ISSN:0975-9646

Fault Recovery in Optimal Task Scheduling and Grid Service Reliability

Ashwini Vedulla, Pramod Reddy A, Krishna Prasad B

Department of Computer Science and Engineering
Bharat Institute of Engineering and Technology
Hyderabad, India

ABSTRACT--- There has been quite some research on the development of tools and techniques for grid systems, yet some important issues, e.g., grid service reliability and task scheduling in the grid, have not been sufficiently studied. This paper examines methods for providing fault-tolerant real-time communication in wide area networks and their related resources and works. An algorithm is introduced for selecting a recovery node which minimizes the requirements while maintaining negotiated traffic characteristics. The End-to-End Recovery source node gives the most **efficient use of resources since the backup path is chosen overall possible paths between source and destination with no cycles. Based on the proposed grid service reliability model, a multi-objective task scheduling optimization model is presented, and an ant colony optimization (ACO) algorithm is developed to solve it effectively. A numerical example is given to illustrate the influence of fault recovery on grid service reliability, and show a high efficiency of ACO in solving the grid task scheduling problem.**

K*ey Words --- Ant colony optimization, fault recovery, task scheduling.*

I. INTRODUCTION

The rapid advancement's in communication technology has changed the landscape of computing. New models of computing, such as business-on-demand, Web services, peer-to-peer networks, and Grid computing have emerged to harness distributed computing and network resources to provide powerful services. The non-deterministic characteristic of the resource availability in these new computing platforms raises an outstanding challenge: how to support Quality of Service (QoS) to meet a user's demand? In this study, we conduct a thorough study of QoS of distributed computing, especially on Grid computing where the requirement of distributed sharing and coordination goes to the extreme. We start at QoS policies, and then focus on technical issues of the enforcement of the policies and performance optimization under each policy. This study provides a classification of existing software system based on their underlying policies, a systematic understanding of QoS, and a framework for QoS of Grid computing.

GRID computing has become apparent as the next-generation parallel and distributed computing methodology. Its instance is to provide a service-oriented infrastructure to enable easy access to and coordinated sharing of geographically distributed resources for solving various kinds of large-scale parallel applications in the wide area network. Now a days, grid computing has been widely accepted, study, and given attention to by researchers [1]. Unlike the traditional file exchange, as supported by the Web or peer-to-peer systems, users in the grid can access the required resource or service in a transparent way as if they were to use local resources or services. However, it gives rise to any of two or more ideas conflict between grid users and resource providers in usage policy of the local resources. For users, in addition to simplicity and easiness, to get desirable service functionalities, some quality of service (QoS) targets associated with the service, such as grid service reliability [2], the financial cost of the resource, and the efficiency of grid service, may be specified when a service is submitted. On the other hand, resource providers receive the compensation from grid users for the consumed resources at the price of sacrificing local task executions [3]. Meanwhile, resource providers may not participate in the grid unconditionally, and they may specify different policies that govern how the resources should be used by the grid such that the resources could still meet the local resource demands [4], [5]. On behalf of grid users with multiple dimensional QoS requirements, multi-objective task scheduling with a set of resource constraints should be solved to obtain satisfied scheduling decisions. Moreover, the likelihood of errors occurring may be exacerbated by the fact that many grid services will perform long tasks that may require several days of computation [10]. Recently, much effort in fault avoidance and fault removal has been invested so as to improve grid service reliability

