IJCSE International Journal of Computer Sciences and Engineering Open Access
Review Paper Volume-5, Issue-6 E-ISSN: 2347-2693

Workflow Scheduling Mechanism Using PCSO n Cloud: Case Study

Prachi Chaturvedi[1*], Sanjiv Sharma[2]

[1*] Dept. of CSE/IT, Madhav Institute of Technology and Science, Gwalior, India
[2] Dept. of CSE/IT, Madhav Institute of Technology and Science, Gwalior, India

**Corresponding Author: pchaturvedi1118@gmail.com, Mob.: +91-9406983296*

Available online at: www.ijcseonline.org

Received: 12/May/2017, Revised: 20/May/2017, Accepted: 17/Jun/2017, Published: 30/Jun/2017

***Abstract*—** Cloud Computing has emerged as a service model that enables on-demand network get right of entry to a massive number of available virtualized resources and applications with a minimal management attempt and a minor rate. The unfold of Cloud Computing technology allowed handling complicated applications together with Scientific Workflows, which consists of a set of extensive computational and data manipulation operations. Cloud Computing enables such Workflows to dynamically provision compute and storage assets necessary for the execution of its responsibilities way to the pliancy asset of those assets. However, the dynamic nature of the Cloud incurs new challenges, as a few allocated assets may be overloaded or out of get entry to all through the execution of the Workflow. Moreover, for data extensive responsibilities, the allocation strategy have to keep in mind the facts placement constraints on the grounds that facts transmission time can growth extensively in this example which implicates the growth of the general of completion time and value of the Workflow. Likewise, for in depth computational responsibilities, the allocation strategy must consider the form of the allocated digital machines, greater specifically its CPU, reminiscence and network capacities. Yet, an essential venture is the way to correctly schedule the Workflow obligations on Cloud resources to optimize its ordinary best of provider. In this paper, we endorse a QoS aware algorithm for Scientific Workflows scheduling that objectives to enhance the overall quality of service (QoS) with the aid of considering the metrics of execution time, data transmission time, price, sources availability and facts placement constraints. We prolonged the Parallel Cat Swarm Optimization (PCSO) algorithm to put in force our proposed method. We tested our algorithm inside pattern Workflows of various scales and we compared the consequences to the ones given by the same old PSO, the CSO and the PCSO algorithms. The consequences display that our proposed algorithm improves the general satisfactory of provider of the tested Workflows.

Keywords— Cloud Computing; Workflow; IaaS; virtual machine; storage; quality of service; scheduling algorithm; Parallel Cat Swarm Optimization...

I. INTRODUCTION

Cloud Computing is a carrier version that permits on-demand Community access to a huge range of available virtualized resources and packages with minimal management attempt and a minor charge. This version can be prepared into 3 layers of services, particularly, Software as a Service (or SaaS) which englobes software packages, Platform As A Service (or PaaS) enclosing platform packages and growing equipment, and Infrastructure As A Service (or IaaS) together with hardware assets including CPU ability, garage and network.

The Cloud computing surroundings gives a couple of advantages for hosting and executing complex applications which include Scientific Workflows. Scientific Workflows are designed particularly to compose and execute a sequence of in depth computational and data manipulation steps [1]. They are often being used to model complex phenomena, to analyse instrumental information, to tie collectively information from disbursed sources, and to pursue other clinical endeavours [2]. They may be finished the usage of IaaS services on the Cloud as virtual machines. Cloud Computing facilitates Workflow programs to dynamically provision compute and garage resources essential for the execution of its responsibilities thanks to the elasticity asset of those sources. It permits, also, Workflows to limit the execution value and meet defined deadlines by allocating resources on-demand the use of the "pay in line with-use" pricing model. However, the dynamic nature of the Cloud Computing surroundings can incur new challenges as a few allocated assets can be overloaded or out of get right of entry to during the execution of the Workflow which might also motive the failure of the general method of execution. For this reason, the allocation method needs to take into account the supply of sources and insure exceptions handling. Moreover, for facts intensive duties (along with a huge quantity of records to manipulate), the allocation strategy must recollect the statistics placement constraints imposed with the aid of the Cloud customers [3] and the constrained garage capacity of computing resources which won't in shape with the amount of records to be dealt with.

In those instances, separate Cloud storage assets like Amazon S3 need to be allocated to store the enter statistics or generated information from the Workflow's responsibilities. Cloud garage is extra notably available than many on-premises storage deployments, with plenty less complexity and a decrease fee. Also, Cloud vendors have made to be had special kinds of virtual machines to fulfil the unceasing users' requirements for more computing capability and reminiscence. Therefore, the allocation approach of Scientific Workflows have to bear in mind the sort of digital machines (its CPU, reminiscence and community capacities) to in shape with the needs of its extensive computational duties. Furthermore, as the allocation strategy focuses typically on sources costs, the selected resources can be geographically very dispersed which can also growth significantly the facts switch time among dependents obligations of the Workflow after which its average execution time. And given that records transfer from one network to some other isn't free in maximum instances; this could further increase the general fee of the Workflow' execution.

To sum up, an essential task for executing Scientific Workflows at the Cloud is the way to efficiently allocate Cloud assets for you to optimize its first-rate of provider objectives.

Therefore, to ensure the performance of such Workflows, powerful scheduling techniques ought to be hired. It consists in calculating all feasible mission-resource mappings and choosing the exceptional ones in keeping with the fine of service objectives [4].

In the Cloud Computing environment, the overall objective of venture scheduling approach is to assure the service-level agreements (SLAs) of allotted sources to make value-efficient selections for workload placement. In our context, we recognition at the challenges cited above. According to our know-how, an effective scheduling approach should bear in mind the type and residences of computing assets (CPU/memory/network capacities), their execution time/price, their availability fee and the records transmission time/value among dependents responsibilities of the Workflow. It has to additionally insure the fine of allocated garage resources (the provision, capacity and fee of facts garage) wanted for the execution of the Workflow.

