

# Quality Engineering of Distributed System with Hassle Free Load Curve Optimization

J. Jeyachitra  
Research Scholar  
Dept. of Computer  
Applications  
School of Information  
Technology  
Madurai Kamaraj University

K. Iyakutti, Ph.D  
Professor  
Dept. of Physics and  
Nanotechnology  
SRM University  
Chennai

K. Alagarsamy, Ph.D  
Associate Professor  
Dept. of Computer  
Applications  
School of Information  
Technology  
Madurai Kamaraj University

## ABSTRACT

Distributed system presents the possible approaches for distribution and aggregation of diverse set of resources such as computers, storage space systems and other specific devices. Normally, these set of resources are processed by the different set of users in different organizations. So, the users might contain diverse objectives and plans and their activities is very complex to differentiate. And also when the demand over the system increases, the load balancing problem arises and the traffic over the systems increases. So, it is necessary to improve the load balancing in the distributed system by means of distributing the application load with software engineering traffic principles. Jun Zhang et.al., presented a novel non-parametric technique for traffic classification which can develop the performance of the classification efficiently by integrating associated information into the classification method. However the traffic classification using non-parametric and non engineering approach fails to balance the load effectively in distributed system. So, in this work, we plan to improve the quality of load balancing in the distributed system by adapting Hassle-free Curved Optimization (HCO) traffic engineering principle. The HCO traffic engineering principle is presented with this research for effective load balancing and communication cost in distributed system. Heuristic algorithm is followed in HCO principle with given communication pattern to produces an optimal solution for effective distributed data/service communication. The experimental performance of HCO traffic engineering principle is evaluated with Solar Flare Data Set from UCI repository against existing non-parametric approach for effective load balancing scheme to attain quality assurance, effective load balancing with minimal communication cost.

## Keywords

Distributed system, load balancing, traffic engineering principle, Hassle-free curved optimization, Heuristic algorithm

## 1. INTRODUCTION

Load balancing in distributed computer systems is the procedure of reallocating the set of tasks for the processors in the system to enhance the system presentation. Numerous set of studies have revealed that allocating the work load randomly to the set of nodes in a distributed system normally enhances the system presentation and enhances the resource utilization. Dynamic load balancing algorithms checks the dynamic changes on the system work load and reallocate the work load consistently.

A load balancing process in the distributed system is normally consists of three set of approaches:

- Transfer approach,
- location approach, and
- information approach.

Initially, transfer approach checks the set of tasks which are necessary for transaction to other set of nodes for processing. Location approach will select the node as a remote node to achieve a transferred task. Information approach is the information center of a load balancing algorithm. It is necessary for providing the position and each node's transfer approach at every path node with the appropriate information desired to provide their decisions. Information approach is a significant section of a load balancing algorithm.

The charge and effectiveness of the load balancing algorithm provides heavily on the presentation of its information approach. The responsibility or access of the load balancing algorithm can provide three diverse set of formats:

- centralized,
- distributed, or
- semi-distributed.

Distributed technology is powerful and however, it is not a universal remedy for all situations. Traffic engineering involves adapting the routing of traffic to

the network conditions, with the joint goals of Quality-of-Service (QoS) load balancing and secure use of network resources. Traffic engineering principles involves a set of performance objectives that express the selection of paths, as well as effectual mechanisms for the routers to decide on paths that gratify these objectives. The end-to-end evaluation of QoS properties in distributed system using the traffic engineering principles is analyzed. As a result, many design flaws that affect QoS are not effective in load balancing and cost effective manner.

One of the today issues in traffic engineering is to find new effective ways to deal intelligently with the increasing complexity of distributed computing systems. The serious role is played by the balancing of the effort load amongst all nodes in a distributed system. It is self-possessed of interconnected nodes that come into and way out the system without following any rule. To address this issue, in this work, Hassle-free Curved Optimization (HCO) traffic engineering principle is introduced.

## **2. LITERATURE REVIEW**

In a distributed load balancing system, several load balancing strategies utilized which might be used according to the case or tailored to improve the presentation of the algorithm for resolving the particular problem, diverse set of strategies are presented where to identify the process of the load balancing algorithm have been proposed in the literature.

In [2], the author presented a absolute resolution for vigorously changing scheme association in a large-scale Byzantine-fault-tolerant scheme. The author presented an examination that trail classification relationship and occasionally informs other structure nodes of relationship changes. On the other hand, existing Byzantine-fault-tolerant storage space systems either presume a fixed set of imitations, or contain limitations in how they grip reconfigurations (e.g., scalability of the clarification or the constancy heights they give).

