CLUSTER ENERGY OPTIMIZATION: A THEORETICAL APPROACH

Vikram Yadav [#1], G. Sahoo [#2]
[#1] Computer Science Department , Birla institute of technology
Mesra, Ranchi, India
[#2]Information Technology , Birla institute of technology
Mesra, Ranchi, India
[1] vikram.yadav117@gmail.com
[2] gsahoo@bitmesra.ac.in

***Abstract*— The optimization of energy consumption in the cloud computing environment is the question how to use various energy conservation strategies to efficiently allocate resources. The need of different resources in cloud environment is unpredictable. It is observed that load management in cloud is utmost needed in order to provide QOS. The jobs at over-loaded physical machine are shifted to under-loaded physical machine and turning the idle machine off in order to provide green cloud. For energy optimization, DVFS and Power-Nap are good strategies. As much of this energy is wasted in idle systems: in typical deployments, server utilization is below 30%, but idle servers still consume 60% of their peak power draw. In this paper, we have proposed an hybrid approach for energy optimization using Ant Colony optimization, Bee Colony optimization, PowerNap, DVFS and RAILS having the constraint QOS .**

Keyword- Cloud Computing, DVFS, Power-Nap, Ant colony algorithm, bee colony algorithm, RAILS etc.

I. INTRODUCTION

The average utilization for a typical industrial data center is as low as 20 to 30 percent - and the computers are still using 60 percent of peak power even when they're doing nothing. To combat this problem, researchers at the University of Michigan have put together an approach that could save up to 75 percent of the energy that power-hungry computer data centers consume. The approach, developed by Thomas Wenisch, David Meisner and Brian Gold, includes PowerNap, an energy-saving plan to put idle servers to sleep, and a technique dubbed Redundant Array for Inexpensive Load Sharing (RAILS), which supplies power more efficiently. To cut down on the power loss, RAILS would replace the one 2,250-watt power supply with a bunch of smaller, 500-watt power supplies. RAILS would be a necessary complement to PowerNap because without it, even sleeping servers would waste energy. Wenisch [1] summarized that both of these approaches can help make data centers green and solve these big energy efficiency challenges.

Organization of this paper is as follows: Related Work is discussed in section 2. Server architecture of a cluster is discussed in section 3. Proposed methodology is discussed in section 4. Proposed algorithm is discussed in section 5. Section 6 gives the conclusion with a direction of the future work.

II. RELATED WORK

David Meisner [2] presented a power nap strategy for eliminating server idle power. Liang Luo [3] presented a resource scheduling algorithm based on energy optimization methods. Eugen Feller [4] summarized the energy consumption and adjustment methods of standalone computer and cluster servers. In Chung-Hsing Hsu’s work [5], They continuously observed the energy consumption and server utilization of server from 2007 to 2010, pointing out that server’s defaults energy model has changed. In Charles lively & Xingtu Wu’s work [6], they use precise experimental probes to collect energy consumption and performance characterstics in different parallel implementations. HoAydin et.al [7] have proposed the minimizing the energy consumption and subsequently the cost for the static system. K.Mukherjee & G. Sahoo [8] have proposed a framework for achieving better load balancing in Grid environment. Again, K. Mukherjee & G. Sahoo [9] proposed an ant colony optimization and Bee colony for service rescheduling. Once again, K. Mukherjee & G. Sahoo [10] proposed an algorithmic approach for Green Cloud.

