Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X
IMPACT FACTOR: 6.017

IJCSMC, Vol. 7, Issue. 12, December 2018, pg.179 – 190

Optimization of Dynamic Provisioning and Scheduling Based on Budget and Deadline Constraints

A.Hemapriya[1], A.Finny Belwin[2], A.Linda Sherin[3], Dr. Antony Selvadoss Thanamani[4]
[1]Research Scholar Department of Computer Science & Bharathiar University, India
[2]Research Scholar Department of Computer Science & Bharathiar University, India
[3]Research Scholar Department of Computer Science & Bharathiar University, India
[4]Professor and Head Department of Computer Science NGM College, Pollachi, India
[1] anushya7373@gmail.com; [2] belwin35@gmail.com; [3] linz15sherin@gmail.com; [4] selvdoss@gmail.com

Abstract— In cloud computing vast scale applications communicated as logical work processes are regularly assembled into gatherings of between related errands. The issue of work process booking undertaking bunches under expense and due date based client requirements parameter in Infrastructure as a Service (Infrastructure-as-a-Service) mists. To built up a few leaving static-based calculations for assignment booking and asset provisioning that depend on exertion structure data and approximations of undertaking runtimes. The past investigation around there overlooks record exchanges between unique work process errands, regularly large affect work process cooperative execution. The proposed dynamic provisioning dynamic booking (DPDS) calculations dependent on static and dynamic planning and asset provisioning. This booking positioned work process bunches under spending plan and due date requirements. The exploratory outcome cloud reenactment utilizing an arrangement of logical work process occupations with a wide scope of spending plan and due date parameters, considering errand granularity, vulnerabilities in undertaking runtime estimations, provisioning deferrals, and cost.

Keywords— Cloud Computing, Scientific Workflows, Dynamic Provisioning, Dynamic Scheduling, Budget and Deadline.

I. INTRODUCTION

Computer technology is an indispensable application for logical examinations. Work processes, the association among researchers and PC frameworks, are an accumulation of computational errands sorted out to achieve a composite task as in atmosphere demonstrating, genome sequencing, seismic examination and oil exploration [3]. Logical work processes incorporate hundreds or thousands of computational assignments which are interconnected after various reliance designs. Work process undertakings routinely require substantial info information documents and additionally play out a remarkable number of guidelines. These variables incite logical work processes to create a high number of blends to disseminate their undertakings on PC assets. As a result, the procedure to choose the ideal dissemination turns into an entangled issue.

These days, diverse PC frameworks have the expected capacities to execute these applications; in any case, cloud computing has the most alluring condition to run logical work processes because of five primary qualities [2]. Initially, cloud computing frameworks have unprecedented measures of figuring power and huge information stockpiling limit. Furthermore, in spite of matrix registering frameworks, each individual as well as foundation can get to cloud assets with no connection. Thirdly, cloud frameworks keep the requirement for their clients to put resources into exorbitant frameworks, for example, supercomputers. Fourthly, in opposition to bunch registering frameworks, cloud clients can scale up/down the quantity of assets. At long last, cloud clients can have quick access to registering assets while supercomputing framework clients are normally required to trust that weeks will approach assets.

The possibility of work process has its underlying foundations in business undertakings as a business procedure displaying apparatus. These business work processes plan to robotize and enhance the procedures of an association, seen as an arranged succession of exercises, and are a mature research area region lead by the Workflow Management Coalition. This idea of work process has reached out to established researchers where logical work processes are utilized to help vast scale, complex logical procedures. They are intended to direct tests and demonstrate logical speculations by overseeing, investigating, mimicking and imagining logical information. Along these lines, despite the fact that both business and logical work processes share a similar fundamental idea, both have explicit prerequisites and need separate thought. A work process is characterized by an arrangement of computational undertakings with conditions between them [1]. In logical applications, usually for the dependence to speak to an information spill out of one undertaking to another; the yield information produced by one assignment turns into the info information for the following one. These applications can be CPU, memory, or I/O escalated (or a blend of these), contingent on the idea of the issue to be illuminated. In a CPU escalated work process, most errands invest the lion's share of their energy performing calculations. Memory-bound work processes are those in which a large portion of the assignments require high physical memory utilization. At long last, I/O escalated work processes are made out of errands that require and create a lot of information and henceforth invest a large portion of their energy performing I/O activities.

