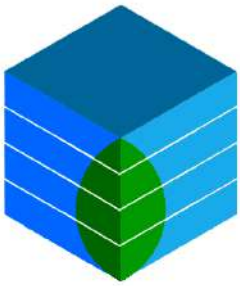

BigStorage

BigStorage: MSCA-ITN-2014-ETN-642963

Storage-based convergence between HPC and Cloud to handle Big Data

Deliverable number **D5.1**

Deliverable title **WP5 Energy Issues**

Intermediate Report

Main Authors **André Brinkmann, Nafiseh Moti, Michael Kuhn, Fotios Nikolaidis, Yacine Taleb, Álvaro Brandón, Athanasios Kiatipis, Yevhen Alforov**

Grant Agreement number	642963
Project ref. no	MSCA-ITN-2014-ETN-642963
Project acronym	BigStorage
Project full name	BigStorage: Storage-based convergence between HPC and Cloud to handle Big Data
Starting date (dur.)	1/1/2015 (48 months)
Ending date	31/12/2018
Project website	http://www.bigstorage-project.eu

Coordinator	María S. Pérez
Address	Campus de Montegancedo sn. 28660 Boadilla del Monte, Madrid, Spain

Reply to	mperez@fi.upm.es
Phone	+34-91-336-7380

Document Identifier	D5.1
Class Deliverable	Document
Version	1.0
Document due date	31 Dec 2016 (M24)
Submitted	30 Dec 2016 (M24)
Responsible	André Brinkmann, JGU
Reply to	brinkman@uni-mainz.de
Document status	Final
Nature	R(Report)
Dissemination level	CO(Consortium)
WP/Task responsible(s)	André Brinkmann, JGU
Contributors	André Brinkmann, Nafiseh Moti, Michael Kuhn, Fotios Nikolaidis, Yacine Taleb, Álvaro Brandón, Athanasios Kiatipis, Yevhen Alforov
Distribution List	Consortium Partners
Reviewers	Maria S. Perez
Document Location	http://bigstorage-project.eu/index.php/deliverables

Executive Summary

This document provides an overview of the progress of the work done until M24 of the Project BigStorage (from 01-01-2015 until 31-12-2016) in WP5 Energy Issues.

There is a pressing need for organisations to store and analyse the data that is constantly generated. This has hastened the growth of data centers without considering the energy consumption involved, becoming one of the most power-hungry industries of our time. That is why energy efficiency has become a rich and promising area of research. In this deliverable, the current techniques and models that are used to minimize power consumption at different levels are examined: from the server component level to the whole datacenter management. In addition, the techniques to storage less data and adapting energy consumption to meet SLAs are reviewed. After surveying the different solutions, the key challenges are identified and also gaps that still need to be solved are examined.

This deliverable is written in form of research paper, since it will be submitted to some journal for its publication.

Document Information

IST Project Number	MSCA-ITN-2014-ETN-642963	Acronym	BigStorage
Full Title	BigStorage: Storage-based convergence between HPC and Cloud to handle Big Data		
Project URL	http://www.bigstorage-project.eu		
Document URL	http://bigstorage-project.eu/index.php/deliverables		
EU Project Officer	Szymon Sroda		

Deliverable	Number	D5.1	Title	Intermediate Report on WP5
Workpackage	Number	WP5	Title	Energy Issues

Date of Delivery	Contractual	M24	Actual	30/12/2016
Status	version 1		final ■	
Nature	prototype □ report ■ dissemination □			
Dissemination level	public □ consortium ■			

Authors (Partner)	André Brinkmann, Nafiseh Moti, Michael Kuhn, Fotios Nikolaidis, Yacine Taleb, Álvaro Brandón, Athanasios Kiatipis, Yevhen Alforov		
Responsible Author	Name	André Brinkmann	E-mail	brinkman@uni-mainz.de
	Partner	JGU	Phone	+49 6131 3926390

Abstract (for dissemination)	There is a pressing need for organisations to store and analyse the data that is constantly generated. This has hastened the growth of data centers without considering the energy consumption involved, becoming one of the most power-hungry industries of our time. That is why energy efficiency has become a rich and promising area of research. In this deliberable, the current techniques and models that are used to minimize power consumption at different levels are examined: from the server component level to the whole datacenter management. In addition, the techniques to storage less data and adapting energy consumption to meet SLAs are reviewed. After surveying the different solutions, the key challenges are identified and also gaps that still need to be solved are examined.
Keywords	Energy efficiency, power consumption, storage systems

Version	Modification(s)	Date	Author(s)
01	First complete draft of the document	<29/12/2016>	André Brinkmann, Nafiseh Moti, Michael Kuhn, Fotios Nikolaidis, Yacine Taleb, Álvaro

			Brandón, Athanasios Kiatipis, Yevhen Alforov
02	Review	<30/12/2016>	Maria S. Perez

Project Consortium Information

Participants	Contact
Universidad Politécnica de Madrid (UPM), Spain	María S. Pérez Email: mperez@fi.upm.es
Barcelona Supercomputing Center (BSC), Spain	Toni Cortes Email: toni.cortes@bsc.es
Johannes Gutenberg University (JGU) Mainz, Germany	André Brinkmann Email: brinkman@uni-mainz.de
Inria, France	Gabriel Antoniu Email: gabriel.antoniu@inria.fr Adrian Lebre Email: adrien.lebre@inria.fr
Foundation for Research and Technology - Hellas (FORTH), Greece	Angelos Bilas Email: bilas@ics.forth.gr
Seagate, UK	Malcolm Muggeridge Email: malcolm.muggeridge@seagate.com
DKRZ, Germany	Thomas Ludwig Email: ludwig@dkrz.de

CA Technologies Development Spain (CA), Spain		Victor Muntes Email: Victor.Muntes@ca.com
CEA, France		Jacque Charles Lafoucriere Email: Charles.LAFOUCRIERE@CEA.FR
Fujitsu Technology Solutions GMBH, Germany		Sepp Stieger Email: sepp.stieger@ts.fujitsu.com

Energy-related Storage Research: WP5 Deliverable

André Brinkmann, Nafiseh Moti
Johannes Gutenberg University
Mainz, Germany
{brinkman, moti}@uni-mainz.de

Yacine Taleb
Inria Rennes Bretagne – Atlantique
Rennes, France
yacine.taleb@inria.fr

Athanasios Kiatipis
Fujitsu Technology Solutions
Munich, Germany
athanasios.kiatipis@ts.fujitsu.com

Michael Kuhn
Universität Hamburg
Hamburg, Germany
michael.kuhn@informatik.uni-hamburg.de

Álvaro Brandón
Universidad Politecnica de Madrid
Madrid, Spain
abrandon@fi.upm.es

Yevhen Alforov
Deutsches Klimarechenzentrum GmbH
Hamburg, Germany
alforov@dkrz.de

Fotios Nikolaidis
CEA/DAM/Ile-de-France
Paris, France
fotios.nikolaidis.ocre@cea.fr

***Abstract*—There is a pressing need for organisations to store and analyse the data that is constantly generated. This has hastened the growth of data centers without considering the energy consumption involved, becoming one of the most power-hungry industries of our time. That is why energy efficiency has become a rich and promising area of research. In this paper, the current techniques and models that are used to minimise power consumption at different levels are examined: from the server component level to the whole datacenter management. In addition, the techniques to storage less data and adapting energy consumption to meet SLAs are reviewed. After surveying the different solutions, the key challenges are identified and also gaps that still need to be solved are examined.**

I. Introduction

Service providers have equipped their data centers with millions of servers, in order to meet the needs of Big Data and large scale Web applications. The energy needed to operate these data centers, both for powering and cooling the equipment, has been increased up to alarming levels [15]. The escalation of the price of power, along with the campaign to reduce the carbon footprint of every power-consumption device on earth, has lead to the investigation of the power consumption of data centers.

As a result, tremendous research efforts have been dedicated to explore, model, and improve the energy efficiency of servers. According to [68], one can think of the energy efficiency at the node level in the following ways:

- Measuring and modeling the energy of nodes;
- Node level optimization (DVFS, software improvements, low-power/sleep states);
- Datacenter level power management (Scheduling, using green energy);
- Cloud and virtualization (VM consolidation, etc.)

In the following, we present an overview of the literature targeting the energy-efficiency in storage systems, according to the above categorization. The goal is not to be exhaustive (at all), but give an insight of contemporary state-of-the-art solutions.

II. Modeling and Measuring Power Consumption

Power consumption is a key issue today for every modern data center and cloud computing system [6]. As data storage requirements are exponentially growing, the development of scalable and efficient IT infrastructure for computing applications and services becomes more complex day by day. The need for robust computing systems with larger servers, powerful processing units and high speed network, with minimal operational cost becomes more evident than ever. Unfortunately, modern processors have already reached their physical limits making further shrinking extremely difficult [28], which on its turn makes further scale-up less efficient than before. To compensate, a great lot of smaller processor are used to provide the same computational power. Similarly, the trend of high-capacity resilient storage disks has been replaced by the trend of deploying more but smaller commodity disks.

Given the heterogeneous nature and scale of modern data centers, the electricity expenditures and carbon dioxide emission to the environment, have become an imminent issue that must be faced. In this way, balance between computing performance and energy consumption becomes a great challenge for the operation and maintenance of data centers and cloud computing systems. Reducing the energy consumption and its impact on environment and operational cost is a non trivial task that must be tackled in hardware, software and management levels.

On hardware aspect there have been efforts to optimize CPU power consumption based on the imposed load. Techniques involve: *dynamic voltage and frequency scaling* (DVFS, see section V), *dynamic concurrency throttling* (DCT) [53], integrated voltage regulators and improving the core of processors [39]. Although these methods can significantly reduce the

consumption, they are only limited within the scopes of the processing units, with the risk of eliminating all the benefits of the face of poorly written applications.

To go around that limitation, it is mandatory to deeply understand how different components of computing system (e.g. CPU and GPU [37], [44], storage disks [36], I/O [59], network, etc.) consume power. Collected information about energy usage can help developers to optimize and improve these systems. There are mainly two approaches [16] to get valuable information on energy consumption:

1) Hardware-based approach - measuring the voltage and current at different computing system units though special physical devices;
2) Using power consumption models or simulators which help to estimate and predict the energy consumption.

Both approaches are also utilized as a mix in hybrid methods. The information provided by them is valuable knowledge to implement *Power and Energy capping* management techniques [79]. These two techniques aiming to limit the amount of energy and power that computing system requires while executing applications within a given time.

A. Measuring

Managers of computing systems pay attention to the reduction of power costs and are interested in useful energy-saving methods as well as software engineers who want to improve energy efficient applications. Thereby measurement and monitoring of power consumption are necessary to evaluate energy efficiency at the various system layers. Obtained measurement values significantly help to discover the energy usage behaviour and to develop more effective power management techniques (e.g. power-aware job scheduling). Thus, many researchers and scientists have introduced different methods and measurement systems to retrieve and analyze energy consumption. Some of them provide measurement values by monitoring computing system work at operating system level or by getting information from hardware performance counters.

