

# [Green database essay sample](https://assignbuster.com/green-database-essay-sample/)

ABSTRACT

Database should utilize several modern hardware capabilities. The purpose of our research study was based on the case of green data base undertaken by T Willis Lang and Jignesh M. Patel Computer Sciences Department University of Wisconsin Madison, USA, where we have suggested and studied the importance of database going ecofriendly and Data Centers and the Database management systems have always ignored the task of managing the energy consumed during query execution and query processing. The economic and environmental factors now require that Data Centers give attention to energy consumption and proper energy utilization. Energy efficiency, a new focus for general-purpose computing, has been a major technology driver in the mobile and embedded areas for some time. Earlier work emphasized extending battery life, but it has since expanded to include peak power reduction because thermal constraints began to limit further CPU performance improvements.

1. INTRODUCTION   
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. 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 shown in fig 1.

Figure 1:   
In order to increase energy efficiency either the throughput is increased keepingthe powerconstant orthe power consumption is decreased by maintaining throughput at certain standards of the systems. This report concentrates second method of increasing energy efficiency. and also continuously monitor the energy consumption characteristics. This report presents two methods to show how changing system settings can save energy consumption with smallamountofperformancepenalty. Processor Voltage/Frequency Control(PVC). This technique uses the under clocking 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 under clock 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.

Fig 1: TPC-H Query 5 on a Commercial DBMS

Fig 2: 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 solidlinerepresentsametriccalledEnergyDelay Product(EDP), which is the product of the energy and the delay and remains constant along the line. As we can see in figure 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.

3. 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 work load 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 of2% based on l\_quantity attribute of the line item table, with predicate drawing from 50 uniformly distributed integer values of TPC‐H benchmark at 0. 5 scale factor. The workload is run on MySqldatabase.

4. QUERY PROCCESSING 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 become to find and execute the most energy-efficient plan that meets the SLA.

5. PROCESSOR ENERGY EFFICIENCT TECHNIQUES   
Energy has become a crucial aspect in the design of modern processors. Processors use a technique called dynamic voltage and frequency scaling (DVFS) to automatically reduce the processor voltage/frequency, moving the processor to lower power/performance states (p-states). Intel’s Speed Step and AMD’s Power Now! And Cool’n’Quiet are examples of built- in automatic processor DVFS. However, many software packages allow programs explicit access to different p-states. P-states are characterized by the combination of CPU multiplier and CPU voltage settings. The CPU frequency is a product of the front side bus (FSB) speed and the CPU multiplier, where the CPU multiplier is dictated by the p-state.

Fig. 3: Query Processing Framework

The CPU voltage is based on the CPU multiplier, and a lower multiplier allows the CPU to operate at a lower voltage. In addition to p-states, users have the ability to more finely tune the CPU and the entire system speed by changing the FSB speed. Over clocking enthusiasts largely use this technique to make the system go faster. But, this technique can destabilize the system, and can reduces the life span of the processor and other hardware components. However, the converse, which is the ability to under clock the system by slowing the FSB speed, does not reduce component lifespan (it may do the reverse and increase the component’s lifespan). It is important to distinguish the difference between CPU frequency modulation through p- state transitioning and underclocking. Traditional methods of

CPU power management through p state manipulation put a hard upper limit on the top p-state that a CPU can achieve. For example, consider a CPU on a 333MHz FSB, with p-state multipliers of 9, 8, 7, and 6 and some corresponding decreasing voltages. When p-state control is used to manage power, the max p-state can be capped to a value. Lets assume a capped value of7. This means that the top frequency the CPU can now achieve is2. 3Ghz (= 7 × 333MHz), instead of 3Ghz (= 9 × 333MHz). In this paper, we use underclocking because instead of capping the multiplier, the FSB speed is now decreased which allows a finer granularity of CPU frequency modulation. Instead of dropping the frequency by 333 MHz with each multiplier cap, we simply modulate the base factor (the FSB speed) that the multiplier acts on. With underclocking, we also retain the full capabilities of p- state transitions (which allow the CPU to reduce its energy consumption dynamically).