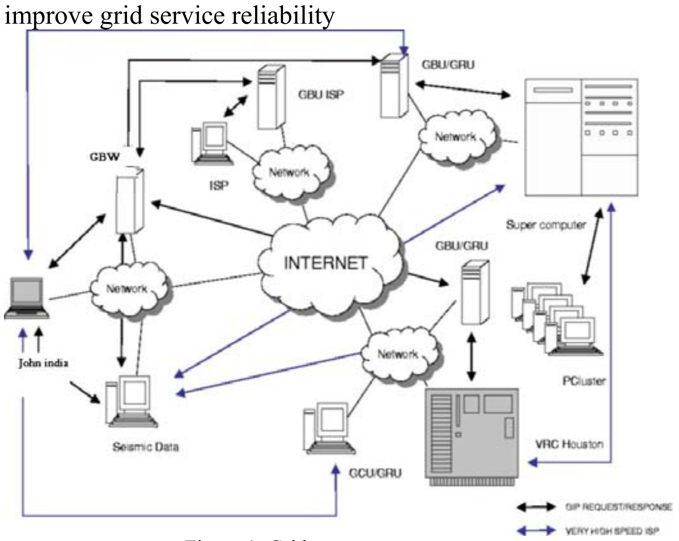

Figure 1: Grid resource manager

The end-to-end fault recovery is introduced. Moreover, because fault recovery modules are located at grid resources, resource providers can set customizable constraints on fault recovery, which makes it easy to achieve distributed management of fault tolerance. Therefore, End-to-End Fault recovery can be used to achieve effective fault tolerance in a grid environment.

II. Reliability Modeling and Analysis with Fault Recovery

A. End-to-End Fault Recovery

Recovery from the source node gives the most efficient use of resources since the backup paths is chosen over all possible paths between source and destination with no cycles. The backup paths are established in this manner. The backup channel does not become active until the source detects the fault. The fault detection time a real-time channel, τ is established between the destination and source to transmit notification packets.

The recovery time is bounded by the transmission time on the real-time channel used to transmit fault notification.

$$WCRT = \sum_{i=n}^{1} l_i + \sum_{i=f}^{1} d_{i,\rho}$$

In case where the fault occurs close to the destination, the time to notify the source node of the fault may be large relative to the time between packets ,thus causing one or more packets to be sent on the failed channel. These packets will need to be resend on the failed channel and these packets will experience an additional delay equal to the time they have already spent in the network.

Thefore the worst case delay is:

$$D_e = \sum_{i=n}^{n} l_i + \sum_{i=1}^{f} d_{i,\psi}$$

The source in the case where zero packet loss has been specified in channel establishment request.

$$B_{e \Leftrightarrow e} = [\frac{\sum_{i=n}^{n} 2l_i + \sum_{i=1}^{f} (d_{i,\psi} + d_{i,\rho})}{T_\psi}] - P_{\max,\psi}$$

End –to-end recovery insures that the least cost path is chosen for recovery but has the worst case delay and packets resent. Since recovery must be guaranteed even from failure at the node/link nearest the destination, recovery by this method can not succeed unless D_ψ is greater than triple the propagation delay between the source and destination. The high delay makes end-to-end recovery unlikely to succeed for much wide area real-time communication applications. This paper is organized as follows. Section II represents reliability modeling and analysis with fault recovery. To be more practical, the optimal task scheduling are presented in Section III. Section IV presents a multi-objective task scheduling model. An ACO-based algorithm is developed to solve this problem with a numerical example to show the positive influence of fault recovery and effectiveness of our algorithms. At last it concludes this paper, and indicates some directions for future study.

Theorem 1: If either local or end-to-end recovery are able to find a backup path, then the hybrid algorithm will also find a path.

Proof: Assume end-to-end recovery is able to find a recovery path ψ_c Then at the link from the source node S, $P_{act,S\psi} \leq P_{\max} + B_s$ must be true(otherwise no guarantee can be given that $P_{\max}$ will not be violated). The hybrid algorithm will attempt to establish a least cost path ψ_r ,form the dource to the destination given these conditions therefore $\psi_c == \psi_r$ and if ψ_c satisfies the source requriments ψ_r must also.

Assume local recovey is able to find a recovery path ψ_l .Let r be the node prior to the failes link/node on ψ . ψ_l is made up of last cost path from node r to the destination which does not pass through the failed link/node. The hybrid algorithm will attempt to establish a least cost path at some node , η on the path of ψ between the source and node r. the hybrid recovery path ψ_r will be made up of the least cost path from the source to η and the least cost path form η to the destination.

