Tracking the Power in an Enterprise Decision Support System

Justin Meza
HP Labs

Mehul A. Shah
HP Labs

Parthasarathy Ranganathan
HP Labs

Mike Fitzner
Hewlett-Packard BCS

Judson Veazey
Hewlett-Packard BCS

ABSTRACT

Enterprises rely on decision support systems to influence critical business choices. At the same time, IT-related power costs are growing and are a key concern for enterprise executives. Yet, there is little work to date characterizing the power use of decision support systems. Towards this end, we present the first holistic measurements and analysis of an audit-class system running the TPC-H decision support benchmark at the 300GB scale. We first provide a breakdown of the system's power use into its core hardware components. We then explore its power-performance tradeoffs. This investigation shows that there is ample room to improve its energy use without sacrificing much performance. Moreover, the most energy-efficient configuration depends on the workload. These results suggest that, going forward, database software has an important role to play in optimizing for energy use.

Categories and Subject Descriptors

H.2 [**Database Management**]: Miscellaneous

General Terms

Measurement, Performance

Keywords

decision support, power, energy, energy efficiency, TPC-H

1. INTRODUCTION

The decision support market is a multi-billion dollar market and is still experiencing double-digit growth rates [17]. Enterprises use decision support systems to quickly perform complex analyses over large amounts of data whose results are used to inform critical business decisions. As technology improves, these companies are demanding larger, faster, and cheaper systems so they can derive value from data that was too costly to mine in the past.

ISLPED'09, August 19–21, 2009, San Francisco, California, USA.

At the same time, an important and growing component of the total cost of ownership for these systems is power and cooling [1]. A recent report by the EPA shows datacenter power consumption in the US doubled between 2000 and 2006, and will double again in the next five years [16]. Uncontrolled energy use in datacenters also has negative implications on density, scalability, reliability, and the environment.

These trends suggest that we should optimize decision support systems for energy-efficiency, yet there is little work on understanding the power characteristics of these systems from a whole system perspective. In this paper, we present the first power measurements and analyses of a system configured similarly to a performance-optimized, audited TPC-H system at the 300GB scale [3]. Our main contributions are as follows.

We first provide a breakdown of the system's power-use into its core subsystems: CPU, memory, disk, and other miscellaneous components (Section 3). Since decision support workloads often need many disks, the storage subsystem, as expected, used more than half of the total power. Interestingly, the miscellaneous components in aggregate, such as fans, disk array controllers, supporting chip sets, and so on, comprised 27% of total power. But, the memory DIMMs comprised an unexpectedly small fraction, 7% of total power.

We also explore the power-performance tradeoffs that this system exhibits and reveal interesting insights from this investigation. In its peak performing configuration, the hardware (and underlying OS) offered little automatic power reduction between full load and at idle. Instead, we found the most effective way to trade performance for power was by repartitioning the database across fewer disks and turning-off the unused ones. Using this knob for reducing power, we found that the initial performance-optimized configuration is grossly over-provisioned; we reduced power use by 45% while sacrificing only 5% of peak performance. Even at smaller memory sizes, we found that the system's energy efficiency is *non-monotonic* with increasing performance (Section 4) . That is, there is a point of diminishing returns after which performance continues to improve as disks are added but energy efficiency drops. Moreover, we found that this peak efficiency point varies and depends on the query workload.

These results suggest the following. First, industry benchmarks should incorporate energy measurements since systems optimized for performance are poorly balanced for power and, thus, may not reflect current customer needs. More-

over, since the most energy-efficient point depends on both hardware and the query workload, database software, e.g. physical design tools, query optimizers, and so on, have an important role to play in optimizing decision support systems for energy.

2. BACKGROUND AND RELATED WORK

Database benchmarks. The transaction processing council (TPC) is an consortium of database hardware and software vendors. They define industry-standard database benchmarks that reflect real-world customer workloads and needs. TPC offers two main categories of benchmarks: online transaction processing (OLTP) and decision support.

