TECHNISCHE UNIVERSITÄT DRESDEN

Energy-Aware Data Management on NUMA Architectures

Kurzfassung der Dissertation

zur Erlangung des akademischen Grades
Doktoringenieur (Dr.-Ing.)

vorgelegt an der
Technischen Universität Dresden
Fakultät Informatik

eingereicht von
Dipl.-Ing. Thomas Kissinger
geboren am 25. April 1985 in Frankfurt (Oder)

Betreuender Hochschullehrer:
Prof. Dr.-Ing. Wolfgang Lehner

Dresden, Januar 2017

Energy-Aware Data Management on NUMA Architectures (Extended Abstract)

Thomas Kissinger

A first study in 2008 revealed that energy consumption of data centers is a critical problem, since their power consumption is about to double every 5 years. However, a follow-up study (2016) points out that this trend was throttled within the past years, due to the increased energy efficiency actions taken by data center operators. The authors emphasize that keeping data centers energy-efficient is a continuous task and that this trend will resume as soon as energy efficiency research efforts and its market adoption are reduced. Data management systems are a fundamental component of nearly every application stack. Modern state-of-the-art database systems are main memory-centric and use non-uniform memory access (NUMA) hardware architectures to scale up.

In this thesis, we investigate energy awareness aspects of large scale-up NUMA systems in the context of in-memory data management systems. To achieve this goal, we design and build ERIS, the first scale-up in-memory data management system that is designed from scratch to implement a data-oriented architecture. With the help of ERIS, we explore our novel core concept for energy awareness, which is *Energy Awareness by Adaptivity*. We present the hierarchically organized Energy-Control Loop (ECL), which is a reactive control loop and provides two concrete implementations of our Energy Awareness by Adaptivity concept, namely the hardware-centric Resource Adaptivity and the software-centric Storage Adaptivity. In our evaluation, we measured a superior scalability and outstanding improvements of energy consumption and query latency.

1 Introduction

The ever-increasing need for more computing and data processing power demands for a continuous and rapid growth of power-hungry data center capacities all over the world. As the first comprehensive analysis of U.S. data centers energy consumption in 2008 [4] concluded: The energy consumption of such data centers is becoming a critical problem, since their power consumption is about to double every 5 years. Moreover, the report stated that in 2006 about 61 billion kilowatt-hours (kWh) –

equal to about $4.5 billion in electricity costs – were consumed by data centers only in the United States. However, a recently (2016) released follow-up study [12] revealed that this threatening trend was dramatically throttled within the past years, due to the increased energy efficiency actions taken by data center operators. The increase of U.S. data centers energy consumption only increased by 24 % from 2005 to 2010 and by 4 % from 2010 to 2014. The results of the study also emphasize that making and keeping data centers energy-efficient is a *continuous task*, because more and more computing power is demanded from the same or an even lower energy budget, and that this threatening energy consumption trend will resume as soon as energy efficiency research efforts and its market adoption are reduced. Such a lack of innovation would have led to an estimated overall energy wasting of 620 billion kWh between 2010 and 2020.

An important class of applications running in data centers are data management systems, which are a fundamental component of nearly every application stack. While those systems were traditionally designed as disk-based databases that are optimized for keeping disk accesses as low a possible, modern state-of-the-art database systems are main memory-centric and store the entire data pool in the main memory, which replaces the disk as main bottleneck. To scale up such in-memory database systems, non-uniform memory access (NUMA) hardware architectures are employed that face a decreased bandwidth and an increased latency when accessing remote memory compared to the local memory.

In this thesis, we investigate energy awareness aspects of large scale-up NUMA systems in the context of in-memory data management systems. To do so, we pick up the idea of a fine-grained *data-oriented* architecture [10, 11] and improve the concept in a way that it keeps pace with increased absolute performance numbers of a pure in-memory DBMS and scales up on NUMA systems consisting of up to 64 sockets and a total of 768 hardware threads and beyond. To achieve this goal, we design and build the first scale-up in-memory data management system – namely *ERIS* – that is designed from scratch to implement a *data-oriented* architecture.

