Volume 1, No. 2, July-August 2010

ISSN No. 0976-5697

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

# Ant Colony Optimization for Replica Management in Distributed Spanning Tree Modeled Peer Network

P. Victer Paul* , T.Vengattaraman, P. Dhavachelvan
Department of Computer Science
Pondicherry University
Puducherry, India
victerpaul@gmail.com, vengat.mailbox@gmail.com, dhavachelvan@gmail.com

R. Baskaran
Department of Computer Science and Engineering
Anna University
Chennai, India
baskaran.ramachandran@gmail.com

V.S.K. Venkatachalapathy
Principal
Sri Manakula Vinayagar Engineering College
Pudücherry, India

***Abstract:*** Data replication is a common and vital approach to reduce the access latency and network traffic in large distributed systems. Global Replica Management (GRM) is the technique to maintain data consistent throughout the network which is very much required in all applications that support multi-peer collaborative works. But GRM is not favorable for many applications because, very large number of message passes needed for replica management process. In our previous work [1], an efficient technique for GRM is devised and analysis elucidate that the efficiency can be improved to some extend using Distributed Spanning Tree (DST) as interconnection structure. In this paper, we make use of Ant Colony Optimization (ACO) technique is to enhance the efficiency of GRM at higher level. Analysis from simulation of our proposed work shows that efficiency of GRM using DST can be much improved by optimizing network with ACO technique.

***Keywords:*** distributed spanning tree; replica management; peer network;Ant Colony Optimization;OMNeT++

## I. INTRODUCTION

Peer-to-peer (P2P) systems are distributed systems in which nodes of equal roles and capabilities exchange information and services directly with each other. Peer-to-peer networks are generally simpler, but they usually do not offer the same performance under heavy loads. Replica Management is a key issue to reduce bandwidth consumption, to improve data availability and to maintain data consistency in large distributed systems, as peer networks [2-4]. Most preferable replica management method is Global Replica Management (GRM), which maintains data consistency globally. But GRM is not favorable for many applications because, very large number of message passes needed for replica management process [1]. Traditional replication schemes [5-7] are static in the sense that the number and placement of replicas are predetermined and fixed. Thus the failure of any peer, which holds the replica, may lead to replica unavailability on the network. Thus it is required to dynamically select the peer that holds replica and on any failure there in, its content has to be transferred to another node so that the replica availability will never be affected. The global consistency of data operations on replicas is not desirable by many applications. Since large number of message passes is required to achieve the global consistency which overload the network with message and cause congestion [7]. So, Peer Network Collaborative work environment requires a system with Global Consistency with minimal number of message passes across the network.

To achieve the various requirements for effective GRM, in our previous work we used Distributed Spanning Tree as interconnection structure which exactly solves the problem to improve data availability and to maintain data consistency. The DST structure is organized into a hierarchy of groups. The nodes are put together in groups and groups are gathered in groups of higher level, recursively. This organization, built on top of routing tables allows the instantaneous creation of spanning trees rooted by any nodes and keeps the load balanced between the nodes [5]. Therefore we virtually converted the peer network into DSTs and each tree should have its root node we call it as Head Node (HN) and others are Leaf Node (LN). Every HN will hold the complete details regarding its LNs and vice versa. These HNs are generated dynamically and hold the replica which is accessible by their corresponding LNs which increases the data item availability and reduces the data latency. Using the replica and details of LNs by each HN in DST, we minimized the complexities like message pass overload and load imbalance that arise in the GRM process [1]. Thus from our previous work, GRM can be achieved effectively and at the same time we cannot assure that every system in the peer network be always connected to the network and so link partition occurs among the peers often. In this paper, a popular optimization technique, Ant Colony Optimization (ACO) is used to enhance the efficiency of GRM in Peer network by finding optimal path between peers in dynamic fashion. ACO [8-10], is a powerful heuristic approach to solve combinatorial optimization problems such as the TSP, Routing in telecommunication networks [11]. So applying ACO approach can enhance the effective routing of message (at low cost) in the network which in-tern reduces the number of message pass required for communication to achieve high level efficiency in GRM applications.