If ψ_l is the least cost path between the source and destination $\psi_r == \psi_l$ since both paths consist of the least cost path from source η to r and the least cost path r to the destination. If the lower cost path exists between η and the destination which does not include r, then the orginal channel ψ can not be a least cost path which is a contradiction.

If ψ_l is the least cost path between the source and destinaiton, then the cost of $\psi_r < \psi_l$, since ψ_r would choose ψ_l as its path if a lower cost were not availible . Since the cost of the path $\psi_r < \psi_l$ if ψ_l is a valid path then so is ψ_r .

B. Reliability Modeling and Analysis with Fault Recovery

Denote by M the number of divided subtasks by the RMS after the RMS receives a service 'S'. Assuming that subtask *i* is assigned to node K, the required processing time of subtask on node K, ξ_{ik} is [2], [6]

$$\xi_{ik} = C_i / S_k \qquad (1)$$

With the introduction of end-to-end fault recovery, an operational state in grid nodes is further classified into executing or recovering. Are covering state means a state where a grid node can be operational but not quite ready to be accessed, as it performs recovery procedures. Denote by N_{ik} the total number of recoverable failures occurring during the execution of subtask *I* on node *k.*

N_{ik} is a random variable. If N_{ik} =n(n>=1) Then denote by $TE_{ik}^{(j)}$ (j=1,2,L,n,n+1) and $TR_{ik}^{(j)}$ (j=1,2,L,n,) the executing times, and recovering times in the execution process of subtask onnode ,respectively.The subtask execution process goes on until the subtask is successfully completed, or it is terminated by an unrecoverable failure. In the former case, the total execution time of subtask *I* on node *k* is ξ_{ik} which can be obtained by (1); in the latter case, the subtask is terminated.

$$\text{TEjk}=\sum_{i=1}^{n+1} TE_{jk}^{(j)} \text{ and } TR_{ik} = \sum_{i=1}^{n} TR_{ik}^{(j)}$$

The lifetime of subtask *I* executed on node k, Tik is

Tik=TEik+TRik (2)

If subtask is successfully completed on node , then its lifetime is

$$Tik=\xi_{ik}+TR_{ik} \qquad (3)$$

Where ξ_{ik} is obtained by (1).

III. Optimal Task Scheduling

After the grid service is divided into some subtasks, the RMS should quickly and effectively schedule those subtasks to the appropriate nodes according to the particular requirements of those subtasks, and the QoS demands of grid users. In the scheduling, it needs to take into account not only the hard constraints of a subtask (the operating system type, available CPU, memory, disk space, etc.), but also software constraints such as the demanded reliability level of grid service, and the constraints on total financial cost. Moreover, with the introduction of End-to-End fault recovery there should be many other factors that may also be taken into account, such as the recoverability of grid nodes, the allowed number of recovery performed, and the life time of subtask execution in located nodes. In this section, a multi-objective task scheduling model, minimizing cost and maximizing reliability, is presented and a search algorithm based on ACO is proposed.

A. Problem Formulation

We consider the optimization problem of task scheduling with the following scenarios.

1) The nodes involved in the grid are heterogeneous. Hence, the nodes may be capacitated with various units of memory and computation resources, and they may have different processing speeds and failure rates. Also, the communication links may have different bandwidths and failure rates.

2) The cost of subtask execution in grid nodes is mainly dependent upon the execution time, and the charging of resource providers per unit time.

3) The nodes involved in the grid may have the different failure recoverability, and the constraints on the life times of subtasks and the numbers of recoveries performed may be different as well.

4) To simplify the problem formulation, one subtask is allowed to be assigned at one node, and one node can only be allowed to execute one subtask at most.

B. Ant Colony Optimization

Ant colony optimization (ACO) is a meta-heuristic optimization technique inspired by research on real ant foraging behavior. The ACO paradigm was first proposed by Dorigo and Blum [7], and has been successfully applied to diverse combinatorial optimization problems including traveling salesman [7], telecommunication networks [8], and scheduling [9].

