FAILURE RECOVERY USING RSWF ALGORITHM FOR ADVANCED RESERVATION IN OPTICAL GRID

[1]MaheshA:Khandke; [2]GaneshM:Kadam [3]AnirudhK:Mangore

Department of Computer Engineering,Gharda Institute of Technology
Lavel, Ratnagiri, M.S., India

[1]maheshkhandke@gmail:com [2]gmkpatil@gmail:com
[3]anirudhmangore@gmail:com

ABSTRACT

For utilizing distributed resources in optical grid environment advanced reservations play a very crucial role. For applications like the co-allocation of distributed resources and deadline driven applications advance reservations are essential. Also for enhancing capabilities of resource brokers advance reservations play a major role. In this paper, failure recovery has been done depending on schemes such as periodic reconfiguration, and so on using Revised Slide Window First (RSWF) algorithm. Here, RSWF algorithm checks one path at a time for all of the scheduling window slots. If a link or node failure occures during the reservation then periodic rescheduling can be done in RSWF to perform failure recovery.

KEY WORDS

Advance reservation, optical grid, RSWF algorithm.

I. INTRODUCTION

Over the past few years it has become evident that local computational resources cannot keep up with the everincreasing demand for processing power. The solution to this problem came in the form of distributed computing, aggregating the power of a multitude of computational resources in one big Grid. This Grid is named after the analogy with the electricity grid, and provides users with on-demand resource usage [13]. Advanced reservation in Grid systems guarantees the availability of resources to users at some specified future time. It can ensure the availability of the Grids heterogeneous resources in future and help a scheduler to produce better schedules [4]. It may be possible that after performing the schedulling of the resources node or link associated with that scheduling fails. Failure recovery needs to be performed for such cases. The RSWF and SWF both the algorithm takes care of this failure recovery.

In this paper, failure recovery has been done depending on parameters such as periodic reconfiguration, and so on using Revised Slide Window First (RSWF) algorithm. Here, RSWF algorithm checks one path at a time for all of the scheduling window slots. If a link or node

DOI : 10.5121/cseij.2012.2603

failure occures during the reservation then periodic rescheduling can be done in RSWF to perform failure recovery.

Rest of the paper is organized as follows: section 2 explains the Wavelength Assignment Schemes and Network Optimization. RSWF algorithm is explained in section 3. Simulation results are explained in section 4. Finally, conclusion and future work is described in section 5.

II. WAVELENGTH ASSIGNMENT SCHEMES AND NETWORK OPTIMIZATION

In this section, we discuss the various Wavelength Assignment Schemes and Network Optimization techniques considered in this paper.

A. Wavelength Assignment Schemes

There are multiple wavelengths on a particular link along the source-destination path, are available for the advance reservation in optical grid. In this case, it is the responsibility of the scheduling algorithm to allocate the wavelength, from the pool of available wavelengths. We use the following two wavelength allocation schemes:

First-Fit : This is a popular method for wavelength assignment as it requires least computation time. Also, the performance is good as compare to others. In this scheme, all wavelengths are arbitrarily numbered. For searching available wavelengths, we always start from first wavelength and proceed sequentially to the last wavelength until a free wavelength is found. By searching the wavelengths in this manner, connections are packed into a smaller number of wavelengths thus freeing other wavelengths for reservations. First-fit scheme does not require global knowledge of the network and it does not required storage to keep the network state [7].

Min-Gap : Min Gap scheme aims at reducing the fragmentation in wavelength usage. In advance reservations, we have the advanced information of the wavelength usage with the associated times, and hence we can use this information when assigning the wavelength in order to reduce the gaps between reservations. But the main drawback of this scheme is that, it requires additional computational overhead as compare to first fit when searching for the least gap throughout the existing reservations [7].

There can be three different ways to minimize the gaps [7]:

- Min-Leading-Gap: This wavelength assignment schememinimizes the unused leading gaps on a wavelength.
- Min-Trailing-Gap: This scheme minimizes the unused trailing gaps on a wavelength.
- Best-Fit: This scheme minimizes the sum of the leading and trailing gaps.

