ISSN:2319-6890)(online),2347-5013(print)
20 May 2016

Network Aware OpenStack Nova Scheduler

Nanditha A
MTECH, Computer Science & Engineering
Cambridge Institute of Technology
Bengaluru, India
nanditha.14scs16@citech.edu.in

Shivakumar Dalali
Associate Professor, Computer Science & Engineering
Cambridge Institute of Technology
Bengaluru, India
skumardalali.cse@citech.edu.in

***Abstract*—Cloud computing is a rapidly growing technology adopted by many companies around the globe. This growth led to huge competition in the business among the cloud providers. Cloud Providers are continuously striving to provide better cloud services to cloud consumers at reasonable cost. Virtualization is the key driver which helps in the optimal usage of cloud resources of datacenters. The virtual machines or instances running on hypervisors should be optimally distributed in the datacenters to provide better performance to the cloud consumers. For optimal placement of instances, a good scheduling algorithm needs to be selected. The scheduling algorithm available considers compute, storage and memory utilization while placing the instances on servers in the datacenter. But as network is a strong pillar in any technology, it should be a compulsory factor in the scheduling algorithms. The proposal of this scheduling algorithm is to include network factors into consideration. OpenStack which is a cloud Software has been used to demonstrate this proposal. OpenStack's Nova scheduler includes only compute and storage filters in its filter scheduler. Network filter is added for optimal placement of instances.**

Keywords—Cloud Computing, Dynamic migration, Hypervisor, Network-Aware, Nova, OpenStack, Scheduling, Virtualization

I. INTRODUCTION

In today's world, different types of data is getting accumulated rapidly. All the people around the world want to save and access their data wherever they are and from all types of devices. So there is always a requirement of central data store to store these users' data. Cloud computing is the technology that helps to build the datacenter and save the client's data for easy access. Data need to be stored and accessed whenever required, so to do this network plays a major role.

Cloud computing has delivery modes such as Software-as-a-Service (SaaS), Platform-as-a-Service (PaaS) and Infrastructure-as-a-Service (IaaS) and has been deployed as private, public, community and hybrid models. Depending on the client's request, cloud services are provided with pay-as-you-go model. Building user's own cloud always require a lot of administration efforts and cost. In current market, there are many cloud providers managing datacenters to provide cloud services. There is a huge competition among them for providing the better services in reasonable cost. Having less physical resources and running multiple virtual machines on them and providing optimal service to the consumers is very challenging. Maintaining the balance between performance and cost is challenging for the cloud providers.

OpenStack is the open-source cloud software, helps the companies to build their own cloud. Compute, Storage and Network are the three important parts of the OpenStack. Virtualization is the key driver for the success of cloud computing. Presently Nova Scheduler schedules new Virtual machines (VM) on physical servers depending on the CPU utilization, RAM and Storage available in the physical servers. But Network also plays a major role in the virtual machine placement and influences the performance and user experience. Virtual machines can be migrated from one server to another depending upon the compute, network and storage factors, to provide optimal usage of resources. There are many virtual machine placement (VMP) algorithms are proposed considering different factors while placing the virtual machine. Network is one of the very important factor need to be considered, which is not taken into consideration in OpenStack Nova scheduler.

The proposal of a network filter algorithm considers network during initial placement and dynamic placement of virtual machines in OpenStack Nova (filter scheduler and weighing). The network interface card status and network bandwidth are considered in initial placement of virtual machines. Along with this a new agents are created for dynamic migration. Dynamic migration is started when data network card in any of the physical servers goes down; all the virtual machines from that server are migrated to other servers.The paper is organized as follows, related work and literature survey is presented in Section II. The proposed system, architecture and the algorithm is described in Section

