International Journal of Computer Applications (0975 – 8887)
Volume 115 – No. 4, April 2015

Realization of Resource Allocation using Centralized Approach by Means of Directed Graph

Sridevi S Taradi
Department of Information Technology
Vidyalankar Polytechnic,
Wadala

ABSTRACT
The mechanism of resource allocation plays a vital role in maximizing the network utilization and serving the requesting nodes. This idea can be effectively demonstrated using the concept of network virtualization. Through Network virtualization an environment conducive to perform tests using similar network conditions , applications and dependencies can be created to ease and enhance implementation of network elements in a topology. One important challenge on network virtualization is the efficient use of the physical resources. To accomplish such efficient use the management of the physical resources should be transparent to the applications running within the virtual networks. In this paper we use directed graph to simulate the network and to reallocate virtual network resources along different physical nodes in order to understand the dynamics of resource allocation.The system generates analytical results through graph plotting and statistical analysis of network parameters**.**

Keywords
Resource allocation,Network Virtualization ,directed graph

1. INTRODUCTION
Resource allocation is the scheduling of activities and the resources required by those activities while taking into consideration both the resource availability and the project time. The use of discrete event simulation packages acts as an aid to modeling and performance evaluation of computer and telecommunication networks, including wireless networks[11].

A resource allocation problem is considered in the sense that the number of processes competing for a particular resource at any time instant is bounded and also at any time instant the number of resources it may get is bounded. One of our solutions assumes the existence of an underlying efficient handshake communication system[6],[8].

The proposed system aims at achieving freedom from starvation and fairness[1]. No process must be waiting for another one to complete beyond a stipulated time and in order to avoid starvation. The property of fairness is essential to be implemented because the must be given access to the resource in the correct order or in the order in which the request is being made.[5]

At the centralized approach, the coordinator holds a list of processes that they are requesting the CS ; grants requests in first in first out order. In this technique, a centralized allocator keeps track of all the available resources. All entities send messages requesting resources and the allocator responds with the allocated resources. The main advantage of this scheme is simplicity.

The reliability of the underlying communications network should be considered while accessing the resources. Messages are not lost or altered and are correctly delivered to their destination in a finite amount of time. Messages are delivered in the order they are sent. There is no message overtaking. Messages reach their destination in a finite amount of time, but the time of arrival is variable[3].

2. ALGORITHM
Variables

G(V,E) where V is a set of vertices and E is a set of edges.

V={P,R} where P is requesting process and R={r1,r2,r3….} where R is the set of resources.

q- queue length

a – number of resources available

ql- queue limit

C- Coordinator

Initialisation

q=0 , a=1 ,ql=2 r=0

Entry Section

Pi[request] -> Ri , A process Pi sends request for the resource Ri

Allocation

Case i>Resource Available

If r<a , the resource is granted to the process

{

C[success]->P

r++ and a—

}

Else

Case ii> Fail

{

C[message=fail]-> Pi && await Pi until r=0

q++ until q<=ql

while(q>ql)

message:=DROPPED

q=ql

}

Deallocation

{

Pi[release]->Ri

C[grant] ->Pr

}

Exit

q=0 && r=0

3. REALISATION

The algorithm is realized using the queueing principle in which the requests are queued before being sent to the resource allocator. Each process maintains an order : the process send a request , the request is being queued before being sent to the resource allocator[9].

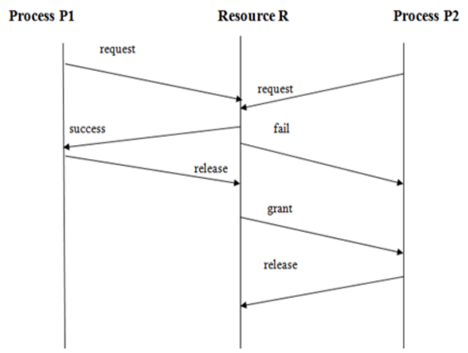

Figure 1: Handshake Mechanism

The resource allocator receives requests from various processes. And based upon the type of scheduling the processes are served.

