**A Survey in Green Cloud Computing Technologies**

**Abstract**

The internet is expanding its viewpoint into each conceivable part of the cutting edge economy. Since quite a while ago unshackled from our web programs today, internet is characterizing our way of life, regardless of whether it's sitting in front of the TV or driving an independent auto. The enchantment of the internet appears to be relatively unbounded. In any case, with each new spell there comes an ever increasing number of data, and interest for computational power. Cloud computing which is an on-request conveyance of computing power, database storage, applications and other IT assets through a cloud benefit stage by means of the internet has violently expanded our computerized lives. While there have been critical advantages as far as accessibility, fluctuation, time and quality in administrations, the unbounded development of our computerized way of life requires monstrous measures of power, especially for the data centres that fill in as the mind of the advanced economy. Data organizations foresee a decrease in the quantity of data centres, as more business close their little data centres and move towards cloud computing. All things considered, the move by shoppers that will rely upon cloud, will build the general energy utilization significantly exceeding any energy productivity increases acknowledged by the move and record for over 70% of data centre development in 2018. The advances in run can possibly diminish the energy utilization of the computing types of gear included; make utilization of efficient power energy, decrease of carbon impression and e-squander. These procedures are supporters of green cloud computing, which is a region focussed on planning and advancing energy-proficient activities to contain inordinate energy utilization in data centres.

In this paper, we survey the state-of-the-art techniques used for lowering the energy consumption in data centres and their components. An In-depth study of the existing literature on data centre power reduction techniques is done. We organize the study of various components of the data centre in a hierarchical manner with two main branches as hardware-centric and software-centric approaches. The third branch focuses on renewable energy and moral filter of both Cloud Service Providers and constitutional bodies. This systematic approach allows us to identify multiple points where considerable energy reductions can be achieved in a data centre.

Keywords- Server, Processor, CPU utilization, Core, GPU, Cache, SRAM, DRAM, HDD, SSD, Cluster of Server

**1. Introduction**

Cloud computing is currently ubiquitous from the fields of spilling music, recordings, document sharing, web and messages, to the incipient of "Internet of Things". The offline world is quickly changing to the online world with a 20% expansion in data for each year [2]. The expansion in volume of BigData is expected to develop hugely, with the rise of modest advanced cells. Internet movement from cell phones expanded by 69% of every 2014 alone with a quick increment in video gushing which is relied upon to surpass what is conveyed over wired associations by 2018. By a wide margin and away, the greatest driver for purchaser internet data is online video. YouTube, Netflix, Hulu and other video gushing administrations have turned into a staple sustenance making up to over 60% of buyer internet activity, which is expected to increase to 76% by 2018 [3]. Subsequently, Information and Communication Technology is detonating as far as gadgets, assembling, systems and datacenters. Electricity is by a wide margin the greatest long haul energy cost, alongside brand and client driven worries about the ecological effect of these online administrations. It has been watched that electricity utilization in data centers has expanded from 15% to 21% and networking from 20% to 29% from 2012 to 2018 [4]. Nonetheless, there has been a reduction in energy utilization of gadgets and assembling part of ICTs from 18% to 16% and 47% to 34% individually. In any case, late investigations have assessed that the aggregate energy utilization of our gadgets, data centres, and systems has bounced from 7.4% of worldwide electricity utilization in 2012 to 12% out of 2017 [5].

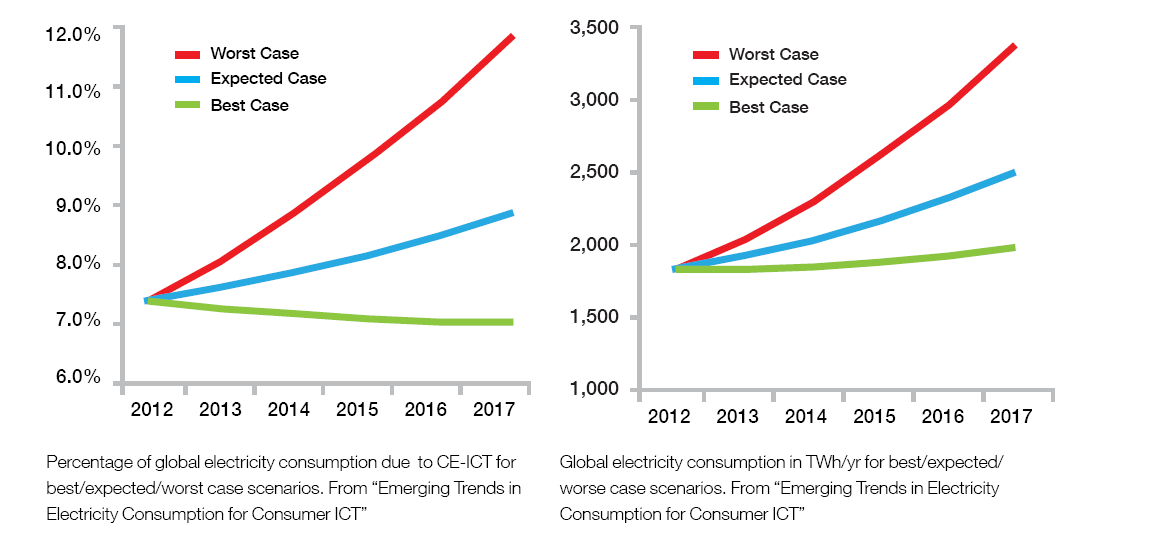


Fig.1. Electricity demand growth of ICT sector [5].

The transition of ICT sector towards energy efficient policies can promote green and sustainable development along with efficient utilization of assets and decrease their liability towards the environment.

Efficient planning of Data centres effectively utilizes the two broad sectors of energy consumption within them: IT equipments (e.g. Servers, networks, storage, etc.) and infrastructure facilities (e.g. cooling and power conditioning systems). A breakdown of energy consumption by different components of a data centre show that 50% power consumption happens in cooling equipments followed by 26% in server and storage, 10% by network hardware and other infrastructure components [6].

Green computing, an active area of research helps to analyse the relation between system utilization and power consumption such that the latter can be reduced while adhering to the Service Level Agreement. However, lowering the power consumption behaviour of the Data centre as a whole or at the individual component level is not straightforward. Data centre energy consumption depends on multiple factors such as hardware specifications, workload, networking infrastructure, types of applications, storage infrastructure and many more. Efficient energy consumption can be achieved by use of state-of-the-art-techniques like virtualisation, resource management, auto scaling of infrastructure, energy aware scheduling, storage space management, live migration of virtual machines, reduction of carbon footprint and many more. These techniques can be applied at appropriate component level and together made to realise a complete system i.e. Data centre as green, which relates to two things: right off the bat, the environment i.e. the abatement in energy consumption and the paper money i.e. cost productivity the cloud computing activity.

One of the key features of green cloud computing technique is its wide range of application and interdependency. A single technique can be applied over various components and used to achieve the results and it may also use various other techniques to achieve the desired goal. For example resource management involves management of servers, switching them off if it’s underutilized. This requires migration of virtual machines from the underutilized server to moderately utilized servers. Migration which involves use of network where also one can optimize power consumption. Thus, green cloud computing techniques are related to each other.