With the help of the *ERIS* platform, we explore our novel core concept for energy awareness, which is *Energy Awareness by Adaptivity*. The concept describes that software and especially database systems have to quickly respond to environmental changes (i.e., workload changes) by adapting themselves to enter a state of low energy consumption. We will show that the *data-oriented* architecture already provides a solid foundation for quick adaptations, but still misses important changes, which are covered by our *Living Partitions* architecture that understands individual data partitions as evolving objects that are not bound to a specific hardware thread anymore. Finally, we present the hierarchically organized *Energy-Control Loop*, which is an reactive control loop and provides two concrete implementations of our *Energy Awareness by Adaptivity* concept: (1) the hardware-centric *Resource Adaptivity* as an holistic approach for managing hardware energy-control knobs at runtime and (2) the software-centric *Storage Adaptivity* that is responsible for continuously adapting the physical storage layout of the database system at runtime. We implemented both *Adaptivity Facilities* in ERIS and evaluate the entire system in terms of scalability, energy consumption, and responsiveness during the adaptation process in the presence of a varying workload pattern.

2 Energy Awareness in Data Management

In this chapter, we introduce the topic of energy awareness in database systems and discuss the important aspects of how to measure and benchmark energy awareness. As the main contribution of this chapter, we finally come up with our core concept of *Energy Awareness by Adaptivity* and formulate a variety of requirements that a database system architecture needs to fulfill to enable specific implementations of this concept.

2.1 Energy Awareness

In this section, we discuss certain metrics describing the energy awareness of a data management system. Beforehand, we need to define what energy awareness exactly is. In our understanding:

Definition 2.1 (Energy Awareness)**.** *Energy Awareness is the ability of software (e.g., a database system) to be conscious of its energy consumption behavior related to the amount of work it is executing.*

While energy is a well-defined physical unit that can be measured by built-in energy counters, the metrics for determining the work respectively performance of a database system depend on the application area. High-level application-specific work measurements are a suitable choice for evaluating standardized application benchmarks (e.g., the TPC family). For low-level performance measurements, performance counters can be used that are implemented in mostly all modern processors and cover a wide range of component-specific measurements.

As the main metrics for energy awareness, we introduce *energy efficiency* and *energy proportionality*. Energy efficiency is defined as the quotient of work and energy respectively performance and power. To improve the metric, modern processors use performance states (P-states) to adjust the trade-off between performance and power at runtime. Due to the non-linear correlation of performance and power as well as the existence of user-defined performance and latency demands, the appropriate choice a performance state is a non-trivial problem. The second metric for energy awareness is energy proportionality, which expresses a proportional relation between work and energy or power and performance and is a critical metric for reducing the energy footprint of a database system. To achieve energy proportionality, modern processors implement processor states (C-States) to turn off unneeded cores.

To assess the energy awareness of a DBMS, benchmarking approaches are required that consider energy proportionality and the dependency between system load and energy efficiency. Thus, we propose to use a combination of a *workload specification* and a *load profile specification* to overcome those weaknesses. While the workload specification is already given by existing benchmarks, load profiles need to be standardized for valuable comparisons. A special focus of such a benchmarking approach for energy awareness are the query latencies, which can either be compared using the energy-delay product (EDP) or by quantifying violations against the service-level agreement (SLA).

2.2 Energy Awareness by Adaptivity

Based on our discussion of energy awareness, we derive our core concept of *Energy Awareness by Adaptivity*, which is inspired by the natural mechanism of evolution. Thus, we understand a database system as an organism in a continuous struggle for energy and frequent adaptations are the appropriate countermeasure to face this challenge. We identified *Resource Adaptivity*, *Storage Adaptivity* and *Data Placement Adaptivity* as *Adaptivity Facilities* implementing our core concept (cf., Figure 1). Nevertheless, to enable those fine-grained adaptations at runtime, a lot of requirements need to be fulfilled. Those requirements either originate from general performance observations or from the individual adaptivity facilities of our energy awareness by adaptivity concept. In the following chapter, we will pick up these requirements and compare existing architectures for their ability to fulfill them and mainly focus on scalability as a prerequisite for energy proportionality.