A network comprises of a collection of **conversing** hosts that provide a subjective network topology through several wireless communication standard. Network communications symbolize a diversification in statement technology essential to resolve the rigorous end-to-end necessities of QoS-based message networks. Of the loads of confronts in this compound distributed system, the trouble of routing supported with a predefined set of customer inclinations, serious to guaranteeing quality-of-service, is the center of this research. Purposely, the paper [3] customized a cluster-based QoS routing algorithm for networks with the endeavor of providing fault acceptance, which is a dangerous characteristic in providing QoS in the connection failure-prone surroundings of the networks.

A novel restricted quality of service (QoS) routing procedure for wireless sensor networks (WSN) is

proposed in [4]. The proposed procedure objects WSN's applications containing diverse kinds of data traffic. It is supported with the QoS requirements consistent with the data type, which allows providing quite a few and modified QoS metrics for every traffic category. The agent calculating model is quickly budding as one of the influential tools for the expansion of large-scale dispersed systems to cope with the indecision in a dynamic environment. The paper [5] scrutinizes an agent-based technique and its applications in dissimilar modes of transportation, as well as several transportations.

In [6], the author presented Saturn, overlay structural design for large-scale data networks continued over Distributed Hash Tables (DHTs) that professionally routes range queries and make sure access load balancing and fault-tolerance. Introducing successive data values in adjacent peers is attractive in DHTs as it quickens choice query processing; though, such an assignment is extremely vulnerable to load imbalances. Self-protection refers to the capability for a coordination to notice prohibited behaviors and to fight-back interruptions with counter-measures. The article [7] presented the proposal, the completion, and the assessment of a self-protected scheme which targets grouped dispersed applications.

The paper [8] proposed a novel fuzzy dynamic load balancing algorithm for homogenous dispersed systems. The proposed algorithm uses fuzzy logic, in commencing with imprecise load information, building load allocation choices, and upholding overall system steadiness [9, 10]. But it gives more importance only to the fuzzy logics by describing the loads and processors in the distributed system.

To enhance and balance the work load among the processors in the distributed system, in this work, Quality engineering principles is applied with HCO technique to obtain the optimal solution

## **3. PROPOSED METHODOLOGY**

Distributed system having various task processing computers, at times get heavily burdened and some others instance gets lightly loaded. The load imbalance in the distributed system hinders the performance of data communication services, which in turn affects the quality of distributed system. The software engineering traffic techniques of load balancing is applied to enhance the quality of the communication across the distributed system. In this work, an effective hassle free curved optimization technique is presented for balancing both the heavy and light load in the distributed systems. The hassle free curved optimization technique (HCO) resolves the traffic hassles present in the load imbalanced distributed system while minimizing the communication overhead among different set of distributed users. Hassles are normally received at any case, in which, heuristic algorithm is applied in the HCO technique to identify the hassles. To remove the hassles, the traffic engineering is defined in the

distributed system to enhance the load balancing strategy. The architecture diagram of the proposed HCO traffic engineering principle for load balancing in distributed system is shown in fig 3.1.

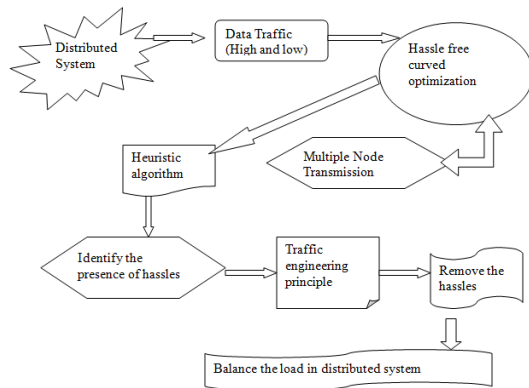


Fig 3.1 Architecture diagram of the proposed HCO traffic engineering principle

From fig 3.1, HCO model and a heuristic algorithmic solution is presented to optimize the balancing of the work load in distributed systems. It finds the optimal assignment for some pre-specified requirements on

total load and load balancing using heuristic algorithm via HCO principles.

Initially, the distributed system consists of set of system with users with resources provide the communication with other set of users in different organizations. If the demand over the set of resource increases, the traffic across the distributed system raises steeply. To resolve the normal traffic, at first, identify hassle curves in the load conditions with heuristic algorithm in the distributed system.