**2. Data Centre Energy Consumption: A System’s Standpoint**

Models are representation of reality; they predict how things will pan out in the near future in the real world. They allow us to design machines and processes that will project what we wish to happen in the reality, because to some level they simulate reality. One can refer to models as deep knowledge and taxonomical approach to achieve objectives. When doing modelling, we have to make a trade-off between the amount of detail that the computations will work with and the accuracy of the predictions, and usually we choose the level of detail that is sufficient for our requirements.

This section models the entire data centre to the required level of details in the terms of energy consumption. An overall view of the Data centre model used in this survey is shown in Fig. 2 below. Like every other computer system a Data centre consist of two main layers: software layer and hardware layer. Both of these layers can be optimized in terms of energy consumption. The software layer can be further categorized into two subcategories Application and OS or virtualization layer. The first half of the survey focuses on the hardware aspects of the Data centre and in the latter part we explore the power consumption modelling for software.

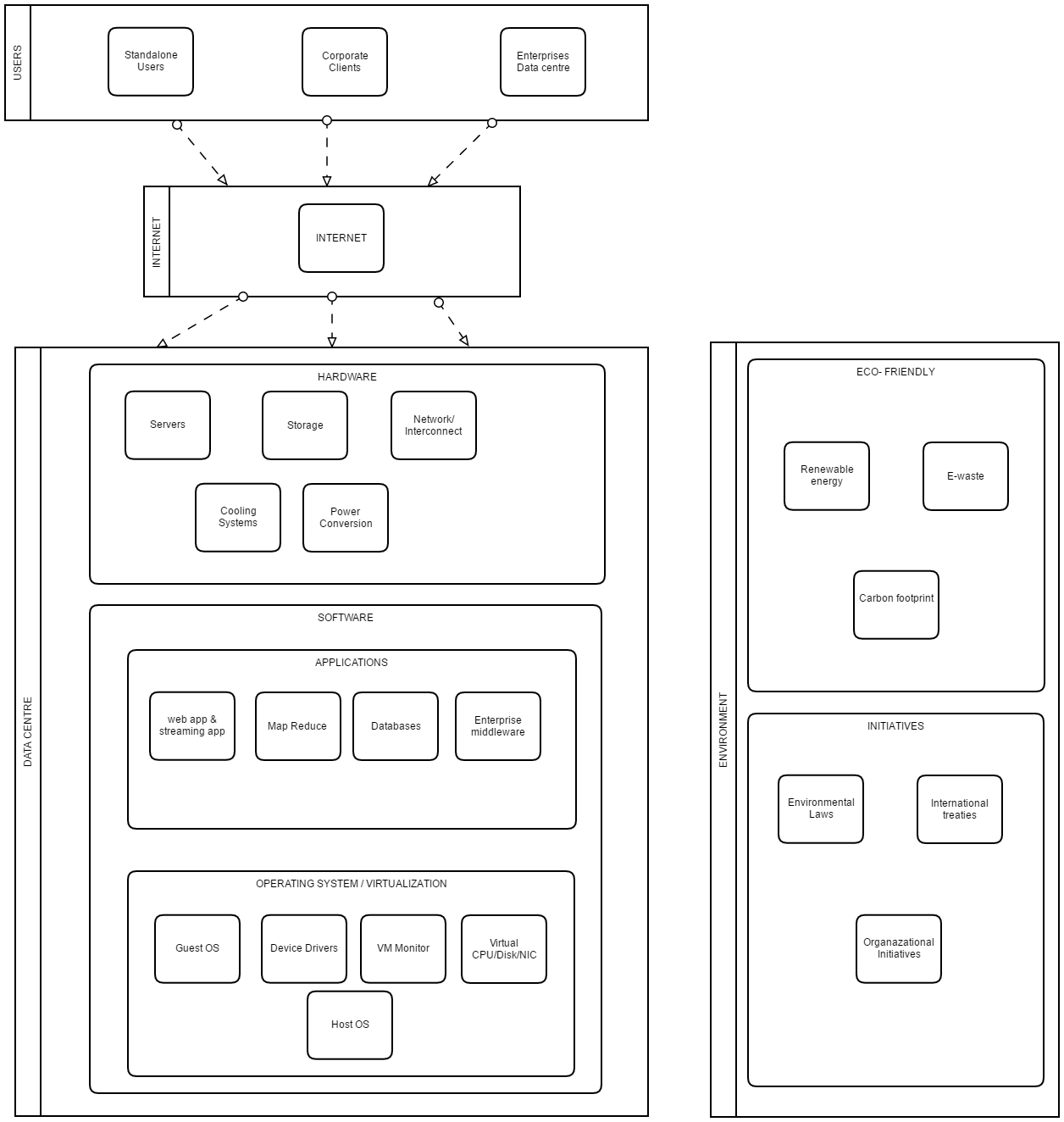


Fig.2. A holistic view of the energy consumption Modelling of the Data centres.

Further, we move on to providing a taxonomical approach as in Fig. 3 below, for implementing the green Data centre. Various techniques at the component level can be applied to achieve the goals of green cloud computing. Beside the software and hardware aspects we also take into account some external agents to the Data centre like renewable energy and organizations or government imposed rules and regulations to also achieve the objectives of green cloud computing.