Rest of the paper is composed as pursues, area I contain diagram of cloud computing and distinctive sort of cloud organization. Segment II contain audit of leaving cloud planning and power product asset calculations, Section III contain proposes framework and module executions, Section IV contain result and talk, execution examination, Section V closes.

II. RELATED WORK

Cloud computing, a distributed computing worldview, empowers circulation of IT assets over the Internet and pursues the compensation as-you-go charging model. Work process planning is one of the major testing issues in distributed computing. In spite of the fact that, work process planning on conveyed frameworks like lattices and bunches have been broadly contemplated, be that as it may, these arrangements are not material for a cloud domain. The reason is a cloud domain varies from other dispersed condition in two noteworthy courses: on-request asset provisioning and pay-as-you-go evaluating model. In this manner, to accomplish the genuine advantages of work process organization onto cloud assets novel methodologies that can features the points of interest and deliver the difficulties explicit to a cloud situation should be produced. The current powerful savvy due date compelled heuristic calculation for booking a logical work process in an open cloud [6]. The proposed system expects to misuse the focal points offered by distributed computing while at the same time considering the virtual machine (VM) execution changeability and occasion procurement deferral to distinguish a without a moment to spare calendar of a due date compelled logical work process at lower costs.

The Pegasus system, its capacity to be tweaked to suit different planning and copy choice calculations, and its capacity to give parcel level disappointment recovery. The existing booking calculations to assess their execution for booking logical work-streams in Grid conditions. The planning calculations include a hereditary calculation like the one exhibited, the notable HEFT calculation, and a "nearsighted" calculation. The HEFT calculation is an expansion for heterogeneous conditions of the traditional rundown booking calculation. Heave is a basic and computationally economical calculation, which plans work processes by making an arranged rundown of undertakings out of the work process, and mapping the assignments to the assets in the most fitting way. Ideally tackle the undertaking booking issue in branches with various successive assignments by demonstrating the branch as a Markov Decision Process and utilizing the esteem cycle technique. A few works have been proposed to address booking issues dependent on clients' due date limitation [10]. Nimrod-G plans free undertakings for parameter-clear applications to meet clients' deadline.

Cloud Partial Critical Paths (IC-PCP), and a two-stage calculation which is called IaaS Cloud Partial Critical Paths with Deadline Distribution (IC-PCPD2) [10]. Both the calculations have a multinomial time unpredictability which make them appropriate choices for booking substantial work processes. Limit the general execution cost while meeting a client characterized due date and was fathomed utilizing the meta-heuristic advancement calculation, PSO. The dynamic versatility empowers clients to boost and limit the crude assets in light of the business volume, execution necessities and other unique practices. Augment the volume of client organized work processes that can be finished given spending plan and due date imperatives.

The hereditary calculation approach for booking work process applications by either limiting the fiscal expense while meeting clients' due date requirement, or limiting the execution time while meeting clients' spending limitations [12].An ACS calculation for a huge scale work process planning issue in computational networks has been proposed. The booking calculation is intended for work process applications in market driven or economy-driven frameworks under the administration situated design. In the calculation, distinctive QoS parameters are considered, including unwavering quality, time, and cost. Adopt a genuinely static approach in characterizing execution expenses of the undertakings of the DAG. In any case, as demonstrated by concentrates on work process booking, it gives the idea that heuristics performing best in a static domain (e.g., HBMCT) have the most elevated potential to perform best in an all the more precisely displayed Grid condition. The sliding limitation is characterized as a component of the primer arrangement, which is extremely natural for the end-client who does not have to figure ahead of time what the correct imperative esteem ought to be. It alters our booking approach for various classes of criteria situated in a novel scientific classification of planning criteria introduced. The existing asset designation strategies with vigor, makespan and cost as its destinations. This is one of the early works in vigorous and blame tolerant work process planning on Clouds, considering due date and spending requirements. A molecule swarm improvement (PSO) based heuristic to lineup applications to cloud assets that considers both calculation cost and information transmission cost.

