ISSN:0975-9646

Power Aware Hybrid Proxy Cache-Prefetch Model using Combined Energy Conservation Policies

Sirshendu Sekhar Ghosh, Aruna Jain

Department of Information Technology, Birla Institute of Technology, Mesra
Ranch-835215, Jharkhand, India

***Abstract*— The World Wide Web (WWW) is growing exponentially in terms of number of users and number of Web applications. Due to enormous traffic in the network and several factors like bandwidth availability, request processing time at server, round trip time and object size, the Web latency is increasing. The sophisticated integration of Web prefetching and caching deployed at proxy server with Web log mining technique is the most attractive and successful solution to reduce this latency and improve Web Quality of Service. To provide uninterrupted services to the Web users', Web servers at the data centres are seamlessly operating always in 24X7 mode. Low power consumption and efficient energy management of individual hardware components, system software, to network applications is a critical and urgent issue in today's highly demanding eco-friendly computing world. In this paper, we incorporate an energy efficient feedback driven control framework along with sleep proxy mechanism in our Hybrid Cache-Prefetch System Model. The proposed model periodically monitors proxy server performance parameters dynamically. We have also developed an optimized load allocation algorithm with the proposed combined strategy for proxy servers' cluster along with some guidelines for energy efficient design solutions towards green IT. Even though we emphasize power saving as much as possible the performance of proxy servers is ensured without sacrificing on client experiences. Experimental results show that using our proposed scheme 28% less power consumption and 18% power efficiency improvement can be achieved.**

***Keywords*— Proxy Cache-Prefetch model, Power and Energy Estimation, Feedback Driven Control Framework, Sleep Proxy, Optimal Load Allocation.**

I. INTRODUCTION

Rapid and explosive growth of the WWW in terms of number of users [1] and number of Web applications [2] has made the network prone to congestion and has increased enormous load on servers. The obvious effect of this scenario resulting in day by day increase in the access times of Web documents i.e. Web latency or User Perceived latency (UPL). A sophisticated integration of Web prefetching and Web proxy caching can significantly reduce the UPL by predicting and storing the next future Web objects to be accessed by users into a proxy cache [3,4]. Web proxies employing combined prefetching and caching also reduce bandwidth consumption and underutilization, network congestion and traffic, improves reliability, can effectively serve more users' requests, reducing the heavy workload from the origin Servers and also protecting them from the "flash crowd" events. We have designed and implemented an integrated Cache-Prefetching System Model Architecture [5] using a novel Hybrid cache replacement policy [6] deployed at proxy server for improving Web Quality of Service towards Web latency reduction. In present days and in future, Eco-friendly or Green computing will be a key challenge for both information technology and business which aims at environmentally sustainable computing and responsible use of computers and related resources. With the limited primary sources of energy and rapid climbing of energy demanded by computing, the commitment to reduce power consumption, efficient energy usage and environmental impact becomes increasingly important. We have to consider green-computing technologies to save unnecessary energy usage and reduce carbon dioxide (CO_2) emissions by minimizing the power consumption of electronic devices, from home appliances to servers at data centres. Based on the report of the US Environmental Protection Agency (EPA), "the servers and data centres in USA alone consumed about 61 billion kilowatt-hours (kWh) at a cost of $4.5 billion, which was about 1.5% of the total U.S. electricity consumption in 2006, and this energy consumption is expected to double by 2011 if continuously powering computer servers and data centres using the same methods [7]." Data centres use a lot of energy, nearly 3% of the electricity consumed in the United States, according to an EPA report to Congress. Web servers at the data centres are always operating in 24X7 mode to provide uninterrupted services to the Web users' explosive and always accessing demand. Because Web servers are at the core of data centres, so the power they consume and the heat they generate drives air conditioning costs are the prime target for energy-savings measures. In our previous work we have designed and implemented a Hybrid Proxy Cache-Prefetch Model with energy efficient prefetch criteria and some energy proportional designs for Web Optimization [8]. Concerning today's highly motivating goal to minimize the overall impact of Information and Communication Technologies (ICT) on the environment and desperate need for eco-friendly performance-power aware server designing objective towards green IT, in this paper we have proposed an energy efficient operation scheme to be deployed at proxy server. It is a sophisticated combination of dynamic feedback control framework and sleep proxy mechanism. We have also developed an optimized load allocation algorithm with the proposed combined strategy for proxy servers' cluster along with some guidelines for energy efficient design solutions. The

rest of this paper is organized as follows: Section 2 reviews the related work and outlines the motivation and contribution of this work. Section 3 describes the Proxy Server Performance-Power Model. Section 4 explains the design and work functionality of Proxy based combined Energy Efficiency module along with optimized load allocation algorithm. Section 5 shows the experimental setup and results discussion. The conclusion along with future work direction is discussed in Section 6.

II. RELATED WORK

The goal of incorporating eco-friendly designing and energy efficiency is to minimize the overall impact of information and communication technologies (ICT) on the environment. The global CO_2 emission in 2009 was estimated at 30.398 billion metric tonnes [28]. Gartner et al. [29] estimates that ICT is responsible for 2% of the global CO_2 emissions due to its power consumption. This level of consumption is equivalent to around $50 billion (considering $0.15/kWh). To deliver effective solutions to the energy efficiency problem, the following considerations can be taken as the solution design guidelines.