III. SERVER ARCHITECTURE

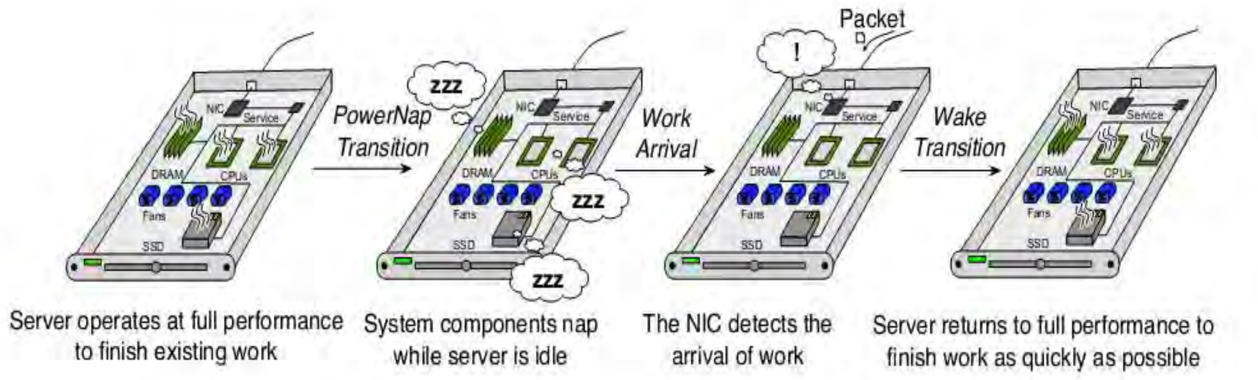


FIG.1 SERVER WORKING STEPS

The rapid transitions and brief intervals of server activity make it difficult to conserve idle power with existing approaches. The recent trend towards server consolidationis partly motivated by the high energy cost of idle systems. By moving services to virtual machines, several services can be time-multiplexed on a single physical server, increasing average utilization. Consolidation allows the total number of physical servers to be reduced, thereby reducing idle inefficiency. However, server consolidation, by itself, does not close the gap between peak and average utilization. Data centers still require sufficient capacity for peak demand, which inevitably leaves some servers idle in the average case. Furthermore, consolidation does not save energy automatically; system administrators must actively consolidate services and remove unneeded systems. Although support for sleep states is widespread in handheld, laptop and desktop machines, these states are rarely used in current server systems. Unfortunately, the high restart latency typical of current sleep states renders them unaccept able for interactive services; current laptops and desktops require several seconds to suspend using operating system interfaces (e.g., ACPI). Moreover, unlike consumer devices, servers cannot rely on the user to transition between power states; they must have an autonomous mechanism that manages state transitions. Recent server processors include CPU throttling solutions (e.g. Intel Speedstep, AMD Cool'n'Quiet) to reduce the large overhead of light loads. These processors use DVFS to reduce their operating frequency linearly while gainingcubic power savings. DVFS relies on operating system support to tune processor frequency to instantaneous load. In Linux, the kernel continues lowering frequency until it observes~20% idle time. Improving DVFS control algorithms remains an active research area. Nonetheless, DVFS can be highly effective in reducing CPU power. However CPUs account for a small portion of total system power. Energy proportional computing seeks to extend the success of DVFS to the entire system. In this scheme, each system component is redesigned to consume energy in proportion to utilization. In an energy-proportional system, explicit power management is unnecessary, as power consumption varies naturally with utilization. However, as many components incur fixed power overheads when active (e.g., clock power on synchronous memory busses, leakage power in CPUs, etc.) designing energy-proportional subsystems remains a research challenge. Energy-proportional operation can be approximated with non-energy-proportional systems through dynamic virtual machine consolidation over a large server ensemble. However, such approaches do not address the performance isolation concerns of dynamic consolidation and operate at coarse time scales (minutes). Hence, they cannot exploit the brief idle periods found in servers.

IV. PROPOSED METHODOLGY

In the proposed approach, we have used following existing algorithms for different purposes.

A. Bee Colony optimization for Load-balancing.
B. Ant Colony optimization for Traversing all nodes.
C. PowerNap and DVFS for energy optimization.
D. RAILS for distribution of power.

A. ANT COLONY OPTIMIZATION

Ant Colony optimization technique was proposed by Marco Dorigo in the early of 90's. In our problem, there are n nodes in a cluster. Some are under-loaded, over-loaded and idle. Ant colony optimization will help in traversing all nodes by shortest path(Traveling salesman problem). After traversing all the nodes, we will find the nodes that are idle and we will not make them turn off. In our practical implementation, we have provided a color code (RGB code) to each node. We will set a Threshold value like summation of RGB must be less than or equal to 300. The node will be turn off or on on the basis of RGB scale. Ants (blind) navigate from source to food with the help of Pheromone. Initially, the ants navigate in all possible paths from source to food uniformly. On the long path, ants take more time to return as there will be more distance between the ants. So, the evaporation will be high as compared to small path, there will be less distance between the ants and the evaporation will be less. In the next iteration, more ants will navigate in that path that has more pheromone means shortest path. So, shortest path is discovered via pheromone trails. Each ant move in random, Pheromone is deposited on path. Ants detect the lead ants path inclined to follow more pheromone path. The probability of path being followed starting node are selected at random path selected at random based on amount of trail present on possible paths from starting node. The path that has higher probability has more trails. Ant reached next node, select path and continues until reaches start node. Finished tour is a solution. A completed tour is analyzed for optimality. Trail amount adjusted for favor better solution. A better solution receive more trails. A worse solution receive less trail. Higher probability of ant selecting path that is a part of better solution. A better solution receive more trail. A worse solution receive less trail. Higher the probability of ant selecting path that is a part of a better performing tour. New cycle is performed repeated until most ants select the same tour on every cycle.

