WattDB: An Energy-Proportional Cluster of Wimpy Nodes

Daniel Schall
AG DBIS
TU Kaiserslautern
Kaiserslautern, Germany
schall@cs.uni-kl.de

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

ABSTRACT

The constant growth of data in all businesses leads to bigger database servers. While peak load times require fast and heavyweight hardware to guarantee performance, idle times are a waste of energy and money. Todays DBMSs have the ability to cluster several servers for performance and fault tolerance. Nevertheless, they do not support dynamic powering of the cluster's nodes based on the current workload. In this demo, we propose a newly developed DBMS running on clustered commodity hardware, which is able to dynamically power nodes. The demo allows the user to interact with the DBMS and adjust workloads, while the cluster's reaction is shown in real-time.

Categories and Subject Descriptors

H.2.4 [**Database Management**]: Systems—*Distributed databases*

General Terms

Design, Measurement

1. INTRODUCTION

Todays servers all over the world are consuming more and more energy. While energy costs are steadily rising, a lot of energy is wasted, because servers are running idle or at low utilization. Servers usually do not employ any energy-saving mechanisms like standby or spin-down idle disks, because transition times to ready state are unacceptably high. A typical database server consumes about 60 % of its peak energy comsumption when idle [6]. Two important reasons why servers consume so much energy are that hard disks are not being spun down automatically and that DRAM needs constant power, independent of the workload.

A server that consumes almost no energy when being idle is highly desirable, unfortunately yet out of reach due to hardware limitations. Even with optimized hardware and the use of power-saving mechanisms, idle power consumption is a big issue. Servers usually come with powerful multi-core processors and plenty amounts of DRAM and storage disks to keep response times low even during peak loads. The downside of this *KIWI* (*kill it with iron*) approach is the high idle power consumption due to all these components. To improve energy efficiency, a new paradigm arose lately, called *energy proportionality* [2]. That means, a server should only consume energy proportionally to its workload-driven utilization. Therefore, a server running at full load is allowed to consume peak energy, while a server on 50 % load should only consume 50% of its peak energy. In fact, todays servers need almost the same energy, whether they run at 50 % or 100 % load.

It is impossible to achieve *energy-proportional* behavior using a single (large / *brawny* [3]) server. There exist some approaches to power down (parts of) the storage system, nevertheless, because of the high transition times and further constant power consumers, the overall outcome is dissatisfying. Therefore, we are going to propose a new approach in this demo by employing commodity hardware to build an energy-proportional cluster. Small-scale server nodes (aka *wimpy nodes* [1] or *Amdahl blades* [5]) can be independently turned on and off. By powering nodes based on the overall workload, the total energy consumption converges to a straight line, thus showing energy-proportional behavior.

In this demo, we are going to introduce our idea of an energy-proportional database cluster and show how we measure power consumption. We will present an interactive front-end that controls the system's workload and illustrates how the system reacts to rising and falling loads.

In Section 2, we will describe our cluster, both the hardware setup as well as key points of our database software running on the nodes. Moreover we scetch the energy measurement hardware we developed. Section 3 and 4 outline what we will show in the Demo Session and how interested visitors can interact with the system. In the last section, we will briefly summarize our idea.

2. CLUSTER OVERVIEW

The cluster consists of 10 identical nodes, two of them have attached 4 hard disks to provide persistent storage. The nodes are interconnected by a 1Gb ethernet switch. Database clients like our workload driver can interact with the cluster by connecting to a dedicated master node, which will provide an SQL interface. All nodes are able to process queries, although the nodes with the hard disks attached will rather focus on storage management, when the load of the cluster increases. Other nodes can request pages (and more coarse-grain data objects) from the storage disks to process them individually. From the physical perspective of the individual compute nodes, our node cluster is a shared-disk system; logically it is a shared-nothing system coordinated by a dedicated master node. Figure 1 illustrates the cluster layout.

SIGMOD'11, June 12–16, 2011, Athens, Greece.
ACM 978-1-4503-0661-4/11/06.

