support both computation long term data storage.

Montage E. Deelman, G. Singh, M. Livny, B. Berriman, and J. Good (2008)is a general engine that is used for computing input images. Montage is a data intensive application. Montage application can be able to run in a resource rich environment where availability of storage resource can be assured. NASA's earth science technology office was funded montage and it is maintained by RSA. Amazon can be used as a basic service model.It can be able to provide both compute and storage resources. Amazon S3 storage is used for REST and HTTP transfer protocol. Pegasus E. Deelman, G. Singh, M.-H. Su, J. Blythe, Y. Gil, C. Kesselman, G. Mehta, K. Vahi, G. B. Berriman, J. Good,A. Laity, J. C. Jacob, andD. S. Katz (2005) is used as workflow management system. For using cloud storage resources, there are three implementation models namely Remote I/O, Regular and Dynamic clean up. The authors use the following three montage workflows in their simulations.

1. Montage 1 degree: A montage workflow for creating a 1 degree square mosaics of the M17 region of the sky. It consists of 203 application tasks. Here the number of processors are independent of data transfer cost. While increase the number of processors, it's storage cost got decline but the CPU cost increase when the number of processors increased, execution time also increased.

2. Montage 2 degree: A montage workflow for creating a degree square mosaics of M17 region of the sky. It consists of 731 applications. The total number of cost for the processors gets increased when the execution time decreases.

3. Montage 4 degree: A montage workflows for creating a 4 degree square mosaics of M17 region of the sky. It consists of 3,027 application tasks. It combines the results of Montage 1 and Montage 2. Here when the number of processors increased ,the cost gets high, although the execution time gets decreased.

This workflow can be created using the mDAG component in the Montage distribution.

After simulation, the authors determine the following metrices:

1. The workflow execution time.
2. The total amount of data transferred from user to storage resources.
3. The total amount of data transferred from storage resources to user.
4. The storage used at the resource in terms of GB-hours.

As a result, the authors' examined the tradeoffs between using three different data management solution.

1) Remote I/O,where tasks access data as needed.
2) Regular, where data are brought in at the computation
3) Cleanup, where data no longer needed are deleted as the workflow progresses.

When the storage cost is higher and transfer rate is lower, it is possible that remote I/O mode would have resulted in the least total cost among the three. Montage (2008) is only the scientific application that can potentially benefit from cloud services. The storage cost with and without cleanup increases when CCR (communication to the computation ratio) increases. When the size of the data increases, the transfer cost also increases. CPU cost also increases due to increase in execution time of the workflow. When CCR increases, both the transfer and storage cost also increases. Hence authors concluded that, when applications become more data-intensive, the transfer cost can be reduced and beneficial to store all the input data in the cloud.

6. SOLVING The NP-Hard Workflow Scheduling Problem In Clouds:

A. Aneka: A Software Platform for .NET-based Cloud Computing:

C. Vecchiola, X. Chu, and R. Buyya (2009) developed a software platform for .NET based cloud computing called Aneka, which provides a runtime environment and exhibits the ability of supporting multiple programming models. It is based on the .net framework and this makes unique as it is available on java based technology. For developing distributed application on the cloud, Aneka is used, which provides platforms and frameworks. It is portable over different platforms and operating systems. It is a collection of physical and virtualized resources connected through internet or intranet. It offers different programming model for different applications. Aneka container represents the run time environment .the container provides the basic management

features. Aneka middleware provides a set of basics features on the top of which each of the Aneka containers can be specialized to perform specific task. It also provides persistence and security hosted by the container.
To develop the application, it has to fulfill the following aspects.
1) Simplify the development of application.
2) Porting existing application in the cloud.
3) Monitoring and managing the Aneka cloud.

Using resource virtualization, the interconnected resources can be modified dynamically according to the user nodes. The Aneka container is the heart for this infrastructure. The container provides the management features. The features of the container include fabrication, Foundation and transversal and execution. Aneka container can easily evolve and integrate new features with minimum setup costs.

It is the lowest level of the software stack in the Aneka container. It can directly interface with the hosting resources and responsible for low level operations. Platform abstraction layer (PAL) allows the container to be completely independent of from the hosting machine and operating systems.