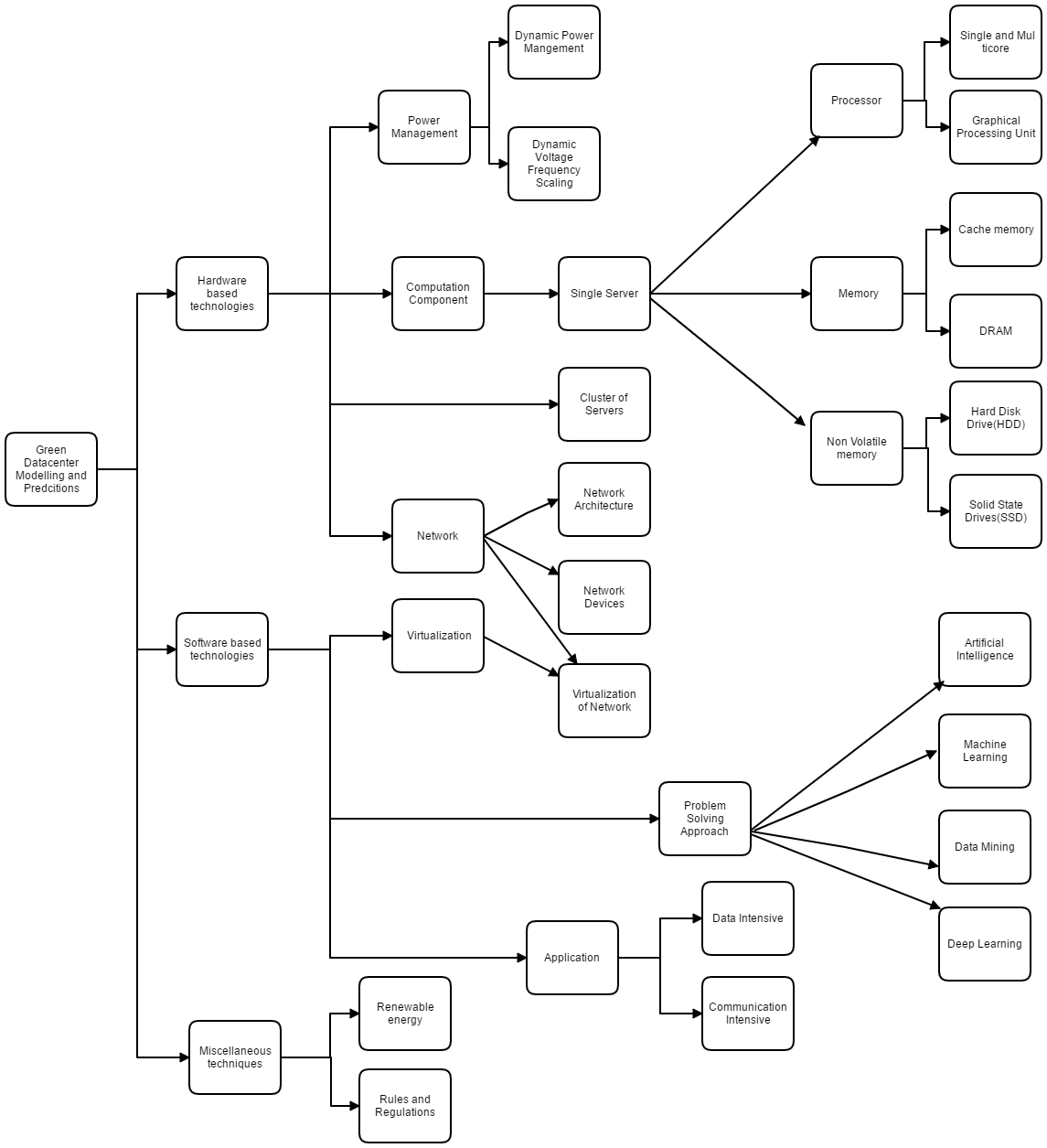


Fig.3. A taxonomy based overview of the Data centre energy consumption modelling.

**3. Data centre Modelling at Single Server Level**

A server is hardware component that is responsible for managing the resources, if any computer device shares resources with the clients as per the request it can be called as server. Servers can be dedicated to a specific task like application server, mail server, file server, print server and etc. or they can also be shared server which can perform multiple responsibilities like a web based server can respond to DNS, multiple websites, FTP based requests. Server in order to respond to the requests often runs a software program upon itself which acts like helper and performs computation. Just like any normal computer system, servers also have memory and storage to store the instructions of the software program and processor to execute the instructions.

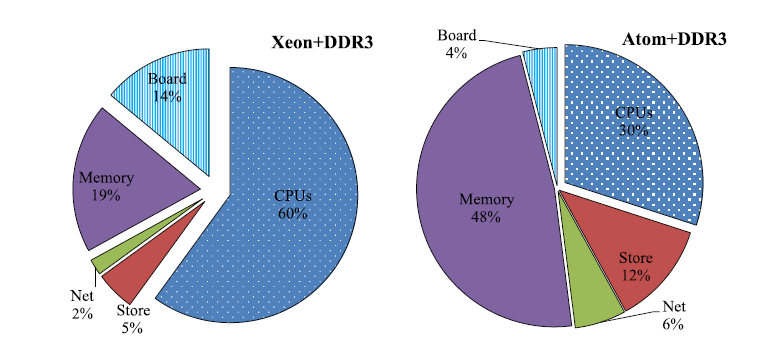
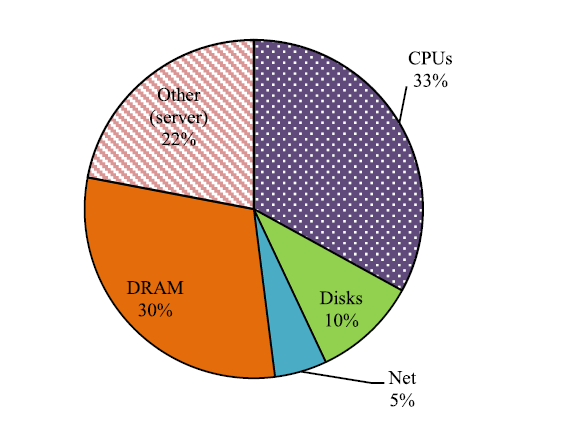


Fig.4. An approximate distribution of peak power in server at Goggle Data centre [7], Atom processor server and Xeon based server [8] (left to right).

Power consumption in server is dynamic it depends upon the operations of circuits as in buses, access to disc drives (I/O) and etc. It depends on the workload and its type being executed on the server, how the workload utilizes the servers CPU, memory, I/O is of utmost importance. Fig. 4 depicts that the power consumption of various components in the server are not fixed. For instance as shown the in the server deployed at Goggle Data centre and Xeon based server the CPU is the largest consumer of power followed by memory. However, in the mobile processor (Atom processor) memory is the largest consumer of power followed by CPU. Disk is another largest power consumption point. So to optimize the power consumption at the level of a single server one should implement policies at its component level which includes CPU, memory, and Storage. One can also optimize the network, I/O, peripheral devices and the rest of the glued circuitry in the server.

In the subsections we discuss the three main components processor, Memory and Non-volatile memory or storage and how policies have been implemented at these component levels to achieve objectives of green cloud computing.

**3.1. Single Server- Processor**

The processor which is a main point of power consumption in the servers consists of cores. Cores are the basic computation unit that execute the instruction cycle, maintain program states, registers, generate control signal and even have their own cache memory consisting of frequently used memory chunk. A processor can have multiple cores to perform the tasks and achieve parallelization; these tasks are usually software processes or threads that the OS schedules. Besides this continuous development in the field of I/O devices introduced special purpose processor for processing of different types of data known as accelerators. Graphical Processing Unit is one such accelerator developed for processing of graphical tasks but now is routinely used by Cloud and server infrastructures for servicing compute tasks such as signal processing data mining, biomedical situations, search and gaming. A number of large scale cloud Service Providers (CSP) like Amazon, Microsoft, and Google offer GPU services.

**3.1.1. Single and Multi-Core Processors**

A multi-core processor is a single computing component with two or more cores which read and execute the program instructions. This enables to run multiple instructions on separate cores at the same time, increasing overall speed of the program amenable to parallel computing. This makes them suitable to run CPU intensive tasks. These OS can have multiple processes and threads i.e. tasks to run but it can run only X such tasks at a time where, X is product of number of cores and the number of hardware threads per core. This leads to basic concerns of multi-core systems like resource management and performance.

Authors in [9, johari 2014] focus on the performance aspect of the multi-core systems. They perform task migration over the cores while handling the issue of load balancing. To perform task migration over the cores authors propose four algorithms sequential search, random search, random search with spatial locality and Average method. The Average method is a load balancing algorithm implemented at the core level that takes in, the global and local view of the cores. It also makes use of threshold to avoid over utilization of cores. However, the proposed load balancing algorithm cannot handle dynamic workload, threshold formulae isn’t sound and task migration over cores has an overhead, consisting not only of the time needed in transfer of state, but also related to cache locality as the data has to be reloaded into the cache of the new core.

