Leveraging energy-efficient non-lossy compression for data-intensive applications

Issam Raïs*, Daniel Balouek-Thomert†, Anne-Cécile Orgerie‡, Laurent Lefèvre§ and Manish Parashar†

*Department of Computer Science, UiT The Arctic University of Norway
†Rutgers Discovery Informatics Institute, Rutgers University, Piscataway NJ, USA
‡Inria, LIP, ENS Lyon, France
§Univ. Rennes, Inria, CNRS, IRISA, Rennes, France

***Abstract*—The continuous increase of data volumes poses several challenges to established infrastructures in terms of resource management and expenses. One of the most important challenges is the energy-efficient enactment of data operations in the context of data-intensive applications. Computing, generating and exchanging growing volumes of data are costly operations, both in terms of time and energy. In the late literature, different types of compression mechanisms emerge as a new way to reduce time spent on data-related operations, but the overall energy cost has not been studied. Based on current advances and benefits of compression techniques, we propose a model that leverages non-lossy compression and identifies situations where compression presents an interest from an energy reduction perspective. The proposed model considers sender, receiver, communications costs over various types of files and available bandwidth. This strategy allows us to improve both time and energy required for communications by taking advantage of idle times and power states. Evaluation is performed over HPC, Big Data and datacenter scenarios. Results show significant energy savings for all types of file while avoiding counter performances, resulting in a strong incentive to actively leverage non-lossy compression using our model.**

***Keywords*-Big Data, Compression, Energy efficiency, HPC, Datacenter**

I. Introduction

Energy consumption is a growing global concern. With the multiplication of connected devices per person around the world, reducing the energy consumption of large scale computing system is a mandatory step to address in order to build a sustainable digital society.

To face this growing concern many solutions have been developed at multiple levels of computing facilities: hardware, middleware, and application. Examples of hardware leverages are Dynamic Voltage and Frequency Scaling (DVFS) [1] or ShutDown Techniques [2]. At the middleware level, energy efficient policies for task, jobs and resource managers can also be labeled as leverages [3]. Last, at the application level, the way an application is implemented has important effects on its energy and power consumption [4].

Nowadays applications tend to generate very large amount of data. Thus, data centric computing has raised since late years in the literature. In a data driven world, the convergence of problematics like HPC and big data are inevitable. Scientists are now struggling to deal with all these generated data on the fly.

One promising technique that currently rises in literature is the lossy compression, that lightly pre-processes data to know if it is relevant to take it into account in the computing process [5]. This leverage is promising but implies a new way of pre-processing data for every application. Thus, this leverage is not generic and has to be evaluated in terms of energy consumption.

In this paper, we tackle the problem from a different perspective. We aim at underlining the relevance of non lossy compression process in various contexts from an energy efficient perspective. By using such a leverage through our proposed solution, we expect a reduction in the overall energy consumption as well as a reduction in response time.

The HPC - BigData context is interesting for the compression leverage. Indeed, workflows made of applications that generate and more importantly exchange a lot of data, creating contention on the network, are good candidates for such a technique. We propose a model that gives binary answers to the usage of compression and decompression to improve energy efficiency of data exchanges.

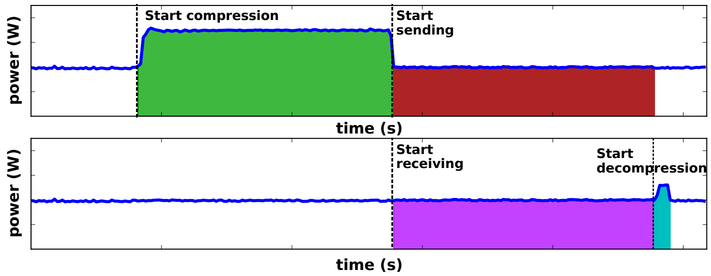
We evaluate our approach on different representative simulated scenarios with extracted traces from HPC and BigData applications running on Titan Supercomputer, on datacenter and Edge facilities.

This paper is structured as follow. Section II will present the compression model as an energy leverage. Section III will focus on the experimental platform while Section IV will analyze evaluation results. Section V presents an overview of selected related works. Section VI presents some conclusion and future works.

