ISSN (ONLINE): 2349-7084
GLOBAL IMPACT FACTOR 0.238
I2OR PIF 2.125

INTERNATIONAL JOURNAL OF COMPUTER ENGINEERING IN RESEARCH TRENDS
VOLUME 2, ISSUE 10, OCTOBER 2015, PP 694-698

Combined Resource Provisioning and Scheduling Strategy for execution of scientific workflows on Cloud Level of IaaS

Chetana Pradip Shravage, Dr. S.T. Singh

Abstract— Cloud computing is that the latest distributed computing model and it offers big opportunities to resolve large-scale scientific issues. However, it presents varied challenges that require to be addressed so as to be with efficiency utilized for progress applications. Although the advancement programing downside has been wide studied, there area unit only a few initiatives tailored for cloud environments. Furthermore, the present works fail to either meet the user's quality of service (QOS) needs or to include some basic principles of cloud computing like the physical property and no uniformity of the computing resources. This paper proposes a resource provisioning and programing strategy for scientific workflows on Infrastructure as a Service (IaaS) clouds. we tend to gift associate algorithm supported the meta-heuristic improvement technique, particle swarm improvement (PSO), that aims to reduce the general workflow execution value whereas meeting point in time constraints. Our heuristic is evaluated victimization CloudSim and numerous well-known scientific workflows of various sizes. The results show that our approach performs higher than the present progressive algorithms.

Index Terms— Cloud computing, resource provisioning, scheduling, scheduling strategy, scientific workflow, Iaas

1. Introduction

THIS Cloud computing is a recently evolved computing terminology or metaphor based on utility and consumption of computing resources. Cloud computing involves deploying groups of remote servers and software networked that allow centralized data storage and online access to computer services or resources. Clouds can be classified as public, private or hybrid.

The criticisms about it are mainly focused on its social implications. This happens when the owner of the remote servers is a person or organization other than the user, as their interests may point in different directions, for example, the user may wish that his or her information is kept private, but the owner of the remote servers may want to take advantage of it for their own business.

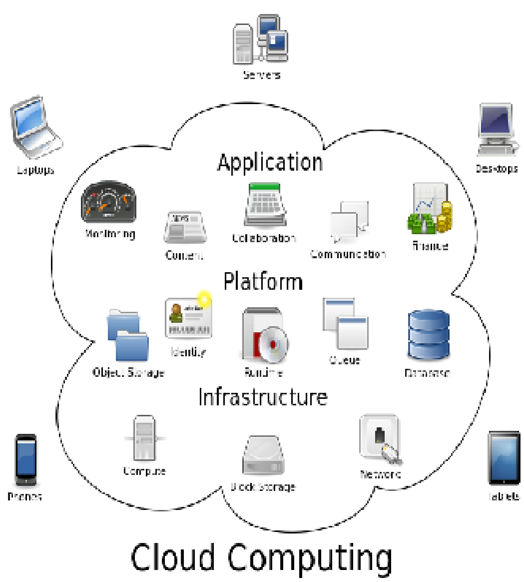

Fig 1: Cloud computing metaphor: For a user, the network elements representing the provider-rendered services are invisible, as if obscured by a cloud.

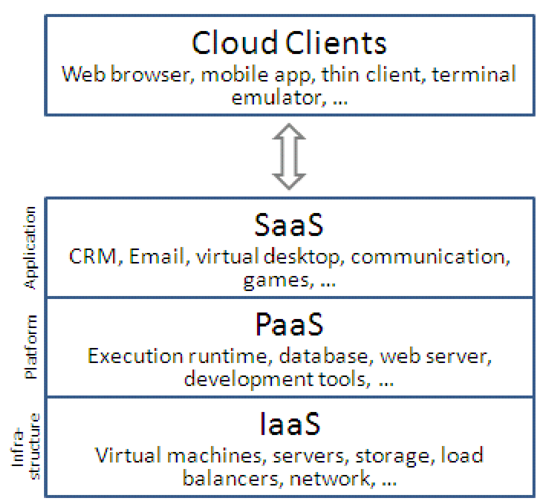

Fig 2: Service models

Service models-Cloud computing providers offer their services according to several fundamental models-Infrastructures as a ser-