An example of these wavelength allocation schemes is shown in figure 1. We can see that the new lightpath (LP) reservation can be accommodated by all the four wavelengths. First-Fit will select λ1:λ2 minimizes the leading gap between the new and previous reservations therefore it will be selected by Min-Leading-Gap scheme, while λ3 minimizes the trailing gaps so it will be selected by Min-Trailing-Gap. The Best-Fit will select λ4, as it minimizes the sum of leading and trailing gaps [7].

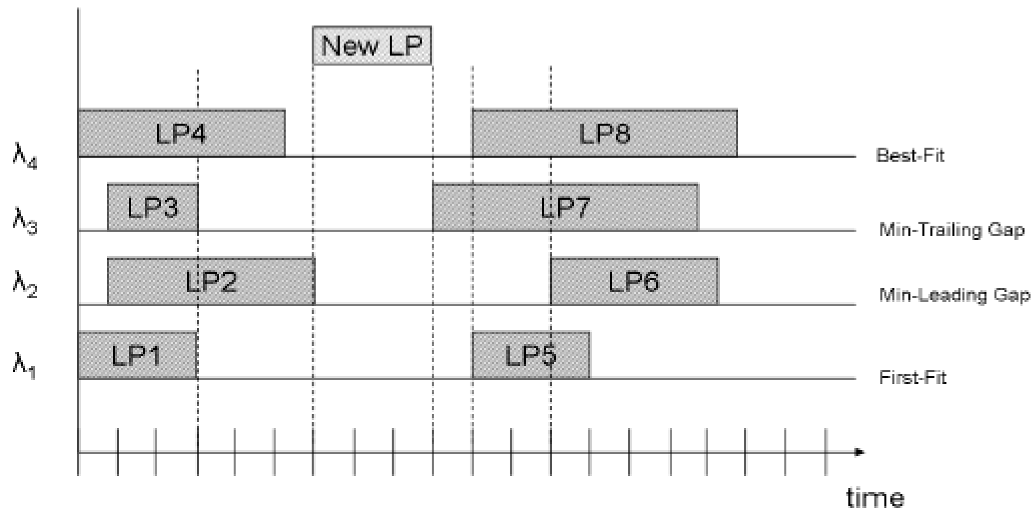

Fig. 1. Wavelength Assignment using Min-Gap.

B. Network Failures

Network failure is possible in accepted advanced reservations. In view of this, it is necessary to make a strategy in order to restore the existing network connections and reschedule future reservations. There are two types of network component failures are present, one is link failure and the other is node failure.

- Link failure: When a link failure occures, all its constituent wavelengths will fail. Link failure is more of a concern because most nodes have built in redundancy.
- Node failure: When a node failure occures, all its constituent link will fail.

We develop appropriate path protection and restoration schemes to prevent or reduce data loss, so that the high data rate on these networks remains unaffected. In case of protection, a path or a link is protected against failures by pre assigning resources for a backup path, while in restoration schemes an alternate route is discovered dynamically for each failed connection after the failure occurs. As compare to restoration schemes protection schemes have faster recovery time and provide guaranteed recovery. But the protection schemes require more network resources which increases cost.

Therefore due to lower costs involved, we use the restoration mechanism. Given a failure, the question of interest is how far into the future should we reschedule the existing reservations. In order to deal with network failures the DNRM must be aware of the current network state. This information can be obtained from the network devices in the form of a link state database using SNMP or other management protocols. Routing protocols, such as OSPF react slowly to network failures. In view of this, monitoring tools that talk directly to the network devices to get the latest state of interfaces and ports should be used as well [9].

