International Journal of Research and Development in Applied Science and Engineering (IJRDASE)

Structural Design for Energy-aware Database Management Systems

Shailja Gupta
Computer Science & Engineering,
A.I.E.T., Lucknow, India
shailja.1225@gmail.com

Atma Prakash Singh
Computer Science & Engineering,
A.I.E.T., Lucknow, India
talk2aps@gmail.com

***Abstract*—This paper presents an open database storing energy-relevant data from a multi-robot flexible factory automation system. with the rising energy prices and energy-inefficient server deployments, it's clear that there's a requirement for DBMSs to think about energy efficiency as a superior operational goal. The traditional query optimizer is primarily concerned with maximizing the query performance. We propose a framework which helps reduce system energy consumption and response time measurements for executing the query using two different query plans – hash join (HJ) and sort-merge join (MJ) at two different system settings on MySQL. The energy measurement is the actual energy that is drawn by the entire server box – i.e., we measure the energy drawn from the wall socket by the entire system. This paper presents a replacement framework for energy-efficient query process. The framework augments question plans created by ancient traditional optimizers with associate energy consumption prediction to provide associate Energy Rejoinder Time Profile for a query. Power Efficient Software.**

Key words*—*Cloud Computing, Database Energy Optimization, and Green Cloud

1. **Introduction**

Cloud Computing has modified the approach applications and computer code are distributed and used additional and more firms are shift to Clouds more their IT value. Thus, it's become vital to handle the energy potency at application and information center level. This aspect has received little attraction since several applications are already on use and most of the new applications are largely upgraded version of or developed victimization antecedently enforced tools. A number of the efforts during this direction are for MPI applications [1], That are designed to run directly on physical machines. Thus, their performance on virtual machine remains indefinite power economical techniques [2] for computer code are largely for embedded devices. Therefore, to attain energy potency at application level, SaaS suppliers ought to listen in deploying computer code and information on proper of infrastructure which might execute the computer code most expeditiously. This necessitates the analysis and analysis of trade-off between performance and energy consumption because of execution of software on multiple platforms and hardware. Additionally, the energy consumption at the using level and code level ought to be thought of by software developers within the style of their future application implementations victimization varied energy-efficient techniques planned.

2. Related work

Willis Lang et al. (2012) Energy is a growing component of the operational cost for many "big data" deployments, and hence has become increasingly important for practitioners of large-scale data analysis who require scale-out clusters or parallel DBMS appliances. Although a number of recent studies have investigated the energy efficiency of DBMSs, none of these studies have looked at the architectural design space of energy-efficient parallel DBMS clusters. There are many challenges to increasing the energy efficiency of a DBMS cluster, including dealing with the inherent scaling inefficiency of parallel data processing. In this work, authors experimentally examine and analyze a number of key parameters related to these challenges for designing energy-efficient database clusters. We they explore the cluster design space using empirical results and propose a model that considers the key bottlenecks to energy efficiency in a parallel DBMS.

Bin Zhang, (2012) In order to fully exploit the potential of optimization algorithms for energy savings in factory automation settings, it is needed to bring together engineers that have knowledge of signal processing algorithms with shop floor engineers that are experienced with real manufacturing processes and factory automation settings. authors presents an open database storing energy-relevant data from a multi-robot flexible factory automation system. The database was populated with data regarding equipment CAMX state events, process/cell/device energy consumption and quality. The database was accessed via a URL and will be further populated with data related to production performance, machine failures and temporary stops etc. The intention is to provide algorithm developers with the needed datasets for design of optimization algorithms for energy efficient manufacturing settings. In order to fully exploit the potential of optimization algorithms for energy savings in factory automation settings, it is needed to bring together engineers that have knowledge of signal processing algorithms with shop floor engineers that are experienced with real manufacturing processes and factory automation settings. For this, the easiest approach would be to have large datasets gathered and stored in the cloud, together with information on how the data can be accessed, so that every interested party can make use of it. This work was for such an initiative.