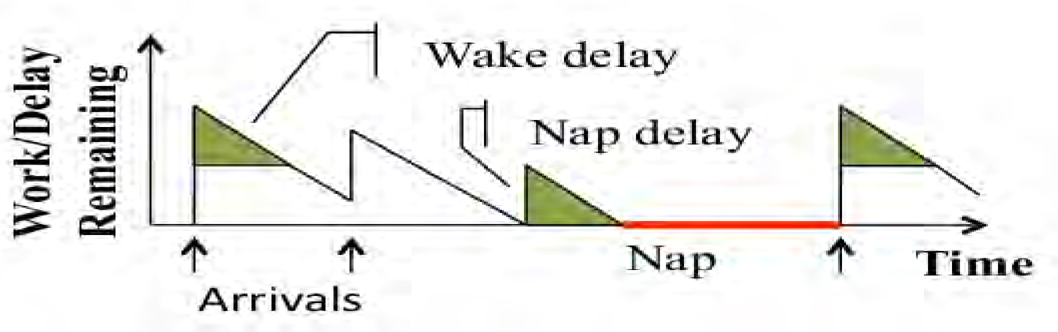
B. BEE COLONY OPTIMIZATION

Bee Colony optimization is used for load balancing. There are many nodes that are under-loaded and some are overloaded. We will uniformly distribute the load among all the node. Honey bee is a social insect. They work in a decentralized and well organized manner . There are two types of bees: One is forger bees, who collect the nectar and food-stores, who store that in hives. Forger bees moves out for searching the nectar(food). They move randomly in any direction. After finding the nectar, they come back on the hive and start dancing on the dance floor. The duration of this dance is closely correlated with the search time experienced by the dancing bees. There are two types of dance, waggle dance which implies poor quality of nectar and tremble dance (round dance) which implies good quality of nectar. If the dance is tremble dance, then the new born bee agents fly to collect the good quality nectar and store them in the hive. After this operation, the old bee agents die and the new born bee agents start to fly with the good quality nectar stored in the hive, and finally mix them with those sources which are holding poor quality nectar. This process of distribution goes on until there is a uniform quality of nectar in all the sources. Similarly, in cloud computing environment, we observe that some CPUs of IaaS are overloaded for processing consumers services, some are under loaded and some are totally idle. We can save the consumption of energy by turning this idle CPUs OFF and rescheduling services from overloaded CPUs to under loaded CPUs.

C. POWERNAP

It is a state when only the system components that detects the arrival of new work is only powered except these units all the other high power consumption units like the CPU, DRAM and all the processing units are turned off and don't utilize power. It can be utilized to achieve enormous energy saving during idle states. It has two operating modes active mode and nap mode.

Mechanism



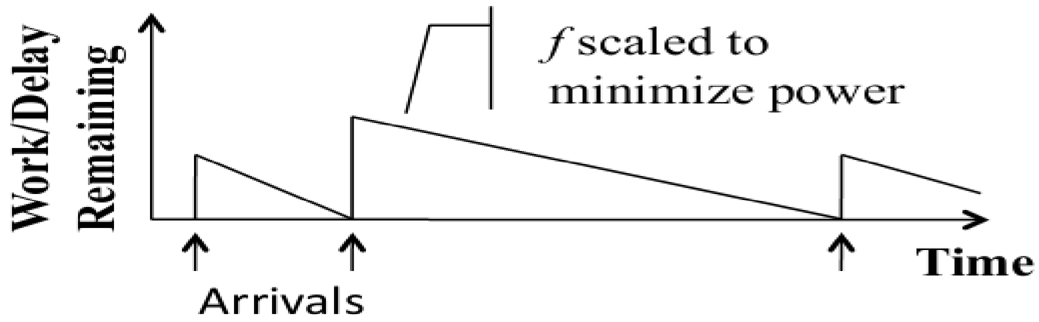
Wake delay: This is the time duration taken by the processing unit to turn on all it's services from the state of nap. Till, the wake delay ends. The work remains fixed.

Nap delay: This is the time allotted between the period when the work/job become zero to that of the nap's starting period.