Rescheduling needs to be done for all the services in connections which are associated with the failed link. Also, the requests that are scheduled but not yet in service that use the failed link are also rerouted for higher performance. Because of these requests are likely to be affected, if the failed link is not restored on time. The easy approach is to re-schedule all the future connections that use the failed link. However if the link is repaired early, the connections will be using sub-optimal paths therefore this may not be good solution. So to minimize the number of dropped reservations a re-routing interval has to be determined to re-schedule future reservations. If a free path cannot be found for the start time that a reservation has already been made the reservations can be dropped when rescheduling them.

Since the failure duration is not known in advance, the re-routing interval has to estimate using different methods. Following two methods are used for estimating rerouting interval [7]:

- Load-based: In this approach, a threshold value η which depends on the network topology is considered. The current load and the booking profile during the next time slot should be less than η
- Feedback based: In this method, the initial re-routing interval is one slot and it is increased based on the percentage of successfully re-routed reservations during the last slot. The interval duration is increased until the percentage of successfully re-routed reservations is sufficiently high .

In this paper, we have used feedback based approach. In this, depending on the knowledge of the duration of previous failures re-routing interval has been set. Normally, we keep record of a moving average of the historical failure durations and from this restoration time can be estimated. If the link does not repaired at the end of the current re-routing interval we increase the re-routing interval by an amount equal to the previous re-routing interval.

C. Network Optimization

With the help of rescheduling of the lightpath requests network optimization can be performed. There are two ways of doing this one is by changing paths only other is by changing both the path and the start time of the request. Second approach is not very desirable because it involves renegotiating the time duration with the users. Therefore we are using the first approach.

There are number of optimization techniques are present for static traffic demand and long term on-demand traffic flows. We are using a scheme in which we are re-configuring the request without changing their reservation times [7]. This scheme is based on the principle of rescheduling. All the existing connections which are time-overlap with blocked request are unscheduled. After that we perform re-scheduling by sorting them according to their start times. Because of this, provisioning of all the requests including the blocked request is possible. If this strategy does not succeed then all connections are restored to their original state without reconfiguration. Here we are periodically re-scheduling all the requests for a specific number of slots in future for every blocked request.

III. REVISED SLIDE WINDOW FIRST (RSWF) ALGORITHM

In this algorithm, we try to find a free period d starting at s + t, where t = 0, 1, 2, ...e - s - d. If the first shortest path is not free for the required duration during the window, the busiest link defined as the one that uses the maximum number of slots during the scheduling window, is removed from the network topology and the procedure is repeated until either an available path is found or a maximum of k paths is considered. The only difference between SWF algorithm and RSWF algorithm is that Dijkstras algorithm is implemented parallely [12] in RSWF algorithm [13].

The pseudo code is as follows :

```
1)  Algorithm 1: The RSWF algorithm
2)  function FindPath(Request r, topology t)
3)  i = 1
4)  while (î ≤  k) do
5)  start time = s
6)  end time = s + d
7)  find shortest path with Parallel Dijkstras algorithm with propagation delay link cost
8)  if A path is found then
9)  while (endtime e) do
10) if wavelengths are available on all links during start time and end time then
11) assign wavelengths, update all tables
12) return
13) else
14) start time = start time + t0
15) end time = start time + d
16) end if
17) end while
18) else
19) remove the busiest link during the window from topology
20) end if
21) i ++
22) end while
23) end function
```

IV. EXPERIMENT RESULTS

GridSim Toolkit [10] is used for conducting simulation run, since it supports advanced reservation. Here, we have taken 44 nodes and 94 unidirectional links. We assume 10 wavelengths on each link and full wavelength conversion at each node.

We assume that requests arrive in a Poisson fashion and all requests need to reserve a lightpath with bandwidth equal to one wavelength. The duration of a reservation is uniformly distributed. The start times of the request are generated within a window of 400 slots. To simulate a more realistic environment, we have generated the intermediate period between the arrival of the request and the start of the reservation using the discrete probability distribution. We assume that more requests will be for the reservation slots in the near future i.e next 24 slots and very few requests will be for reservation slots far into the future e.g. after two or three hours. The source and destination nodes for the requested connection are selected randomly using a uniform distribution.