III. Testing methods and results are explained in Section IV. Final section is Conclusion and Future Work.

II. Related Work

Cloud Computing is a growing technology which gives opportunity for technical research and innovations. There are lot of research is going on considering different factors to balance datacenter performance and maintenance cost. In paper [1], traffic and power are the factors considered in placing virtual machines. The virtual machines running in the datacenters are compute- intensive and some are network-intensive. So while scheduling the virtual machines, traffic due to the communication between the virtual machines running on the different physical servers is not neglected. Traffic consumes network bandwidth and introduces latency in the data communication. Due to huge demand in the cloud services, datacenter are becoming bigger with more physical resources. The power consumed by all the physical resource is increasing, in turn increases CO2 emission and global warming [2].In order to save energy and environment, virtual machines are consolidated in such a way to reduce the number of physical resources required to run them. Energy aware virtual machine placement is classified as Dynamic Server Pool Resizing (DSPR) and Dynamic Processor Scaling (DPS). DSPR save power by turning off the idle and under-utilized servers in the datacenters. On the other hand, DPS save energy by changing the server clock speed with the help of special hardware.Daniel et.al [3] proposes a VMP algorithm considers the online traffic matrix in the network while allocation/ reallocation of VMs on the hosts. VMP algorithm correlates the VMs by checking the traffic among servers; aggregate those servers into clusters of similar traffic patterns. VMP algorithm fits the clusters in separate partitions, which should have enough memory and CPU to manage all the VMs running on it. VMP algorithm runs in four stages: In first stage, Data Acquisition collects CPU and memory allocation of each virtual machines and traffic between them running on all the servers of the datacenter. In second stage, Server partitioning is done by grouping the servers placed on the racks, CPU and memory usage is totalized .In third stage, [4] clustering of virtual machines are done which are frequently communicating using the Clustering algorithm[5]. In final stage, the algorithm outputs the location of virtual machine placement on the server and starts migration.Jing Tai Piao and Jun Yan [6] proposes a virtual machine placement algorithm for optimal data access, by creating virtual machine on physical hosts with less transfer time to the required data. And also migration is started dynamically when the data transfer time of any virtual machines reaches certain threshold due to unstable network. In this paper, data intensive application running in a virtual machine of a server, access the required data continuously stored on some other server of the datacenter, disturbs the network I/O performance of the datacenters. The algorithm proposed helps in the placement of related data-intensive virtual machines. Then dynamic network latency is introduced is handled by migration of related virtual machines [7]. This helps to maintain the Service Level Agreement between cloud service providers and cloud consumers.Sema Oktug et.al [8] explains that while VM placement, computational resources are only taken into account; neglecting the cost of network [9]. To reduce the networking cost, communication pattern of VMs are considered. Frequently communicating VMs are placed in the same rack or very closer by studying the traffic between them. Clustering Algorithm is proposed to reduce the traffic between racks, in turn reducing the communication delay .The arrangement of virtual machines and networking elements should be in such a way of saving energy. A fast clustering technique is proposed to cluster the frequently communicating VMs in the same group. The clustering technique is also applied for dynamically studying of changing traffic rates.In this paper [10], network aware scheduling in nova scheduler (OpenStack) [11] is discussed. OpenStack is open source framework to build public and private clouds. Usually the datacenters are distributed in different geographical regions. The two VMs communicating continuously placed in different geographical regions datacenter incurs heavy traffic. OpenStack Nova scheduler is update with new Nova API, collects the entire information about the VMS and traffic from Neutron. An OpenDayLight controller is used to collect the network topology data and communicate with the Nova scheduler. The new scheduler is designed that takes a group of VMs instead of one virtual machine at a time. All related VMs are placed in the same physical host for reduced communication rate. For grouping the VMs, the information is collected from Neutron.After the initial placement of instances on the hosts, traffic and workload changes dynamically this requires live migration of instances accordingly. Shangruff et.al [12] discusses how the live migration happens in cloud. Migration is the movement of virtual machine from source host to target host. If it is live migration, then user experience and network connections will not be affected. The downtime of VM is zero or minimal. The benefits of doing live migration is maintenance of servers, workload balancing, reduce IT costs, server consolidation, disaster recovery and powering down the unused hosts. He also explains that Live migration happens in three phases: *Push phase:* The migration is started from source, VM is running while pages are pushed across new target .But VM is still running in the source, pages modified is resent. *Stop and Copy phase:* The VM is stopped and completely copied to target. The time taken to copy from source to target is known as downtime. Downtime of instance is few milliseconds depending on the application's memory size running on the source VM. *Pull phase:* The migrated VM starts in the target and pulls the remaining pages from source.

The VM running in the source is suspended.In OpenStack Summit 2015, Dulko et.al [13] presents a talk on Live Migration in OpenStack. Migrations in OpenStack are block migration and True Migration. Live Migration Process happens in five steps:

Pre-Migration: Instance is selected in compute node A, and target compute node B is selected by scheduler.

Reservation: Check the available resources in compute node B and reserve.

Iterative pre-copy: Memory Pages of the Instance is iteratively copied from compute node A to B.

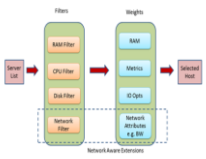
Stop and copy: Suspend the instance in compute node A and copy remaining pages to node B.

