4

Proposed Algorithm for Grid Resource Discovery

4.1 Introduction

Resource discovery in the computational grids is a challenging problem because of the dynamic nature of user requests as well as dynamic availability of resources and their heterogeneous nature [104]. Growing size of grid makes it essential for the grid resource discovery mechanisms to be highly scalable and de-centralized [105]. P2P resource discovery systems [106][107] provide a good example of de-centralized resource discovery. Software mobile agents [115][116], with their autonomous behavior, intelligent decision making capability and mobility are particularly suitable for tasks such a de-centralized resource discovery. At the same time ants (natural mobile agents) based algorithm provide a mechanism for building self-adaptive systems. This paper presents ants colony optimization based algorithm, which is used to guide software mobile agents in efficient resource discovery.

Many resource discovery mechanisms have been proposed in the literature of Grid environments. Some of them, such as MDS2 (Meta-computing Directory Service) [108] and Data Grid Resource Discovery [109], are specific resource discovery services. However, the great majority of them are contained in more general grid proposals such as MDS3 (Monitoring and Discovery Services) [110], NWS (Network Weather Service) [111], VIRD (Vega Infrastructure for Resource Discovery) [112], and Nimrod/G [59].

The resource discovery mechanisms proposed for grids are either hierarchical or centralized. Most of these mechanisms retrieve data related to the machines that compose the grid (operating system used, CPU load, and memory occupation, among others). But for large scale distributed environment like grid the centralized approach is not very much suitable. Hence a de-centralized approach is desirable [112].

As a solution to this problem an ACO based algorithm has been proposed. Mobile agents are used to simulate this ant like natural agents. This further adds the capabilities of mobile agents like distributed and asynchronous processing etc. to this approach. The terms ants, mobile agents, query agents and natural agents are used interchangeably in this thesis.

4.2 Problem formulation

To formulate the problem we consider the whole grid as a network of virtual organizations (VOs) [1] with each VO having a designated contact node. Connection between the VOs is via contact nodes only as shown in Figure 4.1).

