WattDB—a Rocky Road to Energy Proportionality

Theo Härder

Databases and Information Systems Group
University of Kaiserslautern, Germany
haerder@cs.uni-kl.de

Extended Abstract

Energy efficiency is becoming more important in database design, i. e., the work delivered by a database server should be accomplished by minimal energy consumption. So far, a substantial number of research papers examined and optimized the energy consumption of database servers or single components. In this way, our first efforts were exclusively focused on the use of flash memory or SSDs in a DBMS context to identify their performance potential for typical DB operations. In particular, we developed tailor-made algorithms to support caching for flash-based databases [3], however with limited success concerning the energy efficiency of the entire database server.

A key observation made by Tsirogiannis et al. [5] concerning the energy efficiency of single servers, the best performing configuration is also the most energy-efficient one, because power use is not proportional to system utilization and, for this reason, runtime needed for accomplishing a computing task essentially determines energy consumption. Based on our caching experiments for flash-based databases, we came to the same conclusion [2]. Hence, the server system must be fully utilized to be most energy efficient. However, real-world workloads do not stress servers continuously. Typically, their average utilization ranges between 20 and 50% of peak performance [1]. Therefore, traditional single-server DBMSs are chronically underutilized and operate below their optimal energy-consumption-per-query ratio. As a result, there is a big optimization opportunity to decrease energy consumption during off-peak times.

Because the energy use of single-server systems is far from being *energy proportional*, we came up with the hypothesis that better energy efficiency may be achieved by a cluster of nodes whose size is dynamically adjusted to the current workload demand. For this reason, we shifted our research focus from inflexible single-server DBMSs to distributed clusters running on lightweight nodes. Although distributed systems impose some performance degradation compared to a single, brawny server, they offer higher energy saving potential in turn.

Current hardware is not energy proportional, because a single server consumes, even when idle, a substantial fraction of its peak power [1]. Because typical usage patterns lead to a server utilization far less than its maximum, energy efficiency of a server aside from peak performance is reduced [4]. In order to achieve energy proportionality using commodity hardware, we have chosen a clustered approach, where each node can be powered independently. By turning on/off whole nodes, the overall performance and energy consumption can be fitted to the current workload. Unused servers could be either shut down or made available to other processes. If present in a cloud, those servers could be leased to other applications.

We have developed a research prototype of a distributed DBMS called *WattDB* on a scale-out architecture, consisting of n wimpy computing nodes, interconnected by an 1GBit/s Ethernet switch. The cluster currently consists of 10 identical nodes, composed of an Intel Atom D510 CPU, 2 GB DRAM and an SSD. The configuration is considered Amdahl-balanced, i. e., balanced between I/O and network throughput on one hand and processing power on the other.

Compared to InfiniBand, the bandwidth of the interconnecting network is limited but sufficient to supply the lightweight nodes with data. More expensive, yet faster connections would have required more powerful processors and more sophisticated I/O subsystems. Such a design would have pushed the cost beyond limits, especially because we would not have been able to use commodity hardware. Furthermore, by choosing lightweight components, the overall energy footprint is low and the smallest configuration, i. e., the one with the fewest number of nodes, exhibits low power consumption. Moreover, experiments running on a small cluster can easily be repeated on a cluster with more powerful nodes.

A dedicated node is the *master node*, handling incoming queries and coordinating the cluster. Some of the nodes have each four hard disks attached and act as *storage nodes*, providing persistent data storage to the cluster. The remaining nodes (without hard disks drives) are called *processing nodes*. Due to the lack of directly accessible storage, they can only operate on data provided by other nodes (see Figure 1).

All nodes can evaluate (partial) query plans and execute DB operators, e. g., sorting, aggregation, etc., but only the *storage nodes* can access the DB storage structures, i. e., tables and indexes. Each storage node maintains a DB buffer

25th GI-Workshop on Foundations of Databases (Grundlagen von Datenbanken), 28.05.2013 - 31.05.2012, Ilmenau, Germany.