TPC-C and TPC-E are OLTP benchmarks which stress the ACID transaction capabilities of databases and the storage system's ability to handle random I/Os. TPC-E is more modern, reflecting technology ratios (CPU, memory, and disk) seen in current customers deployments.

On the other hand, TPC-H is a decision support benchmark consisting of a suite of business oriented queries. These read-mostly, ad-hoc complex queries stress the query optimization and query processing subsystems of a database. They scan through a large portion of the database combining and summarizing the input. The resulting access patterns stress the sequential I/O capabilities of the storage system.

Currently, all the TPC benchmarks metrics are based on performance and price-performance, but a TPC subcommittee is underway to consider a power-based metric [15].

TPC-H. Our results were gathered from a system which ran the TPC-H benchmark. TPC-H measures how fast a system can processes ad-hoc decision support queries on a synthetic database. There are 22 query templates, and these queries vary in terms of their working set size and runtime. The benchmark consists of two sub-tests. The "Power test" [1], issues these 22 queries back-to-back from a single client. The "Throughput test" issues a query mix simultaneously from multiple clients and more closely resembles a real-world workload. The TPC-H performance score is a composite Queries-per-Hour (QphH), calculated as the geometric mean of the two sub-tests' scores. The individual subtest scores are more complex functions of query and test run times and are defined in the TPC-H specification [15]. Audited runs also include a system's price-performance ($/QphH). Audited results are grouped into "scale factor" categories based on database size which ranges from 100GB to 30TB and ranked by QphH and $/QphH.

System Power. Since power and cooling have been recognized as an important concern, a number of studies have looked at improving the energy use of the various server subsystems: CPU, memory, and disks [4, 5, 8, 9, 19, 20]. Closest to our work, Pinheiro and Bianchini, show that concentrating popular files on a few disks in an array and turning-off the remaining can save energy [9]. Although their approach for power-reduction is similar to ours, their work only considers Web traffic in which file popularity obeys a power law. Unlike our system, their file accesses involve mostly random I/Os, and files are stored over the network rather than on direct-attached arrays.

There is a recent push to build energy-proportional systems whose power use tracks average performance rather than peak [2, 7]. An ideal energy-proportional system uses no power when not used (i.e. delivering no performance) and uses additional power in a perfect linear proportion to delivered performance. Since energy efficiency is the ratio of performance (measured as a rate: work-done/time) to power, such a system exhibits constant energy efficiency at all performance regimes. Unfortunately, current hardware systems are far from ideal; they provide little dynamic range, using significant power when idle. As a result, Tolia et al. [14] try to achieve better proportionality using virtual machine migration across an ensemble of machines. We share this vision and seek novel ways to achieve proportionality in decision support systems through a combination of software and hardware techniques.

More recently, some have proposed power-based benchmarks for specific server workloads. SPEC-Power is a server-side Java benchmark that measures system power at various utilizations [13]. Its workload taxes mainly the CPU and memory subsystems. Rivoire et al. [12] proposed JouleSort which runs an external-memory sort and ranks systems based purely on records sorted per unit energy. This workload stresses all server components including I/O, but does not adequately represent the complexity of decision support queries. For both of these simpler benchmarks, audited systems, to date, have been most energy efficient when configured for optimal performance. That is, they have no practical point of diminishing returns. In contrast, we explore more complex server workloads and reveal that their energy efficiency varies non-monotonically with performance. This results in energy-efficient server configurations that are much different from the best performing.

Poess and Nambiar have analyzed the power use of systems running TPC-C, an online transaction processing benchmark [10]. They also found that the disk subsystem used the most power, but they do not provide a study of power-performance trade-offs for TPC-C.

3. POWER BREAKDOWN

In this section, we show where the power goes in our decision support system. We describe our measurement and estimation methodology, present the power breakdown into the core system components, and present a simple power model for the system.

