ENERGY SAVINGS SUSPICION ACTION IN WEB PORTAL

Pratheek Kumar[1], Anantha Murthy[2]
Asst Professor Dept of MCA, NMAMIT, Nitte
Department of Master of Computer Application
Visvesvaraya Technological University, Belagavi

Abstract
Web portals are composed by thousands of query processing nodes i.e., servers dedicated to process user queries. They may consume a large amount of energy that are more than usual to ensure low latencies, since users expect faster response time. However, users may experience faster response time than they expect. 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. Result shows that PESOS can reduce the energy consumption by CPU up to ~48% compare to a system that running on maximum CPU frequency. PESOS also confronts as the best competitor with the ~20% of power savings.
Keywords: Web Portals, response time, energy consumption, CPU frequency

I. Introduction

Web search engines continuously crawl and index an uncountable number of Web pages to return fresh and relevant results to the users queries. Users requests are processed by query processing nodes, i.e., physical servers dedicated to this task. Web search engines are typically composed by thousands of these nodes, hosted in large datacenters 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., 95-th percentile) to guarantee that most users will receive results in sub-second times (e.g., 500ms), in line with their expectations. In fact, data centers consume a significant amount of energy in order to process the users queries, which may exceed the expenditure of the actual investment cost of the datacenters. Also high emission of carbon-di-oxide will result in global warming. Because of this environment and high expenditure governments are promoting codes of conduct and best practices to reduce the those mentioned impacts of datacenters.

Since energy consumption is the main concern, improving the energy efficiency is the important aspect. To reduce the energy consumption web portals must answer the user requests less faster than the user's expectation. In this work we have to control the power consumption by the CPU's, as per we know CPU's are most energy consumption unit in the search system. To achieve the work Dynamic Frequency and Voltage Scaling(DVFS) technology can be exploited. DVFS technology changes the voltage and frequency of the CPU cores of the servers and slows the response in-order to reduce the energy consumption. DVFS scales the frequency according to the CPU's cores utilization. The core can consume more energy than required to carry out query processing faster than expected, with no benefits for the users.

In this work we propose Predictive Energy Saving Online Scheduling algorithm. This algorithm considers tail latency or queries as its explicit parameter. Over DVFS technology, PESOS algorithm selects most appropriate CPU frequency to process the query on a pre-core basis, while energy consumption is reduced but tail latency is considered. Algorithm is based on its query decision on query efficiency predictor rather than core utilization. Query predictors are those which predicts the time to process a query before it is processed. There are two query efficiency predictors are proposed. One to estimate the number of postings that must be scored to process a query and another one to estimate the response time of a query. PESOS make use of these two predictors to determine which is the lowest possible core frequency that can be used to process a query in-order to reduce the energy consumption by the CPU cores while satisfying required tail latency.

We experimentally evaluate PESOS upon the TREC ClueWeb09 corpus and the query stream from the MSN2006 query log. PESOS, with predictors correction, is able to meet the tail latency requirements while reducing the CPU energy consumption from ∼24% up to ∼44% with respect to performance and up to ∼20% with respect to frequency throttling. This experiment shows that PESOS can further reduce the energy consumption when prediction correction is not used but with much late response time.

II. Background

Here we discuss about the energy related issues in the web portals or search engines, how query processing works and Query efficiency predictors.

2.1 Energy consumption in web portals

Datacenters uses a massive amount of energy to process the user queries. This may result in lack of cooling and power supply system. There are many ways to reduce the energy consumption of the servers hosted in the datacenters. In particular, our work focus on the CPU power management of the query processing nodes. These physical part of the system is dedicated to the query processing task, that may use 65% of whole power plan at peak utilization.

Modern CPU's exposes the two type of energy saving mechanism namely C-state and P-state. C-state represents CPU cores idle state and controlled by the operating system.C0 is operative state, here it performs computing tasks. When no computing tasks CPU may under go deeper C-state. In web portals CPU is always in C0 state such that rarely it may go for idle state. A thread of queries may restrict the CPU to go for C-state reducing the energy saving provided by the C-states in web search engines.

While CPU is in C0 state it can operate at different frequencies. DVFS technology allow us to adjust the frequency and voltage. Higher frequency means faster computation but more energy consumption, where lower frequency means slower computation but less energy consumption. Various configuration of frequency and voltage are mapped to different P-state.

2.2 Query Processing

Web search engines scrolls a large amount of web pages. When a query is submitted to a web search engine, it is transferred to the query processing node. This node then retrieves a ranked list of documents those are relevant to query, and top K documents are related to user's query, sorted in decreasing order of relevance score. To generate the top K results processing node extensively traverses all documents related to the query. This is an exorbitant process since indexed document list can measure a tens of gigabytes so, Dynamic Pruning techniques are adopted. Such techniques avoid to evaluate irrelevant documents, skipping over portions of the posting lists. This decreases the response time. This helps to generate top K document which are most relevant to the user's query. In our work we have used this dynamic pruning technique for processing of queries.

