DYNAMIC RESOURCE MANGEMENT USING ADAPTIVE WORKFLOW SCHEDULING AND NETWORK SECURITY VIRTUALIZATION TECHNOLOGY (NSVT) IN GRID COMPUTING

C.Daniel Nesakumar1, G.Suganya 2 , P.Jayasree 3

1Asst Prof , Dept. of Computer Application , danielnesakumar@gmail.com
2 Asst Prof , Dept. of Computer Application , suganyaag@gmail.com,
3 Asst Prof, Dept. of Computer Application ,kandasamysree@gmail.com,
Hindusthan College of Arts and Science, Coimbatore.

ABSTRACT: Grid computing permits huge amount of resource sharing and collaboration designed for solving complex science and engineering problems under many applications. Major core part of the grid computing is the allocation of resource to users depending on their application tasks to the resources. Allocation of resource to users is done under two ways: static and dynamic scheduling. dynamic scheduling and multi-institutional environment have many challenging problem under security In this work present a new concept of network security virtualization schema for grid computing security which virtualizes security of users and their resources, and consequently maximally make use of security devices. In addition, it permits security protection in the direction of popular networks by means of smallest management cost. In the direction of confirm this concept, additional propose and execute a prototype system, Network Security Virtualization Technology (NSVT) which be able to make use of presented pre-installed security devices and influence software-defined networking technology in the direction of virtualizes network security functions. NSTV contains: 1) a easy script language in the direction of register security services and policies; 2) a position of Scheduling algorithms in the direction of decide schedule workflow tasks in dynamic way designed for varied security policies depending on their various needs; Proposed a new Adaptive Workflow Scheduling (AWS) in the direction of schedule workflow tasks in the direction of the dynamic grid resources depending on their rescheduling method. The AWS algorithm is varied from existing techniques in literature by consideration of changes of both hosts and links availability during load balancing task. The simulation results with randomly generated task graphs conclude that the NSVT with AWS algorithm is capable in the direction of deal by means of fluctuations of resource accessibility and provides generally best performance.

Index terms: Grid computing, DAG grid workflow, Adaptive workflow scheduling, Re-scheduling, Resource monitoring, Network Security Virtualization Technology (NSVT), Security.

1.INTRODUCTION

In recent times, the speedy growth of networking knowledge and web have lead in the direction of the potential of via the use of huge amount of geographically circulated heterogeneous resources own by means of diverse organizations. These developments contain lead in the direction of the establishment of latest model known as Grid Computing [1-2The workflow scheduling in grid is one of the major issues, which handles by means of assigning workflow tasks in the direction of the obtainable grid resources at the same time as maintaining the task dependency restriction and in the direction of gather the Quality of Service (QoS) demands of the user that is minimizing the overall execution time and extend the performance.

In order toward make sure high performance in dynamic and changeable grid environment, let us prefer an adaptive scheduling [8] schemes where scheduling policy of this environment is dynamically changed based on the requirements of users and their resources accessibility. At this time, initial scheduling of each and every one the tasks is carry out statically and then rescheduling is also carryout for unexecuted tasks. The capabilities in the direction of determine and observe the category of resources on run time is essential designed for the adaptive operation of the grid. In this work proposed a new Adaptive Workflow Scheduling (AWS) in the direction of schedule workflow tasks in the direction of the dynamic grid resources depending on their rescheduling methods. It handles by means of the heterogeneous dynamic environment, where the accessibility of computing nodes and links bandwidth fluctuations are predictable appropriate in the direction of survival of local load or load by means of other users. The proposed AWS algorithm designed for grid applications by consideration of workflow tasks toward gather the performance requirements relying on QoS information like accessibility along with the convenience of the resources as point toward by means of Service Level Agreement (SLA).