We have implemented two approaches for wavelength assignment, First Fit and another is Min-Gap as discussed earlier in this paper. Fig.2 shows that as Best- Fit leaves small un-used gaps before and after a reservation, it has worst performance. First Fit has a moderate performance.

Because of the lowest blocking probability, Min-Gap scheme with minimum trailing and leading gap give highest performance.

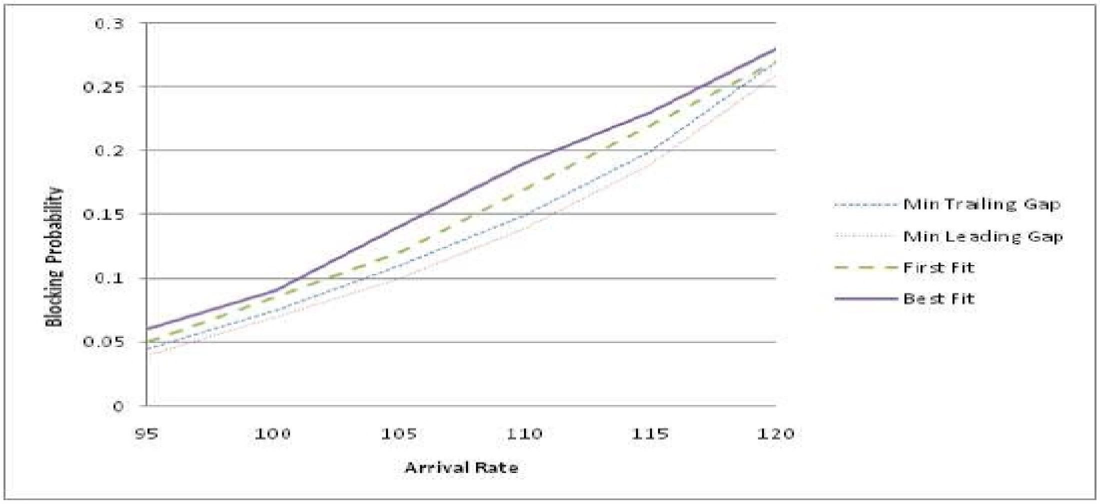

Fig. 2. Wavelength Assignment.

Fig. 3 shows the affect of the requests with short and long durations on each other. To identify this affect we generated requests which have a maximum duration dmax=20 with 50