The most up-to-date review of different power measurement methods is presented in [16]. Authors describe in general existing measurement systems which are available today for current computing systems. Advantages and drawbacks of such measurement systems are discussed in the paper to give a clear understanding of what kind of scenario needs one or another technique. Authors provide comments on different method's implementation for energy consumption profiling and focus on *model specific registers (MSR)* approach which has been employed to monitor CPU power efficiency.

[37] and [44] give a survey of frameworks (*PowerPack, Intel Energy Checker SDK, PerfTrack*) based on the physical measurement approach and interfaces (*Intel's Running Average Power Limiting (RAPL)* and AMD's *Application Power Management (APM)*) which provide model-based power monitoring features of the latest processor generations. [44] focuses on the CPU and socket power consumption measurements. They present a custom solution for an accurate individual component measurement at the node level. In paper [37] authors decided to find out what measurement instrumentation provides the most accurate information about the compute node power usage during the execution of application within a given time slot. After performing an in-depth analysis of several different external AC and internal DC measurement methodologies on the test systems and verification of derived data, researchers conclude that the best results in computing of full node energy consumption can be received by using a professional power meter attached to the power supply units (PSU). Based on the experimental results authors say that the information obtained from modeling approaches of Intel and AMD provide data of a varying quality and should be improved in the future. Moreover they are limited to CPUs and potentially to DRAM.

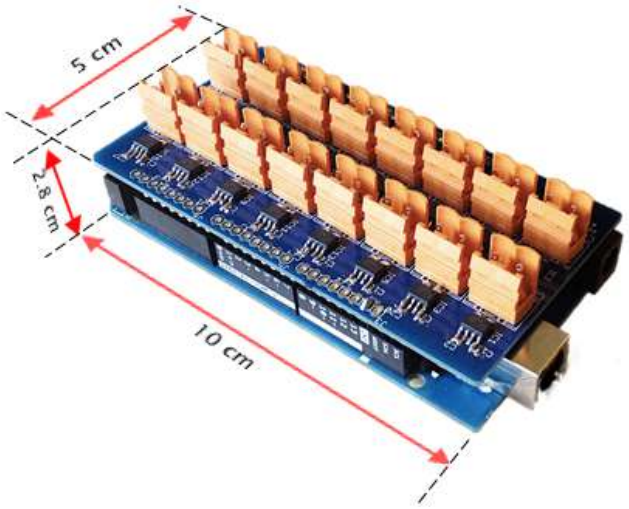

Fig. 1. ArduPower wattmeter, power sensing shield and Arduino Mega 2560 processing board [25]

One more work devoted to the power consumption measurement and energy efficiency of processors has been presented in [39]. This work focuses on new generation of Intel Hasswel processor and provide analysis of its improvements. Multiple *fully integrated voltage regulators (FIVR)* are now included into these type of processors and provide individual voltage for every core. The newly developed features enhance the optimization of such energy efficiency techniques as DVFS and DCT. In addition, information derived from Intel's RAPL model-based approach about power consumption has now a higher level of accuracy.

Even though there are various power monitoring devices that are presented nowadays in a market, Manuel F. Dolz et al. [25] introduced a new, accurate, small and low-cost (around 100 euros per device) wattmeter called *ArduPower*. This is an internal device which basically consist of a shield with 16 Allegro ACS713 current sensors and Arduino Mega 2560 processing board (see Fig. 1). It has been designed to simultaneously measure the DC power consumption of different components (e.g. motherboard, CPU, GPU, disks, etc.) inside of computing system even at very large scale.

ArduPower provides 16 channels to monitor the power consumption with a sampling rate varying from 480 to 5880 Sa/s. This power profiling tool's usage is beneficial for many real-world scientific HPC applications and has been successfully leveraged by Pablo Llopis et al. [59] to measure power costs during data movement (see more details in next subsection).

To evaluate the quality of existing power and energy consumption measurement infrastructures for HPC Hackenberg et al. [38] defined four criteria which include spatial and temporal granularity, scalability and accuracy. Authors introduce the *High Definition Energy Efficiency Monitoring (HDEEM)* infrastructure that fulfills and improves these requirements in computing systems. *HDEEM* aims to optimize energy-aware performance of parallel applications by decreasing the measurement errors (accuracy improving) during real-world application runs which is already less than 0.5% over 270 nodes. As the presented infrastructure based on a novel FPGA architecture, it is able to achieve higher spatial granularity by measuring blade, CPU, and DRAM power consumption separately and temporal granularity with a rate of 1000 Sa/s over 500 nodes. It should be noted that defined criteria must be always taken into account during development of a new measurement tool like it has been done in [25].

Besides the various energy profiling devices and tools there are many monitoring environments which provide not only information on power and energy consumption at different levels of a system, but also many useful features (e.g. visualization of collected metrics and retrieving the knowledge from them). The *Scalable I/O for Extreme Performance (SIOX)* project [51], for instance, provides I/O profiling environment with a Likwid-based plug-in that performs application's and system energy consumption tracking. It employs an intelligent monitoring to detect anomaly energy usage behavior. The knowledge about how power has been used through all computing system is very significant for power models in order to estimate the work of applications and predict their energy consumption before actual execution.

B. Modeling

Power models provide another kind of solutions to estimate energy consumption. Commonly, the estimation can be done by leveraging hardware performance counters with software interfaces which perform monitoring of system components usage at operating system level during the workloads execution [13]. However, many other types of power models can be also developed through using either hardware instrumentation or simulations. Power prediction and estimation models give an alternative to current and voltage meters because they are inexpensive, simple, accurate and portable [26].

Estimation of power consumption is highly required for power and energy capping techniques which aim to significantly increase the energy efficiency by limiting the amount of power/energy that can be consumed during the work of high performance applications. To be able to prevent possible power overloads within a limited amount of energy capping techniques demand the knowledge about computing system components power usage including the application behavior, I/O and running jobs at a node level [79].

An example of predictive model for energy and power consumption of applications running on HPC systems is presented in [13]. Authors propose a Machine Learning (ML) approach which takes as an input all available knowledge about previous application runs and therefore can learn a model to predict accurately the consumption of jobs in the future. It can be done by describing every job running on the computing system with a single approximate value of power consumption, mostly calculated as an average of all the power measurements during the job's lifetime. The quality of prediction depends on the quantity of collected historical data on application runs: the larger amount of information regarding user and application resource requests in the past provides higher model accuracy. Such prediction methodology is very important and necessary for job dispatchers as they have to make an optimal schedule for every application runs.

The similar approach of taking into account available historical information of an application's power and energy consumption has been employed in [79]. They introduced in this paper *Adaptive Energy and Power Consumption Prediction (AEPCP)* high-level model utilizes the knowledge about previous application power and energy behavior from current data center monitoring and resource management tools regarding the number of leveraged computing nodes. This model predicts future energy consumption of parallel HPC applications without any additional adjustments in their code with respect to the number a given compute servers. The associated prediction accuracy is improving with every additional application execution. Produced by AEPCP model results are application specific and can be obtained for any application which runs on HPC system.

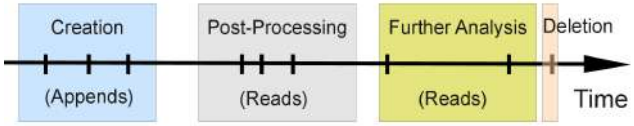

Fig. 2. A scheme of scientific data life cycle [52]

Kunkel et al. [52] introduce a new energy model which aims to quantify the amount of energy consumed for every scientific data file individually during its life cycle (Fig. 2). Having such information promotes the reduction of power costs mainly in the storage systems. The presented estimation model considers *idle* and *active* files and takes into account their migration and replication as well. As collecting of the energy consumption is not doable at the application level, authors suggest to modify the file systems to compute the consumption metric of individual file called the *Total Energy for the File Life cycle (TEFL)*. The technically possible modifications of file systems allow accurate TEFL estimation with I/O accesses accounting. For appropriate storage of obtained metric values authors propose to keep them in the Extended Attributes (EAs)

of file metadata which are supported by local file systems (e.g. ext3, ext4) and parallel file systems like GPFS and PVFS2.

Another research work which also considers I/O operations during analysis and modeling of power consumption in computing systems has been presented by Pablo Llopis et al. [59]. While most of surveyed works are limited to single system's component analysis, these authors offer a study on power usage and energy consumption of the complete I/O stack caused by intra-node data movement across all components. The cost of data movement in modern computing systems us a key issue for I/O-intensive scientific applications and simulations. Better understanding of power consumption through I/O workloads is necessary to develop effective power models and to significantly optimize the energy efficiency of storage technologies. The authors developed the *Pyprocstat* framework to collect system metrics and leveraged ArduPower [25] internal wattmeter (discussed in the previous subsection) to measure a power. They designed a methodology based on the obtained measurement values and applied it on sequential read and write operations. As a result, the provided model can determine which one of system metrics is highly correlated with a power consumption during data movement and I/O workloads. Identification of such metrics is required for development of new sophisticated power models.

The approach that was used to build the aforementioned model has been introduced in Manuel F. Dolz et al. [26]. This work presents an analytical methodology for building simplified and accurate power models when determination of hardware and software metrics is important. The aim of this portable methodology is to derive reduced power models by analyzing all computing system's components through execution of standard benchmarks. These platform dependent models collect all necessary metrics from hardware counters and resource utilization information to perform power estimation at node level. Authors of this methodology state that in the future such models can replace physical wattmeters used on HPC systems.

III. The Impact of New Storage Technologies on Power Consumption

Online storage and main memory systems have long been dominated by magnetic disks and DRAMs. The last years have seen a transfer from these mechanical devices to non-volatile random-access memory, including flash storage and storage class memory like phase-change memory or magnetoresistive random-access memory (MRAM). A huge advantage of these technologies is that NVRAM consumes less power and space and produce less heat.

NVRAM bridges the performance gap between fast, volatile RAM and a traditional disk-based storage, which is orders of magnitude slower. For this reason, NVRAM devices are emerging as disruptive storage for high-performance computing systems. NVRAM can take part in different components of memory hierarchy from cache to main memory and secondary storage. In part A we will introduce different technologies that are called non-volatile memories. In part B we review some approaches for using these new technologies as the secondary storage and in part C we explore possibilities for exploiting these technologies in other parts of the memory hierarchy as the main memory or cache.

