A Conceptual Framework for HPC Operational Data Analytics

Alessio Netti*, Woong Shin†, Michael Ott*, Torsten Wilde‡ , Natalie Bates§

*Leibniz Supercomputing Centre, {alessio.netti,michael.ott}@lrz.de

†Oak Ridge National Laboratory, shinw@ornl.gov

‡Hewlett Packard Enterprise, wilde@hpe.com

§Energy Efficient HPC Working Group, natalie.jean.bates@gmail.com

***Abstract*—This paper provides a broad framework for understanding trends in Operational Data Analytics (ODA) for High-Performance Computing (HPC) facilities. The goal of ODA is to allow for the continuous monitoring, archiving, and analysis of near real-time performance data, providing immediately actionable information for multiple operational uses. In this work, we combine two models to provide a comprehensive HPC ODA framework: one is an evolutionary model of analytics capabilities that consists of four types, which are descriptive, diagnostic, predictive and prescriptive, while the other is a four-pillar model for energy-efficient HPC operations that covers facility, system hardware, system software, and applications. This new framework is then overlaid with a description of current development and production deployments of ODA within leading-edge HPC facilities. Finally, we perform a comprehensive survey of ODA works and classify them according to our framework, in order to demonstrate its effectiveness.**

***Index Terms*—Exascale, Top500, HPC operations, Energy efficiency, Operational data**

I. Introduction

At the verge of *exascale*, the complexity of modern *High-Performance Computing* (HPC) systems has grown to extreme levels, introducing significant operational challenges. These include additional layers of complexity in HPC system design, due to their increase in power and energy footprint. Furthermore, the dynamic large-scale nature of modern HPC systems has physical consequences in terms of power and cooling, leading towards HPC data center designs that have tight coupling between HPC systems and their cooling infrastructure, which in turn introduces a large set of interdependent control knobs in the face of operators. Due to their sheer scale and complexity, merely understanding the current state of HPC systems before even mentioning optimal decisions has become a significant operational challenge.

In addressing the issues above, HPC sites build and implement *Operational Data Analytics* (ODA) systems to acquire up-to-date knowledge that can lead towards better diagnoses and decisions in day-to-day operations [1]. Due to the sheer number and diversity of data streams that can be produced, ODA systems embrace large-scale data and leverage various analysis techniques to achieve this goal. In addition, ODA systems ultimately aim to provide systematic support for deriving optimal parameters that can improve *Key Performance Indicators* (KPIs). Yet, there are many difficulties in understanding, planning, designing, and implementing ODA systems, due to the degree of sophistication required to make sense out of large amounts of data. Coping with these challenges requires selecting from and experimenting with various techniques found in statistics, data mining, data science, machine learning and computer science. In many cases, HPC practitioners face difficulty in navigating through the abundance of techniques in developing use-cases and applications in their operations.

To address this challenge, we propose a conceptual framework that can provide both a holistic integrative picture and an actionable subdivision of ODA to help the HPC community. This is achieved by employing a staged model of data analytics capabilities [2] that consists of four types: descriptive, diagnostic, predictive, and prescriptive. These capabilities are framed within the ”4-Pillar Framework for Energy-Efficient HPC Data Centers” [3], which consists of building infrastructure, system hardware, system software and applications. Combining these two models creates a two-dimensional four-by-four spatial grid that enables users to perform a mapping of various data analytics techniques, HPC ODA systems and tools in the context of their scope (four pillars) and type (four types of analytics). While the scope indicates the comprehensiveness of an ODA capability, the type of analytics indicates a degree of sophistication that helps establish staged roadmaps in planning for HPC ODA systems.

To demonstrate the effectiveness of the framework, we have performed a comprehensive survey on HPC ODA systems and related use cases from both the industry and academia. Use cases in each work have been carefully decomposed to pieces and have been mapped on the proposed spatial grid. With this process, we were able to compare the use cases in terms of various aspects such as similarity and comprehensiveness based on their relative locations within the grid. Moreover, the framework was able to show areas that are rich, as well as gaps in the ODA landscape that need to be explored, eventually leading towards a better understanding of the trends in HPC

This manuscript has been authored by UT-Battelle, LLC, under contract DE-AC05-00OR22725 with the US Department of Energy (DOE). The US government retains and the publisher, by accepting the article for publication, acknowledges that the US government retains a nonexclusive, paid-up, irrevocable, worldwide license to publish or reproduce the published form of this manuscript, or allow others to do so, for US government purposes. DOE will provide public access to these results of federally sponsored research in accordance with the DOE Public Access Plan (http://energy.gov/downloads/doe-public-access-plan).

2168-9253/21/$31.00 ©2021 IEEE

DOI 10.1109/Cluster48925.2021.00086

ODA and helping the community identify the ”next steps” in their own contexts.

II. Background

The concept of ODA encompasses a wide variety of analysis techniques: these share the common goal of continuous monitoring, archiving, and analysis of near real-time performance monitoring data, providing immediately actionable information for multiple operational uses. It can be used for optimizing system operations and improving KPIs like the *Power Usage Effectiveness* [4] (PUE) in a streaming, online fashion [1]. Depending on the use case, the output of ODA models can either be visualized by system administrators and users to aid them in daily operations [5]–[10] or it can be translated into new settings for system knobs, thus enacting control over the monitored system [11], [12] - however, a survey on HPC ODA [13] revealed that most HPC centers use ODA in visualization-oriented scenarios, with control use cases being often out of reach due to their complexity. Most HPC ODA techniques rely on traditional data mining and statistics [14], [15], but use of machine learning is increasingly gaining traction [16], [17].

