Recovery of Failures in Transaction Oriented Composite Grid Service

Dharmendra Prasad Mahato[1], Lokendra Singh Umrao[2], Ravi Shankar Singh[3]
[1,2,3]Department of Computer Science and Engineering, Indian Institute of Technology (BHU)-Varanasi, India

Abstract
Transaction Oriented Composite Grid service is a group of sub services to be executed in Grid environment when transaction management is used. Since Grid services are loosely coupled and dynamic in nature, the transaction management becomes tough task in this environment. As the number of services increase, the chances of failures also increase due to different types of faults occurring in the system. Therefore fault tolerant execution of these tasks is required to maintain the reliability, availability, dependability of the system. In this paper we have implemented coordinated check-pointing approach to tolerate the faults so that resiliency, reliability, availability, and dependability can be enhanced. For recovery of the failed processes we have compared both local node recovery and replicated node recovery by simulating in CPN tool. Here we have considered three types of faults such as hardware faults, communication link faults, and software faults. All the faults have been modelled dynamically in the simulation. The results show that the local node recovery is better than replicated node recovery when the number of services is minimum but in the case of large number of services the replicated node recovery works better. Our results show that using local node recovery we can decrease the failures by 38.86% and when we use replicated nodes recovery we get that results decreasing by 31.34%.

Keywords
Transaction Management, Fault tolerance, reliability, availability, resiliency, dependability, CPN (Colored Petri Nets) tool, local recovery, replicated recovery.

1. INTRODUCTION
Grid computing has become the next-generation parallel and distributed computing methodology to provide a service-oriented infrastructure that leverages standardized protocols and services to enable pervasive access with coordinated sharing of geographically distributed hardware, software and information resources for solving various kinds of large-scale parallel applications in the wide area network. However, it is a big challenge to make service execution in grid systems in a reliable manner [18], [11], [26].

A. Composite Grid Service
In service oriented computing applications, different re-sources in grid systems are encapsulated abstractly as service. A grid service is a computational unit that exists at a high abstraction level, usually closely related to functionality for service consumers [18]. Most of the time single atomic service can not satisfy service consumer's requests; therefore, a new composite service which is collection of all available qualified grid services is built [20]. But running a composite grid service is not an easy task as the resources and applications for execution in grid environment are dynamic and loosely coupled [15].

B Transaction Management in Grid Environment
The interoperation of services often gets affected from different faults like hardware faults, software faults, communication faults, byzantine faults, service expiry faults [5]. Hence selection of transactional grid services will guarantee reliable composition execution [2], [20]. If transactional composite grid services are successfully executed, the grid system will be in a consistent position even instead of fault-occurrence. But, co-ordination transaction for composite grid service is difficult, due to

1) Transaction co-ordination in composite grid service is often time consuming owing to interaction amongst users and latency.
2) Grid services are autonomous hence locking of needed resources is challenging.
3) Transaction always suffers from missing messages as communication is unreliable.
4) Services in grid environment are loosely coupled.
5) If transaction is implemented, the reliability is ensured but execution suffers from some faults [2], [19].

C Reliability of Grid Service
QoS-aware-grid service is affected when transaction-aware grid service is implemented. Hence both QoS-aware and transaction-aware grid service composition is required. Research has shown that the grid system, composed of thousands of heterogeneous resources located at disjoined domains, is very prone to failures due to its extreme complexity. Moreover, the likelihood of failure occurrence is often increased by the fact that many grid services requested by grid users will perform time-consuming tasks that may require several days or even months of computation. Therefore, it is very crucial to assure the quality and reliability of grid service so as to guarantee the correct outcomes of requested services to grid users. The main attributes which are really affected by the occurrence of faults are reliability, dependability, confidentiality, latency, availability, integrity, safety, throughput, and maintainability. Reliability completely depends on latency, throughput and availability factors. Dependability which is meant as trust on the system to execute the services correctly and successfully depends on reliability and availability factors [3], [7]. Hence we can notice that reliability is the important factor which is to be enhanced. As one of the important measures of quality of service (QoS), grid service reliability is considered to be one of the most critical and important issues in grid systems. With any application requirement, a corresponding service combined with the desired operations is created. Under the control of the resource management system (RMS), the service is supposed to execute certain task in the form of software programs. Grid service reliability is defined as the probability that all programs involved in the considered service are executed successfully. Recently, grid service reliability has attracted substantial research and attention [17].