ISSN (ONLINE): 2349-7084
INTERNATIONAL JOURNAL OF COMPUTER ENGINEERING IN RESEARCH TRENDS
VOLUME 2, ISSUE 10, OCTOBER 2015, PP 694-698

vice (IaaS). In the most basic cloud-service model & according to the IETF (Internet Engineering Task Force), providers of IaaS offer computers – physical or (more often) virtual machines – and other resources. (A hypervisor, such as Xen, Oracle Virtual Box, KVM, VMware ESX/ESXi, or Hyper-V runs the virtual machines as guests. Pools of hypervisors within the cloud operational support-system can support large numbers of virtual machines and the ability to scale services up and down according to customers' varying requirements.) IaaS clouds often offer additional resources such as a virtual-machine disk image library, raw block storage, and file or object storage, firewalls, load balancers, IP addresses, virtual local area networks (VLANs), and software bundles.IaaS-cloud providers supply these resources on-demand from their large pools installed in data centers. For wide-area connectivity, customers can use either the Internet or carrier clouds (dedicated virtual private networks).

Workflows are often accustomed model large-scale scientific issues in areas like bioinformatics, astronomy, and physics [1]. Such scientific workflows have ever-growing information and computing needs and therefore demand superior computing surroundings in order to be dead during a affordable quantity of time. These workflows are unremarkably shapely as a collection of tasks interconnected via information or computing dependencies.

The orchestration of those tasks onto distributed resources has been studied extensively over the years that specialize in environments like grids and clusters. However, with the emergence of latest paradigms like cloud computing, novel approaches that address the actual challenges and opportunities of those technologies ought to be developed.

As a years passes, distributed environments have evolved from shared community platforms to utility-based models; the latest of those being cloud computing. This technology enables the delivery of IT resources over the web [2], and follows a pay-as-you-go model wherever user's ar charged based on their consumption. There ar numerous forms of cloud suppliers [2], every of that has completely different product offerings. They're classified into a hierarchy of as-a-service terms: package as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS). This paper focuses on IaaS clouds which provide the user a virtual pool of unlimited, heterogeneous resources that may be accessed on demand. Moreover, they provide the pliability of elastically acquiring or cathartic resources with varied configurations to best suit the necessities of associate degree application. Even though this empowers the users and provides them additional management over the resources, it additionally dictates the event of innovative programming techniques in order that the distributed resources ar with efficiency utilised.

There square measure 2 main stages once designing the execution of a work flow in a very cloud setting. The primary one is that the resource provisioning phase; throughout this stage, the computing resources which will be accustomed run the tasks square measure designated and provisioned. Within the second stage, a schedule is generated and each task is mapped onto the best-suited resource.

The selection of the resources and mapping of the tasks is done in order that completely different user outlined quality of service (QoS) requirements square measure met. Previous works during this space, especially those developed for Grids or Clusters, centered principally on the planning section. the rationale behind this can be that these environments offer a static pool of resources that square measure readily accessible to execute the tasks and whose configuration is known prior to. Since this can be not the case in cloud environments, each issues got to be addressed and combined so as to supply associate economical execution set up.

Another characteristic of previous works developed for clusters and grids is their specialize in meeting application deadlines or minimizing the make span (total execution time) of the work flow whereas ignoring the price of the utilized infrastructure. While this can be well matched for such environments, policies developed for clouds square measure duty-bound to contemplate the pay-per-use model of the infrastructure so as to avoid preventative and supererogatory prices.

