True Energy-efficient Data Processing is Only Gained by Energy-proportional DBMSs

Volker Hudlet
AG DBIS
TU Kaiserslautern, Germany
hudlet@cs.uni-kl.de

ABSTRACT

As energy consumption and related costs are becoming a critical component for operating a data center, system developers as well as database researcher have to deal with this fact and should come up with approaches that increase the energy efficiency of a data center.

Several proposal are already present in the literature which introduce approaches to increase the energy efficiency in a given situation. Nevertheless, a server may still consume more than 50% of its maximal power when running in idle mode. Therefore, we believe that only energy-proportional systems can deliver true energy efficiency, as the power consumption scales with the system load. This paper reviews the current state of research concerning energy efficiency in DB servers and presents our vision of an energy-proportional database system.

1. INTRODUCTION

Energy costs are a growing part of the total cost of ownership of servers, and it is expected that they (as well as the costs for cooling) will outpace the expenses for the server hardware and software in the near future (calculated over a period of three years) [4, 9].

In general, a server is constructed for an expected peak load, i.e., maximal throughput, which is limited by the storage subsystem (in case of data-intensive applications) or by the CPU (in the case of computation-intensive applications). Normally, the peak load corresponds with the maximal energy consumption. In the majority of application situations, this maximal throughput is hardly needed, because the server just utilizes a (small) share of its capacity; the average server utilization often is around 10% – 30% [16]. The remaining capacity is unused, while the server is still consuming (almost the full amount of) energy.

Given the public concern about energy waste, the exclusive focus on performance, where over-proportional energy consumption was acceptable even for only a tiny increase in performance, is not sufficient anymore for future generations of computer systems in general and DB servers in particular. Therefore, attention must be shifted from a solely performance-centric view to energy efficiency.

GvD Workshop'10, 25.-28.05.2010, Bad Helmstedt, Germany.

The buzzword *Green IT* is an umbrella term for the ongoing development of energy-efficient hardware and software as well as the marketing of resulting products. Unfortunately, there are products claiming to be green, however, they just pick up this buzzword to be more attractive on the market.

The remaining parts of this paper are structured as follows: The following Section 2 will briefly define energy efficiency and will discuss why energy proportionality is a natural prerequisite for a true energy-efficient system. Furthermore, related work is considered. Section 3 will explain in more detail how energy proportionality can be achieved for (most of the) system components, whereas Section 4 will disclose our vision of an energy-proportional database system which we are striving for. Finally, we will conclude this paper and give an outlook to future work.

2. ENERGY EFFICIENCY REVISITED

In general, energy efficiency is defined as the quotient of the system's work and the energy consumed while performing this work:

$$EnergyEfficiency = \frac{Work}{EnergyConsumption}$$

This generic model can be adapted to more concrete scenarios such as, in our case, applications of database systems. The following measure can be used to indicate the energy efficiency of a database system.

$$EnergyEfficiency_{(DBS)} = \frac{\#Transactions}{Joule}$$

Note, depending on the transaction mix (varying numbers of long-running and short-running transactions), this measure can be misleading. Meaningful results can only be achieved by using well-defined benchmarks.

In the literature, several ideas have come up to improve energy efficiency. One of such proposals advocates to replace the hard disks of the storage subsystem by flash disks or solid state disks (SSD). While consuming significantly less energy (about 1/10 of the energy a hard disk consumes), SSDs nevertheless deliver substantially higher IOPS rates than hard

disks (at least when read performance is compared). Therefore, SSDs are a natural candidate for achieving better energy efficiency. Until the recent past, SSD technology was still in its infancy and had to struggle with an unbalanced read/write asymmetry: *random reads* were much faster compared with those on hard disks, whereas *random writes* were much slower (approx. ten times of random read access)) and provided only limited write endurance, i.e., the underlying flash cells wore out and became unusable after a given number of rewrites. In the meantime, these disadvantages are almost eliminated. The Intel X-25E claims to be capable of performing one Petabyte of random writes (on a 32GB device) before wearing out[1]. Based on dedicated IO experiments on selected hard disks (HDDs) and SSDs, we come to the conclusion that the asymmetry becomes negligible for SSDs of the newest generation: We have confirmed the performance of random reads at $\sim$ 13K IOPS, while the random-write performance scores at respectable 10K IOPS. Hence, we expect that SSDs will approach the sequential IO behavior of hard disks but, at the same time, provide dramatically better random IO.