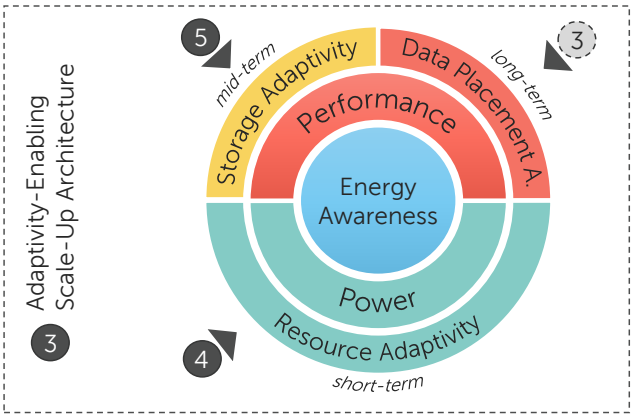

Figure 1: Energy Awareness by Adaptivity concept.

Finally, we propose the *Energy-Control Loop (ECL)* as the hosting framework for our adaptivity facilities as we proposed within our collaborative research center "HAEC" [1]. The ECL employs the design principle of a closed reactive control loop. Hence, the ECL continuously monitors certain database system metrics and responds to workload and system load changes using the measures of the respective adaptivity facility. Moreover, the ECL is organized hierarchically to consider the scope of the specific adaptation facilities including the availability of DBMS metrics and the respective time scale of adaptation.

3 Adaptivity-Enabling Scale-Up Architecture

In this chapter, we start with an exploration of current medium and large scale-up NUMA system architectures to quantify and asses the impact of remote main memory accesses on such architectures. Especially on large-scale NUMA systems, our experiments revealed that latency and throughput differ up to an order of magnitude when accessing main memory remotely, which emphasizes that local main memory access is the key factor for scalability on such hardware platforms. Based on those insights, we classify existing DBMS architectures in terms of their ability to

scale-up on large NUMA systems and their ability to allow fine-grained adaptations at runtime. We conclude that – compared to the transaction-oriented architecture – the data-oriented architecture [10, 11] provides us with the best foundation for fulfilling our requirements for an energy-aware DBMS. However, this architecture still lacks (1) an investigation and appropriate concepts for in-memory DBMSs on large-scale NUMA systems and (2) certain requirements originating from our adaptivity facilities.

3.1 DORA for In-Memory DBMSs on Large-Scale NUMA Systems

To cope with the first issue, we transfer existing concepts of the data-oriented architecture from medium-scale disk-based systems to large-scale in-memory systems, which is mainly a matter of the message passing subsystem that needs to keep pace with the increased speed of data object accesses. We implemented the corresponding proof of concept (PoC) to evaluate our concepts and focus on database primitives such as scans and index accesses. Our evaluation showed that the data-oriented architecture is able to scale up on large-scale NUMA systems in the context of an in-memory database system and clearly outperforms the classic transaction-oriented architecture. Our in-depth evaluation also reflects on the root causes for this scalability gap between both architectures. Moreover, we demonstrate that *Data Placement Adaptivity* can efficiently be done in such an environment.

3.2 ERIS Data Management System

To address the second issue, we extend the data-oriented architecture to enable fine-grained adaptivity at runtime. Hence, we present the *Living Partitions* architecture, which enables a flexible work to hardware thread assignment as well as a late-binding of physical operators. We introduce our in-memory data management system *ERIS*, which is designed from scratch to implement the living partitions architecture as well as our *Adaptivity Facilities*. In contrast to our PoC, ERIS is able to execute comprehensive queries in a transactional environment using constructs like *Tasks*, *Dataflows*, and *Micro Operators*. Furthermore, ERIS employs a hierarchical message passing layer, to deal with the changes introduced by the living partitions architecture.