3. Clouds energy efficient options

There is an excellent concern within the community that Cloud computing may result in higher energy usage by the datacenters, the Cloud computing green inexperienced lining. There are many technologies and ideas used by Cloud suppliers to realize higher utilization and efficiency than ancient computing. Therefore, relatively lower carbon emission is anticipated in Cloud computing owing to extremely energy economical infrastructure and reduction within the IT infrastructure itself by multi-tenancy. The key driver technology for energy economical Clouds is "Virtualization," that permits vital improvement in energy efficiency of Cloud suppliers by leverage the economies of scale related to sizable amount of organizations sharing a similar infrastructure. Virtualization is that the method of presenting a logical grouping or set of computing resources so they will be accessed in ways in which offer edges over the first configuration [7]. By consolidation of underutilized servers within the variety of multiple virtual machines sharing same physical server at higher utilization, corporations will gain high savings within the variety of area, management, and energy.

3.1 Energy efficient database management system

As mentioned higher than, with the rising energy prices and energy-inefficient server deployments, it's clear that there's a requirement for DBMSs to think about energy efficiency as a superior operational goal. In fact, driven by requests from its customers, the group action process Performance Council has moved during this direction, and every one benchmarks currently have a part for coverage the energy consumed once running the benchmarks [12]. Whereas the primary version of this benchmark has resulted in a very compromise that creates this energy coverage non mandatory, the organization clearly expects that "Competitive demands can encourage take a look at sponsors to incorporate energy metrics as before long as attainable." Consequently, info and hardware vendors that want to report TPC energy metrics can have a keen interest in minimizing their "power/performance" results. Mercantilism performance for lower energy consumption will happen as a result of a database management system server could have opportunities to execute the question slower, if the extra delay is suitable.

4. Proposed Framework

In this section we tend to propose a general question process organized framework, that uses each energy and interval toe the optimization criteria. The queries used in this work: (1) the way to fine tune the work of the question optimizer in lightweight of its extra responsibility to optimize for energy usage? (2) Given this new responsibility, however will we style a question optimizer?

4.1 New Role of the Query Optimizer

The traditional query optimizer is primarily concerned with maximizing the query performance. The optimizer's new goal now is to find query plans that have acceptable performance, but consume as little energy as possible. We want the query optimizer to return an energy-enhanced query plan that is to the left of the Service Level Agreement(SLA) -dictated performance requirement, and as low as possible along the y-axis. The main task here is to generate energy rejoinder profile plots like that shown in Figure 1. In Figure 1 we show an energy rejoinder profile for a single equijoin query on two Wisconsin Benchmark relations [10].