Author in [10, mann2016] avoids the problem of task migration by pinning the virtual machine (VM) CPU cores to a specific physical machine (PM) CPU cores. The author formulates a cost problem, whose solution is found using constraint programming. In the first stage the VM-to-PM mappings is done with aim of minimizing the cost function, the number of overloaded CPU’s is calculated and the number of active servers is decreased by accommodating the VM from the lightly used PM’s and overloaded PM’s using the Modified Best First Decreasing (MBFD) Heuristic. In the second stage, optimal mapping of cores is performed using the exhaustive backtrack search and first fail heuristic. The author ensures performance by minimizing the number of migrations, and the number of overloaded physical machine CPU’s and resource utilization by minimizing the number of active servers. However, the size of search space can be further narrowed down using advanced heuristic. Also, author doesn’t consider the memory resources of the processor while scheduling the VM’s.

In order to efficiently use the memory resources, today’s OS have notion of Non-Uniform Memory Access (NUMA) hardware, which optimizes the running applications itself. In this the software threads of an application are scheduled on the cores where memory contents resides which enables fast memory access. A process that is being swapped-in must preferably run on a core it previously ran on as it will benefit from the previous cache contents. Migration of tasks must be infrequent and task and its associated memory contents must be kept on the same core.

With advent of Kernel Virtual Machine (KVM), the traditional Linux Kernel is converted to a hypervisor which schedules every VM as running process but not takes into account the memory requirements. On NUMA hardware with long running tasks such as VM this can cause performance degradation. Author in [11, 4 of saxena2012] propose an approach for dynamic NUMA binding. In this kernel assigns a ‘home node’ to each process. The kernel ensures that the VM process is run on the CPU’s of this node and also cater it for his memory allocations. The scheduler contains each process within home node until load balancing dictates otherwise. Even if process is moved to another node, the kernel service its allocations from home node but if a memory or CPU imbalance occurs the tasks are moved around to new home node. This provides a perfect trade-off between specifying ideal CPUs to run to kernel deciding the ideal CPU to run. However, this is a static approach, once the bindings are set they cannot adapt to change in threads memory association. Author in [11, 6 of saxena2012] propose AutoNUMA approach to resource management. It says that the kernel must keep watching each process, monitor its memory access and move the frequently accessed pages around while the application is unaware of the fact that memory has migrated. However, since it relies on lot of metadata per process and page it can be CPU and memory intensive method which can cause scalability issues on very-large memory systems.

**3.1.2 Graphical Processing Unit**

A graphical processing unit is a computer chip that can perform rapid mathematical calculations. Primarily developed for rendering images, its parallel processing architecture has made it suitable for High Performance Computing (HPC) in cloud and server infrastructures.

Companies like Intel, Nvidia, and ATI are starting to push the boundaries with more general purpose GPU components, making it easier for software developers to utilize the extra processing power on video cards to perform non-graphic operations. Advent of general-purpose programming language and Application programming Interfaces (API) like RapidMind, Brook made it easier for software developers to target GPUs. A major breakthrough occurred with Nvidia’s CUDA, designed to work with programming languages like C, C++, and Fortran, the CUDA platform provides direct access to the GPUs virtual instructions set and parallel computing elements, making virtualization a reality at the GPU level. Amazon has recently announced a new P2 instance type, developed in collaboration with Nvidia which incorporates up to 8 NVIDIA Tesla K80 Accelerators, each running a pair of Nvidia GK210 GPUs. They also include ECC memory protection and double precision floating point operations which makes these instances a great fit for large-scale machine learning, deep learning, and computational finance workloads all being delivered over the cloud platform of Amazon Web Services (AWS).

However, delivering HPC through cloud is still very challenging even after the virtualisation is achieved at the hardware level for the GPUs. Latency becomes major concern with HPC as application kernels may not always fully utilize a GPU. This calls for new scheduling algorithms to optimize co-execution of multiple applications on the GPUs. Authors in [12, ukidave 2016] present Mystic, a framework that schedule workloads on GPU while minimizing the interference between the applications. Before the application begins its execution, a profile is created based on 6 compute resources for which applications contend among themselves. These resources are stream multi-processors, memory resources, and the interconnect network. Out of these 6, 2 resources are measured and the rest are predicted by use of Single-Value Decomposition based collaborative filtering technique. Then if there is an ideal GPU, the application is scheduled upon it else its similarity is calculated with every other application and dispatched to the GPU with lowest similarity score. Mystic improves the throughput by 27.5% and GPU utilization by 16.3%, while maintaining the QoS to 90%. However, the characteristics of the applications are calculated on the basis of previously executed application and before the execution of the current application which may lead to false predictions.

Authors in [13, Xu 2017] present GScheduler which takes into account the function calls made by the application during execution along with resource usage (7 tuple) to detect and reduce the interference between the co-executing applications. The similarity score for key function (function that impact the GPU resource allocation and usage) call graph, and GPU resource usage vector is obtained. On the basis of which interference between two applications is measured and scheduler assigns the application to the GPU with lowest interference. GScheduler out performs Round Robin and Least Loaded schedulers. However, creating a profile for each application can also cause resource exhaustion. Also, authors didn’t compare the framework with other Machine learning frameworks like Mystic, so it can’t be said how efficiently GScheduler performs in respect to them.

**3.2 Single Server- Memory**

Memory in servers or any other computing system is a physical device capable of storing information in form of bits. The cost of memory decreases with increase in size but at the expense of speed. Since processor and memory follow a master-slave architecture, memory should be able to respond to the data request at the speed processor executes the instructions to ensure maximum CPU utilization. Thus, memory was classified as non-volatile and volatile memory. Volatile memory contents are lost as soon as the power goes off. This section focuses on volatile memory often called simply as memory; non- volatile memory often called as Storage is discussed later. Volatile memory can be further classified into two- Static Random Access Memory (SRAM) or Cache and Dynamic Random Access Memory (DRAM).

**3.2.1 Static Random Access Memory (SRAM) or Cache**

SRAM is a kind of semiconductor memory that uses flip-flops to store each bit, these flip flops are implemented using transistor which loose current as soon as the power goes off and thus the data. The power consumption of SRAM chips depend upon the frequency with which they are accessed. It can use a much power as DRAM when accessed at high frequencies also; SRAM suffers with the problem of leakage power. To overcome these authors in [14, fujita2017] suggest the use of embedded spin torque transfer MRAM (e-STT-MRAM) which has highest write access speed and highest endurance, has been developed by decreasing size of Magnetic Tunnel Junction (MTJ) materials. For cloud servers, since the bottleneck of computing performance can be main memory access, short retention MTJs should be used in e-STT-MRAM based Last Level Cache (LLC) to enhance memory bandwidth between CPU and main memory. Simulation shows that it can increase the operation speed and reduce power consumption up to 90%.

