ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Vol 5, Issue 3, March 2018

Predictive Energy Saving In Search Engine Using Query Processing

[1] Brindha T, [2] Anbarasi G, [3]Anitha G, [4]Abirami S

[1][2][3][4] Department of Computer Science and Engineering,V.S.B Engineering College, Karur, Tamilnadu, INDIA.

***Abstract* – Filtering chump seek after effect Application Custom Seek Engine (CSE), you can actualize affluent seek adventures that accomplish it easier for visitors for the acquisition the advice they're searching for on your site. Today we're announcing two improvements to the allocation and clarification of seek after-effects in CSE. We adduce a new Web clarification adjustment based on argument classification. We use samples of for bidden Web pages to characterize the chic of Web page and accomplish the concern processing for activity efficient. Web seek engines are composed by bags of concern processing nodes, i.e., servers committed to action user queries. Such abounding servers absorb a cogent bulk of energy, mostly answerable to their CPUs, but they are all important to ensure low latencies, back users apprehend sub-second acknowledgment times (e.g., 500 ms). However, users can hardly apprehension the acknowledgment times that are faster than their expectations. Hence, we adduce the Predictive Activity Saving Online Scheduling Algorithm (PESOS) to baldest a lot of adapted CPU abundance to action a concern on a per-core basis. PESOS aim at queries by their deadlines, and advantage high-level scheduling advice to abate the CPU activity burning of a concern processing node. As predictors can be inaccurate, in this plan we as well as adduce and investigate a way to atone anticipation errors application the basis beggarly aboveboard absurdity of the predictors.**

1. INTRODUCTION

Web search engines continuously crawl and index an immense number of Web pages to return fresh and relevant results to the users' queries. Users' queries are processed by query processing nodes, i.e., physical servers are dedicated to this task. Web search engines [1] are typically composed by thousands of these nodes, hosted in large data centre which also include infrastructures for telecommunication, thermal cooling, fire suppression, power supply, etc. This complex infrastructure is necessary to have low tail latencies (e.g., 95th percentile) to guarantee that most users will receive results in sub-second times (e.g., 500 ms), in line with their expectations. At the same time, such many servers consume a significant amount of energy, hindering the profitability of the search engines and raising environmental concerns. In fact, data centre consume tons of megawatts of electric power and the related expenditure can exceed the original investment cost for a data centre. Because of their energy consumption, data centres [4] are responsible for the 14% of the ICT sector carbon dioxide emissions, which are the main cause of global warming. For this reason, governments are promoting codes of conduct and best practices to reduce the environmental impact of data centres. Since energy consumption has an important role on the profitability and environmental impact of Web search engines, improving their energy efficiency is an important aspect. Noticeably, users can hardly notice response times that are faster than their expectations. Therefore, to reduce energy consumption, Web search engines should answer queries no faster than user expectations.

The standards used in these technologies [10] will increase the bandwidth, but it is not enough to meet the demand for wireless data. If the access points are increased then only we can overcome this problem. For example, while maintaining full coverage if the access point range is minimum then it will be difficult to provide high efficiency.

II.RELATED WORK

In the past, a large part of data centre energy consumption was accounted to inefficiencies in its cooling and power supply systems. However, [2] report that modern data centres have largely reduced the energy wastage of those infrastructures, leaving little room for further improvement. On the contrary, opportunities exist to reduce the energy consumption of the servers hosted in a data centre. In particular, our work focuses on the CPU power management of query processing nodes, since the CPUs dominate the energy consumption of physical servers dedicated to search tasks. In fact, [17] CPUs can use up to 66% of the whole energy consumed by a query processing node at peak utilization. Modern CPUs usually expose two energy saving mechanism, namely C-states and P-states. C-states represent CPU cores idle states [13] and they are typically managed by the operating system. C0 is the operative state in which a CPU core can perform computing tasks.

Query processing

Web search engines continuously crawl a large amount of Web pages. This collection of documents is then indexed to produce an inverted index. The inverted index is a data structure that maps each term in the document collection to a posting list, i.e., a list of postings which indicates the occurrence of a term in a document. A posting contains at least the identifier (i.e., a natural number) of the document where the term appears and its term frequency, i.e., the number of occurrences of the term in that particular document.