Since mission scheduling hassle on Cloud Computing environments is NP-complete, numerous heuristics had been proposed to remedy it such as Genetic Algorithms, Particle Swarm Optimization [5][6], Ant Colony Optimization [7], and Cat Swarm Optimization [8] algorithms. However, the trouble continues to be hard because of the extraordinarily dynamic surroundings of the Cloud Computing and to the variability and confliction of the user's necessities.

In this paper, we feature on a pleasant of carrier conscious Workflow scheduling method within the Cloud Computing environments. The intention of our proposed approach is to maximise the general great of carrier of the unique types of allocated assets (computing and garage). We formulated the trouble as a mono-objective optimization trouble and we prolonged the Parallel Cat Swarm Optimization set of rules to get higher outcomes than studied works. Our important contributions on this paper are as follows:

• We do not forget both using separate garage sources for the storage of the Workflow records at some stage in its execution and the use of various sorts of computing assets to satisfy the high need of CPU, reminiscence and network capacities for computational in depth duties.

• We recommend a brand new components for the Workflow scheduling problem within the Cloud as a mono-objective optimization hassle that takes into consideration both the computing and garage resources. We don't forget that each one the fine metrics have the same weight of significance.

• We extended the Parallel Cat Swarm Optimization set of rules to clear up the formulated scheduling problem, evaluate and examine it with the PSO, PCSO and CSO algorithms.

II. Related Work

Being an NP-entire hassle, scheduling Workflows duties has usually attracted the attention of researchers. Since the advent of Cloud Computing, the problem of challenge scheduling became greater difficult because of the dynamic nature of this surroundings. In this paper, we recognition at the high-quality of provider pushed scheduling solutions. Several approaches had been proposed in this trouble and can be categorized according to the used algorithm (GA, PSO, ACO, and CSO) to be unique within the following.

Some proposed works used genetic algorithms (GA) to remedy the assignment scheduling trouble in Workflows. For instance

In [9], the authors recommend a hybrid heuristic for records intensive workflow scheduling in hybrid Cloud Computing environment that minimizes the value of execution, whilst satisfying the user's finances, closing date, and statistics placement constraints. The proposed technique adopts a multi-degree scheduling approach the use of the GA and the Dynamic Critical Path heuristic with a view to satisfy the consumer's constraints. Different from this technique, we awareness at the scheduling of intensive medical workflows in natural Cloud environments.

In [5], the authors propose a brand new PSO-based totally set of rules to clear up the scheduling problem of Workflows in Cloud Computing environments. The goal of the scheduling hassle is to optimize the overall execution time and value of the Workflow. The results of the proposed algorithm show its capability to reduce the general value of the Workflow and maintain an even distribution of work packages a few of the allocated sources. An extension of the discrete PSO algorithm was additionally proposed

In [4] to solve the identical problem. The proposed algorithm ambitions to optimize the Workflow's execution cost and records transmission time between offerings, it proposes new formulation for calculating the position and pace of the swarm debris. The outcomes show a progress within the fitness fee of the global solution in comparison to the usual version of PSO. Another approach based on PSO, described in [6], gives a deliver and scheduling solution for clinical Workflows that consists of IaaS Cloud services. The purpose of scheduling is to optimize the general cost of execution of the Workflow and to meet the deadlines constraints. Another technique cited.

In [9] proposes a revised discrete particle swarm optimization (RDPSO) set of rules to time table Workflow packages at the cloud. RDPSO takes into account the computation price and the entire execution time by using satisfying a cut-off date constraint. In [7], the authors use the Ant Colony Optimization set of rules (ACO) to resolve the hassle of offerings scheduling inside the Mobile Cloud Computing environment. The proposed approach considers handiest impartial duties. A current proposed approach used the Cat Swarm Optimization algorithm (CSO). This algorithm is based at the examine of the movement of cats to remedy the Workflow scheduling problem in Cloud Computing environments.

The proposed approach, as described in [8], aims to solve the multi objective Workflows scheduling problem taking into account the completion time, cost and the time of CPU inactivity.

In conclusion, the proposed techniques lack to recollect the availability and reliability metrics of the allocated assets although they can tell about the resource's performance. Also, not one of the above-cited techniques do remember the capacities of the VM instances or the placement constraints of the Workflow's records. They version the Cloud assets only by way of value/time of workload execution and think that the garage ability of computing assets is enough for the garage of facts wished for the Workflow execution which does now not efficiently represent the huge preference of presented assets by way of Cloud providers.

In this paper, we endorse a QoS-conscious scheduling set of rules for Workflow programs that takes into consideration the following quality metrics: the provision of assets, the fee and time

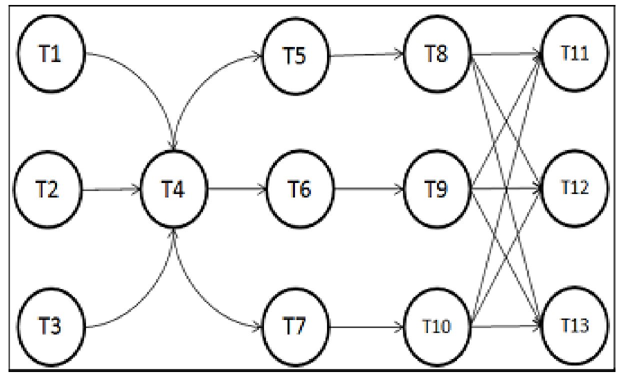

Figure 1. DAG representation for an example of Workflows

The Workflow execution and the information transfer time among dependent tasks. Our proposed method considers additionally the placement constraints of the garage sources based on their availability, potential and price.

III. SCHEDULING PROBLEM FORMULATION

In this segment, we formulate the Workflow tasks scheduling Problem based totally at the first-class of service. We begin by means of modelling Workflows, then we describe a way to specific and compare its normal great of carrier (QoS) and we quit with the formulation of the QoS-aware scheduling problem of Workflow duties.