- *System Components Comprehensive Monitoring:* To save power consumption, we shall first investigate where the power is spent and how to optimize the power usage. Within a computer system acting as proxy, there are generally four energy consumers, namely, processor, disk, memory and I/O devices. Achieving energy efficiency requires improvements in the energy usage profile of every system component.
- *Power-Manageable Hardware Components Incorporation:* Incorporating power-manageable hardware components could help improve energy efficiency. E.g., the voltage of hardware components can be increased or decreased through dynamic voltage scaling (DVS), which is a power management technique in computer architecture, depending upon circumstances. Dynamic voltage scaling to decrease voltage is known as undervolting, and this situation can conserve power [30]. Employing small form factor disk drives, solid state disk drives, large memory configurations, low power processors and memories could decrease power consumption [31]. Web server solution providers like HP and IDC also estimated that about 69% energy reduction can be achieved within a three-year period for IT organizations that migrate to blade self-contained architecture, where blades can span from servers and storage devices to workstations and virtual desktops [32,33].
- *Building Power Models for Computing Systems:* One needs to know how a computing system is constructed and how an energy efficient system operates. It is important to construct a power model that allows the system to know how the power is consumed, and how the system can manipulate and tune that power [34].
- *System Performance Understanding and Measuring:* To counter for performance with the least power consumption, computing systems must have ways to timely understand and measure system performance related to task execution under different dynamic workloads.
- *Constructing Energy Optimizers:* The system must accommodate an energy optimizer component, which is responsible for an energy efficient hardware configuration throughout the system operation at all times. The optimization approaches may be based on either heuristic or analytical techniques, as indicated by Brown et al. [34].
- *Reducing Peak Power:* Barroso et al. [35] explained that current desktop and server processors can consume less than one-third of their peak power at very low activity modes, which can thus save around 70 percent of peak power. Tsirogiannis et al. [36] indicated that almost 50 percent of peak power is actually consumed at idle.

Baliga et al. [9] estimate that the Internet consumes 1% of the total power consumption in broad-band-enabled countries. This consumption could increase to 4% as the access rate increases. Bianzino et al. [10] determine that browsing a single Web page consumes, on average, 4.7W (instantaneous power, i.e. overall energy consumption (kWh) is computing by the product of instantaneous power and browsing time), which can grow to 16W in the case of a streaming video. This is comparable to the power consumption of a single energy efficient bulb over the same time period. In 2009, Google released information about the power consumed during an average Google search [37]. They claim that 0.0003 kWh of energy is consumed per search. Considering an average of approximately 300 million searches per day, the search company consumes 90,000kWh per day, i.e. 32,850,000 kWh per year. Most of the early works related to server power management [11,12,13,14,15,16,17,18] has either focused on the processor specifically or used heuristics to address base power consumption in server clusters [19]. This motivates us to adopt a holistic strategy for the entire proxy server level power management where we exploit the system components interactions and dependencies between different devices that constitute a whole computing system. Dynamic Voltage/Frequency Scaling (DVFS) has been developed as a standard technique to achieve power efficiency of processors [20,21,22]. DVFS is a powerful adaptation mechanism, which adjusts power provisioning according to workload in computing systems. Horvath et al. [38] explored the benefits of dynamic voltage scaling for power management in server farms. They also try to minimize the total energy expenditure subject to soft end-to-end response time constraints. However, this work solved the problem from hardware viewpoint, and very complex and not easy to apply in the real application. Lorch et al. [22] use predictive models to estimate future load and create a power schedule. Unfortunately, the CPU currently contributes less than 50% to overall system power, thus we focus on whole system power management. A control-based DVFS policy combined with request batching has been proposed in [20], which trades off system responsiveness to power saving and adopts a feedback control framework to maintain a specified response time level. A DVFS policy is implemented in [21] on a stand-alone Apache Web server, which manages tasks to meet soft real-time deadlines. Virtual machines can be used to dynamically add or remove machines in response to change in load [23]. Virtual machines take minutes to boot or

migrate and introduce performance over-heads. Kansal et al. [24] focus on power consumption estimation of virtual machines running on the same physical machine. Since the performance counters can be monitored separately for each virtual machine, they attempt to segregate the power consumption. Their estimation achieves accuracy, with errors within 0.4W - 2.4W. Several recent papers [25] have used machine learning to dynamically provision virtual machines while maintaining quality of service goals. Felter et al. [26] addresses base power consumption for Web servers by using a power-shifting technique that dynamically distributes and maintains power budget among components using workload sensitive polices. Contreras et al. [27] present a power estimation model for the Intel XScale® PXA255 processors. The approach exploits the insight into the internal architecture and operation of the processor. It achieves accuracy within an average of 4% error using five processor performance counters. Christensen et al. [39] was first introduced the Sleep Proxy idea. An interesting paper [40] comes in 2007 for an implemented Proxy specifically designed for Universal Plug and Play Protocol (UPnp). Nordman et al. [40] proposed solution for a Sleep Proxy which can manage protocols such as ARP, DHCP, ICMP. Somniloquy et al. [41] gave another scheme similar to Sleep Proxy which offers a hardware implementation of Sleep Proxy in a so-called “gumstix”, thought as a predecessor of future NIC.

III. PROXY SERVER PERFORMANCE-POWER ESTIMATION MODEL

Our proposed Power-Performance model is a combination of proxy based Power and Performance estimation model. Among them, the Power estimation model estimates power changes and energy consumption of proxy servers working in a cluster serving users in the network in different states and loads and also shows which factors can influence the power consumption.

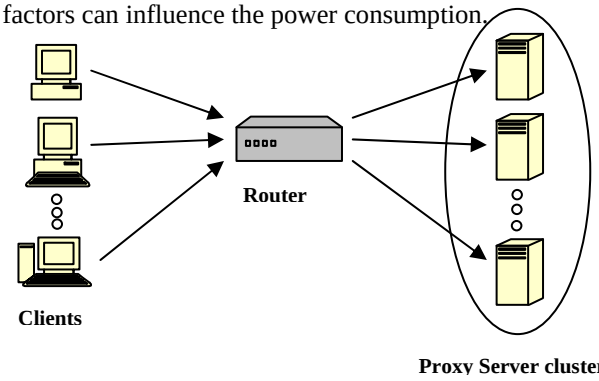


Fig. 1 The network architecture of the client-proxy server cluster system model

For Performance estimation model of a proxy server, we discuss which parameters influence the response time and also estimate performance improvement of proxy server. This model must accomplish the combined design architecture of proxy based feedback driven control framework and sleep proxy mechanism. Fig. 1 shows the basic network architecture of the client-proxy server cluster system model containing three parts: users, a router and the proxy server cluster. The principle of these three parts works as follows: Users issue requests to router acting as load balancer which selects one of the proxy servers from the cluster depending upon the load and then allocates the request to the chosen proxy server. Upon receipt of a service request from a user, a proxy server provides amount of the corresponding service with the certain energy consumption. In general, the electric power consumed by a proxy server is related with the basic running software and hardware. A server consumes energy dependently of the server state which is either SLEEP or ACTIVE. Generally, the basic power consumed by a proxy server’s ACTIVE state is much more than that in the SLEEP state. In general, the larger the current load burdens, the larger power a server consumes. Based on the above discussion the Power estimation model can be defined as follows:

