

Towards Energy Efficient Query Processing in Database Management System

Report by: Ajaz Shaik, Ervina Cergani

Abstract

Rising concerns about the amount of energy consumed by the data centers, several computer hardware manufacturing companies are developing energy efficient components enabling to toggle the amount of power consumed by them at several workloads. But these features are not adopted by the Database Managements Systems which are one of the sources of large energy consumption. This report shows two techniques called Process Voltage/Frequency Control (PVC), Explicit Query Delays (QED) for trading energy efficiency into database systems with their experimental evaluation and also presenting energy efficient query processing framework which shows ease in deploying energy cost model into current query optimization system.

What is Energy Efficiency?

For computing systems and data centers, energy is delivered as electricity. Power is the instantaneous rate of energy used. Energy Efficiency can be defined as

$$\begin{aligned} \text{EnergyEfficiency} &= \frac{\text{FinishedWorkloads}}{\text{Energy}} \\ &= \frac{\text{FinishedWorkloads}}{\text{Power} * \text{Time}} \\ &= \frac{\text{Throughput}}{\text{Power}} \end{aligned}$$

In order to increase energy efficiency either the throughput is increased keeping the power constant or the power consumption is decreased by maintaining throughput at certain standards of the systems. This report concentrates second method of increasing energy efficiency.

CPU Power Management

It is observed from [1] that servers are rarely completely idle or operate to their maximum utilization. Instead most of times operating between 10 and 50 percent of their maximum utilization.

Modern hardware provides more than sixty percent of dynamic CPU power range which are handled internally by the system during various server workloads. Processors adopt technique called Dynamic Frequency and Voltage scaling to vary these power / performance states. These states are implementation dependent with $p0$ being the maximum and $p1$ to pN being lower power consumption states. Some processors support built in DVFS like Intel's SpeedStep and AMDs PowerNow designed for mobile chips and also AMDs CoolnQuite for desktop and server chips. Several external packages which allow programs to customize the performance states are also available.

Different CPU speeds can be achieved by varying CPU Multiplier which measures ratio of internal clock rate to externally supplied clock dictated by p-states and the Front Side Bus which carries data between CPU and Northbridge.

$$FSB * Multiplier = frequency$$

Underclocking is the process of slowing the CPU frequency by varying FSB speed. This allows fine granularity modulation upon which multipliers act on. Frequency changes through multipliers (p-state transition) put hard upper limit on CPU. For example, consider a CPU on a 333MHz FSB with p-state multiplier 9, 8, 7 and 6. Assume the capped value of 7, which means that the top frequency CPU can now achieve is 2.3GHz, instead of 3 GHz (= 9 * 333MHz). However, by modulating the base factor (FSB speed) capping the multiplier with 7 would mean that 2 more transitioning states are left with an FSB speed of 333MHz.

Towards Eco-DBMS

Database should utilize several modern hardware capabilities and also continuously monitor the energy consumption characteristics. This report presents two methods to show how changing system settings can save energy consumption with small amount of performance penalty.

Processor Voltage/Frequency Control (PVC)

This technique uses the underclocking method to trade energy for performance. To explore the energy savings of this method experimental evaluation on two databases commercial and MySql is performed with TPC-H Q5 workload. Each workload consists of 10 Q5 queries using 2 region and 5 date range non overlapping predicates. Finally each workload is run five times to the average of the middle three readings. The workload is performance many times to address the issue of CPU power fluctuation during the actual run.

Experimental setup uses ASUS motherboard and processor, 6-Engine software provided the system to underclock the FSB by 5%, 10%, and 15% and to downgrade the CPU voltage into its preset "small" and "medium" voltage downgrades. The results for commercial DBMS with scale factor of 1.0 and for MySql with scale factor of 0.125.