A. Modelling Workflows

A Workflow utility can be modelled as a group of obligations according to a mixture of manipulate drift and statistics float graphs. The manage flow graph may be represented the usage of UML interest diagrams. The facts drift graph may be modelled thru DAG graphs (Directed Acyclic Graph). In our approach, we recollect modelling the Workflow's facts waft the use of DAG graphs which allow us to determine concurrently the series of execution of the duties and the information flow. A DAG graph G is defined with the aid of the pair G = (V, E), in which V = T1, ..., Tn is the vector of the Workflow's tasks, with "n" the overall quantity of duties. "E" represents all useful hyperlinks between obligations, more particularly, the information dependencies between them, together with (Ti, Tj) ∈ E if the output records of the undertaking "i" are required for the execution of the assignment "j". Is referred to as the source venture and is the vacation spot challenge. Figure 1 indicates the DAG illustration of a pattern Workflow extracted from duties T1, T2, T3 are the "source obligations" of the Workflow. They may be accomplished in parallel (within the identical time). The responsibilities T4, T5 and T8 are completed in sequence. The obligations T11, T12, and T13 are the "exit duties" of the Workflow.

B. Modeling Cloud assets

To execute the duties of the Workflow, we do away with a set of computing assets (digital machines or VM) VM= V M1, , V Mm from the Cloud Computing surroundings. The cloud providers provide specific forms of VM times, suitable for different styles of obligations (Input/output obligations, computing and in depth computing responsibilities, memory tasks) like Amazon EC2 instances kinds. We outline a V Mi by using the pair Ci, Qi in which Ci is a d-dimensional resource ability vector Ci = Ci1, ..., Cid of the 'd' resource varieties of V Mi denoted Ri1, ..., Rid. Cid is the total capability of resource Rid of V Mi. Qi is the vector of first-class of carrier metrics used to evaluate the performances of V Mi. We bear in mind, also, dedicated garage web sites (gadgets) needed especially in the case of information intensive Workflows.

These storage sites can be shared between the Workflow's VMs or each garage website may be associated to a selected VM. We observe by Ds = Ds1, ..., Ds k the vector of "okay" garage web sites used to store the needed data for the execution of the Workflow with "k" identical or less than "m" (the variety of VMs). For each garage website online, we take away a vector of "r" Cloud garage assets like Amazon S3 which are able to store its data with extraordinary performances. Like digital machines, we describe a storage useful resource by way of the pair Ci, Qi wherein Ci is a vector of its resources capacities Ci = Ci1, ..., Cid and Qi is the vector of pleasant of service metrics.

C. Quality of service of a Workflow

In the Cloud Computing surroundings, the best of provider (QoS) of a single IaaS provider includes 3 components (Execution Quality Execution, Communication Quality Communication, and Storage Quality Storage) [10].

It can be evaluated by the sum of the values of these three quality components as indicated in the equation (1). $Q_{GS} = Q_{Execution} + Q_{Communication} + Q_{Storage}$ (1) $Q_{Execution}$ refers to the quality metrics used to assess the performance of a specific virtual machine for executing a workload.

The performance of virtual machines depends on their resources such as CPU, RAM, SDD memory It can be evaluated by the following metrics: execution time, execution cost, availability and reputation... We consider, in this paper the following metrics for the selection and evaluation of virtual machines performance: its CPU/memory/network capacities, its execution cost (for a unit of time (1hour)), the execution time for a single workload and its availability rate.

Communication quality refers back to the QoS for shifting information from one provider to some other the usage of a designated community. Data transfers are determined with the aid of the supply offerings and the destination services [10]. Therefore, communication fine values are calculated handiest on vacation spot offerings and this is performed by adopting the metric of statistics transfer time. Storage excellent refers to the QoS for storing certain amount of information for a certain time the use of particular database carrier [10]. The information to be stored can be the enter records of the provider or produced statistics for the duration of its execution.

In our context, the storage high-quality is largely assessed for storage services. Data is stored at a price described with the aid of statistics unit.

The overall performance of a Cloud garage carrier can be evaluated by the point needed to transmit information from a provider to every other one or to the consumer. Therefore, we remember, on this scope, the metrics of statistics transmission time, storage fee and provider availability to assess the best of service of the Cloud storage services.

To check the global exceptional of the provider of the Workflow for the duration of its execution, we use aggregation features of various considered first-rate metrics as shown within the desk I. "N" is the general quantity of duties of the Workflow.

Ti, Ci, Ai, DTi,i+1 and d i,i+1, are respectively the execution time of a task "i" on a unique VM, its cost of execution, its availability charge, the records drift transmitted between consecutive VMs "i" and "j" and

TABLE 1. Aggregation Functions for Quality Metrics

Quality metric	Execution time	Execution cost	Availability	Data Transfer time
Aggregation function (sequential services)	$\sum_{i=1}^{n} Tps_i$	$\sum_{i=1}^{n} C_i$	$\prod_{i=1}^{n} A_i$	$\sum_{i=1}^{n} \frac{DT_{i,i+1}}{d_{i,i+1}}$
Aggregation function (parallel services)	$\max_{1\leq i\leq n} Tps_i$	$\sum_{i=1}^{n} C_i$	$\prod_{i=1}^{n} A_i$	$\max_{1\leq i\leq n} \frac{DT_{i,i+1}}{d_{i,i+1}}$

TABLE 2. Aggregation Functions for Storage Quality

Quality metric	Storage cost	Availability
Aggregation function	$\sum_{j=1}^{k} c_j * D_j$	$\prod_{j=1}^{k} A_j$

The throughput of community bandwidth between VMs ”i” and ”j” of the Workflow. We assume that the values of the distinct best metrics are furnished as enter to our problem. In the case of parallel responsibilities of the Workflow, the operators of the aggregation characteristic for the execution time and information switch time metrics are replaced via the maximum operator [11].

For the first-class metrics of the Cloud garage resources, the aggregation features for the garage price and availability are defined inside the desk II. For the metric of facts transmission time between exceptional storage assets or among storage sources and the Workflow' VMs, we practice the identical aggregation features as defined in the table II with thinking about that the garage resources having a couple of facts input points like parallel services. Dj is the most information to be stored in the storage aid Dsj related to V Ml with

$$D_j = {}_{l-1}$$
$$_{i=1} DT_{i,l}$$

where $DT_{i,l} = 0$ if the tasks ”i” and ”l” are related in the Workflow. Once the quality metrics are assessed, we use the Simple Additive Weighting (SAW) normalization method described in [11] to determine the value of the overall quality of service of the Workflow.