III. PROPOSED METHODOLOGY

The propose system main aim dynamic resource management and scheduling the cloud tasks when executing scientific work how groups on clouds. To address an important problem of maximizing the number of completed workflows from an ensemble under both budget and deadline constraints. How much computation can be completed given the limited budget and deadline and timeframe of a research project.

The main contributions of this propose system are:

- The dynamic resource allocation prioritized work-flow ensembles under budget and deadline constraints.
- In this propose system using workflow ware dynamic algorithms for task scheduling and resource provisioning that rely on workflow structure information and estimates of task runtimes.
- To evaluate these algorithms using a simulator based on CloudSim, which models the infrastructure and the application, taking into account uncertainties in task runtime estimates, provisioning delays, and failures.
- The performance of the algorithms on a set of synthetic flow ensembles based on important, real scientific applications, using a broad range of different application scenarios and varying constraint values.

It also reports on new results and discussion of task granularity in the context of the resource billing cycle, which is an important aspect of scheduling and resource provisioning on cloud infrastructures.

A. Resource Model

The asset display like Amazon's Elastic Compute Cloud (EC2), where virtual machine (VM) cases might be provisioned on-request and are charged constantly, with fractional hours being gathered together. In spite of the fact that there might be heterogeneous VM types with various measures of CPU, memory, plate space, and I/O, for this paper center around a numerous VM type since it accept that for most applications there will normally be just a single or two VM types with the best value/execution proportion for the application. The expect that a submitted assignment has restrictive access to a VM case and that there is no appropriation.

B. Cloud Model

The cloud demonstrate dependent on java cloudsim condition. The objective applications are groups of logical work processes that can be demonstrated as DAGs, where the hubs in the chart speak to computational errands, and the edges speak to information or control-stream conditions between the assignments. The manufactured work processes that were produced utilizing verifiable information from genuine applications. The applications originate from a wide assortment of areas including: bioinformatics (Epigenomics, SIPHT: sRNA ID convention utilizing high-throughput innovation), stargazing (Montage), seismic tremor science (Cyber Shake), and material science (LIGO). The engineered work processes were produced utilizing java cloudsim code, with assignment runtimes dependent on dispersions accumulated from running genuine work processes.

C. Dynamic Provisioning Dynamic Scheduling

The dynamic framework is an online calculation that arrangements assets and timetables errands at runtime of cloud. Here two imperative advances are considered: a provisioning strategy, and a booking system [23]. dynamic provisioning system depends on asset usage under the limit. The asset distribution begins with a settled number of assets determined dependent on the accessible time and spending plan, and changes the quantity of assets as per how well they are used by powerful edge.

Algorithm 1: Dynamic provisioning algorithm

Require: c: consumed budget; b: total budget; d: deadline; p: price; t: current time; uh: upper utilization threshold; ul: lower utilization threshold; v_{max}: maximum number of VMs

procedure Provision
$V_R \leftarrow$ set of running VMs
$V_C \leftarrow$ set of VMs completing billing cycle
$V_\emptyset \leftarrow \emptyset$ set of VMs to terminate
$n_T \leftarrow 0.$ number of VMs to terminate
if $b - c < |V_C| * port > d$then
$n_T \leftarrow |V_R| - \lfloor (b-c)/p \rfloor$
$V_T \leftarrow$select n_TVMs to terminate from V_C
Terminate (V_T)
else
$u \leftarrow$current VM utilization
if $u > u_h and |V_R| < v_{max} * N_{VM}$ then
Start (new VM)
else if $u < u_l$then
$V_I \leftarrow$ set of idle VMs
$n_T \leftarrow \lceil |V_I|/2 \rceil$
$V_T \leftarrow$ select n_T VMs to terminate from V_I
Terminate (V_I)
end if
end if
end procedure

Algorithm 2: Priority-based scheduling algorithm for DPDS

procedure Schedule
$P \leftarrow$empty priority queue
$IdleVM_s \leftarrow$ set of idle VMs
for root task t in all workflows do
Insert (t, P)
end for
while deadline not reached do
while $IdleVM_s \neq \emptyset \ andP \neq \emptyset$ do
$v \leftarrow$SelectRandom ($IdleVM$)
$t \leftarrow$Pop(P)
Submit (t, v)
end while