Commitment: Compute Node B becomes the primary for the instance.

The migration in OpenStack can be difficult in some cases such as instances with intensive memory workload, compute nodes with different CPU models, heavy network traffic and OpenStack does not allow performing any operations on instances during live migration. All the papers discuss those benefits of considering network during scheduling virtual machine in datacenters. It is important to consider network which helps in reducing IT costs, energy saving, reduce communication latency and improves user experience.

III. Proposed System

A network filter algorithm is proposed, is called by nova scheduler. It helps to identify the network link state of compute nodes. The basic check of the network needs to be done before running any advanced network aware scheduling algorithms. So this proposal can be added in OpenStack as a default filter in Nova scheduler (Filter and Weighting algorithm).



In OpenStack, the process of finding the correct compute node to launch a virtual machine considers CPU and memory allocated for each instance in every compute node. CPU is congestion by the number of cores available in the compute node and required by each instance. Memory (RAM) is measured by the available free memory in the compute node and memory required by each instance.

Compute, Network and storage are the three important pillars in the OpenStack software in managing the cloud infrastructure. Nova is the brain of the OpenStack manages the life cycle management of virtual machine. Nova Scheduler is the service chooses the compute node for running virtual machine [13].User requests for a new virtual machine creation through Horizon, OpenStack dashboard as shown in figure 1. Nova API picks up the request from the queue and forwards to Nova Controller. Nova Controller requests the Nova scheduler for physical host name. Nova scheduler runs the filter scheduler algorithm to find the server to host a new instance. Selected physical server details are sent as response to Nova controller. Nova controller sends the virtual machine creation request to the selected physical host (server).There is a nova agent running in all hosts updates the nova scheduler with all computational resources information.

If data network is down or network interface card is not working in any physical hosts, still nova scheduler does not filter out those hosts. Nova scheduler considers only compute, memory and storage availability in the physical hosts. Scheduling the new virtual machine will not have data connectivity to communicate with other virtual machines running on other physical hosts. So User applications running on it, fails to communicate with other servers. Then administrator has to debug the problem causing network failure.

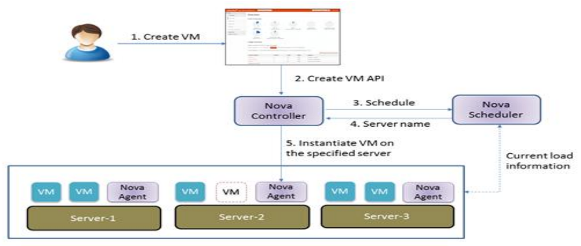


Fig. 1. Scheduling a Virtual machine on a physical host

A. *Enhanced Nova Scheduler with network filter*

Along with CPU, RAM and Disk default filters, Network filter should also be added to select the list of available physical hosts. Now Nova Scheduler is modified to consider network while filtering the list of physical hosts. The network factors like bandwidth is calculated and weight is assigned. Along with the other weights like RAM and Storage, network factors weights are added for each physical host. Now the final sorted list of hosts with weights is obtained. The topmost physical host in the sorted list with less weight is chosen by Nova Scheduler to launch a new VM.

Fig. 2. Network Filter and Weighting

B. *Initial Placement of Virtual Machines*

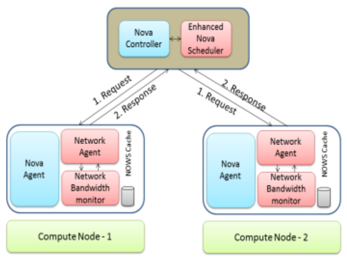


Fig. 3. Network Agents of compute nodes communicating with Enhanced Nova Scheduler

A request comes for launching a new virtual machine to Nova controller. As shown in figure 3, a nova agent is collecting CPU, RAM and disk resources information in every physical host (compute node). Similarly a Network Agent is created in all compute node or physical hosts. During filtering stage, Enhanced Nova Scheduler requests the list of physical hosts for network link state. Network agent responses the network interface card (NIC) state, whether it is up or down. If response is up, then that physical host is considered for the next stage. If the response is down, then that physical host is removed from the list of selected hosts for the next stage. A list of physical hosts is given with required CPU, RAM, storage resources and network link state is up.

The list of selected hosts after filtering stage is given weights depending on the available computational and networking resources required to start the requested virtual machine. The bandwidth monitor agent is used to find the bandwidth of each physical host and weights calculated. A sorted list of physical hosts is obtained from Scheduler. Nova controller picks the compute node with less weight; a new virtual machine is created on it.