D. Scheduling trouble formulation

The normal aim of our approach is to gain efficient schedule of the Workflow's duties primarily based on its global quality of service.

To reap the very best realizable value of the global fine of service of the allotted IaaS services for the execution of the Workflow.

More precisely, we have to maximize the general availability and limit the overall execution time, execution and garage price and facts transfer time of the Workflow's allocated assets.

Since, the nearby objectives of the considered great metrics are conflicting and our goal is to offer a worldwide view of the high-quality of service of the Workflow, we continue by transforming the problem formula from multi-goal into mono-objective one via considering that all the nice metrics have the equal weight.

As the execution exceptional and the communication best are associated with computing sources and the storage excellent depends at the storage sources.We can cut up the formulation of the scheduling trouble into separate ones.

The first one (described by means of equation (2)) is composed to outline the mission computing useful resource mappings that optimize the global first-class of provider of the Workflow and the second (described through equation (3)) is composed to define the mission-storage resource mappings that optimize the worldwide first-rate of storage of the Workflow.

The end result of the primary hassle (the chosen computing resources to execute the Workflow's tasks) is used to clear up the second as we take into account the records transmission time among garage assets and VMs in the assessment of the first-rate of garage of the Workflow.

Maximizing the overall great of carrier QGS of a Workflow may be illustrated via the first equation of (2).

Xi, j is a binary variable that is equal to one if a venture “i” is assigned to a VM ”j” and zero in any other case. We assume that every venture may be related with most effective one VM as indicated by way of the second equation of (2).

QSt represents exceptional metrics used for the selection of garage resources and Qi,j represents great metrics used for the choice of VMs.

$$\begin{cases} \max Q^{GS} = \max \sum_{i=1}^{n} \sum_{j=1}^{m} x_{i,j} Q_{i,j} \\ \forall i = 1..n, \sum_{j=1}^{m} x_{i,j} = 1 \\ \forall i = 1..n, \forall j = 1..m, x_{i,j} \in [0,1] \end{cases} \quad (2)$$

$$\begin{cases} \max Q^{St} = \max \sum_{i=1}^{r} \sum_{j=1}^{k} x_{i,j} Q^{St}_{i,j} \\ \forall i = 1..k, \sum_{j=1}^{r} x_{i,j} = 1 \\ \forall i = 1..k, \forall j = 1..r, x_{i,j} \in [0,1] \end{cases} \quad (3)$$

IV. PARALLEL CAT S WARM OPTIMIZATION ALGORITHM

The “Swarm Intelligence” idea is the take a look at of collective behaviour of decentralized, self-organized systems inclusive of natural or artificial behaviour of swarm particles or ants or bees, so one can exploit it in the field of artificial intelligence for fixing a few NP-complete troubles. Cat Swarm Optimization (CSO) is an algorithm proposed on

2007 through the authors of [12] that may be labelled as a ”swarm intelligence” algorithm.

CSO research the conduct of cats that's characterized by means of fundamental factors: make the most of the ruin time being in non-stop alert or ”Seeking mode” and walking at most velocity to hunt its prey while localized, referred to as “Tracing Mode”. According to [12], the experimental results monitor that CSO is superior to PSO (Particle Swarm Optimization) for the reason that CSO avoids the prolix limits, i.e. The maximum velocities in all iterations, and it could discover the worldwide great solution tons faster than PSO-kind algorithms.

The first step inside the CSO set of rules is to initialize the populace of cats to apply (the number of cats and their characteristics). Each cat has its personal M dimensional function and a velocity for each dimension, a fitness value that represents the lodging of the cat to the health feature and a Boolean price indicating whether the cat is within the “Seeking” or inside the “Tracing” mode. In the second step, cats are organized randomly into businesses, one group of cats is in “Seeking mode” and the alternative in “Tracing mode”. Cats are evaluated according to the fitness characteristic and the cat with the exceptional role (i.e. The pleasant health fee) is retained.

If the cat is in “Seeking mode”, then the system of “Seeking mode” is applied, otherwise the technique of “Tracing mode” is carried out. The details of the “Seeking Mode” and the “Tracing mode” can be discovered in [12]. If the prevent condition has now not been reached yet, then the set of rules restarts from the second one step. The very last answer is the quality role of one of the cats. The set of rules continues the solution until it reaches the end of iterations.

Four essential parameters are to be defined and carried out in the CSO algorithm, particularly:

• SMP (Seeking Memory Pool): the hunt memory pool that defines the dimensions of the looking for memory of each cat.

• SRD (Seeking Range of the chosen Dimension): the mutative ratio for the chosen dimensions.

• CDC (Counts of Dimension to Change): the wide variety of factors of the dimension to be modified.

• SPC (Self Position Consideration): a Boolean number indicating whether or not the contemporary role is to be taken into consideration as a candidate role to be moved on or no longer. To exploit the blessings of parallelization idea, the authors of [13] proposed a new edition of the CSO algorithm called Parallel Cat Swarm Optimization (PCSO), using the concept of department of the population into sub-populations.

The idea is to maintain the independence of people in “Seeking mode” due to the fact that they do now not proportion facts with each other and divide them in “Tracing Mode” in two or greater businesses of folks that share facts approximately the excellent overall answer. The “Tracing Mode” is carried out independently for each group G of cats.

The change of facts is executed consistent with the fee of the parameter ‘ECH’ which suggests the wide variety of iterations to carry out before synchronizing information between the exceptional agencies of cats. The details of the data trade technique may be discovered in [13].

V. OUR PROPOSED EPCSO ALGORITHM

In the context of Workflows scheduling in the Cloud, we propose a QoS-conscious scheduling set of rules, known as EPCSO (Extended Parallel Cat Swarm Optimization) that is an extension of the PCSO algorithm. The proposed extension, described in the first sub-section of this segment, permits improving the performances of the PCSO set of rules. The info of the proposed algorithm is defined within the 2nd sub-phase.

A. Using a inertia variable weight

PCSO as other heuristics (PSO, ACO, GA) may additionally go through from premature convergence and stagnation problems [14]. In the same old model of the algorithm, a further situation at the evolution of the velocity of cats needs to be described as the speed interval.