Another way to increase the performance of applications running upon the virtual machines is by dynamically optimizing the virtual machines memory sizes. Authors in [15, sakamoto2015] optimize the memory ballooning technique. Memory ballooning is a memory reclamation technique used by the hypervisor to reclaim unused memory form the virtual machines. To reclaim the memory, decrease the memory size of virtual machines with cache hit ratio greater than threshold by α% and all others by β%. Authors propose to reallocate this memory to virtual machines with cache hit ratio higher lower the threshold. The allocation size is proportional to cache hit ratio. The memory ballooning technique is invoked after every predefined γ seconds that can be increased to 1.5γ seconds if cache hit ratio exceeds threshold more than 3 times. The technique increases the throughput and indicates that the proposed method avoids the creation of low performance virtual machines. However, authors didn’t design the value for α and β to be considering the swapping problem where a VM is inflated to the point that it runs short on memory to run its own process and start using another VMs memory.

Authors in [16, chen2017] reduce the latency by 50% and double the throughput for read operations by minimizing the distortion and possible bias caused by Least Recently Used (LRU) and Least Frequently Used (LFU). They use a greedy approach which is a combination of LRU and LFU, the data cache on reaching a predefined limit calls the greedy algorithm to free a memory portion of size P. According to the algorithm, the cache data is firstly sorted on LRU scheme, then select 2\*cache size\*P size of cache data from the sorted LRU list and sort it on basis of LFU and select P size of cache data from the list, to evict. The simulation shows that the approach reduces the latency and increase throughput by a promising margin. However, when it comes to write operations the improvement fraction is very small. In a similar work, authors in [17, 18ofchen2017] propose a hyperbolic caching for web applications while focussing on shortness of using eviction data structures. They propose to decouple the data item priorities from eviction data structures like priority queues also a random sampling approach to determine the cache data to be evicted is proposed.

Authors in [18, yang 2015] dive into the architectural aspects to handle the dynamic nature of workload in cloud computing by modifying the architecture of existing CPU/FPGA acceleration system. Author uses the principle of locality and maximum processor utilization to create a CPU-Cache-FPGA architecture which has its own data cache to minimize the data fetch latency. It also employs Bus snooping logic to continue reduce the data access latency. Simulations show that it can result in performance improvement up to 2.6 times. Likewise Authors in [19, han2012] propose Cache as a Service (CaaS) model as an additional service to IaaS. In this a user can specify more cache memory as an additional requirement to the IaaS with the minimum computational capacity. The CaaS model consist of two main components: an elastic cache system as the architectural foundation and a service model with pricing scheme as the economic foundation. The CaaS model helps IaaS improve disk I/O performance which leads to reducing the number of physical machines, increases throughput, and in turn results in profit increase.

**3.2.2 Dynamic Random Access Memory (DRAM)**

DRAM is a type of non-volatile, read write memory that stores each bit of data in tiny capacitor; the charged and discharged states of the capacitor represent the two bits 1 and 0 respectively. The capacitor discharges slowly and thus requires refreshing, which periodically restores them to original charge and prevents loss of data. The traditional DRAM has two major problems first, the continuous need to refresh memory in both idle and working states which consumes a lot of energy and betrays the purpose of green cloud computing. Secondly, the DRAM has reached its upper limit in terms of capacity. As a result architectural changes are being proposed to include Non-volatile memory in every possible aspect as possible. Authors in [20, venkastetan2014] proposed a 3-level cache miss model which works on the basis of Miss Ratio Curves (MRC). MRC describes how miss probability varies with cache size and is determined by the interaction pattern between reference pattern and caching policy. Authors take in both system and user perspective while designing the MRC equations. In system perspective level 1 Cache(DRAM) is fairly portioned among VM by normalizing the miss probability and for user perspective, level 1 and level 2 (Non-volatile memory) cache together called as levela cache is sized appropriately to meet the SLA objectives. Since Non-volatile memory (NVM) is slower as compared to DRAM it is kept at level 2 and the latter at level 1, level 3 is generally the disk and magnetic tapes. Authors apply the model to transcendent memory to show that it can be used to achieve fairness at one level and latency bounds at another.

When it comes to NVM, a number of alternatives like Phase Change Memory (PCM), Ferromagnetic RAM (FRAM), and Magnetic RAM (MRAM) have emerged. All these alternatives for NVM are faster with several orders of magnitude as compared to the disk and flash storage but, have poor write performance and endurance. PCM has high density, non-volatility, positive response to increasing temperature, zero standby leakage, and excellent scalability. PCM works by detecting resistances of different states that lie within range of its amorphous and crystalline states, a recent study by IBM stated that one can identify up to 500 different states in PCM. This gave rise to the Multi-Level Cell (MLC) architecture which allows a single PCM cell to store more than one bit of data by precisely dividing the range into number of levels, which can be identified by different resistances between the chalcogenide material states. However, MLC technique is costly, leads to degradation in performance, endurance, and increment in power consumption. Whereas Single-level Cell (single bit per PCM cell) supports a higher performance and less power consumption with a longer lifetime.

Authors in [21, qiu2015] propose a genetic algorithm for task scheduling and the MLC/SLC PCM mode configuration using morph able PCM cells that can switch operations between SLC and MLC depending upon the workloads. The chromosome can have length n, representing the number of tasks in the application. The interdependencies among the various tasks are always maintained and the task to core scheduling is performed by using a simple list-scheduling algorithm. The PCM mode string corresponding to the chromosome shows the value of bits per cell in which the corresponding PCM task is to be stored. The algorithm begins by random population upon which a fitness function representing average PCM performance of application, in the terms of bits/cell is employed. Out of the population P, using rank-based roulette wheel selection scheme and removing identical chromosomes a new population is create. R pairs of chromosome are selected using same rank-based roulette wheel scheme for crossover procedure, while for mutation Q chromosomes are selected. A bit is selected randomly form each of the Q chromosome and changed to another randomly picked value. The simulation show a reduction in execution time by 10.4% when compared to a heuristic based approach, however the execution of task using SLC only, performs execution in 28000 cycles whereas the GA approach takes 45500 cycles. Also, the authors only concentrate on improving the read cycles.

Authors in [22, wang2016] focus on the write performance and endurance of the PCM, there are two operations SET and RESET in phase change operation. In RESET operation (writing 0), it uses a small current and process is short. However, when performing SET operation (writing 1), it takes a long time and uses large current since phase change material has to be heated to change the status. Authors propose hybrid architecture for main memory consisting of both DRAM and PCM. PCM is designed to store the de-duplicated data blocks and metadata. DRAM has been divided to three parts: Metadata cache- stores metadata in PCM, Memory-mark bit- to mark the PCM storage that is dirty, and Data Cache- caches data block from PCM. Therefore, whenever a DRAM cache miss occurs, a request is sent to PCM. Data de-duplication is performed every time a read/write transaction occurs. Simulations show 85.43-95.28% data reductions in idle situations.

**3.3 Single Server- Storage**

Cloud computing also provides Storage as a Service, which allows users to store the data at remote disks and access them as and when needed. With major market players like Amazon coming up with RDS and S3 for providing storage services to users, secondary storage devices like Hard Disk Drives (HDD) and Flash based Storage (SSD) can be investigated to decrease the power consumption of the servers.