13: Wait for task t to complete on VM_v
14: Update P with ready children of t
15: Insert $(v, IdleVM_s)$
16: end while
17: end procedure

Given a budget plan in dollars b, due date in hours d, and the hourly cost of a VM in dollars p, it is conceivable to compute the quantity of VMs, NVM, to layout with the goal that the whole spending plan is depleted before the deadline:

$$\mathrm{N_{VM}} = \lceil b/(d * p) \rceil \qquad (1)$$

DPDS arrangements N_VM toward the beginning of the troupe execution, at that point it occasionally figures asset usage utilizing the level of inert VMs after some time and modifies the quantity of VMs if the use is above or underneath the given edges. Since it is expected that VMs are charged constantly, DPDS just considers VMs that are moving toward their hourly charging cycle when choosing which VMs to end.

Dynamic work process mindful scheduling

This calculation starts cut down need endeavors just to keep VMs possessed that will end up putting off higher need errands later on, making it more abnormal that higher need work procedures will have the ability to wrap up.

Algorithm 3: Workflow admission algorithm

Require: w: workflow, b: budget, c: current cost

1: procedure Admit (w, b, c)
2: $r_n \leftarrow b \leftarrow c$. Budget remaining for new VMs
3: $r_c \leftarrow$ cost committed to VMs that are running
4: $r_a \leftarrow$ cost to complete workflows previously admitted
5: $r_m \leftarrow 0.1$. Safety margin
6: $r_b \leftarrow r_n + r_c - r_a - r_m$. Budget remaining
7: $c_w \leftarrow$ Estimate Cost(w)
8: if $c_w < r_b$then return TRUE
9: else return FALSE
10: end if
11: end procedure

The dynamic work process mindful calculation expands DPDS by presenting a work process confirmation technique, which is conjured at whatever point propose sees the principal assignment of another work process at the leader of the need line. The confirmation strategy appeared in Algorithm 3 gauges whether there is sufficient spending plan staying to concede the new work process; in the event that there isn't, the work process is rejected and its undertakings are expelled from the line [14].

Algorithm 4: Ensemble planning algorithm for SPSS Require: W: workflow ensemble, b: budget, d: deadline Ensure: Schedule as much of W as possible given b and d

1: procedure Plan Ensemble (W; b; d)
2: $P \leftarrow \emptyset$. Current plan
3: $\leftarrow \emptyset$. Set of admitted DAGs
4: for w in W do
5: P' ←Plan Workflow (w, P, d)
6: if Cost(P') $\leq b$ then
7: $P \leftarrow P'$. Accept new plan
8: $A \leftarrow A + w$. Admit w
9: end if
10: end for
11: return P, A
12: end procedure

To design a work process, the SPSS calculation allots sub-due dates to every individual errand in the work process, and afterward plans each undertaking in order to limit the expense of the undertaking while as yet meeting its doled out sub-due date [12]. In the event that each assignment can be finished by its due date at all costly way, at that point the expense of the whole work process can be limited without surpassing the due date. SPSS allocates sub-due dates to each undertaking dependent on the slack time of the work process, which is characterized as the measure of additional time that a work process can broaden its basic way and still be finished by the troupe due date. For a work process w, the slack time of w is: ST(w)=d-CP(w) where d is the due date and CP(w) is the basic way of w. It expect that CP(w)≤d, generally the work process can't be finished by the due date and should be rejected. For extensive outfits we expect the basic way of any individual work process to be substantially less than the due date.

An undertaking's dimension is the length of the longest way between the assignment and a passage errand of the work process:

$$Level(t) = \begin{cases} 0, & if Pred(t) = \emptyset \\ max_{p \in Pred(t)} Level(p) + 1, & otherwise \end{cases} \quad (2)$$

SPSS disperses the slack time of the stream by level, with the goal that each dimension of the stream gets a bit of the work process' slack time corresponding to the quantity of assignments in the dimension and the aggregate runtime of undertakings in the dimension. The thought is that dimensions containing numerous errands and substantial runtimes ought to be given a bigger part of the slack time with the goal that undertakings in those dimensions might be serialized. Something else, numerous assets should be assigned to run the majority of the undertakings in parallel, which might be costlier.