$$P_i(t) = f_i(\, l_i(\, t\,)\,) + P_i^{ACTIVE} \quad ; \text{for } l_i(t) > 0$$
$$= P_i^{SLEEP} \quad ; \text{for } l_i(t) = 0 \qquad (1)$$

where $P_i(t)$ denotes the amount of electric power a proxy server i consumes at time t, fi denotes the function to show the relation between consumed power and the load, li(t) represents the load of the server i at time t and finally P_i^{ACTIVE} and P_i^{SLEEP} are the power consumptions of the server i in ACTIVE and SLEEP state. In general, if the load is more, the larger the power consumption becomes. The response time is the most important factor in estimating the performance of a proxy server. It is expressed in terms of the round trip time (RTT) between a user and a proxy server and the current amount of load $l_i(t)$ in the server. The round trip time is determined by the distance and the bandwidth between a user and a server. Sometimes, the round trip time gets bigger due to the congestion of a network. Here we have assumed RTT to be a constant. Another important factor which influences the response time is amount of load in a server. That is be-cause: if the current load is larger, each request needs to be kept in the queue of the server; hence each request takes longer time for each request to be proceeded in the server. Based on the above discussion, the response time Ri(t) at time t can be modelled as follows:

$$R_i(t) = g_i(\, l_i(\, t\,)\,) + RTT \qquad (2)$$

where g_i represents the function to show the relation between load $l_i(t)$ and response time $R_i(t)$ at time t. Energy consumption can be generally defined as: Energy = AvgPower × Time, where Energy and AvgPower are measured in Joule and Watt, respectively, and 1 Joule = 1Watt × 1 Second. Energy efficiency is equivalent to the ratio of performance, measured as the rate of work done, to the power used and the performance can be represented by response time or throughput of the computing system.

Energy Efficiency = Work done / Energy = Work done / Power × Time = Performance / Power (3)

The main approach towards energy efficiency is efficient power management. According to equation (3), there are two ways to enhance energy efficient computing: either improving the performance with the same power, or reducing power consumption without sacrificing too much performance. For energy efficient systems, while maximal performance for some tasks (or the whole workload) is still desirable in some cases, the systems must also ensure the energy usage is minimized. Preferably, a computing system consumes the minimum amount of energy to perform a task at the maximal performance level. The relationship between

performance and energy efficiency is not mutually exclusive. A maximal performance could also be achieved by deactivating some resources or lowering certain individual performance without affecting the work-load's best possible completion time or throughput in order to optimize energy usage. Brown et al. [34] treated energy efficiency as an optimization problem. To minimize the total energy, an energy efficient system must adjust the system's hardware resources dynamically, so that only what is needed to execute tasks is made available. Rivoire et al. [42] pointed out two major complementary ways to solve the energy efficiency problem: either building energy efficiency into the initial design of computer components and systems, or adaptively managing the power consumption of systems or groups of systems in response to changing conditions related to the workload or environment.

IV. Design and work functionality of Proxy server based Energy Efficiency module

A. Proxy server based Energy Efficiency Module

In this paper, we propose an energy efficient design module to be deployed at proxy server as shown in Fig. 2. In this energy efficient module we have proposed a combined approach consisting of two interdependent techniques: Dynamic feedback driven provisioning that dynamically reconfigure the proxy server parameters by taking feedback from proxy workload logs, analysing them and also load balancing by incorporating Sleep proxy mechanism into a server cluster that optimally distributes current load among the awaked running servers.