HPC ODA can be used at different operational levels according to a site's specific needs. At the data center level, ODA can help optimize the operation of infrastructure and facility-wide systems such as for cooling, communication and power distribution, as well as diagnose issues [12], [18], [19]. At the HPC system level, on the other hand, ODA can be used to improve resource utilization, energy efficiency and quality of service, for example by employing complex schedulers for the placement of user jobs, usually leveraging additional system information (e.g., energy budgets, thermal limits or I/O features) to operate under specific constraints [20]–[23]. Further, at the compute node level, ODA can sensibly improve long-term energy efficiency and reliability - this can be achieved by using runtime systems capable of tuning system knobs (e.g., CPU frequency) dynamically, according to hardware and application behavior [11], [24], [25], or by employing models to diagnose anomalous behavior and in turn prevent catastrophic failures [16], [26], [27]. Lastly, a number of frameworks supports the optimization of applications themselves via auto-tuning of relevant parameters [28], [29].

HPC system and compute node-oriented ODA techniques can also be augmented with advanced forms of prediction: for example, predicting the duration of user jobs and the respective submission patterns via heuristic or learning techniques can be beneficial to scheduling [30]–[32]. Similarly, predicting the computational profile of jobs (e.g., in terms of energy consumption or network usage) and correlating this to historical data can improve the effectiveness of scheduling, runtime and fault detection systems alike [33]–[36].

III. A Conceptual Framework for ODA

Designing and building HPC ODA solutions to determine the most computationally and energy-efficient operation of data centers and HPC systems is the new frontier for HPC sustainability improvements. By providing a framework to understand trends, help with planning and act as a frame of reference for communicating HPC ODA needs to stakeholders, we hope to enrich this emerging area. ODA is a challenging area in research and development for HPC optimization, since optimizing requires considering all aspects of a data center's operation. Therefore, developing the most optimal solutions requires the break down of traditional siloed (i.e., isolated) components and systems.

We are building our framework on top of two established frameworks from the HPC operations and data analytics domains, respectively: the ”4-Pillar Framework for Energy-Efficient HPC Data Centers” [3] and the ”Four Types of Data Analytics” [2]. The former provides a classification of the silos introduced above, namely in the form of *building infrastructure*, *system hardware*, *system software* and *applications*. The latter, on the other hand, describes the typology of the underlying analytics techniques. The four types are *descriptive*, *diagnostic*, *predictive* and *prescriptive* analytics. We combine the four categories each individual framework uses into a bi-dimensional framework (i.e., a 4x4 matrix), with the *pillars* of the HPC model as columns and the *types* of the data analytics model as rows. Table I depicts the proposed ODA framework, alongside a series of common use cases.

A. Four Pillars of Energy-Efficient HPC

The ”4-Pillar Framework for Energy-Efficient HPC Data Centers” (Figure 1) provides a fundamental structure that allows for a common view of how the community can look at the domains of a data center [3]. The defined pillars are:

- **Building Infrastructure**: every support infrastructure (such as cooling and power distribution) needed to run the HPC systems and supporting the data center's operation as a whole.
- **System Hardware**: the hardware components that constitute an HPC system, such as motherboards and firmwares, CPUs, GPUs, memory and system-internal cooling, as well as network equipment.
- **System Software**: the system-level software stack, including the system management software, the resource management and scheduler, the compute nodes' operating system, as well as all tools and libraries that can be used by the users and their applications.
- **Applications**: individual workloads as well as the workload mix executed on a system. An application can be considered a unit of work, since the goal of an HPC system is to find new scientific insight using software applications.

The original motivation of the 4-Pillar Framework was to help with the understanding of the different facets of a data center related to energy efficiency, and to raise the awareness that a siloed approach is no longer sufficient to achieve it. That said, the framework highlights the fundamental data center pillars that will be touched by any data center-wide solution. By using the defined four pillars as the columns for the ODA framework, the different major areas touched by any ODA

TABLE I
A SERIES OF ODA EXAMPLES CATEGORIZED USING OUR FRAMEWORK.

	Building Infrastructure	**System Hardware**	**System Software**	**Applications**
Prescriptive	• Switching between types of cooling [12] • Tuning of cooling machinery [18], [37] • Responding to anomalies [38], [39]	• Cooling optimization at system level [12] • CPU frequency tuning [11], [24], [40] • Tuning of hardware knobs [20], [25], [41]	• Intelligent placement of tasks and threads [42] • Plan-based scheduling [43] • Power and KPI-aware scheduling [21]–[23]	• Auto-tuning of HPC applications [28], [29], [41] • Code improvement recommendations [44]
Predictive	• Predicting data center KPIs [45] • Predicting cooling demand [37] • Modelling cooling performance [18], [46]	• Forecasting hardware sensors [32], [47] • Component failure prediction [48] • Predicting CPU instruction mixes [11]	• Simulating HPC systems and schedulers [49]–[51] • Predicting HPC workloads [23]	• Predicting job durations [30], [34], [35] • Predicting job resource usage [31], [52], [53] • Predicting performance profiles of code regions [24]
Diagnostic	• Fingerprinting data center crises [38] • Infrastructure anomaly detection [54] • Infrastructure stress testing [39]	• Node-level anomaly detection [17], [26], [47] • System-level root cause analysis [9] • Diagnosing network contention issues [19], [55]	• Diagnosing data locality issues [9] • Detection of software anomalies [16], [56] • Identifying sources of OS noise [57]	• Application fingerprinting [33], [36] • Identifying performance patterns [20], [31], [44] • Diagnosing code-level issues [15], [27]
Descriptive	• PUE calculation [4] • Facility data processing [8], [58] • Facility-level dashboards [1], [7]	• ITUE calculation [59] • System performance indicators [14] • System-level dashboards [7], [8]	• Slowdown calculation [60] • Scheduler-level dashboards [61], [62]	• Job performance models [63] • Job data processing [8] • Job-level dashboards [5], [6], [10]

solution are captured. The 4-Pillar Framework has been cited and used by the green IT HPC community, for example, to find underdeveloped areas [64], to provide an overview of the domain [65], [66], to help with monitoring requirements [67], as a basis for further refinement [68], and was even used for proposing green information technology systems for the mining industry [69].