Explanation: Initially, the cluster is at nap. When the work arrives the services, the servers are powered on and the time take to turn it on from the nap state is referred as wake delay. During this wake delay, the work remains constant and after it all the servers starts functioning and executing the jobs. Again, when the jobs ends/finishes, then the cluster for jobs for a definite time, after which it sends the servers back to nap. This definite amount time is referred as Nap delay. After which the nap remains till any further work is notified.

D. DVFS (DYNAMIC VOLTAGE AND FREEQUENCY SCALING)

This model of job distribution eliminates any idle time arrival between two consecutive jobs. This is achieved by stretching time existing job to fill any idle time until the next job arrives to implement it a prior knowledge of the upcoming jobs should be known.



Explanation: Initially a certain amount of work is taken in the cluster and when it ends, it creates an idle time state for the servers, we have DVFS to scale the voltage and frequency such that the time taken by the servers can be modified, and such intermediate idle states are avoided. This is possible only when the upcoming work's arrival time is predicted or known. So, for each work arrival the previous work is stretched till the new work is arrived.

E. RAILS (REDUNDANT ARRAY OF INEXPENSIVE LOAD SHARING)

It is basically the replacement of the general(common) Power Supply units(PSU) with a combination(parallel) of multiple inexpensive PSU's. This is very feasible for PowerNap since in PowerNap state a very small energy is requirement and with need each PSU is activated and power is taken into the consequently activated servers.

Key features of RAILS are:

1. Efficient dynamic power output switching.
2. Tolerance of PSU failure by using N+K model.
3. Minimal cost.

V. PROPOSED ALGORITHM

In a cluster, many kind of servers like compute servers, data servers (data centers), switching nodes etc. are present. Some nodes may be under-loaded, over-loaded or idle. We have to maxi-mize the server utilization having the constraint energy optimization and QOS. Initially, we will use Ant colony optimization for finding the shortest path among all the nodes and differentiate the nodes into over-loaded, under-loaded and idle.

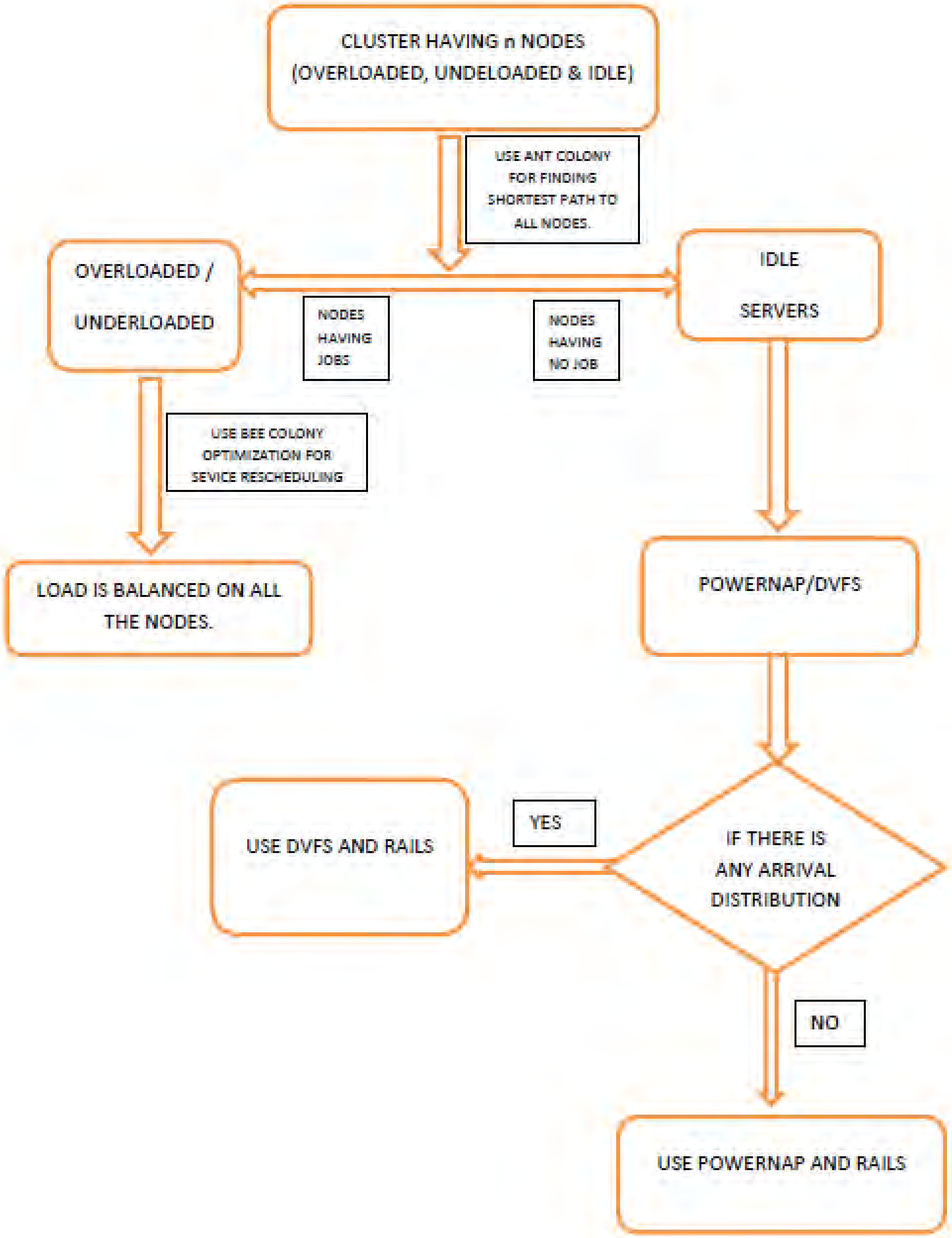
After that we will apply Bee colony optimization for load balancing or service rescheduling. We will not turn off the idle servers. A statistical analysis is done on the arrival jobs interval. We will see: Is it following any distribution i.e. poission, exponential, erlang, normal etc., we will use DVFS model else PowerNap incorporated with RAILS.