The slack time of a level l in workflow w is given by:

$$ST(l) = ST(w)\left[\left(\alpha \frac{N(l)}{N(w)}\right) + \left((1-\alpha)\frac{R(l)}{R(w)}\right)\right] \quad (3)$$

where N(w) is the number of tasks in the workflow, N(l) is the number of tasks in level l, R(w) is the total runtime of all tasks in the workflow, R(l) is the total runtime of all tasks in level l, and α is a parameter between 0 and 1 that causes more slack time to be given to levels with more tasks (large α) or more runtime (small α).

Algorithm 5: Workflow planning algorithm for SPSS

Require: w: workflow, P: current plan, d: deadline Ensure: Create plan for w that minimizes cost and meets deadline d

procedure Plan Workflow (w, P, d)
P' ←copy of P
Deadline Distribution (w, d)
for t in w sorted by DL(t) do
v ←VM that minimizes cost and start time of t
if FinishTime (t, v) <DL(t) then
Schedule (t, v)
else
Provision a new VM v
Schedule (t, v)
end if
end for
return P0
end procedure

The deadline of a task t is then:

$$DL(t) = LST(t) + RT(t) + ST\big(Level(t)\big) \qquad (4)$$

where Level(t) is the level of t, RT(t) is the runtime of t, and LST(t) is the latest start time of t determined by:

$$LST(t) = \begin{cases} 0, & if Pred(t) = \emptyset \\ max_{p \in Pred(t)} DL(p), & otherwise \end{cases} \qquad (5)$$

The Plan Workflow procedure first calls Deadline Distribution to assign sub-deadlines to tasks. Then, Plan Workflow schedules tasks onto VMs, allocating new VMs when necessary

IV. RESULTS AND DISCUSSION

The proposed calculations are building up a cloud work process test system dependent on java CloudSim. The recreation utilizing manufactured work processes demonstrated structures and parameters that were taken from genuine applications. The manufactured work streams from five genuine applications: SIPHT, LIGO, Epigenomics, Montage and Cyber Shake. For every application, work streams with 50, 100, 200, 300, 400, 500, 600, 700, 800, 900 and 1000 assignments were made. For each stream measure, 20 diverse workflow occurrences were produced utilizing parameters and assignment runtime conveyances from genuine stream follows. The aggregate gathering of engineered work processes contains 5 applications, 11 diverse workflow sizes, and 20 work process cases, for a sum of 1100 manufactured work processes.

The spending imperatives are determined by recognizing the littlest spending plan required to execute one of the work streams in the group (Min-Budget), and the littlest spending plan required to execute all work processes in the outfit (Max Budget):

$$MinBudget = min_{w \in e} Cost(w) \qquad (6)$$

$$MaxBudget = \sum_{w \in e} Cost(w) \quad (7)$$

This range [MinBudget, MaxBudget] is then divided into equal intervals to determine the budgets to use in each experiment.

$$MinDeadline = min_{w \in e} CriticalPath(w) \qquad (8)$$

$$MaxDeadline = \sum_{w \in e} CriticalPath(w) \qquad (9)$$

This range [MinDeadline, MaxDeadline] is then separated into equivalent interims. This methodology is to include the quantity of work processes the group that every calculation can finish inside the financial plan before the due date, yet this measurement does not represent the need based utility capacity determined by the client. Utilizing the tallying approach, a less productive calculation might have the capacity to finish an extensive number of low-need work streams by executing the littlest work processes first. So as to represent the need, it utilizes an exponential scoring characterized as:

$$Score(e) = \sum_{w \in Completed(w)} 2^{-Priority(w)} \qquad (10)$$

where Completed(e) is the arrangement of work processes in troupe that was finished by the calculation, and Priority(w) is the need of work process w with the end goal that the most noteworthy need work process has Priority(w) = 0, the following most noteworthy work process has Priority(w) = 1, etc. This exponential scoring capacity gives the most noteworthy need work process a score that is higher than all the lower-need work processes consolidated:

$$2^{-p} > \sum_{i=p+1....} 2^{-i} \qquad (11)$$

This scoring is consistent with our definition of the problem, which is to complete as many works flows as possible, according to their priorities, given a set budget and deadline.