B. Four Types of Data Analytics

For the rows of the ODA framework, the “Four Types of Data Analytics” - *descriptive*, *diagnostic*, *predictive* and *prescriptive* - are used. They are usually implemented in stages, have inherently different purposes and no type of analytics is said to be better than the other. They are interrelated and each of them offers a different insight, mapping to the operational questions that are posed by users and administrators [2], [70], [71]. The staged model is widely used by business consulting companies, as it is a simple and powerful tool for explaining analytics to a broad audience. The defined types are:

- **Descriptive**: the first degree of examination of data that answers the question “what happened?”. In an ODA system, such descriptive analytics manifests in forms of visualizations and dashboards using plots, charts and heatmaps. Example instances of descriptive analytics can be found in dashboards for system administrators and operators that may even include features for automated alerts upon exceeding human-defined thresholds of monitored sensors. This type of analytics can include steps such as normalization, aggregation, outlier removal and dimensionality reduction, but no complex knowledge extraction process (e.g., classification).
- **Diagnostic**: answering the questions “why did something happen?” or ”what problem is this a symptom of?”, in the face of a phenomenon observed from the underlying system, usually via descriptive analytics. This type includes any form of analytics that is able to draw insight which is not obvious to system administrators: this usually comes in the form of machine learning, heuristics or data mining models that ingest multi-dimensional monitoring or log data regarding the current system state, extracting in turn high-level knowledge from low-level data. This kind of processes can be carried out by administrators or users manually relying on their prior experience, but diagnostic analytics aims to provide systematic automation of such diagnoses regardless of who the user is. An ODA system that provides diagnostic analytics capabilities can systematically pinpoint and identify why (or where) a phenomenon happened.
- **Predictive**: predicting or forecasting a system’s state in the near future. Compared to descriptive and diagnostic analytics, which aim for better understanding of the past (*hindsight*), predictive analytics aims for future insights

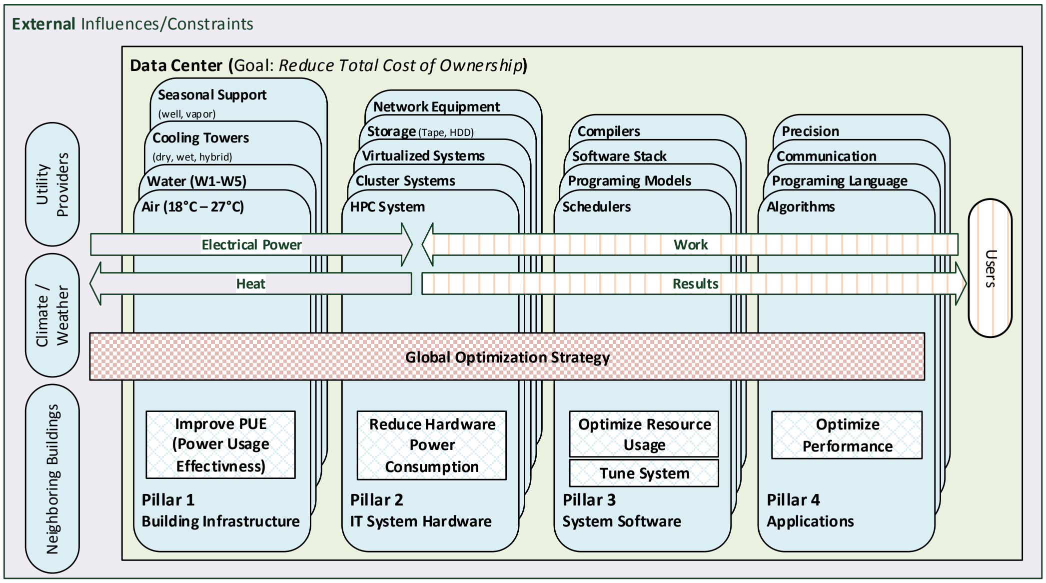

Fig. 1. Four pillars of energy efficient HPC [3]

(*foresight*). This includes, once again, machine learning models or other heuristic techniques that are able to perform forecasting for a certain metric based on current data. In a typical process, current and prior data is used to create a model that can extrapolate or interpolate to an unknown data point given the current state of the system. Unlike in diagnostic and descriptive analytics, this enables anticipating a system's behavior (*proactive* ODA) rather than reacting to it (*reactive* ODA).

- **Prescriptive**: aiming to suggest the best course of actions towards a particular efficiency goal, thus answering the question "what is the best way to manage my resources?". Such analytics involves considering the system's state and converting it into inputs for certain system knobs, which in turn affect the system state itself either in an automated way, or by human inspection. This usually comes in the form of an optimization model that tries to maximize one or more indicators (e.g., energy efficiency or performance) related to the system being considered, and it can once again rely on arbitrarily complex techniques (e.g., machine learning). In some cases, even a linear mapping function may be sufficient.

IV. Applying the Framework

In this section we demonstrate how our framework can be used to describe common ODA use cases effectively: after performing a survey of relevant literature works, we categorize them according to the HPC pillars and data analytics types they belong to. The result of this process is summarized in Table I, which lists several ODA examples for each of the 16 possible classes admitted in our framework. Note that many ODA systems implement a variety of components toward a specific goal, and therefore may cover multiple framework categories at the same time.