Algorithm1 - Network Filter Algorithm for Filtering

```
Input: List of hosts running in the datacenter
Output: Filtered list of hosts satisfying the requirements
1.  for each host in host_list
2.          invoke the REST interface of the Network Agent to get the
            link State
3.          if link is down
4.                  Remove host from host_list
5.          else
6.                  Retain host in the host_list
7.  end-for
8.  return the updated host_list to Filter Scheduler filter function
```

Algorithm1 explains the filtering functionality of network agent. List of all physical hosts running in the datacenter is given as input. In each host, invoke the REST interface of the Network agent to get the link state. If link is down, remove from the master list. At last return the list of remaining hosts for weighing to filter scheduler.

A network bandwidth monitor agent is running in every host (Algorithm 2). Initial bytes sent and received to the particular NIC card of the physical host is captured at time t1.Sleep for specific duration. Final bytes are calculated again in each physical host or compute node. Bandwidth usage is calculated as in equation (1) and (2) in each physical host.

$$delta\_bytes= final\_bytes-initial\_bytes \qquad (1)$$

$$used\_Bandwidth = delta\_bytes/duration \qquad (2)$$

Algorithm 2 – Network Bandwidth Monitor Algorithm

```
Input: Host name
Output: Available bandwidth
1.  while (true)
2.          initial bytes = fetch the current bytes sent and
                received of NIC card
3.          Sleep for specific duration
4.          final bytes = fetch the current bytes sent and
                received of NIC Card
5.          delta bytes = final bytes – initial bytes
6.          BW = delta bytes/duration
7.          Write BW data to file
8.  end-while
```

The used_bandwidth calculated in every selected host is used in finding the available free bandwidth as shown in equation (3).The free bandwidth is obtained to assign weights as explained in Algorithm 3.

$$available\_bandwidth = NIC\_capacity- used\_bandwidth \qquad (3)$$

Algorithm 3 - Network Filter Algorithm for Weighing

```
Input: List of Host names after filtering
Output: Weights assigned
1.  for each host in host_list
2.          invoke the REST interface of the Network Agent to get the Link
            Used Bandwidth
3.          available_BW = NIC_Capacity – used_Bandwidth
4.  end-for
5.  return available_BW to Filter weigher for computing the weight of the
    host
```

The weight is calculated and normalized for network. The normalized weight of each is added with other weights to find the total weight of physical host as in equation (4).

$$Weight\_PhysicalHost1 =$$

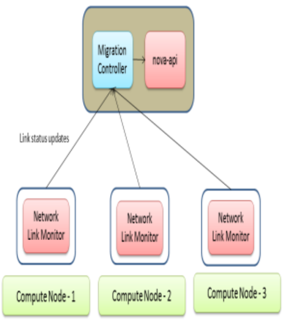
$$Weight1\_RAMWeigher*normalize(Weight1)+ Weight2\_CPUWeigher * normalize (Weight2) +$$

$$Weight3\_NetworkWeigher*normalize (Weight3) +..... \qquad (4)$$

The Nova Scheduler sorts the host in ascending order. Nova controller picks the first physical host in the sorted list for processing the new request.

C. Dynamic Placement of Virtual Machines

Network link state can change due to maintenance of servers, system crash, heavy traffic and energy saving. The user applications running in the virtual machines should not experience many problems. This will adversely affect the business of cloud providers in the market. OpenStack services will help the cloud providers in providing better user experience even in unexpected changes in the datacenter.



Network link monitor of compute nodes communicating with Migration Controller

Network link monitor is an agent running in all compute nodes. It monitors the network interface card state of the compute node as shown in figure 4. If data network fails due to NIC failure, network link monitor sends status update to Migration controller. Migration controller immediately responds by initiating the migration process. It informs nova-api for rescheduling the virtual machines of failure node.

One such scenario, due to some reason, network interface card may fail in some compute node. A user application loses connection with other virtual machines running on other compute nodes. To solve this problem, a Network Link Monitor agent is running in all the compute nodes. The network link state is checked in intervals of time. If NIC failure is found, network link monitor sends a signal to Migration controller running in the controller node as in Algorithm4.