Existing System

Cloud users access the software and utilizes the services [7] by cloning tasks onto multiple virtual machines (VMs).The high-performance computing capacity reminds cloud providers to utilize resources fully due to limitation of resource. Enhancing resource utilization is also essential for achieving cost effectiveness. The SaaS applications support various services such as email, FTP, and multimedia services with different delay requirement. Among these applications multimedia service has relatively low delay requirements.

Disadvantages

• Only using limitation of resource.
• High cost.
• Delay Occur.

III.PROPOSED SYSTEM

A scheduling algorithm is to enhance both deadline guarantee and resource utilization. It modified the conservative backfilling algorithm by utilizing the earliest deadline first (EDF) [5] algorithm and the largest weight first (LWF) algorithm. The proposed algorithm first score all the jobs arrived at the data centre (DC) and sort the jobs in ascending order to serve high priority job first. The proposed algorithm then selects the largest possible backfill job as guaranteeing deadline.
It improves both resource utilization in the cloud DC and deadline guarantee of the all the jobs. The EDF algorithm is used when the first searched backfill job misses its deadline without backfilling to guarantee deadline.

PREDICTIVE ENEGRY SAVING ONLINE SCHEDULING (PESOS) ALGORITHM

A novel Predictive Energy Saving Online Scheduling (PESOS) [9] algorithm is implemented. In the context of Web search engines, PESOS aims to reduce the CPU energy consumption of a query processing node while imposing required tail latency on the query response times.

BLOCK DIAGRAM OF QUERY PROCESSING WEB SEARCH ENGINE

For each query, PESOS [13],[14] selects the lowest possible CPU core frequency such that the energy consumption is reduced and the deadlines are respected. PESOS select the right CPU core frequency exploiting two different kinds of query efficiency predictors (QEPs). The first QEP estimates the processing volume of queries. The second QEP estimates the query processing times under different core frequencies, given the number of postings to score. Since QEPs can be inaccurate, during their training we recorded the root mean square error (RMSE)[8] of the predictions. In this work, we are proposing to sum the RMSE to the actual predictions to compensate prediction errors. We then defined two possible configurations for PESOS: time conservative, where prediction correction is enforced, and energy conservative, where QEPs are left unmodified.

IV.EXPERIMENTAL ANALYSIS

We experimentally evaluated the achievement of PESOS using the ClueWeb09B bulk [4] and processing queries from the MSN2006 log applying two altered activating pruning retrieval strategies: MaxScore and WAND. We compared the performance of PESOS with those of three baselines: perf, which consistently uses the best CPU amount frequency, power, which throttles frequencies according to the amount utilizations, and cons, which throttles frequencies according to the utilization of the concern servers. We begin that time bourgeois PESOS was able to accommodate appropriate appendage cessation of 500 and 1,000 ms for the aforementioned workload acceptable by perf. At the same time, time bourgeois PESOS was able to abate the CPU energy burning of the CPU by 12% with WAND up to 25% with MaxScore, for which we could alternatively added accurate query ability predictors than for WAND. Greater energy savings were appreciable with activity bourgeois PESOS, but at the amount of college latencies. Predictor's alteration is hence necessary to access the appropriate appendage latency, still providing significant activity savings. Moreover, we candy a realistic query workload which reflects the concern arrivals of one day of the MSN2006 log. We begin that time bourgeois PESOS was able to accommodated a 500 ms (with actual few violations) and a1,000 ms appendage cessation requirements, while abbreviation the CPU energy consumption, respectively, by 24% and by 44% when compared to perf. From the aforementioned set of experiments, we reported that ability can abate the CPU

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 5, Issue 3, March 2018

activity consumption by just 4% with account to perf. On the added hand, cons were able to abate the CPU activity burning by 27% but incurring in ample cessation violations. We justified the superior perf provided by PESOS acknowledgment to the application level information exploited by our algorithm, such as the knowledge about the accompaniment of the concern queues and the query efficiency predictions.