3.1 The TPC-H System

Although our results are not audited, our system was configured similarly to an audited TPC-H benchmark result at the 300GB scale [3]. Our system was setup, tuned, and measured with the help of TPC-H performance benchmarking experts. The system consisted of an HP ProLiant DL785 server tray with 8 Quad-Core AMD Opteron processors (8360 QC 2.5GHz), 256GB of memory (PC2-6400 DDR SDRAM), and 204 SCSI drives (15K RPM, 73GB) connected by SAS to 13 HP StorageWorks MSA70 disk trays. We ran a commercial database system also configured similarly to an audited system [3] and ran on Microsoft Windows Enterprise Server 2008. We striped the database across all disks in a RAID 5 configuration and the database was compressed, thereby fitting into 256GB of memory. Through

[1]The term "Power test" is confusing when discussing electric power use. Therefore, we capitalize "Power" when referring to the TPC-H specific test.

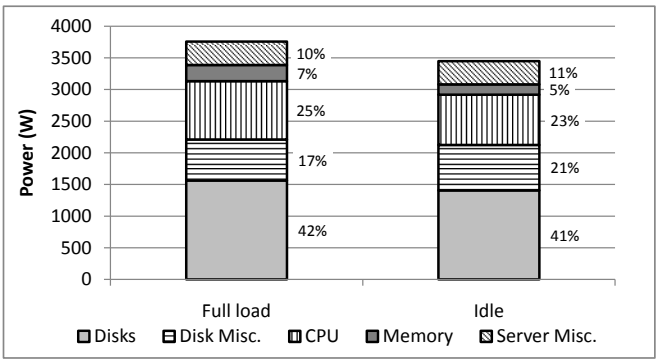

Figure 1: System power breakdown while running TPC-H and at idle.

the BIOS, we set the processors to the fastest, static power state setting, so at full load, they were in the P0 state. When idle, the system simply ran the system idle process. These settings were the same for all the experiments.

3.2 Measuring Power

We logged both power and performance data when running the TPC-H benchmark on our system. We used two "WattsUp?" Pro ES power meters to measure and log the power use. They are rated at 120V, 15A, 60Hz with an accuracy of +/- 1.5% [18]. One meter was connected to the server tray and another was connected to one of the disk trays. Since the Throughput tests took about an hour, we set the meter to collect average power readings at 1 minute intervals. For the Power tests, the meter collected average power readings at one second intervals. We used Microsoft's Performance Monitor (PerfMon) to collect server performance data (CPU utilization, disk I/Os, etc.).

We measured the power of several different disk trays during our experiments, including ones holding the database log and ones holding tables and temporary space. We saw negligible variation in power across these trays for various runs at full load. Thus, we estimated total disk subsystem power throughout by measuring one disk tray and multiplying by the number used. We compute energy as a product of average power over the run and total elapsed time.

3.3 Component Power Breakdown

We first present the TPC-H ratings and power consumption of our system setup designed to achieve the best performance (in QphH). Our system had an estimated TPC-H Throughput@300GB of 43972.9 and an estimated TPC-H Power@300GB of 64213.9 giving an estimated TPC-H composite (Queries-per-Hour) QphH@300GB of 53138.2. The total cost of our system hardware and software was $195,833 resulting in an estimated TPC-H Price-per-QphH@300GB was $3.68. In addition, our system used 3780 Watts (W) on average during the run and 3366 W on average while idle, a dynamic power range of only 12% – far from ideal. Our measurement setup directly gave us a system power breakdown into its two main sub-systems: server and disks, which comprised 58% and 42% of the total, respectively.

To obtain a more detailed breakdown, we ran multiple experiments in different configurations varying the number of components of each type: CPU, DIMM, and disk. We physically removed these components and measured system power using our power meters. In the case of disks, we manually repartitioned the database across all the remaining disks, which often took hours. For simplicity, we kept the tuning parameters for the database software the same. These changes allowed us to estimate the incremental power contribution of each additional CPU, DIMM, and disk. We ran the same procedure for both the system under load running the benchmark and at idle. These experiments also illuminated power-performance characteristics which we analyze in the next section. But first, we present the power breakdown under load.