Algorithm 4 - Network Link Monitor Algorithm

```
1.  while (True)
2.        Check the link state
3.        if nic == down
4.              Increment the counter
5.        else if nic == up
6.              Decrement the counter
7.        Sleep for configured interval
8.        if counter>= threshold_count
9.              Send host network down event to Migration_controller
10. end-while
```

Migration controller agent present in the controller node responds to the signal sent by the Network link monitor agent. Once network down state is received, Monitor controller agent finds the list of virtual machines running in the network-failed compute node. Live migration [14] of virtual machines from failed compute node to other compute nodes is started by scheduling with Enhanced Nova Scheduler. This helps in maintaining optimized workload balancing in the datacenter. Algorithm 5 explains the functionality of Monitor Controller Algorithm. Again when network is up, the agent signals the controller for its activeness in the datacenter.

Algorithm 5 – Migration Controller Algorithm

```
1.  Create A list of network down servers
2.  function Receive host network down message (hostname)
3.        Find the list of virtual machines running on that host
4.        Evacuate all the virtual machines to other host through
          migration
5.  end-function
6.  function Receive host network up message (hostname)
7.        Remove the host from the list of network down servers
8.  end-function
```

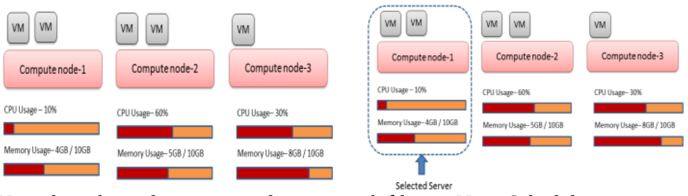
IV. Experimentation

To experiment the scheduling mechanisms of Enhanced Nova scheduler, the minimum hardware required is 2 switches, 1 NFS Server, 1 controller node, 1 network node and 3 compute nodes. All the servers are installed with appropriate OpenStack software and services [15] [16]. The setup is done as shown in figure 5. Compute nodes are connected to data switch and communication happens between them in data network. Controller node, NFS Server and computes nodes connected to management switch. OpenStack services communicate in management network.

A. ***Scenario1:*** *Initial Placement*

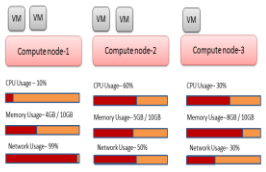
A tested scenario consists of 3 compute nodes. Compute node-1 has 2 instances running on it, consuming 10% CPU, 4GB RAM. Compute node-2 has 2 instances, consuming 60% CPU and 5GB RAM. Compute node-3 has 1 instance running on it, with 30% CPU and 8GB RAM usage.

Consider a new request for creating a virtual machine with 1 logical CPU, RAM -2GB through OpenStack Horizon. Now the datacenter has 3 compute nodes, mentioning the usage indicators as shown in figure 6(a) .The Nova scheduler schedules the creation of new instance in Compute node-1 considering the available CPU and RAM, neglecting the network as shown in figure 6 (b). A new virtual Machine runs network –intensive applications in compute node-1 suffer from network congestion and packet delay &loss.



Virtual machine placements without network filter in Nova Scheduler

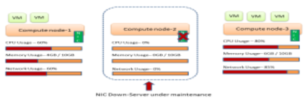
The Enhanced Nova scheduler proposed in our algorithm take care of network while scheduling the creation of new virtual machine. The Enhanced nova scheduler is updated with the network filter as one of its default filters. Our network filter solved the network problems caused to the compute nodes.Now we tested again the same scenario of initial placement of virtual machine with Enhanced nova scheduler. This time Nova scheduler considers network usage indicator along with CPU and RAM as shown in figure 7(a). Previously the filter scheduler has selected compute node-1.But now computes node-2 is selected for launching a new virtual machine as shown in figure 7(b). Now virtual machines running in compute node-2 do not suffer from network problems, because it still has more bandwidth to support network traffic of new VM.



(a)

(b)

Fig. 4. Virtual machine placements with network filter in Nova Scheduler



B. ***Scenario 2****: Migration*

If network interface card is made down or data network failed, virtual machines of compute node loses communication with other compute nodes virtual machines. The network link monitor agent identifies the failure and informs the migration controller. The migration controller starts live migration through management network.

Consider the network interface card of compute nodes-2 fails as shown in figure 8(a). The network link monitor service checks the failure of data network iteratively for some desired times for confirmation. After the confirmation, it informs the migration controller running in the master controller node.

Nova controller running in the controller node starts live migration. Nova controller commands enhanced nova scheduler to find the available hosts for the migration of the virtual machines from failed node. Nova scheduler finds compute node-3 as shown in figure 8(b) satisfies the requirement of computational and networking resources. The VM pages are transferred across management network with few milliseconds downtime. After the complete transfer, the memory and disk are cleared and compute node is sent for maintenance –figure 8(c).