During this scheduling task it considers the computing speed of each processors and network bandwidth accessibility by means of examining the load of grid processing nodes and network links. The working principle of proposed AWS algorithm is varied from other existing methods in the literatures since it is performed based on the dynamic heterogeneous environment by consideration of together computing nodes and communication links appropriate in the direction of existence of local load. It considers (i) Poverty of resource performances particularly computation speed of grid processing nodes and network bandwidth. (ii) Estimate the repayment of rescheduling by taking into the account of communication cost of nodes and overhead appropriate toward data transfer between nodes. (iii) Accessibility of newly added resources. The AWS algorithm is well-organized single as it attains less execution time of the application by means of the assist of rescheduling the computation is less.

our proposed work is varied based on the following consideration (i) Poverty of resource performances particularly computation speed of grid processing nodes and network bandwidth. (ii) Estimate the repayment of rescheduling by taking into the account of communication cost of nodes and overhead appropriate toward data transfer between nodes. (iii) Accessibility of newly added resources. The AWS algorithm is well-organized single as it attains less execution time of the application by means of the assist of rescheduling the computation is less.

Adaptive workflow technique (AWT) [17] is performed based on the makespan of jobs and Earliest Time First (ETF) .AWT methods scheduling of workflow tasks is applied to dynamic grid computing environment based on the job includes ETF determination be executed initially. But for larger applications is not easily applicable.

.

In [20] proposed a new rescheduling approach designed for large scale distributed system (Re-LSDS) toward maintain fault tolerance and resilience. Proposed Re-LSDS algorithm calculates the well-organized mapping of workflow tasks toward grid resources dynamically by means of determining the critical path at every step in DAG model. The simulation results show that the proposed Re-LSDS perform better when compared to heuristic-based scheduling techniques. But it doesn't support rescheduling under heterogeneity of the dynamic grid computing environment which is extended to this work.

2.PROPOSED METHODOLOGY

In the direction of gird security concept, additional propose and execute a prototype system, Network Security Virtualization Technology (NSVT) which be able to make use of presented pre-installed security devices and influence software-defined networking technology in the direction of virtualizes network security functions. NSTV contains: 1) a easy script language in the direction of register security services and policies; 2) a position of Scheduling algorithms in the direction of decide schedule workflow tasks in dynamic way designed for varied security policies depending on their various needs; In this work proposed a new Adaptive Workflow Scheduling (AWS) in the direction of schedule workflow tasks in the direction of the dynamic grid resources depending on their rescheduling methods. The working principle of proposed AWS algorithm is varied from other existing methods in the literatures since it is performed based on the dynamic heterogeneous environment by consideration of together computing nodes and communication links appropriate in the direction of existence of local load. It considers (i) Poverty of resource performances particularly computation speed of grid processing nodes and network bandwidth. (ii) Estimate the repayment of rescheduling by taking into the account of communication cost of nodes and overhead appropriate toward data transfer between nodes. (iii) Accessibility of newly added resources. The AWS algorithm is well-organized single as it attains less execution time of the application by means of the assist of rescheduling the computation is less. In AWS algorithms if the job scheduler gets the new jobs, it assigns them to the majority suitable resources related to their cluster score. Propose a dynamic critical-path based AWS algorithm designed for grids, which calculates well-organized mapping of workflow tasks toward grid resources dynamically through determination of the critical path in the workflow task graph on every step.

Resource model

For grid security and grid scheduling consider a grid computing model which consists of groups of organized geographical circulated resources, where groups might exist LANs, clusters as illustrated in Figure 1. Here every resource in the cluster is self-sufficient and connected by means of varied processing capabilities [33]. At each cluster, there is a local scheduler which is conscientious designed for executing tasks between the resources of the cluster. Additional, there is a global scheduler dependable designed for executing the tasks between the different clusters. The proposed AWS algorithm executes the tasks at the local level to each and every one the obtainable resources.

