Paper Report: Analyzing the Energy Efficiency of a Database Server

Presenter: Ervina Cergani

Tutor: Christian Pölitz
Opponent: Alexander Bunte

Authors: *Dimitris Tsirogiannis*
University of Toronto
Stavros Harizopoulos
HP Labs
Mehul A. Shah
HP Labs

1. Introduction

The main motivations of writing this paper from the authors, was the need of high energy efficiency in large data centers and also the question on how to assemble the newer hardware components, that today's markets offer, into larger energy efficiency systems.

In difference with the previous studies, where the performance played an important role in evaluating the Database Systems, this paper focus on the importance of energy efficiency. The paper studies the energy efficiency in hardware/platform oriented and also in workload-management oriented. There have been some previous work in this direction, mostly oriented in the database software field, but the aim of this paper is to better understand the energy characteristics of database systems on modern hardware.

This paper's focus is first to assess and then explore ways to improve the energy efficiency of a single-machine instance of a database server in a scale-out (shared-nothing) architecture, with standard server-grade hardware components, running a wide spectrum of data management tasks. It focuses on understanding the power performance trade-off for a single database node in a scale out (shared nothing) architecture.

The paper presents several experiments to measure the power of systems components from idle state to fully utilize. It provides a good description of the power profile usage of the different hardware components in one configuration of an 8 core (dual CPU) test machine in the context of database operations. From the workload-management oriented, the paper presents the total CPU power consumption for several operators, in different configurations (varying the number of cores used from 1 to 8, CPU frequency, storage system). The authors used a high performance, open–source database storage engine that supports both columns-oriented and row–oriented database scans. The authors of the paper, furthermore, used two different DBMSs: one open–source system (PostgreSQL) and one commercial (System - X).

In the majority of the experiments they collected over 1’000 data points and observed that for any given database tasks, the most energy-efficient configuration are the highest performing one. Each point in the graphs throughout the paper represents different configurations (number of CPU, CPU frequency etc).

2. Experiments and results

2.1 Micro Benchmarks

The paper analyzes the power profiles of different hardware components in the context of database operations. The goal for this is to find how these operations affect power consumption and to reveal the energy saving potential. For this purpose, the authors designed a set of *micro-benchmarks* to exercise the hardware components of a database server using typical database-centric operations:

1. A hash join kernel, which processes in three steps: partitioning, build and probe,
2. A sort kernel that implements two in-memory parallel sorting algorithms: AlphaSort–S produces cache-sized sorted runs in parallel and then merges this runs using a serial merging phase, AlphaSort–P applies a parallel merging phase
3. A scan kernel, to scan uncompressed rows in memory and compressed columns on disks.

The micro–benchmarks were designed to exercise all cores as well as their shared resources, such as CPU cache and memory bus. Analyzing the power consumption of this three micro benchmarks in performance oriented policy and energy efficiency policy, the authors of the paper found that operators with high up-front costs also exhibited higher memory bus utilization, but could not provide a conclusive explanation for this effect.

The experiment for analyzing power consumption of the three database operators based purely on utilization, were not suitable for predicting CPU power. For this reason, the authors analyzed the power consumption of each individual operator using different configurations (number of cores, scheduling policies, and CPU frequencies). They found that hash join and row scans highly utilize the memory bus, but are able to continue increasing performance with each additional core used, as each core can work on a cache-resident data set. Compressed column scans and sort are not bound by the memory bus, as they require more CPU cycles for each byte read (due to columnar storage and compression) and are bound instead by disk bandwidth.

2.2 Energy vs. Performance

The paper presents a set of experiments to analyze how hardware and software "knobs" affect the energy efficiency of database workloads. For this purpose, they used the micro-benchmark from an energy efficiency perspective and two DBMS engines: PostgreSQL and System-X. For these analyzes they used parameters that have the greatest impact on energy efficiency:

1. Algorithm/plan selection. The authors measured the energy efficiency of a wide range of queries using different algorithms (sort merge and hash join), access methods and join orderings that exercise all components of a database server.
2. Intra–operator parallelism (number of cores running a single operator). In the experiments with parallel hash join and parallel sort, parallel operators show roughly the same energy efficiency/performance rates as the non parallel ones, but they differ in power range usage and the parallel sort can reach a better performance and a little increase in energy efficiency.
3. Inter query parallelism (number of independent queries running in parallel),
4. Physical layout (row vs. column scans),
5. Storage layout (striping),
6. Choice of storage medium (HDD vs. SSD).

From all experiments developed, the authors obtained the same result, that regardless of query complexity and what knobs are used (access method, operator algorithms) and type and level of parallelism, energy efficiency and performance go hand in hand. The reason of this behavior is because of the concave down nature of the power performance curves as well as the fixed power costs. For such curves, the relative performance increases are worth the added relative power, so energy efficiency improves with performance.

3. Strong Points of the Paper

The main contributions of this paper are:

1. A detailed study of the power-performance profiles of core database operators on modern scale-out hardware. The paper concluded that CPU power does not vary linearly with CPU utilization, and utilization is a poor proxy for CPU power. The CPU power used by various operators can vary up to 60%, even when they have the same utilization.
2. A thorough investigation of the effects of both hardware and software knobs on the energy efficiency of complex queries in two widely used engines: PostgreSQL and System-X.
3. Unlike what previous studies have suggested, this paper found that the highest performing configuration is the most energy-efficient and there is no need for DBMS software to optimize for energy efficiency apart from performance.

The paper gives some suggestions on how the results can be used towards two promising directions:

1. On techniques spanning multiple nodes (e.g., resource consolidation in underutilized clusters)
2. On alternative energy-efficient hardware configurations (e.g., low-power non-server grade components)

This is the first study that shows the CPU power use of database-like operators in modern processors as the CPU utilization varies.

In difference from the other literature, the experiments in the paper concluded that SSDs power usage is nearly proportional to utilization.

The results reveal opportunities for cross-node energy optimizations (e.g., the last 30% of a node's CPU computation capacity comes essentially for free) and inefficiencies in server CPUs that new/alternative hardware should address (e.g., need for smaller CPU caches).

The paper analyzes in details the two reasons (idle power and the shape of the performance curves of the workloads), of its main result: The most efficient configuration is typically the best performing one.

4. Weak Points of the Paper

The paper makes series of experiments for the effects of both hardware and software knobs on the energy efficiency of complex queries in two widely used engines: PostgreSQL and System-X, but it never compared the results obtained in each of these two engines used. It could give some results which of these two engines is better according to energy efficiency.

The authors of the paper develop a series of experiments to analyze the power consumption from different hardware components and database operators and explain these results, but do not give concrete suggestions of how to improve the energy efficiency for each of these components. In the paper, they analyzed the power consumption of each individual operator (join, sort and scan) using different configurations (number of cores, scheduling policies, and CPU frequencies), but don't give some suggestions on how to improve the Memory Bus Utilization from the compressed column scan and sort operator.

The authors were more focused in studying the theoretical part of the energy efficiency of a database server and explain the results of the experiments, but they didn't give some practical results on how a individual or company can built a cluster server in order to have performance and energy efficiency.

The experiments were only run on a single hardware configuration where they only varied some settings like the number of used cpu-cores and disks, but they did not try completely different computers like a mainframe. So the question is, if these results also hold for other designs of computers.

In the experiments throughout the paper, the authors simplified their analyzes using a single node machine. In this context their results are an abstract of the reality, since in almost the cases companies operate on a database server running distributed over different nodes. They only give suggestions on how to extend this to a multiple-node-cluster but did not make experiments concerning this topic.