We first removed disks from disk trays and measured the tray power with fewer disks. We took measurements with trays holding 12 and 18 disks and assumed power was a linear function of number of disks. This gives 7.7 W per disk and 50 W of miscellaneous disk tray "overhead", i.e. power estimated at zero disks. This overhead includes power supply overheads, fans, and other chipsets in the tray. These values were similar whether we used one or both redundant power supplies on the disk tray. Across 13 trays, this overhead is 648 W or 17% of total power.

We did the same for memory and processors. We removed DIMMs in increments of 64GB from the system and found that each 4GB DIMM used about 4W. We could only remove half of the CPUs from our server giving us measurements at 4 and 8 CPUs. We found each CPU used about 116W. We estimated the overhead of miscellaneous server tray components, i.e. with neither memory nor processors, by taking the baseline tray power and subtracting the appropriate contribution of power from each of the components. This gave us 373 W for the server's miscellaneous components or 10% of the total system power.

Figure 1 shows the detailed power breakdown for our system running the TPC-H workload and at idle, respectively. The system was configured for peak performance, as described above. There are some important and some unexpected effects to note.

- The system offered little dynamic range in power. We did not scale CPU power. Even if we had, it would have only saved at most another 21% at idle, and only a little at full load since the benchmark was CPU bound. Moreover, we could not easily scale the power of the other components without removing them or turning them off entirely. Thus, there is little difference in the proportions for each of the subsystems.

- Memory drew relatively little power compared to the rest. Recent studies on server systems and workloads have pointed to the increasing contribution of the memory subsystem to the system power [11]. Although our breakdown only reflects the power from DIMMs, it is still dwarfed by the other components, in particular the CPUs and the disk subsystem.

- The disks consumed the largest fraction, as expected, since there are many of them. Thus, they seem to be most effective control point for optimizing energy use in our system.

- The miscellaneous components, which include fans, array controllers, power supply inefficiencies, chipsets and so on, come in second. Miscellaneous components

on both the disk and server trays account for 27% of total power and cannot even be removed or turned off incrementally. This large fraction suggests that we need further research into how to scale the power use of these miscellaneous components along with the core components to achieve true "energy proportionality."

3.4 Modeling the System's Power

In order to build tools that optimize for energy efficiency, we need models that can predict system power consumption. As a start, we present a simple, but useful contribution: a linear power model of our system under full load running TPC-H and at idle. We derive this model directly from the component variation experiments above. Equation 1 models power use under full load while Equation 2 models power use at idle:

$$7.7\delta_d + 49.8\lceil\frac{\delta_d}{25}\rceil + 115.5\delta_p + 3.9\delta_m + 373.1 \quad (1)$$

$$6.9\delta_d + 55.2\lceil\frac{\delta_d}{25}\rceil + 99.5\delta_p + 2.5\delta_m + 367.0 \quad (2)$$

where δ_d, δ_p, and δ_m represent the number of disks, processors, and DIMMs present in our system, respectively. The model assumes that each disk tray is filled to capacity (25 disks for us) before another is used, manifested in the second term that accounts for disk tray overheads. The last scalar term represents the power contribution of the server tray overheads. We compared the model to actual power measurements for other configurations of our system running TPC-H and saw less than 3% error in all cases. Although not universal, this model works for our system setting because its components offer little dynamic power range.