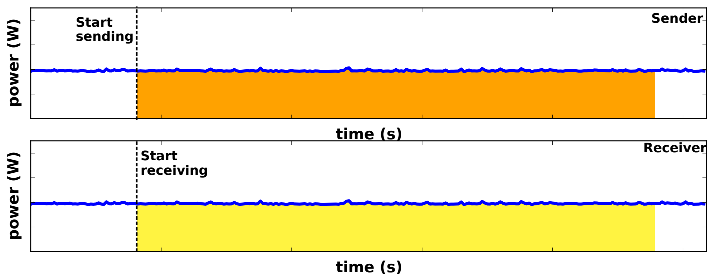
II. Compression model as energy leverage and modelisation

The aim of the proposed compression model is to answer the following question: "Is it energy efficient to compress a file on a specific sender node and decompress it on a receiver node for a given size, file format, and available bandwidth?".

Figure 1 explains the comparison of energy costs addressed by the model. Figure 1a presents the cost of the model with compression and decompression on sender and receiver nodes,



(a) Model exchange cost for sender and receiver



(b) No model (No Compression) exchange cost for sender and receiver

Fig. 1: Model of energy cost with or without compression and decompression

respectively, while Figure 1b presents the cost of exchanging an uncompressed file for sender and receiver, respectively. For our model, on the sender side, sending the file starts only when the compression is done. Thus the reception starts only when the compression is finished, on the receiver side. Decompression starts only if the file is completely received, on the receiver node. The energy consumed by the sender and the receiver during these actions is compared to what would happened during the exchange of the original file without the steps of compression and decompression, as displayed in Figure 1b.

$CompressionEnergyCost$(Green) + $EmissionCompressedFileEnergyCost$(Red) + $ReceptionCompressedFileEnergyCost$(Purple) + $DecompressionEnergyCost$(Blue) $<$ $EmissionFileEnergyCost$(Orange) + $ReceptionFileEnergyCost$(Yellow)

Thus the model is summed up with the previous formula (in parenthesis the corresponding color in Figure 1) where the answer is a binary statement. The left part of the equation corresponds to the model Figure 1a while the right side corresponds to Figure 1b.

A. Modeling energy cost of compression and decompression phases

To understand the implication of compression and decompression process to energy consumption, we used a compression tool on various file formats and for each of these, various file sizes.

Table I presents results of energy consumption of compression and decompression of pbzip tool with various formats and sizes of file on Nova server nodes (from the Grid'5000 experimental testbed described on Table II).

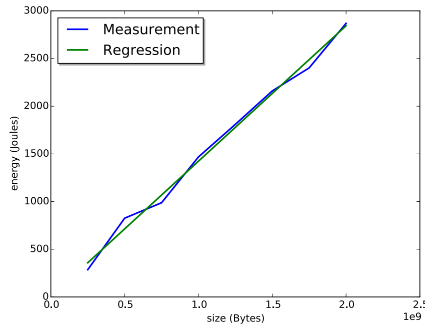
The compression rate is computed by dividing the original size of a file by the size obtained after compression. We noticed that for every experience, on every file structure and size of file generated, the compression rates are very close (the worst case is 1.3185% for random binary files and the best case is 250 % for XML file), showing a high correlation between the file structure and the compression rate.

The proposed energy consumption prediction model is thus based on the size and the structure of the file. This prediction model is a linear regression based on results obtained for various explored sizes. Every file structure has its own regression model for compression and decompression. Thus, for a given size and a file structure, the regression tells us how much energy is consumed to compress or decompress the file. On a production scenario, this regression could be built on the fly and refined every time a file is compressed and decompressed.

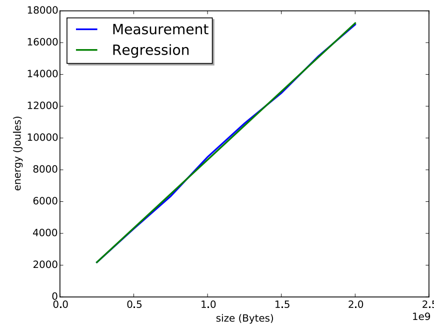
Figure 2 shows results of energy consumption for compression in function of size of various file structures. The obtained linear regression and actual measurements are represented for every case as a green and blue lines, respectively. Every energy consumption obtained is very close to the obtained linear reduction.