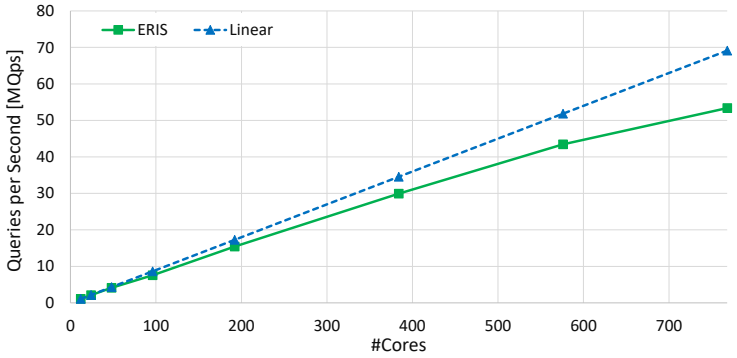

Figure 2: Scalability of the TATP-Mix in ERIS (SF 10).

In the evaluation of ERIS, we use a systematic series of microbenchmarks as well as the standardized transactional TATP benchmark [6] (cf., Figure 2) and demonstrate the superior scalability of ERIS on large-scale NUMA systems (64 sockets and 768 hardware threads). Hence, ERIS and its living partitions architecture are an excellent foundation for investigating our adaptivity facilities with the overall goal to build an energy-aware data management system.

4 Resource Adaptivity

In this chapter, we present *Resource Adaptivity* as the hardware-centric implementation of our *Energy Awareness by Adaptivity* concept. While previous research [9, 13, 15] mainly focused on disk-based DBMSs, resource adaptivity aims at investigating and optimizing the energy consumption of highly parallel state-of-the-art in-memory database systems that make heavy use of the main power consumers – CPUs and main memory – and are thus, an attractive target for energy optimizations. Our in-depth energy analysis of a current server system shows that modern processors provide a rich set of energy-control facilities, but lack the capability of controlling them appropriately.

4.1 Energy Profiles

We specify the concept of *Configurations*, which represent a specific system state in terms of hardware energy-control settings for a single processor. Configurations are evaluated in the context of a specific workload to be enriched by information about the power consumption, the delivered performance, and the effective energy efficiency. A set of configurations is aggregated to an *Energy Profile* (cf., Figure 3). This set of configurations is generated with the help of a configuration generator, which tries to cover the most important supporting points of the big exploration space. We show that the cardinality of the configuration set can be kept low, while still reaching a good quality of the energy profile. Moreover, we demonstrate that the shape of the energy profile is highly workload dependent.

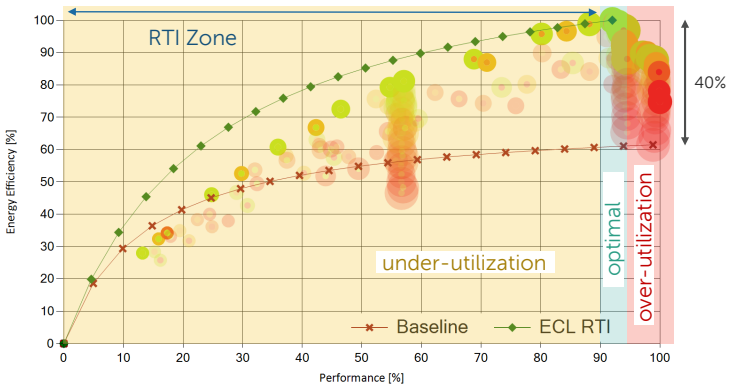

Figure 3: Energy profile of a memory-intensive workload.