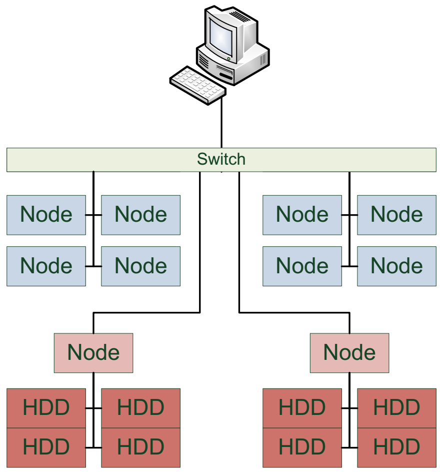

Figure 1: Cluster hardware layout

2.1 Database Nodes

Each node has 2 GB of DRAM and an Intel Atom D510 CPU, a rather lightweight processor without frequency scaling options, but an overall low thermal design power. The power supply is *80plus Gold* certified. Typically, a node consumes about 23 watts when idle and 26 watts at 100 % CPU load. The data nodes have attached 4 hard disk drives each to their SATA ports, each disk having 250 GB storage space, thus a data node consumes max. 41 watts. We are using low-power hard disks to reduce the energy footprint of the system. The operating system of the nodes is booted from an attached USB stick. Due to the limited bandwidth provided by the ethernet and the hard disks, a lightweight CPU is sufficient to process data. We consider this combination *amdahl-balanced* [5].

2.2 Database Software

In order to fully support dynamic node powering, we are developing our own database management system, called **WattDB**. The system is capable of reacting to changing load situations and able to adapt the cluster to the current workload. By switching nodes on and off, the overall energy consumption is made proportional to the load, thus energy-proportional. Each node is providing feedback about its resource utilization to a coordinator, which uses this information to re-balance the cluster by automatically powering up and down nodes. The key challenges faced here are to equally balance load over a minimally necessary set of nodes and to redistribute data before powering down or after powering up a node. Depending on a given power policy, a node determined surplus is suspended (suspend-to-RAM) or shutdown. The former still consumes about 3 watts, but can be powered up via wake-on-LAN in less than 5 seconds. Furthermore, a *lazyness* parameter can defer the power policy to counteract oscillating workloads and to enhance responsiveness by sacrificing energy.

WattDB is designed from scratch to include energy proportionality as a first-class goal. Although we build on top of the database engine we developed for last year's programming contest at SIGMOD 2010, the system is in an early development state. Currently, WattDB is capable of distributing and processing simple queries on multiple nodes.

Figure 2: Measurement device for a cluster of nodes

2.3 Energy Measurement

In order to measure the energy consumption of a whole cluster of nodes, we developed a measurement device tailored for this purpose. The focus is not on the power consumption of computer components, but on the overall energy consumption of each node. Figure 3 depicts the internal wiring diagram. This device is able to track the power consumption of 10 server nodes independently at 1 % accuracy with 10 Hz resolution. We are able to measure the effective power of each node and the total apparent power. Additionally, power consumption of the environment, i.e. the ethernet switch, is reported as well. All measurements can be read out using an A/D converter and a connected PC. Figure 2 shows a picture of the device. In the live demo, we will show the current power consumption of the cluster [4].

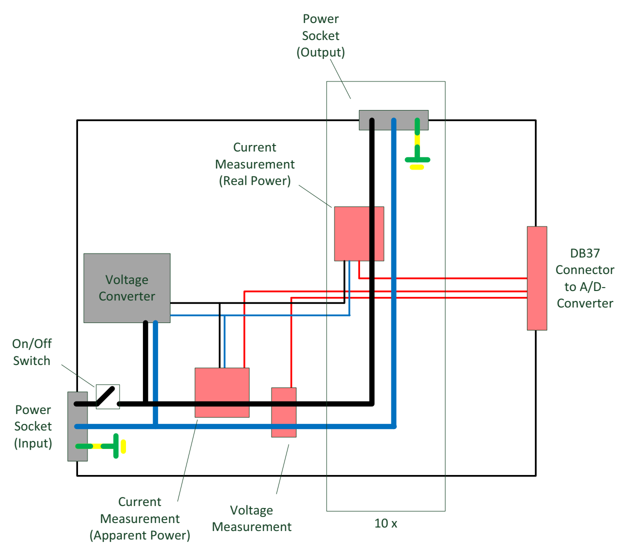

Figure 3: Schematic of the measurement device

3. DEMO SETUP

To demonstrate the capabilities of our database cluster, a simple workload driver can be used to trigger a workload to the system. The database system adapts to the workload by dynamically powering nodes. The entire progress is visible on a GUI as well as on a webcam that shows the state of the cluster in real-time. Power consumption of every node is displayed by a separate graph, as well as the total power consumption. Another graph shows both target and current workload. Figure 4 depicts a screenshot of the user interface: The form on the left shows the various states (either *on*, *off* or *suspended*) of each of the nodes. On the right-hand side,