4. POWER VS. PERFORMANCE

In this section, we illustrate the power and performance trade-offs in our decision support system by varying the system components. For experiments in this section, we report actual measurements of average power consumption, not estimates. We first show that the most effective means of achieving energy-efficiency is by repartitioning our database across fewer disks. This knob effectively trades performance for power and shows that the initial performance-optimized configuration is grossly over-provisioned. We can achieve near peak performance while reducing system power by 45%. We then use more a modest memory size (64GB), and show that the system's energy-efficiency profile is non-monotonic. Unlike other server benchmarks (see Section 2), there is a point of diminishing returns, and the most energy-efficient configuration provides less than peak performance. We also show that the peak efficiency point depends on workload. We then discuss the interesting opportunities for database software that these observations suggest.

Varying Memory and CPU. We start by varying the amount of memory to affect both power use and performance. The purpose of this experiment is to illustrate the energy-efficiency profile of a system that has poor energy proportionality. Figure 2 shows how power and performance vary as we change the the amount of memory in the system from 64GB to 256GB when running the Throughput test on 32 cores and across 204 disks. To clearly show the relationship between power and performance, we use the test's

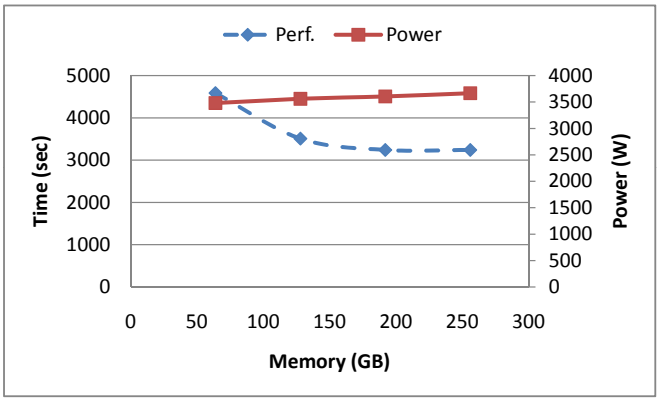

Figure 2: Power-Time vs. Memory Size. Throughput test, 32 cores, 204 disks

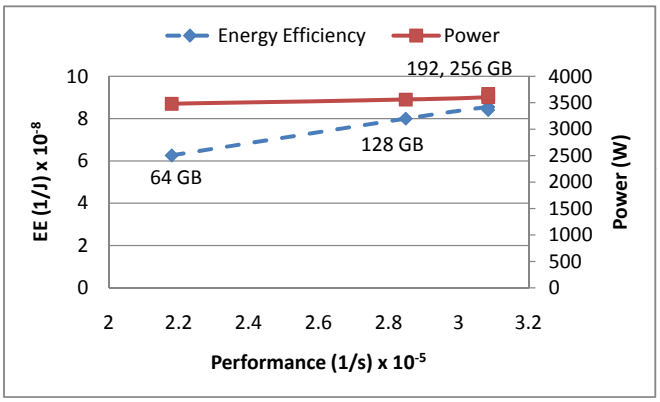

Figure 3: Energy-efficiency vs. Memory Size. Throughput test, 32 cores, 204 disks