D Fault Tolerant Execution of Grid Service
The faults which occur during the execution of the trans-action oriented grid services are Byzantine faults caused by many types

of faults such as hardware faults, communication link faults, and software faults. Therefore, for efficient execution of these services, fault tolerant mechanism is required by tolerating these Byzantine faults. Fault tolerance in grid environment is a necessary requirement so that that the system can continue even in the occurrence of faults [22].

Therefore, Co-ordinated Check-pointing fault tolerant mechanism which is well suited in this scenario must be required to enhance the reliability with efficient resource utilization [4], [9].

Basically the Co-ordinated check-pointing mechanism which is the commonly used mechanism for fault tolerance stores the information of the current application state, and then it is used for resuming the execution in case of failure [11], [13], [14], [24].

E Star Topology

How computational grid executes transaction oriented composite grid services are as follows:

The Open Grid Services Architecture (OGSA), widely adopted in industry and research, can develop a grid from a computing grid, data grid, or other dedicated grids to a service grid. We can say that a grid consists of a distributed server where all applications are provided in packages as services. Users submit their jobs to the grid client as service requests and the clients receives those service requests and decides sequence according to the scheduling rules and sends the results to the scheduler [1], [5].

1. Scheduler agent decides which one processor is to be assigned for particular service request.
2. Now processors assign services by dividing them to multiple sub-services and distributing them to sub-processors.
3. Thereafter, transaction management is used for reliable execution of distributed sub-services.
4. Results (failed or successful) of the jobs are submitted to users upon successful completion of the jobs by monitor. The monitor agent keeps the status of progress of every service and then sends these messages to the client, and scheduler agent. When sub-services are completed, the results are merged and sent to the client.

All the activities from distribution of services to different sub services to integration of those sub services after execution are following star topology.

F Recovery of Failed Services

In grid system there are two recoveries which can be achieved with the help of the information taken during check-pointing, 1) Local faults recovery; 2) Migration faults recovery [2]. Migration fault recovery: When failures occur at a grid node, the check-pointing information is migrated to other node at which the execution of sub-service is to be restarted. Local fault recovery: When sub-service is resumed on the same node where fault has occurred after recovery. This type of recovery is known as local fault recovery. It is better than migration fault recovery because it can save the migration time. But in the case of maximum number of services the replicated or migration recovery is better than local recovery [14], [1].

2. PROBLEM STATEMENT

How computational grid executes transaction oriented composite grid services are as follows:

i. Users submit their jobs to the grid client as service requests and the clients receives those service requests and decides sequence according to the scheduling rules and sends the results to the scheduler.

ii. Scheduler agent decides which one processor is to be assigned for particular service request.

iii. Now processors assign services by dividing them to multiple sub-services and distributing them to sub-processors of processors.

iv. Thereafter, transaction management is used for reliable execution of distributed sub-services.

v. Results (failed or successful) of the jobs are submitted to users upon successful completion of the jobs by monitor. The monitor keeps the status of the progress of every service and then sends these messages to the client, and scheduler agent. When sub-services are completed, the results are merged and sent to the client.

But, such a computational transactional oriented grid environment consists of two major draw backs:

a. If a fault occurs at a grid resource, the whole job is roll backed or aborted to maintain the ACID properties (Atomicity, Consistency, Isolation, and Durability) of transaction. This leads to inefficient use of resources as most of the successful executed sub-services are also roll backed. [10]

b. In grid environments, the resources accomplish the norm of deadline limitation, but they have an inclination to failure. In this situation, the scheduler selects the same resource that the grid resource has promised to meet the requirements of the user.

For the first problem, a job check-pointing strategy has been proposed to tolerate the faults, because it can restart the partially completed job from the last checkpoint. For the second problem, the check-pointing strategy should be made adaptive.