In our example, capping the multiplier at 7 would mean that only 2 transitioning states are left available with an FSB speed of 333Mhz. Underclocking retains all the multiplier settings while globally reducing the frequency of all available p-states. This means we have all 4 settings corresponding to multiplier values ranging from 6 through 9, but these values are multiplied against a smaller (slower) FSB speed. Consequently, lowering the FSB speed is a finer grained tool for changing the CPU frequency. Underclocking also affects other components connected to the Northbridge hub. Main memory is on the Northbridge, and its operating frequency is a multiple of the FSB(usuallyadifferentmultiplierfromtheCPU multiplier). Thus, under clocking also slows the main memory, which in turn reduces the amount of energy consumed by main memory. We examine the effects of these smaller frequency modulationsthroughunderclocking ontheenergy consumption of the CPU.

6. SYSTEM UNDER TEST   
The system that we use in this paper has the following main components: ASUS P5Q3 Deluxe Wi-Fi-AP motherboard, Intel Core2-Duo E8500, 2×1GKingston DDR3 main memory, ASUS GeForce 8400GS 256M, and a Western Digital Caviar SE16 320G SATA disk. The power supply unit (PSU) used

was a Corsair VX450W PSU, which is labeled as an energy- efficient PSU under 80plus. org. System power draw was measured by a Yokogawa WT210 unit (as suggested by SPEC power benchmarks). The operating system used was Microsoft Windows Server 2008. All client applications accessing the database were written in Java 1. 6 using JDBC connection drivers to the different database systems we used. To measure the CPU power consumption, we used a hardware sensor provided by the mother board manufacturer. The ASUS motherboard has an EPU processor that directly measures the CPU power. The ASUS P5Q3 Deluxe 6-Engine software displays information gathered from this onboard hardware sensor. Unfortunately, the only display mechanism is a GUI that shows the current CPU wattage.

As a workaround, we recorded the CPU wattage by graphically sampling the GUI every second throughout the execution of the workload. CPU joules was recorded as the average sampled wattage multiplied by the workload execution time. There are drawbacks to using this method as the refresh rate of the 6- Engine is about 1 second. Furthermore, since we allowed Intel Speed step to act freely, the CPU power fluctuates during an actual run. To address these issues, we create workloads that contain 10 queries for a given type of query. Each of these 10 queries has input predicate ranges that don’t overlap. Then we take measurements for the entire workload, which is usually many minutes long. Finally, we run each workload five times and discard the top and bottom readings, and average the middle three readings. This average is reported in our results. In all our experiments, we did not create any database indices.

7. CPU POWER MANAGEMENT   
It is observed 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 sever 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 pNbeing lower power consumption states. Some processors support built in DVFS like Intel’s Speed Step and AMDsPower Now designed for mobile chips and also AMDs Cool n Quite 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   
Under clocking 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.

8. PVC   
The Processor Voltage/Frequency Control (PVC) technique uses the under clocking technique described above to trade energy for performance. We now present an experimental evaluation of PVC. For our tests, we use TPC-H as our workload. To keep the experiments manageable, we only ran TPC-H Q5. This query has a response time that is often close to the geometric mean of the power tests in many published results. This query has a six table join and a group by clause on one attribute. We ran TPC-H on both a commercial DBMS and

MySQL 5. 1. 28. To explore the energy savings resulting from under clocking the system, we first run the workload in “ stock setting” – which corresponds to no under clocking and represents the traditional way of processing queries. Then, we used the 6-engine software provided by ASUS to under clock the FSB by 5%, 10%, and 15%. In addition, we used this software to downgrade the CPU voltage into its preset “ small” and “ medium” voltage downgrades. We also ran the ASUS PC Probe II program, which continuously monitors the CPU settings and warns when the system settings result in instability. We found that using the “ small” and “ medium” CPU voltage downgrades, the system operated without any warnings. Other motherboard settings were as follows: CPU load line was set to “ light,” chipset voltage downgrade was set to “ on,” and CPU fan settings were set to “ bios setting.” Thus, for each workload, we have 7 CPU power results: stock (no altered settings) and the 3 under clocking settings, for the 2 CPU voltage downgrades. Unless stated otherwise, results will be presented as a ratio compared to the “ stock” CPU and motherboard settings.