## II. BACKGROUND INFORMATION NEEDED

### *A. Interconnection Technique*

Distributed Spanning Tree (DST) [12,13] is the interconnection formation we follow as in [1], which improve the routing and reduce the number of message passes required for any communication in Peer network. DST systematizes Peer network into a hierarchy of groups of nodes. The nodes are put together in groups recursively. This systematization is built on top of routing tables allows effective routing in Peer network. The DST is an overlay structure designed to be scalable [14]. The DST is a tree without bottlenecks which automatically balances the load between its nodes. So we virtually convert the Peer network into DST and each tree should have its root node we call it as Head Node (HN) and others are Leaf Node (LN). Every LN will hold the details of its own HN. Likewise every HN will hold the complete details regarding its LNs and all other HNs in the network. The details stored in HNs and LNs in the DST is used to enhance the efficient routing with minimum message pass. During the formulation of DST in Peer network, LNs and HNs are chosen randomly and dynamically with some requirement criteria which improve the Fault Tolerance of the system.

### *B. Global Replica Management (GRM)*

A detailed mechanism for GRM using DST as a communication structure in peer network can be found in [1]. In Peer network, GRM can be accomplished efficiently by using algorithm consists of 3 phases of three rounds of message exchanges.

*a. Read Operation*

Phase 1: The Read requested peer send the Read Lock request it's HN which is forwarded to all LNs.

Phase 2: The HN send the requested peer wait message or latest copy of replica based on its status.

Phase 3: After receiving the latest replica the read requested peer will send the Read Unlock request to its HN which is forwarded to all LNs.

*b. Write Operation*

Phase 1: The Write requested peer send the Write Lock request it's HN with updated data which is forwarded to all LNs.

Phase 2: The HN send the requested peer wait message or update the replica copy with the received updated copy of replica based on its status.

Phase 3: After receiving the latest replica the read requested peer will send the Write Unlock request to its HN which is forwarded to all LNs.

These procedures are executed in distributed fashion among the peers in network which are engaged in GRM. Thus each peer performs the efficient replica management globally using DST as interconnection technique

## III. PROPOSED ANT COLONY OPTIMIZATION OF DST MODELED PEER NETWORK

ACO is a probabilistic technique which search for optimal path in a graph, which is based on behaviors of ants seeking a path between their colony and source of food. So, by applying the ACO on the DST, we can obtain optimal path for message pass among the peers in the network. ACO is also capable to reform a new optimal path in case of any problem with the current optimal path which improve the effective routing in real-time. In this paper the Ant Colony Optimization Algorithm has been modified and proposed for finding an optimal path in DST of the peer network.

The Ant Colony Optimization Algorithm for DST peer network is presented as in Fig.1. This algorithm uses four procedures. Procedure *Optimize()* is the entry point for the algorithm. It takes the Peer Network as Graph G as its parameter and calls other procedures *transmit(), formation(), daemonaction()* on some criteria.

Procedure *Optimize()* performs two operations, to find the dynamic optimal path between every HN and between HN and its every LNs.