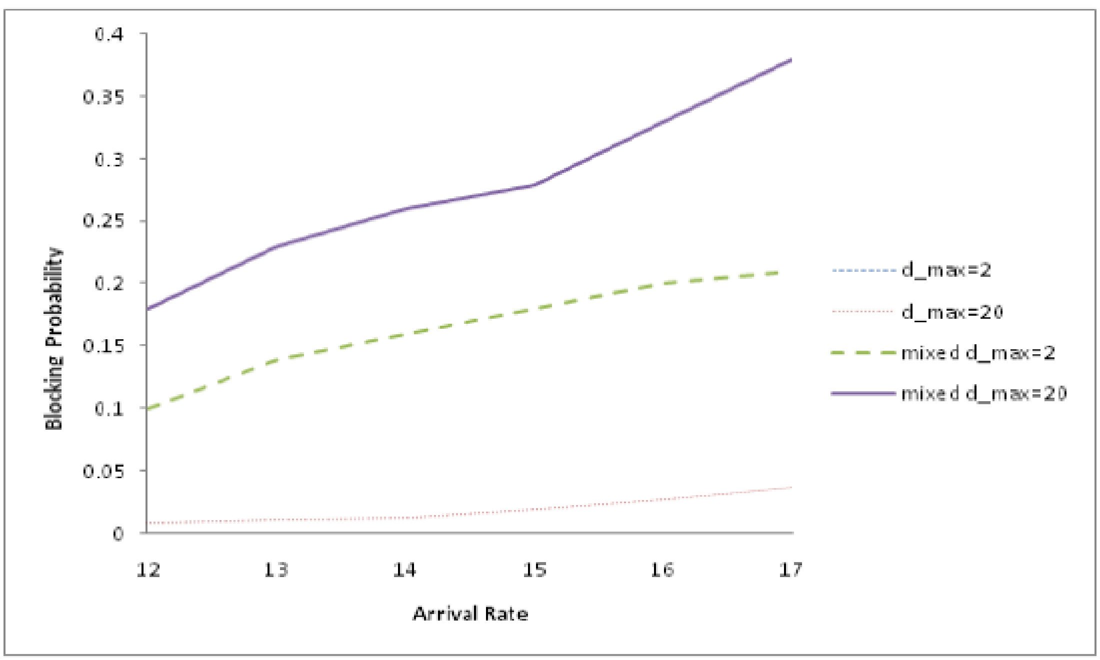

Fig. 3. Requests for long and short durations simultaneously.

A. Failure Recovery

this simulation run, we have considered link failures for the failure recovery. With a one link failure all wavelengths associated with that link are failing. We consider only one link failure at a time and we are randomly selecting that link from all the links. We have distributed mean time to failure

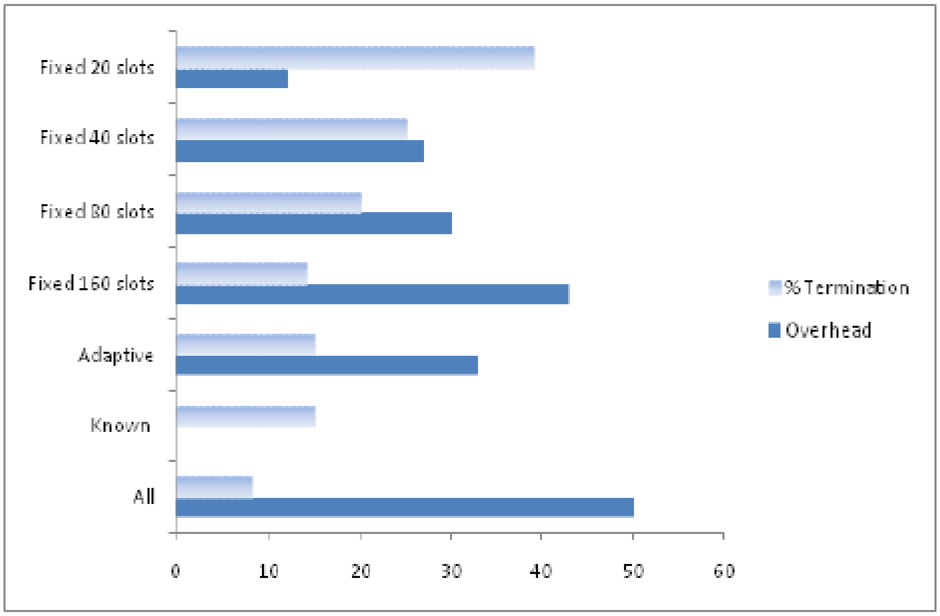

Fig. 4. Failure Recovery.

exponentially with a mean of 80 slots. The recovery time is also exponentially distributed with a mean of 48 slots.

Here we have calculated parameters such as the percentage termination and the overhead. Percentage termination is the percentage of reserved connections that cannot be re-routed and are dropped. The overhead is defined as the percentage of reserved connections that are unnecessarily rerouted as they were un-affected by the failure because the reservation start time turns out to be after the instant that link is repaired [7]. Affect of blocking probability on failure recovery process is also measured.

Following three policies are used to calculate the re-routing interval [7]:

1) The Fixed re-routing interval with feedback: The length of the re-routing is fixed. When the link goes down the connections are re-routed for this fixed interval. If the link does not come up at the end of the interval, the re-routing interval is increased by an amount equal to the fixed value.
2) Adaptive re-routing interval with feedback: In this case the re-routing interval is calculated using a moving average of the historical recovery times. If the link does not come up at the end of the interval, the re-routing interval is increased by an amount equal to the last rerouting interval length.

