

A Particle Swarm Optimization Approach for Workflow Scheduling with CPOP on Cloud Resources priced by CPU Frequency

¹Manzoor Mohammad, ²Meduri Manohar, ³Baduguna Swapna

Joginpally B.R Engineering College, Bhaskar Nagar, Yenkapally,Hyd-Telangana, India

Abstract: Due to low cost, high performance and energy efficient computing cloud computing attracts many organizations to choose cloud service providers. In this paper we propose an approach considering particle swam optimization (PSO) to guide the user to partition CPU total frequency among resources in order to minimize the execution time (makespan) of the workflow. The proposed procedure was derived and compared with a naive approach, which selects only identical CPU frequency configurations for resources. We apply CPOP algorithm as a makespan scheduler which is a famous DAG scheduling algorithm. By using makespan as fitness value the PSO moves the particle position further at each repeat to get the optimal solution.

Key terms- cloud computing, makespan, work flow scheduling, PSO;

I. INTRODUCTION

Executing workflows using public cloud providers having many advantages, while working with execution of workflows with data dependencies jobs can be assigned to different resources. The makespan of the workflow depends on computational power

allocated for each resource. A scheduling agent does the job of distributing workflows to different resources in order to speed up the execution of the application[1]. To minimize the makespan, scheduling agent tries to select resources based on their processors speed based on its CPU frequency configuration.

In this paper we assume cloud service providers who own large server clusters and cloud resources provides their service priced by profits that they assure by charging the end users for the service access. If the CPU

capacity is split equally for different resources it is observed that it may not be the best choice to minimize the makespan of the workflow. Dividing

an amount of CPU frequency into equal sizes for the all resources is considered a naïve approach . In this paper there is a need to find out which is best CPU frequency configuration to minimize the makespan of workflow.

The main focus of this paper is to propose a procedure to help the user to select an appropriate CPU frequency for resources to ensure the makespan of the workflow is minimized. Therefore for a given number of resources and capacity of CPU frequency to be split among these resources. This paper shows the user that splitting CPU frequency appropriately for resources can speed up the execution of workflow.

The proposed procedure applies particle swam optimization [2] strategy in conjunction with the CPOP scheduling algorithm[3]. The proposed strategy objective is to help the user to split the total CPU capacity for resources that minimizes the makespan of the various workflow. As a result it is flexible to the available budget. In this paper we assume that the user has a budget to which is enough to pay for this amount of CPU capacity that makes use of the naïve approach. Simulation results shows that the proposed system can reduce the makespan up to 20% over the naïve approach.

This paper is ranged as follows. Section II describes the related work. Section III includes background and key concepts. Proposed procedure discussed in section IV. In section V we discussed evaluation and section VI results. Section VII concludes the paper.

II. RELATED WORK

Due to features of low cost, high performance and energy efficient computing, cloud computing attract many organizations to choose cloud services. In cloud Workflow applications can be denoted by a

directed acyclic graph (DAG) represented by $G=(N, E)$ where $N=\{T_1, T_2, \dots, T_n\}$ is the no of tasks & E represents the data dependencies between these tasks. For example $f_{ij}=(T_i, T_j) \in E$ then T_i is the resources grant by the i^{th} task and T_j resources acquired by T_j .

Pietri et al. described an algorithm [4] that achieves low cost provisioning by selecting different CPU frequencies for every resource in order to execute workflows within the estimated time.

III. BACKGROUND AND KEY CONCEPTS

a. Workflow

A workflow application basically represented by a DAG (directed acyclic graph) in which nodes represent tasks; edges represent task dependencies [5]. Each task describes how much computing power a task requires to be executed, and each edge describes how much data must be transferred between two tasks. A task can start after its entire parent tasks have been completed and all necessary data has been received.

b. Workflows scheduling in clouds

Cloud computing environment provides the facility that executing workflows by providing dynamically virtualized resources on demand on pricing per resource basis.

c. Cloud computing environment

In this paper we are assuming that a company has a broker in its surroundings, which is responsible for leasing resources from cloud providers and preparing those resources to deploy and run the applications. The broker receives the requests for workflow execution and leases resources based on the schedules provided by the scheduler. We also assume that the resources are on-demand provisioned during the execution of applications and the user is priced according to the total of CPU frequency allocated to the resources.