A. Building Infrastructure

ODA in this pillar targets the supervision and optimization of building infrastructure systems, including cooling and power distribution machinery among others. Here, *descriptive* ODA entails the calculation of energy efficiency indicators like the PUE [4], as well as basic processing of facility-level monitoring data [8], [58]. Another common use case is the use of graphical dashboards to visualize monitoring data and thus allow system operators to spot issues and inefficiencies [1], [7]. *Diagnostic* ODA includes techniques to detect classes of anomalies in infrastructure components, such as water pumps and power supplies [54] - sometimes, this process is aided by periodical stress testing to improve detection accuracy [39]. Other techniques target the data center as a whole and aim to identify crises that may span multiple sites at a time [38].

The *predictive* type focuses on forecasting a variety of KPIs for data center-level energy efficiency [45], as well as energy and cooling demand in general [37]. Many other works propose theoretical models for cooling systems, which can be useful to forecast the impact of hardware and configuration changes [18], [46]. Finally, the *prescriptive* type supplies models to drive the tuning of infrastructure knobs with the aim of optimizing energy efficiency and reliability. This includes models to switch between multiple cooling technologies according to current demand [12], as well as to determine new optimal settings for knobs such as the inlet water temperature [18], [37]. Response systems to data center-level anomalies (automated or recommendation-based) are also common [38], [39].

B. System Hardware

Here, ODA revolves around the management of HPC hardware, ranging from network components to compute nodes and CPUs. *Descriptive* ODA computes indicators such as the *IT-Power Usage Effectiveness* [59] (ITUE), for system-level energy efficiency, or the *System Information Entropy* [14] (SIE), to characterize system state transitions. Dashboards are once again a common tool to visualize monitoring data and simplify operations [7], [8]. A common ODA application of the *diagnostic* type is node-level detection of hardware anomalies [17], [26], [47], which is an effective tool for improving system reliability. Root cause analysis extends anomaly detection, diagnosing generic behaviors that are not necessarily fault-related [9]. Other approaches pinpoint issues at the HPC system level, such as network contention between concurrent jobs [19], [55].

Predictive system hardware ODA deals with issues such as catastrophic failure prediction in components [48]. More common, however, is the forecasting of sensors (e.g., compute node energy or temperature) [32], [47] or of CPU instruction mixes [11], which is the foundation of many *prescriptive* approaches. Here, in fact, we find a variety of ODA techniques for tuning CPU frequencies [11], [24], [40] or other hardware knobs, such as fan speeds [20], [25], [41], with the aim of maintaining a certain trade-off between efficiency and performance. While most of these techniques can function in a reactive way using real-time data, predictions allow them to have a proactive effect, improving their effectiveness.

C. System Software

ODA techniques in this pillar are centered around the optimization of HPC software components, with a particular focus on scheduling systems. Here, *descriptive* ODA includes computation of metrics such as the job *slowdown* [60] to estimate the quality of service delivered to HPC users by the scheduler. Further, a variety of tools supplies scheduler-oriented data visualization [61], [62] to gain insight over allocation decisions and the resulting system states. On the other hand, the *diagnostic* type is mostly software-oriented, with ODA approaches for detecting software anomalies such as CPU contention or memory leaks [16], as well as the source of floating point errors [56] or of OS and kernel-level noise [57]. Anomaly detection is once again a typical topic, in this case focusing on software-level issues such as data locality and migration in distributed storage systems [9].

Predictive system software ODA includes a variety of scheduling-oriented approaches for simulating HPC systems [49]–[51], enabling the identification of optimal scheduling policies in function of a site's application workload. Other approaches forecast the overall workload of an HPC system in terms of user jobs [23]. All of this - including, if necessary, the output of techniques in other pillars and ODA types - can be used to improve the effectiveness of *prescriptive* techniques. These include intelligent scheduling policies to improve system-level energy efficiency and other KPIs [21]–[23] as well as plan-based scheduling [43]. Other works focus on specific issues, such as the placement of tasks and threads under certain constraints [42].

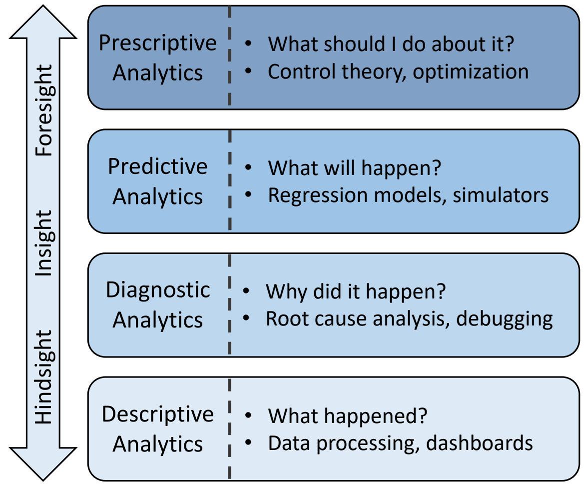

Fig. 2. The four types of data analytics [70].

D. Applications

This pillar is the only one where ODA is partly outside of the system administrators' control and in the users' hands. In terms of *descriptive* ODA, we find performance models such as the *roofline* one, to highlight I/O bottlenecks in applications [63]. Dashboards for application-related data are also common: these allow to visualize performance indicators on a per-job basis, and may include both sensor monitoring data and profiling data resulting from instrumentation [5], [6], [10]. On a similar note, tools enabling processing of job-related data are also available [8]. The *diagnostic* type enables the identification of performance patterns in user codes (e.g., compute or memory boundedness) [20], [31], [44] as well as fingerprinting of entire applications to detect, for example, cryptocurrency miners [33], [36]. Tools able to diagnose code-level issues (e.g., inefficient loops), on the other hand, are useful to both users and developers [15], [27].

Predictive application-level ODA is often at the base of system hardware and software-level prescriptive ODA. Here we find approaches for predicting the duration of user jobs [30], [34], [35], as well as their resource usage [31], [52], [53] based on submission data. Some techniques focus on predicting the duration and performance profile of specific code regions at a high granularity [24]. Finally, as *prescriptive* ODA we find recommendation systems for the improvement of HPC applications from a code perspective [44] and auto-tuning frameworks to optimize application-specific settings under certain performance objectives [28], [29], [41].