Any cost found outdoor the defined interval may be rounded to a price within this range. Other strategies were additionally proposed to stand the hassle of premature convergence for other heuristic algorithms. For Instance, for the PSO set of rules, we note the use of the idea of inertia weight [15]. The inertia weight has capabilities which can be reminiscent as the temperature setting in the simulated annealing.

In PSO, a big inertia weight facilitates a worldwide search while noting that a small inertia Weight facilitates nearby search.

By linearly lowering the inertia weight from a notably large cost to a small price thru the execution of the algorithm, this latter tends to have a worldwide research capability at the start of the execution and local research ability near the give up of the algorithm.

As a part of our work, we advise a variable inertia weight function adapted to our problem the usage of the search status of the PCSO algorithm to improve its search capability for a most excellent answer.

The proposed inertia weight function "w" is introduced inside the pace assessment equation (4) this is changed through the equation (6).

Vk,d(t) = wvk, d(t− 1)+ r1c1[xlbest, d(t− 1) − xk,d(t− 1)] (6)

The proposed inertia weight feature is inspired form the work of [16] in which the authors endorse a new equation for the evaluation of the rate of particles in PSO set of rules.

We changed the parameters of the proposed equation to conform it to our hassle. Equation (7) illustrates the proposed inertia Weight characteristic.

W = w0−wacc (f(fx(lbest,dxlbest,
d(t (−t))1)) −wagr Avgf(x(lbest,df(xl,d((tt))))) (7)

w0 denotes the preliminary weight set to 1, wacc denotes the acceleration weight measuring the trade inside the pleasant fitness cost of the cats at the new release 't' in comparison to their fine health value on the preceding new release. Wagr denotes the aggregation weight measuring the gap of the satisfactory health cost of the cats at the iteration 't' to the average health values Avg(f(xl,d(t))) of all the cats at the modern generation. The values of the 3 weights of the equation (7) are covered in the c language [0, 1].

As is set to at least one, the experimental consequences show that high-quality values for the weights wacc and wagr are respectively (0.5) and (0.3).

B. Steps of the proposed algorithm

In this phase, we detail the stairs of the proposed algorithm EPCSO.

The advantages of the proposed set of rules is that it improves, on one hand, the search system for the exceptional answer via decomposing the preliminary populace of cats into numerous corporations (through using the parallel technique of PCSO set of rules), and the best of the first-class solution of the hassle, on the other hand, with the aid of refining the quest process in keeping with the best fitness value discovered in previous iterations the usage of the proposed extension of the PCSO algorithm.

1) Mapping between PCSO set of rules and the problem formulation:

To clear up the scheduling problem described by way of the optimization hassle (in equation (2)), the desk III illustrates the mapping among our hassle method and the PCSO algorithm.

TABLE 3. The Mapping Between our Problem Formulation and the PCSO Algorithm

PCSO Algorithm	**Problem formulation**
Dimension of a cat	<$Task_i$,$Resource_j$>
Cat (a set of the cat's dimensions)	Scheduling solution (a set of <T aski ,Resourcej > for i from 1 to n)
Fitness function	Objective function of optimization problem (3)
PCSO solution (cat with the best fitness value)	Workflow scheduling solution

2) EPCSO set of rules:

The following algorithm describes he proposed EPCSO algorithm. We begin with the aid of creating a preliminary population of cats using create Cats characteristic.

Each cat has as dimensions because the quantity of responsibilities in the Workflow. We initialize the cats' parameters, the use of the initialize Cats Parameters characteristic via assigning randomly to each venture a VM and every storage web page a storage aid. We break up randomly the created cats into a predefined quantity of groups using cut up Cats into Groups.

The feature set racing Mode Cats alternatives randomly a fixed of cats in line with MR parameter and set them to tracing mode. Then, for each cat, we compare its fitness function and apply Tracing Mode Procedure or Seeking Mode Procedure strategies according to its mode.

Algorithm 1: EPCSO Algorithm

```
Input : Var catDimensionsNumber;   > the number of tasks of the workflow

Var setOfAvailableResources;   > number of VMs
Var cats;   > represent scheduling solutions (tasks to resources mapping)
Var initialP opulationSize;   > general parameter of PCSO
Var numberOfGroups;   > general parameter of PCSO
Var MaxIterations;   >general parameter of PCSO
Var ECH;   > number of iterations before information exchange
Output: Var BestF itnessV alue;  >optimal quality of service of the workflow
begin
  cats = createCats(initialP opulationSize);
    >create initial population of cats
    initializeCatsP arameters (cats, catDimensionsNumber,
    setOfAvailableResources);
     setAllCatsT oSeekingMode(cats);
    > set all cats to seeking mode
    splitCatsIntoGroups(cats, numberOfGroups);
    for all the cat ∈ cats do
      setT racingModeCats(cats, MR, output :
      tracingCats[]);

   for i := 1toMaxIterations do
     for all the cat ∈ cats do
         varfitnessV alue = EvaluateF itnessV alue(cat);
         if fitnessV alue > BestF itnessV alue then
           BestF itnessV alue = fitnessV alue;

         if cat.Mode == SeekingMode then
           SeekingModeP rocedure(cats);

         else
           TracingModeP rocedure(cats);
           randomlyP ickT racingModeCats
           (cats, MR, tracingCats[]);
         if i == ECH then
           performInformationExchange(cats);
           >synchronize
```

3) The Seeking Mode: The searching for mode lets in exploring all the viable solutions (all combinations of resources to duties assignments) through developing copies of the cats (the use of Make Position Copies feature) and mutating a part of the cat's dimensions (through Modify CDC Value feature) and then calculating and updating the solution to transport to (the use of circulate To Next Position).