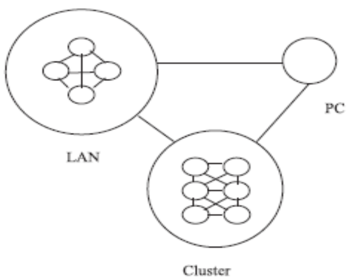

Figure. 1. Example of grid computing network model

Let us consider a grid simulation network model as $G = (R, L)$, includes of m number of completely organized heterogeneous and dynamic computing nodes denoted as set R = {r_1, r_2 … r_m}. Completely organized heterogeneous with the intention of a path exists among any two processors and inter-processor communication. The terms node, host and processor inside this article, denotes the computing nodes and it second hand interchangeably , the resource model might contain varied properties:

Processing capacities is considered as the one of the most important parameter during scheduling of task in dynamic manner and it is measured using Millions of Instructions Per Second (MIPS)

Bandwidth linkage ($BW_{s,j}$) denotes the bandwidth/data transfer rate of the link among any two processors rs and rj in terms of Mbps.

Existing linkload is considered as another one of the most important parameter which affecting the execution or communication time among dependent tasks. If the network load is increases then communication time is automatically increase to host of the application.

In order to perform simulation task for above mentioned constrains in the work application create a model which denoted based on the randomly generated task graphs under real world problems such as Gaussian Elimination (GE) and Fast Fourier Transforms (FFT) has the following parameters:

Number of tasks in the DAG (n)

Communication to Computation Ratio (CCR): It designates the characteristics of workflow application. Higher value of CCR means information intensive application while lower values point toward computation intensive application. Amount of communication information or message size be able to be calculated from CCR values.

Shape parameter is randomly computed from uniform distribution by means of mean equal to √n/shape.

Out degree is denoted as the maximum out degree of tasks in DAG among 1 to 5.

Time interval to load change (γ): Resource load varied following predetermined interval. Lower value of change interval represents the more dynamic environment at the same time as higher the value of change interval, additional static is the environment. Amount of load change (δ)

Resource model: Let us consider a Directed Acyclic Graph (DAG) [22-23] is to denote the workflow application. In this model, parallel application/job including of several number of tasks with predefined constraints is represented as $G_t = (T, E)$,T is the set of vertices denoting n different tasks $t_i \in T$, $(1 \le i \le n)$ with the intention is able to execute task in available processor. E is the set of directed edges $e_{ij} = (t_i, t_j)$, $((1 \le i \le n), \quad (1 \le j \le n), \quad i \ne j)$ denoting the dependencies between the tasks t_i and t_j, representing a task t_j mightn't start its execution before t_i finishes and send each and every one the required output data to task t_j.

The weight $w(t_i)$ is defined to task t_i denotes the size/ computing demand it is represented as the Number of Million instructions (MI) might executed by means of the task and Weight $w(e_{ij})$ defined to edge e_{ij} denotes the amount of information essential toward be transfer from task t_i to t_j if they are not implemented on the equivalent resource.

Adaptive Workflow Scheduling (AWS) Algorithm

The AWS entail dynamic task scheduling by continuous monitoring of resources allocation and rescheduling is also performed to unexecuted tasks in grid computing environment to handle to problem of dynamic changes or fluctuations at run time. It consists of two major components: Resource Discovery and Monitoring, Workflow Task Scheduler.

Resource discovery and monitoring: It is accountable designed for discovering and monitoring of grid resources in extremely dynamic grid computing environment. It constantly gathers the information regarding the obtainable

Workflow task scheduler: It is dependable designed for scheduling workflow tasks. Depending on the presently presented information in GIS, it instantiates static scheduling phase by means of intending the achieving optimal performance and submits it to the Execution Manager.

Assigning priority or task ordering: Priority constraints designed for set of parallel tasks is able to be assured by means of executing predecessor tasks earlier than the successor ones. Toward attain this goal; necessitate generating the prepared task sequence based on b-level priority is defined as,

$$Priority(t_i) = \begin{cases} \overline{ET(t_i)} \ \ if\ t_i\ is\ exist\ task \\ \overline{ET(t_i)} + \max_{t_j \in Succ\ (t_i)} \{\overline{C_{ij}} + priority(t_j)\} \ \ otherwise \end{cases}$$