2. LITERATURE SURVEY

Workflow programming on distributed systems has been widely studied over the years and is NP-hard by a discount from the digital computer programming downside [7]. Therefore it is not possible to get associate degree optimum answer inside polynomial time and algorithms target generating approximate or near-optimal solutions. Numerous
Algorithms that aim to search out a schedule that meets the user's QoS necessities are developed. a colossal vary of the proposed solutions target environments similar or equal to community grids. this implies that minimizing the application's execution time is usually the programming objective, a restricted pool of computing resources is assumed to be accessible and therefore the execution price is never a priority.
For instance, Rahman[9] propose an answer supported the workflow's dynamic crucial ways, Chen and Zhang [10] elaborate associate degree formula supported hymenopterous insect colony optimization that aims to fulfill completely different user QoS necessities and, finally, Yu and Buyya use Genetic Algorithms to implement a budget strained programming of workflows on utility Grids [11].
The same solutions offer a valuable insight into the challenges and potential solutions for advancement scheduling. However, they're not optimum for utility-like environments like IaaS clouds. There area unit varied characteristics specific to cloud environments that require to be thought-about when developing a programming formula. Mao and Humphrey propose a dynamic approach for programming advancement ensembles on clouds [12]. They acknowledge that there area unit varied styles of VMs with completely different prices which they'll be chartered on demand, depending on the application's necessities. What is more, they tailor their approach in order that the execution price is decreased based on the cloud's valuation model, that is, VMs are paid by a fraction of your time that in most cases is one hour. They try to reduce the execution price

ISSN (ONLINE): 2349-7084
INTERNATIONAL JOURNAL OF COMPUTER ENGINEERING IN RESEARCH TRENDS
VOLUME 2, ISSUE 10, OCTOBER 2015, PP 694-698

by applying a collection of heuristics like merging tasks into one one, characteristic the most cost-efficient VM kind for every task and consolidating instances. Though this is often a legitimate approach capable of reducing the execution price of workflows on clouds, the answer planned solely guarantees a discount on the value and not a near-optimal answer.

Another recent work on advancement ensemble developed for clouds is given by Malawski et al. [13]. They propose various dynamic and static algorithms that aim to maximize the number of labor completed, that they outline as the range of dead workflows, whereas meeting QoS constraints like point in time and budget. Their solutions acknowledge completely different delays gift once handling VMs chartered from IaaS cloud suppliers like instance acquisition and termination delays. What is more, their approach is powerful within the sense that the task's calculable execution time might vary supported a homogenous distribution and they use a price margin of error to avoid generating a schedule that goes over budget. Their work, however, considers solely a single kind of VM, ignoring the heterogeneous nature of IaaS clouds.

While the algorithms given by Mao and Humphrey [12] and Malawski et al. [13] area unit designed to figure with workflow ensembles, they're still relevant to the work done in this paper since they were developed specifically for cloud platforms and as therefore embrace heuristics that attempt to embed the platform's model. Additional in line with our work is the solution given by Abrishami et al. [14] which presents a static formula for programming one advancement instance on associate degree IaaS cloud. Their formula is predicated on the Workflow's partial crucial ways and it considers cloud options Such as VM no uniformity, pay-as-you-go and time.

Other authors have used PSO to unravel the advancement scheduling downside. Pandey et al. [15] propose a PSO primarily based algorithm to reduce the execution price of one advancement while reconciliation the task load on the accessible resources. While the value minimization objective is extremely desired in clouds, the load reconciliation one makes additional sense in an exceedingly non-elastic setting like a cluster or a grid. The execution time of the advancement isn't thought-about within the programming objectives and so this worth may be significantly high as a results of the value minimization policy. The authors do not contemplate the physical property of the cloud and assume a set of VMs is out there beforehand

Wu et al. [16] conjointly use PSO to provide a near-optimal schedule. Their work focuses on minimizing either price or time whereas meeting constraints like point in time and budget. Despite the very fact that their heuristic is ready to handle heterogeneous resources, even as Pandey et al. [15], it assumes an initial set of VMs is out there beforehand associate degreed therefore lacks in utilizing the physical property of IaaS clouds.

Finally, Rodriguez and Buyya et al. [25] use PSO algorithm for optimization problem which aims to minimize the whole execution cost.

3. PROBLEM DEFINATION

There are two main stages when planning the execution of a workflow in a cloud environment. The first one is the resource provisioning phase; during this stage, the computing resources that will be used to run the tasks are selected and provisioned. In the second stage, a schedule is generated and each task is mapped onto the best-suited resource. The selection of the resources and mapping of the tasks is done so that different user defined quality of service (QoS) requirements is met. The selection of initial resource pool as it has a significant impact on the performance of a cloud computing environment.