Query efficiency predictors (QEPs) [14] are techniques that estimate the execution time of a query before it is actually processed. Knowing in advance, the execution time of queries permits to improve the performance of a search engine. Most QEPs exploit the characteristics of the query and the inverted index to pre-compute features to be exploited to estimate the query processing times. For instance, Macdonald et al. propose to use term-based features (e.g., the inverse document frequency of the term, its maximum relevance score among others) to predict the execution time of a query. They exploit their QEPs to implement on-line algorithms to schedule queries across processing node, in order to reduce the average query waiting and completion times. The works, instead, address the problem to whether parallelize or not the processing of a query. In fact, parallel processing can reduce the execution time of long-running queries but provides limited benefits when dealing with short-running ones. Both the works propose QEPs to detect long-running queries. The processing of the query is parallelized only if their QEPs detect the query as a long-running one. Rather than combining term-based features, propose the analytical model for the query processing stages and to use such model to predict the execution time of queries. In our work, we modify the QEPs described to develop our algorithm for reducing the energy consumption of a processing node while maintaining low tail latencies.

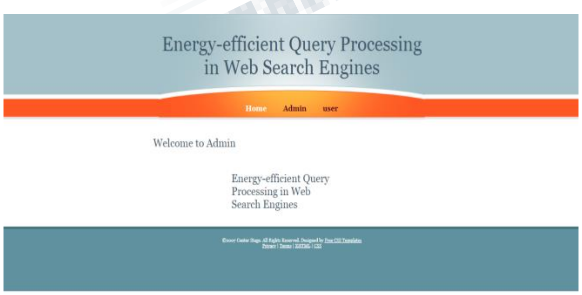

Fig1. Home page

Fig2. Login page

Fig3. Query results

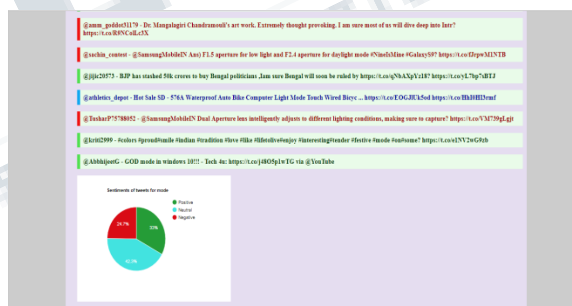

Fig4. Query results with chart

Fig4. Admin Page

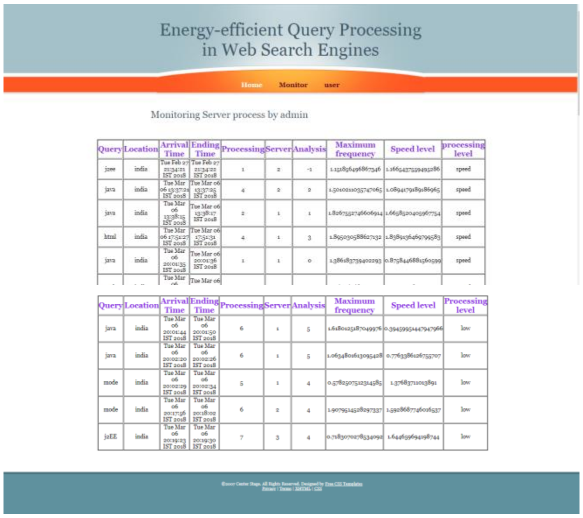

Fig5. Back end processing

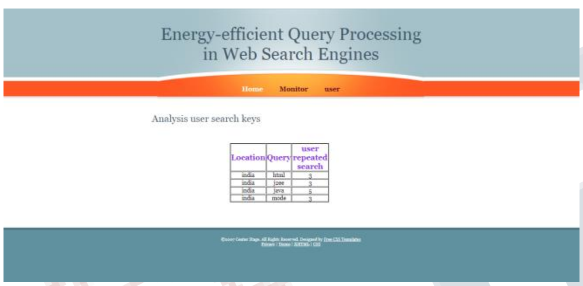

Fig6. Frequency of each queries

V.CONCLUSION AND FUTURE WORK

Web search engines can consume tons of megawatts of electric power to operate; there is only a limited body of research that aims to reduce the energy expenditure of Web search engines. These works can be divided in three categories which focus on different level of a Web search engine architecture: 1) geographically distributed data centres, 2) processing clusters within a data centre, and 3) a single query processing node. This work focuses on multi-site Web search engines, i.e., search engines composed by multiple and geographically distant data centres. These studies propose to use query forwarding, i.e., to shift the query workload between data centres. consider a scenario where data centres hold the same replica of the inverted index. They propose to use query forwarding to exploit the difference in energy price at different sites, due to the different data centre locations and time zones. In this way, they aim to minimize the energy expenditure of the search engine. At the same time, the approach ensures that the remote sites can process forwarded queries without exceeding their processing capacity. It extends this idea by forwarding queries towards data centres that can use renewable energy sources that are both environmentally friendly and economically convenient. Instead, consider a scenario where each site holds a different inverted index. In this approach, we use query forwarding to maximize the quality of search results, collecting relevant document from the different sites, while satisfying energy cost budget constraints. In future, Query forwarding techniques may be applied in conjunction with PESOS to deploy more energy-efficient architectures.

