**Energy-efficient Query Processing in Web Search Engines**

**ABSTRACT:**

Web search engines are composed by thousands of query processing nodes, i.e., servers dedicated to process user queries. Such many servers consume a significant amount of energy, mostly accountable to their CPUs, but they are necessary to ensure low latencies, since users expect sub-second response times (e.g., 500 ms). However, users can hardly notice response times that are faster than their expectations. Hence, we propose the Predictive Energy Saving Online Scheduling Algorithm (PESOS) to select the most appropriate CPU frequency to process a query on a per-core basis. PESOS aims at process queries by their deadlines, and leverage high-level scheduling information to reduce the CPU energy consumption of a query processing node. PESOS bases its decision on query efficiency predictors, estimating the processing volume and processing time of a query. We experimentally evaluate PESOS upon the TREC ClueWeb09B collection and the MSN2006 query log. Results show that PESOS can reduce the CPU energy consumption of a query processing node up to \_48% compared to a system running at maximum CPU core frequency. PESOS outperforms also the best state-of-the-art competitor with a \_20% energy saving, while the competitor requires a fine parameter tuning and it may incurs in uncontrollable latency violations.

**EXISTING SYSTEM:**

* Kayaaslan et al. consider a scenario where datacenters 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 datacenter locations and timezones. 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.
* Blanco et al. extend this idea by forwarding queries towards datacenters that can use *renewable energy sources* that are both environmentally friendly and economically convenient.
* Teymorian et al., instead, consider a scenario where each site hold a different inverted index. In their approach, the authors use query forwarding to maximize the quality of search results, collecting relevant document from the different sites, while satisfying energy cost budget constraints. Query forwarding techniques may be applied in conjunction with PESOS to deploy more energy-efficient architectures.

**DISADVANTAGES OF EXISTING SYSTEM:**

* The query processing node can consume more energy than necessary in providing query results faster than required, with no benefit for the users.
* Web search engines process a large and continuous stream of queries. As a result, query processing nodes are rarely inactive and experience particularly short idle times. Consequently, there are little opportunities to exploit deep C-states, reducing the energy savings provided by the C-states in a Web search engine system

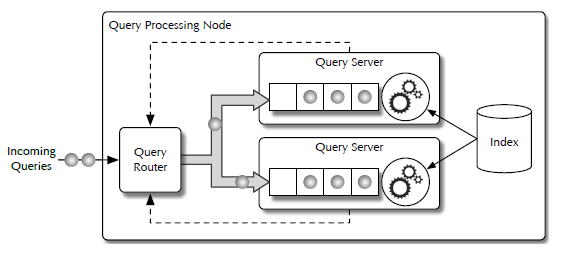
**PROPOSED SYSTEM:**

* In this work we propose the Predictive Energy Saving Online Scheduling algorithm (PESOS), which considers the tail latency requirement of queries as an explicit parameter. Via the DVFS technology, PESOS selects the most appropriate CPU frequency to process a query on a per-core basis, so that the CPU energy consumption is reduced while respecting a required tail latency.
* The algorithm bases its decision on *query efficiency predictors* rather than core utilization. Query efficiency predictors are techniques to estimate the processing time of a query before its processing. They have been proposed to improve the performance of a search engine, for instance to take decision about query scheduling or query processing parallelization. However, to the best of our knowledge, query efficiency predictor have not been considered for reducing the energy consumption of query processing nodes.
* PESOS exploits these two predictors to determine which is the lowest possible core frequency that can be used to process a query, so that the CPU energy consumption is reduced while satisfying the required tail latency. As predictors can be inaccurate, in this work we also propose and investigate a way to compensate prediction errors using the root mean square error of the predictors.

**ADVANTAGES OF PROPOSED SYSTEM:**

* We compare the performance of our approach with those of three baselines: perf, which always uses the maximum CPU core frequency, power, which throttles CPU core frequencies according to the core utilizations, and cons, which performs frequency throttling according to the query server utilization.
* PESOS, with predictors correction, is able to meet the tail latency requirements while reducing the CPU energy consumption with respect to perf and with respect to cons, which however incurs in uncontrollable latency violations. Moreover, the experiments show that energy consumption can be further reduced by PESOS when prediction correction is not used, but with higher tail latencies.

**SYSTEM ARCHITECTURE:**



**SYSTEM REQUIREMENTS:**

**HARDWARE REQUIREMENTS:**

* System : Pentium Dual Core.
* Hard Disk : 120 GB.
* Monitor : 15’’ LED
* Input Devices : Keyboard, Mouse
* Ram : 1 GB

**SOFTWARE REQUIREMENTS:**

* Operating system : Windows 7.
* Coding Language : JAVA/J2EE , .Net
* Tool : Netbeans 7.2.1
* Database : MYSQL