Modeling Query Energy Costs in Analytical Database Systems with Processor Speed Scaling

Boming Luo[1], Yuto Hayamizu[1], Kazuo Goda[1], and Masaru Kitsuregawa[1,2]

[1] The University of Tokyo, Tokyo, Japan
{luo,haya,kgoda,kitsure}@tkl.iis.u-tokyo.ac.jp
[2] National Institute of Informatics, Tokyo, Japan

Abstract. Energy efficiency in analytical database systems is becoming increasingly important because of the rapid growth in energy consumed by data centers driven by the recent big data boom. Previous studies showed that processor speed scaling has the potential to improve energy efficiency of analytical queries. These results, however, were obtained from measurement of specific queries. The power–performance characteristics of processor speed scaling specific to analytical database systems still remains unexplored despite their importance in energy efficient analytical query processing. We tackle this problem by modeling the energy costs of analytical queries with processor speed scaling based on query processing throughput. Our experimental evaluation shows that our energy model can be fitted within an error of 1.65 % and can be used to identify power–performance characteristics of analytical queries.

1 Introduction

Energy management is a primary concern in current data centers. The consumption of energy in data centers has been rapidly growing and is forecasted to reach 8 % of worldwide energy production by 2020 [1]. Because the rate of worldwide data generation is increasing rapidly [2], an increasing amount of IT resources, e.g., servers and storage systems, have been installed in data centers to develop large-scale data analytics platforms. Therefore, energy efficiency in analytical database systems—the key component of the platforms—is becoming a serious concern.

Processor speed scaling, also referred to as Dynamic Voltage and Frequency Scaling (DVFS), is a well-known power–performance tuning knob. Prior studies of processor speed scaling adopted a *measurement-based approach* for investigating its power–performance characteristics in analytical database systems. Tsirogiannis et al. [3] analyzed power–performance profiles among various hardware configurations and reported that the higher operating frequency resulted in the better energy efficiency. In contrast, Götz et al. [4] demonstrated that the most energy-efficient operating frequency was not always the highest one and largely varied depending on workload characteristics. While these measurement-based studies provided practical insights about the power–performance characteristics of processor speed scaling in analytical database systems and the

guidelines on energy management specific to the measured queries, they cannot be applied to a wide spectrum of analytical queries.

Herein, we tackle a *model-based approach* to investigating the power–performance characteristics of processor speed scaling for analytical query processing. In this paper, we focus on modeling an energy cost of analytical queries comprising sequential scans as basic building blocks, which has not been studied in the literature as far as we know. The presented model enables us to quantitatively analyze the effect of processor speed scaling on power–performance characteristics of a wide range spectrum of analytical queries.

2 Analytical Database Systems with Processor Speed Scaling

Processor Speed Scaling. For improving energy efficiency in analytical database systems, processor speed scaling, also referred to as DVFS, is a well-known tuning knob for runtime energy management adaptive to workload shifts. Usually, a processor defines available frequency levels and preset voltage levels for each frequency level like Intel SpeedStep Technology in Intel Xeon processors [5], thus an operating system or an application program can manage the processor power consumption of processor cores through this interface. Some studies have reported that processor speed scaling can potentially improve the energy efficiency of analytical query processing [3,4,6,7]; however the power–performance characteristics of processor speed scaling in analytical databases are still largely unexplored mainly because their measurement-based approaches on specific queries.

Energy Saving Opportunity for Analytical Database System. Because energy efficiency is defined as query processing throughput per power consumption, the key problem is correctly quantifying the balance between these two metrics. When a query is compute intensive and query processing throughput is restricted by processor performance, query processing throughput and processor power consumption cased by query processing are approximately proportional to the operating frequency of the processor. Because power is also consumed by other components and peripherals, energy efficiency is considered higher for higher operating frequencies. Conversely, when query processing throughput is restricted by other factors, e.g., storage I/O, there are potential ways for reducing power consumption with little impact on performance. As sequential I/O patterns are known to account for the bulk of analytical query I/O workloads [8], instruction execution in the processor can be easily overlapped with I/O operations using common I/O read-ahead techniques. In this scenario, as long as processor computing throughput is higher than I/O throughput, lowering the operating frequency should not result in performance penalty and may possibly improve energy efficiency. To the best of our knowledge, these opportunities for energy efficiency improvement in analytical database system have mentioned [9] in previous studies but have not been quantitatively modeled.