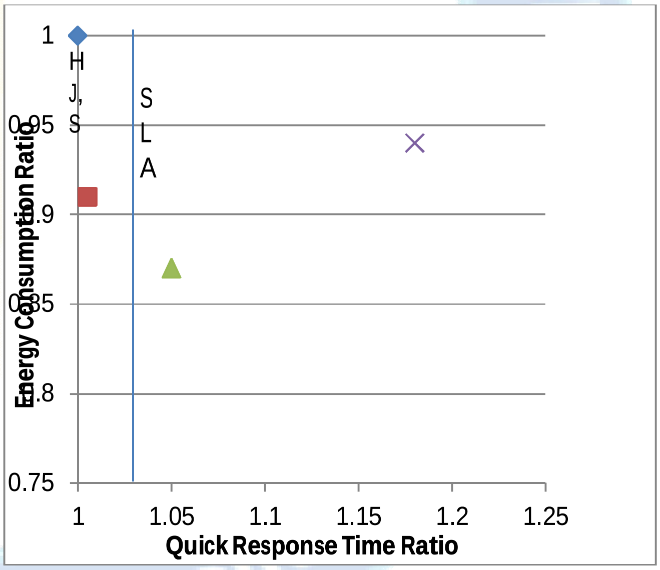


Fig. 1. Energy rejoinder profile for a single equijoin

The plot shows the system energy consumption and response time measurements for executing the query using two different query plans – hash join (HJ) and sort-merge join (MJ) at two different system settings (labeled 'S', 'M'), on MySQL. System setting 'S' corresponds to the "stock" (high power/performance) system settings. System setting 'M' is the power/performance setting that can save energy by reducing the memory capacity as we simulated in Figure 1. The energy measurement is the actual energy that is drawn by the entire server box – i.e., we measure the energy drawn from the wall socket by the entire system. In this figure, we have plotted all the other data points proportionally relative to (HJ, S), for both the energy consumption (on the y-axis), and the response time (on the x-axis). A response time SLA is represented by a dashed vertical line. To generate these plots, the query optimizer needs to quickly and accurately estimate both the response time and the energy consumption for each query plan. Query optimizers today are fairly good at predicting the response time, but doing so while accounting for varying hardware configuration is a new challenge. Of course, this challenge of predicting the energy consumption of a given query plan for a specific system setting also requires that the query optimizer understands the power/performance settings that the hardware offers.

4.2Framework

The question is provided as input to the question arrange generator in SQL Server, that is then requested to list all the promising question plans that the optimizer has known. These question plans at the side of the data regarding accessible system settings is provided as input to the Energy value reckoner, that generates the energy rejoinder profile exploitation associate analytical model for predicting the energy consumption. The generated energy rejoinder profile is then employed by the combined energy-enhanced question optimizer to decide on the foremost energy-efficient arrange that meets SLA constraints. Then, a command to change to the chosen system in operation state is distributed to the hardware, followed by causation the best query commit to the execution engine.

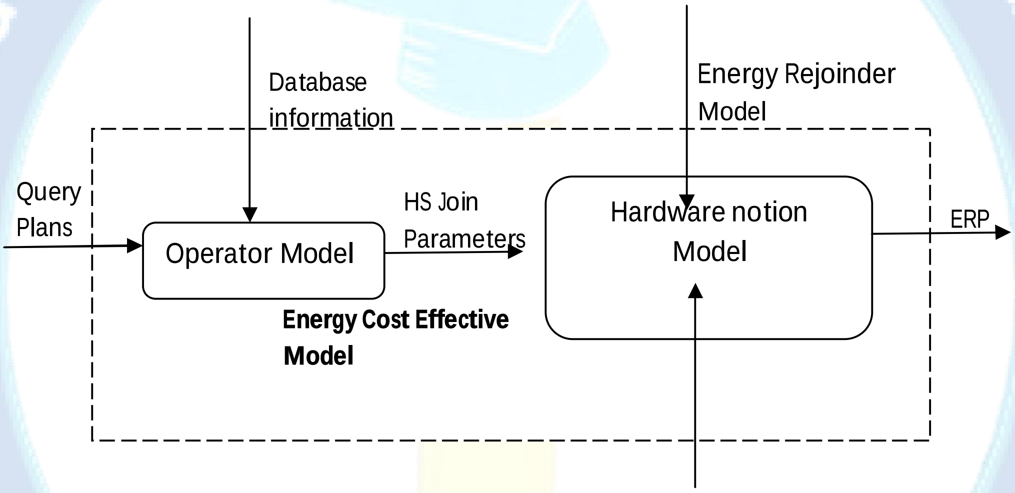


Fig. 2. an summary of the energy price model. The operator model estimates the question parameters needed by the hardware abstraction model for every query set up from the information statistics on the market. The hardware abstraction model uses the query parameters calculable and also the system parameters learnt to accurately estimate the energy price.