Härder et al. [8] analyzed the impact of the replacement of HDDs with SSDs in a database systems. They compare the energy efficiency in XTC [10], a native XML DBMS, by running a selected subset of the TPoX [15] benchmark. The results gained in these experiments show a slight increase of energy efficiency for CPU-bound DB applications (0,176 TA/Joule vs. 0,166 TA/Joule), whereas more than a doubling was obtained for IO-intensive DB applications, i.e., for an IO-bound DB server (0,850 TA/Joule vs. 0,368 TA/Joule). Hence, it is obvious that differing load situations may imply entirely different energy-efficiency levels. But this is not the desirable behavior of a DB server.

One could argue that switching the server completely off would be the most energy-efficient alternative, but again this is just another energy-efficiency level (namely the point of origin) and the cost of resuming operation could not be neglected, e.g., loading the DB buffer anew.

The approach mentioned above is hindered by the fact that the capacity costs (GB/$) for SSDs still exceed the ones for HDDs by at least a factor of 10. Although analysts forecast a considerable price drop within the next two years [11], at the moment, SSDs might still be unattractive for a large data center.

To overcome this drawback, hybrid approaches, like those described in [12] or [13], have been proposed. These ideas combine the use of SSDs and hard disks and thereby allow to benefit from the advantages of both storage types while still having a cost-effective storage subsystem. Right now, these approaches just focus on the combination of several heterogeneous storage types for maximum performance. But it is conceivable to come up with a hybrid storage subsystem which focuses on the energy-efficiency aspect as well.

[1]Using 3.3K IOPS of random 4KB writes—the maximum random-write speed specified by the manufacturer—, a maximum write endurance of $>\sim 8 * 10^7$ sec is obtained. This is close to three years, approximately the lifetime of a hard disk.

Apart from the storage subsystem, dedicated proposals aim at energy-efficient usage of the CPU. To evaluate the benefits gained from energy-efficient approaches, the *energy delay product* (EDP) [6] has been proposed as a reasonable measure. This factor is defined as $energy \cdot delay$: for a constant EDP, the change in the energy consumed is therefore matched by an equal change in the response time. Lower EDP values are, of course, desirable as they embody a larger percentage of energy saving. In this case, however, system response time is likely to be increased, which may not be wanted by the user.

In contribution [14], Lang and Patel propose two techniques which are evaluated towards their resulting EDP. The first technique, called *explicit query delay*, delays queries and places them into a queue upon arrival. When the queue reaches a given threshold, all queries in the queue are examined to determine whether or not they can be aggregated into a small number of groups, such that the queries of a group can be jointly evaluated. Hence, this approach tries to minimize redundant evaluation of queries thereby saving energy. It has shown that, using a simplistic scenario, this kind of grouping could decrease the EDP by 26%.

Besides this technique, it is possible to influence the CPU behavior and thereby its energy consumption by *processor voltage/frequency control* (PVC) techniques, e.g., by underclocking the *front-side bus* or by downgrading the CPU voltage. Again, PVC techniques embody a static approach which could leverage the energy efficiency only at a certain load level, but which could eventually also impinge upon the query execution time and imply higher energy consumption than the default setting. Thus, in general, it is highly desirable to dynamically adjust the server's energy consumption such that the best possible energy efficiency is accomplished at all load levels.

This is an objective where energy-proportional systems come into play. The notion of energy proportionality has been first coined by Barroso and Hölzle [3] and characterizes the behavior of a server whose energy consumption proportionally scales with its load. An adaptive PVC would be an initial step towards this design goal. Nevertheless, the entire system architecture should be reconsidered, because building energy-proportional systems requires a holistic approach. Ranganathan [16] comes to the same conclusion that instead of having several small and local energy-aware optimizations, a holistic focus supposably results in an even better energy-efficient system.

Recently, Tsirogiannis et al. [20] claimed that, within a single node system (intended for use in scale-out architectures), *the most energy-efficient configuration is typically the highest performing one*. Obviously, their empirical "observation" is also closely dependent on the absence of energy-proportional runtime behavior in current servers. Furthermore, the authors hypothesize that better saving opportunities might be found when cross-node, energy-efficiency techniques are to be applied.

In the following section, the key components of a server are examined towards their ability to reach energy-proportional behavior.

