# A MEMORY UTILIZATION AND ENERGY SAVING MODEL FOR HPC APPLICATIONS

[1]Santosh Devi, [2]Radhika, [3]Parminder Singh
[1,2]Student M.Tech (CSE), [3]Assistant Professor
Lovely Professional University, Phagwara, Punjab, India

***ABSTRACT***
*Cloud computing is an emerging trend in today's era. So with the advancement in technology there has been advancement in the aspects of cloud and its utilization. In this research we are going to discuss about live migration in clouds and how the single memory channel has been a bottleneck to the performance of CPU, communication and memory subsystem. A multi- memory channel model is presented in which all these issues which are affecting the overall performance of our system are handled. Memory channel partitioning approach is also discussed to get the better results and to solve the problem of degrading performance of the system due to single memory channel. Earliest deadline first algorithm has been applied to find the solution to the starving requests so that they don't have to wait in the queue for processing. The model proposed will therefore aim at achieving the high performance and to reduce the overhead of migration on memory subsystem and various other aspects of the system. There has been continuous increase in the cores of CPU which is increasing the performance but on the other hand it is an overhead, and this overhead is to be tackled carefully. The performance of virtual machines gives better result with the more number of CPU. The rapid increase in the cloud and its infrastructure has lead to increase in the various aspects of the cloud computing. This has resulted in significant increase in the virtualization technology and lays more emphasis on virtual machines, live migration and so on. So these issues will be dealt with high attention to get solutions.*

***KEYWORDS:*** *Single memory channel, Multi-memory channel, virtual machines, virtual queue, banks.*

## I. INTRODUCTION

Cloud computing is an advancement in distributed system, where large number of computers are connected through a real-time communication network such as internet, intranet and extranet. Generally it is referred to network-based services, which are provided by real server hardware, and are served by virtual hardware, simulated by software running on one or more real machines. A virtual machine is a software implementation of computer in which the programs are executed just like a physical machine. Virtual machines can be classified into two groups based on their use and degree of correspondence to any real machine.

1. A system virtual machine provides a complete system platform which supports the execution of a complete operating system. [6]
2. A process virtual machine is designed to run a single program, which means that a single program is supported by it. [6]

Currently cloud computing hosts various heterogeneous applications with different performance requirements including high performance computing, web applications. How-ever there are many challenges in providing reliable and guaranteed performance service in such consolidation environment. Researchers are always trying to find new solutions of these challenges and enhance virtual machine scheduling algorithm, resource allocation and migration strategies. In cloud computing problem of some dirty pages also occur, which are to be cleaned so that the pages of the application can be prevented from over writing. A dirty page is a page that has been modified in main memory (physical memory) but yet not written in the disk. [1] The concept of dirty pages can be made

clearer by knowing about virtual RAM. Virtual memory is software representation of RAM. New operating systems do not allow the direct access to RAM instead they create virtual RAM. Virtual memory is memory taken from the HDD and converted to RAM. [6]

The topic of the research is how to migrate data lively in the VMs so that the rate of data redundancy does not increase and the data is easily accessible by the user. Currently clouds use virtualization technologies to provide an isolated execution environment. [2] As this type of environment help the faster and quicker movement of the data. Use of detailed simulation is necessary as if there will be less details in the model, the result will be incorrect. Live migration can be defined as a process moving data from one VM to another. Live migration is a solution to hosts, load balancing and maintenance.

Generally live migration consists of two phases named as pre-copying and stop-and-copy. [1]

a. Pre-Copying

In pre-copying phase the physical memory of the VM is copied to the target via the network while the VM which is migrating is active. [1]

b. Stop-and-copying

In this phase the migration process is slowed down to match the copying of next memory pages.[1]

Mostly it is seen that the migration algorithms considers only the CPU utilization to trigger live migration but they do not consider the results of migration overhead on other server resources such as memory bus and network. [1] The architecture of cloud computing involves multiple cloud components interacting with each other about the various data they are holding on too, and helping the user to get the required data on a faster rate. Cloud computing is sharing of resources on a layer scale which is cost effective and location independent. Resources on the cloud can be deployed by the vendor, and used by the client. It also shares necessary software and on demand tools for various IT industries. Amazon is the first company to look into the growing importance of cloud computing.[1] There are many benefits of the cloud computing, the most important is that the customer do not need to buy the resources from a third party vendor, but they can use the resources and pay for its service. It saves customer's time and money.