Aneka middleware provides runtime support for execution service and application. It constitutes Directory and membership-responsible for setting up and maintaining node information, Resource reservation-it identifies the ability of reserving a set of nodes, Storage management-to implement the distributed system ,availability of disk space is more important. It provides flexibility and extensibility. It is responsible for robust, file based storage for application and Licensing, accounting and pricing, to protect the systems from misuse. It controls on the number of resources. It co-ordinates the execution of application and responsible for job collection. The task model, thread model and the map reduce model are implemented under this service. Using Aneka software kit, we can develop the application. It contains basic class libraries that provide the feature of developing application. Programming model defines the way of design. Types of programming model includes task programming model, thread programming model and map reduce model. While deploying on clouds using Aneka the aspects to be taken into consideration are traversal service, persistence, security and portability and interoperability. After development, it is essential to maintain and execute their application. The set of operation to be performed for maintenance are Setting up cloud computing, Remote installation and configuration, Controlling containers and System load monitoring and maintenance.

*B. **CloudSim:***

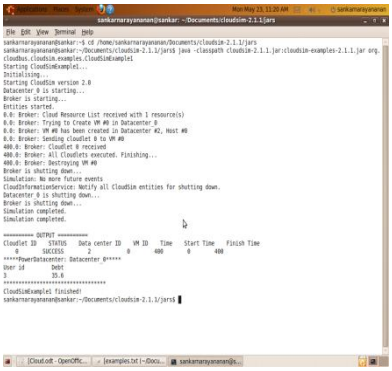
A Novel Framework for Modelling and Simulation of Cloud Computing Infrastructures and Services

CloudSim (33) is a multilayered design with a architectural components in it. CloudSim is a SimJava event simulation engine which will supports the core functionalities which will includes the network queening techniques, event processing and Cloud systems entities. The cloud sim entities are such that data centers, virtual machine, services, hosts, brokers. The CloudSim defines the communications between the cloud system entities and as well as the Simulation clock.

The cloudsim uses the SimJava hence simulation of data centers and virtual machines, storage, memory and bandwidth can be virtually simulated and also the Cloud providers must have a efficiency in communications with the VMs (Virtual Machines). The cloud has to work concurrently for the different VMs in a same period of time for the SaaS therefore the QoS have to be enriched. Not even this several scenarios must be overcomes such as the 1) workload mix in the datacenters, 2) Robusted configurations based scenarios in the cloud. 3) Implementation of the own applications techniques in the clouds

The CloudSim consists of the different layers with different simulating components, each components have its own specified unique tasks to be completed. The cloudsim package effectively tackles the Infrastructure as a service (IaaS) and the application as a service (SaaS) complexities. So that the core algorithm, policies, executing algorithms can be easily defined.

Since the cloudsim tackles the IaaS lets us see about the modeling which is related to the Infrastructure. In this IaaS we are going to deal about the simulating work of the extending the number of data centers and its entities, the number of hosts assigned to the Virtual Machines (VM). VM allocation policies in the cloud provider, VM creation, VM destruction, migration of the virtual machines, applications provided to the different VMs. The datacenters will manages the number of hosts connected to it, and manages the VMs during their life cycle period, provisioning policies for memory, allocations, storages, processing capabilities. For both the single-core nodes and the multi-core nodes. To simulate the user own programs the cloud must have minimum configurations of a basic system with a CloudSim package and the java toolkit 1.6 or any other enhanced version.



Fig, 1: One data centre with a single Virtual Machine.

In Fig 1 we created a data center with one virtual Machine. Like this the programmers can develop according to their own organization now we try for a scalable simulation

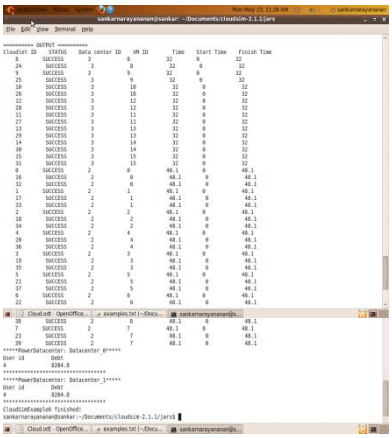


Fig. 2: Creating a scalable Cloud with 'n' nodes and 'm' Virtual Machines.