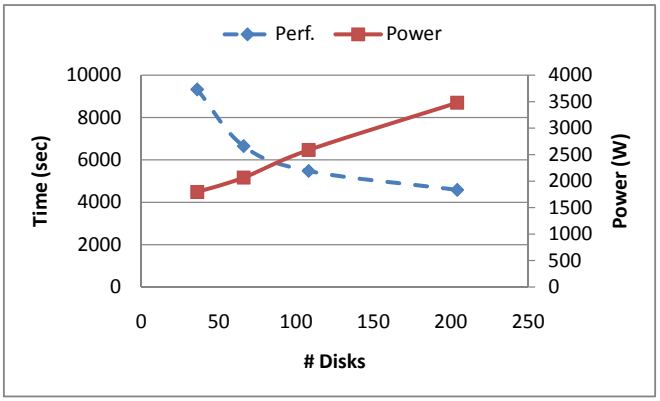

Figure 4: Power-Time vs. Disks. Throughput test, 32 cores, 64GB

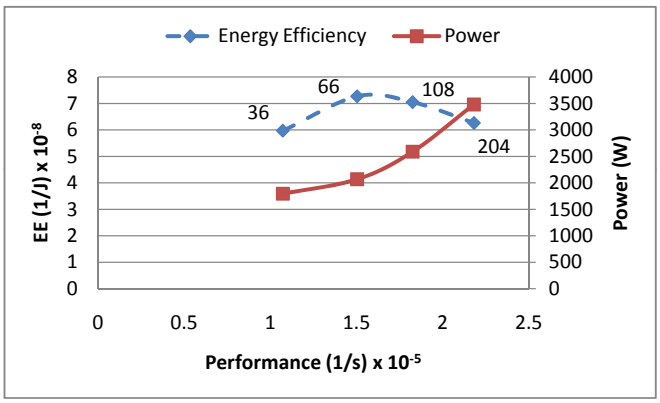

Figure 5: Energy-efficiency vs. Disks. Throughput test, 32 cores, 64GB

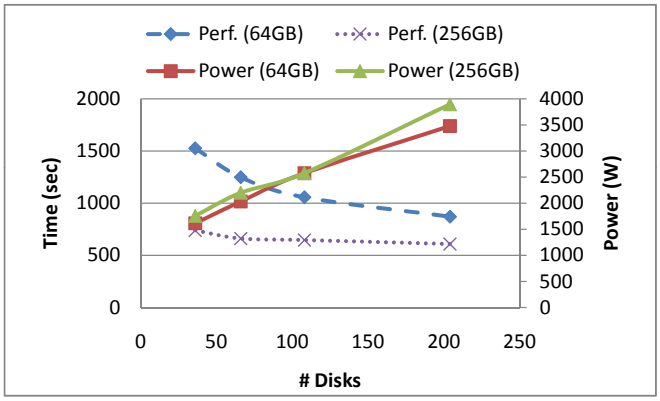

Figure 6: Power-Time vs. Disks. Power test, 32 cores

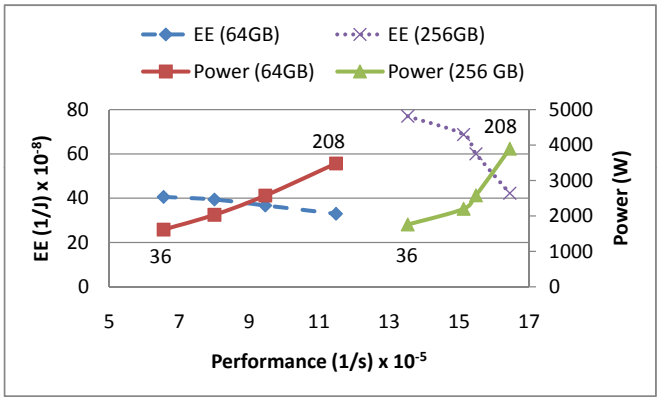

Figure 7: Energy-efficiency vs. Disks. Power test, 32 cores

elapsed time as the performance metric rather than a complicated QphH metric. At higher sizes (>= 256GB), the database fit entirely into memory, so the system was CPU bound and adding more memory did not improve performance. At lower memory sizes (< 192GB) the compressed database did not fit into memory, making the system more I/O bound and reducing performance. Since memory power accounted for only a small fraction total power, reducing memory resulted in a small, linear drops in power with relatively large drops in performance (or increase in time).

To measure energy-efficiency, we need to examine the ratio of those two curves. Figure 3 plots a parametric curve using the same data, i.e. both the x and y axis are dependent on the memory size. Performance is on the x-axis (note the origin is not at 0) and power and energy-efficiency are on the y-axis. In this case, performance is measured as a rate: inverse of the time taken to complete the test. Since energy-efficiency is the ratio of performance to power, its units are 1/Joules. This plot shows two effects when only varying memory size. First, the system power-performance profile is linear but not "energy-proportional" since power use at low performance is still significant. Second, the system's most energy-efficient point is also its best performing.