Here, the area of research is the single memory channel model and its drawbacks and how this model can be replaced by multi-memory channel to improve the better utilization of the system and in the energy consumption. Three situations aroused which are stated below.

a. Overhead of live migration on CPU, Memory Subsystem and Network.
b. Communication overhead on the CPU.
c. Job execution time.

## II. LITERATURE REVIEW

In the literature review many aspects of live migration are and all other issues which are to be handled during the study are seen. There are many ways that shows how the performance of the system can be upgraded and the cost can be minimized to increase the speed and to save both energy and time. The problems which are faced due to the single memory channel are discussed. Though before going deep into the architecture, the concept of live migration and what the single memory channel is explained below.

Live migration is the one of the fascinating feature of virtualization technologies. It is used as a solution for hosts load balancing and maintenance. Live migration consists of two phases: pre-copying and stop-and-copying. In this paper rather than taking the CPU utilization only they have considered the other overheads such as memory bus, network.

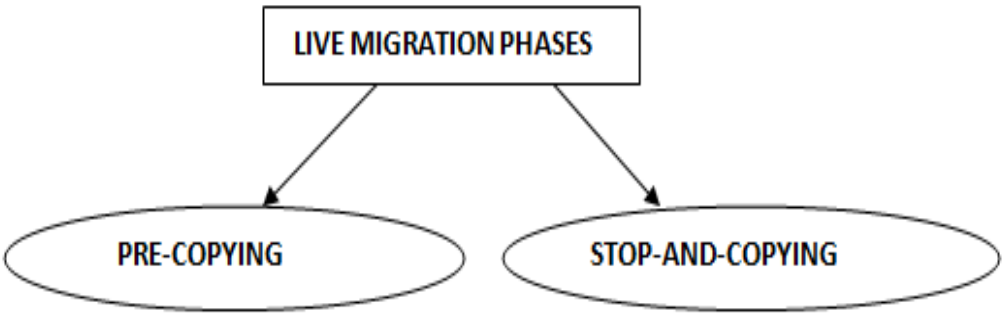


**Figure1:** Phases of Live Migration

There are different communication techniques of VMs that are simulated including shared memory for multi-threaded applications and network for multi-processes applications. Here the influence of memory bus is shown on multithreaded applications and multi-processes applications. Next thing which they have given is the NBP benchmark analysis. The simulation of the demands behavior and communication patterns of each benchmark. After going through the live migration concept it is advisable to know about the single memory channel model and how it lacks in certain aspects.