Here, we have a tendency to in brief describe AN analytical model that estimates the energy price of executing a query at a specific power/ performance system setting. Our model abstracts away the energy price of a query in terms of system parameters that may be learnt through a "training" procedure, and query parameters (CPU directions, memory fetches, etc.) that may be calculable from on the market information statistics. we would like to develop a straightforward, portable, practical, and correct methodology to estimate the energy price of a query set up. sadly, previous techniques accustomed estimate energy consumption fail to satisfy one or a lot of those goals. as an example, a circuit-level model of hardware elements accurately predict the energy consumption, however these models even have a high process overhead that build them impractical for query optimisation. On the opposite hand, a better level model that treats the complete system as a recording machine, though easy and transportable. In our models, the ability drawn by totally different hardware elements area unit abstracted into learn-able system parameters that area unit combined with a straightforward operator model. Figure 2 offers an summary of the energy price model that we've designed.

The operator model takes as input the question set up and uses the database statistics to estimate the question parameters that's needed by the hardware abstraction model to estimate the energy price. The hardware abstraction model that we have a tendency to describe below needed estimations of parameters from the operator model: the quantity of hardware directions, the quantity of memory accesses, the quantity of disk scan and write requests anticipated throughout question execution. Our operator model provided estimates for these four query parameters for 3 basic operations: choice,

International Journal of Research and Development in Applied Science and Engineering (IJRDASE)

projection, and joins. The hardware abstraction model then uses these four question parameters and also the latency model (since latency relies on system settings) to estimate the energy price of evaluating {a question|a question |a question} employing a specific query set up at a specific power/performance system setting, primarily computing the energy rejoinder profile .

Table 1: Both queries have the template (SELECT * FROM T1, T2 WHERE <predicate>). These queries are used for the ERP plotted in Figure 4.15. All relations are modified (100 byte tuples)

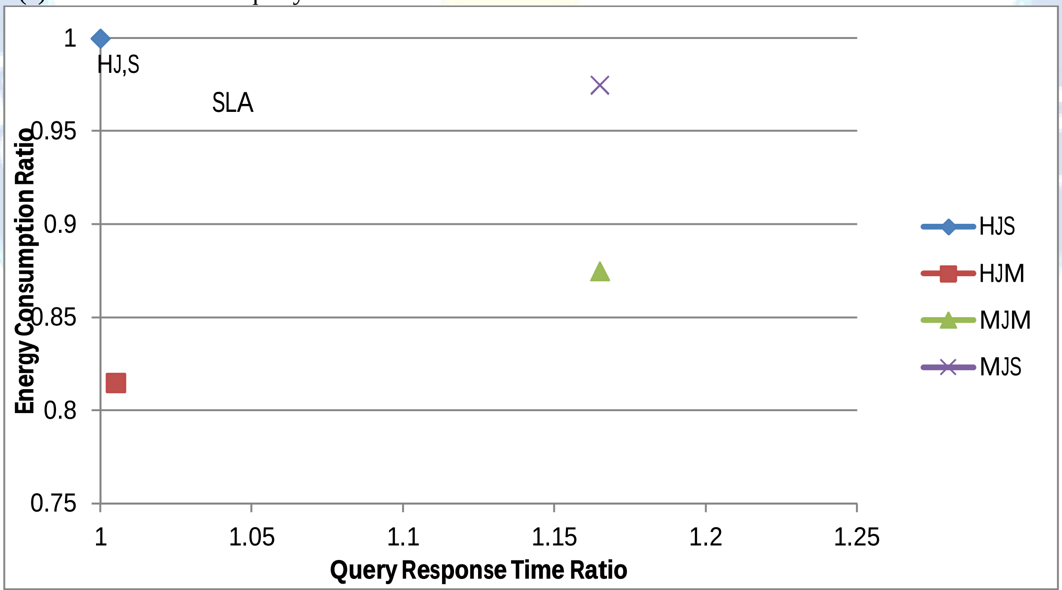
Query	Predicate	Size of T1, T2
A	T1.unique< 0.1*‖T1‖AND T1.unique1=T2.unique2	1GB,1GB
B	T1.unique2=T2.unique2	5GB,5GB