TABLE I
A COMPARISON OF DIFFERENT MEMORY TECHNOLOGIES [49]

Device Type	HDD	DRAM	NAND Flash	FRAM	MRAM	STTRAM	PCRAM	NRAM
Maturity	Product	Product	Product	Product	Product	Prototype	Product	Prototype
Present Density	400Gb/in^2 [7]	8Gb/chip [9]	64Gb/chip [10]	128Mb/chip	32Mb/chip	2Mb/chip	512Mb/chip	NA
Cell Size (SLC)	(2/3)F^2	6F^2	4F^2	6F^2	20F^2	4F^2	5F^2	5F^2
MLC Capability	No	No	4bits/cell	No	2bits/cell	4bits/cell	4bits/cell	No
Program Energy/bit	NA	2pJ	10nJ	2pJ	120pJ	0.02pJ	100pJ	10pJ [11]
Access Time (W/R)	9.5/8.5ms [8]	10/10ns	200/25us	50/75ns	12/12ns	10/10ns	100/20ns	10/10ns [11]
Endurance/Retention	NA	10^{15}/64ms	10^5/10yr	10^{15}/10yr	10^{15}/10yr	10^{15}/10yr	10^5/10yr	10^{15}/10yr

Device Type	RRAM	CBRAM	SEM	Polymer	Molecular	Racetrack	Holographic	Probe
Maturity	Research	Prototype	Prototype	Research	Research	Research	Product	Prototype
Present Density	64Kb/chip	2Mb/chip	128Mb/chip	128b/chip	160Kb/chip	NA	515Gb/in^2	1Tb/in^2
Cell Size	6F^2	6F^2	4F^2	6F^2	6F^2	N/A	N/A	N/A
MLC Capability	2bits/cell	2bits/cell	No	2bits/cell	No	12bits/cell	N/A	N/A
Program Energy/bit	2pJ	2pJ	13pJ	NA	NA	2pJ	N/A	100pJ [12]
Access Time (W/R)	10/20ns	50/50ns	100/20ns	30/30ns	20/20ns	10/10ns	3.1/5.4ms	10/10us
Endurance/Retention	10^9/10yr	10^5/Months	10^9/days	10^4/Months	10^5/Months	10^{15}/10yr	10^5/50yr	10^5/NA

A. Non-Volatile Memory Technology Overview

The NVM technologies are not new and they were offered in early super computers but the initial design was not optimized for energy efficiency [63]. But with advances in technologies they have become more reliable and energy efficient. Kryder et al. [49] studied these technologies with regard to their hardware characteristics and speed. You can see the comparison between these technologies in Table I. Since some of these technologies are out of scope of this study we just mention the most important ones in our context.

The most common type of NVMs are NAND Flash memories. These memories are mature now and are being developed by many vendors. Flash memories are mostly used in solid state drives. Since these drives don't have moving parts, and are resistance to shock and heat they are a good replacement for HDDs. Agrawal et al. [2] studied the characteristics of SSDs and showed that different hardware software designs would affect the performance of SSDs. NAND flash memories suffer from degrading and failing over time with so many writes. This is a reason why researches in SSDs and Flash memories are focusing on wear-leveling and write minimization.

Another class of NVMs are storage class memories or SCMs which includes different types of technologies such as PCM, STT-RAM and ReRAMs. The main difference between these SCMs and Flash memories is that they SCMs byte addressable. SCMs can offer near DRAM read speed and have better endurance than Flash memories.

B. Non-Volatile Memories As A Secondary Storage

There are several studies which exploit NVMs as secondary storage in HPC systems. These studies mostly focus on using the best features of HDDs and NVM technologies for replacing

HDDs as the main storage, checkpointing or as metadata storage in HPC filesystems.

Narayanan et al. [67] analyzed the performance and energy consumption of SSDs when being used as the main storage in HPC systems. They have shown that the energy price which would be saved over 5-years of using SSDs would compensate the additional purchase cost. SSDs can also be used alongside with HDDs in order to exploit the advantages of both. Kim et al [47] suggested a hybrid HDD-SSD model which dynamically decides about providing service to requests. It also migrate pages between HDD and SSD to achieve better configuration in term of SSD write minimization.

The other researches on SSDs are focused on reducing write traffic, Huang et al. [40] classified blocks to semantic blocks and data blocks they used deduplication techniques on data blocks and encryption methods on semantic blocks. Some researchers focus on changing FTL and translation methods to improve the reliability of Flash memories and SSDs. Lu et al. [60] proposed an object-based FTL layer and used lazy indexing to minimize write and provide a byte addressable layer to the application.

Dong et al. [27] suggests to use PCMs instead of HDDs to improve checkpointing performance. Since the energy consumption of checkpointing depends on the checkpoint storage hardware replacing HDDs with PCM can improve checkpointing energy efficiency. Another scope in which the HPC systems can explpoit NVMs is using them as burst buffers. Current HPC architectures do not include HDDs in local nodes. Instead they try to offload their I/O over network which leads to congestion in the bandwidth of the system. To improve scalability and performance of these system different NVM technologies are used as a fast and energy efficient storage which enables applications to save their current state and checkpoints on the local node storage [57].

There are also other opportunities to use other types of NVMs especially PCMs in storage systems. Sun et al. [82] proposed a Flash-PCM hybrid architecture to improve energy consumption and performance of storage system. They used PCM as a log region for NAND Flash, and exploited the in-place updating of PCMs for improving energy efficiency and reliability of Flash storage. There are also works which used PCM in storage system as a faster layer between HDDs and SSDs. Kim et al. [46] exploited PCMs in the storage hierarchy alongside HDDs and SSDs and explored different combinations of these three to achieve higher cost/performance ratio.

C. Hybrid Memory Hierarchies

One of the main components which affects system's overall power consumption is main memory. Two components of today's memory hierarchy are caches and main memory. There are many opportunities for using NVMs in the memory hierarchy from strict all persistent memory systems to deploying them as a replacement for different layers in the system's memory or using these technologies alongside an existing layer.

Narayanan et al. [66] proposed a system where all memory is non-volatile. This helps applications to recover fast after a system failures, in their approach the transient state of the system is flushed from cache and registers only on failures. This method reduces cache flush overhead while improving the performance of suspend/resume events. They also mentioned the cases when only certain parts of the memory is persistent, "dangling references" might happen which refers to the non-volatile references which point to the volatile ones after a crash.

A number of works on using NVMs in memory hierarchy, combine different technologies and exploit advantages and potentials of each type of memory alongside others to achieve the best performance. Qureshi et al. [73] introduce a PCM based hybrid memory system where a small DRAM buffer is used as a supplement for the memory. This architecture benefits from lower latency of DRAM and capacity of PCM. They used different policies for migration and writing to memories.

IV. Saving Power by Storing Less Data

More and more HPC applications generate vast amount of data sets especially when dealing with information produced by simulations or program runs of climate change, weather forecasting, seismic and other scientific models. The tremendous growth of these digital data that is observed today leads to the increasing of storage, energy and overall costs for every data center. Thus, different techniques and algorithms which can significantly reduce the power consumption and improve storage efficiency are highly demanded.

A straightforward approach for saving energy in storage systems is to reduce the amount of data that is stored. For storage systems handling live data such as parallel file systems, less data directly results in less storage hardware that has to be procured and operated. Since the hardware used in such systems (for example, SSDs and HDDs) is typically not powered down completely, this results in energy savings. However, even for tape systems, costs can be reduced.

To be able to deploy data reduction on a storage system level, only lossless data reduction techniques can be considered. Two techniques that can be deployed in a way that is transparent for users of the storage systems are deduplication and compression. In this section, these two techniques will be discussed. Each of them has its own basic concepts, advantages and disadvantages which are summarized below. Besides HPC, these approaches have also found their applicability in game modeling, sensor networks, cloud computing, etc. Both data deduplication and compression provide the base for further scientific research work in order to improve existing and acquire new energy saving solutions. In common, the main benefits that bring the usage of these techniques are storage capacity optimization, network bandwidth reduction, reduction of operational costs and energy saving.

A. Deduplication

It can happen that different data sets contain fully or partially identical pieces of data. This situation usually occurs after data backup or recovery processes but can also appear accidentally because of inefficient coding or computation complexity. Deduplication provides an approach where duplicated copies of repeated data are detected within a dataset or a file system (in case of file redundancy) and replaced by a pointer to original data or a file [91]. It works by splitting up data into (possibly variably-sized) blocks and storing each unique block of data only once. Duplicate data is not stored explicitly but instead a reference to the original block is created.

Generally, the common process of data deduplication consist of 4 major steps [61], [91], [92]:

1) *Chunking*
   At first, the deduplication system splits the incoming data (e.g. files, database snapshots, virtual machine images, etc.) into equal or similarly sized multiple parts called *"chunks"*. The size of chunks impacts the deduplication ration and the overall performance of this data reduction approach. Depending on the chunking granularity exist several methods which can be used for the first step. Among them are whole-file chunking, fixed-size chunking, content-defined chunking, and Two-Threshold Two-Divisor (TTTD) method. More detailed discussion about these chunking methods and their algorithms performance is outlined in [86].
2) *Fingerprinting*
   On the second step each data chunk is identified by using a cryptographically secure hash signature (e.g. SHA-1 or SHA-256 algorithms). Every unique hash value calculated by the cryptographic function is called *"fingerprint"* and is necessary in the next step of deduplication process. In this way traditional byte-by-byte comparison is avoided.
3) *Fingerprint lookup*
   Calculated fingerprints are used in the lookup process to identify which data chunks are redundant or unique. The comparison is based on the idea that two fingerprints are the same only in the case if two corresponding data chunks are the same.
4) *Storage management*
   In the end, after the uniqueness verification of data chunks, deduplication system gets rid of redundancy by removing the duplicated data chunks. Only new data which is nonduplicate is stored on a disk (or transfered via network) and the reference to these data is provided instead of duplicated ones. In case if some application tries to modify the original data part, it is copied into the modification environment to let other applications which depend on the same data continue to work with unmodified part. Such method excludes the data corruption.

Deduplication strategies [92] can be different depending on:

- range - deduplication can be local (when a single data sources is employed) or global (different data sources are integrated before being used);
- position - client-side (when network bandwidth reduction is needed whole process is performed on a client side before data transferring) or server-side (in case when data have been already stored on the server storage system) are distinguished;
- time - data reduction can be performed as In-line (or real time reduction when data is written to the disk) or Off-line (post-processing deduplication after whole data were saved).

Data deduplication has been widely used storage saving technique in various computing systems. Commonly, it provides many benefits for backup, archive and network environments (secondary storage) but is also successfully adopted by primary storage, cloud storage, virtual machines and others. A study conducted using the online file systems of four HPC data centers (BSC, DKRZ, RENCI, RWTH) shows that 20–30 % of their data can be removed through deduplication [61]. Even using full-file deduplication, it is still possible to reach savings of 5–10 %.

From the other side data deduplication approach has problems which computer scientists aim to solve in the future. Most of these concerns are caused by consuming more system resources and time. Deduplication remains a costly technique as it requires additional memory and CPU performance during the data reduction and needs a lot of time while working with large datasets.

The main drawback of deduplication is the fact that the use of all data blocks has to be tracked in the fingerprint index. For fast access, these tables are typically supposed to be stored in main memory or at least low-latency storage devices such as SSDs. These tables are usually in the range of 10 GB per 1 TB of data, which is especially problematic for very large file systems. The analysis in [50] shows that the amount of additional main memory can actually increase energy consumption, rendering deduplication unviable.

In usual practice, to achieve costs reduction goal at a high ratio data deduplication is often used in conjunction with other forms of data reduction such as compression which will be discussed further.

B. Compression

Another type of data reduction techniques which provides storage capacity saving, faster data transferring and helps to decrease the energy costs is data compression. Programs which perform this technique are using special compression algorithms (e.g. LZW, DEFLATE, lz4) that determine the way the data size should be reduced. There are two types of data compression algorithms that can be distinguished:

- **Lossless** algorithms provide the possibility to restore the data exactly how it was before the compression process without any loss of information. This is a typical approach used by many applications which work mostly with texts, source codes and rarely with images.