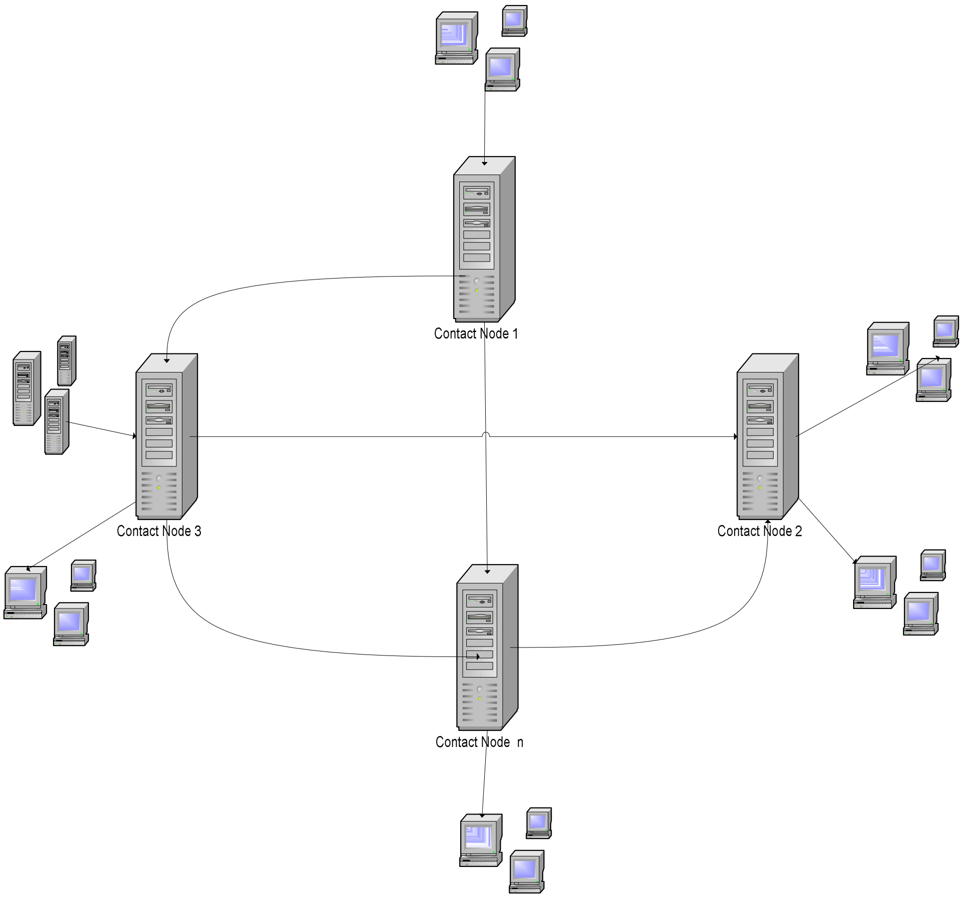

Figure 4.1: High level view with Peer Contact Nodes

The subnet of contact nodes can be represented by a directed graph. Machines/ contact nodes are represented by nodes in the graph and an edge between two nodes represent a relation between a pair of machines / contact nodes (an edge between nodes *A* and *B*, means machine *A* has knowledge of node *B*). Communication can only happen between the machines that know each other, rather than some broadcast message from central machine to which every machine responds. The problem is to find the set of matching resources that can full fill the request. The resource request is specified in the form of resource attributes / characteristics that need to be matched. The set of resource attributes that is used to find matching resources is also known as resource query.

4.3 Mobile Agents

Software agent is a computing entity that performs some task or tasks on behalf of somebody [113]-[115]. A mobile agent is a software agent that has an additional property that it is not bound to operate only in the system in which it started. One special thing about mobile agent is that during its lifetime it can be stopped at one point, its state can be saved and code can be moved to another machine and then can resume its execution. Mobile agents have many benefits over other technologies [116].

Mobile agents can reduce the need for bandwidth. They can package the interaction between two machines as a discrete packet of network traffic and helps to make all interactions locally rather than having all individual interactions over the network [117]. Hence MAs reduce the network traffic by moving the processing to raw data instead of moving raw data to processing [118]-[121]. Also they can grow dynamically to accommodate more data. Mobile agents can be dispatched asynchronously. Autonomy in the MAs enables them to take dynamic decisions based on the previous learning. MAs can be easily cloned and dispatched too and finally, mobile agent based solution are very fault tolerant. If some MA gets destroyed in between, the results from other surviving MAs can still be used. So they can be used to benefit or to simplify different types of application areas. Some examples of these application areas include e-commerce, distributed information retrieval, telecommunication networks services, and monitoring and notification [117].

4.4 Proposed Algorithm for Resource Discovery based on ACO guided Mobile Agents (ACORD)

This section discusses the ACO guided mobile agents based proposed resource discovery algorithm for computational grids. Ants are simulated using Mobile agents and pheromone trails laid down by ants are used to guide the mobile agents in their search. This provides the basis for efficient and de-centralized resource discovery mechanism for computational grids. Different components of this algorithm are described below.

4.4.1 Fitness function

Fitness function is used to differentiate between high and low quality solutions. It is defined in terms of objectives. For resource discovery our objective is to find the set of resources matching to the query made by the grid resource management system. So we define our fitness function as the ratio of resources found to the resources requested.

Let Q denotes the query, i.e. the set of the resources requested, and M_N is the set of matching resources found till Node N. Then the fitness function, $F_N(Q)$, at any Node N is defined as

$$F_N(Q) = \xi(M_N) / \xi(Q) \tag{4.1}$$

where $\xi(M_N)$ denotes the resources in Set M_N

As the ants move from one node to another, the fitness function, $F_N(Q)$, is evaluated at every node N, pheromone is deposited based on the fitness and Ants move to the next node till the exit criteria is reached.

4.4.2 Pheromone Trail Definition

In ACO based algorithms ant's search is highly influenced by the pheromone trails, $\tau(N)$, left by other ants. So the definition of pheromone trail is one of the important decisions in ACO algorithms. Since, grid is highly distributed and heterogeneous environment, so different nodes may have different availability of resources. So the pheromone trail, $\tau(N)$ will be representing the Query Hits along the path of node N.

4.4.3 Heuristic Information

Heuristic information, $\eta\ (N)$, in the ACO algorithms is another factor that affects the move of an ant. This is the piece of information other than the pheromone trail that decides the probability of the node to be chosen next. In our problem we define this as the number of resources available at node N. Hence

$$\eta\ (N) = \xi(R_N) \tag{4.2}$$

where R_N is the set of available resources at node N.