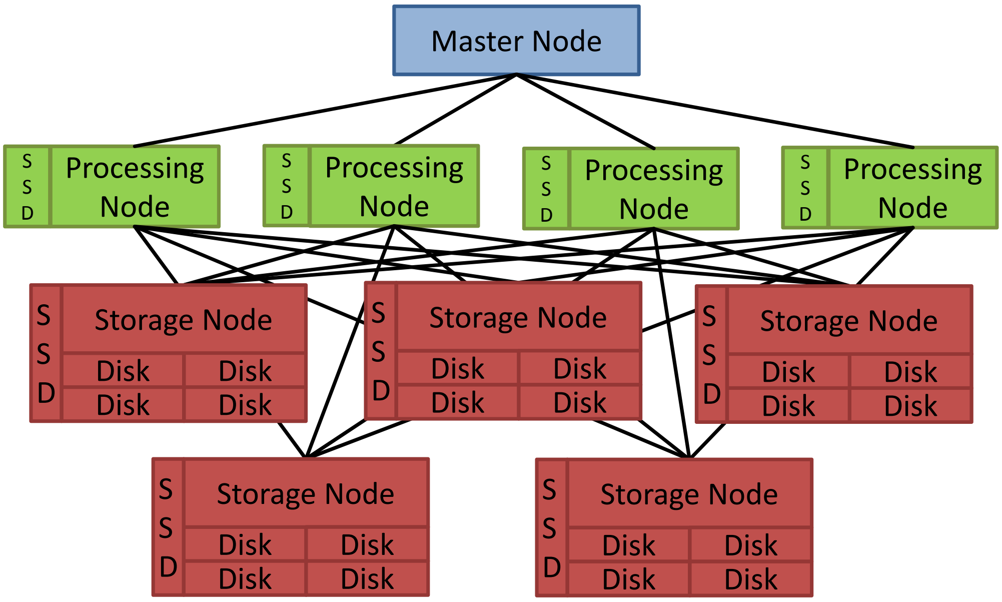

Figure 1: Overview of the WattDB cluster

to keep recently referenced pages in main memory, whereas a processing node does not cache intermediate results. As a consequence, each query needs to always fetch the qualified records from the corresponding storage nodes.

Hence, our cluster design results in a *shared-nothing architecture* where the nodes only differentiate to those which have or have not direct access to DB data on external storage. Each of the nodes is additionally equipped with a 128GB Solid-State Disk (Samsung 830 SSD). The SSDs do not store the DB data, they provide swap space to support external sorting and to provide persistent storage for configuration files. We have chosen SSDs, because their access latency is much lower compared to traditional hard disks; hence, they are better suited for temp storage.

In WattDB, a dedicated component, running on the master node, controls the energy consumption, called *EnergyController*. This component monitors the performance of all nodes in the cluster. Depending on the current query workload and node utilization, the *EnergyController* activates and suspends nodes to guarantee a sufficiently high node utilization depending on the workload demand. Suspended nodes do only consume a fraction of the idle power, but can be brought back online in a matter of a few seconds. It also modifies query plans to dynamically distribute the current workload on all running nodes thereby achieving balanced utilization of the active processing nodes.

As data-intensive workloads, we submit specific TPC-H queries against a distributed shared-nothing DBMS, where time and energy use are captured by specific monitoring and measurement devices. We configure various static clusters of varying sizes and show their influence on energy efficiency and performance. Further, using an *EnergyController* and a load-aware scheduler, we verify the hypothesis that energy proportionality for database management tasks can be well approximated by dynamic clusters of wimpy computing nodes.

1. REFERENCES

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

[2] T. Härder, V. Hudlet, Y. Ou, and D. Schall. Energy efficiency is not enough, energy proportionality is needed! In *DASFAA Workshops, 1st Int. Workshop on FlashDB, LNCS 6637*, pages 226–239, 2011.

[3] Y. Ou, T. Härder, and D. Schall. Performance and Power Evaluation of Flash-Aware Buffer Algorithms. In *DEXA, LNCS 6261*, pages 183–197, 2010.

[4] D. Schall, V. Höfner, and M. Kern. Towards an Enhanced Benchmark Advocating Energy-Efficient Systems. In *TPCTC, LNCS 7144*, pages 31–45, 2012.

[5] D. Tsirogiannis, S. Harizopoulos, and M. A. Shah. Analyzing the Energy Efficiency of a Database Server. In *SIGMOD Conference*, pages 231–242, 2010.