1.5em

Cuckoo Search-Ant Colony Optimization Based Scheduling in Grid Computing

Dharmendra Prasad Mahato

BIT Sindri, Dhanbad
Jharkhand, India

dpmahato.rs.cse13@iitbhu.ac.in

47th International Conference on Parallel Processing Companion

August 13-16, 2018

Overview

Introduction

- Scheduling transactions in grid computing system is known to be NP-hard problem.

Introduction

- Scheduling transactions in grid computing system is known to be NP-hard problem.
- In order to solve this problem, this paper introduces a hybrid approach named cuckoo search-ant colony optimization.

Introduction

- Scheduling transactions in grid computing system is known to be NP-hard problem.
- In order to solve this problem, this paper introduces a hybrid approach named cuckoo search-ant colony optimization.
- We use cuckoo search method for making clusters of resources based on their load.

Introduction

- Scheduling transactions in grid computing system is known to be NP-hard problem.
- In order to solve this problem, this paper introduces a hybrid approach named cuckoo search-ant colony optimization.
- We use cuckoo search method for making clusters of resources based on their load.
- We use ant colony optimization for selecting the appropriate and optimal resources.

Introduction

- Scheduling transactions in grid computing system is known to be NP-hard problem.
- In order to solve this problem, this paper introduces a hybrid approach named cuckoo search-ant colony optimization.
- We use cuckoo search method for making clusters of resources based on their load.
- We use ant colony optimization for selecting the appropriate and optimal resources.

Problem Definition/Research Question

Block 1

the transaction processing model considering load balanced scheduling is formulated as follows:

$$\begin{aligned} \text{Minimize} \quad & L \\ \text{subject to} \quad & proc_j \leq makespan(T), \forall j = 1, \ldots, n. \end{aligned} \tag{1}$$

where the constraint states that the processing time ($proc_j$) of all the transactions on node N_j should lie within its makespan.

Table : Divide the solution to K categories based on random cluster numbers in the solution S_1.

c_1	c_2	c_3
3	5	1
6	7	2
–	8	4

Table : Calculated cost value for all solutions.

Solution	N_1	N_2	N_3	N_4	N_5	N_6	N_7	N_8	*Cost*
S_1	3	3	1	3	2	1	2	2	**6.5**
S_2	3	3	3	1	2	2	3	2	7.0
S_3	3	1	2	1	1	3	2	2	7.5

Table : Calculated fitness value for dataset objects by S_1, S_2 and S_3.

Nodes	load	Fitness of S_1	Fitness of S_2	Fitness of S_3
1	3	0.5	0.5	0
2	8	3.0	3.0	3.0
3	5	0	1.5	0
4	2	0	0	0
5	4	0	0	1.0
6	6	0.5	1.0	2.0
7	7	1.5	0	1.0
8	6	1.0	1.0	0.5

Table : Parameters of LBTS_ACO

Parameters	Value
α	0.5
β	0.5
ρ	0.2
d	0.25

LBTS_ACO

```
Select the node from the appropriate cluster using Cuckoo Search Algorithm
k = 1                                                  ▷ Iteration starts
P_k = 0 (P_k ≤ 1) L_j ⩽ n/2 L_bs ← L_j
Update the pheromone_load of selected node             ▷ pheromone_load update
Calculate the quality of the node                      ▷ Finds quality of the node
L^p ← L_bs
k = k + 1
P_k = log(k)/log(K)
select random nodes N_r1 and N_r2
find L_r1 and L_r2
Calculate φ(L_j, L_r1, L_r2)  (0 ⩽ φ(L_j, L_r1, L_r2) ⩽ 1.5) L_r1 ≤ L_r2
L_bs ← L_r1
L_j ← L_r1
L_bs ← L_r2
L_j ← L_r2
Update the pheromone_load of the node                  ▷ pheromone_load update
```

LBTS_CSACO algorithm

1: Initialize the transaction T_i at node N_j
2: Find the load L_j of the node N_j
3: make N_j the best-so-far node N_{bs} All the transactions are not scheduled
4: Call LBTS_ACO
5: Make N_j the best-so-far node
6: Increment the load of N_j by 1

This experiment evaluates the performance among the algorithms concerning throughput. Here the average throughput of the grid transaction at different processing levels are presented. The transaction throughput is first measured for the grid transaction processing system (depicted in Figure 1) which shows that the LBTS_CSACO algorithm outperforms the LBTS_ACO, EO, GA, Randomized, DLB, and HLBA algorithms.

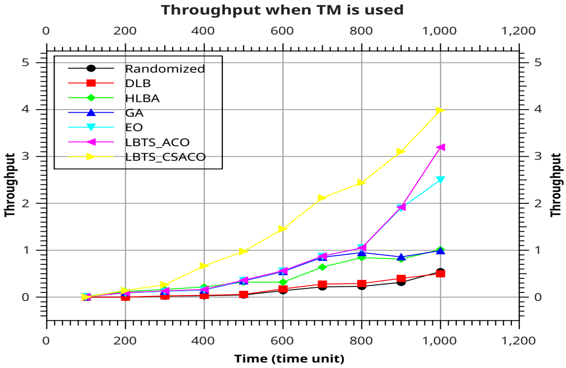

Figure : Throughput with Transaction Management

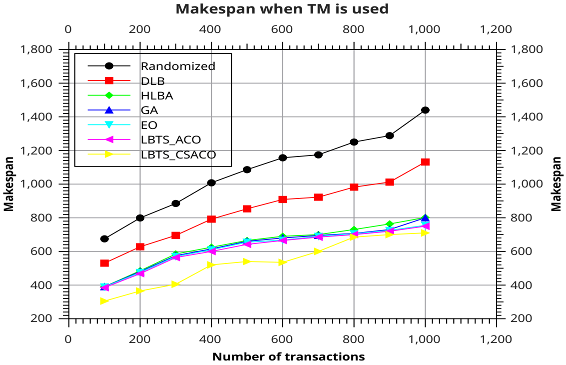

Figure : Makespan with Transaction Management

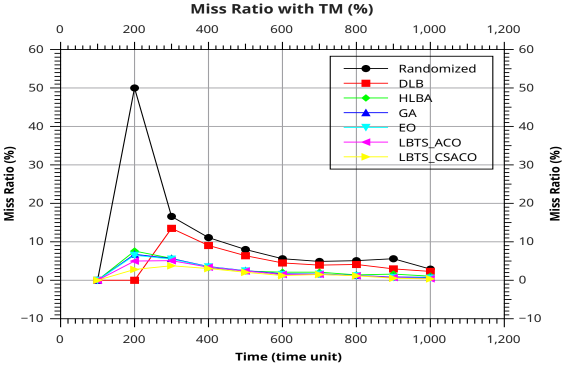

Figure : Miss Ratio with Transaction Management

Thank You