- **Lossy** algorithms eliminate less important or unnecessary bits of information during the compression process. In this way the size of data compression output is much smaller, however data is not equal to the original one after decompression. This type of algorithms is mostly adopted by programs which work with multimedia (video, graphics, images).

While compression allows reducing the amount of data, it is important to keep the compression algorithms' overhead in mind. Fast and modern algorithms such as lz4 and zstd offer high throughputs, their compression ratios are typically not as good as those of slower algorithms such as DEFLATE. However, DEFLATE can usually not be used for HPC workloads due to its low throughput because the decreased throughput results in longer job execution times and consequently an increase in energy consumption. The impact of compression on I/O throughput is studied in [89]; the results show that the achievable throughput is highly dependent on the chosen algorithm and data as slow algorithms or incompressible data can decrease throughput significantly. One way to compensate for this drawback is to implement these algorithms in hardware. In [1], the authors have implemented gzip on FPGAs using OpenCL. Their implementation offers a throughput of 3 GB/s in comparison to 300 MB/s for a highly-optimized CPU implementation. Moreover, the FPGA implementation's performance-per-watt ratio is twelve times better than that of the CPU implementation. Not all accelerator-based implementations are faster than CPU-based implementations, though. In [71], the authors have implemented bzip2 on an NVIDIA GTX 460 and found that their implementation is more than two times slower than the original. One of the reasons for this is the fact that all data has to be transferred via the PCIe bus to the GPU and back.

Compressing the data already on the compute nodes allows increasing the network's throughput, which can also result in energy savings. By closely integrating technologies, data has to be compressed only once on the compute nodes and can then be stored in its compressed form by the storage servers.

For instance, there are several approaches to virtually increase network throughput by compressing messages sent and received via the MPI (Message Passing Interface). CoMPI adds a compression layer to the ADI of MPICH to compress MPI messages [31]. While the study shows that this is beneficial in many cases, the evaluation has been performed using Fast Ethernet and its applicability to recent HPC systems is thus limited. Adaptive-CoMPI is an improved approach that allows developers to specify guiding information and uses this information to select the compression algorithm at runtime [32]. The evaluation shows that Adaptive-CoMPI is able to improve throughput for some of the tested scientific applications and may only degrade performance slightly While the previous implementations are bound to specific MPI implementations, the PRAcTICaL-MPI wrapper transfers the strategy to a library that utilizes the MPI standard profiling interface (PMPI), which can be deployed without changing code or MPI implementation [30].

There is a wide range of compression algorithms. General purpose compression algorithms such as DEFLATE consider data to be an array of characters. However, applications usually make use of and store more complex data types and compound structures. Therefore, more specialized algorithms are often used to improve compression ratios. For instance, several lossless compression algorithms specifically tailored to floating-point data are available. In [75], a compression algorithm for arrays of 64-bit floating-point values is presented. It predicts the next value in the array based on previous values and uses XOR to encode the difference between the predicted and actual value. Therefore, the algorithm's efficiency greatly depends on its predictor. The authors of [56] present another compression algorithm that is aimed at improving I/O throughput to eliminate application stalls. Application stalls lead to idle CPUs, which in turns needlessly wastes energy. While the algorithm is slightly slower than the one in [75], it provides significantly higher compression ratios and can therefore improve throughput.

As mentioned before, the efficiency of algorithms aimed at floating-point values often heavily depends on the quality of their predictors. The authors in [42] present a compression algorithm that is specifically targeted at climate data and can therefore exploit knowledge about the data structure. For example, the values for land regions do not change when only the values for ocean regions are computed. This additional domain knowledge allows improving compression ratios even further and can therefore be used to reduce energy consumption.

Another work [20] presents decision algorithm for MapReduce users to decide whether to use compression or not. The key factor here is a data compressibility which determines the cases when compression is worthwhile. The impact of compression on performance and energy efficiency for MapReduce data-intensive workloads has been analyzed by the authors. The results of their analysis showed that compression provides up to 60% energy savings for some jobs.

In order to improve the performance of data compression techniques computer scientists offered several approaches. Most of them are based on the idea of porting compression methods to accelerators. The implementation of compression algorithms in hardware [10] increases speed, minimizes energy consumption needed for compressed data transferring and reduces memory traffic.

C. Durability kills Energy Saving

Although the aforementioned techniques can significantly improve storage savings, if deployed on a real long-running system, the disaster is inevitable since failures are not taken into account. If data are tampered due to medium failure the reconstructed data will not be the original. If network or storage nodes becomes unavailable, data will not be retrievable. Further studies [83] have been focused on enlightening the relation between energy saving and failure rate. Voltage undervaluing reduces the energy consumption, but at the same time increases the failure rate. All the above, make clear the need for data redundancy. Redundancy in storage systems can

be implemented through replication, parity schemas or erasure codes, with each method having its benefits and drawbacks. Replication consumes more storage space but data can be directly fetched from multiple sources, improving the overall performance. Parity schemas occupy less storage than full replicas, but can only tolerate small numbers of concurrent failures. Erasure codes fits between the two. Requires less space than replication, for the same reliability and availability levels, can tolerate more failures than parity schemes, but involve coding and decoding overhead.

The impact of redundancy configuration on disk energy consumption has been extensively investigated in [72]. The main contribution is called diverted access model, and is motivated by the key observation that redundant data are only read 1) during periods of high demand for disk bandwidth, to increase performance 2) when disk failures occur, to guarantee reliability and availability.

The goal of diverted access it to keep redundant disks in low-power mode without compromising neither performance nor availability. To achieve it

- Reads are directed to the original disks only, unless redundant disks are needed to be activated to offload the original disks
- Writes are performed on all disks, on high load. On light or moderate load, only m disks are synchronously written with the write propagated to the rest $m-n$ disks asynchronously.

Generalizing the above, if replication is accounted for a single, isolated datacenter, the most commonly deployed technique is to turn off underutilized storage servers, with regard to the replica availability [83]. However, if the replication across data-centers is the goal, the request is to minimize the power consumption of the backbone network [14]. Proposed methods estimate the number of required replicas as a factor of data loss probability (e.g storage failures, network outage) to the required reliability model. For that model, replica placement decisions can be made based on runtime analysis of the data popularity, in terms of access patterns, within a time-window.

To improve the performance and minimize the latency, usually the data storage resources are brought closer to the computational infrastructure, where the applications are running. Although this strategy aims at increasing the overall system bandwidth and data availability, it does not take into account the energy consumption factors like:

1) Electricity cost on different geographical location
2) Energy efficiency of the target infrastructure
3) Overhead of maintaining exact replicas across locations

V. Adapting Power Consumption and Performance Needs

A. Tranferring the idea of Dynamic voltage and frequency scaling (DVFS)

DVFS is a technique successfully used in microprocessors, which consists of tuning up and down the voltage level and frequency of the CPU according to its actual processing requirements [21], [77]. The impact on the dynamic processor power consumption can be directly derived from the (slightly simplified) equation $P \propto V^2 \cdot f$ [34], showing the quadratic impact of the voltage level V and the linear impact of the frequency f on the overall power consumption P. Concepts closely related to DVFS have also been applied in storage related areas, like Big Data processing [43].

Nevertheless, it has been shown that the energy consumption of server environments is, even when applying DVFS, often not proportional to the currently required CPU performance [7]. The server systems, which have been investigated in 2007, still used more than 50% of their peak power consumption even when nearly running in idle mode. The typical operating region of these servers has been between 15% and 55% of their peak performance, leading to huge wastes of energy. The basic idea of power-proportional (storage) systems is therefore to design server (and storage) systems, which do only require power which is proportional to their actual performance. The power consumption of power-proportional storage is therefore allowed to be higher if more data is moved and should be significantly reduced in idle periods.

Transferring DVFS to (magnetic) storage systems is not straightforward. Magnetic disks have to run at a constant speed to enable the disk heads to read or write data on a track. Even the static power consumption of spinning disks at this constant rate already consumes between 50% and $2/3$ of the disks maximum power consumption [18] [36].

Completely turning off magnetic disks on the other side leads to unwanted side-effects and it can take between 3 and 15 seconds until the first byte can be read again after the platters have completely stopped rotating [22]. The characteristics of some application areas can be used to turn off disks to safe power and energy. An application domain is, e.g. the archiving of data, where data is often considered to be mostly written and rarely read. *Massive Arrays of Idle Disks* or MAID systems can be built, where the initial MAID approach uses a number of cache drives, which are always turned on, and data drives, which can be turned-off after they have been inactive for a specified time frame [22] (see Section VI for a more detailed discussion of the different approaches to build active and passive archives).

Chen et al. instead proposed to use the acoustic modes of disk drives to reduce the actual power consumption [18]. The seek speed can be reduced in the quiet mode, so that the overall power consumption of a disk can be capped. They recommend to use these modes to cap power if the overall power budget within a data center would be, but they also mention that the total energy to perform a task can be higher if the quite mode is applied.

B. Power-proportional Storage and Dynamic Power Management(DPM)

The Dynamic Power Management (DPM) is a method that increases energy efficiency by shutting down components that are idle or partly utilized [11]. This method could enable a system to become energy-proportional, meaning that it

will consume energy according to the workload required [8]. Specifically for hard disks (HD), this could be applied to the power required by components such as the HD motor, the controller, for seeking and for data transfer. The application of DPM is based on two assumptions. Firstly, on the fact that most systems encounter nonuniform workloads. Secondly, on the fact that the upcoming workloads can be possibly predicted. The typical operation state of a disk is between the *active* and the *idle* mode. The DPM also adds the *standby* mode to the available options. The difference between the idle and the standby mode is that the latter stops the disk spin and lifts the head off the platters, putting it in a resting position. There are various DPM techniques that are stated below. The first technique used is the predictive technique. In a system, the future, input events are not known and the predictions are made based on uncertainties. Generally, a predictive technique examines the correlation between the past history of the event and its near future. The most frequent technique is the fixed timeout, which analyses the idle time of the machine, in order to make decisions on whether the machine should be turned off. The second technique used is the adaptive. When the work that is being performed is not known beforehand, the "static" predictive techniques are not capable to improve efficiency to a great extend. In this case, a flexible and adaptable technique is required. For example, in the timeout case, several different timeout times are used, based on how successful each one was on previous uses. Simply, this is a creation of a table with many different scenarios, where each scenario is used accordingly. The third technique is the stochastic control. This technique does not use prediction, but develops policy optimization under uncertainty. An example is to examine the whole power-management with Markov processes [76]. By using this technique, the model can find trade-offs between performance and power consumption, find globally optimum policies for power control, design complicated systems with many power states (not only on and off but intermediate states too) and finally to create a model that analyses the uncertainty in power consumption.

In a different scope, an apparent solution for better energy efficiency is the usage of multispeed disks. Carrera *et al.* [17] proposed the solution of modulating the disk speed according to the load. This could hamper performance but it will benefit the energy efficiency.