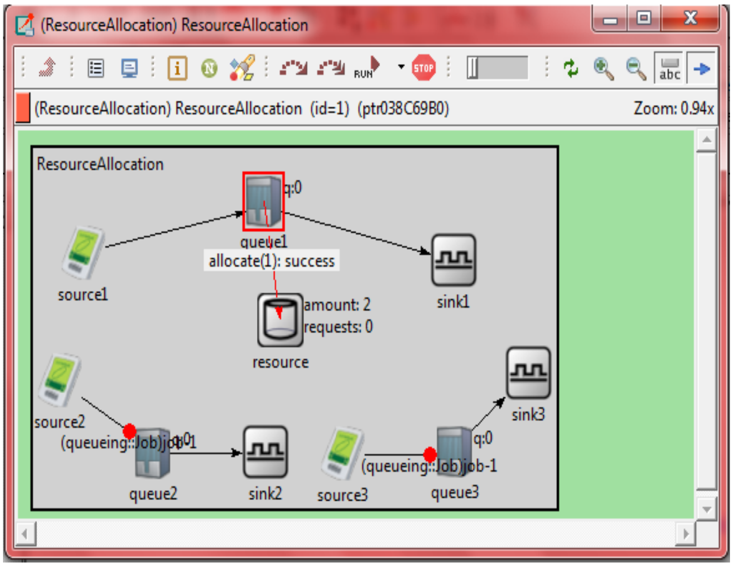

Figure 2: Checking status of availability of resource

A log is maintained by the resource allocator which contains the number of requests and resources available.

The amount list is incremented whenever a resource is freed and decremented whenever a resource request is sent. [1]Once a resource is found available it a grant message is sent by the co-ordinator.

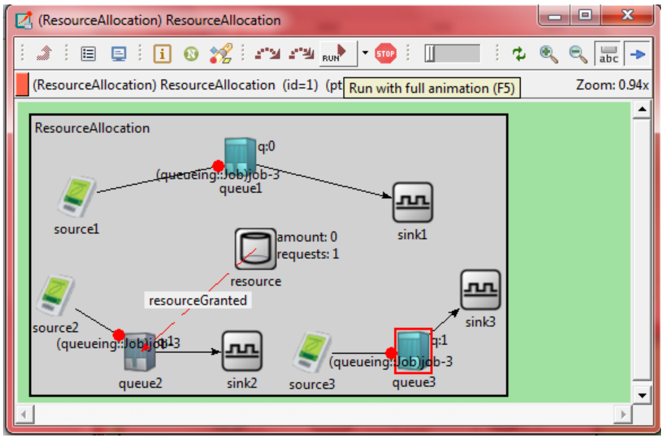

Figure 3: Allocation of resource upon availability of resource

While one process is being served and no resource is available if another process one sends a request it is denied and delayed until the resource is released by the former one.The queue length can be limited by putting a restriction on its maximum length. Any request made by a process will be dropped in order to avoid starvation.

4. ANALYSIS

4.1 Performance at each node can be monitered using the key measurement parameters. Key measurements produced are:

- Queue Length
- Average Wait Time
- Idle Time
- Total Wait Time

Table 1: Statistical Analysis of performance parameters

Sources	Parameter name	Count	Mean	StdDev
Terminal1	Queuelength	3	0.3333	0.577
Terminal2	Queuelength	4	0.5	0.577
Terminal3	Queuelength	4	0.5	0.577
Dest1	Lifetime	1	1.0	n.a
Dest2	Lifetime	1	2.0	n.a
Dest3	Lifetime	1	3.0	n.a

4.2 Statistical analysis can be done by applying mean, sliding window average etc as in fig 5.

4.3 Sequence diagram can be generated by recoding the flow of events ; it helps in understanding the order of execution of processes as in fig 6.[10]

4.4 Scheduled Events

Jobs – Requests are represented through directed graphs flowing from one node to another. Messages – The state or form of the process request is represented through messages.