4.2 Resource Adaptivity-Specific ECL

Finally, we propose the *Resource Adaptivity-Specific ECL* as a holistic software-based approach for adaptive energy-control on scale-up in-memory database systems that obeys a query latency limit as a soft constraint and actively optimizes energy awareness and performance of the DBMS. Resource adaptivity effectively implements the CPU-level and the system-level of the overall *Energy-Control Loop* by employing a *Node ECL* per processor and a *Global ECL*. Node ECLs rely on adaptive workload-dependent energy profiles that are continuously maintained at runtime using the *Online Adaptation* and *Multiplexed Adaptation* maintenance strategies. The global ECL monitors and projects the current query latencies to influence the resource allocation strategy of the Node ECLs.

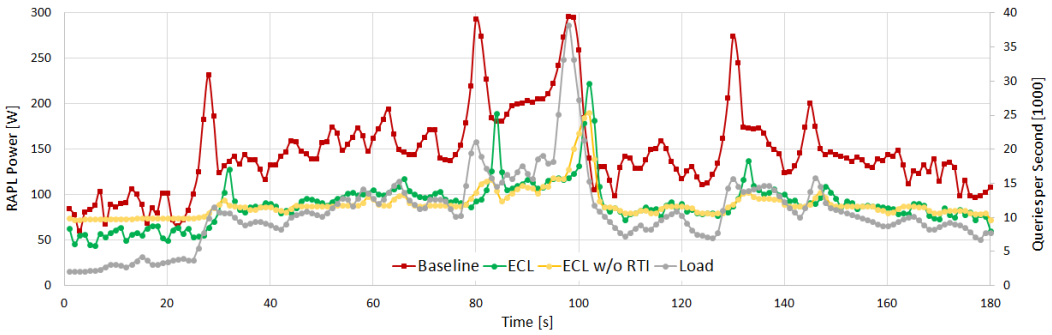

Figure 4: Load profile and power consumption of the baseline and the ECL.

In our evaluation, we observed energy savings of up to about 40 % for real world load profiles (cf., Figure 4). Moreover, we demonstrate that the Node ECLs are able to quickly and efficiently adapt their energy profile in case of workload changes without inducing a significant overhead.

5 Storage Adaptivity

Modern application scenarios require data management systems to cope with a vast variety of datasets and query types that are not known beforehand and change over time. To still provide a superior performance and energy efficiency in all of the potential scenarios, the database system needs to adapt its physical storage layout to the current workload, because the data organization has a significant impact on the query execution performance and there exists no one-size-fits-all physical storage layout. Current solutions for the storage adaptation problem are very limited, because they are either designed as offline approaches [5, 14] or address only a small subset of the available storage layout tuning knobs [2, 3, 7, 8].

In this chapter, we presented *Storage Adaptivity* as a holistic software-centric approach for increasing the performance and energy efficiency of a database system in the presence of varying workloads and data characteristics. Our approach consists of two main components: (1) the *1-Storage* storage manager and the (2) *Storage Adaptivity-Specififc Energy-Control Loop*.

5.1 1-Storage

The first component is *1-Storage*, which is a storage manager that is able to organize the data in a multitude of physical layouts combining the advantages of row-wise and columnar data organizations in combination with adaptive indexing. Furthermore, 1-Storage uses the concept of extensible storage modules to provide support for any kind of access path implementation. Since 1-Storage is designed to operate within the *Living Partitions* architecture, each living partition of a relation is additionally able to apply a different physical storage layout. To decouple database operators from the actual physical storage layout, 1-Storage employs an indirection layer that induces a certain overhead and decomposes the query plan compilation into a *Macro Query Execution Plan* and *Micro Query Execution Plans*.

5.2 Storage Adaptivity-Specifiifc Energy-Control Loop

The *Storage Adaptivity-Specifiifc Energy-Control Loop* leverages this implicit partitioning of the architecture for enabling a fine-grained incremental adaptation of the physical storage layout in case of a changing workload. Moreover, this component integrates well with our *Resource Adaptivity-Specific ECL*, which results in a sophisticated ECL hierarchy that is able to control the adaptation of hardware as well as software at runtime while trying to stay within a user-defined query latency limit.