$\eta\ (N)$ is normalized with respect to the availability of resources in the neighborhood of node N.

4.4.4 Pheromone Trail Updating

After every successful search pheromone trail needs to be updated along the search path so as help other ants to find resource quickly. The amount of pheromone to be deposited at any node N is defined as

$$\tau(N) = \rho\,.\ \tau(N) + F_N(Q\ ) / F_{end}(Q) \tag{4.3}$$

where $F_N(Q\ )$ is the fitness at Node N for query Q and $F_{end}(Q)$ is the fitness at the last node in the search Path and ρ is the rate of pheromone evaporation.

4.4.5 Building the Solution

A specified number of ants are initialized and start building their tours with respect to the query Q. At any Node I, the probability of moving to the next node K is given by Equation 4.4.

$$Prob(K) = \frac{[\tau(K)]^{\alpha} \cdot [\eta(K)]^{\beta}}{\sum^{n}_{i=1} [\tau(i)]^{\alpha} \cdot [\eta(i)]^{\beta}} \qquad (4.4)$$

where n is the number of nodes in the forwarding table of Node I.

New node K is selected according to the probability calculated using the equation 4.4 and ant is moved to next node. This process is repeated till the fitness at newly selected node exceeds the threshold value ("query hit"). As soon as the ant reaches at the node where there is query hit, the ant performs the backwards journey from final / end node to the source node and on its way back, it keeps on updating the pheromone trail on all the nodes in its path. On the other hand if the ant has visited maximum nodes, it simply returns to the source node with a "query miss" without updating the pheromone trails.

4.4.6 Exit Criteria

An exit criterion specifies when to stop further exploration. Proposed algorithm uses two exit criteria. 1) If the fitness at any node I exceeds fitness threshold, which is a "query hit" or 2) If ant has visited maximum number of nodes as specified, this is a "query miss". In either of the cases further exploration is stopped.

4.5 Algorithm to be executed by Query Agent

During the process of resource discovery, a Query Agent goes through different states. States diagram shown in Figure 4.2 show the transition of Query agent to different states.

I. Query Agents are created by a contact node in VO and initialized with the query according to which they have to search for resources. Query is specified in terms of resource characteristics.