**3.3.1 Hard Disk Drives (HDD)**

Hard Disk Drive is the most used secondary storage media in the data centre servers. HDD contains disk platters on a rotating spindle and read-write head floating above the platters. The three main components that consume power in an HDD are: The Spindle Motor (SPM), Voice Coil Motor (VCM) and the electronics [23, 194of07279]. However, in the context of HDD the electromechanical component SPM accounts for the major power consumption. To decrease this, traditional energy conservation techniques proposed for HDD RAIDs mainly focus on switching a subset of disks to the power off state.

Hibernator [24, 200of07279] is a disk array energy management system which meets the response time performance goals while saving on the energy consumption. Authors propose a disk-speed-setting algorithm called Coarse-grain Response (CR) which uses the observed workload to determine optical disk speed setting that minimize energy consumption without violating performance goals. This works along with an energy and time efficient data migration scheme, called randomized shuffling, that performs reconfiguration quickly and allows data layout to adopt workload changes. However, the paper uses a complex power model.

Authors in [25, tomes2017] show that HDD RAIDs are better when considering sequential write performance therefore; Authors in [26, pinhoreio2014] propose a technique called Popular Data Concentration (PDC). The idea is to concentrate the most popular (frequently miss in the main memory cache) disk data by migrating it to subset of disks. This makes other disks less loaded and can eventually be sent to lower-power modes to conserve energy. More specifically, PDC lays data across the array so that the first disk stores the most popular data, the second disk stores the next set of most popular data and so on. In fact, last few disks have data that is frequently hit in the main memory cache. To avoid performance degradation, PDC only migrate data onto a disk until the expected load on the disk is close to maximum bandwidth of the workload.

Authors in [25, tomes2017] also show that HDD RAIDs are more power hungry when the storage system is in idle state. Authors in [27, 18oflee2010] perform task consolidation using the traditional bin-packing problem with two main characteristics i.e. CPU and disk usage. The proposed technique attempts to consolidate tasks balancing and deduction in energy consumption on the basis of Pareto frontier (optimal points). The two main steps involved are the determination of optimal points for profiling data, and energy aware resource allocation using Euclidean distance between the current allocation and optimal point at each server.

**3.3.2 Solid State Drives (SSD)**

Solid State Drive employs flash memory technology which is based upon the use of floating gate transistors. Floating gate transistors are electrically isolated, therefore can hold charge i.e. data for a long time. They are strongly advocated for primary storage as SSD have better energy efficiency and faster random access when it comes to read operations. But, write operations require significant amount of energy and also damage the oxide layer (energy barrier in transistors) due to collisions. Thus, limiting the number of write cycles that can be performed.

Authors in [25, tomes2017] show that traditional power-off based energy conservation techniques applied in HDD RAIDs are not suitable for SDD RAIDs due to its high operational cost and minimal energy conservation potential. Authors also show that for all common server and workloads of web, file, and mail, SSD RAIDs are one to two orders of magnitude more energy efficient than HDD RAIDs. Authors also showed that for write intensive and mixed workloads, energy efficiency of RAID0 is consistently superior to other RAID levels, and mirroring with RAID10 provides better energy efficiency than parity based protection techniques of RAID5/6 for SSD RAIDs. However, authors didn’t consider the case of hybrid arrays where both HDD and SSD are used.

Authors in [28, 10.1109@1cc] increase the lifespan of the SDD by decreasing the randomized access to the disk and exploring sequential access patterns. The authors employ a sequence recognition module, to recognize sequential requests; they also dynamically divide the cache into RAM and SSD. The sequential requests are sent to the sequential part of the RAM where if a miss occurs, the system fetch the data from HDD arrays. The random requests are sent to the random part of the array and if a miss occurs the related data is fetched from the SSD or HDD arrays. Simulation shows that SSD performance can be improved up to 45%. However, authors use one LRU list per user for read requests and one for write requests. This can create memory contention at the cache level.

Authors in [29, gao2017] propose a Load Aware Data Migration scheme for distributed surveillance using hybrid storage architecture. Video chunks are migrated from the local HDD to the local SSD so that node performance can be improved. Also, a Cluster Level Data Migration (CLDM) is used to periodically migrate the video chunks from the high load nodes to low load nodes for achieving the overall load balance of cluster. Although authors makes use of hybrid architecture for the secondary storage yet the CLDM algorithm identifies high and low load nodes through a novice approach by comparing it with average load. Clustering and Classification techniques could have produced better and accurate results. Authors in [30, tan2014] apply the hybrid secondary storage architecture for Big Data services. A Hadoop job is a pipeline consisting of distributed file system IO, CPU processing, local IO and network transfer. Using SSD would reduce the latency in both DFS and local IO. Authors show that using SSD for storing local temporary data even in I/O intensive jobs such as TeraSort can achieve good utility. They also show that for latency bound workloads such as HBase queries, data with higher access density should be given priority to use SSD. Also, when replacing HDD with SSD, Hadoop configurations such as the number of map and reduce slots should be increased so that they can leverage the bandwidth of SSD.

**4. Cluster of Servers**

The power consumption reduction techniques defined in this paper until this point has been focussed on reducing energy consumption at individual component level. Energy consumption depends primarily on CPU utilization, and memory consumption at both disk and cache level. Efficient utilization of resources can lead to significant power savings and decrease the operational cost of the data centre.

Authors in [31, 32, 33] leverage the Particle Swarm Optimization (PSO) technique to solve the task-resource mapping problem. Authors in [31, aina2010] schedule task considering both computation and data cost, with the focus on reducing the total cost of executing the applications. A model is formulated for task-resource mapping to decrease the overall execution cost and PSO is used to achieve the mapping. The algorithm considers communication cost of all tasks, including various dependencies between them. Authors in [32, multi] take the previous work a step forward and also minimize the transfer and execution time of tasks. Authors regard task scheduling as a mapping and create a Directed Acyclic Graph (DAG) with each node having weight equal to the amount of data to be performed by the task and edge weight as the amount data transfer between two respective tasks. Authors in [33, verma2014] take the budget constraints and deadline as important factors while scheduling tasks. In Bi-Criteria Priority based PSO (BPSO) the tasks are executed in order of their priority computed using the bottom level. The assigned priority is used to initialize PSO and task are sorted in descending order and sent to different processors in order.

Authors in [34, 08166] saves on energy consumption and energy costs both by dynamically migrating the virtual machines between data centres such that one can decrease the cost incurred in executing the tasks. The optimization problem is formulated in two steps, firstly the incoming tasks are placed on the lowest-electricity cost Data centre and secondly, migrating tasks to the lowest costs data centre. The algorithm makes use of the fluctuations in the energy prices. However, the power model used to calculate the energy consumption of the server is linear and only takes into account the CPU utilization. In a different work, authors in [35, 10.1.1.] perform task scheduling as well as use renewable energy to power the servers. Authors formulate an optimization problem while avoiding server overloads and limiting the brown energy (fossil fuels) consumption. The optimization problem tends to decrease the overall cost of the data centre while maintain the SLA requirements.

When decreasing power consumption at the cluster level, power consumption of the individual servers should be decreased, which can be decreased by decreasing the power consumption of servers in the cluster and the networking interface (discussed further) involved between the servers. The individual server power consumption can be decreased by decreasing the power consumption of its main components like memory and processor.