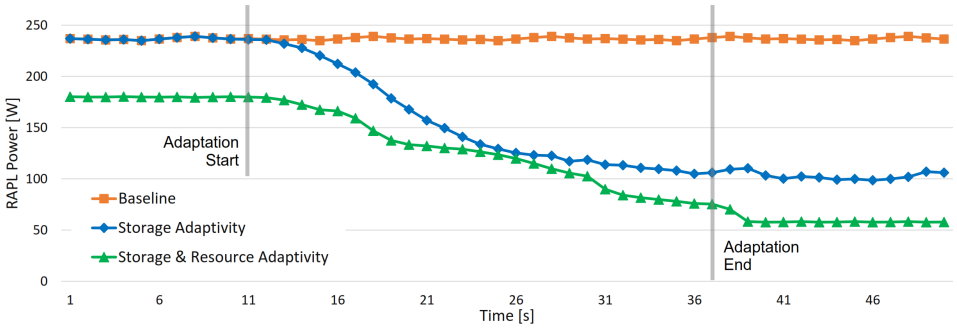

Figure 5: Power consumption over time during the adaptation process.

Our evaluation demonstrates that all ECLs work hand in hand and we observe energy savings of about 65 % while additionally improving the query latency by orders of magnitude (cf. Figure 5). Nevertheless, we also revealed the limitations of the current state of our approach and suggested just-in-time compilation, forecasting, redundancy, and *Data Placement Adaptivity* as augmenting techniques to overcome these limitations.

6 Summary and Conclusions

Recent studies revealed that the energy consumption of server hardware already became a critical problem, especially in data centers. Since data management systems are an application class that amounts to a high portion of the overall deployments, they are responsible for a high share of the energy draw. Another trend is the ongoing move from disk-based to in-memory database systems, which run on hardware

that exhibits more and more non-uniform memory access (NUMA) related effects. In this thesis, we investigated how the energy consumption of such in-memory database systems that run on mid and large scale-up NUMA hardware platforms can be reduced.

In the first place, we discussed the nature of energy in the context of data management systems and derived the term energy awareness as the ability of a DBMS to actively optimize its energy efficiency as well as energy proportionality. We came up with our core concept of *Energy Awareness by Adaptivity*, which aims at active software-driven adaptations at runtime, especially in the presence of changing workloads and data characteristics as it is becoming increasingly common in today's applications. To actually implement this concept, we defined a rich set of requirements that need to be fulfilled to build an energy-aware database system. Those requirements either originate from the general scalability prerequisite or from the ability to enable fine-grained adaptations at runtime.

Our exploration of existing database architectures concluded that none of the known architectures fulfills our requirements for an energy-aware DBMS. Nevertheless, we decided to use the *data-oriented architecture* as a starting point, because of its scalability advantages. We optimized this architecture for large scale-up in-memory database systems and achieved superior scalability results as well as absolute performance numbers that clearly outperform the traditional transaction-oriented architecture. To enable our *Adaptivity Facilities* on this architecture, we proposed the *Living Partitions* architecture that treats *Living Partitions* as autonomous self-adapting objects and presented the database system *ERIS* that is based on this novel architecture. Our evaluation showed superior scalability of ERIS for transnational workloads on a large scale-up NUMA system.

Using ERIS and the living partitions architecture as a solid foundation, we investigated two *Adaptivity Facilities* that implement our core concept. The first implementation is the hardware-centric *Resource Adaptivity*, which actively adapts the hardware configuration by controlling the rich set of available energy-control knobs of current processors. Our resource adaptivity approach implements the *Resource Adaptivity-Specific Energy-Control Loop (ECL)*, which consists of a system-level *Global ECL* and a CPU-level *Node ECL*. While the global ECL keeps track of the current average query latency, the Node ECL maintains an adaptive *Energy Profile* to manage the hardware configurations. In our evaluation, we measured energy savings ranging from 20% to 40% for a real-world load profile.