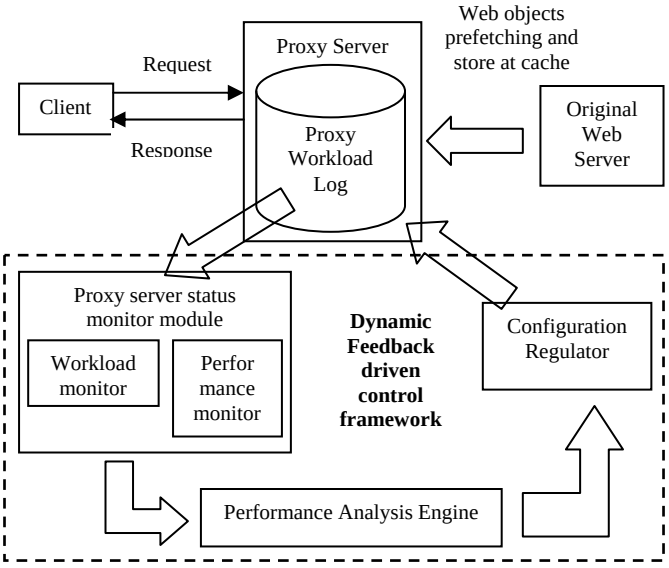


Fig. 2 Architecture of Proxy server based Energy Efficient module design

The proxy server continuously monitors the response time of individual requests as measured by the difference between the time the request was received at the server and the time the first pack-et of the response was sent to the client. We have deployed a feedback driven control framework at proxy server based on performance analysis of the server and adaptively adjusts policy parameters to increase energy savings when measured response times are lower (better) than the response time goal (a threshold value), or to decrease energy savings when system is not meeting the specified threshold value. In this framework, the Proxy Server Status Monitor Module periodically monitors and measures power and performance characteristics of Proxy workload based on the current server configuration. It performs the statistical analysis and feedback those results to Performance Analysis Engine. Server Status Monitor Module consists of Workload Monitor and Performance Monitor. Workload Monitor collects information about received service re-quests, including request numbers, request types and average service time of accessing a client, and so on. Performance Monitor collects various performance data such as utilization of CPU and Memory. Also, Server Status Monitor Module monitors if processes works properly. Performance Analysis Engine receives workload data and performance data from Server Status Monitor Module, calculates the value of load balancing under different configurations, evaluate deviation between actual and expected load value, predicts the proxy server power-performance budget for the next observation interval based on the statistical analysis as well as history-based knowledge and finally determine whether to change proxy configuration. Configuration Regulator adaptively adjusts and reconfigures Proxy server power-performance parameters towards optimal settings such that the system can meet the power budget determined by the Analysis engine. In our proposed energy efficient strategy we have combined Sleep proxy mechanism with dynamic feedback driven control framework in a cluster of proxy servers and the associated serving clients.

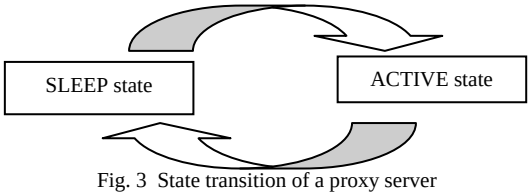
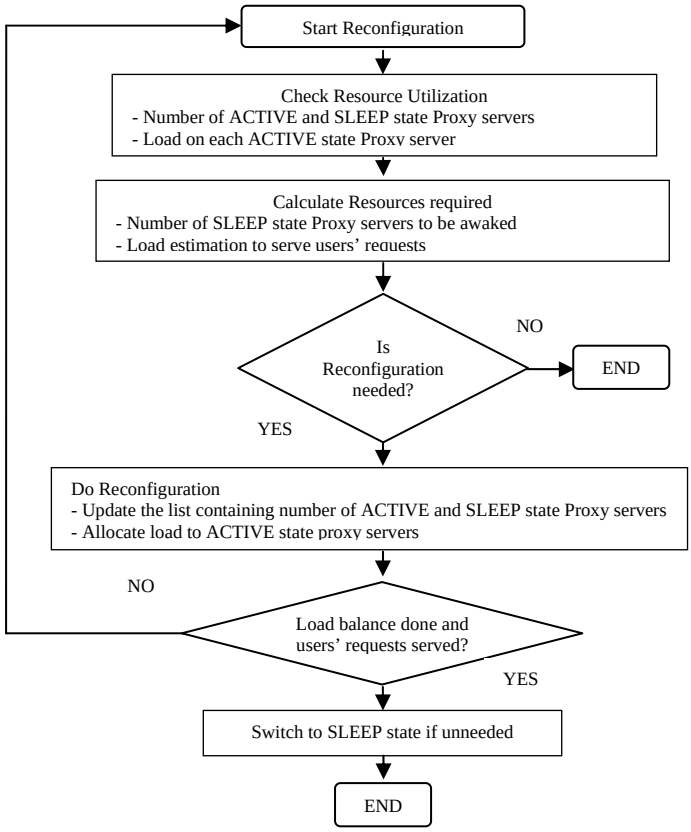


Fig. 3 State transition of a proxy server

We assume each proxy server has two states: SLEEP and ACTIVE. In SLEEP state there is no request from users to the proxy server and in the ACTIVE state the proxy is providing service for the users. Therefore, in general, the consumed energy in SLEEP state is much less than that in ACTIVE state. Fig. 3 depicts a proxy server state transition diagram between SLEEP state and ACTIVE state. Each client in the network will login in the proxy server and provide necessary information that the proxy server will need in order to maintain load balancing of the entire network. This will be done by periodically sending an INFO message that contains these data: node-id (client-id), IP address, MAC address, Operating System, state, local time and amount of data transferred. Proxy servers by listening the INFO messages from clients and store these data in tables to keep track and information about active nodes in the network. The network load will be calculated and the threshold value must be set depending on serving number of clients and amount of data to be exchanged. In the cluster of proxy servers each server will be either in ACTIVE or SLEEP state depending upon the threshold value set for the whole network the cluster is serving and their status will be maintained in a queue. Also, each proxy

server will check periodically (preferably twice the time set for periodic sending of INFO) whether the clients it was serving are active or not. Proxy servers change status from SLEEP to ACTIVE according as maintained in queue whenever the threshold value or load limit exceeds. Optionally we can configure the default time for periodic sending of INFO, IP address of server and communication port with the server. Also we can set the number of server limits at a specific time of the day or night whenever the load is almost average and the clients of the network can be optimally served with that minimum number of servers. Our proposed techniques are applicable to individual proxy server systems and complement energy management policies for proxy servers' cluster. The following flowchart describes the reconfiguration module:



B. Optimized Load Allocation algorithm

Based on our proposed framework we introduce the Proxy server Optimized Load Allocation algorithm to allocate the incoming client requests to a collection (cluster) of proxy servers. The main idea of our proposed algorithm is as follows: there are two states ACTIVE and SLEEP of servers in the cluster as we mentioned before.