3) All future reservations are re-routed: In this case there is no re-routing interval, all the reservations on the failed link are re-routed.

We compare all the three approaches with the ideal scenario where the failure duration is known and only the flows affected are re-routed.

Figure 4, shows the results of our simulation. We can observe that the smaller the re-routing interval the higher is the termination percentage, because at an earlier time more alternative paths are free and the reservations can be re-scheduled. If the reservations are re-routed just before their starting time, the remaining alternate routes are more likely to be busy. But with the longer re-routing intervals the overhead increases. Since both the requirements are conflicting with each other, we are using the tradeoff between the two approaches. When for determining the re-routing intervals we use the adaptive method we get the best results.

B. Periodic Re-configuration

We have implemented periodic re-configuration of existing reservations to improve the network performance. The reconfiguration process occurs every 24 slots and re-configures the existing reservations for the next 24 slots without changing the reservation times. For this we consider start times of the requests. Depending on the start times requests are sorted in ascending order. These sorted requests are then re-scheduled one at a time starting from the request with the earliest start time. This process can result in terminations, in which case we restore the reservation table to its last state (i.e. no re-configuration). We can see from the fig. 5 that the reconfiguration process failed most of the time, i.e., because of the at least one termination, especially at high rates, the reservation table was restored to its last state. Process success rate is high when we allowed small percentage of terminations.

Figure 5 shows the blocking probability as a function of the arrival rate with and without re-configuration. We can observed that, with increase in the re-configuration overall locking probability increases. Also, the overall blocking probability in the network increases with re-configuration which allows 2

From the results, we can conclude that the link failure can be controlled by using this method of periodic re-configuration for offline optimization. Therefore periodic re-configuration is a good solution to a failure recovery.

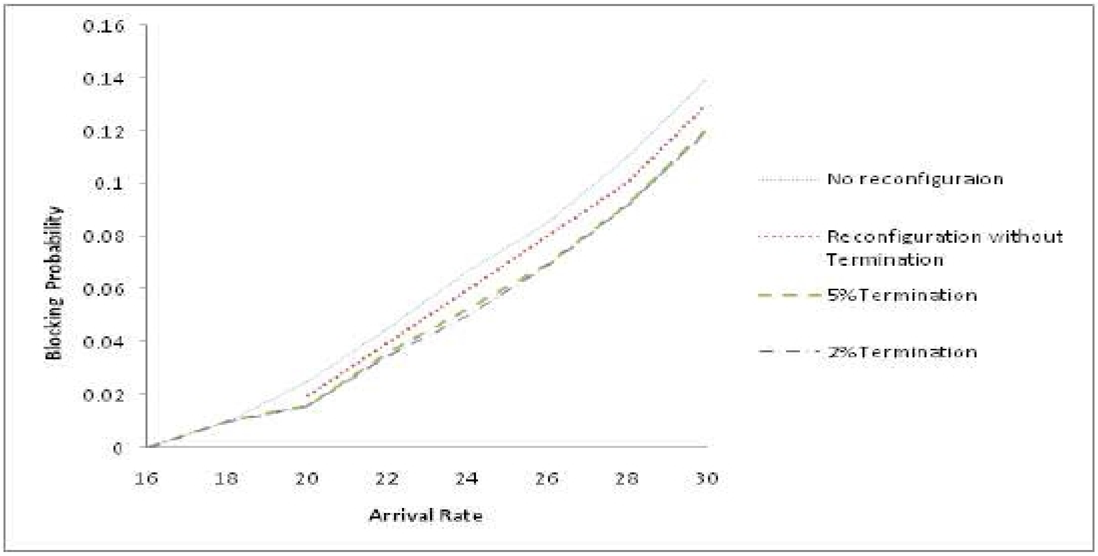

Fig. 5. Periodic Reconfiguration.