**5. Network**

In this section we discuss about green computing techniques used upon group of servers connected via data centre network as well as scenarios such as multiple data centre systems being linked via wide area networks as in NFSNet. When modelling energy consumption at such high level of abstraction, energy consumption of communication links and the intermediate hardware even plays major role in power consumption. Green cloud computing techniques focusing on the intra data centre networks consider the entire data centre’s network energy consumption. It starts with efficient architectural planning of the data centre network, followed by decreasing the energy consumption of network elements, such as network switches, routers and transmission system. This involves running the data centre components at high utilization rates and putting the under-utilized components like switches, servers, and etc. to lower-energy modes.

**5.1 Network Architecture**

According to authors in [37, tucker] this requires efficient planning of the architecture of the network as it evolves in order to take advantage of the most appropriate technologies at each point in the network. Authors in [38, densref-1forguzek] consider a three tier tree of hosts and switches consisting of core tier at the root of the tree, aggregation tier responsible for routing, and the access tier holds the cluster of servers. The access tier uses the standard 1gbps links as they provided sufficient capacity, and rack connectivity is achieved using the Top-Of-Rack (TOR) switches. In a different work [39, chiesa2014] authors consider the CLOS network. CLOS network topology was proposed to overcome the performance and cost related challenges of switches. CLOS works on the principle that organizing the switches in a hierarchy can achieve non-blocking performance in the switching array without the need of the n-squared ports. However, both these topologies consider a finite number of levels. Authors in [40, widijaja2013] show that a fat tree network can be developed having arbitrary number of levels. A fat tree maintains constant bisection bandwidth as one traverses from switches at the bottom of the tree to switch at the root. Beside these architectures, BCube, FiConn, uFix etc. are also some of the other popular architectures.

**5.2 Network Links**

Authors in [41, 227of07279] show that the total network energy consumption can be based on the level of power consumption by a link. Another work by authors in [42,237of07279063] extended the link power consumption by including the network traffic flows. Specifically they modelled the total amount of network energy used for transmitting a group of traffic flows in data centre. Which calls for one of the main objectives of architectural planning in cloud computing is to create effective access to remote resources by optimizing the procedures of Traffic Engineering (TE) for transmission of data. All these non-blocking multipath network topologies utilize optimal routing techniques between adjacent layers to help eliminate bandwidth bottlenecks in the core layers. In optimal routing, traffic between the source and destination is split at tactical points to allow the gradual altering of traffic along the alternative paths. This removes the problem of traffic contention, particularly along the shortest paths by using alternative paths. This is where multipath routing algorithms come into play, used to identify the maximum number of disjoint paths in the graph which overcomes the problem in the junction area estimation. Authors in [38, densref-1forguzek] utilize Equal Cost Multi-Path (ECMP) routing technique in the 3 layered fat tree architecture to perform a per-flow load balancing, which differentiates the flows by computing a hash function on the incoming packet headers. Similarly authors in [39, chiesa2014] also considered the standard model of traffic with ECMP and proved that ECMP can probably achieve optimal traffic flow for the important category of CLOS data centre networks. They also addressed the shortcomings of ECMP in the suboptimal routing of large flows by presenting a suitable algorithm, for scheduling with provable approximation. Similarly, Authors in [40, 19of 4b413] consider multipath routing from TE perspective in a multi-commodity setting through linear programming. They showed that a multipath measure is zero or close to zero under certain traffic conditions and topological structures, hence implying limited multipath gain compared to single-path routing. In another work, authors in [41, 20of4b413] introduce AMPLE, which is based on offline link weight optimization. Using this, one can monitor network dynamics at short timescales, thereby coping almost optimally with unpredictable traffic dynamics.

In a similar work DENS [38, densref1forguzek], authors select the best fit computing resources like rack uplink and etc. for job execution and designed specifically for the communication potential of the data centre components. The communication potential is defined as the amount of end to end bandwidth provided to individual servers or a cluster of servers by the data centre architecture. The communication potential relies on the rack uplink buffer occupancy, such that it reacts to the growing congestion. However, DENS is strictly limited to three-tier architecture, e-STAB [47, estab-ref2forguzek] which is a two stage algorithm that requires online knowledge about the full data centre network utilization. As a result the load is better distributed among the rack opposed to that in DENS. e-STAB favours low utilization and thus prevents consolidation. However, still both of these works are based on homogenous servers, HEROES [48, guzek2015] contributes with heterogeneity-aware decision making approach. It tolerates any standard network topology, as it operates at rack level. To maximally reduce the energy consumption, server selection function promotes DNS, but prevents too high utilization levels. The decision making is based on aggregated information and characterized by low computational complexity.

**5.3 Network Devices**

In the skeletal structure of the data centre, network devices such as routers, switches etc. play a main role. In most of the situations networking devices such as routers are provisioned for peak loads, yet they operate at a low average utilization levels consuming 80%-90% of the peak power [43, 238of0727]. Significant energy savings can be gained if the transport system and network switches operate at high utilization rates. The systems need to dimension to handle high traffic, but putting under-utilized network elements or some of their components like ports in switches can achieve energy savings. Authors in [44, mahadevan2010] show that energy consumption of a switch depends upon the number of active ports and their operation speed. This same model was used by the authors in [38, dens-ref1forguzek] to measure the power consumption of a switch. In a different work, authors in [45, 247of072790] consider the power consumption of the edge routers. They found that the power consumption is directly proportional to the link utilization and the packet sizes. Therefore, authors in [46, 5ofwidijaja] show that inter-server communication energy cost can be proportional to the amount of data being transferred and gives an algorithm to dynamically tune the individual links to match the required load while consuming as little power as possible. Also, authors in [40, widijaja] show that a topology with many small switches is more energy efficient than topology with large switches when the servers are communicating in close proximity. They also determine the right switch sizes required in the network topology considering the fact that number of servers required is given. They propose the use of sleep mode for the switches to achieve significant network energy savings.

**5.4 Virtualization at Network level**

Virtualization technology was developed to maximize the utilization of the hardware resources as the expensive mainframe computers were going underutilized. The main aim of virtualization is to run multiple VMs parallel on single server. To improve the Cloud Data centre efficiency, different type of resource management strategies like server consolidation, load balancing, server up-gradation are applied through migration of single/multiple VMs. This helps in combining the scattered server workload onto a few physical servers and switch off the idle one. This can lead to significant energy savings. Besides this, to keep up with SLA requirements, running VMs can be migrated from a low performing to better performing physical servers. But, it should be seen that running different VMs on few servers shall not create resource contention and hence not lead to performance degradation. To resolve this issue, often hypervisor migrate VMs from over-utilized servers to normally-utilized servers and from under-utilized servers to the moderately-utilized servers. Switching off the under-utilized servers can make significant energy savings as servers in idle mode consume 70% of their peak power [47, 13ofcompletesurvey].