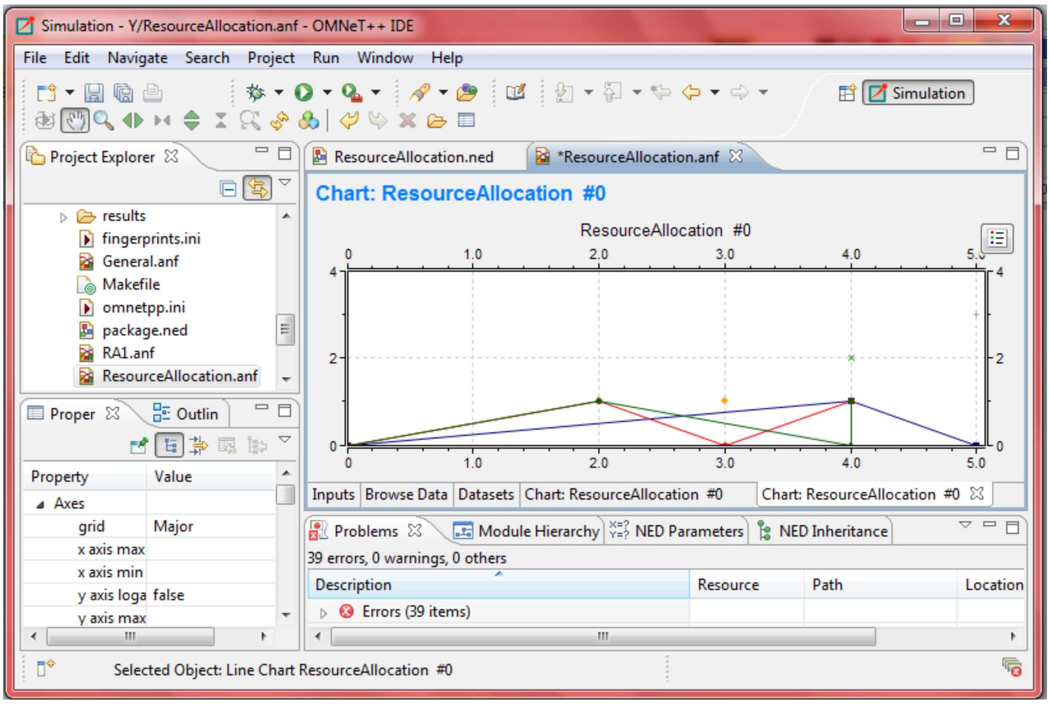

Figure 5: Analysis of Resource Allocation Chart

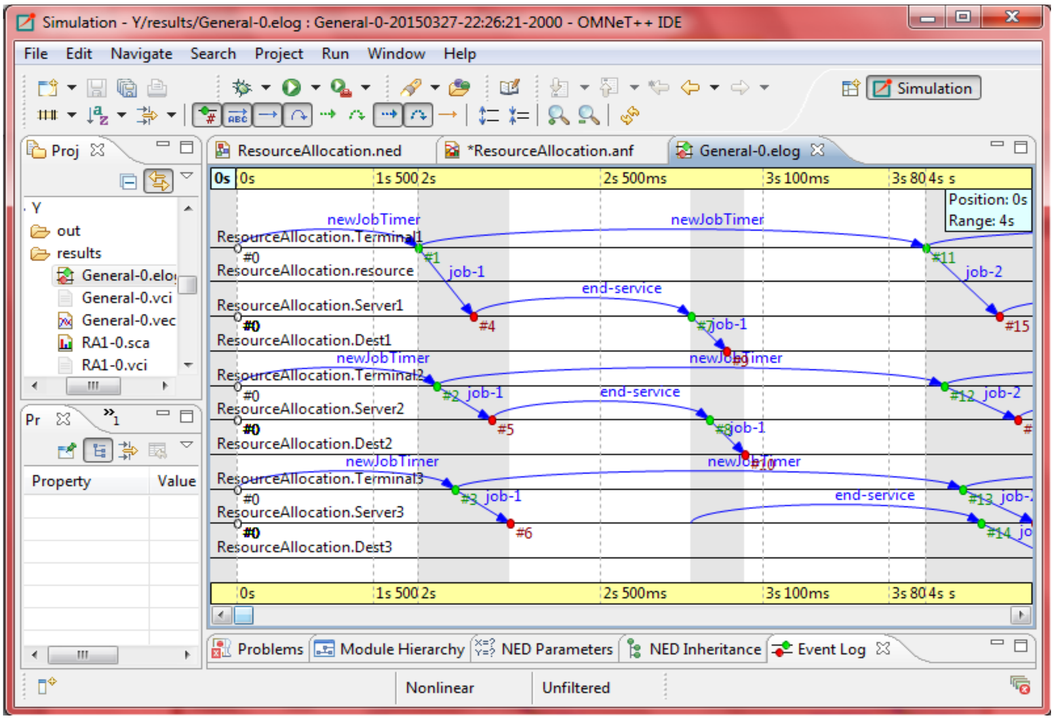

Figure 6: Sequence Diagram showing flow of events

5. CONCLUSION

Over the last decade modelling and simulation of systems have attracted a great deal of attention. Resource sharing, parallel processing, system availability and communication are the four major reasons among others[4]. By distributing a computation among various sites, processes are allowed to run concurrently and to share resources, but still work independently of each other[9],[14],[15]. The paper describes the design and implementation of centralized approach of resource allocation and its simulation. The design allows dynamic simulation scenarios to be created, where requests, grants and messages can enter and exit the scenario during the course of a simulation run[9],[13]. Varying parameter values give different outputs which help in understanding the working of the system in different circumstances[12]. The model presented in this research work raises number of challenges for further research such as validate the feasibility of proposed models in real environments by extending the mechanism by providing means for a domain to redirect tasks across several domains and incorporating more sophisticated load forecasting technique .

6. REFERENCES

[1] Alex A. Aravind, IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, VOL. 22, NO. 6, JUNE 2011

[2] J.H. Anderson, Y.-J. Kim, and T. Herman, "Shared-Memory Mutual Exclusion: Major Research Trends Since 1986," DistributedComputing, vol. 16, pp. 75-110, 2003.

[3] Aravind and W. Hesselink, "A Queue Based Mutual Exclusion Algorithm," Acta Informatica, vol. 46, pp. 73-86, 2009.