***Procedure** Seeking Mode Procedure*

```
Input : Cats
Output: Cats
begin
   NumberOfCopies = SMP ;
   PreviousFitnessV alue = null ;
   if SP C == true then
     NumberOfCopies = SMP - 1 ;

   MakePositionCopies(cat, NumberOfCopies,
   PositionsCopies[]) ;

   forall the position ∈ PositionsCopies[] do
     ModifyCDCV alue(SRD, CDC) ;

   forall the cat ∈ GetSeekingCats(Cats) do
     if GetFitnessV alue(cat)   = PreviousFitnessValue
     then
       CalculatePosition Probability(cat,
       positionProbability);

   if AllFitnssesAreEquals() == true then
     setCatPositionProbability(positionProbability, 1);
                  > set position probability to 1
     moveToNextPosition(cat, positionProbability);
```

4) The Tracing Mode: The proposed extension to the PCSO algorithm is illustrated by way of the usage of Calculate Inertia Weight characteristic to update the rate of cats the usage of Update Velocity characteristic. Then the placement of the cat is up to date using Update Cat Position characteristic.

***Procedure** TracingModeProcedure*

```
Input : Cats
Output: Cats
begin
  forall the cat in Cats do
    CalculateInertiaWeight(cat, inertiaW eight) ;
               > returns the inertia weight of cat
    cat.velocity = UpdateVelocity(cat) ;
    if cat.velocity ≥ VelocityLimit then
      cat.velocity = VelocityLimit ;
               > keep velocity inside velocity interval
    UpdateCatPosition(cat) ;
```

VI. EXPERIMENTS AND RESULTS

In this section, we describe the test eventualities used to validate the proposed algorithm and the specific parameters for its execution. Then, we present and speak our approach results.

A. Test scenarios

We have identified two sample Workflows from for the assessment of our proposed set of rules. Each Workflow is

TABLE 4. Definition Intervals for the Used Quality Metrics

Quality metric and unit	Definition interval
Execution time (minutes)	[0.01,2.0]
Execution cost ($)	[0.01,5.0]
Throughput (MB/sec)	[0.01,0.2]
Data amount per service(GB)	[0.01,2.0]
Storage cost ($/GB)	[0.01,1.0]
Availability	[0.9750,0.9999]

TABLE V. Generic Parameters of the CSO Algorithm

Parameter	MP	SRD	CDC	MR	cl	Rl
Value	5	20%	80%	10%	2.05	(0,1)

represented with its DAG graph, a hard and fast of VMs which can carry out each project of the Workflow (each VM is described by means of its CPU potential, memory ability) a set of garage resources that can store the Workflow's information (everyone is defined with the aid of its memory ability) and six tables representing the taken into consideration exceptional metrics' values of every aid. All the take a look at information (the quality metrics values) had been generated with random statistics that fit inside the reality. The first check Workflow, illustrated in Figure 1, represents the ING-N sub-Workflow. It incorporates 13 duties (with 10 applicants' VMs for every one), and 8 storage web sites (with five candidates garage sources for every one). The second check Workflow is a sub-workflow of the Ligo Ihope Workflow. It consists of a hundred and fifteen tasks (with 50 candidates' VMs for each one), and 25 garage sites (with 20 applicants storage resources for every one). The values of the high-quality metrics of each candidate VM (execution time, execution value, records switch time, availability), and people of storage sources (availability, storage value, quantity of statistics saved according to provider) are randomly generated inside a fixed c program language period with a uniform chance as proven inside the desk .

B. Parameters of the PCSO and PSO algorithms

We will examine our experimentation effects for the proposed set of rules EPCSO with the same old PSO set of rules with a varying inertia weight and both the CSO (Cat Swarm Optimization) set of rules used in and the PCSO (Parallel Cat Swarm Optimization) set of rules. The table V illustrates the general parameters of the CSO algorithm. For the unique parameters of the PCSO set of rules, we lessen the range of companies to 2 and the ECH parameter to twenty. The initial populace size of cats is about to 32. The most variety of iterations is ready to 2000. For the same old PSO algorithm, we recollect c1=c2=2, and the inertia weight 'w' various from 0.4 at the start to 0.9 at the prevent of execution of the set of guidelines.

C. Results and interpretations

The following figures summarize the check outcomes for all tested algorithms, namely: EPCSO, PCSO, CSO and PSO. The algorithms had been carried out on the two take a look at scenarios provided above.

Each take a look at state of affairs modified into executed 50 instances and the effects supplied on this phase are the average values. More precisely, the determine 2 indicates the evolution of the global best of storage of the primary examined Workflow for the three algorithms.

This determine attests that the proposed algorithm (EPCSO) gives higher results than the other ones (CSO and PCSO).The discern three confirms this end for the second examined Workflow. The distinction among the consequences of the PCSO and EPCSO seems

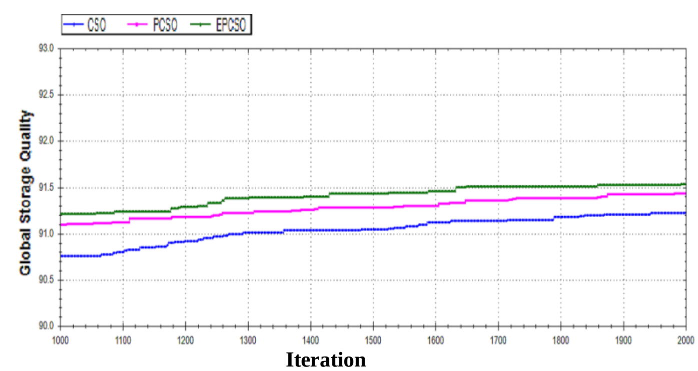

Figure 2. Quality of storage for the first scenario

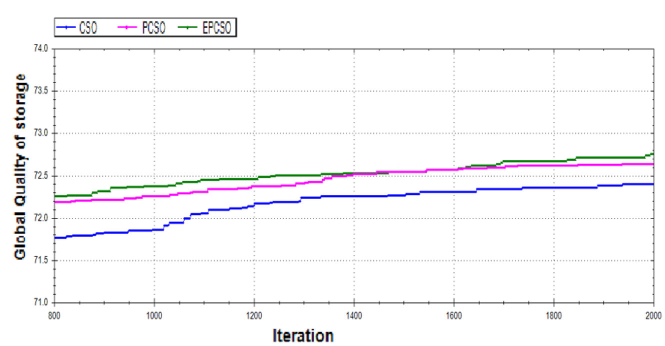

Figure 3. Quality of storage for the second scenario

low for a few iterations, but the proposed set of rules usually reaches at the give up a better end result than the others. Figure 4 shows the evolution of the overall pleasant of carrier of the second one Workflow for the EPCSO, CSO and PSO algorithms. It proves that the proposed set of rules plays better than the CSO and PSO algorithms and affords the great cost of world high-quality of provider. Also, the evolution of the solution furnished by the EPCSO and CSO algorithms is actually uniform in contrast to those supplied by using the PSO algorithm. This proves that the proposed algorithm allow averting premature convergence of the algorithm. The Figure 5 indicates the evolution of the overall great of carrier of the same algorithms for the first check situation and affirms the capability of the proposed set of rules to achieve higher values of the global fine of provider than PSO and CSO algorithms.