B. Modeling energy cost of emission and reception phases

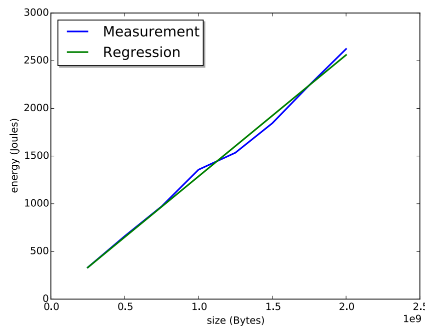
The energy cost during a file exchange between two nodes results from the usage of three major components: the sender,



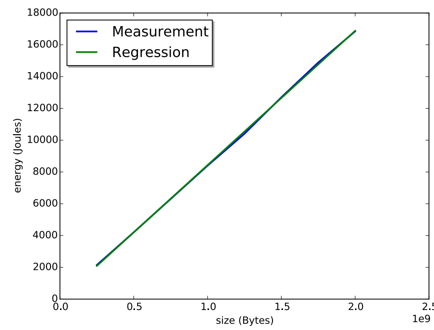
(a) Random binary files



(b) BP files



(c) SQL files



(d) XML files

Fig. 2: Compression energy consumption for various size and linear regression

the network, the receiver.

For simplicity, we consider that the cost of network is, at full or idle load, almost constant [6]. Thus, network will have a constant energy cost here. We place this leverage in the case of contention in the network. Therefore, available bandwidth is also a parameter for our answer. The idle power consumed by the sender and receiver are also parameters of our model. We only take into account point-to-point communications.

For the sender and the receiver, the best scenario is when the nodes send and receive the expected data while consuming the same energy as being in their idle state. Indeed, it is, after its shutdown state, the lowest reachable power state. Thus, from an energy point of view, to determine if it is beneficial to use compression and decompression or send as generated, we compare the best case (idle for the initial size on both sides) with the energy cost exposed by our regression on compression and decompression for the given file structure and size.

Thus, needed information to estimate this best case scenario is the idle power for both sender and receiver, the available bandwidth during the transit of the file, and the size of the raw file to send.

With this information, the model answers yes to the question if the cost of idle during compression sending and decompressing is lower than sending raw generated file over the contended network, from an energy point of view.

III. Experimental setup

In order to evaluate our model, we explore parameters that consider different bandwidths, sizes of data exchanges and energy consumption of nodes. In this section, we describe the infrastructure, parameters and traces that were used.

A. Calibration architecture

Our evaluation is carried out on the Grid'5000 infrastructure. Grid'5000 is a large-scale and versatile testbed for experiment-driven research in all areas of computer science, with a focus on parallel and distributed computing including Cloud, HPC and Big Data [7]. It provides a software stack, including management tools and monitoring frameworks. This includes the Kwapi utility that allows a user to monitor and profile the power consumption of platform nodes (servers and network equipment) [8]. Kwapi gives us access to external wattmeters on the Grid'5000 Lyon site, that monitor entire nodes with a 0.125 Watts accuracy. Characteristics of nodes used to calibrate the energy consumption model, named *Nova*, are given in Table II.

B. Simulation parameters

Bandwidth We define three evaluation scenarios for the bandwidth parameter: *Datacenter*, *SuperComputer* and *Edge* scenario. The *Datacenter* scenario relies on the Gigabit Ethernet as the network technology as actually implemented on Grid'5000. We simulate a randomly equally distributed contention, varying between 1 to 100 in order to factor the

TABLE I: Energy consumption of pbzip under various sizes' files

	Energy cost (Joules)		
Size (Bytes)	Compression	Decompression	Compression Rate
	Rand Binary Files		
250MB	286.0	192.0	1.3185
500MB	826.0	322.0	1.3185
750MB	988.0	631.0	1.3185
1.0GB	1468.0	650.0	1.3185
1.25GB	1810.0	807.0	1.3185
1.50GB	2160.0	945.0	1.3185
1.75GB	2402.0	1291.0	1.3185
2.00GB	2869.0	1324.0	1.3185
	BP files		
250MB	2180.0	160.0	6.5482
500MB	4290.0	226.0	6.5486
750MB	6334.0	368.0	6.5496
1.0GB	8806.0	494.0	6.5497
1.25GB	10919.0	613.0	6.5517
1.50GB	12827.0	628.0	6.5516
1.75GB	15140.0	793.0	6.5511
2.00GB	17136.0	848.0	6.5509
	SQL files		
250MB	332.0	178.0	11.7680
500MB	660.0	178.0	11.7621
750MB	972.0	305.0	11.7592
1.0GB	1357.0	450.0	11.7592
1.25GB	1535.0	468.0	11.7623
1.50GB	1844.0	651.0	11.7619
1.75GB	2242.0	756.0	11.7608
2.00GB	2623.0	895.0	11.7600
	XML files		
250MB	2149.0	545.0	250.9874
500MB	4205.0	138.0	250.9998
750MB	6298.0	155.0	250.9972
1.0GB	8394.0	293.0	251.0136
1.25GB	10404.0	301.0	251.1332
1.50GB	12698.0	349.0	251.1085
1.75GB	14886.0	326.0	251.0894
2.00GB	16846.0	347.0	251.0799