| | | |
|---|---|---|
| **transmit()** | | |
| | Step 1: | $\forall$e such that G(v,e) $\in$ {G(v) – v} promote p to G(v,e) |
| **formation()** | | |
| | Step 1: | Initialize val(x), trial(x) as $\varnothing$ |
| | Step 2: | Let v'(G) is a set of non-visited vertices in between v and x. Choose y such that y $\in$ v'(G) |
| | Step 3: | val(x) $\longrightarrow$ val(x) + $\tau_{(v,y)}$ <br> trial(x) $\longrightarrow$ trial(x) $\cup$ y and set y as visited |
| | Step 4: | If v'(G) ≠ $\varnothing$ , goto **step 2**. |
| **DaemonAction()** | | |
| | Step 1: | $val(x)_i \longrightarrow$ val |
| | Step 2: | If val(x) < $val(x)_i$ <br> val(x) $\longrightarrow val(x)_i$ <br> Path(v,x) $\longrightarrow$ v, trail(x), x <br> Path(trail(x),x) $\longrightarrow$ trail(x) - 1, x |
| | Step 3: | inform and send Path(v,x) to v. |
| **Optimize()** | | |
| | Step 1: | Let G is a graph of vertices v and edges e such that G is given as G(v,e). <br> $\forall$ v in G such that v is an HN, v $\in$ {$HN_1$, $HN_2$…$HN_n$} where 'n' is the number HN in the Network. |
| | Step 2: | $\forall$ x such that x is HN other than v, x $\in$ {{$HN_1$, $HN_2$…$HN_n$} – v} repeat Step 3. |
| | Step 3: | Create Probe message p on v, <br> Invoke transmit() by v <br> Invoke formation() by v <br> Invoke DaemonAction() by x |

*Figure 1. Ant Colony Optimization Algorithm for DST peer network*

*Operation 1:* Let $HN_i$ is a HN among {$HN_1$, $HN_2$...$HN_n$} where 'n' is the number HNs in the Network. $HN_i$ use Probe message *p* to find optimal path between $HN_i$ and other HNs. First procedure *transmit()* is called which takes Graph G, $HN_i$ as v and probe message p as parameters. Probe *p* is flooded through all the possible paths from $HN_i$. Then procedure *formation()* is called by $HN_i$ which takes Graph *G*, start HN *v*, specific end HN *x* and the measure concerned with edge between each Peers along the way between *v* and *x*, $\tau$ whose value is used to decide the optimal path, usually $\tau$ will be Hop count, Congestion occurrence, Propagation delay etc.

Procedure *DaemonAction(val)* is called by end HN *x*, which takes '*val*' as a parameter and decide the optimal path between the HNs *v* and *x* based on the value of $\tau$ along the path of each probe *p*. Every probe reaches *x* with its '*val*' then *x* decides the optimal path based on the '*val*' and the component type of *τ*. HN x inform the identified optimal path to the HN *v*. So, the operation of finding an optimal path between HNs has been successfully done.

*Operation 2:* In this operation optimal path between each HN and their LNs is identified.

As said in operation 1, operation 2 is carried on such that the value of τ counted along the path between HN and all its LNs. Number of optimal paths identified in operation 2 is given as,

$$n(paths) = \sum_{i=0}^{p} \sum_{j=0}^{q} \sum_{k=0}^{r} (LN_{ijk}) \qquad -(1)$$

where,

- *'p'* is number of HNs in the Network.
- '*q*' is number of LNs for $i^{th}$ HN in the Network
- *'r'* is number of Peers that replicate the probe
- '$LN_{ijk}$' is $j^{th}$ LN for $i^{th}$ HN through $k^{th}$ peer in the Network

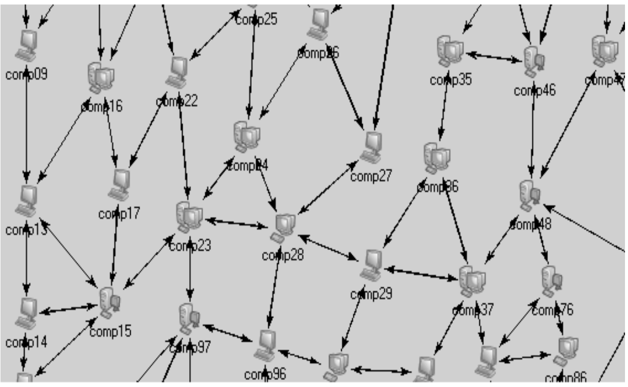


*Figure 2. Simulated Peer Network in OMNeT++ Simulator*

From the available optimal path $n(paths)$ between the HN and all its LNs, a best path has been chosen for the message pass. Communication through the best (optimal) path found using ACO needs low cost and improve overall efficiency of the GRM.