3 Throughput-Based Energy Cost Formulation

We take a model-based approach to identify the power–performance characteristics of processor speed scaling on analytical database server comprising processors, storage devices, memory modules, and other peripherals, e.g., a motherboard, cooling fans, and power supply unit. By modeling the power–performance characteristics, we can identify the behavior of power consumption and performance over operating frequencies and voltages of a processor. We focus on single query execution and analytical queries comprising sequential scans as basic building blocks as a first step toward understanding the power–performance characteristics of a vast variety of analytical workloads.

We adopt the pipeline model for energy cost modeling as it is the general concept in database systems and used in the literature [9, 10]. For query execution, database system generates a query execution plan comprising a tree structure whose nodes are database operators. By grouping database operators which can be concurrently executed as a single pipeline, one or more subtrees are formed and we refer to these subtrees as *basic execution blocks*. The workload is steady and the fluctuate of the system power consumption is relatively small in a single basic execution block, thus we formulate an energy cost model at this granularity.

For basic execution blocks comprising sequential scan and join operators, the throughput θ (tuples per unit time) of a basic execution block can be limited either by the computational throughput θ^{CPU} of the processor, which is generally proportional to the operating frequency f of the processor, or by the I/O throughput θ^{IO} of the storage. Hence, θ can be expressed as follows:

$$\theta = \min\left(\theta^{\mathrm{CPU}}, \theta^{\mathrm{IO}}\right), \quad \theta^{\mathrm{CPU}} = af$$

where a is a coefficient that depends on the characteristics of the given processor and workload. Given that N is the number of tuples processed in a basic execution block, the execution time T of the basic execution block can be calculated as follows:

$$T = N/\theta = N/\min\left(af, \theta^{\mathrm{IO}}\right) \tag{1}$$

θ is limited by θ^{CPU} and T is inversely proportional to f when $f < \theta_{\mathrm{IO}}/a$, whereas T is independent of f when $f > \theta_{\mathrm{IO}}/a$. $f = \theta_{\mathrm{IO}}/a$ is the balance point between processor and storage throughput and we refer to this frequency as the *boundary frequency*.

Let us now consider power consumption of the system. We assume that power consumption of the active processor cores and storage devices stays steady within a single basic execution block and power consumption of other components of the server stays constant regardless of the types of basic execution blocks [1]. Fan et al. [11] suggested a nonlinear model of processor power consumption of analytical

[1] Although this assumption does not always hold on realistic environments, modeling power consumption as a time-varying function and considering variation of power consumption of other components are beyond the scope of this paper.

workload as function of CPU utilization, therefore, we adopt this model. In this model, the difference in power consumption from the idle state ΔP^{CPU} can be expressed as $\Delta P^{\mathrm{CPU}} \propto 2u - u^r \quad (1 \leq r)$. On the assumption that u is solely determined based on θ, then u can be expressed as $u = \theta/\theta^{\mathrm{CPU}}$. It is known that the higher the operating frequency f, the higher the operating voltage V required for stable operation [12]. Although the actual relationship between f and V depends on processor implementations, we assume that $V(f)$ is a stepwise function of f following common implementations. Consequently, P^{CPU} can be expressed as follows:

$$P^{\mathrm{CPU}} = \left\{AfV^2 + Bf + C(V - V_{\mathrm{idle}})\right\}(2u - u^r) + P^{\mathrm{CPU}}_{\mathrm{idle}} \tag{2}$$

where A, B, C are coefficients and $V_{\mathrm{idle}}, P^{\mathrm{CPU}}_{\mathrm{idle}}$ represent the voltage and power dissipation during the idle state, respectively. In the curly brackets, each term corresponds to transition power dissipation, short-circuit power dissipation and leakage power dissipation.

Next, we assume that storage power consumption P^{IO} is proportional to the utilization of disk. P^{IO} can be expressed as follows:

$$P^{\mathrm{IO}} = D\theta/\theta^{\mathrm{IO}} + P^{\mathrm{IO}}_{\mathrm{idle}}$$

where D is the amount of increase in power under the maximum utilization of disk and $P^{\mathrm{IO}}_{\mathrm{idle}}$ is power dissipation during the idle state.