We have also tested the evolution of each satisfactory metric (execution time, cost, availability, and records switch time) for the specific take a look at situations. The outcomes show that the proposed algorithm presents higher outcomes as compared to the alternative tested algorithms. Figure 6 describes the evolution of the worldwide fee of the Workflow for the distinct algorithms and shows that the proposed set of rules presents the nice result. Likewise, figure 7 illustrates the evolution of the general availability of services for the primary check state of affairs.

This determine indicates that the proposed algorithm achieves a better availability charge as compared to the ones provided by using other tested algorithms. Finally, the determine eight, describing the evolution of the value of the worldwide information transmission time among the specific obligations of the second test scenario, certifies the ability of the proposed algorithm to offer a higher outcome for all the examined nice metrics and for the global excellent of provider.

In end, the proposed set of rules allowed attaining better quality of service fee as compared to the PSO, CSO and PCSO algorithms. The performances executed through our set of rules are because of the enormous desire of pace and function.

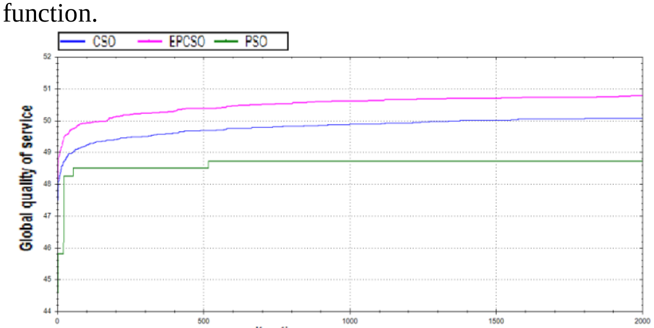

Figure 4. Evolution of the global quality of service for the second test scenario

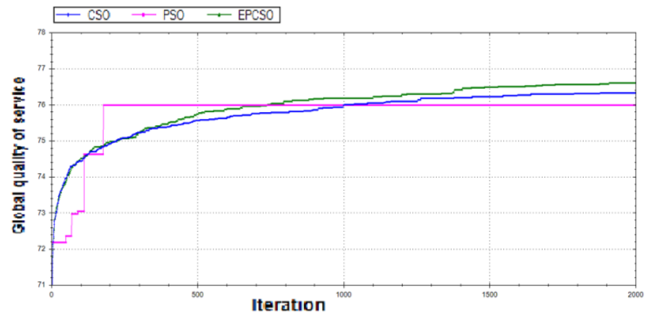

Figure 5. Evolution of the global quality of service for the first test scenario

Updating within the set of rules (using the inertia weight feature) not like the random evolution strategy adopted in well-known CSO and PCSO algorithms. The proposed extension for the PCSO set of rules doesn't have an effect on the computational complexity of the algorithm since it uses the values generated by using the set of rules for the computation of the inertia weight, however in the contrary it permits finding the global quality solution quicker by using guiding the seek process.

VII . CONCLUSION

In this work, we offered a QoS-conscious scheduling set of rules for Workflow applications inside the Cloud Computing surroundings primarily based on Parallel Cat Swarm Optimization (PCSO) algorithm. The goal of the proposed algorithm is to optimize the general great of carrier of the Workflow. To do that, we taken into consideration the first-rate of carrier as three sub-excellent additives: the first-class of execution of the computing resources that can perform the Workflow's tasks (decided on consistent with their CPU/memory/network capacities), the fine of garage of wished facts and the best of communique among all

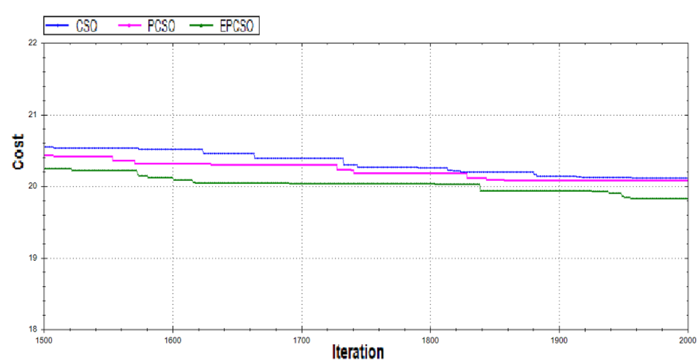

Fig. 6. Evolution of the global cost for the first test Workflow

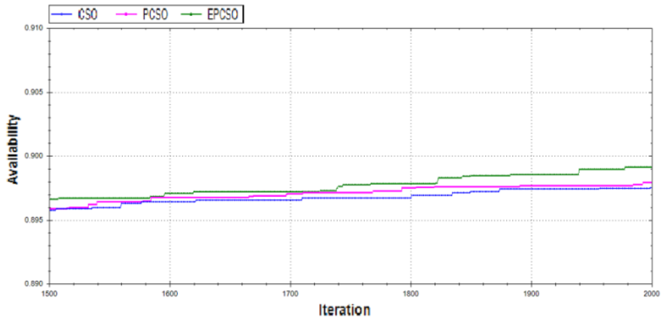

Figure 7. Evolution of the global availability of the first test Workflow

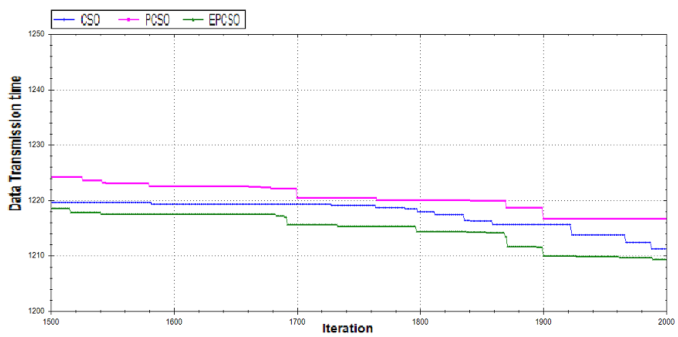

Figure 8. Evolution of the global data transmission time for the first test Workflow