We observed a similar effect when varying the number of CPUs at 256GB for the Throughput test (not shown). In this case, we saw lower power use as we removed CPUs, but performance was significantly degraded because the system is CPU-bound even with 8 CPUs (32 cores).

Varying Disks. Repartitioning the database to use fewer disks, however, produced a dramatically different result. We varied the number of disks from 36 to 204 for the Throughput test at 256GB (not shown). The difference in the best and worst performance in this range was small, about 4.7%, but the drop in system power was substantial, about 45%. To meet a publication goal for a TPC-H performance benchmark, the disks are needed but grossly over-provisioned for typical deployments. The most energy-efficient configuration was at 36 disks, for an overall efficiency improvement of 91% from the peak performance point.

Next, we reduced the memory to 64GB to reflect a more typical case in which the database size is larger than memory size. Figure 4 illustrates how power and performance vary with the number of disks for the Throughput test at 64GB. Unlike when we varied memory, we see both significant power savings as we remove disks and significant performance improvements as we add more disks.

Figure 5 plots the energy-efficiency curve for this data using the same parametric axes as before. In this case, the power-performance curve is *non-linear* as we repartition across various numbers of disks. Moreover, the energy-efficiency curve is non-monotonic; it peaks before the system reaches the best performing configuration. At 64GB, the dataset cannot be cached entirely, so with few disks (< 66), adding disks is worth it. System power increases by a fraction but the additional I/O capacity increases performance by a larger fraction. With more disks (> 66) the fractional increase in performance does not outweigh the fractional increase in power. In this experiment, the most efficient point offers a 40% drop in power and 31% drop in performance, for an overall 14% increase in energy efficiency.

Finally, Figures 6 and 7 are similar plots that show the power and performance tradeoffs as we vary disks for the

Power test with 64GB and 256GB of RAM. Like earlier, these plots also show that there is a point of diminishing returns: adding more disks (> 36) improves performance but not enough to warrant the increase in power consumed. Interestingly, the peak efficiency point for the Power test at 64GB is different than for the Throughput test at 64GB. This effect is a result of the difference in workload. The Power test executes queries back to back rather than simultaneously processing a query mix as the Throughput test does. In the Power test a few queries need the disk bandwidth offered by additional disks, but most do not. Since the system resources are not shared across queries in this experiment, the additional disks are not worth their power cost. For the Power tests, at 256GB, the most efficient configuration offers an 82% increase in efficiency for only a 22% drop in performance. At 64GB, the most efficient point offers only a 22% increase in efficiency for a 43% drop in performance.

4.1 Implications

We showed that the most energy-efficient configurations for decision support systems depend not only on the hardware but also the workload, and finding these configurations is non-trivial. Using more power-elastic hardware will help, but will not solve the problem since perfectly proportional hardware is hard to find. Thus, going forward, we believe database software will have a new role to play in helping customers optimize not only for performance and cost but also for power. An immediate next step from this work is to incorporate energy constraints and power models into physical design tools. But, in general, all aspects of database systems, such as query optimization, buffer management, logging and recovery, and so on, will need to be re-thought with energy in mind. Harizopoulos et al. offer a taxonomy of general approaches for tackling this redesign [6].

5. CONCLUSION

In this paper, we investigated the energy-efficiency characteristics of a performance-optimized, audit-class TPC-H system. We provided a breakdown of power consumption into its core components: CPU, memory, and disks. We found that the miscellaneous support components consume a large fraction of the total power, so, going forward, we will need mechanisms to scale their power along with the other components. We also explored the power-performance tradeoffs of this system by physically adding and removing components. We found that the best way to improve efficiency is to repartition the database across fewer disks. Using this knob, we found that our system has a point of diminishing returns for energy efficiency, and this point depends on the workload. This suggests that database software has an important role to play in optimizing for energy.