VI.REFERENCES

[1]. C. D. Manning, P. Raghavan, and H. Sch¨utze, Introduction to Information Retrieval. Cam-bridge University Press, 2008. [18] M. Catena, C. Macdonald, and I. Ounis, “On inverted index compression for search engine efficiency,” in Proc. ECIR, 2014, pp. 359–371. [19] J. Dean, “Challenges in building large-scale information retrieval systems: Invited talk,” in Proc. WSDM, 2009.

[2]. S. Robertson and H. Zaragoza, “The Probabilis-tic Relevance Framework: BM25 and Beyond,” Found. Trends Inf. Retr., vol. 3, no. 4, pp. 333–389, Apr. 2009.

[3]. A. Z. Broder, D. Carmel, M. Herscovici, A. Sof-fer, and J. Zien, “Efficient query evaluation us-ing a two-level retrieval process,” in Proc. CIKM, 2003, pp. 426–434.

[4]. H. Turtle and J. Flood, “Query evaluation: Strate¬gies and optimizations,” Inf. Process. Manage. vol. 31, no. 6, pp. 831–850, Nov. 1995. [23] H. Wu and H. Fang, “Analytical per-formance modeling for top-k query processing,” in Proc. CIKM, 2014, pp. 1619–1628.

[5]. A. Freire, C. Macdonald, N. Tonellotto, I. Ou¬nis, and F. Cacheda, “Hybrid query scheduling for a replicated search engine,” in Proc. ECIR, 2013, pp. 435–446

[6]. L. A. Barroso, J. Clidaras, and U. H¨olzle, the Datacenter as a Computer: An Introduction to the Design of Warehouse-Scale Machines, 2nd ed. Morgan & Claypool Publishers, 2013.

[7]. I. Arapakis, X. Bai, and B. B. Cambazoglu, "Impact of response latency on user behavior in web search," in Proc. SIGIR, 2014,pp. 103–112.

[8]. U.S. Department of Energy, "Quick start guide to increase data center energy efficiency," 2009. [Online]. Available:http://goo.gl/ovDP26

[9]. The Climate Group for the Global e-Sustainability Initiative, "Smart 2020: Enabling the low carbon econ¬omy in the information age," 2008. [Online]. Available: http://goo.gl/w5gMXa

[10]. European Commission - Joint Research Centre, "The European Code of Conduct for En-ergy Efficiency in Data Centre." [Online].Available: http://goo.gl/wmqYLQ

[11]. U.S. Department of Energy, "Best Practices Guide for Energy-Efficient Data Center Design." [Online]. Available: http://goo.gl/pikFFv

[12]. D. C. Snowdon, S. Ruocco, and G. Heiser, "Power Management and Dynamic Voltage Scaling: Myths and Facts," in Proc. Of Workshop on Power Aware Real-time Computing, 2005.

[13]. The Linux Kernel Archives, "Intel P-State driver." [Online].Available: https://goo.gl/w9JyBa

[14]. D. Brodowski, "CPU frequency and voltage scaling code in the Linux kernel." [Online]. Available: https://goo.gl/QSkft2

[15]. C. Macdonald, N. Tonellotto, and I. Ounis, "Learning to predict response times for online query scheduling," in Proc. SIGIR, 2012, pp. 621–630.

[16]. M. Jeon, S. Kim, S.-w. Hwang, Y. He, S. Elnikety, A. L. Cox, and S. Rixner, "Predictive paralleli¬zation: Taming tail latencies in web search," in Proc. SIGIR, 2014, pp. 253–262.

[17]. S. Kim, Y. He, S.-w. Hwang, S. Elnikety, and S. Choi, "Delayed dynamic-selective (dds) prediction for reducing extreme tail latency in web search," in Proc. WSDM, 2015, pp. 7–16.

[18]. M. Catena, C. Macdonald, and N. Tonellot-to, "Load-sensitive cpu power management for web search engines," in Proc. SIGIR, 2015, pp. 751–754.