Finally, we assume that power consumption of other components (P^{others}) is constant. Thus, the total power consumption of system P is calculated as follows:

$$P = P^{\mathrm{CPU}} + P^{\mathrm{IO}} + P^{\mathrm{others}}$$

Because energy consumption E is the product of power consumption P and execution time T, E can be expressed as follows:

$$\begin{aligned} E = PT &= \left(P^{\mathrm{CPU}} + P^{\mathrm{IO}} + P^{\mathrm{others}}\right) T \\ &= N\left[\frac{AfV^2 + Bf + C(V - V_{\mathrm{idle}})}{\theta}(2u - u^r) + \frac{D}{\theta^{\mathrm{IO}}} + \frac{P^{\mathrm{CPU}}_{\mathrm{idle}} + P^{\mathrm{IO}}_{\mathrm{idle}} + P^{\mathrm{others}}}{\theta}\right] \end{aligned}$$

Let us now consider the operating frequency f_{opt} that minimizes energy consumption. The value of f_{opt} depends on where throughput-determining factor is. We assume that the processor has n levels of operating frequencies ($\{f_1, \cdots, f_n\}$, $f_i < f_{i+1}$) and there are m rising edges of $V(f)$ ($\{f_{s_1}, \cdots, f_{s_m}\}$) such that $V(f_{s_j}) > V(f_{s_j - 1})$. When throughput θ is limited by storage ($f > \theta_{\mathrm{IO}}/a$), E monotonically increases with f. When throughput θ is limited by processor ($f < \theta^{\mathrm{IO}}/a$), if there exists no rising edges in $V(f)$, E monotonically decreases with f. If not, E increases by the increase of $V(f)$ at a rising edge f_{s_j} and might take a local minimum value at $f_{s_j - 1}$. Therefore, f_{opt} can be categorized into three cases and determined through the proposed model:

(a) the minimum frequency f_1, when $f_i > \theta^{\mathrm{IO}}/a$ $(i = \{1, \cdots, n\})$
(b) the boundary frequency f_b or $f_{s_j - 1}$ $(s_j < b)$, when there exists f_b that satisfies $f_b > \theta^{\mathrm{IO}}/a$ $(1 < b \leq n)$ and $f_k < \theta^{\mathrm{IO}}/a$ $(k = \{1, \cdots, b-1\})$
(c) the maximum frequency f_n or $f_{s_j - 1}$, when $f_i < \theta^{\mathrm{IO}}/a$ $(i = \{1, \cdots, n\})$

4 Experimental Evaluation

Experimental Environment. We used HP Z440 Workstation as the server, equipped with Intel Xeon E5-1603 v4, 8GB memory, Seagate BarraCuda (3.5 inch, 2TB, 7200rpm). The operating frequency ranged from 1.2 to 1.9 GHz and from 2.1 to 2.8 GHz in 0.1 GHz steps ($f = \{f_1, \cdots, f_{16}\}$). Power consumed by the whole server system was recorded at 20 Hz sampling using Yokogawa WT1800. We used Linux 3.10.0 as kernel, PostgreSQL 9.6.3 as DBMS and the TPC-H dataset with a scale factor of 100. We observed the system had a jump in power consumption at 2.8 GHz in micro-benchmark as preliminary experiment, thus the CPU operating voltage in our system was assumed to be V_1 when $f \leq 2.7$ GHz and V_2 when $f = 2.8$ GHz.

Query. Using the TPC-H dataset, we defined evaluation queries based on full table scan. These queries are named as queries A, B and C.