1) Construction of Solutions: A permutation of subtask scheduling subject to the constraint is termed a feasible solution. In the algorithm, the process of constructing a feasible solution can be divided into *m* steps. In the *ith* *(0<i<m)* step, ant *j* finds one accessible node for subtask *i*, and then moves to the *i+1st* step for subtask *i+1* scheduling until all the subtasks have been assigned, which can guarantee all the subtasks are scheduled as stated in (8). In the scheduling, all infeasible moves of ant *j* must be stored into a T abu list denoted by T abu_j This list is the memory of ant *j* saving the index of infeasible allocations, and is initialized according to the hard constraint, as shown in (10).

2) Selection Probability: In each step, an ant in a subtask chooses a node as the location of the corresponding subtask. This move can be represented by edge ***(i,k)*** which is the assignment of subtask *I* to node *k* for traversing by the probability

$$P_{ik} = \frac{\propto \tau_{ik} + (1-\alpha)/\eta_{ik}}{\sum_{X \notin Tabu_j}^{n} [\alpha \ \tau_{is} + (1-\alpha)\eta_{i\sigma}]}$$

Where τ_{ik} is the quantity of pheromone on the edge *(i,k)* and η_{ik} is the desirability of assigning subtask *I* to node *k* which is written as

$$\eta_{ik} = C_{ik} + \chi / R_{ik}$$

Where C_{ik} is execution cost of sub_i in Node K and R_{ik} is the reliability of sub_i in node K , χ is the scaling factor.

3) Pheromone Updating Rule: Pheromone updating is a process of changing the quantity of pheromone over time on each edge. Before activating the next iteration, the quantity of pheromone on each edge is updated by the pheromone updating rule so as to avoid local convergence, and explore more search space as well. According to the original ACO algorithm [8] in the single objective function problems, the pheromone updating rule on each edge is

$$\tau_{ik}' = (1-\rho)\tau_{ik} + \Delta\tau_{ik}$$

Where τ_{ik}' is the updated quantity of pheromone on the edge *(I,k) and(0<* ρ *<1)* is the evaporation rate of pheromone.

ACO Algorithm

```
♦Initialization:
 a. Set initial parameters that are system: variable, states, function, input, output, input trajectory, output trajectory.
 b. Set initial pheromone trails value.
 c. Each ant is individually placed on initial state with empty memory.
♦While termination conditions not meet do
 a. Construct Ant Solution:
 Each ant constructs a path by successively applying the transition function the probability of moving from state to state depend on: as the attractiveness of the move, and the trail level of the move.
 b. Apply Local Search
 c. Best Tour check: If there is an improvement, update it.
 d. Update Trails:
 - Evaporate a fixed proportion of the pheromone on each road.
 - For each ant perform the "ant-cycle" pheromone update.
 - Reinforce the best tour with a set number of "elitist ants" performing the "ant-cycle"
 d. Create a new population by applying the following operation, based on pheromone trails.
 The operations are applied to individual(s) selected from the population with a probability based on fitness.
 • Darwinian Reproduction
 • Structure-Preserving Crossover
 • Structure-Preserving Mutation
 End While
```

Applications of ACO

Ant colony optimization algorithms have been applied to many combinatorial optimization problems, ranging from quadratic assignment to protein folding routing vehicles , flooding, shortest path and a lot of derived methods have been adapted to dynamic problems in real variables, stochastic problems, multi-targets and parallel implementations. It has also been used to produce near-optimal solutions to the travelling salesman problem, Knapsack problem. They have an advantage over simulated annealing and genetic algorithm approaches of similar problems when the graph may change dynamically.

The first ACO algorithm was called the Ant system and it was aimed to solve the traveling salesman problem, in which the goal is to find the shortest round-trip to link a series of cities. The general algorithm is relatively simple and based on a set of ants, each making one of the possible round-trips along the cities. At each stage, the ant chooses to move from one vertex to another according to some rules:

1. It must visit each vertex exactly once;
2. A distant vertex has less chance of being chosen (the visibility);
3. The more intense the pheromone trail laid out on an edge between two vertex's, the greater the probability that that edge will be chosen;
4. Having completed its journey, the ant deposits more pheromones on all edges it traversed, if the journey is short;
5. After each iteration, trails of pheromones evaporate.