the used resources. We used the following quality metrics: execution time, execution and storage cost, availability of resources and data transmission time between computing and storage resources of the Workflow. We formulate our scheduling problem in two mono-objective linear programs. The first one aims to select the best storage resources that optimize the quality of storage of the workflow. The second one aims to select the best computing resources that optimize the overall quality of service of the Workflow. We extended the PCSO algorithm to solve our optimizations problems. We compared the results obtained by our algorithm to those of the standard PSO, the CSO and the PCSO algorithms. We found that the proposed algorithm can achieve better results than the other ones. The algorithm we proposed is generic as it can be used, for example, to solve the problem of task scheduling of other test benchmarks. As part of our future work, we would like to deal with the provisioning of Cloud resources during the execution of Workflows applications in the Cloud Computing environment.

REFERENCES

[1] L. Guo, S. Zhao, S. Shen, and C. Jiang, " Task scheduling optimization in cloud computing based on heuristic algorithm", Journal of Networks, Vol.7, Issue.3, pp.547–553, 2012.

[2] E. Deelman, "*Grids and clouds: Making workflow applications work in heterogeneous distributed environments*", International Journal of High Performance Computing Applications, Vol.24, Issue.3, pp.284-298, 2010.

[3] MRahman, X. Li, and H. N. Palit, "*Hybrid heuristic for scheduling data analytics workflow applications in hybrid cloud environment*", In Proceedings of the 25th IEEE International Symposium on Parallel and Distributed, ser. IPDPS Workshops. Anchorage (Alaska) USA, Anchorage (Alaska) USA:966-974, May 2011.

[4] Z. Wu, Z. Ni, L. Gu, and X. Liu, "*A revised discrete particle swarm optimization for cloud workflow scheduling*" CIS10, 2010.

[5] S. Pandey, L. Wu, S. M. Guru, and R. Buyya, "A particle swarm optimization-based heuristic for scheduling workflow applications in cloud computing environments", AINA2010, 2010.

[6] M. A. Rodriguez and R. Buyya, "*Deadline based resource provisioning and scheduling algorithm for scientific workflows on clouds*", *IEEE TRANSACTIONS ON CLOUD COMPUTING*, 2014.

[7] R. Achary, V. Vityanathan, P. Raj, and S. Nagarajan, "*Dynamic job scheduling using ant colony optimization for mobile cloud computing*", Advance in Intelligent Systems and Computing, Springer International Publishing, Switzerland, Vol.321, Issue. pp. 70-71, 2015.

[8] S. Bilgaiyan, S. Sagnika, and M. Das, "*A multi-objective cat swarm optimization algorithm for workflow scheduling in cloud computing environment*", Intelligent Computing, Communication and Devices, Advances in Intelligent Systems and Computing, Vol.308, Issue.4, Springer, India, 2015.

[9] Z. Wu, Z. Ni, L. Gu, and X. Liu, "*A revised discrete particle Swarm optimization for cloud workflow scheduling*", 6th International Conference on Computational Intelligence and Security, ser. CIS'2010, page 184-188, 2010.

[10] V.P.Muthukumar and R.Saranya, "*A Survey on Security Threats and Attacks in Cloud Computing*", International Journal of Computer Sciences and Engineering, Vol.2, Issue.11, pp.120-125, 2014.

[11] L. Zeng, B. Benatallah, A. H. Ngu, M. Dumas, J. Kalagnanam, and H. Chang, "*Qos-aware middleware for web services Composition*" Software Engineering, Vol.30, Issue.5, pp.311–327, 2004.

[12] S. C. Chu and P. W. Tsai, "*Computational intelligence based on the behaviour of cats*", International Journal of Innovative Computing Information and Control, Vol.3, Issue.1, pp.163–173, 2007.

[13] P. W. Tsai, J. S. Pan, S. M. Chen, B. Y. Liao, and S. P. Hao, "*Parallel cat swarm optimization*", In Proceedings of the seventh International conference on machine learning and cybernetics, Kunming, China, Vol.1, ISSN:3328-3333, 2008.

[14] T. T. Nguyen, S. Yang, and J. Branke, "*Evolutionary dynamic optimization: a survey of the state of the art*", Swarm and Evolutionary Computation, Vol.6, Issue.3, pp.1–24, 2012.

[15] Mandeep Kaur, Manoj Agnihotri, "*A Hybrid Technique Using Genetic Algorithm and ANT Colony Optimization for Improving in Cloud Datacenter*", International Journal of Computer Sciences and Engineering, Vol.4, Issue.8, pp.100-105, 2016.

Authors Profile

Prachi Chaturvedi,pursued Bachlors of engineering from Institute of Technology and Management (ITM), Gwalior (MP), India in 2014.She is currently pursuing her Master of Engineering from Madhav Institute of Technology and Science, Gwalior (MP), India.

Sanjiv Sharma PhD, M.Tech(IT), B.E.(IT) is an Assistant Professor in the Department of Computer Science Engineering and Information Technology at Madhav Institute of Technology & Science Gwalior (MP), India. He received his PhD degree (Computer Science & Engineering) from Banasthali University, Jaipur (Raj.), India in 2014 and M.Tech (Information Technology) with honors from School Of Information Technology, Rajiv Gandhi Proudyogiki Vishwavidyalaya, Bhopal(MP), India in 2007. His current research interests include Social Network Analysis, Data

Mining, Network Security, Adhoc Network and Mobile Computing and their interdependency.