CONCLUSION and Future Work:

In a workflow, the application is divided into set of dependent tasks and scheduling process maps and manages execution of inter-dependent tasks on distributed resources and allocates necessary resources to workflow tasks so that the execution can be completed to satisfy objective(s) of by user. This paper presented a detailed literary review on applying of PSO in solving np-hard problems like scheduling and task allocation. This paper presented a detailed investigation on solving np-hard problems of scheduling and task allocation in cloud computing environment. The paper also focused on delivering a survey on existing tool support for solving scheduling problems in Grid Environment. This includes Grid Ant, V-LAM, a Grid Based Virtual Laboratory and Gridflow A Workflow Management Systems for Grid Computing. It also gave some methodologies to migrate the Workflow Scheduling problems onto clouds and provides the overview of necessary tool support for solving the np-hard problem of scheduling on clouds which includes Aneka: a Software platform for .NET based cloud computing and CloudSim, a novel framework for modeling and simulation of cloud computing infrastructure and service.

As future work, we are conducting our research in employing PSO and its variants (PSO with fixed inertia, PSO with variable inertia, PSO with elitism, Parallel PSO, Hybrid PSO, Orthogonal PSO and Parallel orthogonal PSO) to develop and to propose new strategies and algorithms to minimize the cost incurred to schedule task onto efficiently selected resources, transfer data from/to resources at reduced cost and map the developed heuristic strategy to execute workflow application.

REFERENCES

Zhang, L., Y. Chen, R. Sun, S. Jing and B. Yang, 2008. “A task scheduling algorithm based on pso for grid computing,” *International Journal of Computational Intelligence Research*, 4-1.

RajkumarBuyya, SurajPandey and Christian Vecchiola, 2005. "Cloudbus Toolkit for Market-Oriented Cloud Computing", E. Alba, E.-G. Talbi, G. Luque, and N. Melab, *Parallel Metaheuristics: A New Class of*

Algorithms, ser.Wiley Series on Parallel and Distributed Computing. Wiley, ch. 4.Metaheuristics and Parallelism, 79-104.

Yu, J., R. Buyya and K. Ramamohanarao, 2008. "Workflow Scheduling Algorithms for Grid Computing," in *Metaheuristics for Scheduling in Distributed Computing Environments*, ser. Studies in Computational Intelligence. Springer Berlin / Heidelberg, 146: 173-214

Pandey, S., W. Voorsluys, M. Rahman, R. Buyya, J.E. Dobson and K. Chiu, 2009. "A grid workflow environment for brain imaging analysis on distributed systems," *Concurrency and Computation: Practice & Experience*, 21(16): 2118-2139.

Vecchiola, C., M. Kirley and R. Buyya, 2009. "Multi-Objective Problem Solving With Offspring on Enterprise Clouds," in *Proceedings of the 10th International Conference on High-Performance Computing in Asia-Pacific Region (HPC Asia 2009)*. Kaohsiung, Taiwan: IEEE.

Cao, J., S.A. Jarvis, S. Saini and G.R. Nudd, 2003. "GridFlow: Workflow Management for Grid Computing," in *CCGRID '03: Proceedings of the 3st International Symposium on Cluster Computing and the Grid*. Washington, DC, USA: IEEE, 198.

Amin, K., G. von Laszewski, M. Hategan, N.J. Zaluzec, S. Hampton and A. Rossi, 2004. "GridAnt: A Client-Controllable Grid Worklow System," in *HICSS '04: Proceedings of the Proceedings of the 37th Annual Hawaii International Conference on System Sciences - Track 7*. Washington, DC, USA: IEEE.

Afsarmanesh, H., R.G. Belleman, A.S.Z. Belloum, A. Benabdelkader, J.F.J. van den Brand, G.B. Eijkel, A. Frenkel, C. Garita, D.L. Groep, R.M.A. Heeren, Z.W. Hendrikse, L.O. Hertzberger, J.A. Kaandorp, E.C. Kaletas, V. Korkhov, C.T.A.M. de Laat, P.M.A. Sloot, D. Vasunin, A. Visser and H.H. Yakali, 2002."Vlam-g: A grid-based virtual laboratory," *Scientific Programming*, 10(2): 173-181.

Deelman, E., G. Singh, M. Livny, B. Berriman and J. Good, 2008. "The cost of doing science on the cloud: the Montage example," in *SC '08: Proc. of the 2008 ACM/IEEE conference on Supercomputing*, Piscataway, NJ, USA, 1-12.

Vecchiola, C., X. Chu and R. Buyya, 2009. High Speed and Large Scale Scientific Computing. IOS Press, ch. Aneka: A Software Platformfor .NET-based Cloud Computing, 267-295, iSBN: 978-1-60750-073-5.

http://aws.amazon.com/

Deelman, E., G. Singh, M.H. Su, J. Blythe, Y. Gil, C. Kesselman, G. Mehta, K. Vahi, G. B. Berriman, J. Good, A. Laity, J.C. Jacob and D.S. Katz, 2005. "Pegasus: Aframework for mapping complex scientific workflows onto distributed systems," Scientific Programming, 13(3): 219-237.