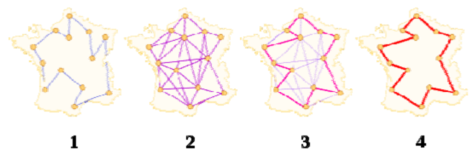

Figure 2: Path of sub tasks

C. Numerical Example

Suppose a service S of Ant colony optimization $M = (Pf\Omega)$. Initialize Pheromone values(τ) where ξ_{bs} is initialized to null while the termination conditions do not meet then $\eta_{iter} = \phi$ where $\sigma = (2,4,5,9,10........n_a)$ then if ξ is valid then construct τ and perform a linear search

$$\tau(\{x\}) = \sqrt{(\tau')^2 + 2\tau'\tau''\frac{x_2}{2R} + (\tau'')^2}$$

$$\tau' = \frac{P}{\sqrt{2}x_1x_2}, \quad \tau' = \frac{MR}{J}$$

$$M = P\left(L + \frac{x_2}{2}\right), \quad R = \sqrt{\frac{x_2^2}{4} + \left(\frac{x_1 + x_3}{2}\right)^2}$$

$$J = 2\left\{\sqrt{2}x_1x_2\left[\frac{x_2^2}{12} + \left(\frac{x_1 + x_3}{2}\right)^2\right]\right\}$$

$$\sigma(\{x\}) = \frac{6PL}{x_4x_3^2}, \quad \delta(\{x\}) = \frac{4PL^3}{Ex_3^3x_4}$$

$$P_c(\{x\}) = \frac{4.013E\sqrt{\frac{x_3^2x_4^6}{36}}}{L^2}\left(1 - \frac{x_3}{2L}\sqrt{\frac{E}{4G}}\right)$$

$$L = 14in, \quad P = 6{,}000lb$$

$$E = 30 \times 10^6 psi, \quad G = 12 \times 10^6 psi$$

Table I: Attributes Of Subtasks

Subtask I	1	2	3	4	5
Software failure intensity $\lambda_{i}s(10^{-5}/s)$	1.0	1.5	1.4	1.6	1.0
Subtask Complexity C_i	6.0	8.0	7.0	10	8.5
Exchanged data b_{ik}	12	16	14	20	17

Thus, the ACO can often find a near optimal solution even though it may not guarantee the optimum every time. The average execution time is about 0.0157 seconds per run, which is much efficient.

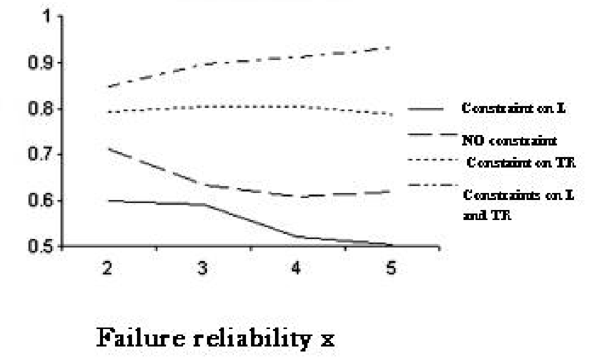

Figure. 3. Grid service reliability with respect to failure recoverability in difference situations.

The values of grid service reliability are increasing while the increasing of grid service reliability without any constraint is the fastest among three situations. In particular, when when X=1 R=0.9767, R^2=0.9678,R^3=0.9447. From the above analysis, we can see that fault recovery has a positive influence on grid service reliability. However, the

Table II: Statistics Of Grid Service Reliability And Total Cost For 100 Aco Runs

	Max	Min	Ave	Std.Derivation
Grid Service reliability	0.8527	0.8501	0.8527	3.8257x10^{-4}
Total Cost	215.3	213.3	213.3	0.2927
Execution Time(s)	0.0214	0.0146	0.0157	0.0016

constraints on both the life times of subtasks and the numbers of recoveries performed, which is customized by resource providers, have a negative impact on grid service reliability. To balance the contradiction of them, the optimization of task scheduling is necessary so as to satisfy the demand of users.