After the identification of hassles in the distributed system, the traffic engineering principle is applied and resolved the hassles. The HCO traffic engineering principle efficiently process, the communication among the users and balance the communication load in the distributed system.

### 3.1 System model

The HCO principle model is designed with a network, associated with blocked bus and the nodes in the network are to be sensibly prearranged from 1 to n, where n is the total number of nodes in the entire system as shown in Figure 1.

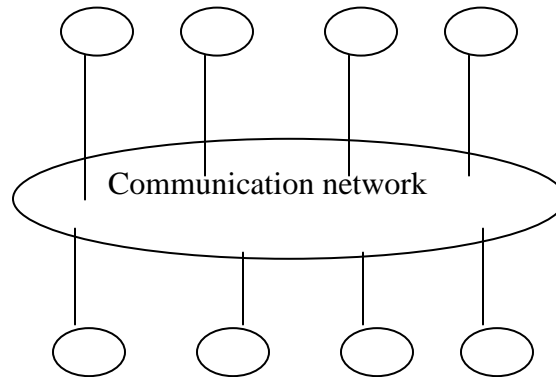


Fig 1 Distributed system model

The topology of the network in HCO principle model is efficiently configured. The nodes are designed in pairs, distributed, where no contentions suppose to happen; i. e., every node in the network does not create contact with any other node for the purpose of load balancing. Nevertheless, for every process in the load balancing, these nodes are to be changed. The group of nodes that share the load consists of three nodes each.

The presentation of the HCO principle model is presented as follows. Assume a set of number of nodes N which are associated by a network where every node in the network comprises a set of resources. Each task arrived to a set of nodes could either arrived from outside or from other nodes in the network. All nodes in the network should assign the same average arrival rate of tasks from the network.

- ☞ Nodes of the distributed system are efficiently sorted from 1 to N, where N is the total number of nodes in the distributed system. Assign id to each set of node in the system.
- ☞ Associate the nodes using the broadcast network and the processing gap of communication for two nodes is the same.
- ☞ Each node in the network has some state S with number of tasks  $T_n$ .

### 3.2 Load balancing using heuristic technique

The heuristic technique is utilized here for identifying the presence of hassles in the distributed system. The hassles are nothing but more than one task in the task set requests the same set of resource in the available resource pool. While requesting the resource like the

way, the processing task by that resource might get interrupted. So, there is a great chance of task being discarded. So, it is necessary to build a distributed system, hassle-free. At first step, the presence of hassles is needed to identify in the distributed system. The occurrence of data traffic both in the high and low level indicates the presence of hassles. The data traffic occurrence is determined using heuristic technique which balances the process of assigning tasks to the resources.

Consider a distributed system with number of tasks as  $t_n$  and the set of resources be  $R$ , where  $n$  is the total number of tasks ready to take over the available resources. These set of tasks are maintained by the agent  $A$ . If any new resource  $R_i$  enters into the network, then the capacity of the resources are identified based on the two elements, one is based on the processing elements  $P$  and the other one is based on MIPS rating of the processing element.

$$Cap(R_i) = P \times MIPS(P) \dots\dots\dots \text{Eqn 1}$$

where

$$MIPS(P) = \frac{t_i}{T \times 1,000,000}$$

Through the equation 1, the capacity of each of the resources in the network is determined. Based on the capacity, tasks are assigned to the set of resources depends on its requests. Set a flag  $f$  for each of the resource to determine whether the number of successful tasks in the system.

If any new task (new  $t$ ) arrives to the network and wants to assign it to the particular resource  $R_i$ , then the flag  $f$  is set according to the assignment.

$$R_i = newt, \dots\dots\dots \text{eqn 2}$$

Based on the assignment of flag, the loads in the distributed system are balanced. For instance, if the new task arrives, it will search for the resource needed to process the task. Before assigning the task to the particular resource, it is necessary to check the flag of the particular resource. If the flag is 1, then the resource is already processing the tasks for a certain interval of time. If the flag is 0, then assign the task to the particular resource. If more than one task requests the same set of resources, then the data traffic occurs. When the traffic occurs, the task in the queue will get hassled. The presence of hassles is identified based on the occurrence of data traffic in the distributed system. If new task is successfully done with the assigned resource, then increase the value of flag. The flag value increases based on the number of successful tasks processing in the particular resource is done. The possibility of assigning the new task with the resource  $R_i$  determined as,

$$P(R_i) = \frac{C_i(t)^a * init(R_i)}{\sum_n (C_i(t)^a * init(R_i))} \dots\dots\dots$$

eqn 3

Where  $i$  and  $n$  is the available set of resources  
 $C_i(t)$  – current processing task in the resource  
 Init ( $R_i$ ) – initial value of the resource  $i$   
 $a$ - intensity of the initial performance values