- Initially, all the servers are into SLEEP state.
- If there is a request from users, the router first wakes up one SLEEP state server, and then allocates the request to the server.
- After that, if a new request arrives at the router, the router allocates the request to an ACTIVE state server whose number of requests is fewer than the trade off point, i.e., li < $l_i^{balance}$, where li is the current load and $l_i^{balance}$ is the maximum load of each server i.
- If all the ACTIVE state servers' loads are equal to their balance point, a router wakes up a SLEEP state server and allocates such a request to the new waked-up servers. Simultaneously the server's state changes from SLEEP state to ACTIVE state.
- Now, there are more than or equal to one sever is in ACTIVE state. Thus, on receipt of a re-quest, a router first checks the loads of every ACTIVE state server. If there is at least one ACTIVE state server whose load is lighter than the balance load, then the request is allocated to one of that ACTIVE state server.
- If the load of every ACTIVE state server is heavier than the balance point $l_i^{balance,}$ a router is regarded to wake up a new SLEEP state server and allocate the request to that newly awaked server.
- An ACTIVE state server after finishing execution of a request, if there is no request from users being executed in a server; the server changes the state to SLEEP state.
- Finally, by using this load allocation algorithm, the entire network is efficiently served by the cluster of energy efficient proxy servers.

V. EXPERIMENTAL SETUP AND RESULT DISCUSSION

The Web workload to study our proposed combined energy efficient proxy designing is obtained from running proxy server of Birla Institute of Technology (BIT), Mesra, Ranchi, Jharkhand which is extremely popular among students, faculty members and staffs of as many twenty five departments along with various administrative sections, hostels and quarters. In our experiment the proxy traces refer to the period from 13/Feb/2012:06:45:04 to 19/Feb/2012:00:00:02 of one week. The trace is composed of 9,043 nodes and 1,165,845 Web requests with average of 1,700 users per day. Table I summarizes the characteristics of the proxy trace workload. The simulations were performed at different network loads. Also Fig. 4 and Fig. 5 demonstrate the Proxy server request rate observed of a busy working day.

TABLE I
CHARACTERISTICS OF PROXY SERVER WORKLOAD

Workload	Proxy trace
Avg requests / sec	15
Peak requests / sec	30
Avg requests / conn	45
Files	658,232
Total file size	7,856 MB
Requests	9,290,196
Total response size	24,172MB

The experimental environment is composed of six client computers and one server acting as proxy. Six client computers have these configurations: processor Intel® Pentium Dual Core CPU 2.0 GHz, RAM 1 GB, HDD 160 GB. Four of them having OS 32-bit Windows 7 and TFT monitors whereas remaining two having OS 32-bit Windows XP and CRT monitors. The system specification of the typical ACPI-compliant proxy server is with processor Intel® Core2Duo 3 GHz, 8 GB RAM, 500GB hard disk, and single Gigabit Ethernet card installed. The Proxy server runs Windows Server 2008 R2 OS and it is the IIS 7.5 Web server system.

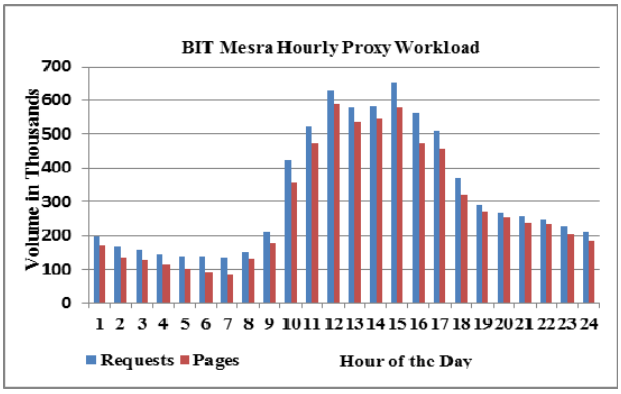


Fig. 4 BIT, Mesra daily Web proxy workload

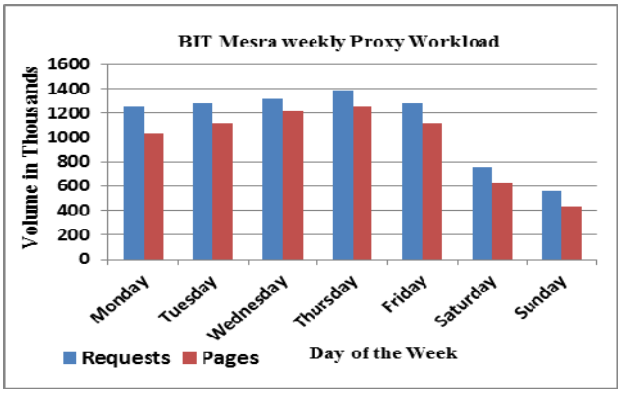


Fig. 5 BIT, Mesra weekly Web proxy workload

We have performed two sets of experimental testing one with implementing our combined sleep proxy and optimal load allocation strategy and another without the strategy. We have measured the consumed energy and the network traffic (number of packets) passing to them during the Institution working hours 08:00 to 16:00 for three continuous days with the above mentioned Web proxy workload and each client is connected with the proxy server through the Institution's proxy client application "Cyberoam Captive Portal". The first set of testing, Test Set 1 (TS1), without combined strategy and the second set of testing, Test Set 2 (TS2) when our combined strategy is implemented on the experimental network setup. Table II shows results regarding the power consumption and total number of packets for the two test sets TS1 and TS2. As we can see from the Table II, Computer C1 and C6 have higher power consumption and this happens because of the CRT screen that consumes more power.

TABLE II
3-DAY AVERAGE POWER CONSUMPTION AND TOTAL NUMBER OF PACKETS

	TS1 (without combined strategy)		TS2 (with combined strategy)	
	Power Consumption (W)	Total number of packets	Power Consumption (W)	Total number of packets
C1	133.7	208734	132.8	107337
C2	116.7	53440	117.4	49413
C3	114.15	138331	115.17	71492
C4	113.46	172610	115.5	72880
C5	117.9	195652	118.16	97320
C6	135.1	288957	134.16	122369

Fig. 6 shows the power consumption vs. CPU utilization with and without implementing the combined strategy while changing the CPU utilization by using variable workloads, where the horizontal axis indicates the average CPU utilization reported by the OS, and the vertical axis indicates the average power consumption measured at the proxy server.