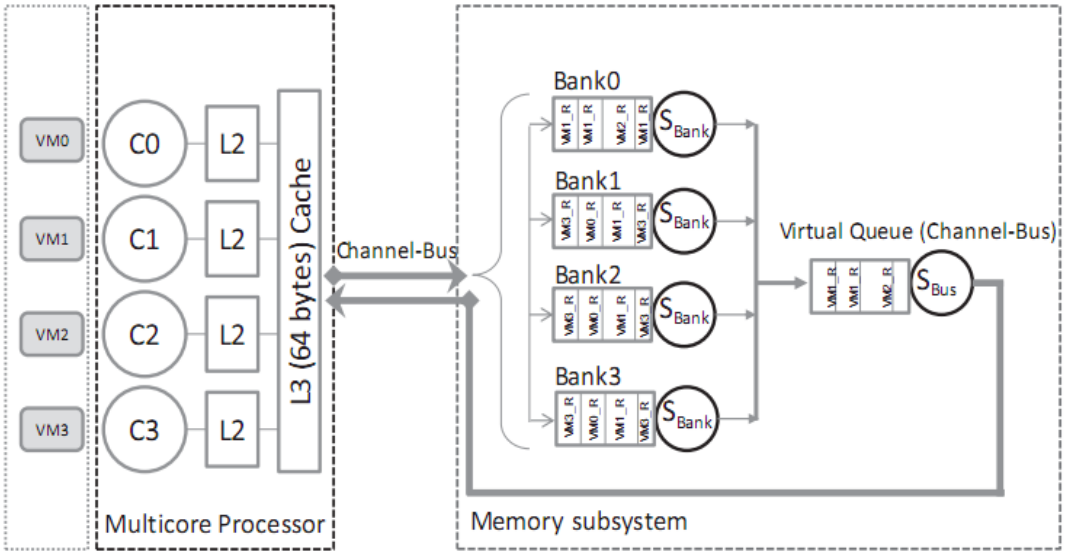


**Figure2:** A Queue Model Of a single Memory Channel

In this diagram they presented a memory-bus and its implementation into cloudsim. The model implemented by them is similar to the one proposed by Deng-et-al. the thing they did is that they attached virtual queue to the model the waiting time that a memory spends when the memory-bus is locked by another request. [1] The processor is a single stage queue of process sharing; meanwhile memory subsystem is a two stage tandem queue of a First Come First Serve (FCFS). In the memory subsystem the output of the banks queues is the input of the bus queue. Thus when a request is issued by a VM to the bank, the request might wait for some time before being served. [1]

The request is moved from the bank to the bus, after already served at the bank. DRAM allows this to occur only when the bus is free. So the requests are held at the bank until the bus become free. To simulate the waiting time of a request from memory to the processor they have that that the memory-bus behaves as a single server queue.

The second thing in research is to know how to partition the single memory channel. So here it is discussed about what is channel partitioning. So, first of all it is said that the performance benefits of mapping the pages of applications with largely different memory intensities to separate channels.[2]

Conventional page mapping - in conventional page mapping the requests have to wait until the earlier requests are processed. Though the requests coming from the second source are creating a disturbance to the other but still it continues to run the processes which have arrived earlier.

Channel Partitioning - in this approach the requests that are coming do not wait and are processed at the time they arrive. So there latency of all the requests are eliminated and thus the there is increase in the processing.

MCP consists of three steps which are performed at different intervals:

Profiling of an application - under this the statistics related to MPKI and RBH are collected and it is seen that due to which application how much harm is caused to the other applications.

Assignment to preferred channel - then after the profiling the applications are assigned preferred channels, the goal of this is to (a) separate low memory- intensity applications from that of a high memory-intensity applications, (b) among the high memory-intensity application, data of low row-buffer locality applications from that of high row-buffer locality applications.

MPKI (t) = Average of the last level cache misses per kilo instructions of all applications and multiplying it by a scaling factor.

MEMORY INTENSITY

MPKI<MPKI (t) = low intensity

MPKI>MPKI (t)= high intensity

HIGH INTENSITY

RBH>RBH (t) = high row buffer locality.

RBH<RBH (t) = low row buffer locality. [2]

Memory channels are assigned to application groups and not to individual applications. Channel will be allocated proportional to the number of applications in that group. By this it can be said that if the system doesn't achieve the high intensity then the performance is degraded. Next is about the scheduling algorithm which is to be used for our model. The algorithm chosen is the earliest deadline first. This algorithm is scheduled in the multiprocessors. It is a real time operating system algorithm, whish picks the dynamic task priorities, and the job with the nearest absolute deadline gets highest priority. And most of all it is an online algorithm. It is a dynamic algorithm used in real time OS to place processes in a priority queue. Whenever a task finishes the queue will search a task which is closest to its deadlines.