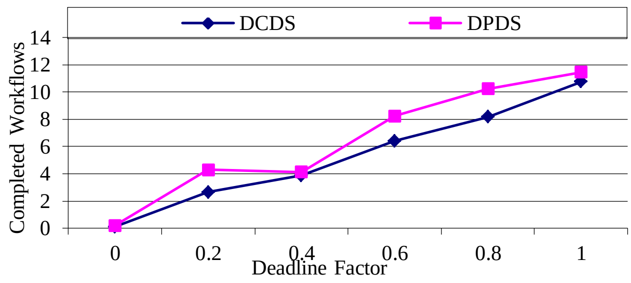

Fig.1.Comparison of completed workflows based on deadline factor existing with proposed system

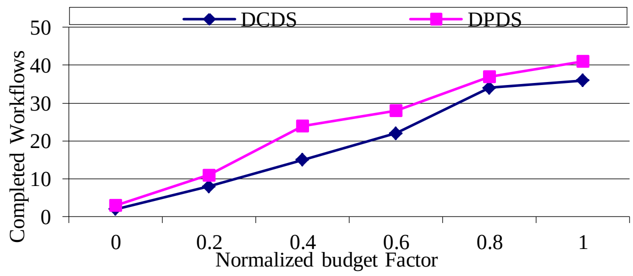

Fig.2.Comparison of completed workflows based on budget factor existing with proposed system

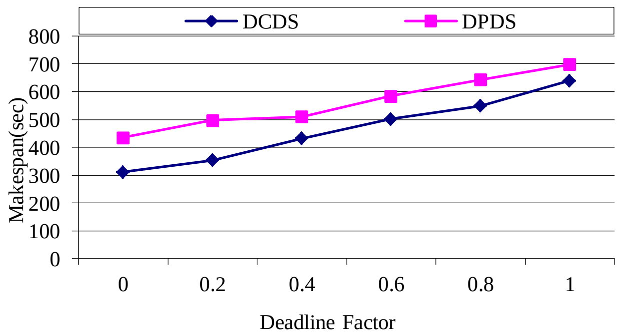

Fig.3.Comparison of completed workflows based on budget factor existing with proposed system

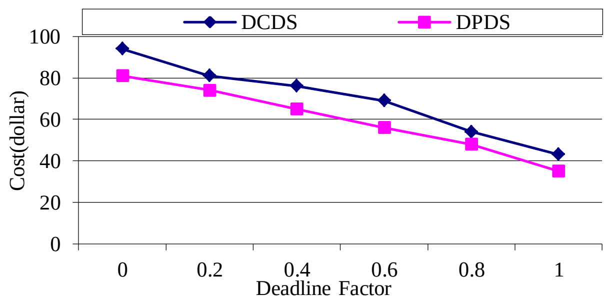

Fig.4.Comparison of computation cost existing with proposed system

V. CONCLUSION

In this paper consider primary issue of dynamic scheduling and resource provisioning for logical work stream on IaaS clouds. The propose framework mostly center around augment the quantity of client organized work processes that can be finished given spending plan and deadline requirements. The expansion of the quantity of positioned work processes finished from the community is additionally novel and requires work streams to be conceded or dismissed dependent on their assessed asset requests. The propose work processes, for example, Cyber-Shake and Montage, nonetheless, the dynamic calculations much of the time create execution that is as great or superior to existing, on the grounds that they are better ready to pack the littler errands onto inactive VMs near the deadline than existing, which conveys due dates to assignments in a way that keeps them from beginning late. This makes the dynamic algorithm substantially more alluring for expansive scale issues.

REFERENCES

[1] E. Deelman, G, Singh, M. H, Su, J. Blythe, Y. Gil, C. Kesselman, D. S. Katz, "Pegasus: A framework for mapping complex scientific workflows onto distributed systems", Sci. Program., vol. 13, no. 3, pp. 219-237, 2005.