5. Experimental Results: Joins Effectiveness tested on ERP

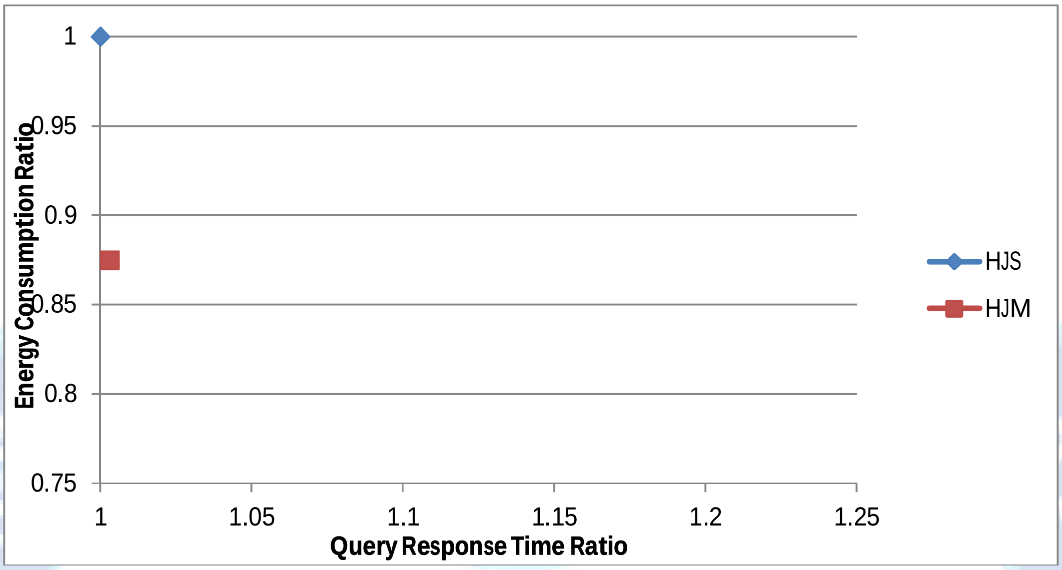
We now present end-to-end results using the techniques that we've proposed in this paper. We have a tendency to use the 2 system settings delineated within the starting of this Paper; particularly, (1) S – the default stock settings of our system, and (2) M – a reduced memory setting wherever the memory is reduced from 4GB to 2GB. Here we present results with the 2 queries shown in Table 1. each queries are join queries on two changed tables, where the tuple length has been reduced to 100 bytes. question A may be a 100 pc selectivity join on 2 small 1GB tables whereas query B may be a full join on the sorted keys of 2 5GB tables.

First, allow us to examine the scenario when switching to a lower power/performance state has little effect on the reaction time. With solely 2GB of memory, we expect that a query whose peak main memory requirement is a smaller amount than 2GB will take approximately constant amount of your time to execute, and thence can provide significant energy savings. Figure 3 (a) shows the ERP of query A in Table 1. As we are able to see, retaining constant hash join plan but using system setting 'M' reduces the energy consumption by 18% but increases the query reaction time by only one. compared, changing the join algorithm to sort-merge incurs significantly higher reaction time penalties.

Query B in Table 1, is an equijoin on two tables that are clustered on the join attribute 'unique2'. the 2 5GB tables are comparatively large compared to the amount of available main memory, but since the tables are already sorted on the join attribute the inputs don't got to be sorted before joining using a merge join operation. query B is I/O-intensive, requires minimal computation, and has a low peak memory requirement. For this question , as shown in Figure 3 (b), using the sort-merge join beside reducing the memory requirement produces a win of 12% energy savings for fewer than 1% performance penalty. The response times for the hash join plans are way larger than sort-merge for query A, and are therefore left out of Figure 3 (b).



(a)



(b)

Fig. 3. Usage of two equijoin query classes in : ERPs: (a) Low memory requirement (b) Low memory and I/O heavy. Two join are used: hash join (HJ) and sort-merge join (MJ); along with 'stock' (S) and low memory (M).