*Definition 1:* Let $n(ACO_{msg})$ be the total number of messages required to ACO optimize DST which is defined as,

$$n(ACO_{msg}) = (1+2+....+p)*s + \sum_{i=0}^{p} q*n(s) \qquad -(2)$$

where,

- 'p' is number of HNs in the Network.
- 'q' is number of LNs for $i^{th}$ HN in the Network
- 's' is number of Peers that replicate the probe
- 'n(s)' is the number of Peer that each 's' has within its range.

## IV. SIMULATION AND ANALYSIS

This section illustrates the simulation results obtained during the analysis and we used OMNeT++, which is an object-oriented modular discrete event network simulator. We simulated a peer network by placing 100 nodes randomly within a region as shown in Fig.2 and propagation delay is set as 100ms.

To route message from source to destination Ant Colony Optimized routing technique is followed which improve the message pass efficiency of DST because of dynamically identified optimal route between every HN and LN Peers. As explained in the previous sections, formulation of ACO optimization of DST is happen as the next step of the network after the formulation of DST, which is achieved by sending Probe Request messages from HN to *LNs and HN to all other HNs.* Fig.3 shows the Number of messages required by each peer to ACO optimizes DST.

On simulation proceeds, it is observed that ACO optimized network perform nearly 181 read operations are performed by nearly 56 Peers in 600 seconds whereas mere DST network can perform 121 read operations are performed by nearly 49 Peers at same time. Thus by using DST as a interconnection structure and ACO as routing optimization technique we are reducing the message pass for Read Operation at high level which makes the operation fast and consistent. Fig.4 shows Number of Read Operations performed by Peers on ACO optimized DST.

By collecting data of Write Operation in simulation, we found that 25 Write Operations were performed by 18 Peers in a time of 30 seconds.Fig.5 shows number of Write Operations performed by Peers on ACO optimized DST.

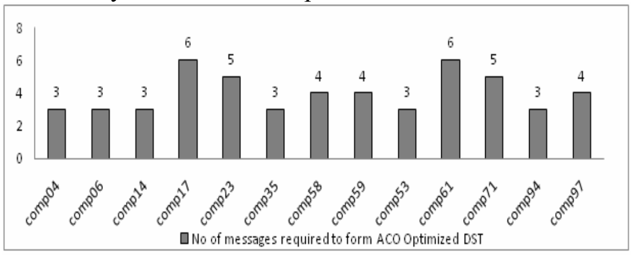


*Figure 3. Number of messages required by each peer to form ACO optimized DST*

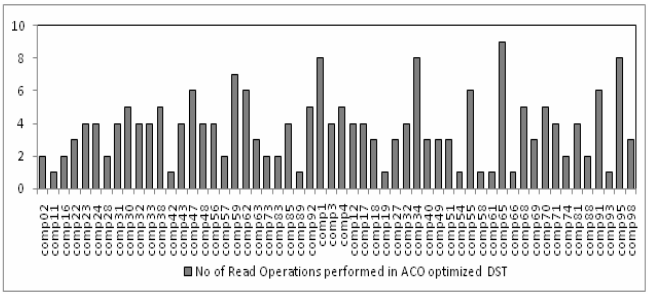


*Figure 4. Number of Read operations performed by each peers in ACO optimized DST*

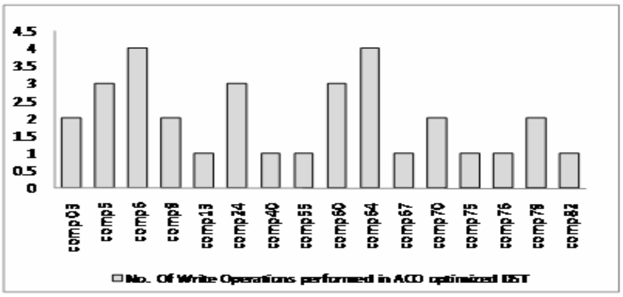


*Figure 5. Number of Write operations performed by each peers in ACO optimized DST*

From the result analysis, it is clear that ACO optimized DST can perform read/write operations at higher speed and consistent. On analyzing the various criteria measurements we obtained from simulation we conclude that ACO Optimized DST can perform nearly 45% faster than mere DST network.