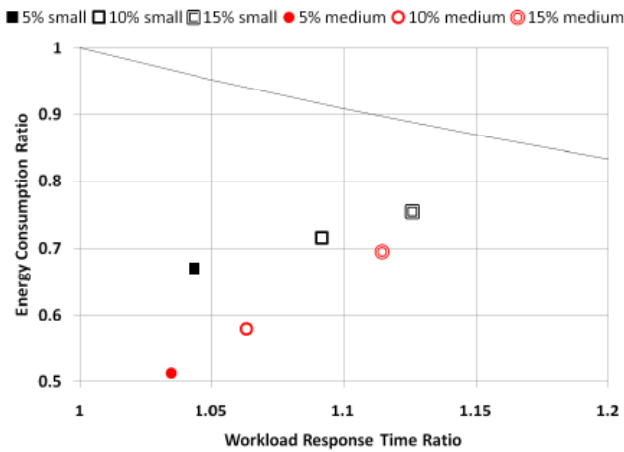


Figure 2: TPC-H Query 5 on a Commercial DBMS

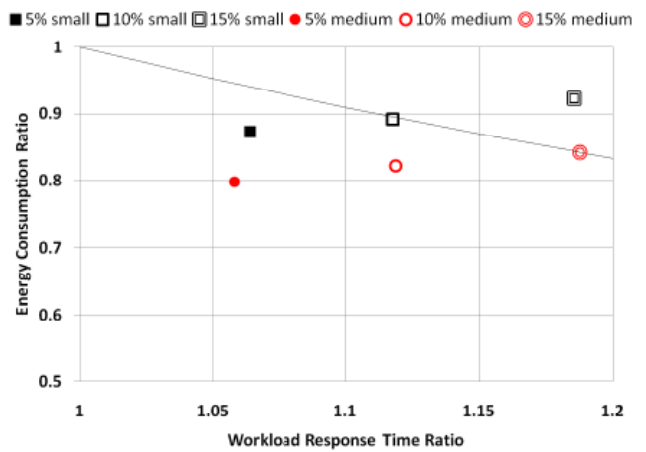


Figure 3: TPC-H Query 5 on MySQL

The graphs are plot with workload response of changed settings to stock settings ratio on X axis and Energy consumption of changed settings to stock settings ratio on Y axis. The stock settings can be referred to top left corner of the graphs. The solid line represents a metric called Energy Delay Product (EDP), which is the product of the energy and the delay and remains constant along the line.

As we can see in figure 2 the 5% settings dramatically reduces the energy by only a small drop in performance on commercial databases. However the change is not huge in MySQL as one could imagine that this database adopt less optimization methods compared to the other.

Explicit Query Delays (QED)

In traditional DBMS with workload of single table select-only queries with same but different range of selection predicates will result in each query being evaluated individually and one after other. In QED, queries are delayed and placed into a queue on arrival. When the queue reaches a certain threshold all the queries in the queue are examined to determine if they can be aggregated into a small number of groups, such that the queries in each group can be evaluated together. Evaluating this workload can trade average query response time for reduced overall energy consumption. Single group queries can be run in the DBMS at a lower energy cost than the individual queries.

For evaluating QED, simple workloads of a series of single table queries each having a selectivity of 2% based on `l_quantity` attribute of the `lineitem` table, with predicate drawing from 50 uniformly distributed integer values of TPC-H benchmark at 0.5 scale factor. The workload is run on MySQL database.

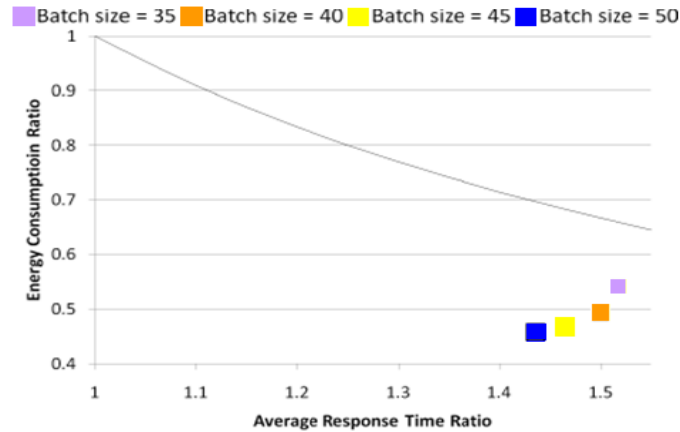
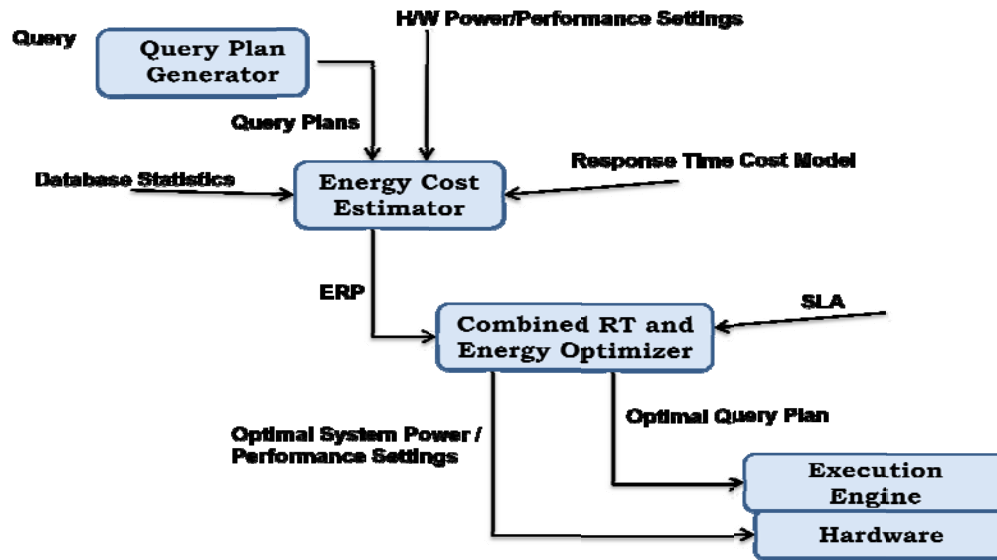


Figure shows the average response time of changed over stock settings ratio on X axis and energy consumption ratio on Y axis. As the batch size (threshold) increases the energy consumption decreases with very less change in response time. This is because single queries consume more energy in overall when compared to queries which run in groups.