The testing proves that the network is such an important factor. Network need to be considered as a default filter in the OpenStack Nova Scheduler. This helps the cloud providers to build more optimal cloud and provide services to consumers.

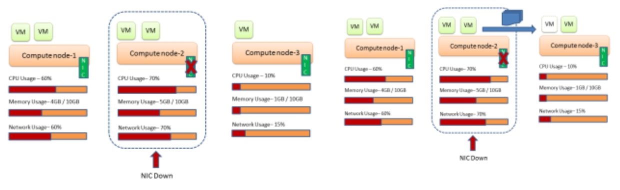
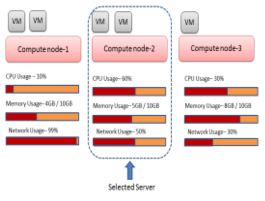


Fig: Live migrations to optimize the workload

V. CONCLUSION AND FUTURE WORK

The paper proposed a network filter algorithm, is added in nova scheduler. It helps to identify the network link state of compute nodes. The basic check of the network needs to be done before running any advanced network aware scheduling algorithms. So this proposal can be added in OpenStack as a default filter in Nova scheduler (Filter and Weighting algorithm).



The initial placement of virtual machine is now scheduled by enhanced nova scheduler. Now it considers all the three important factors of OpenStack. Compute, storage and network. Network bandwidth algorithm finds bandwidth usage pattern in all the compute nodes. Rebalancing of workload using live migration improves customer experience and VM performance.The future enhancement to the Network filter is to add more network metrics like network latency and number of hops. Live migration can be done studying traffic and communication pattern between virtual machines.

VI. REFERENCES

[1] Soonwook Hwang and Hieu Trong Vu, "A traffic and power aware algorithm for virtual machine placement in cloud datacenter", International Journal of Grid & Distributed Computing,Vol. 7 Issue 1, 2014.

[2] Anton Beloglazov, Jemal H. Abawajy and Rajkumar Buyya and "Energy-efficient management of data center resources for cloud computing: A vision, architectural elements, and open challenges" ResearchGate, arXiv:1006.0308, 2010.

[3] Lu´ıs Henrique M. K. Costa and Daniel S. Dias "Online Traffic-Aware Virtual Machine Placement in Data Center Network" University of Brazil,2013.

[4] P. J. Mucha, J.-P. Onnela, and M. A. Porter, "Communities in network,",2009.

[5] M. E. J. Newman, "Fast algorithm for detecting community structure in networks," Phys. Rev. E 69, 066133 (2004).

[6] Jun Yan and Jing Tai Piao, "A Network-aware Virtual Machine Placement and Migration Approach in Cloud Computing", 9th International Conference on Grid and Cloud Computing, 2010.

[7] T. S. E. NG and W. Guohui, "The Impact of Virtualization on Network Performance of Amazon EC2 Data Center", INFOCOM, IEEE Proceedings, 2010.

[8] Sema Oktug and Tevfik Yapicioglu, "A Traffic Aware Virtual Machine Placement for Cloud based Datacenters", International Conference on Utility and Cloud Computing, IEEE/ACM, 2013.

[9] P. Patel, A. Greenberg, D. A. Maltz and J. Hamilton, "The cost of a cloud: Research problems in data center networks", ACM SIGCOMM Computer Communication Review, vol. 39, pp. 68-73, 2009.

[10] Guido Marchetto, Francesco Lucrezia, Vinicio Vercelloney and Fulvio Risso "Introducing Network-Aware Scheduling Capabilities in OpenStack", First IEEE Conference on Network Softwarization, London, March 2015.

[11] "Network Aware Schedule" in OpenStack Compute (nova) https://blueprints.launchpad.net/nova/+spec/network-aware-scheduler/

[12] Shangruff Raina and Ashima Agarwal, "Live Migration of Virtual Machines in Cloud", International Journal of Scientific and Research Publications, Volume 2, Issue 6, June 2012.

[13] Kevin Jackson, Cody Bunch and Egle Sigler "OpenStack Cloud Computing Cookbook" , Third Edition, August2015.

[14] Michat Dulko, Michat Jastrzebski,Pawet Koniszewski,"Dive Into VM Live Migration",OpenStack Liberty Summit, Vancouver, 2005.

[15] OpenStack Cloud Software, "Installation Guide", http://docs.openstack.org/juno/install-guide/install/apt/content/ch_basic_environment.html.

Ubuntu Documentation , "Setting Up NFS server", https://help.ubuntu.com/community/SettingUpNFSHowTo