TABLE II: Calibration nodes characteristics

Features	Nova node
Server model	Dell PowerEdge R430
CPU model	Intel Xeon E5-2620
Number of CPU	2
Cores per CPU	8
Memory (GB)	64
Storage (GB)	2 x 300 (HDD)

maximal capability of this technology. The *SuperComputer* scenario is based on the specifications of Cray Titan XP7[1], using a bandwidth of 100 Gb/s between each computing node. We make the hypothesis that the quality of links between the computing nodes and the data nodes are the same as the *Datacenter* scenario. For the *Edge* scenario, a 10 Mbps link is used for edge links, which according to [9] corresponds to a higher connectivity than the average connection speed in USA. Due to the high instability of the connection over Internet, we raise the maximal contention factor to 1000.

Size of exchanges We define three scenarios for the size of exchanges: *Regular*, *S3D* and *BigData* size. *Regular* size is based on the settings used to generate the regression models: size varies from 250MB to 2GB files, chosen randomly for a fixed number of exchanges. The *S3D* size replays the data exchanges of the S3D application on Titan Cray XK7. S3D is a direct numerical simulation of combustion flow that runs in production. Following the obtained traces, every exchange between compute and data nodes is around 0.2 MBytes (exactly 238328 Bytes). The *BigData* size relies on [10]. In this paper, authors describe pictures exchanges of size between 6MB and 20MB, per chunks of 30 to 160. Thus, we used values comprised between 960 MB and 3.2 GB as parameters.

Power and Energy characteristics The energy consumption of the nodes that send and receive the data is split in two parameters. The *Nova* parameter corresponds to the idle value of energy consumption of the chosen node, as described in Table II. The average monitored value is equal to 100 W. The *Little* parameter corresponds to a low-consumption processor present on smartphones or edge devices. We used a value of 4 W, as referred in [11] where authors evaluated the specifications of a Big-Medium-Little architecture composed of 3 different processors to adapt computing needs to demand. A summary of the simulation parameters is given in Table III.

TABLE III: Summary of simulation parameters

Bandwidth	SuperComputer	100 Gb/s
	Datacenter	1 Gb/s
	Edge	10 Mb/s
Size of exchanges	Regular	250MB to 2GB
	BigData	960MB to 3.2GB
	Edge	0.2MB
Power Consumption (Idle state)	Nova	100W
	Little	4 W

C. Simulation Scenarios

The simulation scenarios combine the parameters described in the previous section. Such scenarios are representative of datacenter management, Big data applications and HPC applications. A summary of the simulation scenarios is given in Table IV.

The scenario A describes what could happen during the lifetime of a datacenter, when exchanging files at application level or between nodes in a point-to-point fashion. It represents the exchange between two nodes with homogeneous architectures. The scenarios B and C describe the file exchanges of an application performing large scale medical image processing (as presented in [10]) where huge amounts of data are exchanged, during every iteration of the application. B represents the exchange between two regular nodes belonging to the same supercomputer, while C considers a regular node and a low-consumption node at the edge of the network.

TABLE IV: Simulation scenarios

Scenario	Data exchange	Bandwidth	Sender	Receiver
A	Regular	Datacenter	Nova	Nova
B	Big Data	SuperComputer	Nova	Nova
C	Big Data	Edge	Nova	Little
D	S3D	Edge	Nova	Nova

[1]https://www.top500.org/system/177975

The scenario D describes the file exchanges of the S3D numerical simulation [12] based on a collected trace where small amounts of data are exchanged, during every iteration of the application.

For every file sent, the model takes into account the linear regression to predict the cost of compression and decompression. From this predicted cost, the actual bandwidth and the node power characteristics, the model answers if it is energy efficient to use compression on the sender node and decompression on the receiver for the given format and size of file.