IV. CONCLUSION

In this paper, a fault recovery mechanism is introduced into the grid, and the modeling of grid service reliability considering fault recovery is presented. In order to make it more practical, a constraint on recovery amount is discussed in the modeling of grid service reliability. As for the implementation of fault recovery in grid resources, it can be achieved by embedding a fault recovery module in grid clients. In the module, there are options such as the allowed life times of grid subtasks, and the allowed numbers of recoveries performed. By those options, resource providers can be free to choose appropriate fault recovery strategies according to the local situations. The future study is based on that, a task scheduling optimization model to maximize grid service reliability and minimize the total cost simultaneously is proposed, and an ACO algorithm is used to solve this task scheduling problem. Since Ant colony algorithm may produce redundant states in the graph, its better to minimize such graphs to enhance the behavior of the inducted system. A colony of ants moves through system states X, by applying Genetic Operations. By moving, each ant incrementally constructs a solution to the problem when an ant complete solution, or during the construction phase, the ant evaluates the solution and modifies the trail value on the components used in its solution. Ants deposit a certain amount of pheromone on the components; that is, either on the vertices or on the edges that they traverse. The amount of pheromone deposited may depend on the quality of the solution found.

REFERENCES

[1] I. Foster, "The Grid: A new infrastructure for 21st century science," Physics Today, vol. 55, no. 2, pp. 42–47, 2002.

[2] Y. S. Dai, M. Xie, and K. L. Poh, "Reliability of grid service systems," Computers and Industrial Engineering, vol. 50, no. 1–2, pp. 130–147,2006.

[3] S. C. Guo, H. Wan, G. B. Wang, and M. Xie, "Analysis of grid resourcecompensation in market-oriented environment," Eksploatacja I Niezawodnoœæ—Maintenance and Reliability, vol. 45, no. 2, pp. 36–42, 2010.

[4] K. Krauter, R. Buyya, and M. Maheswaran, "A taxonomy and survey of grid resource management systems for distributed computing," Software—Practice and Experience, vol. 32, no. 2, pp. 135–164, 2002.

[5] C. L. Li and L. Y. Li, "Multiple QoS modeling and algorithm in computational grid," Journal of Systems Engineering and Electronics, vol.18, no. 2, pp. 412–417, 2007.

[6] G. Levitin and Y. S. Dai, "Grid service reliability and performance in grid system with star topology," Reliability Engineering and System Safety, vol. 92, no. 1, pp. 40–46, 2007.

[7] M. Dorigo and C. Blum, "Ant colony optimization theory: A survey," Theoretical Computer Science, vol. 344, no. 2-3, pp. 243–278, 2004.

[8] G. D. Caro and M. Dorigo, "AntNet: distributed stigmergetic control for communications networks," Journal of Artificial Intelligence Research, vol. 9, no. 2, pp. 317–365, 1998.

[9] D. Merkle, M. Middendorf, and H. Schmeck, "Ant colony optimization for resource-constrained project scheduling," IEEE Transactions on Evolutionary Computation, vol. 6, no. 4, pp. 333–346, 2002.

[10] V. Maniezzo, "Exact and approximate nondeterministic tree-search procedures for the quadratic assignment problem," INFORMS Journal on Computing, vol. 11, no. 4, pp. 358–369, 1999.

Mrs. Ashwini Vedulla received Master of Technology from Jawaharlal Nehru Technological University Hyderabad. Currently working as Assistant Professor in the department of Information Technology at Bharat Institute of Engineering and Technology, Hyderabad

A. Pramod Reddy received Master of Technology from Jawaharlal Nehru Technological University Hyderabad. He also certified Microsoft engineer since 2005. Currently working as Assistant Professor in the department of Information Technology at Sri Indu college of Engineering and Technology, Hyderabad.

Mr.B.krishna Prasad received his M.S from university of Michigan,USA currently working as Associate Professor in department of Information Technolgy at Bharat Institute of Engineering and Technology. His interested areas are computer networks.