Thus, the checkpoint can be introduced whenever it is necessary. The application can only be restarted from the last known state, if the checkpoint is available. To increase the availability of checkpoint, Co-ordinated Check point Scheme can be used. Using this scheme, the current application state can be taken at any time. Simulation experiments show that the proposed fault tolerance mechanism is able to tolerate the faults by taking appropriate measures. [1]

3. METHODOLOGY

The Co-ordinated Checkpointing proposed here considers fault tolerance in transaction oriented grid environment to optimize user-centric metrics like execution time and jobs completed within deadline even in the presence of faults. In general, a fault in grid environment occurs when a resource is not able to accomplish its job in a given time limit. When a fault like this occurs, the information about fault occurrence at the grid resource is updated. This information of fault occurrence is used while making a job check-pointing before allocating job to the grid resource [25]. Generally when a fault occurs at a grid resource, the service whose sub service executes at this resource is roll backed or is aborted. Hence the resource utilization is very much affected. Using Co-ordinated Checkpointing approach the sub-service which faces failure is not roll-backed but is again rescheduled for the execution so that better resource utilization can be guaranteed [20], [1].

4. EXPERIMENT AND IMPLEMENTATION

Here we have used Coloured Petri Nets tool for the simulation of our approach. CPN is a language for modelling and validation of system in which concurrency, communication, and synchronization play a major role. This language makes it possible to organize a model as a set of modules, and it includes a time concept for representing the time taken to execute events in the modelled system. Using CPN, information can be modelled by tokens and types of information can be modelled by the token colours. First of all Null Hypothesis has been modelled in which the transaction management works without fault tolerance mechanism. Then Alternate Hypothesis has been modelled with fault tolerance mechanism where number of failed process becomes less compared to the first model. Here the model consists of three participants agents: clients, processors, sub processors and two control agents: scheduler, and monitor. We assume that there are 100 clients, 100 processors, and 10 sub processors for every processor in a grid environment. Here the control agents are used to schedule, dispatch, and monitor grid services. At first service requests are sent to the client agent from the users [27], [28].

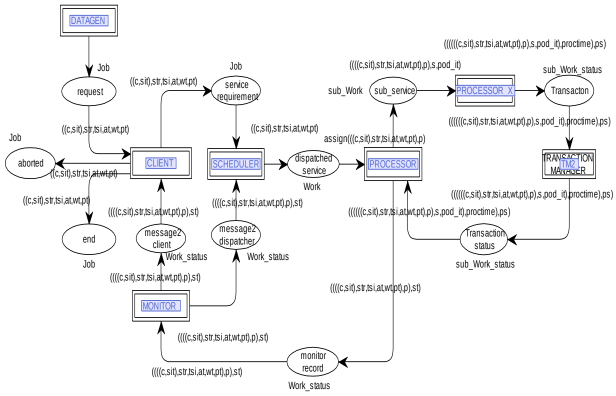

Fig.1.CPNGRID

The service requests are then transferred to the scheduler. The scheduler agent receives the service request and decides its sequence according to the schedule rules and dispatches these results to processor agent. Also the scheduler agent decides which one processor to execute the service request.

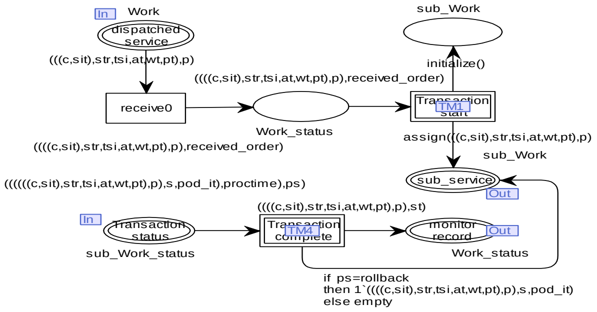

Fig.2.Loacal Recovery

For the execution of the service requests of the clients, sub service requests are transferred by processor agent to sub processor agent. Information to the client is sent that the monitor has accepted requests of the service. Thereafter, after tracking the progress status of service, the monitor agent sends these messages to the client and the scheduler agent. Fig.1 shows the Grid infrastructure in CPN. Fig.2 represents the implementation of Local node recovery and Fig.3 represents the implementation of Replicated node recovery in CPN.