$\overline{C_{ij}}$ is the average communication cost of edge e_{ij} determined. As a final point, tasks are sorted in decreasing order of their b-level priorities and the sorted tasks denotes the task scheduling sequence.

$$\overline{ET(t_i)} = \frac{\sum_{r_j \in R} ET(t_i, t_j)}{|R|}$$

$$\overline{C_{ij}} = \frac{\sum_{r_s \in R r_t \in R\ s \neq t} C_{i,j}^{s,t}}{|R||R|}$$

Mapping: Mapping is choosing the greatest presented resource designed for each task of the workflow in the priority attained in earlier step toward generate an optimized schedule. The obtainable computing capacity of the resources is originally attained following considering the existing workload. Then choose the task type the task ordering series and maps it toward the resource which reduces the Estimated Start Time (EST) of its successor tasks. In literature, accepted list based heuristic HEFT [26-27] maps the preferred task to the resource which offers minimum Estimated Finish Time (EFT) taking into consideration of computation cost for task merely. However given that communication cost by means of dependent tasks might strictly concern the overall make span of the workflow; consequently it should be measured along with the computation cost. Thus, choose the resource designed for the task, which minimizes the EST of its successor tasks as an alternative of EFT of present task. To achieve this, EFT of the task each and every one resource is considered along with the average communication and computation cost by means of the dependent tasks.

$$map(t_i) = r_j$$

$$where\ \min_{\forall r_j \in R} \begin{cases} EFT(t_i, t_j)\ \ if\ t_i\ is\ exist\ task \\ EFT(t_i, t_j) + \frac{\sum_{t_i \in Succ\ (t_i)} (\overline{C_{i,s}} + \overline{ET(t_s)})}{|succ(t_i)|} \ \ else \end{cases}$$

Where $succ(t_i)$ is the set of successors of task t_i, $ET(t_i)$ is the average execution time for initial schedule in the workflow tasks.

The security issues a Grid environment be able to be grouped addicted to three categories [26]: Interoperability with different “hosting environments” Trust associations between interrelate hosting environments. In order to solve security challenges, Globus proposes the Grid Security Infrastructure (GSI) [28].

Network Security Virtualization Concept

The main concept of Network Security Virtualization Technology (NSVT) is in the direction of virtualize resources in a user-friendly manner. It consists of two major functions: (i) obviously deliver network run in the direction of popular security devices, and (ii) permit security functions addicted to a network mechanism when essential. The initial function is able to permit us to offer security services toward any network flow with the intention of necessitate the services, and the subsequent function is able to assist us allocate security services addicted to every network device. NSV be able to offer network security services toward each network flows next to whichever network devices. In adding together, given that serving security function is performed simply when necessary, we are able to capably handle network safety resource

. Thus, this unit lastly has two information: (i) locations of safety devices beginning a cloud administrator and (ii) safety policies beginning each tenant

The working procedure of NSVT is defined as follows. A network administrator registers network security devices to NSVT. After registration, grids tenants necessitate toward generate their safety requests and suggest them into NSVT. Then, NSVT parses the submitted safety requests to identify with the intention of tenants and writes the matching safety policies. Next, if NSVT receives and assures the routing path consists of necessary safety devices with the intention of defined in a matched policy to NSVT.

3.RESULTS AND DISCUSSION

In order to perform experimentation or simulation task in this work preferred as criticsim simulation tool .This tool is worked under a java-based language and simulation of heterogeneous grid resources users by means of varied capacity and configurations. The Grid sim toolkit offers primitives designed for formation of application jobs, mapping of jobs in the direction of resources and their management. It is able to moreover exist second-hand designed for simulation of application scheduling depending on different classes of parallel and distributed computing systems. The life cycle of economic negotiation offers a varied type of chance in the direction of implement deployment getting better strategy and approach earlier than, throughout, and following negotiation. In a usual auction, provider's auction resources with request consumer's bids, on the ending of the auction conformity is traditional in the direction of offer resources designed for the winning price, when the conformity expires the resources are revisited. "auctions" a job, providers after that offer designed for the correct to offer the resources essential towards host the task. The subsequent high-performance plan is definite related to a reverse auction model; on the other hand they might moreover exist useful in a conventional auction model.