V. Discussion

In this section we discuss the insights our framework provides when applied to arbitrary ODA systems, specifically regarding the complexity of ODA spanning across multiple

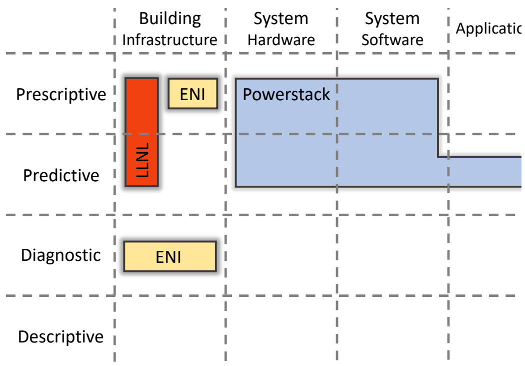

Fig. 3. Examples of complex ODA systems categorized with our framework.

pillars and types, as well as its limitations in certain corner cases. Our discussion is supported by practical examples of ODA systems framed within our model, shown in Figure 3.

A. *Analytics across Multiple Types*

Many opportunities and challenges are connected with implementing ODA across multiple analytics types. First, this allows for more powerful and effective systems: enhancing a prescriptive ODA system with predictive capabilities (see Section IV) allows it to optimize system knobs in a proactive manner, thus anticipating state transitions and preventing adverse effects, rather than in a reactive way. In almost all cases, this has a positive effect on the KPIs that are targeted by ODA systems. The same applies when leveraging the knowledge extracted by diagnostic techniques to make more informed control decisions. Unfortunately, this comes at a significant price: combining the four ODA types not only entails additional complexity in terms of implementation and maintenance but, most importantly, it requires the fusion of heterogeneous disciplines. For example, while implementing prescriptive ODA requires knowledge about system operations and control theory, the diagnostic and predictive types often require expertise in fields such as machine learning. This translates into added complexity from a recruitment and coordination standpoint.

As a practical example, we consider the ODA system proposed by Bortot et al. to detect and respond to infrastructure-level anomalies at the ENI data center located in Pavia, Italy [39]. It includes a diagnostic component which identifies anomalies in hardware components, aided by periodical stress testing, plus a prescriptive one which determines optimal set-point temperatures for the cooling infrastructure, as well as the system settings to reach them in a cost-effective way. While both components operate within the building infrastructure pillar, implementing the diagnostic one requires use of sensor data science techniques; the prescriptive one, on the other hand entails a robust understanding of how infrastructure machinery functions, including access to the related control interfaces.

B. *Single-pillar vs. Multi-pillar Use Cases*

Another dimension of challenge and opportunity comes from the fact that ODA systems can be designed to cover use cases incorporating data or control parameters spanning multiple pillars. Due to the complexity of crossing the pillars that typically represent the traditional operational boundaries within an HPC data center, most use cases are single-pillar ones: single-pillar use cases, in fact, are easier to manage as they can be implemented as *closed* systems, with relatively little concern for other system components and frameworks. On the other hand, multi-pillar use cases require more careful planning and holistic design, often integrating multiple systems with one another and requiring orchestration mechanisms. We observe that the use cases we present in this work are influenced by such trade-off, with a prevalence of single-pillar systems rather than multi-pillar ones, due to the necessity of holistic monitoring and control. However, with the increased popularity in data center designs that have tight coupling between HPC systems and cooling infrastructure, there are opportunities that can come from multi-pillar ODA.

We have found several efforts for incorporating multi-pillar use cases in data center operations. *Powerstack* [41], for example, is a multi-year effort dedicated to identifying the interactions and interfaces to implement a cross-pillar system for HPC power management (*prescriptive*) leveraging intelligent techniques to make more informed scheduling, hardware and software decisions (*predictive*). While multi-pillar use cases such as Powerstack are still at the design stage, it is foreseeable that their potential for energy efficiency will render them more popular in the near future.

C. *ODA beyond Building Infrastructure*

While the 4-Pillar Framework touches on the connections of data centers to neighbouring buildings and the grid, at its core it concentrates on the four domains that are under the control of data center operators. However, from an abstract point of view, neighbouring buildings and the grid can be seen as components of the building infrastructure – albeit at larger scale – as the connection to them typically only encompasses infrastructure such as heating, cooling, and electricity. In ODA, use cases that extend beyond the data center are quite rare, not least because the availability of monitoring data and control typically ends at the outer walls of the data center. Such use cases can still be classified under the building infrastructure pillar in our ODA framework, but it should be noted that their practical implementation is more challenging compared to traditional data center-level ODA systems.

One example where ODA extends beyond the data center is the contractual relation with the respective utility, which in many cases requires data centers to notify them of imminent changes in their power consumption. The *Lawrence Livermore National Laboratory* (LLNL), for example, is required to notify their utility whenever its power consumption goes up or down by more than 750kW over a 15-minute time window [72]. Using *Fourier* transformations on historical monitoring data, they were able to identify power spike patterns and use these to forecast power consumption and meet the requirements of their utility.