1) Rabbit: Rabbit has been one of the first systems to transfer the idea of power-proportional systems to storage systems [4]. The main idea behind Rabbit is to shut down as many cluster-based storage nodes as possible to save energy. To do so, they propose a data-layout that is able to keep the same performance with a subset of nodes. More precisely, Rabbit proposes to keep M replicas only on a subset of nodes then $M-1$ on the next subset, and so on. By doing so, the system can enforce an equal work distribution and therefore it achieves power-proportionality. An important aspect in Rabbit is its fault-tolerance. Availability and durability are still achieved, as powered-down nodes receive replicas in an asynchronous way, i.e., a set of nodes selected among the powered-up nodes are receiving replication requests on behalf of these powered-down nodes, and forward these replicas asynchronously.

2) Sierra: Sierra achieves power-proportionality in a very similar way to Rabbit [85]. While the overall approach and the data layout are quite similar to Rabbit, it also introduces a module that has knowledge of the history of the I/O load and that tries to predict the load. By doing so, it is capable of predicting the number of required nodes to power-on/off in order to have the best energy consumption.

VI. Energy-optimized Storage Systems

A. General-purpose storage systems

FawnKV tries to answer this simple question [5]: Can we build a cost-effective cluster for data-intensive workloads that uses less than a tenth of the power required by a conventional architecture, but that still meets the same capacity, availability, throughput, and latency requirements? To do so they build a log-structured storage system based on flash and low-power embedded CPUs. They prove that it is still possible to saturate the I/O capability of their hardware.

Knightshift provides a number many interesting contributions [90]. First it introduces a set of metrics very useful to assess the energy efficiency of a system. Afterwards, it introduces Knightshift, a system based on the assumption that today's multi-core machines have a close power consumption either at low or high utilization. It uses two kind of machines, low-spec machines and multi-core ones. The low-spec machines exhibit lower power consumption and are used to serve workloads, until a certain threshold, then only, the multi-core machines (that are put into idle mode) will be woken up to serve requests.

B. Active archives

Due to a vast amount of information generated by Big Data analytics applications, modelling and simulations there is an extreme need in storage systems designed for efficient, reliable and long tern data archiving. One of such huge systems becomes an *Active Archive* when the data stored in it is accessible at any time. Every archive system currently faces with a growing requirements associated with reliability, low power consumption, easy maintenance, capacity and high performance within a given short time slot. Widely used archiving storage systems nowadays rely mostly on sequential-access tapes, which unfortunately can not fulfill all the mentioned above requirements in an adequate manner. They are not suitable for all use cases, provide poor random-access, have ineffective performance during auditing, searching, checking and other operations. Thereby, arose a need to replace tapes with other more energy efficient and reliable long-term storage architectures. The investigated solutions proposed by researchers are based on MAID (Massive Arrays of Idle Disks) and RAID (Redundant Array of Independent Disks) system architecture schemes.

Mark W. Storer et al. [81] introduce a solution for replacing tape archives with a help of *Pergamum* disk-based

storage. *Pergamum* is a distributed network of intelligent devices (called "tome") which provides reliable, energy efficient archival storage. Every Pergamum Tome, consist of four hardware components: a commodity hard drive for persistent, large-capacity storage; on-board flash memory for persistent, low-latency, metadata storage; a low-power CPU; and a network port.

Another disk-based storage solution has been presented in Matthias Grawinkel et al. [35]. Authors introduced well-suited for large scale archives RAID scheme called *LoneStar RAID* that can be used for cold and active long-term storage environments. *LoneStar RAID* provides fault-tolerance, strong reliability mechanisms and at the same time low energy consumption. It has been designed for archival workloads when data access is performed rarely.

VII. Storage Management and Cloud Computing

The computing power of the cloud can be used to run Big Data workloads in an effortless and more accessible way. These workloads consist of tasks running in VMs that are assigned to different machines. There are several techniques in the literature that focus on saving energy when executing them. One of the most popular is VM consolidation. Through bin packing algorithms, the user can concentrate the load in a set of machines and turn-off spare servers [55].

When allocating resources to workloads in data centers there is an inevitable tradeoff between the throughput, quality of service and energy consumed. More virtual machines will mean more workloads executed, but also higher power consumption and vice-versa. Some work focus on this area and tries to achieve a balance between the three. In [58] the authors model the behavior of the virtual and physical machines applying machine learning with information from previous executions and metrics collected with Nebula [80]. An example of this metrics include number of requests, average requested bytes or bandwidth. Initially, linear regression and MP5 are used to predict information that is not available or it is uncertain like the CPU, IO and memory load of the virtual machines currently running. These previous results together with the load of the VM's running inside the same physical machines are used to predict the SLA agreement level of a VM. With all this information, accurate decisions can be made regarding migrating virtual machines or adjusting the granted resources to them.

The power consumption of Hadoop workloads in the cloud has also been evaluated in [29]. The authors evaluate the impact on the performance by running a TeraSort Benchmark, both in physical and virtual clusters. Also, two different scenarios are shown where data is either colocated and separated from the compute services. The results show that separated data increases the energy consumption and that this degradation depends on the data to compute ratio, application and data size. It also shows that a high number of map slots in comparison to reduce slots creates idle time and therefore energy waste. Another interesting observation is the energy fluctuation across the map, shuffle and reduce phases. As the reduce and map phases completes, power consumption decreases and during the shuffle there is a drop in energy consumption.

An additional line of work is to use renewable energies to power data centers for cloud computing [24]. This poses several challenges, since renewable energy sources like solar or wind are not stable. Cloud computing requires constant availability and reliable services. In addition the demand is dynamic but renewable energy cannot be scheduled at certain times. To tackle this challenges a series of approaches are shown. Prediction models for renewable energy focus on scheduling workloads depending on the forecasted power generated through green sources like sun or wind. Capacity planning helps in evaluating the energy that is going to be needed for one given workload and allocating both renewable and fossil energy sources to meet their demand. In contrast intra-datacenter workload schedulers aim to find scheduling strategies that maximizes the use of renewable energy sources. Some of these strategies rely on the trade-off between energy consumed and the job's deadline, changing the server power state (e.g. DVFS) or load migration. Finally, inter-data center load balancing takes advantage of some characteristics of geo-distributed data centers, like scheduling and migrating workloads to the locations where there is sun or wind available. Lately, Microsoft and Google built a data center facility close to the Columbia River (Washington, USA), utilizing the cold air for inexpensive cooling and the power from a nearby hydroelectric dam [3]. GreenHadoop is another system that aims to use renewable energy [33]. MapReduce jobs are scheduled so they can use as much green energy as possible while meeting their deadlines. This renewable energy comes from a photovoltaic solar array. If the job will not complete within its bounds then brown energy is taken from the electrical grid to meet the SLA.

Parallel applications in the cloud and energy consumption reduction have also been studied in [41]. Again the trade-off between SLA and energy consumption is used to build a DAG of n vertices that represent tasks and edges representing its precedence constraints with a weight that denotes its communication cost. By scaling the supply voltage and clock frequency of the tasks that are not in the critical path we can save energy while still meeting the deadline like in Figure 3. Also the algorithm tries to schedule tasks near each other on a uniform frequency to get better results.

GreenCloud [48] provides a simulator to evaluate the energy consumed in data centers. To do that it divides the elements of a data center into the follow sections: Servers, switches and links and workloads. Through a fine grained modeling of the energy consumption of these three previous elements and an extension to a network simulator different strategies like voltage scaling, frequency scaling or shutting down machines/network components can be evaluated. In [23] the authors provide an analysis on the power consumption of a Hadoop sentiment analysis workload. They measure the power consumption directly with an external device and perform an analysis that considers three aspects of power

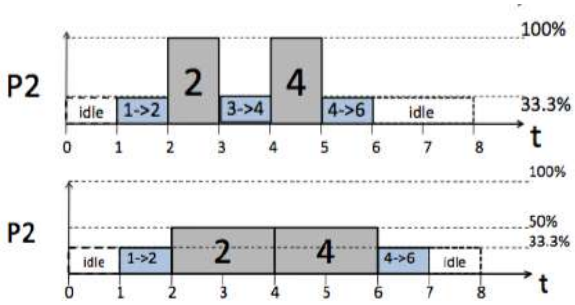

Fig. 3. An example of how DVFS can be used to minimize energy consumption and meet job deadlines in [41]. The upper caption shows a workload consisting of a series of tasks and the voltage levels with the default assignment. The bottom picture shows how the slack time can be leveraged to decrease voltage levels and still meet the 8 seconds deadline

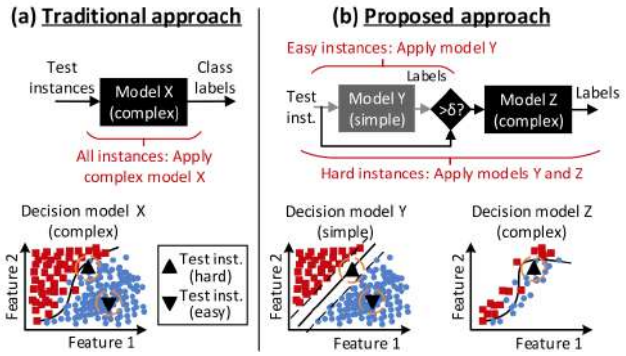

Fig. 4. An example of the approach in [87]. The traditional approach on the left learns the model for all of the input data, whereas the right one applies a simple model and then a complex model to only some data

consumption: the power consumption when turning on/off machines, the maximum and minimum values and the effect that the workload has on the a single node running several virtual machines. These analysis lead the authors to define six different power consumption profiles (e.g standby, idle, VM's running, etc...). Finally it provides an analysis that considers a tradeoff between energy and latency when adding nodes. The conclusion is that there is a point where adding nodes does not provide much benefit to the processing time but increases the energy consumption posing a challenge when finding a sweet spot between the two.

Other lines of work focus on the hardware used, mixing high-performance nodes with low energy consumption nodes [74]. By modeling the type of workload the execution time and energy consumption can be predicted. Then a combination of machines can be found that minimises the power consumption. Efficient scheduling algorithms are also an effective technique to reduce energy consumption in a datacenter. In [19] the author tackles the problem of scheduling real-time tasks. This tasks have dynamic arrival times and their duration is unknown. A novel scheduling algorithm named PRS is presented which consider the trade-off between tasks SLA, system resource utilization and energy consumption. Proactive and reactive methods are used to compensate from this lack of information about the workload and an evaluation with real world traces from Google are presented.

A. Energy Management and Machine Learning

One of the latest techniques that were used in DC energy management is Machine Learning (ML). It has been used extensively to perform an assessment of a new DC configuration, to pinpoint a possible optimization opportunity or to assess the whole energy performance. Some certain tasks that are performed in a DC can be tracked and used as a training kit for a ML algorithm, in order to be able to predict the workloads for similar future tasks. Using this technique, i.e. to create a model from past experience, is more efficient in order to save energy than using an explicit model which was designed by an expert user [12].

The usage of Reinforcement Learning (RL) to improve energy efficiency on a web application server yielded an improvement of 10% on server power, but kept the desired performance levels [84]. The method that was used by Tesauro *et al.* was Hybrid RL, but other RL methods are also available, such as Apprenticeship Learning, Differential Dynamic Programming and also a fitted policy iteration minimizing Bellman residuals.