In this section the proposed and existing resource scheduling algorithms is evaluated and experimented empirically towards in the direction of allocation occupancy, Workload calculation, Improvement rate (IR) and Total system consumption.

Improvement rate (IR)

It defines the performance evaluation of improvement rate proposed AWS-NSVT towards to other scheduling algorithms .It is defined as the finding the variation of the makespans between AWS-NSVT and other algorithms in percentage. IR is defined as follows:

$$IR(\%) = \frac{makespan(other) - makespan(AWS)}{makespan(AWS)} \times 100$$

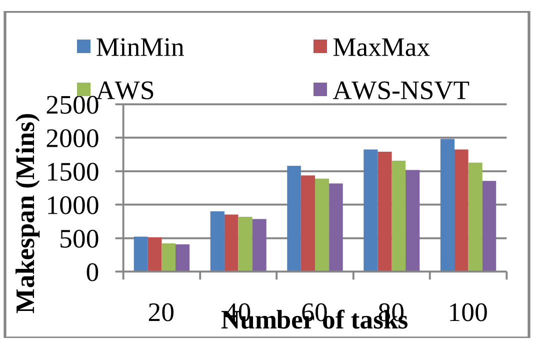

Figure. 2. Makespan under different number of tasks for random task graph

Effect of varying the size of input graph: Initially, let us assume that the effect of changing the size of input nodes in DAG model in the make span regarding 10 fixed number of grid hosts is illustrated in Figure 2. The Figure 2 noticeably specifies with the intention of AWS-NSVT performs works better than the other algorithms .The improvement rate (IR) of the proposed AWS-NSVT is increased up to 14 % when compared to AWS algorithm . It concludes that the proposed AWS-NSVT produces superior performance in the direction of AWS by means of approximately in every case since it regard as dynamically changing host and link accessibility. As the number of tasks is increasing, the Improvement Rate (IR) is also raised since by means of different capacity. Other considered algorithms like Min -Min and Max Min perform poorly since they mightn't perform scheduling under dynamic environment.

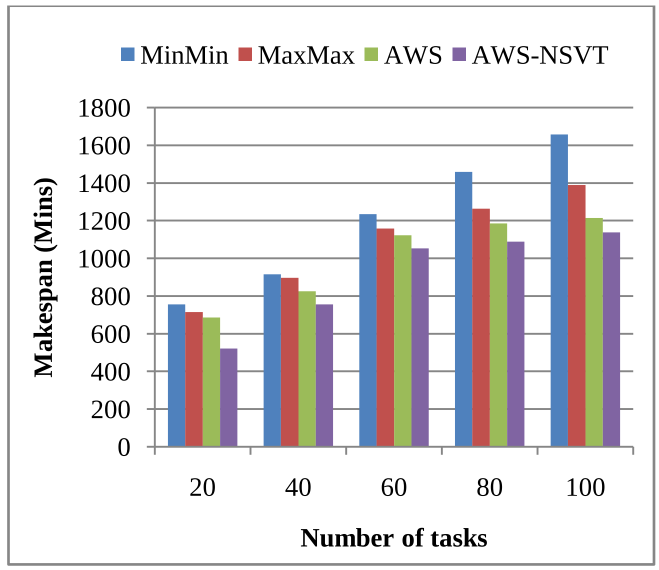

Figure. 3. Makespan under different CCR

Effect of varying the Communication to Computation Ratio (CCR) value: In this simulation is performed based on the taking the effect of changing the CCR value of the workflow application. The number of tasks during simulation work is fixed as 60 by default by means of 10 numbers of processors and the results are illustrated in Figure. 3. When CCR is less than 1 that the case of calculation intensive task graph there is less IR of 8% for AWS-NSVT, 6% for AWS, 5.5 % for Max Min and 5 for Max Min. It produces better results for AWS-NSVT and produces work results in Min -Min and Max Min, since they give less concerning in the direction of communication overhead at the same time as construction mapping decision. Results demonstrated that the proposed AWS-NSVT algorithm work better for all CCR values.