6. Conclusion

This paper presents a replacement framework for energy-efficient query process. The framework augments question plans created by ancient traditional optimizers with associate energy consumption prediction to provide associate Energy Rejoinder Time Profile for a query. These energy rejoinder profile s will then be utilized by the software in varied attention-grabbing ways that, together with finding the foremost energy-efficient question set up that meets sure performance constraints (dictated by SLAs). To modify the framework, a dbms wants associate energy consumption model for queries, and that we have developed a straightforward, portable, practical, and correct model for a very important set of database operations and algorithms. we've got used our framework to reinforce a poster dbms using actual energy measurements, and incontestable that important energy savings area unit possible in some cases. Additionally, the energy consumption at the using level and code level ought to be thought of by software developers within the style of their future application implementations victimization varied energy-efficient techniques planned.

References

[1] Freeh, V. W., Pan, F., Kappiah, N., Lowenthal, D. K., and Springer, R. 2005. Exploring the energy-time trade-off in MPI programs on a power-scalable cluster, *Proceedings of the 19th IEEE International Parallel and Distributed Processing Symposium*, CA, USA. 22

[2] Saxe, E. 2008. Power Efficient Software. *Communication of the ACM*. 53(2) 44-48. 24

[3] Großschädl, Avanzi R.M., Savas E., and Tillich S. 2005. Energy-efficient software implementation of long integer modular arithmetic, *Proceedings of 7th Workshop on Cryptographic Hardware and Embedded Systems*, Edinburg, Scotland. 25

[4] Beloglazov, A, Buyya, R, Lee, YC, and Zomaya, A. 2011. A Taxonomy and Survey of Energy-Efficient Data Centers and Cloud computing Systems, *Advances in Computers*, M. Zelkowitz (ed), ISBN 13: 978-0-12-012141-0, Elsevier, Amsterdam, The Netherlands. 48

[5] Ranganathan P, 2010, Recipe for efficiency: principles of power-aware computing. *Communication. ACM*, 53(4):60–67. 17

[6] Greenberg, S., Mills, E., Tschudi, B., Rumsey, P., and Myatt, B., 2008, Best Practices for Data Centers: Lessons Learned from Benchmarking 22 Data Centers. ACEEE Summer Study on Energy Efficiency in Buildings. Retrieved September 4, 2008, from http://eetd.lbl.gov/emills/PUBS/PDF/ACEEE-datacenters.pdf 18

[7] Smith, J. and Nair, R. 2003. *Virtual Machines: Versatile Platforms for Systems and Processes*. Morgan Kaufmann: Los Altos, CA. 20

[8] Accenture Microsoft Report. 2010. Cloud computing and Sustainability: The Environmental Benefits of Moving to the Cloud, http://www.wspenvironmental.com/media /docs/newsroom/Cloud_computing_and_Sustainability_-_Whitepaper_-_Nov_2010.pdf. 7

[9] R. Ayoub, K. R. Indukuri, and T. S. Rosing. Energy Efficient Proactive Thermal Management in Memory Subsystem. In ISLPED, pages 195–200, 2010.

[10] D. J. DeWitt. TheWisconsin Benchmark: Past, Present, and Future. In J. Gray, editor, The Benchmark Handbook for Database and Transaction Systems (2nd Edition). Morgan Kaufmann, 1993.

[11] M. Poess and R. O. Nambiar. Energy Cost, The Key Challenge of Today's Data Centers: A Power Consumption Analysis of TPC-C Results. In VLDB, pages 1229–1240, 2008.

[12] Transaction Processing Council. http://www.tpc.org/tpc_energy/default.asp.

[13] D. Tsirogiannis, S. Harizopoulos, and M. A. Shah. Analyzing the energy efficiency of a database server. In SIGMOD, pages 231–242, 2010.

[14] H. David, C. Fallin, E. Gorbatov, U. R. Hanebutte, and O. Mutlu. Memory Power Management via Dynamic Voltage/Frequency Scaling. In ICAC, pages 31–40, 2011.