Our simulation is made with channel of propagation delay of 100ms. By decreasing the delay we can improve the throughput of our GRM technique.

## V. CONCLUSION AND FUTURE WORK

The work we presented in this paper has described the Ant colony Optimization of Global Replica Management (GRM) technique in peer network using DST Model. We have proved that the proposed model will improve the performance of GRM application in terms of various parameters like time taken to read the latest replica, to update the latest copy of replica and number of message passes required for operation. From the simulation analysis, it is shown that by employing ACO applied DST in Peer network, with cost of only some message pass we can promise improved GRM technique can be achieved in a network. This work can also be extended for many other replica management types as like local, time, off-line and on-line in future.

## VI. REFERENCES


[1] P. Victer Paul, T.Vengattaraman, M.S. Saleem Basha, P. Dhavachelvan, "Distributed Spanning Trees for Effective Replica Management", IEEE International Conference on Advances in Communication, Network, and Computing – CNC 2010.

[2] Xin Sun, Jun Zheng, Qiongxin Liu, Yushu Liu, Dynamic Data Replication Based on Access Cost in Distributed Systems, 2009 Fourth International Conference on Computer Sciences and Convergence Information Technology, IEEE 2009

[3] Stockinger H, Samar A, Allcock B, Foster I, Holtman K, Tierney B (2002) File and object replication in data grids. J Cluster Comput 5(3):305–314

[4] Chervenak A, Foster I, Kesselman C, Salisbury C, Tuecke S (2001) The data grid: towards an architecture for the distributed management and analysis of large scientific datasets. J Netw Comput Appl 23:187–200

[5] A.A. Helal, A.A. Heddaya, and B.B. Bhargava, Replication Techniques in Distributed Systems. Kluwer Academic, 1996, ISBN:0-7923-9800-9.

[6] B.Ciciani, D.M. Dias, and P.S. Yu, "Analysis of Replication in Distributed Database Systems," IEEE Trans. Knowledge and Data Eng., vol. 2, no. 2, pp. 247-261, June 1990.

[7] S.-Y. Hwang, K.K.S. Lee, and Y.H. Chin, "Data Replication in a Distributed System: A Performance Study," Proc. Conf. Database and Expert Systems Applications, pp. 708-717, 1996

[8] Frank Neumann, CarstenWitt,"Runtime Analysis of a Simple Ant Colony Optimization Algorithm", Springer Science+Business Media, LLC 2007.

[9] Dorigo, M., Maniezzo, V., Colorni, A.: "The ant system: An autocatalytic optimizing Process". Tech.Rep. 91-016 Revised, Politecnico di Milano, Italy (1991).

[10] A. Colorni, M. Dorigo, V. Maniezzo (1991)," Distributed optimization by ant colonies",In Proceedings of ECAL'91 European Conference on Artificial Life, Elsevier Publishing,Amsterdam, The Netherlands, pp 134-142.

[11] M. Dorigo, V. Maniezzo, A. Colorni (1991),"The ant system: an autocatalytic optimizing process", Technical Report TR91-016, Politecnico di Milano.

[12] Sylvain Dahan, Jean-Marc Nicod and Laurent Philippe, The Distributed Spanning Tree ,IEEE transactions on parallel and distributed systems, vol. 20, no. 12, december 2009

[13] A. Rowstron and P. Druschel, "Pastry: Scalable, Decentralized Object Location, and Routing for Large-Scale Peer-to-Peer Systems," Proc. IFIP/ACM Int'l Conf. Distributed Systems Platforms, vol. 2218, pp. 329-350, 2001.

[14] Sylvain Dahan, Distributed Spanning Tree Algorithms for Large Scale Traversals, 11th International Conference on Parallel and Distributed Systems (ICPADS'05), 2005 IEEE.