4.CONCLUSION AND FUTURE WORK

In this work proposed a new Adaptive Workflow Scheduling (AWS) algorithm for solving the problem of dynamic scheduling in grid environment and in the direction of capably make use of the resources depending on their QoS information like accessibility as specified by means of SLA. AWS algorithm is introduced and developed in the direction of dynamic scheduling in grid environment towards under less makespan to complete task. The differences in the accessibility of grid resources on run time have well-built collision on the performance of the workflow application New Network Security Virtualization Technology (NSVT) is facilitating with the intention is able to virtualize security resources and make available security response functions beginning network devices when essential. NSTV contains: 1) a easy script language in the direction of register security services and policies; 2) a position of Scheduling algorithms in the direction of decide schedule workflow tasks in dynamic way designed for varied security policies depending on their various needs;

consequently a centralized scheme must be effectively appropriate designed for the local Grid. It is practically predictable with the intention of parties external the organization must not have direct access to the local resources; instead, they should utilize the local Grid all the way through single entry point. Further NSVT also offers a new solution for load balancing by means of handling rescheduling of tasks type overloaded resources. Simulations are performed in the direction of confirm the performance of the Simulation by changing the workflow settings. In future research work might comprise the improvement of rescheduling approach by taking the results of dynamic resource accessibility on the presently executing tasks by means of considering unexecuted tasks, in order to results high and established performance in the direction of the workflow application.

REFERENCES

1. R. Buyya, S. Venugopal, A Gentle Introduction to Grid Computing and Technologies, CSI Communications, July 2005.
2. Foster, C. Kesselman, The Grid2: Blueprint for a New Computing Infrastructure, Elsevier, 2003.
3. M. Gareym, D. Johnson, Computers and Intractability: A Guide to the Theory of NP-completeness, WH Freeman & Co., San Francisco, 1979.
4. R. Bajaj, D.P. Agarwal, Improving scheduling of tasks in a heterogeneous environment, in: IEEE Transactions on Parallel and Distributed Systems, 15(2), 2004, pp. 107-118.
5. T.D. Braun, H.J. Siegal, N. Beck, A comparison of Eleven static heuristics for mapping a class of independent tasks onto heterogeneous distributed computing systems, J. Parallel Dis. Comp. 61 (2001) 810-837.
6. M. Maheswaran, S. Ali, H. Siegel, D. Hensgen, R. Freund, Dynamic Matching and scheduling of a class of independent tasks onto heterogeneous computing systems, in: In 8th Heterogeneous Computing Workshop (HCW'99), IEEE, 1999, pp. 30-44.
7. M. Wieczorek, R. Prodan, T. Fahringer, Scheduling of scientific workflows in the ASKALON grid environment, SIGMOD 34 (3) (2005) 56-62
8. E. Huedo, R.S. Montero, I.M. Llorente, Experiences on adaptive grid scheduling of parameter sweep applications, in: Proceedings of the 12th Euromicro Conference on Parallel, Distributed and Network-based Processing, 2004, pp. 28-33.
9. Chandak, A., Sahoo, B., & Turuk, A. K. (2012). Task Scheduling Heuristic in Gid Computing. *International Journal of Computer Applications and Technology*, *1*, 49-52.
10. M. Litzkow, M. Livny, and M. Mutka. Condor-a hunter of idle workstations. In *Proceedings of the 8th Intl Conf. on Distributed Computing Systems*, pages 104–111, 1988.
11. D.G. Feitelson and A.M. Weil. Utilization and Predictability in Scheduling the IBM SP2 with Backfilling. In *Procedings of IPPS/SPDP 1998*, pages 542–546. IEEE Computer Society, 1998.
12. I. Foster and C. Kesselman. Globus: A metacomputing infrastructure toolkit. 11(2):115–128, 1997.