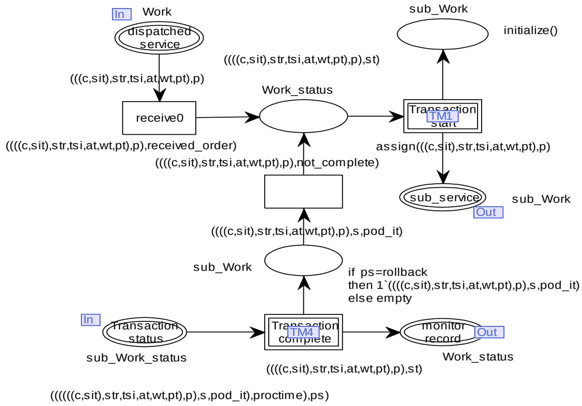

Fig.3.Replicated Recovery

For ensuring reliable execution of services in grid environment, transaction management is needed. In general, when transaction management is used in distributed grid environment, roll back or abort of even one sub service due to failure occurrence on link or node causes the roll back and abort of whole service to fulfil ACID properties of transactions. This leads to less utilization of resources.

After the completion of sub services, the results are merged and sent to the client. After getting the delivery, the client replies and sends acknowledgement to the processor [21].

5. RESULTS

We have compared the results of the three models; the first without fault tolerance mechanism, second with fault tolerance mechanism tolerating hardware fault and communication faults, and third with fault tolerant mechanism tolerating hardware faults, communication faults, and software faults. We have compared the results when recovery is done on local nodes and replicated nodes.

Here Fig.4 shows number of failed services of null and alternate hypotheses against time.

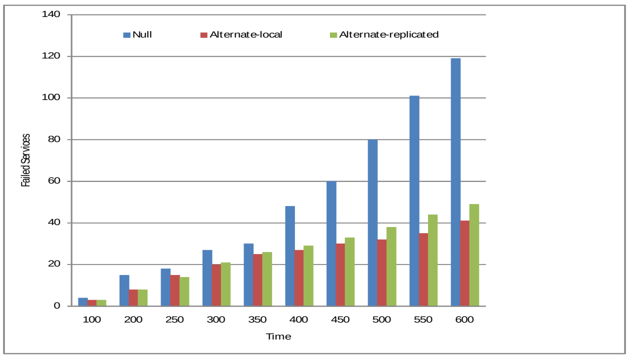

Fig.4. Time Vs Failed Services

Here Fig.5 shows how loads vary on different scenarios after fault tolerance mechanism is implemented.

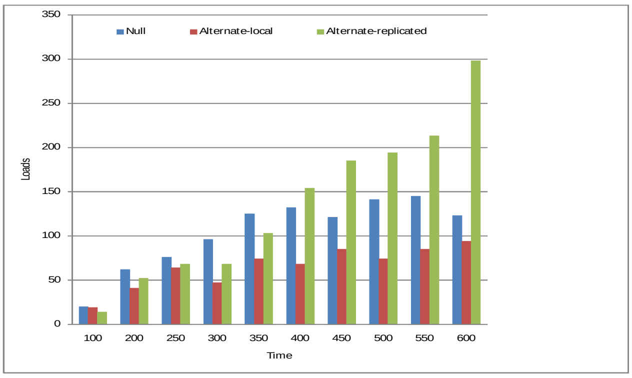

Fig.5. Time Vs Loads

The Fig.6 indicates the successful services against time in different scenarios after fault tolerance implementation.

The results show that the local node recovery is better than replicated node recovery when the number of services is less but in the case of large number of services the replicated node recovery works better. Our results show that using local node recovery we can decrease the failures by 38.86% and when we use replicated nodes recovery we get that results decreasing by 31.34%.

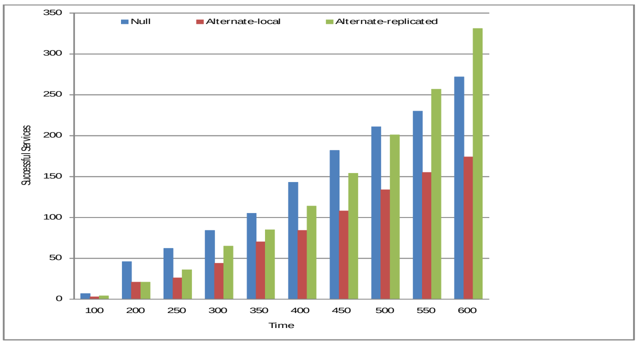

Fig.6. Time Vs Successful services

6. CONCLUSION

This paper analyses fault tolerance mechanism in grid system and presents the modelling of composite grid service reliability considering fault recovery when transaction management is used. Under these constraints, grid service reliability is modelled and analysed. Although the modelling and analysis of grid service reliability in our work are based on some simplified assumptions, our work addresses the important issue of adopting fault tolerance mechanism in grid system, and the models developed could be of practical use.