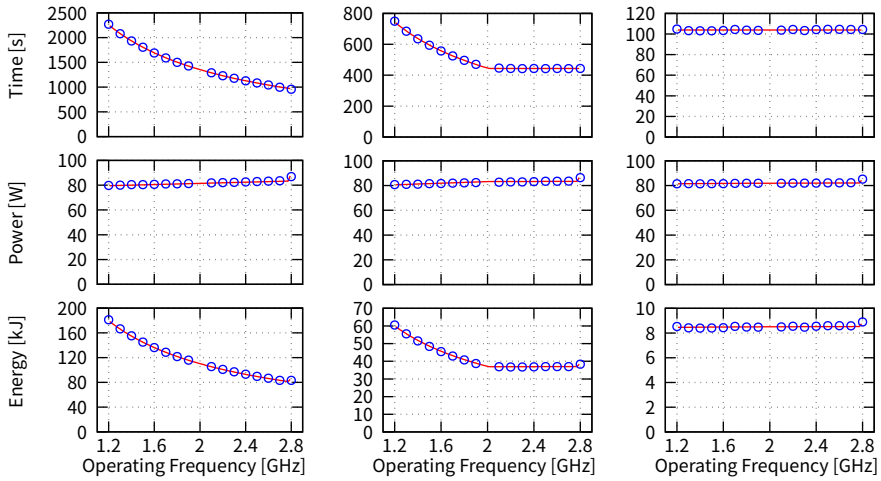
- **Query A**: $\sigma(\texttt{LINEITEM})$ with five aggregate expressions and two grouping variables.
- **Query B**: $\sigma(\texttt{LINEITEM})$ with an aggregate expression.
- **Query C**: $\sigma(\sigma(\texttt{ORDERS}) \bowtie \texttt{LINEITEM})$

We confirmed that full table scan was used for the execution plan of queries A and B. Additionally, full table scan and hash join were used for query C. The execution plans of both query A and B consisted of one basic execution block. The execution plan of query C consisted of two basic execution blocks because it performed hash join and we refer to them as C(1) and C(2).

Parameter Calibration. We measured the execution time and power consumption of queries A, B, and C for each operating frequency and then calibrated the parameters of each basic execution block in following steps: (i) obtained values of N for each basic execution block from the estimation of PostgreSQL query planner, (ii) calibrated the parameters which are not specific to basic execution blocks, and (iii) calibrated basic-execution-block-specific parameters.

Result. Fig. 1 shows power-performance curves of each basic execution block (the result of query C(2) has omitted by the limitation of space). We can find the slope of the power consumption graph differed among the range of frequencies. In particular, the slope was smaller in the IO-bound condition in comparison with the CPU-bound condition. CPU utilization (u) was inversely proportional to operating frequency (f) when operation was IO-bound; hence, the effect of the increase in power consumption because of the increase in the operating frequency in Eq. (2) was nullified.

With regard to the frequency f_{opt} that minimized energy consumption, it was estimated to be 2.7 GHz for query A, while the actual minimum energy point was 2.8 GHz. The minimum energy point of queries B were 2.1 GHz by the model,



(a) Query A (b) Query B (c) Query C(1)

Fig. 1: Power–performance characteristics of processor speed scaling: measured (with blue circle) and modeled (with red line)

whereas they were 2.2 GHz in the actual measurement. One possible cause of these discrepancies was a factor influencing execution time which the proposed model was unable to capture; however this requires further investigation. The increase in energy consumption because of these misestimation was 0.1 % for query A, 0.4 % for query B; however that for query C was 1.2 %.

Overall, the measured results and the graph described by the fitted model almost agreed with each other in terms of execution time, power consumption, and energy consumption. The error in energy consumption between the measured value and the model-fitted value was within 1.65 %. These results confirmed the effectiveness of the presented model.

5 Related Work

Processor speed scaling, also referred to as DVFS, is a common method for adjusting power and performance by switching the operating frequency and voltage of the processor. The basic approach of DVFS energy-saving policies is lowering the operating frequency at low CPU utilization to reduce power consumption with small performance penalty, as the general method taken by OS. In the context of transaction processing workload, there are studies about application-aware control of processor speed scaling by constructing power model [13, 14] or SLA-based control mechanism [14, 15].

For analytical query processing, the energy efficiency of hardware configurations have been comprehensively investigated in the late 2000s [16, 17]. Recently,

there have been several studies on energy-aware query optimization techniques and frameworks [18–20]. With regard to energy management with processor speed scaling, conventional approaches were measurement-based. Tsirogiannis et al. [3] extensively measured the energy efficiency of processor configurations with various types of analytical queries, analyzed the power–performance profiles, and concluded that the highest frequency tended to be the most energy efficient. Contrary to their study, Götz et al. [4] reported that the best frequency varied based on query workloads and proposed a technique for calibrating frequency with workload recording and query benchmarking. Manousakis et al. [7] proposed a feedback DVFS controller using real-time power sensor measurement values. In contrast to these studies, we adopted a model-based approach to formulate the power–performance characteristics of analytical query processing with processor speed scaling.