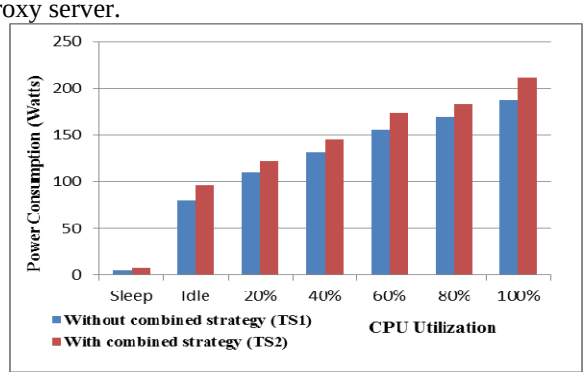


Fig. 6 Power consumption vs. CPU utilization

We have observed two important facts. First, the power consumption increases almost linearly with CPU utilization, as reported in other studies. Second, an idle server consumes up to 66% of the peak power, because even when a server is not loaded with user tasks, the power needed to run the OS and to maintain hardware peripherals, such as memory, disks, master board, PCI slots, and fans, is not negligible. Fig. 5 also implies that if we allow user connections for login requests to a limited number of proxy servers, and keep the rest of the servers hibernating, we can achieve significant power savings. However, the consolidation of login requests results in high utilization of those servers, which may downgrade performance and user experiences. Our measure leads to the following observations: although when in the idle state a proxy server can have null performance, it has inherent power consumed by some resources to maintain the basic routines of the system. Moreover, the power consumption shows only slight variations when in the idle state and the average power consumption is approximately constant. According to our observation, the power consumption, while in the busy state, covers a wider range of value, followed by the utilization of the servers. A power consumption histogram, or probability distribution function, serves as one useful measure of the power consumption variation with time. Examples of probability distribution functions for the three utilizations as measured above for utilization = 0.15, 0.5 and 0.85 appeared in Fig. 7.

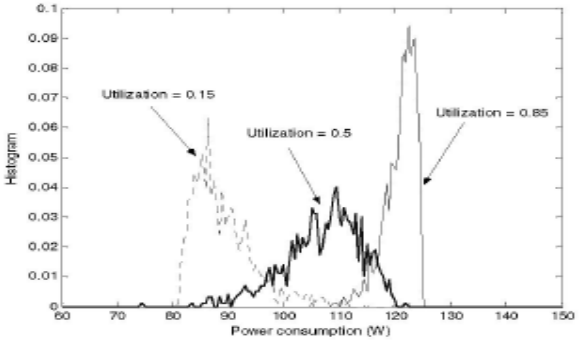


Fig. 7 Histogram of power consumption of a proxy server

Obviously, the different utilization results in a different distribution of the power dissipation. The larger the utilization, the more likely the mean is to take on larger values and the variance is to take on smaller values.

TABLE III
TS1 PERFORMANCE, AVERAGE POWER AND POWER EFFICIENCY

	Without combined strategy (TS1)		
Load Level	**Performance (Responses/ Second)**	**Avg. Power (Watts)**	**Power Efficiency (Performance/ Watts)**
100%	21,959	324	68
90%	19,752	318	62
80%	17,552	312	56
70%	15,360	292	53
60%	13,163	277	48
50%	10,969	274	40
40%	8,777	271	32
30%	6,584	268	25
20%	4,390	264	17
10%	2,196	261	8
0%	0	256	0

Table III and IV depict the Performance, Average Power and Power Efficiency at various load levels of our experimental setup with (TS1) and without (TS2) implementing the proposed combined energy efficient strategy whereas Table V shows the comparison result and average improvements. Also Fig. 8-10 shows the comparisons of TS1 and TS2 against three performance measures.

TABLE IV
TS2 PERFORMANCE, AVERAGE POWER AND POWER EFFICIENCY

	With combined strategy (TS2)		
Load Level	**Performance (Responses/ Second)**	**Avg. Power (Watts)**	**Power Efficiency (Performance/ Watts)**
100%	69,978	264	110
90%	62,918	253	96
80%	55,910	236	87
70%	48,921	221	73
60%	41,922	211	65
50%	34,923	205	52
40%	27,936	197	45
30%	20,935	188	37
20%	13,962	182	28
10%	6,985	175	17
0%	0	119	0

TABLE V
TS1 AND TS2 POWER CONSUMPTION AND POWER EFFICIENCY COMPARISON

Load Level	**Difference in Power Consumed**	**Difference in Power Efficiency**
100%	18%	42%
90%	20%	34%
80%	24%	31%
70%	24%	20%
60%	24%	17%
50%	25%	12%
40%	27%	13%
30%	30%	12%
20%	31%	11%
10%	33%	9%
0%	54%	-
Averages:	**28%**	**18%**

Over the 10 target load levels tested, the experimental setup with combined energy efficient strategy (TS2) an average of 28% less power consumed than without implementing the combined strategy (TS1). In addition, for TS2 the performance-to-power ratio is, on average, 18% higher than TS1. Also, TS2 consumed less power than TS1 across all target loads. At 10% target load, it consumed 33% less power than the TS1; at 50% target load, the TS2 used 25% less power than the TS1.