Prefetching optimization is also used, along with ML, to improve energy efficiency, as stated by Liao *et al.* [54]. This optimisation (a parameter search problem) is done with ML techniques, which proves to be a very efficient way to find the best configuration. During this research, the main target is to find the best prefetching technique, which is the means for energy efficiency, but ML is the statistical tool that helps find the best technique.

Another different line of work is to optimize the energy consumed by machine learning algorithms. This type of workloads can have long running times and they are computationally intensive. The objective is to sacrifice some accuracy of the trained model for less power consumption. In [87] the authors propose to apply a chain of classifiers depending on the difficulty of the input data. A graphical explanation can be seen in Fig. 4. A similar concept is used in [69] for neural networks. The authors explain the concept of a conditional deep learning network (CDN) where a linear classifier is used to activate the next layer of the neural network if the data is too complex to be classified linearly. This technique results in shorter execution times and so in energy savings.

B. Energy Efficiency with CFD optimisation

Data centre heat generation and subsequently the required cooling is one of the major factors of energy consumption. About 15% of the total energy is used for the environmental control operation. The usage of Computational Fluid Dynamics could simulate the data centre heat generating algorithm, showing which areas are difficult to cool and would probably require more energy for this reason, as shown on Figure 6.

Bash and Forman [9] have shown that the investigation of the infrastructure heat generation, along with the application of a CFD simulation on the DC room, can show which parts generate more heat. After performing the simulation, they

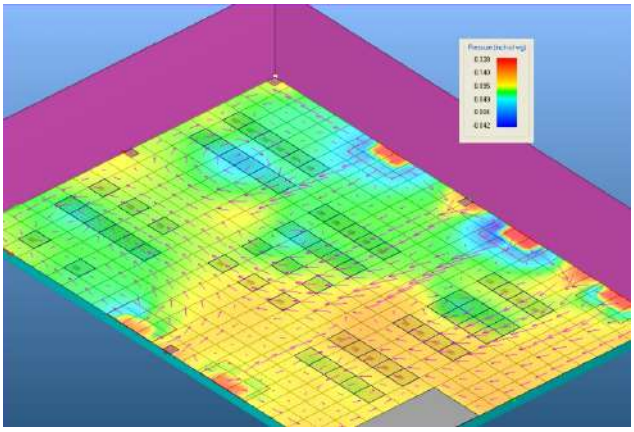

Fig. 5. A CFD analysis of the pattern of the airflow rates in a DC, by analysing the pressure distribution

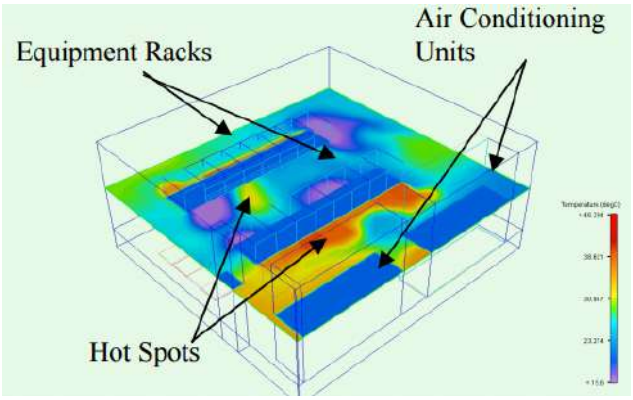

Fig. 6. A sample CFD simulation for a datacenter

applied some changes that were based on the results and they have observed that many hot spots could be avoided by a better design of the room and the placement of the racks. The allocation of resources (VM, HD usage) could prove pivotal in keeping a better temperature allocation across the room. The *temperature-aware workload placement* has been examined before, among others, by Moore *et al.* [64] and [78], by applying an approach that schedules workloads in a manner that minimizes the energy consumed by the cooling infrastructure.

Karki *et al.* [45] simulated the airflow through special perforated floor tiles, in a real life data center with a raised floor. This simulation, along with similar references, has lead to the development of software like TileFlow®, which can be used to simulate airflows in data centers, as seen in Fig. 5.

The initial dimensioning of the air-conditioning (AC) system, by estimating the heat generation and dissipation using CFD, was also examined by Patel *et al.* [70]. Before this technique, the AC systems were designed intuitively and were also oversized, in order to ensure that the DC will never have overheating problems. Using complex CFD techniques for initial dimensioning could improve the sizing of the AC and can avoid the usage of an over sized system, which could eventually consume excess power for no apparent reason.

Regarding a storage server, the investigation of the allocation of data along specific hard disks could prevent over-heated areas. This workload-specific allocation could be achieved by configuring the system, in order for it to allocate data to different areas. CEPH [88] can perform this, by using different placement groups for the data. For example, if a specific corner of the DC seems to be overheated, data could be allocated to hard disks that are in different places in that room, in order to distribute heat generation better.

C. Impact of Process Reliability Mechanisms

Additionally to data availability, it is also desired to ensure process availability. If a process is continuously faulting and re-spawned, data transfer will be on a continuous demand, canceling all the deployed optimization. Proposed techniques involve a combination of checkpoint/restart and process shadowing [62]. More specifically, consistent snapshots of a process are replicated into multiple computational nodes which later can proceed either lock-stepped, independently, or remain frozen until they are needed.

VIII. Acknowledgement

This work is part of the "BigStorage: Storage-based Convergence between HPC and Cloud to handle Big Data" project, funded by the European Union under the Marie Sklodowska-Curie Actions (H2020-MSCA-ITN-2014-642963).

References

[1] M. S. Abdelfattah, A. Hagiescu, and D. Singh, "Gzip on a chip: high performance lossless data compression on fpgas using opencl," in *Proceedings of the International Workshop on OpenCL, IWOCL 2013 & 2014, May 13-14, 2013, Georgia Tech, Atlanta, GA, USA / Bristol, UK, May 12-13, 2014*, 2014, pp. 4:1–4:9. [Online]. Available: http://doi.acm.org/10.1145/2664666.2664670

[2] N. Agrawal, V. Prabhakaran, T. Wobber, J. D. Davis, M. Manasse, and R. Panigrahy, "Design tradeoffs for ssd performance," in *USENIX 2008 Annual Technical Conference*, 2008, pp. 57–70.

[3] D. Alger, *Grow a greener data center*. Pearson Education, 2009.

[4] H. Amur, J. Cipar, V. Gupta, G. R. Ganger, M. A. Kozuch, and K. Schwan, "Robust and flexible power-proportional storage," in *Proceedings of the 1st ACM Symposium on Cloud Computing (SoCC), Indianapolis, Indiana, USA, June 10-11, 2010*, 2010, pp. 217–228.

[5] D. G. Andersen, J. Franklin, M. Kaminsky, A. Phanishayee, L. Tan, and V. Vasudevan, "FAWN: a fast array of wimpy nodes," in *Proceedings of the 22nd ACM Symposium on Operating Systems Principles 2009 (SOSP), Big Sky, Montana, USA, October 11-14, 2009*, 2009, pp. 1–14.

[6] U. Awada, K. Li, and Y. Shen, "Energy consumption in cloud computing data centers," *International Journal of Cloud Computing and services science*, vol. 3, no. 3, p. 145, 2014.

[7] L. A. Barroso and U. Holzle, "The case for energy-proportional computing," *IEEE Computer*, vol. 40, no. 12, pp. 33–37, 2007.

[8] L. A. Barroso and U. Hölzle, "The case for energy-proportional computing," *IEEE Computer*, vol. 40, no. 12, pp. 33–37, 2007. [Online]. Available: http://dx.doi.org/10.1109/MC.2007.443

[9] C. Bash and G. Forman, "Cool job allocation: Measuring the power savings of placing jobs at cooling-efficient locations in the data center," in *Proceedings of the 2007 USENIX Annual Technical Conference, Santa Clara, CA, USA, June 17-22, 2007*, 2007, pp. 363–368. [Online]. Available: http://www.usenix.org/events/usenix07/tech/bash.html

[10] L. Benini, D. Bruni, A. Macii, and E. Macii, "Hardware-assisted data compression for energy minimization in systems with embedded processors," in *2002 Design, Automation and Test in Europe Conference and Exposition (DATE 2002), 4-8 March 2002, Paris, France*, 2002, pp. 449–453. [Online]. Available: http://dx.doi.org/10.1109/DATE.2002.998312

[11] L. Benini and G. D. Micheli, "System-level power optimization: techniques and tools," *ACM Trans. Design Autom. Electr. Syst.*, vol. 5, no. 2, pp. 115–192, 2000. [Online]. Available: http://doi.acm.org/10.1145/335043.335044

[12] J. L. Berral, R. Gavaldà, and J. Torres, "Adaptive scheduling on power-aware managed data-centers using machine learning," in *12th IEEE/ACM International Conference on Grid Computing, GRID 2011, Lyon, France, September 21-23, 2011*, 2011, pp. 66–73. [Online]. Available: http://dx.doi.org/10.1109/Grid.2011.18

[13] A. Borghesi, A. Bartolini, M. Lombardi, M. Milano, and L. Benini, "Predictive modeling for job power consumption in HPC systems," in *High Performance Computing - 31st International Conference, ISC High Performance 2016, Frankfurt, Germany, June 19-23, 2016, Proceedings*, 2016, pp. 181–199. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-41321-1_10

[14] D. Boru, D. Kliazovich, F. Granelli, P. Bouvry, and A. Y. Zomaya, "Energy-efficient data replication in cloud computing datacenters," *Cluster Computing*, vol. 18, no. 1, pp. 385–402, 2015.

[15] R. Buyya, A. Beloglazov, and J. H. Abawajy, "Energy-efficient management of data center resources for cloud computing: A vision, architectural elements, and open challenges," *CoRR*, vol. abs/1006.0308, 2010. [Online]. Available: http://arxiv.org/abs/1006.0308

[16] G. Calandrini, A. G. Vicente, I. B. Muñoz, P. A. Revenga, J. L. Lázaro, and F. J. Toledo-Moreo, "Power measurement methods for energy efficient applications," *Sensors*, vol. 13, no. 6, pp. 7786–7796, 2013. [Online]. Available: http://dx.doi.org/10.3390/s130607786

[17] E. V. Carrera, E. Pinheiro, and R. Bianchini, "Conserving disk energy in network servers," in *Proceedings of the 17th Annual International Conference on Supercomputing, ICS 2003, San Francisco, CA, USA, June 23-26, 2003*, 2003, pp. 86–97. [Online]. Available: http://doi.acm.org/10.1145/782814.782829

[18] D. Chen, G. Goldberg, R. Kahn, R. I. Kat, and K. Z. Meth, "Leveraging disk drive acoustic modes for power management," in *26th IEEE Symposium on Mass Storage Systems and Technologies (MSST), 2012, Lake Tahoe, Nevada, USA, May 3-7, 2010*, 2010, pp. 1–9.

[19] H. Chen, X. Zhu, H. Guo, J. Zhu, X. Qin, and J. Wu, "Towards energy-efficient scheduling for real-time tasks under uncertain cloud computing environment," *Journal of Systems and Software*, vol. 99, pp. 20–35, 2015.