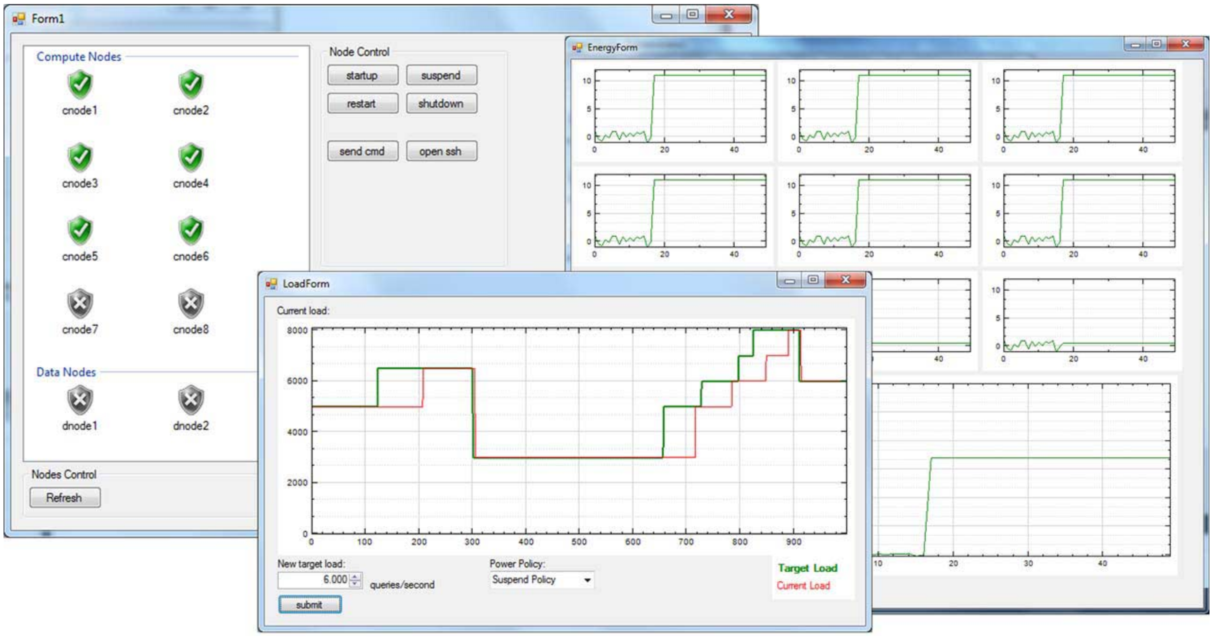

Figure 4: Various GUI forms reporting the node states

the power consumption over time is plotted. At the front of Figure 4, a specific form visualizes the progress of the target and current workloads. Additionally, the average response times of the queries are plotted to demonstrate how queries impact the cluster and how the cluster adapts to the new workload.

4. USER INTERACTION

The user can alter the workload of the system using a GUI. The *power policy* can be changed as well. This policy controls the way nodes are powered up and down, what sleep states the nodes should enter and how aggressive the power management is done. The effect of the actions is immediately visible – on the GUI as well as through the webcam. Attendees can see how the system reacts to changing workloads and how the power consumption is affected. The audience will also see a webcast of the cluster as depicted in Figure 5.

5. SUMMARY

Our database cluster exhibits energy-proportional behavior and we'd like to emphasize the importance of this topic by showing a demo at the SIGMOD conference. Saving energy is one of the major topics in computer science and our research project will help developing better energy-efficient hardware and software systems. We are trying to show that lightweight nodes are more flexible than oversized servers. Although the peak computational power of our cluster in terms of *Queries/Watt* is lower than comparable single-server-based solutions, our implementation will easily reduce TCO by saving energy. By showing our demo, we'd like to encourage researchers to think about the energetic advantages of wimpy nodes and we'd like to demonstrate that it's possible to develop an energy-proportional, yet high-performance database cluster.

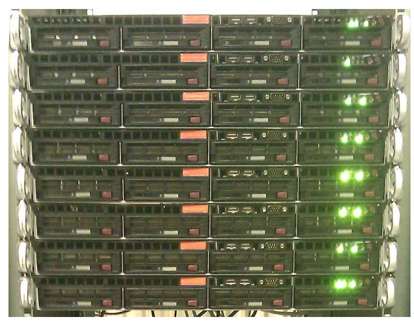

Figure 5: Webcast of the cluster

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

[3] U. Hölzle. Brawny Cores Still Beat Wimpy Cores, Most of the Time. *IEEE Micro*, 2010.

[4] V. Hudlet and D. Schall. Measuring Energy Consumption of a Database Cluster. In *Proc. 14-th GI-Conference Database Systems for Business, Technology and Web, LNI - P 180*, pages 734–737, 2011.

[5] A. S. Szalay, G. C. Bell, H. H. Huang, A. Terzis, and A. White. Low-Power Amdahl-Balanced Blades for Data Intensive Computing. *SIGOPS Oper. Syst. Rev.*, 44(1):71–75, 2010.

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