4. EXISTING SYSTEM

4.1 Particle Swarm Optimization-

The rule could be a random optimization technique within which the foremost basic conception is that of particle. To find out the position of a particle.

PARTICLE SWARM OPTIMIZATION ALGORITHM-

1. Set the dimension of particle d.
2. Initialise the population of particle with random po sition and velocity.
3. For each particle, calculate its fitness value
   - 3.1 Compare the particle fitness value with the particle's pbest. If the current value is better than pbest then set pbest to the current value and location.
   - 3.2 Compare the particle fitness value with the particle's gbest. If the current value is better than pbest then set gbest to the current value and location.
   - 3.3 Update the position and velocity of the particle.
4. Repeat from step 3 until the stopping criteria met.

4.2 Schedule generation

The pseudo-code to convert a particle's position into a Schedule. Initially, the set of resources to lease R and the set of task to resource mappings M are empty and the total execution cost TECand time TET are set to zero.

ISSN (ONLINE): 2349-7084
INTERNATIONAL JOURNAL OF COMPUTER ENGINEERING IN RESEARCH TRENDS
VOLUME 2, ISSUE 10, OCTOBER 2015, PP 694-698

ALGORITHM 2
SCHEDULE GENERATION

Input: Set of workflow tasks T
Initial resource pool $\mathcal{R}_{initial}$
An array $pos[|T|]$ representing a particle's position
Output: A Schedule S
1. Initialize schedule components
  1.1. $\mathcal{R} = \emptyset, \mathcal{M} = \emptyset$
  1.2. $TEC = 0, TET = 0$
2. Calculate $ExeTime[|T| \times |R_{initial}|]$
3. Calculate $TransferTime[|T| \times |T|]$
4. **for** $i = 0$ to $i = |T| - 1$
  4.1. $t_i = T[i]$, $r_{pos[i]} = \mathcal{R}_{initial}[pos[i]]$
  4.2. **if** t_i has no parents
    $ST_{t_i} = LET_{r_{pos[i]}}$
  else
    $ST_{t_i} = max\left(max\left\{ET_{t_p} : t_p \in parents(t_i)\right\}, LET_{r_{pos[i]}}\right)$
  end if
  4.3. $exe = exeTime[i][pos[i]]$
  4.4. **for each** child t_c of t_i
    if t_c is mapped to a resource different to $r_{pos[i]}$
      $transfer += TransferTime[i][c]$
    end if
  end for each
  4.5. $PT_{t_i}^{r_{pos[j]}} = exe + transfer$
  4.6. $ET_{t_i} = PT_{t_i}^{r_{pos[j]}} - ST_{t_i}$
  4.7. $m_{t_i}^{r_{pos[i]}} = (t_i, r_{pos[i]}, ST_{t_i}, ET_{t_i})$
  4.8. $M = M \cup \{m_{t_i}^{r_{pos[i]}}\}$
  4.9. **if** $r_{pos[i]} \notin R$
    $LST_{r_{pos[i]}} = max(ST_{t_i}, bootTime)$
    $R = R \cup \{r_{pos[i]}\}$
  end if
  4.10. $LET_{r_{pos[i]}} = PT_{t_i}^{r_{pos[j]}} + LST_{r_{pos[i]}}$
5. Calculate *TEC* according to equation (4)
6. Calculate *TET* according to equation (5)
7. $S = (R, M, TEC, TET)$

5. PROPOSED SYSTEM

5.1 Genetic Algorithm

Genetic Algorithms (GAs) is adaptive heuristic search algorithmic program supported the organic process concepts of action and genetic science. Associate in Nursing of itself} they represent an intelligent exploitation of a random search accustomed solve optimization issues. Though irregular, GAs ar by no suggests that random, instead they exploit historical info to direct the search into the region of higher performance among the search house. The essential techniques of the GAs ar designed to simulate processes in natural systems necessary for evolution, particularly those follow the principles initial set down by Charles Robert Darwin of "survival of the fittest.". Since in nature, competition among people for scanty resources leads to the fittest people dominating over the weaker ones

Genetic algorithms belong to the larger category of organic process algorithms (EA), that generate solutions to optimisation issues victimisation techniques impressed by natural evolution, like inheritance, mutation, selection, and crossover.