V. CONCLUSION

Advanced reservation represents an important mechanism in Optical Grid which allows applications to request resources for use at a specific time in the future. For advanced reservation in Optical Grid many algorithms are present.

In this paper, several approaches have been analyzed for failure recovery. All this approaches are depending on rerouting interval of all the wavelengths of each failed link. All the reservations which are associated with failed link are rescheduled. Simulation results shows that a long time interval re-routes unaffected connections while short re-routing interval gives large number of terminated connections. So we have used tradeoff between these two approaches to get the best possible solution. When the re-routing interval is depends on the moving average of the historical failure times the performance is increases.

REFERENCES

[1] Ahmar Abbas, Grid Computing: A Practical Guide To Technology and Applications, Firewall Media, 2008.
[2] Ian Foster, Carl Kesselman, Steven Tuecke , “The Anatomy of the Grid”,2001.
[3] Ian Foster, Carl Kesselman, Steven Tuecke , Jeffrey M. Nick, “The Physiology of the Grid”, 2002.
[4] Mark Bartlett, Alan M. Frisch, Youssef Hamadi, Ian Miguel, Chris Unsworth, “Efficient Algorithms for Selecting Advanced Reservations”, IEEE,2006.
[5] Jawad Ashraf and Thomas Erlebach, “A New Resource Mapping Technique for Grid Workflows in Advance Reservation Environments”, IEEE, 2010.
[6] Hidemoto Nakada, Atsuko Takefusa, Katsuhiko Ookubo, Makoto Kishimoto Tomohiro Kudoh, Yoshio Tanaka, Satoshi Sekiguchi, “Design and Implementation of a Local Scheduling System with Advance Reservation for Co-allocation on the Grid”, IEEE, 2006.
[7] Savera Tanwir, Lina Battestilli, Harry Perros and Gigi Karmous-Edwards, ”Dynamic Scheduling of Network Resources with Advance Reservations in Optical Grids ”, 2007.

[8] Min Zhu, Wei Guo, Shilin Xiao, Anne Wei, Yaohui Jin, Weisheng Hu, and Benoit Geller, ”Availability-Driven Scheduling for Real-Time Directed Acyclic Graph Applications in Optical Grids”, Opt. Comm. Network, 2010.
[9] Wei Guo, Weiqiang Sun, Weisheng Hu, Yaohui Jin, ”Resource Allocation Strategies for Data-Intensive Workflow-based applications in Optical Grids”, IEEE, 2006.
[10] GridSim Website. http://www.gridbus.org/gridsim, 2008.
[11] Anthony Sulistio, Rajkumar Buyya, ”A Simulation Infrastructure for Resource Allocation with Advanced Reservation in Global Grids”, 2004.
[12] Fang Zhou Lin, Nan Zhao, ”Parallel Implementation of Dijkstra’s Algorithm”, 2004.
[13] Santosh Sable, G.V.Chowdhari, Mahesh A. Khandke, ”Revised Slide Window First Algorithm for Advanced Reservation in Optical Grid”, CSEIJ, vol 2, No.3, June 2012.

Authors

Mr. Mahesh A. Khandke received his Bachelor’s & Master’s Degree in Computer Science and Engineering from Dr. B.A.T.U. Lonere, Maharashtra, India. His area of interest is Data Structure, Distributed Computing and Grid computing. He has published 2 papers in international journals.

Mr. Ganesh M. Kadam received his Bachelor’s Degree in Information Technology from Shri Ramanand Tirth University, Nanded, Maharashtra, India. Now he is pursuing Master’s Degree in Computer Science and Engineering from Jawaharlal Nehru Technological University, India. His area of interest is Design of Algorithms and Grid computing.

Mr. Anirudh K. Mangore received his Bachelor’s Degree in Computer Science and Engineering from Shivaji University, Kolhapur, Maharashtra, India. Now he is pursuing Master’s Degree in Computer Science from Jawaharlal Nehru Technological University, India. His area of interest is Distributed Database and Grid computing.