Schedulability test for EDF is

$U=\sum i=1 \text{ to } n, Ci/Ti \leq 1$

Where, Ci is the worst case computation time of the n processes.[3]

Ti is their arrival periods.

It guarantees that the total CPU utilization is not more than 100%. The new algorithm proposed to restrict the task migrations. There is no total utilization constraint. The new algorithm proposed restricts the task migrations. There no total utilization constraint. Scheduling algorithms can be categorized into (1) static (2) dynamic but fixed (3) fully dynamic.

DEADLINE: It is a time by which execution of transactions should be completed, after the transaction is released.[3]

Flow of EDF is as follows:

Calculate deadline of each transaction present in queue.

Deadline= arrival time+ slack factor*average execution time.

Schedule the deadline.

Serve transaction with earliest deadline.

This paper tells the concept, which says that memory scheduling algorithms should be designed to handle the memory requests from different threads. This can provide better system throughput and the fairness in the working of the system. A new algorithm known as "priority based fair scheduling" is discussed in which it is said that in it classifies threads memory access behavior by dynamically updated priorities. Here it says that the threads those are latency sensitive they have top priority for giving the throughput to the system. [4]

From the study of this paper it can be known that the dynamic web content is very small(less of CPU cycles). And for the static web loads there is networking overhead of (upto 25% of CPU cycles). [5]

## III. RESEARCH METHODOLOGY

To get rid of the problems and to achieve the performance of our system we proposed an architecture which we named as MULTI_MEMORY CHANNEL MODEL. Our model tries to cover all the

limitation that we found in the existing single memory channel and to partition the channel in such a way so that high memory intensity applications get the channels. Our research follows the approach in which first of all we will convert the single channel into multi-channel by using the portioning approach. The next step is to apply any scheduling algorithm on the virtual queue so that the requests are not halted and they can be processed at the time they are arriving. We also apply an algorithm before the virtual queues so that if a job is coming from any VM and at that particular time the virtual queue of that VM is not free then the job of that VM will be shifted to the virtual queue which is free at that particular time. The jobs will be provided with a token-id so that after processing in another queue it gets back to the same VM to which it belongs. So that is no conflict of the processes and all the jobs reach to their destination without any delay.