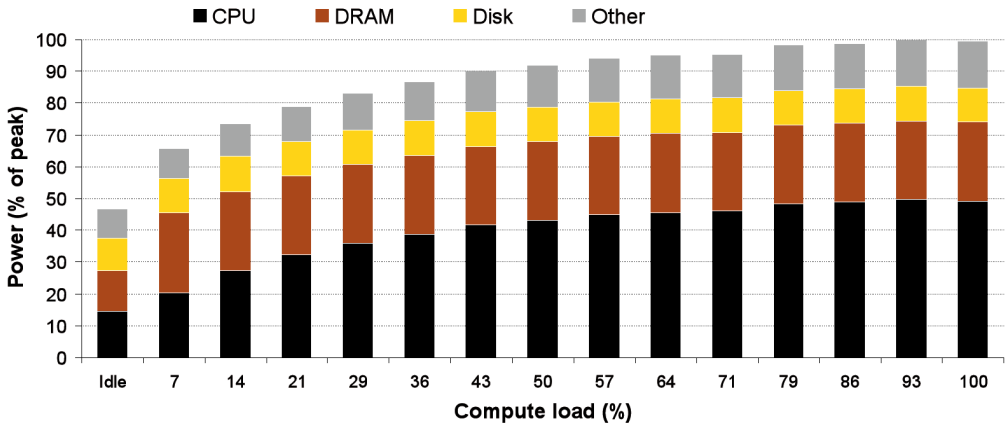

Figure 1: Relative power consumption of a server at different activity levels derived by Google [18]

3. ENERGY PROPORTIONALITY OF A DB SERVER

Before we come up with a proposal how an energy-proportional system should be preferably composed, it makes sense to examine existing (DB) server systems to find out how energy proportionality can be achieved.

When considering a server as a whole, Spector [18] as well as Tsirogiannis et al. [20] come to the conclusion that a normal server consumes already more than 50% of its maximal power (and much more especially, when a huge memory is present) when running in idle mode. Figure 1 illustrates how the power consumption looks like at different load situations. It is remarkable that the power consumption (starting already at 50%) quickly converges with a small increase of utilization close to the peak consumption, i.e., the 100% level. Obviously, a server in its default settings does not exhibit an energy-proportional behavior at all. For these reasons, a closer look at the key components will be helpful.

Storage In contribution [20], experiments using hard disk RAIDs and SSD RAIDs show that, unlike hard disks, SSDs provide an energy-proportional behavior. We also performed some load test using a selected set of hard disks and SSDs of different generations (cf. figure 2), but we draw another conclusion: SSDs just have a slightly better energy-proportional behavior, yet at a much lower power level (1/10 of that of hard disks).

In the recent past, several approaches have been proposed for hard-disk-based storage subsystems, which spin down idle disks in order to save energy [5, 21, 22]. Depending on the respective approach, data is relocated during run time in order to increase the idle time of a disk that is already spun down. Otherwise, there is a time penalty to spin up the disk again. As an overall effect, energy-proportional behavior can be approximated [7]. In order to further decrease the power consumption, these approaches could be adapted to hybrid or SSD-only storage subsystems.

CPU Modern CPUs behave in an energy-proportional way to some degree. In addition to the control via PVC techniques, the current trend towards many-core processors favors energy-efficient operation. It is possible that unused cores enter a *sleep mode* where they just consume a fraction of the power needed in *idle mode*. The Intel Core i7 processor combines both techniques by disabling unused cores (especially in the case of single-threaded applications) and by increasing the clock rate of the remaining one. Finally, there are also low-energy processors (e.g. Intel Atom) available.

DRAM memory Main memory is the primary concern when thinking about energy proportionality. As it permanently consumes a given amount of power (independent of the load), this component is not energy-aware at all. One current trend is to build large (in the range of Terabyte) main-memory databases[2]. This will result in just the opposite of an energy-proportional system as RAM will be responsible for the overwhelming share of the energy consumed by the server—at a constant rate.

Therefore, it is critical to evaluate how much internal memory is needed to approximate energy proportionality without sacrificing drops in performance by utilizing an insufficient amount of memory.

In a nutshell, the previous methodology using large-scale servers (scale-up) is still burdened by large energy consumption in idle mode.