So the heuristic algorithm is planned with the aim of manipulating the presentation of the distributed system which depends upon numerous aspects and these aspects could be cut down for dropping the difficulties of the technique to be utilized. Load Balancing based on heuristic algorithm for the identification of hassles is presented below:

**// Algorithm**

**Input: Set of resources R, set of tasks T**

**Begin**

Initialize the available set of resources in the distributed system

Determine the processing tasks  $P$

Determine the MIPS rating of the particular task  $t_n$

Identify the number of tasks to be processed as  $n$

**Do**

**Begin**

Choose the task  $t_i$  from  $t_n$

Identify the resource  $R_i$  to assign the task  $t_i$  based on the  $cap(R_i)$

Assign the task

**If** task successfully completed

Flag=flag+1

**Else**

Keep the task in the task set  $T$

Wait for the appropriate resource

**End If**

**While** ( $t_n = \text{NULL}$ )

**End do**

**End**

All tasks are allocated to the resources depending on the capacity of the resources, which differs along with the position of the task at a particular resource. Resource with the highest capacity rate would obtain the task previous to any other resource.

**3.3 HCO traffic engineering principle**

HCO based traffic engineering principle is done based on resolving the hassles present in the distributed system while transmitting the data from source to destination. When more number of users request the same set of resources at the same time, then the traffic occurs. In that case, the utilization of the network decreases. With the heuristic technique, the presence of hassles are identified. To resolve the hassle, the HCO traffic engineering principles are used here.

To increase the network utilization, mapping the data traffic flows against the substantial topology and generate a consistent circulation of traffic. HCO based traffic engineering optimizes the effectiveness of the network through the mapping and distribution of traffic over the network to promise reasonable message delivery, and to evade or reduce congestion on any single path.

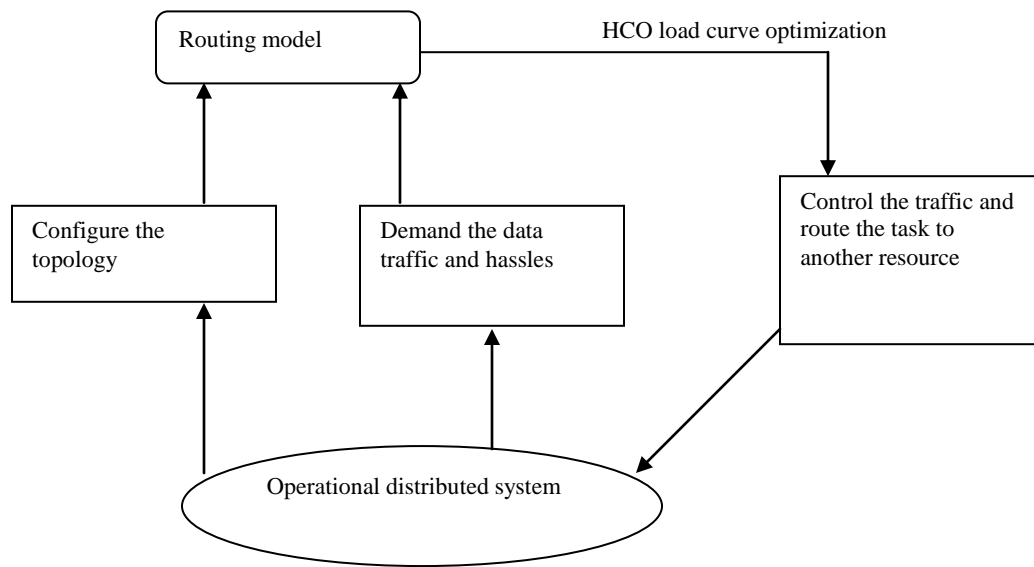
In the HCO based traffic engineering, it chooses the node for processing the task based on their service so as to plan the traffic flows onto a distributed system domain. HCO based traffic engineering algorithm processes a set of user requests received at the system domain. Each user's request specifies the amount of resources they needs and the set of service. Based on the information provided in the users' request query, the algorithm tries to situate the path that best gathers the request. Once the path is determined, it is

registered with the table, which is maintained for every node in network that determines the status of the links.