2.3 Query Efficiency Predictors

Query Efficiency Predictors(QEP's) are techniques that estimates the execution time of the query before it is processed. Knowing the execution time can improve the performance of the search engines. Most of QEP's utilizes the features of the query and inverted index to pre-compute features to be used to estimate the execution time. In this work, problem is to parallelize the query processing or not. In fact, parallelizing the process will result in reducing the execution of the long running queries, but less benefit for the short running one. Both proposes the QEP's to find long running queries. The processing of the query is parallelized only if their QEPs detect the query as a long-running one. In our work, we modify he QEP's for reduce the energy consumption for a processing node.

III. Problem Formulation

In this section, introduce the operative scenario of the query processing node, formalizing the general minimum energy scheduling problem.

3.1 Operative Scenario

A query processing node is a physical server made by several CPU cores with a shared memory which holds inverted indexes. These indexes are divided into shards and distributed across other processing nodes. Here we focus on reducing power consumption of a single query processing node. The Query server process is always executed on top of CPU cores. All query processors access the shared index stored in the main memory (See figure 1). Each query servers are having the queues to store the incoming queries. When the CPU core is idle the first query is processed, in fact all queues follow the FCFS (First Come Firs Serve) policy. All queries will arrive processing node as stream S={Q1, Q2,….., Qn}. Once all queries reached the processing node are transmitted to the query server by query router. Query router will send query to the least loaded query server. Query process node may have a single query queue.

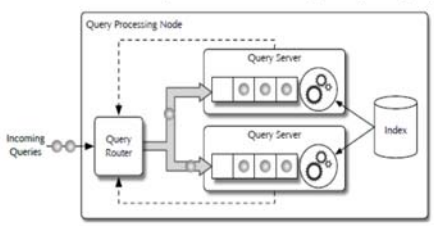

Fig 1. The architecture of a query processing node.

In our work, queue-per-core architecture is assumed to schedule job across CPU cores to minimize the energy consumption and to schedule the queries across the different query servers.

3.1 Minimum energy scheduling problem

Consider the following scenario, where a single core CPU must execute a set J = {J1, . . . Jn} of generic computing jobs rather than queries. Job must be done within the time interval. Every Job Ji has its own arrival time Ai and deadline Di. A schedule is feasible if each job in J is completed within the deadline, Minimum Energy Scheduling Problem (MESP) purposes to find a feasible schedule such that total energy consumption is minimized. It is an offline version of our problem, where jobs corresponding to the queries are pre-emptible. The YDS algorithm solves the MESP in polynomial time.

Algorithm 1 explains YDS algorithm,

Algorithm 1: The YDS algorithm

Data: A set of jobs $J = \{j_1, \ldots, j_n\}$ to schedule in $[t_0, t_1]$
Result: A feasible schedule S for J minimizing $E(S)$
OYDS(J):
$\psi \leftarrow \{\}$
$\phi \leftarrow \{\}$
while $J \neq \{\}$ **do**
  Identify $I^* = [z, z']$ and compute $g(I^*)$
  Set processor speed to $g(I^*)$ for jobs in J_{I^*} in ψ
  Schedule jobs in J_{I^*} according to EDF in ϕ
  Remove I^* from $[t_0, t_1]$
  Remove J_{I^*} from J
  foreach $J_i \in J$ **do**
    if $a_i \in I^*$ **then**
      $a_i \leftarrow z'$ // Update arrival times
    if $d_i \in I^*$ **then**
      $d_i \leftarrow z$ // Update deadlines
return $S = (\psi, \phi)$

YDS works by analyzing each possible time interval ***I*** in [t0,t1]. Then it finds critical interval ***I*** and maximizes the intensity. YDS schedules the job Ji in Earliest Dead First (EDF) policy. Then, if not pre-empted, the jobs in Ji will terminate in Ri = Vi - g(I) time units since the beginning of their execution. Then jobs in Ji are removed from J and Interval I from [t0,t1]. So, YDS updates the arrival times and deadlines of the remaining jobs to be outside I. Finally, YDS frequently finds a new critical interval for the remaining jobs are eventually scheduled.

Some issues with the YDS are as follows,

1. YDS is an offline algorithm to schedule generic queries, cannot be used to schedule online queries.
2. YDS need to know the processing volumes of job in advance.
3. YDS schedules job using processing speed which is continuous and unbounded.
4.

IV. Conclusion

In this paper we proposed Predictive Energy Saving Online Scheduling (PESOS) algorithm. In the perspective of web search engines, PESOS aims to reduce the CPU energy consumption of a query processing node while considering the tail latency on query processing time. PESOS select the lowest CPU core frequency so that energy consumption is reduced and time limit respectively. Also PESOS exploits two types of query efficiency predictors (QEP's), one to estimate the processing volumes, another for estimate the query processing time under different core frequencies. We then defined two possible configurations for PESOS: time conservative and energy conservative. Also we explained Minimum Energy Scheduling Problem and how to overcome with that. Then YDS offline algorithm was explained, such that how YDS works, what are its cons.

References

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

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

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