[2] M. Wieczorek, R. Prodan, T. Fahringer, "Scheduling of scientific workflows in the ASKALON grid environment", ACM SIGMOD Rec., vol. 34, no. 3, pp. 56-62, 2005.

[3] C. Hoffa, G. Mehta, T. Freeman, E. Deelman, K. Keahey, B. Berriman, J. Good, "On the use of cloud computing for scientific workflows", Proc. IEEE 4th Int. Conf. eScience, pp. 640-645, Dec. 2008.

[4] S. Ostermann, A. Iosup, N. Yigitbasi, R. Prodan, T. Fahringer, D. Epema, "A performance analysis of EC2 cloud computing services for scientific computing", Proc. 1st Int. Conf. Cloud Comput., pp. 115-131, 2010.

[5] G. Juve, E. Deelman, G. B. Berriman, B. P. Berman, P. Maechling, "An evaluation of the cost and performance of scientific workflows on amazon ec2", J. Grid Comput., vol. 10, no. 1, pp. 5-21, 2012.

[6] J. Yu, R. Buyya, C. K. Tham, "Cost-based scheduling of scientific workflow applications on utility grids", Proc. 1st Int. Conf. e-Sci. Grid Comput., pp. 140-147, Jul. 2005.

[7] S. Abrishami, M. Naghibzadeh, D. H. Epema, "Cost-driven scheduling of grid workflows using partial critical paths", IEEE Trans. Parallel Distrib. Syst., vol. 23, no. 8, pp. 1400-1414, Aug. 2012.

[8] E. K. Byun, Y. S. Kee, J. S. Kim, S. Maeng, "Cost optimized provisioning of elastic resources for application workflows", Future Gener. Comput. Syst., vol. 27, no. 8, pp. 1011-1026, 2011.

[9] G. Juve, A. Chervenak, E. Deelman, S. Bharathi, G. Mehta, K. Vahi, "Characterizing and profiling scientific workflows", Future Gener. Comput. Syst., vol. 29, no. 3, pp. 682-692, 2013.

[10] S. Abrishami, M. Naghibzadeh, D. H. Epema, "Deadline-constrained workflow scheduling algorithms for Infrastructure as a service clouds", Future Gener. Comput. Syst., vol. 29, no. 1, pp. 158-169, 2013.

[11] M. A. Rodriguez, R. Buyya, "Deadline based resource provisioning and scheduling algorithm for scientific workflows on clouds", IEEE Trans. Cloud Comput., vol. 2, no. 2, pp. 222-235, Apr.–Jun. 2014.

[12] M. Mao, M. Humphrey, "Auto-scaling to minimize cost and meet application deadlines in cloud workflows", Proc. Int. Conf. High Perform. Comput. Netw. Storage Anal., pp. 49, Nov. 2011.

[13] M. Malawski, G. Juve, E. Deelman, J. Nabrzyski, "Cost-and deadline-constrained provisioning for scientific workflow ensembles in iaas clouds", Proc. Int. Conf. High Perform. Comput. Netw. Storage Anal., pp. 22, Nov. 2012.

[14] J. Yu, R. Buyya, "Scheduling scientific workflow applications with deadline and budget constraints using genetic algorithms", Sci. Program., vol. 14, no. 3, pp. 217-230, 2006.

[15] W. N. Chen, J. Zhang, "An ant colony optimization approach to a grid workflow scheduling problem with various QoS requirements", IEEE Trans. Syst. Man Cybern. C: Appl. Rev., vol. 39, no. 1, pp. 29-43, Jan. 2009.

[16] Y. Yuan, X. Li, Q. Wang, X. Zhu, "Deadline division-based heuristic for cost optimization in workflow scheduling", Inf. Sci., vol. 179, no. 15, pp. 2562-2575, 2009.

[17] R. Sakellariou, H. Zhao, E. Tsiakkouri, M. D. Dikaiakos, "Scheduling workflows with budget constraints", Proc. CoreGRID Integr. Workshop Integr. Res. GRID Comput., pp. 189-202, 2007.