Another possibility for system engineering is scale-down / scale-out: Instead of using a single, large server, several small-scale servers are deployed. In the literature, there have been proposals for such a network of small-scale servers like *Amdahl blades* [19], FAWN (Fast array of wimpy nodes) [1] or TerraServer bricks [2]. As each server is independent, this is the appropriate granularity for scaling the whole system as well as the appropriate granularity to switch nodes on and off. In the end, this will result in a true energy-proportional system (to the extent possible).

[2]"SAP-Module gewinnen an Tempo" (Computer-Zeitung, June 22, 2009). Using main-memory data management, SAP tries to speed up the response times of applications by a factor of 100.

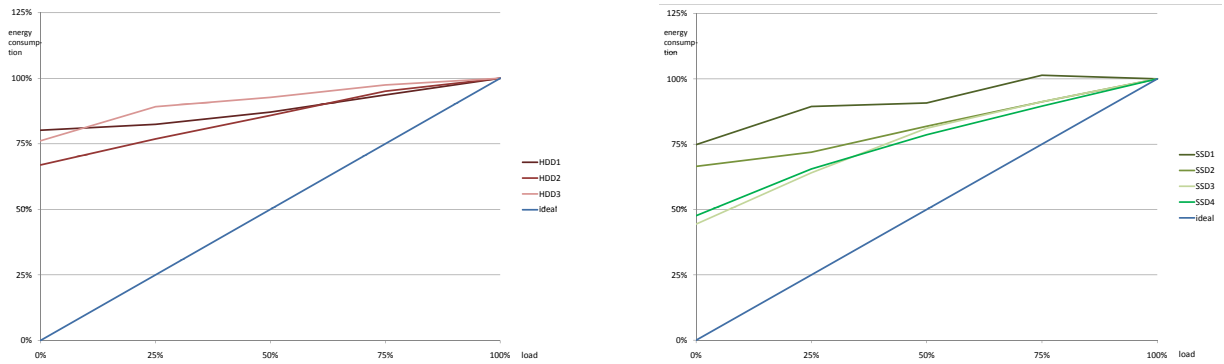

Figure 2: Energy proportionality of selected hard disks and SSDs

In the next section, we will explain our vision of an energy-proportional system in more detail.

4. OUR VISION

As it has become obvious in the preceding section that a single (large-scale) server node can't establish an energy-proportional behavior, we will focus on the scale-out approach consisting of several small-scale nodes connected via network adapters.

We envision a distributed database system which runs on several small-scale nodes. While FAWN tackles a distributed key-value store, we will focus on a traditional relational database system. Although much research work on distributed database systems has delivered substantial scientific results and engineering techniques during more than 20 years, it is nevertheless fundamental to reevaluate this "body of knowledge and experience" with respect to modern hardware and energy efficiency.

As every node is constructed in a small-scale manner and, thus, consumes little energy, we have at least energy proportionality at the granularity of nodes. Depending on the load situation, nodes can be switched on and off, so this approach will approximate the ideal energy-proportional system. We believe that small-scale distributed systems are the key concept to achieve energy proportionality. By applying ad-hoc adaptivity mechanisms, energy consumption will scale with the given load.

At the moment, we are about to implement a first software prototype in the context of the SIGMOD 2010 programming contest [17] whose goal is set to come up with a distributed database engine. After having finished the contest, we will expand its functionality towards adaptivity and energy efficiency.

For the future, we consider an architecture which comprises two types of specialized nodes: *Data nodes* for accessing the base relations and performing simple operations (e.g. selection and projection) and *computation nodes* for CPU-intensive operations like joins. Of course, there are many open and challenging questions while refining this approach, amongst others to find out how the overall energy efficiency is affected by the data distribution or how to come up with an energy-efficient query optimizer for distributed systems.

Another issue that needs further investigation is how much energy consumption is introduced by the network infrastructure and the data transmission between nodes transfer and whether it proportionally scales with respect to the load as well.

5. CONCLUSION

As we have shown, the current trend towards energy efficiency and Green IT is relevant for the database research community as well. Several ideas of limited scope have already been proposed; nevertheless, we believe that only a holistic approach will be the road to success in the end.

Present approaches try to be energy-efficient under high workloads or even peak load situation (e.g., explicit query delays). Our approach aims especially at increasing the energy efficiency at low load levels by introducing the concept of energy proportionality.

Furthermore, we want to provide some evidence whether or not the claims of Tsirogiannis et al. [20] are *true*, i.e., whether our findings will support their hypothesis.