As a start, we believe that industrial benchmarks should adopt power consumption as a reported metric. Currently, peak performance is used as a design guide, followed by price-performance, in many audited benchmarks, not just TPC-H. In these cases, even minimal changes in performance can cause an expected publication goal to be missed. In our experiments, we showed that best performance does not imply best energy efficiency by a wide-margin. In real-world deployments pursuing the energy tradeoff is practical, and benchmarks should reflect this choice.

As part of ongoing work, we are continuing to do additional experiments with the our audit-class system and are interested in examining the impact of per-component energy proportionality. We are also examining audit-class TPC-C and TPC-E workloads and are interested in understanding similarities and differences for energy efficiency across different classes of data processing workloads. Finally, we are interested in turning energy-efficiency behavior we observed into insights for future server hardware design.

6. REFERENCES

[1] L. Barroso. The price of performance. *ACM Queue*, 3(7), Sept. 2005.

[2] L. Barroso and U. Holzle. The case for energy-proportional computing. *IEEE Computer*, pages 33–37, Dec. 2007.

[3] H.-P. Company. TPC Benchmark H, Full Disclosure Report, Nov 2008. Result ID:108111702, `http://www.tpc.org/tpch/results/tpch_result_detail.asp?id=108111702`.

[4] X. Fan, C. Ellis, and A. Lebeck. Memory controller policies for DRAM power management. In *Low-Power Systems and Design (ISLPED)*, 2001.

[5] R. Gonzalez and M. Horowitz. Energy dissipation in general purpose microprocessors. *IEEE Journal of Solid-State Circuits*, 31(9):1277–1284, Sept. 1996.

[6] S. Harizopoulos, M. Shah, J. Meza, and P. Ranganathan. Energy efficiency: The new holy grail of data management systems research. In *CIDR*, Jan 2009.

[7] R. N. Mayo and P. Ranganathan. Energy consumption in mobile devices: Why future systems need requirements-aware energy scale-down. In *PACS*, pages 26–40, Dec. 2003.

[8] P. Pillai and K. G. Shin. Real-time dynamic voltage scaling for low-power embedded operating systems. In *SOSP*, pages 89–102, 2001.

[9] E. Pinheiro and R. Bianchini. Energy conservation techniques for disk array-based servers. In *ICS*, June 2004.

[10] M. Poess and R. Nambiar. Energy cost, the key challenge of today's data centers: A power consumption analysis of tpc-c results. In *VLDB*, 2008.

[11] K. Rajamani et al. Power management for computer systems and datacenters. ISPLED Tutorial, Aug. 2008.

[12] S. Rivoire, M. Shah, P. Ranganathan, and C. Kozyrakis. Joulesort: A balanced energy-efficiency benchmark. In *SIGMOD*, June 2007.

[13] Standard Performance Evaluation Corporation. SPEC power and performance committee. Online. http://www.spec.org/specpower/.

[14] N. Tolia, Z. Wang, et al. Delivering energy proportionality with non energy-proportional systems – optimizing the ensemble. In *HotPower*, 2008.

[15] TPC Council. Transaction Processing Performance Council. Online, 2009. `http://www.tpc.org/`.

[16] United States Environmental Protection Agency. Report to Congress on Server and Data Center Energy Efficiency, Public Law 109-431, Aug. 2007.

[17] D. Vesset, B. McDonough, K. Wilhide, M. Wardley, R. McCullough, and D. Sonnen. Worldwide Business Analytics Software 2007-2011 Forecast Update and 2006 Vendor Shares. Technical Report 208699, IDC, Sept. 2007.

[18] Watts Up? Meters. Onlne, 2009. `https://www.wattsupmeters.com/secure/products.php`.

[19] Y. Zhang, S. Gurumurthi, and M. R. Stan. Soda: Sensitivity based optimization of disk architecture. In *DAC*, 2007.

[20] Q. Zhu, Z. Chen, L. Tan, et al. Hibernator: Helping disk array sleep through the winter. In *SOSP*, 2005.