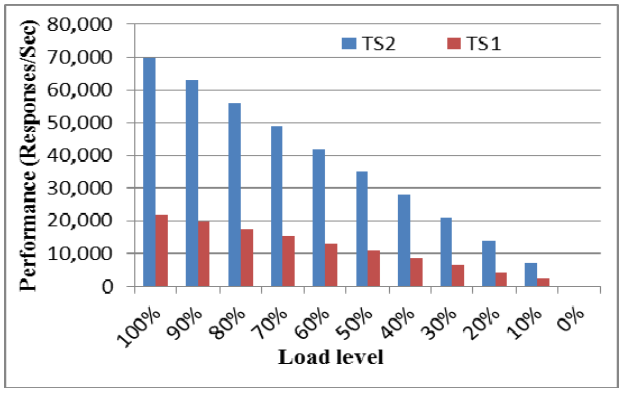


Fig. 8 Performance (Responses/Sec) comparison

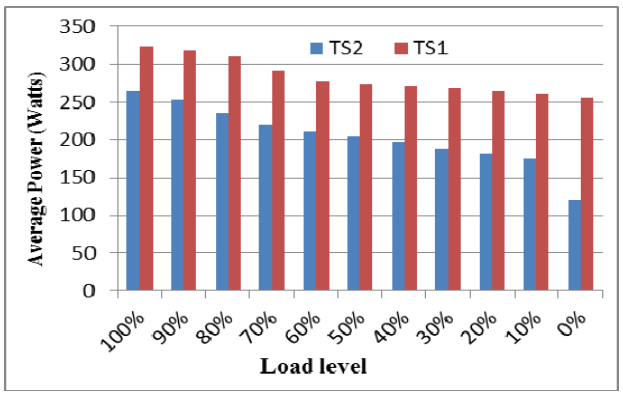


Fig. 9 Average Power (Watts) comparison

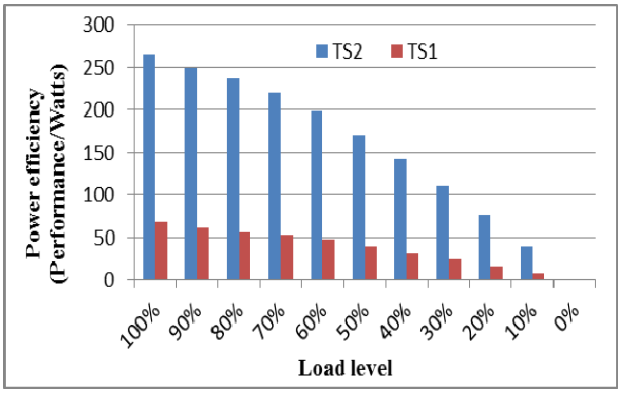


Fig. 10 Performance efficiency (Performance/Watts) comparison

VI. Conclusions

This paper presents a power aware proxy server design model using combined energy conservation strategy. The principal objective is to incorporate efficient power management into our Hybrid Prefetch-Cache system model. This objective enables always active proxy servers to improve its energy efficiency under fluctuating loads, and to dynamically match both load and power consumption. The combined strategies of Dynamic feedback driven provisioning and Sleep proxy mechanism in proxy server presented in this paper can adaptively reconfigure proxy performance parameters by efficiently measuring existing client loads, do optimal client serving load allocation among the running (ACTIVE state) proxies and also provide energy efficiency by sending proxies into SLEEP state. Moreover to evaluate the performance of the proposed scheme from the view point of energy consumption, we implemented a small experimental client-proxy server setup with one proxy server and six client systems. The experimental result shows that the proposed scheme could reduce power consumption by 28% and improve power efficiency by 18%. Our combined energy efficient strategy along with optimal load allocation algorithm is relatively simple yet still manages to save a significant amount of energy. In the future, we plan on looking at more sophisticated power and energy prediction model which will allow the intelligent proxy server to handle a greater variety of workloads. Furthermore, proxy based power efficient workload management and simulation tool buildup are also our next-step work.

Acknowledgment

We wish to acknowledge the network support staffs of our University Birla Institute of Technology, Mesra for supplying valuable proxy server workload traces and helping us to establish the experimental proxy servers' cluster network setup.