CONCLUSION AND DIRECTIONS FOR FUTURE WORK   
In our paper, we students, have presented some proposal for considering energy efficient query processing in a database management system thus hereby helping the environment by reducing the energy consumption and helping the environment go ecofriendly. Green database management is a wider scope and provides a good scope for the researchers to create an energy efficient and ecofriendly system for the increasing demands of corporates employing utilization of database. Our proposals center around techniques that can trade energy for performance. We have also described two techniques that work within this framework.

Fig. 4: Energy versus Average per Query Response Time with QED

Actual experiments on two DBMSs demonstrate that these techniques achieve significant savings in energy, and often reduce the energy consumption by a larger amount compared to the degradation in response time (i. e. operate at lower EDPs). For future work, we plan on better understanding and measuring the energy consumption of all the hardware components in a DBMS server. This task is challenging as modern motherboards are complex (multi-layered) and simply tapping into the components (such as memory banks) is not trivial. Consequently, we may need to design some indirect ways of measuring the energy consumption of some components. Designing a DBMS to balance the response time versus energy consumption opens a wide range of research issues that per correlate through nearly all aspect of a DBMS, including query evaluation strategies, query optimization, query scheduling, physical database design, and dynamic workload management. In addition, there is also an opportunity for the database community to collaborate with the architecture community to influence the design of the next generation of hardware that will be more energy efficient, and to work towards building mechanisms that allow the DBMS to leverage the full potential of the energy saving features that the hardware will provide.

REFERENCES

[1] Report To Congress on Server and Data Center Energy   
Efficiency. EPA Technical Report, 2007.   
[2] L. A. Barroso and U. H¨olzle. The Case for   
Energy-Proportional Computing. IEEE Computer,   
40(12): 33–37, 2007.   
[3] C. Belady. In the Data Center, Power and Cooling Costs   
More than the IT Equipment it Supports. In Electronics   
Cooling, volume 23, 2007.   
[4] K. G. Brill. Data Center Energy Efficiency and Productivity. In The Uptime Institute – White Paper, 2007.   
[5] J. Dean and S. Ghemawat. MapReduce: Simplified Data   
Processing on Large Clusters. In OSDI, 2004.   
[6] Enermax. http://www. enermaxusa. com/.   
[7] X. Fan, W.-D. Weber, and L. A. Barroso. Power Provisioning for a Warehouse-Sized Computer. In International   
Symposium on Computer Architecture, pages 13–23, 2007.   
[8] G. Graefe. Database Servers Tailored to Improve Energy   
Efficiency. In Software Engineering for Tailor-made Data   
Management, pages 24–28, 2008.   
[9] S. Greenberg, E. Mills, W. Tschudi, P. Rumsey, and   
B. Myatt. Best practices for data centers: Lessons learned   
from benchmarking 22 data centers. ACEEE Summer Study   
on Energy Efficiency in Buildings, 2006.   
[10] J. Hamilton. Where Does the Power go in DCs & How to get it Back. 2008.   
[11] J. G. Koomey. Estimating Total Power Consumption by   
Servers in the US and the World. 2007.   
[12] S. Rivoire, M. A. Shah, P. Ranganathan, and C. Kozyrakis. JouleSort: A Balanced Energy-Efficiency Benchmark. In   
SIGMOD Conference, pages 365–376, 2007.   
[13] R. R. Schmidt, E. E. Cruz, and M. K. Iyengar. Challenges of Data Center Thermal Mangement. IBM Journal of Research   
and Management, 49(4), 2005.   
[14] T. K. Sellis. Multiple-query optimization. ACM Trans.   
Database Syst., 13(1): 23–52, 1988.   
[15] S. Siddha, V. Pallipadi, and A. V. D. Ven. Getting Maximum Mileage Out of Tickless. In Proceedings of the Linux   
Symposium, pages 201–208, June 2007.   
[16] M. Stonebraker. Operating System Support for Database   
Management. Communications of the ACM, 24(7): 412–418,   
[17] Transaction Processing Council.   
http://www. tpc. org/tpch/.   
[18]. EPA Technical Report, 2007.   
Towards Ecofriendly Database Management Systems   
[19]. Willis Lang and Jignesh M. Patel.   
[20] Willis Lang and Jignesh M. Patel   
Computer Sciences Department   
University of WisconsinMadison,