VI. Conclusion

Operational Data Analytics in HPC is as complex as HPC operations themselves as it may cover building infrastructures, system hardware, system software as well as applications. Therefore, as more and more HPC sites are implementing and deploying ODA systems to improve their operations, there is a need for well-defined terminology to facilitate meaningful discussion among stakeholders. Also, tools are required to analyze, assess, and categorize such systems in order to better understand what they do and how they work. To this end, we have combined two well-established frameworks from the HPC and data analytics domains to create a framework that is applicable to ODA in HPC and covers all potential use cases in data centers: in order to demonstrate the usefulness of this new framework, we have applied it on a sizeable amount of use cases for ODA in the HPC domain that we had identified in an extensive literature survey. Grouping the individual use cases into the categories defined by the framework helps in understanding the challenges associated with each particular use case and in comparing multiple use cases with one another. In that sense, a major contribution of this paper is not just the framework itself but also the overview and analysis it gives of the current state and activities of the ODA community: it may also be useful for HPC sites who are planning to implement ODA systems to identify similar activities at other sites and benefit from their lessons learned.

Acknowledgment

Alessio Netti was supported by the European Union's Horizon 2020/EuroHPC research and innovation programme under grant agreement No. 956560 (REGALE). This work was supported by, and used the resources of, the Oak Ridge Leadership Computing Facility, located in the National Center for Computational Sciences at ORNL, which is managed by UT Battelle, LLC for the U.S. DOE (under the contract No. DE-AC05-00OR22725).

References

[1] N. Bourassa, W. Johnson, J. Broughton, D. M. Carter *et al.*, "Operational data analytics: Optimizing the national energy research scientific computing center cooling systems," in *Proc. of the ICPP 2019 Workshops*. ACM, 2019, pp. 5:1–5:7.

[2] K. Lepenioti, A. Bousdekis, D. Apostolou, and G. Mentzas, "Prescriptive analytics: Literature review and research challenges," *International Journal of Information Management*, vol. 50, pp. 57–70, 2020.

[3] T. Wilde, A. Auweter, and H. Shoukourian, "The 4 Pillar Framework for energy efficient HPC data centers," http://dx.doi.org/10.1007/s00450-013-0244-6, pp. 241–251, 2014.

[4] J. Yuventi and R. Mehdizadeh, "A critical analysis of power usage effectiveness and its use in communicating data center energy consumption," *Energy and Buildings*, vol. 64, pp. 90–94, 2013.

[5] J. Eitzinger, T. Gruber, A. Afzal, T. Zeiser *et al.*, "Clustercockpit—a web application for job-specific performance monitoring," in *Proc. of CLUSTER 2019*. IEEE, 2019, pp. 1–7.

[6] C. Guillen, W. Hesse, and M. Brehm, "The persyst monitoring tool - A transport system for performance data using quantiles," in *Proc. of the Euro-Par 2014 Workshops*. Springer, 2014, pp. 363–374.

[7] E. Bautista, M. Romanus, T. Davis, C. Whitney *et al.*, "Collecting, monitoring, and analyzing facility and systems data at the national energy research scientific computing center," in *Proc. of the ICPP 2019 Workshops*. ACM, 2019, p. 10.

[8] B. Schwaller, N. Tucker, T. Tucker, B. Allan *et al.*, "HPC system data pipeline to enable meaningful insights through analysis-driven visualizations," in *Proc. of CLUSTER 2020*. IEEE, 2020, pp. 433–441.

[9] U. Demirbaga, Z. Wen, A. Noor, K. Mitra *et al.*, "Autodiagn: An automated real-time diagnosis framework for big data systems," *IEEE Transactions on Computers*, 2021.

[10] L. Adhianto, S. Banerjee, M. Fagan, M. Krentel, G. Marin, J. Mellor-Crummey *et al.*, "HPCtoolkit: tools for performance analysis of optimized parallel programs," *Concurrency and Computation: Practice & Experience*, vol. 22, no. 6, pp. 685–701, 2010.

[11] J. Eastep, S. Sylvester, C. Cantalupo, B. Geltz *et al.*, "Global extensible open power manager: A vehicle for HPC community collaboration on co-designed energy management solutions," in *Proc. of ISC 2017*. Springer, 2017, pp. 394–412.

[12] W. Jiang, Z. Jia, S. Feng, F. Liu *et al.*, "Fine-grained warm water cooling for improving datacenter economy," in *Proc. of ISCA 2019*. ACM, 2019, pp. 474–486.

[13] M. Ott, W. Shin, N. Bourassa, T. Wilde *et al.*, "Global experiences with HPC operational data measurement, collection and analysis," in *Proc. of CLUSTER 2020*. IEEE, 2020, pp. 499–508.

[14] Y. Hui, B. H. Park, and C. Engelmann, "A comprehensive informative metric for analyzing HPC system status using the LogSCAN platform," in *Proc. of FTXS Workshop 2018*. IEEE, 2018, pp. 29–38.

[15] I. Laguna, S. Mitra, F. A. Arshad, N. Theera-Ampornpunt *et al.*, "Automatic problem localization via multi-dimensional metric profiling," in *Proc. of SRDS 2013*. IEEE, 2013, pp. 121–132.

[16] O. Tuncer, E. Ates, Y. Zhang, A. Turk *et al.*, "Online diagnosis of performance variation in HPC systems using machine learning," *IEEE Transactions on Parallel and Distributed Systems*, 2018.

[17] A. Borghesi, A. Bartolini, M. Lombardi, M. Milano *et al.*, "A semisupervised autoencoder-based approach for anomaly detection in high performance computing systems," *Engineering Applications of Artificial Intelligence*, vol. 85, pp. 634–644, 2019.

[18] C. Conficoni, A. Bartolini, A. Tilli, G. Tecchiolli *et al.*, "Energy-aware cooling for hot-water cooled supercomputers," in *Proc. of DATE 2015*. IEEE, 2015, pp. 1353–1358.

[19] R. E. Grant, K. T. Pedretti, and A. Gentile, "Overtime: A tool for analyzing performance variation due to network interference," in *Proc. of Exascale MPI Workshop 2015*. ACM, 2015, p. 4.

[20] C. Imes, S. Hofmeyr, and H. Hoffmann, "Energy-efficient application resource scheduling using machine learning classifiers," in *Proc. of ICPP 2018*. ACM, 2018, p. 45.