HCO traffic engineering works based on the three engineering principles, namely,

- ☞ measure,
- ☞ model and
- ☞ control.

The measure describes the process of measuring the presence of hassles in the distributed system. The model criteria describes the process of predicting the configuration model of the network to determine the flow of hassles. Control describes the process of controlling the hassles and route the task to some other resource in the distributed system. The main elements of the HCO traffic engineering principle is defined in fig 3.2.



**Fig 3.2 Main components of HCO traffic engineering framework**

The topology and configuration information of the distributed system are reasonably noted. The capacity of the available set of resources are identified for configuring the topology and assists to route the task into another set of resource in resource pool. Here, Simple Network Management Protocol (SNMP) is utilized to provide the information regarding the status of the processing elements. In addition, it is necessary to monitor the topology in the operational distributed system. The operational distributed system determines the volume of hassles among the routing path.

When all of the tasks in the task set choose the same path, then the tasks are moved to some other route in the distributed system. Normally, larger distributed systems are typically divided into several small networks with some set of resource pool to process the task. For resources in diverse set of areas, the resource

selection is done based on the information given by the flag set described in heuristic algorithm. A HCO principle simply splits the data traffic evenly over each of the routers in the distributed system. Through this procedure, the data traffic is cleared and balance the load evenly in the distributed system.

#### **4. EXPERIMENTAL EVALAUTION**

Experimental evaluation is done to estimate the performance of the proposed quality engineering of distributed system with hassle free load curve optimization. Experiments are conducted with the Solar Flare Data Set from UCI repository against existing non-parametric approach for the process of enhancing the load distribution system.

The description of solar flare data set is defined as follows (Table 1):

The solar flare database contains 3 potential classes, one for the number of times a certain type of solar flare occurred in a 24 hour period. Each instance represents captured features for 1 active region on the sun. The total number of instances used here are 323 and second set of data instances are measured to be 1066. The total number of attributes used here are 13 with 3 set of class attributes.

**Table 1 Details of solar flare dataset**

Dataset name	No. of instances	Total number of attributes	Number of class attributes
Solar flare dataset	323, 1066	10	3

The values of the attribute information are described in the subsequent table (table 2).

**Table 2 Attribute description**

Attribute	Values
Code for class	(A,B,C,D,E,F,H)
Code for largest spot size	(X,R,S,A,H,K)
Code for spot distribution	(X,O,I,C)
Activity	1-reduced, 2-unchanged
Evolution	1-decay, 2 – no growth, 3 - growth
24 hour flare activity code	1-nothing as big as an m1, 2- one M1
Historically complex	1-yes, 2- No
Area	1-small, 2- large
Area of the largest spot	1 = <=5, 2 = >5

From all these predictors three set of flare classes are predicted, which are represented in the last three columns.

C - Common flares - Class flares production in 24 hours;

M - Moderate flares - Class flares production in 24 hours;

X - Severe flares - Class flares production in 24 hours;

With these set, the experiments are conducted to estimate the performance of the proposed quality engineering scheme. The performance of the proposed Quality engineering of distributed system with hassle free load curve optimization approach is measured in terms of data traffic, hassles removal rate and load balancing rate.

Data traffic is measured in terms of rate at which the tasks in the task set are successfully routes to another set of available resource pool. The data traffic is measured based on the training dataset. The dataset taken for HCO principle approach experiments are measured based on the resource set available.

Hassles removal rate is measured in terms of rate at which how efficient the presence of hassles at the data traffic is removed. The hassles removal rate is measured based on the number of tasks in the task set.

Load balancing is measured in terms of rate at which how efficient the workloads are distributed among the multiple systems in the network. Load balancing rate is measured based on the total number of tasks in the task set.