This opens dimensions for research, VM consolidation. VM consolidation consist of identification of under and over utilized servers, followed by identification of VMs for migration and then lastly selecting a server for it to be migrated upon. All this is done while following the SLA and avoiding resource contention and performance degradation. Authors in [48, khoshkholghi2017] perform power aware VM consolidation by detecting overloaded hosts while keeping in mind all the three resources: CPU, RAM and bandwidth. If either of the resources is utilized above the dynamic threshold calculated using Iterative Weighted Linear Regression (IWLR) the host is marked as overloaded. Authors further propose a policy named maximum power reduction policy, time and power trade-off policy, and violated-mips VM policy for selecting the VMs to be migrated. This is followed by the last step, placing the selected VMs on the host using the VM placement algorithms like best RAM and bandwidth placement algorithm. They also consider under-utilized server for the VM migration so that they can be moved into sleep modes. Although the authors propose various approaches providing a complete view of the VM consolidation process considering all the resources involved like RAM, CPU and bandwidth. The algorithm considers RAM as the total amount of memory to be transferred and ignores all the other types of memory contents completely. In a different work [49, chaudhry2016], authors focus upon finding VMs for migration and the destination host for VMs using the Bin packing problem. However, the authors only consider CPU utilization as the only factor; they completely ignore the memory and network components. Just like authors in [48, khoshkholghi2017] authors in [50, shaw2017] also uses smoothing factor in the linear regression to predict the VMs to be migrated based on the current and future loads. They use Holt-Winter Double exponential smoothing technique. They detect the overloaded hosts using the dynamic threshold whose values are found using Median Absolute Deviation (MAD) technique. However, the authors only focus on CPU utilization and also they migrate VMs only if the prediction and present state both signify overload status and ignore else wise they leave the server in overloaded state and do nothing. This can lead to future performance degradation due to contention in resources.

In different works authors have also used sophisticated Machine Learning techniques like Clustering, Classification, Neural networks to solve the VM consolidation problem. Authors in [51, rp2] use k-means clustering technique to segregate the servers into four clusters little, light, moderate and over loaded servers. Then they select VM from little and over loaded hosts on the basis of the migration factor and migrate it to a destination host. However, the selection process isn’t autonomous and requires the intervention of cloud service provider (CSP). In a different work, authors in [52, 7of SDClouds] propose a VM management method considering the network topology specifically. A VM with a high level of network usage is migrated to the destination host to reduce the network cost for VMs and the traffic over the data centre. The algorithm calculates the current communication cost for each VM-to-VM flow and estimates the new communication cost if the VM is migrated onto the other end of the flow. If it can find a new location for the VM which can reduce the total communication cost, it is migrated. However, the approach does not consider the capacity of the computing nodes so a large VM may never be migrated. Also, the migration targets are limited as we have to consider hosts which have similar VM of the flow. Therefore Authors in [53, SDClouds] have considered a group of hosts instead of individual hosts, as a candidate for the VM placement. Besides that the VM placement algorithm proposed works for heterogeneous cloud infrastructure and is also topologically aware.

Live VM migration consist migration of the memory content, which involves the VM configured memory (physical memory allocated by the hypervisor to the VM), VM used memory, application requested memory and application actively dirtied memory. Besides this, if LAN connection uses Network Attached Storage (NAS) and the migration is performed over the LAN within the data centre there is no need to transfer the storage contents. But, if the transfer is over a WAN, to some other data centre storage contents need to be transferred. Storage contents mostly include Virtual disk size, VM used Blocks and Hypervisor Allocated blocks. Live VM migration consists of pre-copy techniques and post-copy techniques. In pre copy, initially all pages are transferred while VM runs continuously runs at source host, in further iterations dirty pages are resent and the iterations are terminated when the number of iterations exceed pre-defined iterations, or total amount of memory sent surpasses the threshold or the page dirty rate falls below a certain threshold. Once the iterations terminated post-copy phase begins, VM is suspended at source server and processor states and remaining dirty pages are transferred. Then VM is resumed at the destination host. In post copy technique, processor state is transferred before the memory content and the VM could be started at the destination server. Post copy techniques make use of demand paging, active push, pre-paging and dynamic self- ballooning.

In live migration of virtual machines a single VM can also be migrated, multiple VMs can also be migrated. Authors in [54, rp3] propose a live migration based on the time series analysis done using the artificial intelligence based simulated annealing concept. Authors avoid migration of highly dirtied pages in the selection phase and also limits the number of pages to be migrated by only migrating the pages with probability higher than 0.75. Authors terminate the live migration and begin post-copy when the live migration time increases at the expense of the downtime thus, determining the iterations to be performed dynamically. However, the authors assume a single VM is migrated at a moment as the entire bandwidth of the link is given to the VM. In a different approach where Multiple VMs are simultaneously migrated, techniques become complex due to reasons like insufficient resources at destination server and concurrent migration of VM. Authors in [55, 86ofCompleteSurvey] try to achieve optimal scheduling of multi-tier VM and propose vHaul to control multi-VM migration and figure out an optimal scheduling strategy. They are able to minimize the service downtime by 70% and also improve application throughput by 52%. In the similar field of multiple VM migration authors in [56, 88ofcompletesurvey] perform load balancing and to achieve this perform live migration of VMs in their proposed dual strategies: push and pull. Simulations performed in the OMNet show that a load balanced system can be achieved into 4-15 minutes. Authors in [57, 90ofcompletesurvey] focus specifically upon the problem of multiple VM migration and propose a serial migration strategy called M mixed migration strategy and developed queuing models. Authors performed detailed analysis by measuring performance metrics like average waiting time, blocking ratio, average waiting queue length, and average queue length of each migration request.

While the VM is being migrated, Virtual Machine Monitor detects multiple copies of same page on single or multiple VMs or different number of servers; this is leading cause for useless memory pages migration. This requirement of large number of pages to be migrated can lead to increase in requirement of the bandwidth or increase network traffic. This has called for various memory compression techniques. One such technique is replication based strategy in which same memory page is spread on multiple servers for simultaneous computing and fault recovery. Authors in [58, 94ofcompletesurvey] focus on migrating large sized VM over a WAN which has low bandwidth effectively. To achieve this they combine VM replication with VM scheduling and use de-duplication techniques. Algorithm is able to minimize the migration latencies of the VMs. Another technique is De-duplication based VM migration where one identifies the similar memory pages on a single VM and avoids transfer of such pages, this leads to effective bandwidth utilization. Authors in [59, 99ofcompletesurvey] reduce the total time of VM migration by detecting duplication between memory pages and storage blocks while also maintaining an up-to-date map of duplicated pages. This leads to 30% reduction of total migration time and 60% reduction for certain benchmark overloads. Another important technique is Redundancy based VM migration where identical memory blocks belonging to different VMs on the same server or large blocks consisting of zero bytes entries. The avoidance of transferring redundant pages leads to reducing power consumption, load and cost of live VM migration. Authors in [60, 101ofcompletesurvey] reduce the downtime and cost of VM migration using the concept of distance and redundancy elimination mechanism. The proposed algorithm reduces the power consumption, load and cost of VM migration. Another technique is compression based techniques in which the memory compression leads to data transfer during migration process, this reduces cost of transferring VM memory, storage contents during migration, and service downtime gets reduced. Authors in [61, 103ofcompletesurvey] propose MECOM, which live migrates VM by adaptively compressing memory pages using the adaptive zero-aware compression algorithm. This reduces the transfer data, total migration time, and downtime.

**6. Power Management**

**Conclusion and Future Scope**