Wieczorek, M., R. Prodan and T. Fahringer, 2005. Scheduling of Scientific Worklows in the ASKALON Grid Enviornment, ACM SIGMOD Record, 34(3): 56-62.

Metropolis, N., *et al*., 1953. Equations of state calculations by fast computing machines. Joural of Chemistry and Physics, 21: 1087-1091.

Topcuoglu, H., S. Hariri and M.Y. Wu, 2002. Performance-E®ective and Low-Complexity Task Scheduling for Heterogeneous Computing, IEEE Transactions on Parallel and Distributed Systems, 13(3): 260-274.

Sakellariou, R. and H. Zhao, 2004. A Hybrid Heuristic for DAG Scheduling on Heterogeneous Systems, The 13th Heterogeneous Computing Workshop (HCW 2004), Santa Fe, New Mexico, USA, April 26, 2004.

R. Bajaj and D. P. Agrawal, Improving Scheduling of Tasks in a HeterogeneousEnvironment, IEEE Transactions on Parallel and Distributed Systems, 15: 107-118.

Feo, T.A. and M.G.C. Resende, 1995. Greedy Randomized Adaptive Search Procedures, Journal of Global Optimization, 6: 109-133.

Wang, L., *et al*., 1997. Task Mapping and Scheduling in Heterogeneous Computing Environments Using a Genetic-Algorithm-Based Approach, Journal of ParallelBand Distributed Computing, 47: 8-22.

Menascµe, D.A. and E. Casalicchio, 2004. A Framework for Resource Allocation in Grid Computing, The 12th Annual International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunications Systems (MASCOTS'04),Volendam, The Netherlands, 5-7.

Yu, J. and R. Buyya, 2005. A Taxonomy of Work°ow Management Systems for Grid Computing, Journal of Grid Computing, Springer, 3(3-4): 171-200. Spring Science Business Media B.V., New York, USA, Sept.

Yu, J., R. Buyya and C.K. Tham, 2005. A Cost-based Scheduling of Scientific Workflow Applications on Utility Grids, The ¯rst IEEE International Conference on e-Science and Grid Computing, Melbourne, Australia, 5-8.

Yu, J. and R. Buyya, 2006. Scheduling Scientific Workflow Applications with Deadline and Budget Constraints using Genetic Algorithms, Scientific Programming,b 14(3-4): 217 - 230, IOS Press, Amsterdam, The Netherlands.

Graham Rilchie, 2003. "Static Multiprocessor Scheduling with Ant Colony optimization and Local search", Master of search", Master of science Thesis, University of EdiEdinburgh.

Osman, I., Li, 1993. "Metastrategy Simulated annealing and tabu search algorithms for vehicle routing problems", Annals of operations Research, 41(4): 421-451.

Edwin. S.H., Hou, NinvanAnsan and Hong Ren, 1994. "A genetic Algorithm for Multiprocessor Scheduling", IEEE Transactions on Parallel and Distributed Systems, 5-2.

Consard, M. and M. Loi, 1995. Automatic task Graph Generation Techniques Parallel Processing letters, 5(4): 527-538.

Wu, M.Y. and D.D. Gajski, 1990. Hypertool: A Programming Aid for Message Passing Systems. IEEE Transactions on Parallel and Distributed Systems, 1(3): 330-343.

EI. Rewini, H. and T.G. Lewrs and H.H. Ali, 1994. Task Scheduling in Parallel and Distributed Systems, Eaglewood Cliffs, New Jersey Prentice Hall.

Yu, J., R. Buyya and Ramamanohararao, 2008. Workflow Scheduling Algorithms for Grid Computing, Volume 146/ 2008, Chapter, 7: 173-214. Springer Berlin/ Heidelberg.

Behrouz, A. Forouzan, 2006. Sophia Chung Fegan, "Data Communication and Networking" Fourth Edition, Tata McGrawHill, 254-256.

Buyya, R., 1990. "Higher Performance Cluster Computing-Architecture and Systems", Volume, Pearson Education, ISBN 81-317-1693-7.

http://www.buyya.com/gridbus/cloudsim/

Ullman, J.D., 1975. "NP-complete scheduling problems," Journal of Computer and System Sciences, 10: 384-393.