6 Conclusion

In this paper, we took a model-based approach to identifying power–performance characteristics of processor speed scaling for analytical query processing. We presented throughput-based formulation of an energy cost model. The presented model enabled us to quantitatively analyze the effectiveness of processor speed scaling for analytical queries comprising sequential scans as basic building blocks. The experimental evaluation showed that the energy model agreed with observations within an error of 1.65 %.

As part of our future research, we plan to expand our model to workloads with index table scan or sorting and to examine the effectiveness of the proposed model on various hardware configurations.

References

1. Chalise, S., Golshani, A., Awasthi, S.R., Ma, S., Shrestha, B.R., Bajracharya, L., Tonkoski, R.: Data Center Energy Systems: Current Technology and Future Direction . In: PES GM '15. (July 2015)
2. Reinsel, D., Gantz, J., Rydning, J.: Data Age 2025: The Evolution of Data to Life-Critical. Issue paper (April 2017)
3. Tsirogiannis, D., Harizopoulos, S., Shah, M.A.: Analyzing the Energy Efficiency of a Database Server. In: SIGMOD '10. (June 2010) 231–242
4. Götz, S., Ilsche, T., Cardoso, J., Spillner, J., Kissinger, T., Aßmann, U., Lehner, W., Nagel, W.E., Schill, A.: Energy-Efficient Databases Using Sweet Spot Frequencies. In: UCC '14. (February 2014) 871–876
5. Intel: Enhanced Intel SpeedStep Technology for the Intel Pentium M Processor. White paper (March 2004)
6. Lang, W., Patel, J.: Towards Eco-friendly Database Management Systems. In: CIDR '09. Volume cs.DB. (2009)
7. Manousakis, I., Marazakis, M., Bilas, A.: FDIO: A Feedback Driven Controller for Minimizing Energy in I/O-Intensive Applications. HotStorage '13 (June 2013)

8. Yu, P.S., Chen, M.S., Heiss, H.U., Lee, S.: On Workload Characterization of Relational Database Environments. IEEE Transactions on Software Engineering **18**(4) (April 1992) 347–355
9. Roukh, A., Bellatreche, L.: Eco-Processing of OLAP Complex Queries. In: Big Data Analytics and Knowledge Discovery. Springer, Cham, Cham (2015) 229–242
10. Kunjir, M., Birwa, P.K., Haritsa, J.R.: Peak Power Plays in Database Engines. In: EDBT '12. (March 2012) 444–455
11. Fan, X., Weber, W.D., Barroso, L.A.: Power Provisioning for a Warehouse-sized Computer. In: ISCA '07. (June 2007)
12. Flynn, M.J., Hung, P.: Microprocessor Design Issues: Thoughts on the Road Ahead. IEEE Micro **25**(3) (July 2005) 16–31
13. Korkmaz, M., Karsten, M., Salem, K.: Towards dynamic green-sizing for database servers
14. Xu, Z., Wang, X., Tu, Y.C.: Power-aware throughput control for database management systems. 315–324
15. Hayamizu, Y., Goda, K., Nakano, M., Kitsuregawa, M.: Application-Aware Power Saving for Online Transaction Processing Using Dynamic Voltage and Frequency Scaling in a Multicore Environment. In: ARCS '11. (February 2011) 50–61
16. Meza, J., Shah, M.A., Ranganathan, P., Fitzner, M., Veazey, J.: Tracking the Power in an Enterprise Decision Support System. In: ISLPED '09. (August 2009) 261–266
17. Poess, M., Nambiar, R.O.: Tuning Servers, Storage and Database for Energy Efficient Data Warehouses. In: ICDE '10. (March 2010) 1006–1017
18. Xu, Z., Tu, Y.C., Wang, X.: PET: Reducing Database Energy Cost via Query Optimization. VLDB **5**(12) (August 2012) 1954–1957
19. Roukh, A., Bellatreche, L., Ordonez, C.: EnerQuery: Energy-Aware Query Processing. In: CIKM '16. (October 2016) 2465–2468
20. Tu, Y.C., Wang, X., Zeng, B., Xu, Z.: A System for Energy-Efficient Data Management. ACM SIGMOD Record **43**(1) (May 2014) 21–26