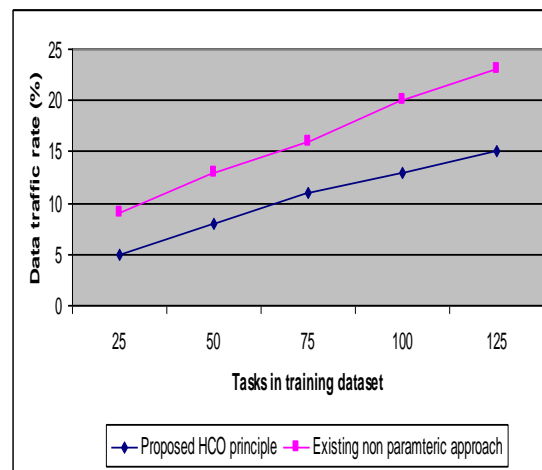
## 5. RESULTS AND DISCUSSION

Quality engineering of distributed system with hassle free load curve optimization approach is compared with the existing non-parametric technique for traffic classification in measuring the rate of balancing the load, clearing the data traffic on the distributed system, providing quality on communication pattern from source to destination. The below table and graph describes the performance of the proposed Quality engineering of distributed system with hassle free load curve optimization approach and existing non-parametric technique.

**Table 5.1 tasks in training dataset vs. data traffic rate**

Task in training dataset	Data traffic rate (%)	
	Proposed HCO principle	Existing non-parametric approach
25	5	9
50	8	13
75	11	16
100	13	20
125	15	23

The data traffic rate is measured based on the number of tasks in the distributed system. The value of the proposed HCO principle is compared with the existing non-parametric approach.



**Fig 5.1 tasks in training dataset vs. data traffic rate**

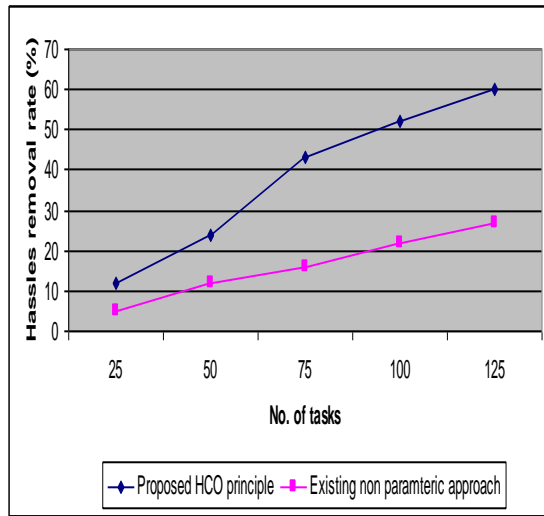
Fig 5.1 describes the occurrence of data traffic based on the number of tasks presents in the distributed network. Compared to the existing non-parametric

approach, the proposed HCO principle provides low data traffic rate since it followed the heuristic technique for the clearing the data traffic by means of identifying the presence of hassles in the system. But the existing non parametric approach fails to accomplish he identification and directly implement the technique for load balancing. So, the variance in the occurrence of data traffic rate is 5-10% less in the proposed HCO principle.

**Table 5.2 Number of tasks vs. hassles removal rate**

Number of tasks	Hassles removal rate (%)	
	Proposed HCO principle	Existing non-parametric approach
25	12	5
50	24	12
75	43	16
100	52	22
125	60	27

The hassles removal rate is measured based on the number of tasks in the distributed system. The value of the proposed HCO principle is compared with the existing non-parametric approach.



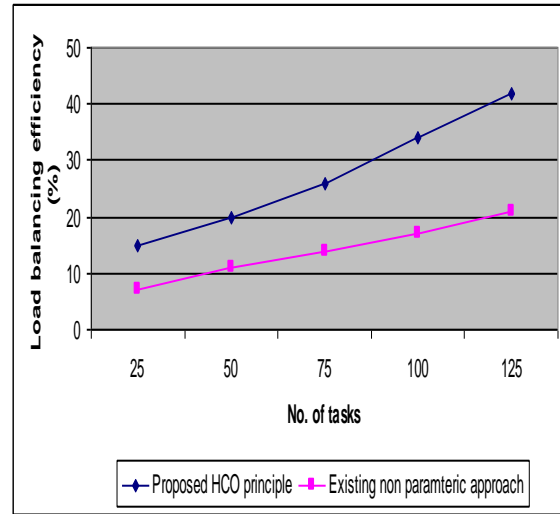
**Fig 5.2 Number of tasks vs. hassles removal rate**

Fig 5.2 describes the process of removal of hassles to accomplish the task execution based on the number of tasks presents in the distributed network. Compared to the existing non-parametric approach, the proposed HCO principle removes all the hassles presents while the presence of data traffic by means of applying the traffic engineering principle in the system. But the existing non parametric approach fails to transit the tasks to another set of resources in the system. So, the variance in the removal of hassles in terms of rate is 15-20% high in the proposed HCO principle.