Query Processing Framework



Typical servers have many opportunities to execute the query slower if the Service Level Agreement (SLA) permits the additional response time penalty. Since SLA's are written to meet peak or near peak demand they often have some slack in the frequent low utilization periods, which could be exploited by the DBMS. In the above framework it is assumed that queries have some response time goal,

potentially driven by an SLA. The query optimization problem now becomes to *Find and execute the most energy-efficient plan that meets the SLA.*

The Energy Response Time Profile (ERP) is a structure that details the energy and response time cost of executing each query plan at every possible power/ performance setting under consideration. Given a query Q with a set of possible plans $P = \{P_1, .. ,P_n\}$ to be executed on a machine with system operating settings $H=\{H_1, .. ,H_m\}$ to Energy Cost Estimator generates the ERP which contains one point for every plan for every system operating setting.

The generated ERP is then used by the combined energy enhanced query optimizer to choose the most energy efficient plan that meets SLA constraints. Then, a command to switch to the chosen system operating state is sent to the hardware, followed by sending the optimal query plan to the execution engine.

Energy Cost Model

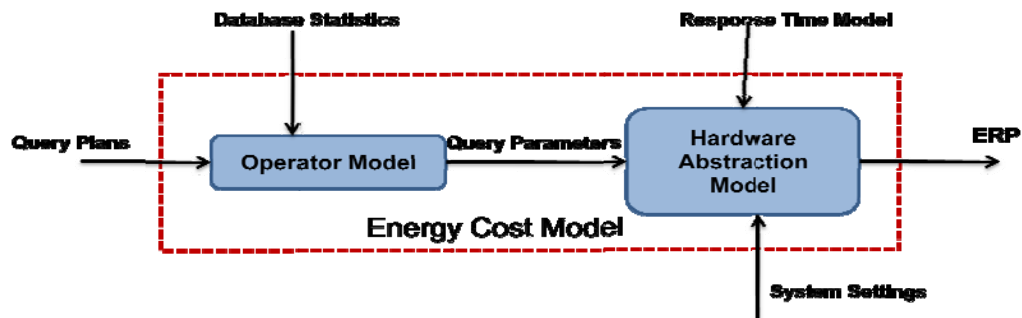


Figure above gives an overview of energy cost model,

- Operator model estimates the query parameters (CPU Instructions: I_Q disk page reads: R_Q , disk page writes: W_Q and memory page accesses: M_Q) for selection, projection, join of each query plan from the database statistics.
- Hardware abstraction model uses query parameters and system parameters (CPU: C_{cpu} , disk write: C_W , disk read: C_R memory access: C_{mm} , remaining system: C_{other}) that can be learnt through ,training procedures to accurately estimate the energy cost.

Equations below show the model for average power (Watt) during a query response time T.

$$P_{av} = C_{cpu} * \frac{I_Q}{T} + C_R * \frac{R_Q}{T} + C_W * \frac{W_Q}{T} + C_{mm} * \frac{M_Q}{T} + C_{others}$$

Conclusion

This report on [2, 3] shows how energy and performance can be traded into current database management system with emerging modern hardware mechanism that can provide significant energy savings for DBMS. And also present Energy aware Query Optimization by introducing Hardware Abstraction model in addition to the traditional Operator Model. The ERP analysis the relationship between different query plans and different hardware power / performance settings, we can trade decreased query performance for increased energy efficiency in the presence of slack in SLA's.

Remark

Notice from paper that in order to achieve a gain in efficiency there should be certain amount of performance degrades. Authors from the paper showed good evaluations of how tuning database systems according to the hardware setting can lead to saving of energy consumption of servers with small downgrade in application performance. For example considering method QED where individual query execution energy consumption is more when compared to executing in batch. However, authors did not mention of how QED method will be efficient with several tables and disjoint predicates. The proposed energy saving query processing model can be easily deployed into existing database systems. But the energy cost model explanation is abstract, details about how hardware and query parameter values are obtained. And also the papers consider only one application running on server, but in normal scenario this is not the case. So question of how applicable the proposed methods and model are when other applications are also running on the same server. In overall the authors give initiative direction towards eco friendly database management systems.

References

[1] *The Case for Energy Proportional Computing*, Luiz Barroso and Urs Holzle Google.

[2] *Towards Eco-friendly Database Management Systems*, Willis Lang and Jignesh M. Patel Computer Science Department, University of Wisconsin-Madison, USA

[3] *Rethinking Query Processing for Energy Efficiency: Slowing Down to Win the Race*, Willis Lang, Ramakrishnan Kandhan, Jigesh M. Patel Computer Sciences Department Univeristy of Wisconsin – Madison.