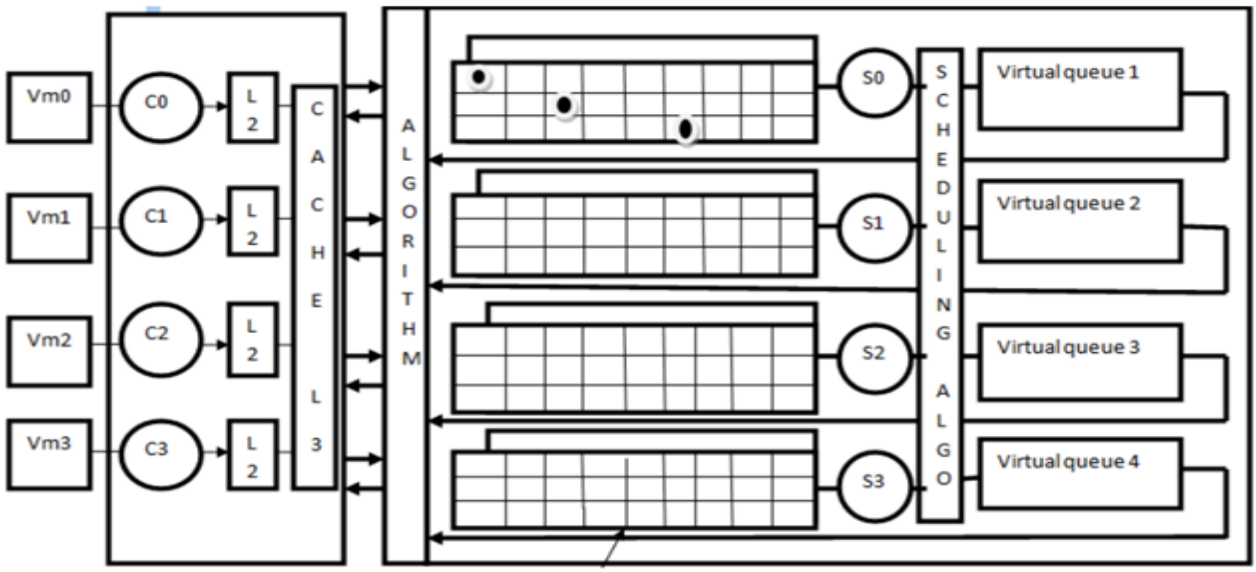


**Figure3:** A Queue Model Of Multi-Memory Channel

## IV. FUTURE WORK

As it is discussed in introduction that single memory channel has various limitations so far. This model will be able to increase the speed of the system and to use the energy in a more efficient way. Moreover there are certain aspects which were ignored in the previous model and the aim of study is to study those areas and bring out the better solution to them. Here in this model channel partitioning approach known as Multi Channel Partitioning followed by an algorithm is used. The model proposed will have more efficiency and it can be further used in upgrading the system. If this model is applied over the cloud then the large number of problems such as the starvation of requests can be sorted. This model if used in the processors then the speed of the processors can be increased and so as the system throughput. The response of the system towards the requests will be quicker and thus the performance of the system automatically increases.

Memory channel partitioning allows the flow of many requests towards the memory and at the same time many requests can be handled. Earlier the system was too slow as the channel for flow of requests was one and thus only one request moved to the memory one by one. Job scheduling is one aspect which can be improved to get the faster result and better results. Waiting time of the requests will be reduced which will help in the faster execution of the processes. After cleaning the dirty pages from the memory, the memory can be utilized in more efficient way and also the demand for clean pages from the memory is reduced. It simply increases the utilization of memory and also reduces the overhead on the memory due to those dirty pages. Dirty pages also effects the processor and its speed, so once these dirty pages are treated then the performance of the processor also increases and thus due to this the overall output of the system can be increased.

## V. CONCLUSION

Though from going through the single memory channel it can be said that there are many problems in it and due to these problems the performance of the system can be degraded. Thus by applying the approach provided in the paper can improve the performance and throughput of the system and following objectives can be achieved which are enlisted as follows:

a. Reduce the overhead of live migration on CPU, Memory Subsystem and Network.
b. Reduce communication overhead on the CPU.
c. Reduce Job execution time.
d. To reduce the dirty pages in the memory those are generated in memory.
e. To partition the channels using some algorithm.

## REFERENCES

[1]. Ibrahim Takouna, Wesam Dawoud and Christoph Meinel (2012) "Analysis and Simulation of HPC Applications In Virtualized Data Centers" 2012 IEEE International Conference on Green Computing and Communications, Conference on Internet of Things, and Conference on cyber, Physical and Social Computing.

[2]. Sai Prashanth Muralidhara, Lavanya Subramajan, Onur Mutlu, Mehmut Kandemir, Thomas Moscibroda. "Reducing Memory Interference in Multi-core Systems via Application- Aware Memory Channel Partitioning".

[3]. James H.Anderson, Vasile Bud and UmaMaheswari C.Devi "An EDF-based Restricted-Migration Scheduling Algorithm for Multiprocessor Soft Real-Time System" The University of North Carolina at Chapel Hill, NC 27599 USA.

[4]. Chongmin Li, Dongsheng Wang, Haixia Wang, and Yibo Xue "Priority Based fair Scheduling: A Memory Scheduler Design for Chip-multiprocessor system". Tsinghua national laboratory for information Science and technology, Beijing 100084, China.

[5]. Jaidev P. Patwardhan, Alvin R. Lebeck, Daniel J.Sorin. "Communication Breakdown: Analyzing CPU usage in Commercial Web Workloads".

[6]. http://www.slideshare.net/umairamjadawan/archi-15659322