[21] A. Verma, P. Ahuja, and A. Neogi, "Power-aware dynamic placement of HPC applications," in *Proc. of ICS 2008*. ACM, 2008, pp. 175–184.

[22] C. Bash and G. Forman, "Cool job allocation: Measuring the power savings of placing jobs at cooling-efficient locations in the data center," in *Proc. of USENIX 2007*, vol. 138, 2007, p. 140.

[23] Y. Fan and Z. Lan, "DRAS-CQSim: A reinforcement learning based framework for HPC cluster scheduling," *Software Impacts*, vol. 8, p. 100077, 2021.

[24] J. Corbalan and L. Brochard, "EAR: Energy management framework for supercomputers," in *Proc. of IPDPS 2018*. IEEE, submitted.

[25] X. Lin, Y. Wang, and M. Pedram, "A reinforcement learning-based power management framework for green computing data centers," in *Proc. of IC2E 2016*. IEEE, 2016, pp. 135–138.

[26] Q. Guan and S. Fu, "Adaptive anomaly identification by exploring metric subspace in cloud computing infrastructures," in *Proc. of SRDS 2013*. IEEE, 2013, pp. 205–214.

[27] D. Shaykhislamov and V. Voevodin, "An approach for dynamic detection of inefficient supercomputer applications," *Procedia Computer Science*, vol. 136, pp. 35–43, 2018.

[28] R. Miceli, G. Civario, A. Sikora, E. César *et al.*, "Autotune: A plugin-driven approach to the automatic tuning of parallel applications," in *Proc. of PARA Workshop 2012*. Springer, 2012, pp. 328–342.

[29] C. Ţăpuş, I.-H. Chung, J. K. Hollingsworth *et al.*, "Active harmony: Towards automated performance tuning," in *Proc. of SC 2002*. IEEE, 2002, pp. 1–11.

[30] M. Naghshnejad and M. Singhal, "Adaptive online runtime prediction to improve HPC applications latency in cloud," in *Proc. of CLOUD 2018*. IEEE, 2018, pp. 762–769.

[31] J. Emeras, S. Varrette, M. Guzek, and P. Bouvry, "Evalix: Classification and prediction of job resource consumption on HPC platforms," in *Proc. of JSSPP Workshop 2015*. Springer, 2015, pp. 102–122.

[32] J. Xue, F. Yan, R. Birke, L. Y. Chen *et al.*, "PRACTISE: Robust prediction of data center time series," in *Proc. of CNSM 2015*. IEEE, 2015, pp. 126–134.

[33] E. Ates, O. Tuncer, A. Turk, V. J. Leung *et al.*, "Taxonomist: Application detection through rich monitoring data," in *Proc. of Euro-Par 2018*. Springer, 2018.

[34] M. R. Wyatt II, S. Herbein, T. Gamblin, A. Moody *et al.*, "PRIONN: Predicting runtime and IO using neural networks," in *Proc. of ICPP 2018*. ACM, 2018, p. 46.

[35] R. McKenna, S. Herbein, A. Moody, T. Gamblin *et al.*, "Machine learning predictions of runtime and IO traffic on high-end clusters," in *Proc. of CLUSTER 2016*. IEEE, 2016, pp. 255–258.

[36] O. DeMasi, T. Samak, and D. H. Bailey, "Identifying HPC codes via performance logs and machine learning," in *Proc. of CLHS Workshop 2013*. ACM, 2013, pp. 23–30.

[37] M. B. Kjærgaard, K. Arendt, A. Clausen, A. Johansen *et al.*, "Demand response in commercial buildings with an assessable impact on occupant comfort," in *Proc. of SmartGridComm 2016*. IEEE, 2016, pp. 447–452.

[38] P. Bodik, M. Goldszmidt, A. Fox, D. B. Woodard, and H. Andersen, "Fingerprinting the datacenter: automated classification of performance crises," in *Proc. of EuroSys 2010*. ACM, 2010, pp. 111–124.

[39] L. Bortot, W. Nardelli, and P. Seto, "Data centers are a software development challenge," in *Proc. of ICPP 2019*, 2019.

[40] A. Auweter, A. Bode, M. Brehm, L. Brochard *et al.*, "A case study of energy aware scheduling on SuperMUC," in *Proc. of ISC 2014*. Springer, 2014, pp. 394–409.

[41] X. Wu, A. Marathe, S. Jana, O. Vysocky *et al.*, "Toward an end-to-end auto-tuning framework in HPC powerstack," in *Proc. of CLUSTER 2020*. IEEE, 2020, pp. 473–483.

[42] J. Li, X. Ma, K. Singh, M. Schulz *et al.*, "Machine learning based online performance prediction for runtime parallelization and task scheduling," in *Proc. of ISPASS 2009*. IEEE, 2009, pp. 89–100.

[43] X. Zheng, Z. Zhou, X. Yang, Z. Lan *et al.*, "Exploring plan-based scheduling for large-scale computing systems," in *Proc. of CLUSTER 2016*. IEEE, 2016, pp. 259–268.

[44] H. Zhang, H. You, B. Hadri, and M. Fahey, "HPC usage behavior analysis and performance estimation with machine learning techniques," in *Proc. of PDPTA 2012*, 2012, p. 1.

[45] H. Shoukourian and D. Kranzlmüller, "Forecasting power-efficiency related key performance indicators for modern data centers using LSTMs," *Future Generation Computer Systems*, vol. 112, pp. 362–382, 2020.

[46] H. Shoukourian, T. Wilde, D. Labrenz, and A. Bode, "Using machine learning for data center cooling infrastructure efficiency prediction," in *Proc. of the IPDPS 2017 Workshops*. IEEE, 2017, pp. 954–963.

