HPC and energy efficiency using V-nets*

John W. Vásquez-Capacho[1] [0000−0003−3710−1086]

CAGE-SC3, Universidad Industrial de Santander, Bucaramanga, COLOMBIA
jwvascap@uis.edu.co
https://uis.edu.co/en/

Abstract. In today's era of exascale machines, the pressing issue of energy efficiency is more crucial than ever. This study explores the potential of V-nets, initially tested on small-scale machines, to be scaled up for larger systems that support parallelism. By capturing real-time data as sequences of discrete events, this project investigates how V-nets can effectively analyze these event sequences to diagnose system behavior in HPC systems. The focus is on constructing temporal patterns to assess the energy performance of scalable computing systems. While no specific system is tested, the analysis emphasizes the significance of this innovative formalism. It showcases the ability of V-nets to identify simultaneous event occurrences, detect partial sequences, and mitigate false positives. This research aims to bridge the gap between theoretical analysis and practical implementation in Industry 4.0, ultimately advancing the optimization of scalable computing systems.

Keywords: Scalable computing systems · V-nets · HPC Energy performance · Industry 4.0 · DES diagnosis

1 Introduction

Large-scale systems that consume a significant amount of energy must prioritize energy efficiency, which can be achieved through the use of tools like discrete event analysis. In today's world, there is a growing demand for energy and a corresponding increase in concern over the environmental impact of energy consumption, particularly in light of climate change. Through careful monitoring and analysis of energy usage using tools such as discrete event analysis, it is possible to optimize the energy efficiency of high-performance computing systems. This approach is applicable to a broad range of industries and systems, including manufacturing, transportation, and power generation. Implementing energy-efficient practices and using discrete-event analysis can result in significant cost savings and environmental benefits for large-scale systems. Large-scale systems can simultaneously reduce their environmental impact and improve their financial performance by decreasing operating costs and increasing their performance. Identifying bottlenecks and inefficiencies in the system can further improve overall efficiency and performance. The adoption of energy-efficient practices can

* CAGE-SC3 UIS

also improve a system's reputation and competitive edge by appealing to environmentally conscious consumers and stakeholders. As the use of large-scale infrastructure for IT applications has grown in recent years, the energy costs of their operation have become increasingly important. Many top supercomputers are now used for HPC industrial applications like cloud providers or streaming services, making the need to control energy costs even more critical. In summary, optimizing energy consumption and achieving energy efficiency is essential for large-scale systems, and the use of tools such as discrete event analysis can help identify areas for improvement and lead to significant environmental and cost-saving benefits [1]. The issue of energy consumption in large-scale computing systems has become a topic of intense debate in both research and policy circles. The challenge is that increasing the performance of these systems often leads to an increase in resource usage, resulting in higher energy consumption and a negative impact on the environment. Various strategies have been developed to address this issue, including hardware changes, energy-efficient software usage, and new rules for computing resource usage. However, two critical factors must be considered when seeking to improve energy efficiency in large-scale or cluster computing: operational cost and system reliability. By improving the energy efficiency of cluster systems, it is possible to reduce electricity consumption, excess heat, operational costs, and system reliability issues, providing significant benefits [2].

The following paper is structured into five sections for ease of understanding. The first section serves as the introduction to the paper. In section two, the current state of the art is discussed. Section three introduces "V-nets," a new formalism used to diagnose energy efficiency in large-scale systems. A case study highlighting the importance of detecting simultaneity and repetition is presented in Section Four. Finally, the fifth section concludes the paper and outlines possible avenues for future research.

2 State of the art

To ensure the high performance of electronic devices in an energy supply system, it is crucial to identify the most relevant discrete events that have a significant impact on energy efficiency. These discrete events occur at specific moments and can affect the device's energy consumption. By identifying the most relevant events, energy consumption can be optimized and associated costs can be reduced. For example, scheduling energy savings measures during peak energy consumption times can be an effective strategy to reduce energy costs. Now, by examining and researching techniques to boost energy storage, the article [3] discusses strategies to improve the energy efficiency of electric transportation, and it is organized into categories based on techniques, strategies, and answers. Furthermore, in [4] the contribution lies in the exploration of the feasibility of achieving runtime adaptiveness through formal methods for enhanced energy efficiency analysis in computer systems; the survey of existing approaches, along with challenges and initial outcomes, supports this perspective. The paper [5]