The second *Adaptivity Facility* we investigated was *Storage Adaptivity*, which is a software-centric approach for adapting the physical storage layout at runtime. Our approach uses the extensible *1-Storage* storage manager that is capable of organizing its data in a wide variety of physical representations covering columnar and row-wise data organizations as well as adaptive indexing. To actually adapt the physical storage layout, we presented the *Storage Adaptivity-Specific Energy-Control Loop*, which leverages the implicit partitioning of our living partitions architecture to incrementally adapt the storage layout at runtime. We described how to integrate the different ECLs with each other and ended up with a sophisticated ECL hierarchy that is doing hardware and software-centric adaptations at runtime, while trying to stay within a user-defined query latency limit. Our evaluation showed that all

ECLs work hand in hand and we achieved superior energy savings and query latency improvements for various workload mixtures.

In our opinion, highly adaptive database systems are the only way to cope with the vast amount of application domains database systems are being exposed today. The scalable and adaptivity-enabling *Living Partitions* architecture as well as our *Adaptivity Facilities* are a first milestone towards such a highly adaptive DBMS, which opens up new horizons for further research.

References

[1] SFB 912: Highly Adaptive Energy-Efficient Computing. [Online] `https://tu-dresden.de/ing/forschung/sfb912`.

[2] I. Alagiannis, S. Idreos, and A. Ailamaki. H2O: a hands-free adaptive store. In *International Conference on Management of Data, SIGMOD 2014, Snowbird, UT, USA, June 22-27, 2014*, pages 1103–1114, 2014.

[3] J. Arulraj, A. Pavlo, and P. Menon. Bridging the Archipelago between Row-Stores and Column-Stores for Hybrid Workloads. In *Proceedings of the 2016 International Conference on Management of Data, SIGMOD Conference 2016, San Francisco, CA, USA, June 26 - July 01, 2016*, pages 583–598, 2016.

[4] R. E. Brown, E. R. Masanet, B. Nordman, W. F. Tschudi, A. Shehabi, J. Stanley, J. G. Koomey, D. A. Sartor, and P. T. Chan. Report to Congress on Server and Data Center Energy Efficiency: Public Law 109-431. 06/2008 2008.

[5] S. J. Finkelstein, M. Schkolnick, and P. Tiberio. Physical Database Design for Relational Databases. *ACM Trans. Database Syst.*, 13(1):91–128, 1988.

[6] IBM Software Group Information Management. Telecommunication Application Transaction Processing (TATP) Benchmark Description. 2009. [Online] `http://tatpbenchmark.sourceforge.net/TATP_Description.pdf`.

[7] S. Idreos, M. L. Kersten, and S. Manegold. Database Cracking. In *CIDR*, pages 68–78, 2007.

[8] M. L. Kersten and S. Manegold. Cracking the Database Store. In *CIDR*, pages 213–224, 2005.

[9] M. Korkmaz et al. Towards Dynamic Green-Sizing for Database Servers. In *ADMS*, 2015.

[10] I. Pandis et al. Data-Oriented Transaction Execution. *PVLDB*, 2010.

[11] D. Porobic, E. Liarou, P. Tözün, and A. Ailamaki. ATraPos: Adaptive Transaction Processing on Hardware Islands. In *IEEE 30th International Conference on Data Engineering, Chicago, ICDE 2014, IL, USA, March 31 - April 4, 2014*, pages 688–699, 2014.

[12] A. Shehabi, S. J. Smith, D. A. Sartor, R. E. Brown, M. Herrlin, J. G. Koomey, E. R. Masanet, N. Horner, I. L. Azevedo, and W. Lintner. United States Data Center Energy Usage Report. 06/2016 2016.

[13] Y. Tu et al. A System for Energy-Efficient Data Management. *SIGMOD Record*, 2014.

[14] G. Valentin, M. Zuliani, D. C. Zilio, G. M. Lohman, and A. Skelley. DB2 Advisor: An Optimizer Smart Enough to Recommend Its Own Indexes. In *ICDE*, pages 101–110, 2000.

[15] Z. Xu et al. Power-Aware Throughput Control for Database Management Systems. In *ICAC*, 2013.