[47] A. Netti, D. Tafani, M. Ott, and M. Schulz, "Correlation-wise smoothing: Lightweight knowledge extraction for HPC monitoring data," in *Proc. of IPDPS 2021*. IEEE, 2021.

[48] A. Sîrbu and O. Babaoglu, "Towards operator-less data centers through data-driven, predictive, proactive autonomics," *Cluster Computing*, vol. 19, no. 2, pp. 865–878, 2016.

[49] C. Galleguillos, Z. Kiziltan, A. Netti, and R. Soto, "Accasim: a customizable workload management simulator for job dispatching research in HPC systems," *Cluster Computing*, vol. 23, no. 1, pp. 107–122, 2020.

[50] P.-F. Dutot, M. Mercier, M. Poquet, and O. Richard, "Batsim: a realistic language-independent resources and jobs management systems simulator," in *Proc. of JSSPP Workshop 2015*. Springer, 2015, pp. 178–197.

[51] D. Klusáček, M. Soysal, and F. Suter, "Alea–complex job scheduling simulator," in *Proc. of PPAM 2019*. Springer, 2019, pp. 217–229.

[52] A. Sîrbu and O. Babaoglu, "Power consumption modeling and prediction in a hybrid CPU-GPU-MIC supercomputer," in *Proc. of Euro-Par 2016*. Springer, 2016, pp. 117–130.

[53] A. Matsunaga and J. A. Fortes, "On the use of machine learning to predict the time and resources consumed by applications," in *Proc. of CCGrid 2010*. IEEE, 2010, pp. 495–504.

[54] A. Todd, A. Purkayastha, H. Egan, D. Sickinger *et al.*, "Artificial intelligence for data center operations (AI ops)," National Renewable Energy Laboratory, Hewlett-Packard Enterprise, Tech. Rep., 2021.

[55] S. Jha, J. Brandt, A. Gentile, Z. Kalbarczyk *et al.*, "Characterizing supercomputer traffic networks through link-level analysis," in *Proc. of CLUSTER 2018*. IEEE, 2018, pp. 562–570.

[56] J. L. Gustafson, *The end of error: Unum computing*. CRC Press, 2017.

[57] K. B. Ferreira, P. Bridges, and R. Brightwell, "Characterizing application sensitivity to OS interference using kernel-level noise injection," in *Proc. of SC 2008*. IEEE, 2008, p. 19.

[58] G. L. Stewart, G. A. Koenig, J. Liu, A. Clausen *et al.*, "Grid accommodation of dynamic HPC demand," in *Proc. of the ICPP 2019 Workshops*, 2019, pp. 1–4.

[59] M. K. Patterson, S. W. Poole, C.-H. Hsu, D. Maxwell *et al.*, "TUE, a new energy-efficiency metric applied at ORNL's jaguar," in *Proc. of ISC 2013*. Springer, 2013, pp. 372–382.

[60] D. G. Feitelson, "Metrics for parallel job scheduling and their convergence," in *Proc. of JSSPP Workshop 2001*. Springer, 2001, pp. 188–205.

[61] N. Chan, "A resource utilization analytics platform using grafana and telegraf for the savio supercluster," in *Proc. of PEARC 2019*, 2019, pp. 1–6.

[62] J. T. Palmer, S. M. Gallo, T. R. Furlani, M. D. Jones *et al.*, "Open XD-MoD: A tool for the comprehensive management of high-performance computing resources," *Computing in Science & Engineering*, vol. 17, no. 4, pp. 52–62, 2015.

[63] S. Williams, A. Waterman, and D. Patterson, "Roofline: an insightful visual performance model for multicore architectures," *Communications of the ACM*, vol. 52, no. 4, pp. 65–76, 2009.

[64] G. Ozer, S. Garg, N. Davoudi, G. Poerwawinata *et al.*, "Towards a predictive energy model for HPC runtime systems using supervised learning," in *Proc. of PMACS Workshop 2019*. Springer, 2019, pp. 626–638.

[65] M. Ott and D. Kranzlmüller, "Best practices in energy-efficient high performance computing," in *Workshops der INFORMATIK 2018*. Köllen Druck+Verlag GmbH, 2018, pp. 167–176.

[66] N. Bin Mohd Nor and M. H. Bin Selamat, "Review on green data center frameworks," in *Proc. of WICT 2014*. IEEE, 2014, pp. 338–343.

[67] S. E. Michalak, A. M. Bonnie, A. J. Montoya, C. B. Storlie *et al.*, "A temperature monitoring infrastructure and process for improving data center energy efficiency with results for a high performance computing data center," *Analytic Methods in Systems and Software Testing*, p. 329, 2018.

[68] S. M. Hussain, A. Wahid, M. A. Shah, A. Akhunzada *et al.*, "Seven pillars to achieve energy efficiency in high-performance computing data centers," *Recent Trends and Advances in Wireless and IoT-enabled Networks*, pp. 93–105, 2019.

[69] C. Bai, S. Kusi-Sarpong, and J. Sarkis, "An implementation path for green information technology systems in the ghanaian mining industry," *Journal of Cleaner Production*, vol. 164, pp. 1105–1123, 2017.

[70] Gartner, "Gartner IT glossary," *Technology Research*, 2013. [Online]. Available: https://www.gartner.com/en/glossary

[71] R. Soltanpoor and T. Sellis, "Prescriptive analytics for big data," in *Proc. of ADC 2016*. Springer, 2016, pp. 245–256.

[72] G. Abdulla, A. M. Bailey, J. Weaver, and R. Bockmon. (2018) Forecasting extreme site power fluctuations using fast fourier transformation. Lawrence Livermore National Laboratory. [Online]. Available: https://eehpcwg.llnl.gov/assets/sc18_workshop_forecasting_adbulla.pdf