[18] R. Prodan, M. Wieczorek, "Bi-criteria scheduling of scientific grid workflows", IEEE Trans. Autom. Sci. Eng., vol. 7, no. 2, pp. 364-376, Apr. 2010.

[19] D. Poola, S. K. Garg, R. Buyya, Y. Yang, K. Ramamohanarao, "Robust scheduling of scientific workflows with deadline and budget constraints in clouds", Proc. IEEE 28th Int. Conf. Adv. Inf. Netw. Appl., pp. 858-865, May 2014.

[20] S. Pandey, L. Wu, S. M. Guru, R. Buyya, "A particle swarm optimization-based heuristic for scheduling workflow applications in cloud computing environments", Proc. 24th IEEE Int. Conf. Adv. Inf. Netw. Appl., pp. 400-407, Apr. 2010.

[21] Thilagavathi, D. and A. S. Thanamani, 2012, A Survey on Dynamic Job Scheduling in Grid Environment Based on Heuristic Algorithms, International Journal of Computer Trends and Technology Vol. 3, Issue 4, pp. 531-536.

[22] D. Thilagavathi , Dr. Antony Selvadoss Thanamani, An impression on performance metrics for scheduling problem in grid computing environment, INTERNATIONAL JOURNAL OF RESEARCH IN COMPUTER APPLICATIONS AND ROBOTICS ISSN 2320-7345.

[23] D. Thilagavathi , Dr. Antony Selvadoss Thanamani ,A Survey on Dynamic Job Scheduling in Grid Environment Based on Heuristic Algorithms,International Journal of Computer Trends and Technology (IJCTT).

[24] R.Sunitha, Dr. Antony Selvadoss Thanamani,"Load sharing for cloud procesors using RPF and Genetic algorithm concepts",International journal of Engineering research and Technology(IJERT)ISSN:2278-0181.

[25] N.Sumathi, Dr. Antony Selvadoss Thanamani, "Evaluation of Energy Efficient Reactive Routing Protocols in QoS Enabled Routing for MANETS", International Journal of Computer Applications (0975 – 8887) Volume 14– No.2, January 2011.

[26] D. Thilagavathi Dr. Antony Selvadoss Thanamani, "A Survey on Dynamic Job Scheduling in Grid Environment Based on Heuristic Algorithms', International Journal of Computer Trends and Technology-volume3Issue4- 2012.

[27] N. Sumathi, Antony Selvadoss Thanamani, "ENERGY EFFICIENT BANDWIDTH CONSTRAINED QOS ENABLED MULTIPATH ROUTING FOR MANETs", International Journal of Computing, 2011, Vol. 10, Issue 3, 270-277.

[28] Lakshmi, Dr. Antony Selvadoss Thanamani, "DYNAMIC DATA REPLICATION USING ENHANCED MODIFIED BANDWIDTH HIERARCHY REPLICATION", International Journal of Engineering Technology and Computer Research (IJETCR) Available Online at www.ijetcr.org Volume 5; Issue 4; July-August: 2017; Page No. 43-49 Journal Approved by UGC.

[29] N.Sumathi, Dr. Antony Selvadoss Thanamani, "Pipelined backoff scheme for bandwidth measurement in QoS enabled routing towards scalability for MANets", proceedings of the 1st Amrita ACM-W Celebration on women in india.

[30] K.Sashi[a]Antony SelvadossThanamani[b], "Dynamic replication in a data grid using a Modified BHR Region Based Algorithm", Future Generation Computer Systems Volume 27, Issue 2, February 2011, Pages 202-210.

[31] R.NANDHAKUMAR1 AND ANTONY SELVADOSS THANAMANI, "A Survey on E-Health Care for Diabetes Using Cloud Framework", International Journal of Advanced Research Trends in Engineering and Technology (IJARTET) Vol. 4, Issue 10, October 2017.

[32] N.Raveendran1 Dr. Antony Selvadoss Thanamani, "Autonomic Cloud Services to Enhance Secure Data Sevices for Digital Library", American Research Journal of Computer Science and Information Technology (ARJCSIT) Volume 1, 10 pages.