[20] Y. Chen, A. Ganapathi, and R. H. Katz, "To compress or not to compress - compute vs. IO tradeoffs for mapreduce energy efficiency," in *Proceedings of the 1st ACM SIGCOMM Workshop on Green Networking 2010, New Delhi, India, August 30, 2010*, 2010, pp. 23–28. [Online]. Available: http://doi.acm.org/10.1145/1851290.1851296

[21] K. Choi, R. Soma, and M. Pedram, "Fine-grained dynamic voltage and frequency scaling for precise energy and performance tradeoff based on the ratio of off-chip access to on-chip computation times," *IEEE Trans. on CAD of Integrated Circuits and Systems*, vol. 24, no. 1, pp. 18–28, 2005.

[22] D. Colarelli and D. Grunwald, "Massive arrays of idle disks for storage archives," in *Proceedings of the 2002 ACM/IEEE conference on Supercomputing, Baltimore, Maryland, USA, November 16-22, 2002*, 2002, pp. 56:1–56:11.

[23] J. Conejero, O. Rana, P. Burnap, J. Morgan, B. Caminero, and C. Carrión, "Analyzing Hadoop power consumption and impact on application QoS," *Future Generation Computer Systems*, vol. 55, pp. 213–223, 2016. [Online]. Available: http://dx.doi.org/10.1016/j.future.2015.03.009

[24] W. Deng, F. Liu, H. Jin, B. Li, and D. Li, "Harnessing renewable energy in cloud datacenters: opportunities and challenges," *IEEE Network*, vol. 28, no. 1, pp. 48–55, 2014.

[25] M. F. Dolz, M. R. Heidari, M. Kuhn, T. Ludwig, and G. Fabregat, "ARDUPOWER: A low-cost wattmeter to improve energy efficiency of HPC applications," in *Sixth International Green and Sustainable Computing Conference, IGSC 2015, Las Vegas, NV, USA, December 14-16, 2015*, 2015, pp. 1–8. [Online]. Available: http://dx.doi.org/10.1109/IGCC.2015.7393692

[26] M. F. Dolz, J. M. Kunkel, K. Chasapis, and S. Catalán, "An analytical methodology to derive power models based on hardware and software metrics," *Computer Science - R&D*, vol. 31, no. 4, pp. 165–174, 2016. [Online]. Available: http://dx.doi.org/10.1007/s00450-015-0298-8

[27] X. Dong, N. Muralimanohar, N. Jouppi, R. Kaufmann, and Y. Xie, "Leveraging 3d pcram technologies to reduce checkpoint overhead for future exascale systems," in *Proceedings of the Conference on High Performance Computing Networking, Storage and Analysis*, 2009, pp. 57:1–57:12.

[28] H. El-Aawar, "Structural and hardware complexities of microprocessor design according to moore's law," *International Journal of Computer Science & Information Technology*, vol. 6, no. 4, p. 175, 2014.

[29] E. Feller, L. Ramakrishnan, and C. Morin, "Performance and energy efficiency of big data applications in cloud environments: A hadoop case study," *Journal of Parallel and Distributed Computing*, vol. 79, pp. 80–89, 2015.

[30] R. Filgueira, M. P. Atkinson, A. Nuñez, and J. Fernández, "An adaptive, scalable, and portable technique for speeding up mpi-based applications," in *Euro-Par 2012 Parallel Processing - 18th International Conference, Euro-Par 2012, Rhodes Island, Greece, August 27-31, 2012. Proceedings*, 2012, pp. 729–740. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-32820-6_72

[31] R. Filgueira, D. E. Singh, A. Calderón, and J. Carretero, "Compi: Enhancing MPI based applications performance and scalability using run-time compression," in *Recent Advances in Parallel Virtual Machine and Message Passing Interface, 16th European PVM/MPI Users' Group Meeting, Espoo, Finland, September 7-10, 2009. Proceedings*, 2009, pp. 207–218. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-03770-2_27

[32] R. Filgueira, D. E. Singh, J. Carretero, A. Calderón, and F. García, "Adaptive-compi: Enhancing mpi-based applications' performance and scalability by using adaptive compression," *IJHPCA*, vol. 25, no. 1, pp. 93–114, 2011. [Online]. Available: http://dx.doi.org/10.1177/1094342010373486

[33] I. Goiri, K. Le, T. D. Nguyen, J. Guitart, J. Torres, and R. Bianchini, "Greenhadoop: leveraging green energy in data-processing frameworks," in *Proceedings of the Seventh European Conference on Computer Systems (EuroSys), Bern, Switzerland, April 10-13, 2012*, 2012, pp. 57–70.

[34] R. Gonzalez, B. Gordon, and M. A. Horowitz, "Supply and threshold voltage scaling for low power cmos," *IEEE Journal of Solid-State Circuits*, vol. 32, no. 8, pp. 1210–1216, 1997.

[35] M. Grawinkel, L. Nagel, and A. Brinkmann, "Lonestar RAID: massive array of offline disks for archival systems," *ACM Transactions on Storage (TOS)*, vol. 12, no. 1, p. 5, 2016.

[36] M. Grawinkel, T. Schäfer, A. Brinkmann, J. Hagemeyer, and M. Porrmann, "Evaluation of applied intra-disk redundancy schemes to improve single disk reliability," in *19th Annual IEEE/ACM International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS), Singapore, 25-27 July, 2011*, 2011, pp. 297–306.

[37] D. Hackenberg, T. Ilsche, R. Schöne, D. Molka, M. Schmidt, and W. E. Nagel, "Power measurement techniques on standard compute nodes: A quantitative comparison," in *2012 IEEE International Symposium on Performance Analysis of Systems & Software, Austin, TX, USA, 21-23 April, 2013*, 2013, pp. 194–204. [Online]. Available: http://dx.doi.org/10.1109/ISPASS.2013.6557170

[38] D. Hackenberg, T. Ilsche, J. Schuchart, R. Schöne, W. E. Nagel, M. Simon, and Y. Georgiou, "HDEEM: high definition energy efficiency monitoring," in *Proceedings of the 2nd International Workshop on Energy Efficient Supercomputing, E2SC '14, New Orleans, Louisiana, USA, November 16-21, 2014*, 2014, pp. 1–10. [Online]. Available: http://dx.doi.org/10.1109/E2SC.2014.13

[39] D. Hackenberg, R. Schöne, T. Ilsche, D. Molka, J. Schuchart, and R. Geyer, "An energy efficiency feature survey of the intel haswell processor," in *2015 IEEE International Parallel and Distributed Processing Symposium Workshop, IPDPS 2015, Hyderabad, India, May 25-29, 2015*, 2015, pp. 896–904. [Online]. Available: http://dx.doi.org/10.1109/IPDPSW.2015.70

[40] P. Huang, G. Wan, K. Zhou, M. Huang, C. Li, and H. Wang, "Improve effective capacity and lifetime of solid state drives," in *Proceedings of the 2013 IEEE Eighth International Conference on Networking, Architecture and Storage*, 2013, pp. 50–59.

[41] Q. Huang, S. Su, J. Li, P. Xu, K. Shuang, and X. Huang, "Enhanced energy-efficient scheduling for parallel applications in cloud," in *Proceedings of the 2012 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (ccgrid 2012)*. IEEE Computer Society, 2012, pp. 781–786.

[42] X. Huang, Y. Ni, D. Chen, S. Liu, H. Fu, and G. Yang, "Czip: A fast lossless compression algorithm for climate data," *International Journal of Parallel Programming*, vol. 44, no. 6, pp. 1248–1267, 2016. [Online]. Available: http://dx.doi.org/10.1007/s10766-016-0403-z

[43] S. Ibrahim, T. Phan, A. Carpen-Amarie, H. Chihoub, D. Moise, and G. Antoniu, "Governing energy consumption in hadoop through CPU frequency scaling: An analysis," *Future Generation Comp. Syst.*, vol. 54, pp. 219–232, 2016.

[44] T. Ilsche, D. Hackenberg, S. Graul, R. Schöne, and J. Schuchart, "Power measurements for compute nodes: Improving sampling rates, granularity and accuracy," in *Sixth International Green and Sustainable Computing Conference, IGSC 2015, Las Vegas, NV, USA, December 14-16, 2015*, 2015, pp. 1–8. [Online]. Available: http://dx.doi.org/10.1109/IGCC.2015.7393710

[45] K. C. Karki, A. Radmehr, and S. V. Patankar, "Use of computational fluid dynamics for calculating flow rates through perforated tiles in raised-floor data centers," *HVAC&R Research*, vol. 9, no. 2, pp. 153–166, 2003.

[46] H. Kim, S. Seshadri, C. L. Dickey, and L. Chiu, "Evaluating phase change memory for enterprise storage systems: A study of caching and tiering approaches," in *Proceedings of the 12th USENIX Conference on File and Storage Technologies*, ser. FAST'14, 2014, pp. 33–45.

[47] Y. Kim, A. Gupta, B. Urgaonkar, P. Berman, and A. Sivasubramaniam, "Hybridstore: A cost-efficient, high-performance storage system combining ssds and hdds," in *2011 IEEE 19th Annual International Symposium on Modelling, Analysis, and Simulation of Computer and Telecommunication Systems*, 2011, pp. 227–236.

[48] D. Kliazovich, P. Bouvry, and S. U. Khan, "Greencloud: a packet-level simulator of energy-aware cloud computing data centers," *The Journal of Supercomputing*, vol. 62, no. 3, pp. 1263–1283, 2012.

[49] M. H. Kryder and C. S. Kim, "After hard drives ;what comes next?" *IEEE Transactions on Magnetics*, vol. 45, no. 10, pp. 3406–3413, 2009.

[50] J. Kunkel, M. Kuhn, and T. Ludwig, "Exascale Storage Systems – An Analytical Study of Expenses," *Supercomputing Frontiers and Innovations*, pp. 116–134, 06 2014. [Online]. Available: http://superfri.org/superfri/article/view/20

[51] J. M. Kunkel, A. Aguilera, N. Hübbe, M. C. Wiedemann, and M. Zimmer, "Monitoring energy consumption with SIOX," *Computer Science - R&D*, vol. 30, no. 2, pp. 125–133, 2015.

[52] J. M. Kunkel, O. Mordvinova, M. Kuhn, and T. Ludwig, "Collecting energy consumption of scientific data - energy demands for files during their life cycle," *Computer Science - R&D*, vol. 25, no. 3-4, pp. 197–205, 2010.

[53] D. Li, B. R. de Supinski, M. Schulz, K. W. Cameron, and D. S. Nikolopoulos, "Hybrid mpi/openmp power-aware computing," in *24th IEEE International Symposium on Parallel and Distributed Processing, IPDPS 2010, Atlanta, Georgia, USA, 19-23 April 2010 - Conference Proceedings*, 2010, pp. 1–12. [Online]. Available: http://dx.doi.org/10.1109/IPDPS.2010.5470463

[54] S. Liao, T. Hung, D. Nguyen, C. Chou, C. Tu, and H. Zhou, "Machine learning-based prefetch optimization for data center applications," in *Proceedings of the ACM/IEEE Conference on High Performance Computing, SC 2009, November 14-20, 2009, Portland, Oregon, USA*, 2009. [Online]. Available: http://doi.acm.org/10.1145/1654059.1654116

[55] C.-C. Lin, P. Liu, and J.-J. Wu, "Energy-aware virtual machine dynamic provision and scheduling for cloud computing," in *Cloud Computing (CLOUD), 2011 IEEE International Conference on*. IEEE, 2011, pp. 736–737.