**Table 5.3 Number of tasks vs. load balancing efficiency**

Number of tasks	Load balancing efficiency (%)	
	Proposed HCO principle	Existing non-parametric approach
25	15	7
50	20	11
75	26	14
100	34	17
125	42	21

The load balancing rate is measured based on the number of tasks in the distributed system. The value of the proposed HCO principle is compared with the existing non-parametric approach.



**Fig 5.3 Number of tasks vs. load balancing efficiency**

Fig 5.3 describes the process of identifying the efficiency of balancing the workload to accomplish the task execution based on the number of tasks presents in the distributed network. Compared to the existing non-parametric approach, the proposed HCO principle achieves better load balancing rate since the workload are evenly balanced only after clearing the data traffic through hassles. But the existing non parametric approach fails to balance the load in terms of task set via network classification. So, the variance in the efficiency load balancing rate in terms of rate is 10-15% high in the proposed HCO principle.

## 6. CONCLUSION

HCO traffic engineering principle is established as a proposed work for effective load balancing and communication cost in distributed system. Heuristic algorithm is followed in the HCO traffic engineering principle with given communication pattern to produces an optimal solution. Heuristic algorithm used here for identifying the presence of hassles based on the trial and error method. It defines the user's resource requests and capabilities of assigning resources in distributed system. It accesses the various

areas of the software recursively. Finally, the HCO traffic engineering principles for load balancing are incorporated efficiently for the load balancing in distributed system. Experimental evaluation is done with the dataset extracted from UCI repository for evaluating the performance of the proposed HOC traffic engineering principle and compared with the existing non parametric approach. Performance valuation showed that compared to existing non parametric approach, the proposed HCO traffic engineering principle achieved 20% efficiency in terms of load balancing in the distributed system.

## REFERENCES

- [1] Jun Zhang., Yang Xiang., Yu Wang, Wanlei Zhou., Yong Xiang, and Yong Guan., "Network Traffic Classification Using Correlation Information," IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, 2013
- [2] Rodrigo Rodrigues., Barbara Liskov., Kathryn Chen, Moses Liskov, and David Schultz., "Automatic Reconfiguration for Large-Scale Reliable Storage Systems.," IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING, VOL. 9, NO. 2, MARCH/APRIL 2012
- [3] Larry C. Llewellyn., Kenneth M. Hopkinson., and Scott R. Graham., "Distributed Fault-Tolerant Quality of Wireless Networks," IEEE TRANSACTIONS ON MOBILE COMPUTING, VOL. 10, NO. 2, FEBRUARY 2011
- [4] Djamel Djenouri., and Ilangko Balasingham., "Traffic-Differentiation-Based Modular QoS Localized Routing for Wireless Sensor Networks," IEEE TRANSACTIONS ON MOBILE COMPUTING, VOL. 10, NO. 6, JUNE 2011
- [5] Bo Chen., and Harry H. Cheng., " A Review of the Applications of Agent Technology in Traffic and Transportation Systems," IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, VOL. 11, NO. 2, JUNE 2010
- [6] Theoni Pitoura., Nikos Ntarmos., and Peter Triantafillou., "Saturn: Range Queries, Load Balancing and Fault Tolerance in DHT Data Systems," IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 24, NO. 7, JULY 2012
- [7] Noel De Palma., Daniel Hagimont, Fabienne Boyer., and Laurent Broto., "Self-Protection in a Clustered Distributed System.," IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, VOL. 23, NO. 2, FEBRUARY 2012
- [8] Ali M. Alakeel et. Al., 'A Fuzzy Dynamic Load Balancing Algorithm for Homogenous Distributed Systems', World Academy of Science, Engineering and Technology 61 2012
- [9] M. Rantonen, T. Frantti, and K. Leiviskä, "Fuzzy expert system for load balancing in symmetric multiprocessor systems," Journal of Expert Systems with Applications, Vol. 37 No. 12, pp. 8711-8720, December, 2010.
- [10] L. Singh, A. Narayan, and S. Kumar, "Dynamic fuzzy load balancing on LAM/MPI clusters with applications in parallel master-slave implementations of an evolutionary neuro-fuzzy learning system," IEEE International Conference on Fuzzy Systems, pp.1782-1788, June 2008.