IV. Evaluation

In this section, we evaluate the different scenarios previously described. For each scenario, we measure three different values over 10,000 data exchanges: (i) *No Compression* : the baseline energy consumption value without any use of the model, (ii) *Active model* : using the model as a leverage when it benefits a communication, (iii) *Always ON* : where the compression is always used without the model regulating the choices.

For all file formats and scenarios, activating the model of compression is always beneficial from an energy perspective. For the scenario *A*, and for the most complicated file format to compress, the *Random binary* files, the percentage of gain is around $8,6\%$. Although, for very compressible files, like *SQL* files, we noticed excellent percentage of energy savings (up to 79%). Intermediately structured files also show significant energy savings, 19% and 31% for BP and XML files, respectively.

Regarding the always active leverage on the *A* scenario, one can observe that for every file format, the proposed model outperforms the energy gains of this configuration. For example, almost 19% and 24% of difference concerning energy gain are seen for XML and BP files, respectively. Note that for BP files, the *Always ON* results in a negative percentage of gain compared to the *No Compression* scenario as some files should not have been compressed. In other words, not using compression would have been better than always using it, but using the proposed model permits energy savings.

For all scenarios from SuperComputer to Edge configurations, the model permits to save important amounts of energy. In the worst case, scenarios *B* and for random binary files, 14.48% of energy is gained using our model. In the best cases, scenarios *C* with *XML* files and *D* with *BP* file, 97.95% and 97.30% of energy is gained compared to the same exchanges without model. This evaluation over different file formats and extensive scenarios demonstrates the relevancy of non-lossy compression from an energy consumption perspective. As seen, the best case is not always the same. We underline the fact that, under all scenarios and studied file format, the proposed model enables significant energy savings while ensuring no counter performances.

V. Related work

The impact of lossless compression techniques on energy consumption has been studied in the literature. *Barr and al.* demonstrate that, using several typical compression algorithms, there is an energy consumption increase when compression is applied before wireless transmission [13]. On the contrary, in [14], authors investigate the need of compression to reduce the energy consumed to save battery on hand-held devices. Their results show that on key cases for devices downloading files, the gain of energy consumption when using lossless compression could be interesting. *Welton and al.* allocate idle CPU resources to compress network traffic, thus reducing the amount of data transferred over the network and increasing the effectiveness of network bandwidth [15]. In [16], the authors explore register file data compression for GPUs to improve power efficiency. Compression reduces the width of the register file during read and write operations, which in turn reduces dynamic power.

Satish and al. show that current cluster implementations suffer from high latency data communication with large volumes of transfers across nodes, leading to inefficiency in performance and energy consumption [17]. Authors conclude that these constraints can be overcome by using a combination of efficient low-overhead data compression techniques to reduce transfer volumes along with latency-hiding techniques.

Very recent papers [18]–[21] have studied lossy compression as a key leverage to reduce the data to process. It consists in preprocessing data using detectors to detect if data is interesting to be sent to the computing kernel. This differs from the focus of this paper, as we propose a generic model having no input nor strong domain-specific knowledge from the application, resulting in no insights on what is the relevance of the data exchanged to allow losses.

Despite several investigations on using lossy or non lossy compression as leverage to consume less energy, none of the previous papers presented a simple model that could answer, at a given time, if the compression could be beneficial from a current state of the system (size of file currently exchanged, actual available bandwidth, energy consumption of nodes) and applied it to the complexity of datacenter, supercomputer and edge systems. Our approach underlines a model that helps an automated system to determine, at runtime, if compressing and decompressing the needed file could lead to a reduction of the energy consumption. It provides a straightforward binary answer to this problem and proves to be easily integrable into third-party systems.

VI. Conclusion and Future work

Evaluating the energy efficiency and its impact on application performance becomes increasingly challenging when computation units are a moving target and data exchange are growing in size and diversity. It is of paramount importance to address data centric applications as the convergence of HPC and Big Data domain to design and leverage energy efficient techniques. In this work, we propose a model that determines the relevancy of using the non-lossy compression leverage on a target setting, for energy efficiency purpose.

Evaluation highlights compression performance of various heterogeneous workloads while considering supercomputer

TABLE V: Simulated energy gains of non-lossy compression model over 4 different scenarios. Each value represents an exchange of 10,000 files.