[56] P. Lindstrom and M. Isenburg, "Fast and efficient compression of floating-point data," *IEEE Trans. Vis. Comput. Graph.*, vol. 12, no. 5, pp. 1245–1250, 2006. [Online]. Available: http://dx.doi.org/10.1109/TVCG.2006.143

[57] N. Liu, J. Cope, P. H. Carns, C. D. Carothers, R. B. Ross, G. Grider, A. Crume, and C. Maltzahn, "On the role of burst buffers in leadership-class storage systems." in *MSST*. IEEE Computer Society, 2012, pp. 1–11.

[58] J. Ll Berral, R. Gavaldà, and J. Torres, "Empowering automatic data-center management with machine learning," in *Proceedings of the 28th Annual ACM Symposium on Applied Computing*. ACM, 2013, pp. 170–172.

[59] P. Llopis, M. F. Dolz, F. J. G. Blas, F. Isaila, M. R. Heidari, and M. Kuhn, "Analyzing the energy consumption of the storage data path," *The Journal of Supercomputing*, vol. 72, no. 11, pp. 4089–4106, 2016. [Online]. Available: http://dx.doi.org/10.1007/s11227-016-1729-4

[60] Y. Lu, J. Shu, and W. Zheng, "Extending the lifetime of flash-based storage through reducing write amplification from file systems," in *Presented as part of the 11th USENIX Conference on File and Storage Technologies (FAST 13)*, 2013, pp. 257–270.

[61] D. Meister, J. Kaiser, A. Brinkmann, T. Cortes, M. Kuhn, and J. M. Kunkel, "A study on data deduplication in HPC storage systems," in *SC Conference on High Performance Computing Networking, Storage and Analysis (SC), Salt Lake City, UT, USA - November 11 - 15, 2012*, 2012, p. 7.

[62] B. N. Mills, T. Znati, R. G. Melhem, K. B. Ferreira, and R. E. Grant, "Energy consumption of resilience mechanisms in large scale systems," in *22nd Euromicro International Conference on Parallel, Distributed, and Network-Based Processing (PDP), Torino, Italy, February 12-14, 2014*, 2014, pp. 528–535.

[63] S. Mittal, "A survey of architectural techniques for dram power management," *Int. J. High Perform. Syst. Archit.*, vol. 4, no. 2, pp. 110–119, 2012.

[64] J. D. Moore, J. S. Chase, P. Ranganathan, and R. K. Sharma, "Making scheduling "cool": Temperature-aware workload placement in data centers," in *Proceedings of the 2005 USENIX Annual Technical Conference, April 10-15, 2005, Anaheim, CA, USA*, 2005, pp. 61–75. [Online]. Available: http://www.usenix.org/events/usenix05/tech/general/moore.html

[65] D. Narayanan, A. Donnelly, and A. I. T. Rowstron, "Write off-loading: Practical power management for enterprise storage," in *6th USENIX Conference on File and Storage Technologies (FAST), February 26-29, 2008, San Jose, CA, USA*, 2008, pp. 253–267.

[66] D. Narayanan and O. Hodson, "Whole-system persistence," in *Proceedings of the Seventeenth International Conference on Architectural Support for Programming Languages and Operating Systems*, 2012, pp. 401–410.

[67] D. Narayanan, E. Thereska, A. Donnelly, S. Elnikety, and A. Rowstron, "Migrating server storage to ssds: Analysis of tradeoffs," in *Proceedings of the 4th ACM European Conference on Computer Systems*, ser. EuroSys '09, 2009, pp. 145–158.

[68] A. Orgerie, M. D. de Assunção, and L. Lefèvre, "A survey on techniques for improving the energy efficiency of large-scale distributed systems," *ACM Comput. Surv.*, vol. 46, no. 4, pp. 47:1–47:31, 2013.

[69] P. Panda, A. Sengupta, and K. Roy, "Conditional deep learning for energy-efficient and enhanced pattern recognition," in *2016 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 2016, pp. 475–480.

[70] C. D. Patel, C. E. Bash, C. Belady, L. Stahl, and D. Sullivan, "Computational fluid dynamics modeling of high compute density data centers to assure system inlet air specifications," in *Proceedings of IPACK*, vol. 1, 2001, pp. 8–13.

[71] R. A. Patel, Y. Zhang, J. Mak, A. Davidson, and J. D. Owens, "Parallel lossless data compression on the gpu," in *2012 Innovative Parallel Computing (InPar)*, May 2012, pp. 1–9.

[72] E. Pinheiro, R. Bianchini, and C. Dubnicki, "Exploiting redundancy to conserve energy in storage systems," *SIGMETRICS Perform. Eval. Rev.*, vol. 34, no. 1, pp. 15–26, Jun. 2006. [Online]. Available: http://doi.acm.org/10.1145/1140103.1140281

[73] M. K. Qureshi, V. Srinivasan, and J. A. Rivers, "Scalable high performance main memory system using phase-change memory technology," in *Proceedings of the 36th Annual International Symposium on Computer Architecture*, ser. ISCA '09, 2009, pp. 24–33.

[74] L. Ramapantulu, B. M. Tudor, D. Loghin, T. Vu, and Y. M. Teo, "Modeling the energy efficiency of heterogeneous clusters," in *2014 43rd International Conference on Parallel Processing*. IEEE, 2014, pp. 321–330.

[75] P. Ratanaworabhan, J. Ke, and M. Burtscher, "Fast lossless compression of scientific floating-point data," in *2006 Data Compression Conference (DCC 2006), 28-30 March 2006, Snowbird, UT, USA*, 2006, pp. 133–142. [Online]. Available: http://dx.doi.org/10.1109/DCC.2006.35

[76] S. M. Ross, *Introduction to probability models*. Academic press, 2014.

[77] G. Semeraro, G. Magklis, R. Balasubramonian, D. H. Albonesi, S. Dwarkadas, and M. L. Scott, "Energy-efficient processor design using multiple clock domains with dynamic voltage and frequency scaling," in *Proceedings of the Eighth International Symposium on High-*

Performance Computer Architecture (HPCA), Boston, Massachusettes, USA, February 2-6, 2002, 2002, pp. 29–42.

[78] R. K. Sharma, C. Bash, C. D. Patel, R. J. Friedrich, and J. S. Chase, "Balance of power: Dynamic thermal management for internet data centers," *IEEE Internet Computing*, vol. 9, no. 1, pp. 42–49, 2005. [Online]. Available: http://dx.doi.org/10.1109/MIC.2005.10

[79] H. Shoukourian, T. Wilde, A. Auweter, and A. Bode, "Predicting the energy and power consumption of strong and weak scaling hpc applications," *Supercomputing frontiers and innovations*, vol. 1, no. 2, pp. 20–41, 2014.

[80] B. Sotomayor, R. S. Montero, I. M. Llorente, and I. Foster, "Virtual infrastructure management in private and hybrid clouds," *IEEE Internet computing*, vol. 13, no. 5, pp. 14–22, 2009.

[81] M. W. Storer, K. M. Greenan, E. L. Miller, and K. Voruganti, "Pergamum: Replacing tape with energy efficient, reliable, disk-based archival storage," in *6th USENIX Conference on File and Storage Technologies (FAST), February 26-29, 2008, San Jose, CA, USA*, 2008, pp. 1–16.

[82] G. Sun, Y. Joo, Y. Chen, D. Niu, Y. Xie, Y. Chen, and H. Li, "A hybrid solid-state storage architecture for the performance, energy consumption, and lifetime improvement," in *HPCA - 16 2010 The Sixteenth International Symposium on High-Performance Computer Architecture*, 2010, pp. 1–12.

[83] L. Tan, S. L. Song, P. Wu, Z. Chen, R. Ge, and D. J. Kerbyson, "Investigating the interplay between energy efficiency and resilience in high performance computing," in *2015 IEEE International Parallel and Distributed Processing Symposium (IPDPS), Hyderabad, India, May 25-29, 2015*, 2015, pp. 786–796.

[84] G. Tesauro, R. Das, H. Chan, J. O. Kephart, D. Levine, F. L. R. III, and C. Lefurgy, "Managing power consumption and performance of computing systems using reinforcement learning," in *Advances in Neural Information Processing Systems 20, Proceedings of the Twenty-First Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 3-6, 2007*, 2007, pp. 1497–1504.

[85] E. Thereska, A. Donnelly, and D. Narayanan, "Sierra: practical power-proportionality for data center storage," in *Proceedings of the Seventh European Conference on Computer Systems (EuroSys), Salzburg, Austria, April 10-13, 2011*, 2011, pp. 169–182.

[86] A. Venish and K. S. Sankar, "Study of chunking algorithm in data deduplication," in *Proceedings of the International Conference on Soft Computing Systems*. Springer, 2016, pp. 13–20.

[87] S. Venkataramani, A. Raghunathan, J. Liu, and M. Shoaib, "Scalable-effort classifiers for energy-efficient machine learning," *Proceedings of the 52nd Annual Design Automation Conference on - DAC '15*, pp. 1–6, 2015. [Online]. Available: http://doi.acm.org/10.1145/2744769.2744904{\%}5Cnhttp://dl.acm.org/citation.cfm?doid=2744769.2744904

[88] S. A. Weil, S. A. Brandt, E. L. Miller, D. D. E. Long, and C. Maltzahn, "Ceph: A scalable, high-performance distributed file system," in *7th Symposium on Operating Systems Design and Implementation (OSDI '06), November 6-8, Seattle, WA, USA*, 2006, pp. 307–320. [Online]. Available: http://www.usenix.org/events/osdi06/tech/weil.html

[89] B. Welton, D. Kimpe, J. Cope, C. M. Patrick, K. Iskra, and R. B. Ross, "Improving I/O forwarding throughput with data compression," in *2011 IEEE International Conference on Cluster Computing (CLUSTER), Austin, TX, USA, September 26-30, 2011*, 2011, pp. 438–445. [Online]. Available: http://dx.doi.org/10.1109/CLUSTER.2011.80

[90] D. Wong and M. Annavaram, "Knightshift: Scaling the energy proportionality wall through server-level heterogeneity," in *45th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO), Vancouver, BC, Canada, December 1-5, 2012*, 2012, pp. 119–130.

[91] W. Xia, H. Jiang, D. Feng, F. Douglis, P. Shilane, Y. Hua, M. Fu, Y. Zhang, and Y. Zhou, "A comprehensive study of the past, present, and future of data deduplication," *Proceedings of the IEEE*, vol. 104, no. 9, pp. 1681–1710, 2016. [Online]. Available: http://dx.doi.org/10.1109/JPROC.2016.2571298

[92] X. Xu and Q. Tu, "Data deduplication mechanism for cloud storage systems," in *2015 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery, CyberC 2015, Xi'an, China, September 17-19, 2015*, 2015, pp. 286–294. [Online]. Available: http://dx.doi.org/10.1109/CyberC.2015.71