As for the implementation of fault recovery in grid resources, it can be achieved by embedding fault recovery module in grid clients located at grid nodes. In the module, there are some options, such as the allowed life times of grid subtasks and the allowed numbers of recoveries performed. By those options, resources providers can be free to choose appropriate fault recovery strategies according to the local situations. Yet more in-depth research on grid service reliability modelling and analysis is needed. For example, in realistic grid system, some precedence constraints on the order of subtask execution may be imposed and the usage amount of grid resources may be dynamic during the execution of grid subtask.

Here in the simulation we have seen that the recovery can be achieved by using both of the recovery methods either local recovery or replicated recovery. We have seen that local node recovery is better than replicated recovery, but in minimum number of services executing in the environment. For maximum number of services it is better to use replicated recovery.

During the simulation we have also seen that the load on the network increases when we use checkpointing mechanism. In our future work we will work on load balancing when checkpointing mechanism is used in transaction oriented composite grid service.

REFERENCES

[1] Guo, Suchang, Hong-Zhong Huang, and Yu Liu. "Modeling and Analysis of Grid Service Reliability Considering Fault Recovery." New Generation Computing 29.4 (2011): 345-364.

[2] Bohm, Matthias and Habich, Dirk and Lehner, Wolfgang and Wloka, Uwe “An advanced transaction model for recovery processing of inte-gration processes”, ADBIS (local proceedings), pp.90–105, 2008.

[3] Bolosky, W. J., Douceur, J. R., Ely, D. and Theimer, M., “Feasibility of a Serverless Distributed File System Deployed on an Existing Set of Desktop PCs”, in Proc. of the ACM International Conference on Measurement and Modeling of Computer Systems 2000, ACM Press, pp. 34-43, 2000. M. R. Yudith Cardinale, “Fault tolerant execution of transactional composite web services: An approach,” Services Computing, IEEE Transactions on, vol. 5, pp. 158 –164, jan.-april 2011.

[4] Affaan, M. and Ansari, M. A., Grid and Cooperative Computing, 2006. GCC 2006. Fifth International Conference, “Distributed Fault Management for Computational Grids”, 2006, pp.363–368.

[5] Dai, Y. S., Levitin, G. and Wang, X. L., “Optimal Task Partition and Distribution in Grid Service System with Common Cause Failures”, Future Generation Computer Systems, 23, 2, pp. 209-218, 2007.

[6] Yuan-Shun Dai and Levitin, G., Reliability, IEEE Transactions on, “Reliability and performance of tree-structured grid services”, 2006, 55, 2, pp.337-349.

[7] Dai, Y. S., Pan, Y. and Zou, X. K., “A Hierarchical Modeling and Analysis for Grid Service Reliability”, IEEE Transactions on Coers, 56, 5, pp. 681-691, 2007.

[8] Dai, Y. S., Xie, M. and Poh, K. L., “Reliability of Grid Service Systems”, Computers and Industrial Engineering, 50, 1, pp. 130-147, 2006.

[9] Jin Liang and Tong WeiQin and Tang JianQuan and Wang Bo, Industrial Informatics, 2003. INDIN 2003. Proceedings. IEEE International Conference on, “A fault-tolerance mechanism in grid”, 2003, pp.457–461.

[10] Foster, I., “The Grid: a New Infrastructure for 21st Century Science”, Physics Today, 55, 2, pp. 42-47, 2002.

[11] Foster, Ian, “The grid: A new infrastructure for 21st century science”, Grid Computing: Making the Global Infrastructure a Reality, pp.51–63, 2003, John Wiley & Sons, Chichester.

[12] Shi, Xuanhua and Pazat, Jean-Louis and Rodriguez, Eric and Jin, Hai and Jiang, Hongbo, “Adapting grid applications to safety using fault-tolerant methods: Design, implementation and evaluations”, Future Gener. Comput. Syst., February, 2010, 26, 2, 2010, 0167-739X, pp.236– 244,

9, acmid:1630314, Elsevier Science Publishers B. V., Amsterdam, The Netherlands, The Netherlands.

[13] Hwang, Soonwook and Kesselman, Carl, Journal of Grid Computing, 1, 3, “A Flexible Framework for Fault Tolerance in the Grid”, Kluwer Academic Publishers, pp.251–272, English2003.