ALGORITHM

Clster_optimiz()

```
Begin
  traverse shortest path among n nodes using ACO
    search for idle, over-loaded & under-loaded nodes.
     for(idle nodes)
      if(arrival time distribution is identifiable)
       use DVFS and RAILS
       else
       use PowerNap  and RAILS
     endfor
     for(under-loaded & over-loaded nodes )
       load balancing or service rescheduling using BCO
     end for
End
```

FLOWCHART



VI. CONCLUSIONS WITH FUTURE DIRECTION OF WORK

From the above theory, we can observe that idlepower costs are high, idle period length too short for conventional shutdown and system wide coordination simplifies power management. We can conclude that PowerNap: eliminating server idle power and RAILS provides PSU efficiency at low cost. We can use statistics for finding the next job arrival using distribution. Powernap has better power saving schemes as well as better response time as compared to DVFS. Due to Power nap unique power requirement, we have introduced RAILS, a novel power delivery system that improves power conversion efficiency, provides graceful degradation in the event of PSU failures, and reduces cost.

REFERENCES

[1] http://www.gizmag.com/powernap-save-data-center-energy/11218/.
[2] D. Meisner and B. Gold, “PowerNap: Eliminating Server Idle Power”, ACM 978-1-60558-215-3/09/03.
[3] Liang Luo, Wenjun Wu, “A Resource Scheduling Algorithm of Cloud Computing based on Energy Efficient Optimization Methods”, IEEE-2010.
[4] E. Feller, D. Leprince and C. Morin, “State of the art of power saving in clusters results from the EDF”, case study. 2010.

[5] C. H. Hsu and S. W. Poole,“Power Signature Analysis of the SPECpower_ssj2008 Benchmark[C] Performance Analysis of Systems and Software (ISPASS)”, 2011 IEEE International Symposium, 2011: 227-236.
[6] C. Lively, X. Wu, V. Taylor, S. Moore, H. Chang and K. Cameron,“Energy and performance characteristics of different parallel implementations of scientific applications on multicore systems”, International Journal of High Performance Computing Applications, 2011, 25(3): 342-350.
[7] H. Aydin, R. G. Melhem D. Mosse and P. Mejia-Alvarez. “Power-Aware Scheduling for Periodic Real-Time Task”, IEEE Transactions on Computers, May 2004, , 53(5).
[8] K. Mukherjee and G. Sahoo, “A framework for achieving better load balancing and job scheduling in Grid environment”, International Journal of Information Technology and knowledge Management, volume 2, No - 1, pp- 199- 202, January- June, 2009.
[9] K. Mukherjee and G. Sahoo,“A framework for achieving better load balancing and job scheduling in Grid environment”, International Journal of Information Technology and knowledge Management, Volume 2, No - 1, pp- 199- 202, January- June, 2009.
[10] K. Mukherjee and G. Sahoo,“Green Cloud: An Algorithmic Approach”, International Journal of Computer Application Volume9 – No.9 November 2010.