IV. PROPOSED PROCEDURE BY APPLYING PARTICLE SWARM OPTIMIZATION

a. Particle swarm optimization

Particle swarm optimization (PSO) is an intelligent optimization algorithm belongs to meta-heuristic optimization algorithms [6]. PSO is inspired by social behavior of animals such as Fish

and Bird. It is a simple, yet powerful optimization algorithm. It can be applied to various fields of engineering. Every particle in swarm has a position vector $X[]$ and velocity vector $V[]$ which has magnitude and direction. Velocity describes the movement of particle in search space. Each particle performance which is calculated by fitness function f_p influences the position of the particle. For every movement the particle has a best value $p_{best,i}$ which is minimum or maximum based on problem of fitness function value of the particle among all particles. The best fitness value becomes the global best G_{best} . The velocity of each particle in swarm is measured for an instance as

$$V_i(t+1) = wV_i(t) + c_1r_1(p_{best,i}(t) - X_i(t)) + c_2r_2(G_{best} - X_i(t)) \quad \text{--- (1)}$$

Here w is inertia weight which converges the solution to origin, c_1 is acceleration coefficient and c_2 is social component, r_1, r_2 are random numbers distributed from [0 to swarm size]. Different inertia weights effects largely in converging the solution space [7].

M. Clerc and J. Kennedy introduced new inertia and c_1, c_2 coefficient values to converge the solution space [8]. In this paper W, c_1, c_2 values obtained by

$$W = \mu$$

$$c_1 = \mu \emptyset_1$$

$$c_2 = \mu \emptyset_2$$

$$\mu = \frac{2k}{12 - \emptyset_1 - \sqrt{\emptyset_1^2 - 4\emptyset_2}} \quad \text{----- (2)}$$

$$\text{Where } k=1, \emptyset_1=2.05, \emptyset_2=2.05,$$

$$\emptyset = \emptyset_1 + \emptyset_2 \quad 4$$

The position of particle at an instance can be measured as follow

$$X_i(t+1) = X_i(t) + V_i(t+1) \quad \text{----- (3)}$$

The particle velocity V_i value can be bounded to the range $[V_{min}, V_{max}]$ to control the particle movement in search space. The particle searches in the solution space until certain condition or Maximum iterations reached.

b. Problem formulation

In PSO the swarm size is number of particles that search the solution of the problem. At first every particle is initialized with random values in the range of Var_{min}, Var_{max} in search space with initial velocity as zero. A particle in PSO is determined by its position in $nVar$ dimensions,

where $nVar > 0$. In this work m is the total available resources to be allocated by user and the value assigned to each particle's dimension represents the value for the CPU frequency allocation of corresponding resource of that dimension.

	D1	D2	D3	D4	D5	D6
Particle	f_{r1}	f_{r2}	f_{r3}	f_{r4}	f_{r5}	f_{r6}
	R1	R2	R3	R4	R5	R6

Fig1: Particle of $nVar=5$ dimensions for each resource R_i the frequency value f_{ri} is assigned.

After initialization of particle fitness value will be calculated using fitness function and that value is treated as particles best value P_{best} , among all particles global best value is selected as G_{best} . Now all the particles move towards G_{best} particle.

Algorithm 1:

```

Initialize particle all particles
Repeat for all particles
    Calculate fitness value using fitness function
    If current fitness value is best then update  $P_{best}$ 
    If  $P_{best} < G_{best}$  Update  $G_{best}$  as  $P_{best}$ 
End

```

c. Fitness function

To measure best value for particle we need fitness function. In this paper we use a workflow scheduler which is responsible for calculating the performance of the particle by measuring the makespan of workflow by using the CPU frequency of resources given by the particle. We apply CPOP algorithm as a makespan scheduler which is a famous DAG scheduling algorithm. By using the makespan as fitness values, The PSO moves the particles position further at each repeat to get the optimal solution.

The cloud resource provides determines a minimum CPU frequency f_r min which can be assigned to a resource. The total capacity of CPU is F_{tot} to be split into m resources. We supply F_{tot} and m values as inputs to PSO and measure particle position. The solution is eligible only if

$$f_{ri} \leq F_{tot} \text{ and } f_{rmin} \leq f_{ri} \leq f_{rmax} \text{ ----- (4)}$$

Where f_{ri} is sum of all frequencies for a particle, f_{rmin} and f_{rmax} are lower and upper bound frequencies to assign for a resource. f_{rmax} calculated as

$$f_{rmax} = F_{tot} - [(m-1) * f_{rmin}] \text{ ----- (5)}$$

After creating particle each particle initialized for all dimensions with the values between f_{rmin} and f_{rmax} . Then these particles with values are given as input to fitness algorithm(Scheduler) to calculate fitness value.