[4] D. Fooladivanda and C. Rosenberg, "Joint resource allocation and user association for heterogeneous wireless cellular networks," Wireless Communications, IEEE Transactions on, vol. 12, no. 1, pp. 248–2257, 2013.

[5] S. Adabi, A. Movaghar, A. M. Rahmani, H. Beigy, Market-based grid resource allocation using new negotiation model, Journal of Network and Computer Applications, (2012).

[6] S. Wang, F. Huang, M. Yuan, and S. Du, "Resource allocation for multiuser cognitive OFDM networks with proportional rate constraints," International Journal of Communication Systems, vol. 25, no. 2, pp. 254–269, 2012.

[7] H. Attiya, R. Guerraoui, D. Hendler, and P. Kouznetsov, "Synchronizing without Locks Is Inherently Expensive," Proc.25th ACM Symp. Principles of Distributed Computing (PODC '06),pp. 300-307, 2006.

[8] M. Ben-Ari, Principles of Concurrent and Distributed Programming.Addison-Wesley, 2006.

[9] http://www.omnetpp.com

[10] http://canu.informatik.uni-stuttgart.de/mobisim.

[11] J. Yeo, D. Kotz, and T. Henderson, \CRAWDAD: a community resource for archiving wireless data at Dartmouth," SIGCOMM Comput. Commun. Rev.,vol. 36, no. 2, pp. 21{22, 2006.

[12] Varga, \The OMNeT++ Discrete Event Simulation System," in Proc. of European Simulation Multiconference (ESM2001), June 2001.

[13] http://www.idosi.org/wasj/wasj23(8)13/8.pdf

[14] http://home.ubalt.edu/ntsbarsh/simulation/sim.htm#ri ntroduction W. , S. Sroka, V. Handziski , A. K opke, and H. Karl, \A Mobility Framework for OMNeT++," in Proc. of 3rd International OMNeT++ Workshop, Jan. 2003.

[15] K. Czajkowski, I. Foster, C. Kesselman, Agreement-based resource management, Proc. IEEE, Mar. 2005, 93 (3) (2005) 631-643