GAs simulates the survival of the fittest among people over consecutive generation for resolution a tangle. Every generation consists of a population of character strings that area unit analogous to the body that we have a tendency to see in our deoxyribonucleic acid. Every individual represents a degree during a search area and a potential resolution. The people within the population area unit then created to travel through a method of evolution.

GAs area unit supported AN analogy with the genetic structure and behavior of chromosomes inside a population of people exploitation the subsequent foundations:

- Individuals during a population contend for resources and mates.
- Those people most booming in every 'competition' can turn out additional offspring than those people that perform poorly.
- Genes from `good' people propagate throughout the population in order that 2 smart oldsters can generally turn out offspring that area unit higher than either parent.
- Thus every serial generation can become additional suited to their atmosphere.

Based on action Implementation details-After associate initial population is indiscriminately generated, the algorithmic rule evolves the through 3 operators:

- selection that equates to survival of the fittest;
- crossover that represents pairing between individuals;
- Mutation that introduces random modifications.

6. CONCLUSION

In this paper I presented a combined resource provisioning and scheduling strategy for executing scientific workflows on IaaS clouds. The scenario was modeled as an optimization problem which aims to minimize the overall execution cost while meeting a user defined deadline and was solved using the meta-heuristic optimization algorithm, PSO. The proposed approach incorporates basic IaaS cloud

Principles such as a pay-as-you-go model, heterogeneity, elasticity, and dynamicity of the resources. Furthermore, our solution considers other characteristics typical of IaaS platforms such as performance variation and VM boot time. Also I will use genetic algorithm for optimization strategies

REFERENCES

[1] G. Juve, A. Chervenak, E. Deelman, S. Bharathi, G. Mehta, and K. Vahi, “Characterizing and profiling scientific workflows,” Future Generation Comput. Syst., vol. 29, no. 3, pp. 682–692, 2012.

[2] P. Mell, T. Grance, “The NIST definition of cloud computing— recommendations of the National Institute of Standards and Technology” Special Publication 800-145, NIST, Gaithersburg, 2011.

[3] R. Buyya, J. Broberg, and A. M. Goscinski, Eds., Cloud Computing: Principles and Paradigms, vol. 87, Hoboken, NJ, USA: Wiley, 2010.

[4] J. Kennedy and R. Eberhart, “Particle swarm optimization,” in Proc. 6th IEEE Int. Conf. Neural Netw., 1995, pp. 1942–1948.

[5] Y. Fukuyama and Y. Nakanishi, “A particle swarm optimization for reactive power and voltage control considering voltage stability,” in Proc. 11th IEEE Int. Conf. Intell. Syst. Appl. Power Syst., 1999, pp. 117–121.

[6] C. O. Ourique, E. C. Biscaia Jr., and J. C. Pinto, “The use of particle swarm optimization for dynamical analysis in chemical processes,” Comput. Chem. Eng., vol. 26, no. 12, pp. 1783–1793, 2002.

[7] T. Sousa, A. Silva, and A. Neves, “Particle swarm based data mining algorithms for classification tasks,” Parallel Comput., vol. 30, no. 5, pp. 767–783, 2004.

[8] M. R. Garey and D. S. Johnson, Computer and Intractability: A Guide to the NP-Completeness, vol. 238, New York, NY, USA: Freeman, 1979.

[9] M. Rahman, S. Venugopal, and R. Buyya, “A dynamic critical path algorithm for scheduling scientific workflow applications on global grids,” in Proc. 3rd IEEE Int. Conf. e-Sci. Grid Computing., 2007, pp. 35–42.

[10] Maria Alejandra Rodriguez and Rajkumar Buyya, “Deadline based Resource provisioning and Scheduling Algorithm for scientific workflows on clouds”, IEEE Transaction on cloud computing, vol. 2, no. 2, pp. 222-235, April-June 2014.