References

[1] http://www.internetworldstats.com/stats.htm
[2] http://www.worldwideWebsize.com/
[3] S.Sulaiman, S.M.Shamsuddin, A. Abraham, S. Sulaiman, *"Web Caching and Prefetching: What, Why, and How?"*, IEEE, 2008.
[4] T. M. Kroeger, D. D. Long, and J. C. Mogul, *"Exploring the bounds of Web latency reduction from caching and prefetching"* in Procc. of the 1st USENIX Symp. on Internet Technologies and Systems, Monterey, USA, 1997.
[5] S.S.Ghosh and A.Jain, *"Hybrid Cache Replacement Policy for Proxy Server"*, International Journal of Advanced Research in Computer and Communication Engineering (IJARCCE), Vol. 2, Issue 3, pp. 1527 - 1532, 2013.
[6] S.S.Ghosh, V.Kumar and A.Jain, *"A Novel Hybrid Policy for Web Caching"*, International Journal of Engineering Research and Development (IJERD), Volume 6, Issue 11, pp. 15 - 22., 2013.
[7] R. Brown, E. Masanet, B. Nordman, B. Tschudi, A. Shehabi, J. Stanley, J. Koomey, D. Sartor, P. Chan, J. Loper, et al. Report to congress on server and data center energy efficiency. Public law, pages 109–431, 2007.
[8] S.S.Ghosh and A.Jain, *"An Energy Efficient Hybrid Proxy Cache-Prefetch Model for Web Optimization"*, International Conference on Eco-friendly Computing and Communication Systems (ICECCS 2013), 2013.
[9] J. Baliga, K. Hinton, and R. Tucker, *"Energy consumption of the internet"* in Joint International Conference on Optical Internet, 2007 and the 2007 32nd Australian Conf. on Optical Fibre Technology., june 2007, pp. 1 –3.
[10] A. Bianzino, A. Raju, and D. Rossi, *"Greening the internet: Measuring Web power consumption"* IT Professional, vol. 13, no. 1, pp. 48 –53, jan.-feb. 2011.
[11] P.Bohrer, E.N.Elnozahy, T.Keller, M.Kistler, C.Lefurgy, C.McDowell, and R.Rajamony, *"The Case for Power Management in Web Servers"*, Power Aware Computing, Klewer Academic Publishers, 2002.
[12] L.Mastroleon, N.Bambos, C.Kozyrakis and D.Economou, *"Autonomic Power Management Schemes for Internet Servers and Data Centers"*, In Proc. GLOBECOM, November 2005.
[13] M.Elnozahy, M.Kistler and R.Rajamony, *"Energy Conservation Policies for Web Servers"*. In Proc. 4th USENIX Symposium on Internet Technologies and Systems, March 2003.
[14] E.N.(Mootaz)Elnozahy, M.Kistler and R.Rajamony *"Energy-Efficient Server Clusters"*. In Workshop on Mobile Computing Systems and Applications, February 2002.
[15] V.Sharma, A.Thomas, T.Abdelzaher, K.Skadron, Z.Lu, *"Power-aware QoS Management in Web Servers"*, Proceedings of the 24th IEEE International Real-Time Systems Symposium, p.63, December 03-05, 2003.
[16] T.Abdelzaher and V.Sharma. *"A synthetic utilization bound for a-periodic tasks with resource requirements"*. In Euromicro Conference on Real Time Systems, Porto, Portugal, July 2003.
[17] C. Lefurgy et al., *"Server-Level Power Control"* ICAC, p. 4, In Proc. ICAC, 2007.
[18] J. S. Chase et al., *"Managing energy and server resources in hosting centers"*, In Proc. the eighteenth ACM symposium on Operating systems principles, October 21-24, 2001, Banff, Alberta, Canada.
[19] E.Pinheiro, R.Bianchini, E.V.Carrera and T.Heath, *"Load Balancing and Unbalancing for Power and Performance in Cluster-Based Systems"* In Proc. the Workshop on Compilers and Operating Systems for Low Power, September 2001; Technical Report DCS-TR-440, Department of Computer Science, Rutgers University, New Brunswick, NJ, May 2001.
[20] M. Weiser, B.Welch, A. J. Demers, and S. Shenker. *"Scheduling for reduced cpu energy"*. In OSDI, pages 13–23, 1994.
[21] D. Grunwald, P. Levis, K. I. Farkas, C. B. M. III, and M. Neufeld. *"Policies for dynamic clock scheduling"*. In OSDI, pages 73–86, 2000.
[22] J.R.Lorch, A.J.Smith, *"Improving Dynamic Voltage Scaling Algorithms with PACE"*, In Proc. SIGMETRICS, pp50-61, ACM 2001.
[23] D. Kusic, J. O. Kephart, J. E. Hanson, N. Kandasamy, and G. Jiang. *"Power and performance management of virtualized computing environments via lookahead control"*. In ICAC '08:, 2008.
[24] A. Kansal, F. Zhao, J. Liu, N. Kothari, and A. A. Bhattacharya, *"Virtual machine power metering and provisioning"* in In 1st ACM symposium on Cloud computing, ser. SoCC '10. New York, NY, USA: ACM, 2010, pp. 39–50.
[25] P. Bodik, R. Griffith, C. Sutton, A. Fox, M. Jordan, and D. Patterson. *"Statistical machine learning makes automatic control practical for internet datacenters"*. HotCloud'09, 2009.
[26] W.Felter, K.Rajamani, T.Keller, C.Rusu, *"A Performance-Conserving Approach for Reducing Peak Power Consumption in Server Systems"*, In Proc. ICS, Cambridge, MA, June 2005.

[27] G. Contreras and M. Martonosi, *"Power prediction for Intel XScale® processors using performance monitoring unit events"* in Int. symposium on Low power electronics and design 2005. ACM, 2005, pp. 221–226.
[28] http://co2now.org/
[29] http://www.gartner.com/it/page. jsp?id=503867
[30] J. Chen and C. Kuo. *"Energy-efficient scheduling for real-time systems on dynamic voltage scaling (DVS) platforms"*. In RTCSA, pages 28–38. IEEE, 2007.
[31] M. Poess and R. Nambiar. *"Energy cost, the key challenge of today's data centers: a power consumption analysis of TPC-C results"* Proceedings of the VLDB Endowment, 1(2):1229–1240, 2008.
[32] HP Blade System c-Class portfolio.http://hl8004.wwwl.hp.com/products/blades/components/c-class-components.html.
[33] IDC White Paper. Forecasting Total Cost of Ownership for Initial Deployments of Server Blades. ftp://hp.pl/pub/c-products/blades/idc-tco-deployment.pdf, 2006.
[34] D. Brown and C. Reams. Toward energy-efficient computing. Communications of the ACM, 53(3):50–58, 2010.
[35] L. Barroso and U. Holzle. *"The case for energy-proportional computing"*. IEEE Computer, 40(12):33, 2007.
[36] D. Tsirogiannis, S. Harizopoulos, and M. Shah. *"Analyzing the energy efficiency of a database server"*. In SIGMOD, pages 231–242. ACM, 2010.
[37] http://googleblog.blogspot.com/2009/01/powering-google-search.html
[38] T. Horvath, T. Abdelzaher, K. Skadron and X. Liu, *"Dynamic voltage scaling in multi-tier Web servers with end-to-end delay control"*, IEEE Transaction on Computers, Volume 56, Issue 4, Pages 444-458, April 2007.
[39] Kenneth J. Christensen, Franklin 'Bo' Gulledge – *"Enabling Power Management for Network-attached Computers"*, International Journal of Network Management, Vol. 8, Nr.2, pp. 120-130, 1998.
[40] M. Allman, K. Christensen, B. Nordman, dhe V. Paxson, *"Enabling an Energy-Efficient Future Internet Through Selectively Connected End Systems"* Sixth Workshop on Hot Topics in Networks (HotNets-VI), November 2007.
[41] Yuvraj Agarwal, Steve Hodges, James Scott, Ranveer Chandra, Paramvir Bahl, Rajesh Gupta *"Somniloquy: Maintaining Network Connectivity While Your Computer Sleeps"*, 2008.
[42] P. Ranganathan, S. Rivoire, and J. Moore. *"Models and metrics for energy-efficient computing"*. Advances in Computers, 75:159–233, 2009.