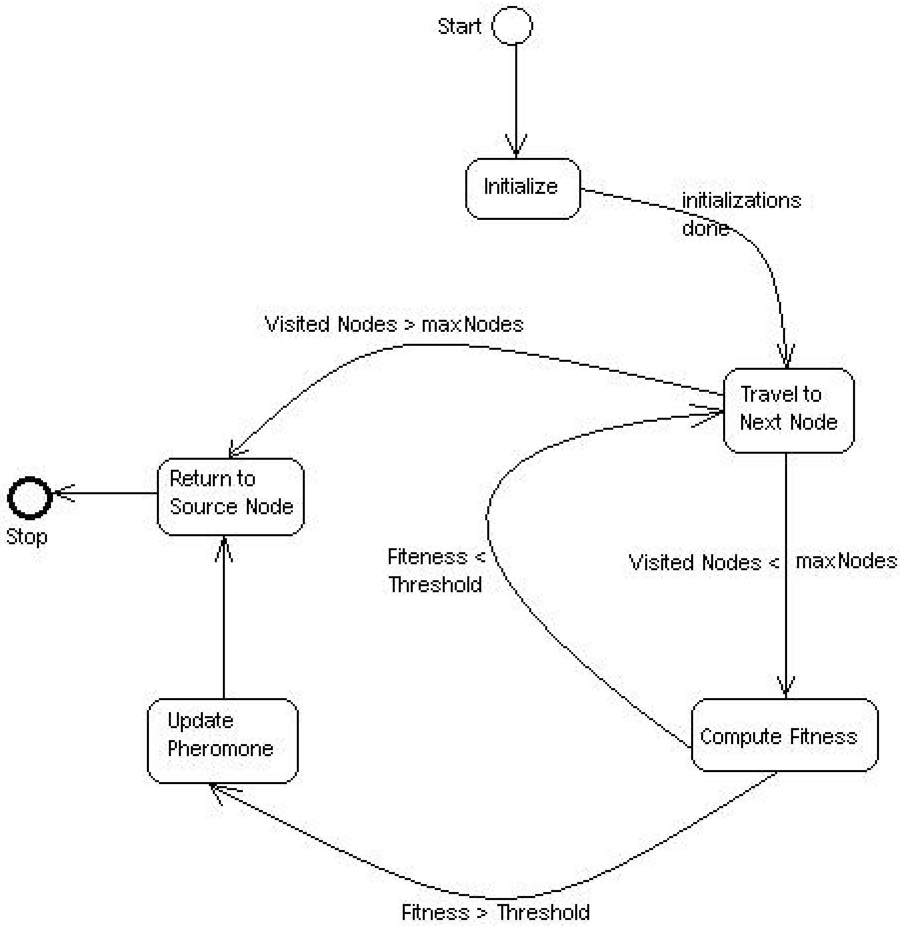

Figure 4.2: Different States of a Query Agent

II. *Branching factor* limits the number of query agents to be created for a specific query.

III. QA is also initialized with *MaxNodes* that it can travel at most. *MaxNodes* is the maximum number of nodes a QA is allowed to visit. It ensures that a QA should return in finite time to source contact node.

IV. After arriving at any node it increments the *Visited Nodes* count, check for the resources at the current node and computes the fitness and accordingly take the decision to move to next node or return to source node as per the algorithm mentioned in the previous section.

V. After submitting the results to the source contact node (the creator of QA) it dies.

VI. Source contact node collects results (resource characteristics) from all the QAs dispatched, and allocate resources to the different jobs as per the requirement.

VII. Pheromone trails are also evaporated after regular interval of time to avoid saturation.

4.5.1 Algorithms (Pseudo Code)

```
// Initialize Variables
initialize BranchingFactor
initialize MaxNodes
initialize Threshold

// Create Query Agent Array
QueryAgentList QueryAgent[BranchingFactor]

// Initialize Query Agents
for ( i = 0; i < BranchingFactor; i++ )
{
     QueryAgent[i].initialize();
}

Repeat following steps for Every QueryAgent[i]
{
     Repeat at every Node
     {
          //Increment VisitedNodes
          QueryAgent[i].visitedNodes += 1;

          if ( QueryAgent[i].visitedNodes <= MaxNodes )
          {
               // Compute Fitness
               fitness=QueryAgent[i].ComputeFitness(Node);

               if ( fitness < Threshold )
               {
                    // Find Next Node.
                    NextNode=QueryAgent[i].findNode(Node);

                    // Travel to next Node
                    NextNode.Move(QueryAgent[i]);
               }
               else
               {
                    // Update Pheromone Trails.
                    QueryAgent[i].UpdatePheromone();
```

```
                    // Return to source Node. (Query Hit)
                    QueryAgent[i].returnToSource(HIT);

               }
          }
          else
          {
               // Return to Source Node (Query Miss)
               QueryAgent[i].returnToSource(MISS);

          }

     } // End Repeat at every node
}
```

4.6 Advantages of proposed ACO based Resource Discovery Algorithm (ACORD)

Proposed ACO based resource discovery algorithm (ACORD) provides several advantages over traditional grid resource discovery system. Some of them are list below.

De-Centralized System: Proposed algorithm provides a resource discovery model which employs the benefits of a peer to peer system and, hence, is de-centralized system. A grid is considered as a peer to peer network of virtual organizations (VOs) where the contact nodes are allowed to communicate with other peer contact nodes.

Scalability: As mentioned above, proposed system is considers grid as a collaboration of different virtual organizations. This can be viewed as graph of contact nodes. With the de-centralized control it can easily cope with the increased number nodes / addition of new virtual organizations.

Parallel resource discovery: More than one QA are generally dispatched corresponding to a query and all search in parallel through different paths in the network for matching resources.

Fault Tolerant: Since more than one QA are usually dispatched, so if some QA is crashed in between, the results from other QA can provide the solution without affecting the quality.

ACO guided search: Proposed model provides ACO guided search. QAs from previous searches deposit pheromone trails that guide next QAs to find the fruitful nodes.

4.7 Summary

Details of the proposed ACO guided mobile agent based resource discovery algorithm are provided in this chapter. This algorithm assumes a model where each virtual organization has a designated contact node and one or more virtual organization interact with each other to find a set of matching resources via contact nodes. Mobile agent contains the query and where as pheromone trails laid down by ants provides the learning and aid in quick and efficient resource discovery.