Algorithm 2: Fitness Function

Input: a particle $P[]$, scheduler S , workflow w , a set of resources $R[]$

If equation 3 satisfied then

$$P[i] = f_{ri}$$

Schedule W on R using *scheduler S*

Return *makespan* of W on R as fitness value

end if

otherwise,

Return *Infinity* as *makespan*

In algorithm, if the solution is qualified then scheduler schedules the workflow and return's makespan otherwise return's ∞ as makespan. As we used CPOP as scheduler, The time complexity is $O(m*p)$ where m is no of resources($nVar$) and p is no of work flows.

V. EVALUATION METHODOLOGY

To compare the performance of proposed PSO based CPU scheduling with naïve approach we simulate both algorithms in MATLAB(Math Works) .

a. Workflows

TABLE I: Samples of cloud resources

I/P	samples					
nVar (No. of Recourses)	A	B	C	D	E	F
	2	4	6	8	10	12
F(GHz)	2	4	6	8	10	12

for testing we took six samples the configuration of problem definition shown in Table 1.

Table 2 shows the minimum frequency to be allotted for each resource and measured maximum frequency can be assigned to a resource.

TABLE II: Parameters used by the fitness function

FFP	samples					
f_{min} (GHz)	A	B	C	D	E	F
	0.8	0.8	0.8	0.8	0.8	0.8
f_{max} (GHz)	1.2	1.6	2	2.4	2.8	3.2

*FFP is Fitness Function parameters

b. Naive approach

In This paper naïve approach is selecting the identical frequencies for resources from cloud. So the use of naïve approach was considered in this paper with the purpose of comparing the efficiency of the results obtained from the PSO-based approach.

VI. RESULTS

Impact on the Makespan by using the PSO-based procedure

The fig 1 shows the results for various CPU frequencies result in various makespan estimates. By using the PSO based proposed procedure ,the scheduler achieves better results when we compared with the naïve approach for all the samples used in the evaluation.

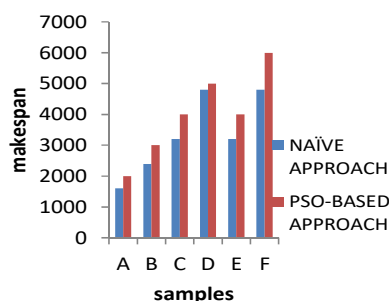


Table III Makespan of work flow

Sample	Naïve Based	PSO Based
	Makespan(MHz)	Makespan(MHz)
A	2000	1600
B	3000	2400
C	4000	3200
D	5000	4800
E	4000	3200
F	6000	4800

VII. CONCLUSION

In this paper we proposed a PSO (particle swarm optimization) based approach to know the user in splitting an amount of CPU capacity among fixed number of resources to minimize the execution time of the workflow(makespan). This paper results are compared with the naïve approach. Simulation results shows that the proposed procedure PSO can decreases the total makespan by 20%.

References

- [1] M. L. Pinedo, Scheduling: Theory, Algorithms, and Systems. Springer Publishing Company, 2008.
- [2] J. Kennedy and R. Eberhart, "Particle swarm optimization", in IEEE Int'l Conf. on Neural Networks, vol.4,Nov 1995, pp. 1942-1948 vol. 4.
- [3] H. Topcuoglu and S. Hariri "Task Scheduling Algorithms for Heterogeneous Processors" IEEE Int'l Conf. on Neural Networks, vol.4,Apr 1999, pp. 1097-5209.
- [4] I. Pietri and R. Sakellariou,"Cost-efcient provisioning of cloud resources priced by CPU frequency", in 7th International Conference on Utility and Cloud Computing, December 2014.
- [5] Guoqi Xie, Renfa Li, Xiongren Xiao and Yuekun Chen "A high performance DAG task scheduling algorithms for heterogeneous networked embedded systems". IEEE Int'l Conf. on Advanced Information Networking and Applications, 2014.
- [6] S.Pandey, L. Wu, S. Guru, and R. Buyya, "Aparticle swarm optimization based heuristic for scheduling work flow applications in Cloud computing environments", in 24th IEEE Int'l Conf. on Advanced Information Networking and Applications, April2010, pp.400407.
- [7] J. C. Bansal, P. K. Singh, M. Saraswat, A. Verma, S. S. Jadon and A. Abraham, Inertia Weight strategies

-
- in Particle Swarm 2011 Third World Congress on Nature and Optimization, in Biologically Inspired Computing, 2011:633640.
- [8] M Clerc and J Kennedy “The Particle Swarm – Explosion, Stability, and convergence in a multi dimensional complex space”. Vol 6, Issue 1, Feb 2002.
- [9] Wang Zhen, Zhang Jihui, SI Jianfei “ Application of particle swarm optimization with stochastic Inertia weight Strategy to Resource scheduling and assignment problem in cloud manufacturing environment”. July 2014.