In the future, we will further explore how (distributed) database systems have to be designed to exploit the given system architecture best. By introducing adaptivity, the database system will dynamically interact with its underlying hardware to increase energy efficiency.

6. REFERENCES

[1] D. G. Andersen, J. Franklin, M. Kaminsky, A. Phanishayee, L. Tan, and V. Vasudevan. FAWN: a fast array of wimpy nodes. In *SOSP*, pages 1–14, 2009.

[2] T. Barclay, J. Gray, and W. Chong. TerraServer Bricks – A High Availability Cluster Alternative, Microsoft Research (MSR-TR-2004-107). Technical report, 2004.

[3] L. A. Barroso and U. Hölzle. The Case for Energy-Proportional Computing. *Computer*, 40(12):33–37, 2007.

[4] C. L. Belady. In the Data Center, Power and Cooling Costs More Than the IT Equipment it Supports. Electronics Cooling. vol. 13, no. 1; http://electronics-cooling.com/articles/2007/feb/a3/, 2007.

[5] D. Colarelli and D. Grunwald. Massive arrays of idle disks for storage archives. In *ACM/IEEE conference*

on Supercomputing, pages 1–11. IEEE Computer Society Press, 2002.

[6] V. De and S. Borkar. Technology and design challenges for low power and high performance. In *International Symposium on Low power electronics and design*, pages 163–168, 1999.

[7] J. Guerra, W. Belluomini, J. Glider, K. Gupta, and H. Pucha. Energy proportionality for storage: impact and feasibility. *SIGOPS Operation Systems Review*, 44(1):35–39, 2010.

[8] T. Härder, K. Schmidt, Y. Ou, and S. Bächle. Towards Flash Disk Use in Databases - Keeping Performance While Saving Energy? In *BTW*, volume P-144 of *LNI*, pages 167–186, 3 2009.

[9] S. Harizopoulos, M. A. Shah, J. Meza, and P. Ranganathan. Energy Efficiency: The New Holy Grail of Data Management Systems Research. In *CIDR*, 2009.

[10] M. P. Haustein and T. Härder. An Efficient Infrastructure for Native Transactional XML Processing. *Data&Knowledge Engineering*, 61(3):500–523, 6 2007.

[11] J. Janukowicz, D. Reinsel, and J. Rydning. Worldwide Solid State Drive 2008-2012 Forecast and Analysis. Technical report, IDC, Juni 2008.

[12] S.-H. Kim, D. Jung, J.-S. Kim, and S. Maeng. HeteroDrive: Reshaping the Storage Access Pattern of OLTP Workload Using SSD. In *International Workshop on Software Support for Portable Storage*, 10 2009.

[13] I. Koltsidas and S. Viglas. Flashing up the storage layer. *PVLDB*, 1(1):514–525, 2008.

[14] W. Lang and J. M. Patel. Towards Eco-friendly Database Management Systems. In *CIDR*, 2009.

[15] M. Nicola, I. Kogan, and B. Schiefer. An XML transaction processing benchmark. In *SIGMOD*, pages 937–948, 2007.

[16] P. Ranganathan. Recipe for efficiency: principles of power-aware computing. *Communications of the ACM*, 53(4):60–67, April 2010.

[17] P. Senellart, C. Genzmer, S. Abiteboul, M. Balazinska, S. Madden, and M. Stonebraker. SIGMOD 2010 Programming Contest – Distributed Query Engine. http://dbweb.enst.fr/events/sigmod10contest/, 2010.

[18] A. Z. Spector. Distributed Computing at Multi-dimensional Scale. In *International Middleware Conference*, 2008. Keynote.

[19] A. S. Szalay, G. C. Bell, H. H. Huang, A. Terzis, and A. White. Low-power amdahl-balanced blades for data intensive computing. *SIGOPS Operation Systems Review*, 44(1):71–75, 2010.

[20] D. Tsirogiannis, S. Harizopoulos, and M. Shah. Analyzing the Energy Efficiency of a Database Server. In *SIGMOD*, 2010.

[21] C. Weddle, M. Oldham, J. Qian, A.-I. A. Wang, P. Reiher, and G. Kuenning. PARAID: A gear-shifting power-aware RAID. *ACM TOS*, 3(3):13, 2007.

[22] Q. Zhu, Z. Chen, L. Tan, Y. Zhou, K. Keeton, and J. Wilkes. Hibernator: helping disk arrays sleep through the winter. In *SOSP*, pages 177–190, 2005.