	A		*B*		*C*		*D*	
Scenario	Energy(J)	%gain	Energy(J)	%gain	Energy(J)	%gain	Energy(J)	%gain
Rand Binary Files								
No Compress.	1.12e+08	0.0	2.45e+08	0.0	6.50e+09	0.0	2.47e+07	0.0
Active model	1.02e+08	8.760	2.10e+08	14.48	4.95e+09	23.79	1.91e+07	22.88
Always ON	1.06e+08	4.836	2.10e+08	14.48	4.95e+09	23.79	1.91e+07	22.88
BP files								
No Compress.	1.12e+08	0.0	2.45e+08	0.0	6.49e+09	0.0	3.00e+07	0.0
Active model	9.07e+07	18.976	1.47e+08	39.87	1.10e+09	83.019	8.08e+05	97.30
Always ON	1.17e+08	-4.860	1.47e+08	39.87	1.10e+09	83.018	8.08e+05	97.30
SQL files								
No Compress.	1.12e+08	0.0	2.47e+08	0.0	6.50e+09	0.0	2.47e+07	0.0
Active model	2.31e+07	79.437	3.65e+07	85.22	5.69e+08	91.25	2.28e+06	90.76
Always ON	2.36e+07	79.002	3.65e+07	85.22	5.69e+08	91.25	2.28e+06	90.76
XML files								
No Compress.	1.13e+08	0.0	2.46e+08	0.0	6.49e+09	0.0	2.47e+07	0.0
Active model	7.73e+07	32.068	1.07e+08	56.41	1.33e+08	97.95	2.87e+06	88.40
Always ON	9.88e+07	13.152	1.07e+08	56.41	1.33e+08	97.95	2.87e+06	88.40

nodes, edge devices and datacenter specifications. We show that besides enabling energy savings without counter performance, our model is also useful to explore the optimization opportunity in infrastructures and system design.

Future work includes the extension of this model to stream processing technologies. Compression of machine-generated data from Edge and IoT devices would extend the battery life of sensors. Also, by including the latency criteria into the compression leverage, the model would contribute to the cost-efficient enactment of stream processing topologies under changing data volume.

Acknowledgments

This research is partially supported by the French FSN ELCI project and the Inria-Rutgers SUSTAM (Sustainable Ultra Scale compuTing, dAta and energy Management) associate team, by NSF via grants numbers ACI 1339036, ACI 1441376 and by the Research Council of Norway (RCN) IKTPluss program, project number 270672. Experiments were carried out using the Grid'5000 testbed, supported by a scientific interest group hosted by Inria and including CNRS, RENATER and several Universities as well as other organisations (https://www.grid5000.fr).

References

[1] D. Suleiman, M. Ibrahim, and I. Hamarash, "Dynamic voltage frequency scaling (dvfs) for microprocessors power and energy reduction," in *4th International Conference on Electrical and Electronics Engineering*, 2005.

[2] A. Benoit, L. Lefèvre, A.-C. Orgerie, and I. Raïs, "Reducing the energy consumption of large-scale computing systems through combined shutdown policies with multiple constraints," *The International Journal of High Performance Computing Applications*, vol. 32, no. 1, pp. 176–188, 2018.

[3] Y. Georgiou, D. Glesser, K. Rzadca, and D. Trystram, "A scheduler-level incentive mechanism for energy efficiency in hpc," in *Cluster, Cloud and Grid Computing (CCGrid), 2015 15th IEEE/ACM International Symposium on*. IEEE, 2015, pp. 617–626.

[4] I. Raïs, L. Lefèvre, A.-C. Orgerie, and A. Benoit, "Exploiting the table of energy and power leverages," in *Algorithms and Architectures for Parallel Processing*, J. Vaidya and J. Li, Eds. Cham: Springer International Publishing, 2018, pp. 175–185.

[5] J. Calhoun, F. Cappello, L. N. Olson, M. Snir, and W. D. Gropp, "Exploring the feasibility of lossy compression for pde simulations," *The International Journal of High Performance Computing Applications*, vol. 33, no. 2, pp. 397–410, 2019.

[6] T. T. Ye, L. Benini, and G. De Micheli, "Analysis of power consumption on switch fabrics in network routers," in *Design Automation Conference, 2002. Proceedings. 39th*. IEEE, 2002, pp. 524–529.