[14] Kovcs, Jzsef and Kacsuk, Pter, Grid Computing, 3165, Lecture Notes in Computer Science, Dikaiakos, MariosD., “A Migration Framework for Executing Parallel Programs in the Grid”, Springer Berlin Heidelberg, pp.80–89.

[15] Bubak, Marian and Funika, Wdzimierz and Bali, Bartosz and Wismller, Roland, Parallel Processing and Applied Mathematics, 2328, Lecture Notes in Computer Science, Wyrzykowski, Roman and Dongarra, Jack and Paprzycki, Marcin and Waniewski, Jerzy, “A Concept of Grid Application Monitoring”, Springer Berlin Heidelberg, pp.307–314, En-glish2002.

[16] Foster, I. and Kesselman, C. and Nick, J.M. and Tuecke, S., Computer, “Grid services for distributed system integration”, 2002, 35, 6, pp.37–46, 0018–9162,1997.

[17] Abdelsalam Heddaya and Abdelsalam Helal, “Reliability, Availability,Dependability and Performability: A User-centered View”, 1997.

[18] Mache, Jens, “Hands-on grid computing with Globus Toolkit 4”, J. Com-put.Sci. Coll., December 2006, 22, 2, pp.99–100, 2, acmid:1181921, Consortium for Computing Sciences in Colleges, USA.

[19] An Liu and Qing Li and Liusheng Huang and Mingjun Xiao, “FACTS: A Framework for Fault-Tolerant Composition of Transactional Web Services”, IEEE Transactions on Services Computing, 3, 1, 1939-1374, 2010, pp.46-59 ,IEEE Computer Society, Los Alamitos, CA, USA.

[20] Cardinale, Yudith and Rukoz, Marta, “Fault Tolerant Execution of Transactional CompositeWeb Services: An Approach”, UBICOMM 2011, The Fifth International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies, pp.158–164, 2011.

[21] Chen, Jun and Gu, Yuesheng and Liu, Yanpei, Grid Service Concurrency Control Protocol, Journal of Networks, 7, 4, 707–714, 2012. Acmid: 1692817, Springer-Verlag, Berlin, Heidelberg.

[22] Haider, Sajjad and Ansari, Naveed Riaz and Akbar, Muhammad and Perwez, Mohammad Raza and Ghori, Khawaja Moyeez Ullah, “Fault Tolerance in Distributed Paradigms”, Proc. of Fifth International Conference on Computer Communication and Management, IACSIT Press, Singapore, 2011.

[23] Wang, Dexiang and Kumar, Arvindhan and Sivakumar, Madhan and McNair, Janise Y., “A fault-tolerant backbone network architecture targeting time-critical communication for avionic WDM LANs”, Proceedings of the 2009 IEEE international conference on Communications, ICC’09, 2009, 978-1-4244-3434-3, Dresden, Germany, pp.2596–2600, 5, acmid:1817752, IEEE Press, Piscataway, NJ, USA.

[24] Lopes, Rafael Fernandes and da Silva e Silva, Francisco Jose, ”Fault tolerance in a mobile agent based computational grid”, Cluster Computin and the Grid, 2006. CCGRID 06. Sixth IEEE International Symposium on, 2, pp.8–pp, 2006, IEEE.

[25] Kov´acs, J´ozsef and Kacsuk, Peter and Januszewski, Radoslaw and Jankowski, Gracjan, ”Application and middleware transparent checkpointing with TCKPT on ClusterGrids”, Future Generation Computer Systems, 26, 3, pp.498–503, 2010, Elsevier.

[26] Krpska, Elbieta and Kielmann, Thilo and Sirvent, Ral and Badia, RosaM., Achievements in European Research on Grid Systems, Gorlatch, Sergei and Bubak, Marian and Priol, Thierry, ”A Service for Reliable Execution of Grid Applications”, Springer US, pp.179–192,2008.

[27] Lai, Hong Feng. "Modeling grid workflow by coloured grid service net." Advances in Grid and Pervasive Computing. Springer Berlin Heidelberg, 2010. 204-213.

[28] Ma, Hua. "An Approach on Grid Services Transaction Management for Grid Workflow." Information Engineering and Computer Science, 2009. ICIECS 2009. International Conference on. IEEE, 2009.