[7] D. Balouek, A. Carpen Amarie, G. Charrier, F. Desprez, E. Jeannot, E. Jeanvoine, A. Lèbre, D. Margery, N. Niclausse, L. Nussbaum, O. Richard, C. Pérez, F. Quesnel, C. Rohr, and L. Sarzyniec, "Adding virtualization capabilities to the Grid'5000 testbed," in *Cloud Computing and Services Science*, ser. Communications in Computer and Information Science, I. Ivanov, M. Sinderen, F. Leymann, and T. Shan, Eds. Springer International Publishing, 2013, vol. 367, pp. 3–20.

[8] F. Clouet, S. Delamare, J.-P. Gelas, L. Lefèvre, L. Nussbaum, C. Parisot, L. Pouilloux, and F. Rossigneux, "A Unified Monitoring Framework for Energy Consumption and Network Traffic," in *TRIDENTCOM - International Conference on Testbeds and Research Infrastructures for the Development of Networks & Communities*, Vancouver, Canada, Jun. 2015, p. 10. [Online]. Available: https://hal.inria.fr/hal-01167915

[9] Akamai, "State of the internet report," https://www.akamai.com/us/en/about/our-thinking/state-of-the-internet-report/, 2016.

[10] S. Bao, Y. Huo, P. Parvathaneni, A. J. Plassard, C. Bermudez, Y. Yao, I. Llyu, A. Gokhale, and B. A. Landman, "A data colocation grid framework for big data medical image processing-backend design," *arXiv preprint arXiv:1712.08634*, 2017.

[11] I. Raïs, A.-C. Orgerie, M. Quinson, and L. Lefèvre, "Quantifying the Impact of Shutdown Techniques for Energy-Efficient Data Centers," *Concurrency and Computation: Practice and Experience*, 2018. [Online]. Available: https://hal.archives-ouvertes.fr/hal-01711812

[12] J. H. Chen, A. Choudhary, B. De Supinski, M. DeVries, E. R. Hawkes, S. Klasky, W.-K. Liao, K.-L. Ma, J. Mellor-Crummey, N. Podhorszki *et al.*, "Terascale direct numerical simulations of turbulent combustion using s3d," *Computational Science & Discovery*, vol. 2, no. 1, p. 015001, 2009.

[13] K. C. Barr and K. Asanović, "Energy-aware lossless data compression," *ACM Trans. Comput. Syst.*, vol. 24, no. 3, pp. 250–291, Aug. 2006. [Online]. Available: http://doi.acm.org/10.1145/1151690.1151692

[14] R. Xu, Z. Li, C. Wang, and P. Ni, "Impact of data compression on energy consumption of wireless-networked handheld devices," in *Distributed Computing Systems, 2003. Proceedings. 23rd International Conference on*. IEEE, 2003, pp. 302–311.

[15] B. Welton, D. Kimpe, J. Cope, C. M. Patrick, K. Iskra, and R. Ross, "Improving i/o forwarding throughput with data compression," in *Cluster Computing (CLUSTER), 2011 IEEE International Conference on*. IEEE, 2011, pp. 438–445.

[16] S. Lee, K. Kim, G. Koo, H. Jeon, M. Annavaram, and W. W. Ro, "Improving energy efficiency of gpus through data compression and compressed execution," *IEEE Transactions on Computers*, vol. 66, no. 5, pp. 834–847, 2017.

[17] N. Satish, C. Kim, J. Chhugani, and P. Dubey, “Large-scale energy-efficient graph traversal: a path to efficient data-intensive supercomputing,” in *High Performance Computing, Networking, Storage and Analysis (SC), 2012 International Conference for*. IEEE, 2012, pp. 1–11.

[18] S. Di and F. Cappello, “Optimization of error-bounded lossy compression for hard-to-compress hpc data,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 29, no. 1, pp. 129–143, 2018.

[19] D. Tao, S. Di, Z. Chen, and F. Cappello, “Exploration of pattern-matching techniques for lossy compression on cosmology simulation data sets,” in *International Conference on High Performance Computing*. Springer, 2017, pp. 43–54.

[20] ——, “Significantly improving lossy compression for scientific data sets based on multidimensional prediction and error-controlled quantization,” in *Parallel and Distributed Processing Symposium (IPDPS), 2017 IEEE International*. IEEE, 2017, pp. 1129–1139.

[21] I. Foster, M. Ainsworth, B. Allen, J. Bessac, F. Cappello, J. Y. Choi, E. Constantinescu, P. E. Davis, S. Di, W. Di *et al.*, “Computing just what you need: online data analysis and reduction at extreme scales,” in *European Conference on Parallel Processing*. Springer, 2017, pp. 3–19.