centers on energy-efficient design in Ethernet PONs, utilizing formal methods to optimize energy consumption while upholding QoS demands. Verification-guided policies demonstrate substantial energy savings, emphasizing the practical implications of this approach. On the other hand, in [6], a new approach for detecting and diagnosing failures in complex systems with discrete event dynamics is proposed. This approach involves a decentralized protocol, where individual agents are assigned to monitor specific aspects of the system and report any deviations to a central coordinator. By analyzing the data reported by the agents, the coordinator can identify the most likely cause of the failure, allowing for timely intervention and maintenance. This decentralized protocol has potential applications in the analysis of energy efficiency in complex systems, as it enables real-time monitoring and diagnosis of potential inefficiencies or failures. Through the use of decentralized monitoring, the system can accurately detect and diagnose issues, leading to more efficient energy usage and reduced waste. In summary, this approach offers a promising avenue to improve the efficiency and reliability of complex systems in the energy sector. However, optimization of energy efficiency in large-scale computer systems, such as cluster systems, is a pressing issue due to the high energy costs and environmental impact. To enhance energy efficiency, various techniques and strategies are being employed, including hardware changes, energy-aware software usage, and new rules for the utilization of computing resources. Two main approaches are used for energy management in these systems: static power management, which uses low-energy hardware components, and dynamic power management, which employs extensible components and software to optimize energy consumption. Although dynamic voltage and frequency scaling algorithms are popular techniques to reduce energy consumption in high-performance computing systems, they can potentially compromise system performance [7]. However, improving energy efficiency in cluster systems presents unique challenges, such as ensuring the reliability of the system while reducing energy consumption. Failure to address these challenges may increase the risk of system failures and data loss. Despite these challenges, optimizing energy efficiency in cluster systems through a combination of energy management techniques and strategies can lead to significant economic and environmental benefits. In [8], the significance of energy efficiency in scientific computing is discussed, particularly in high-performance computing (HPC) and high-performance computing (HTC). The article presents a comparison between the ARMv7 and Intel Xeon architecture using CMS workloads and describes the tools and techniques available to measure and monitor energy consumption on HTC systems. IgProf, an open-source application performance profiler, and its recently added energy profiling features and 64-bit ARM support are also introduced. The techniques used include both instrumentation and sampling profiling, as well as external probing devices and onboard chips for monitoring energy consumption. The article concludes that a full server-grade system is necessary for a proper comparison of power efficiency for an architecture and that software optimizations for energy efficiency can be supported by mapping energy consumption measurements to functions and methods within an

executing process. Now, in [9], a comprehensive overview of the tools, methods, and challenges associated with measuring and optimizing the energy efficiency of HPC systems is provided. The article covers topics such as energy efficiency metrics, power and energy measurement tools, performance measurement and analysis suites, energy models, and tuning approaches. It also discusses the challenges of optimizing performance under a maximum amount of electrical power available to an HPC system, including hardware overprovisioning, scheduling, node configuration, and load imbalance. Additionally, the importance of integrating performance analysis and energy efficiency optimizations in a unified environment is highlighted. The authors conclude by discussing future work in the development of performance counter-based energy models and the integration of tuning cycles and performance measurement. Finally, in [10], a comprehensive literature review is provided on the use of artificial intelligence (AI) and machine learning (ML) to improve energy efficiency in HPC and 5G networks. The article covers various applications and techniques, including architectural design, hardware and software technologies, advanced cooling technologies, waste heat reuse, and deep reinforcement learning. It also discusses the benefits and limitations of using AI in HPC systems and highlights the potential benefits of using AI in data center optimization, such as reducing cooling bills and improving resource allocation. The challenges of achieving energy efficiency in HPC and big data are also discussed, as well as the need for more thorough evidence on why AI is needed in HPC systems. The article provides valuable information on the use of AI and ML to improve energy efficiency in HPC and 5G networks. When it comes to large-scale computer systems such as cluster systems, achieving energy efficiency without sacrificing system performance is crucial. There are various techniques and strategies available to improve energy use, including hardware modifications, energy-aware software, and resource usage rules. Two approaches to energy management are static and dynamic, with the former focusing on low-power hardware components and the latter using expansible components and software to optimize energy consumption. While dynamic voltage and frequency scaling algorithms are commonly used to reduce energy consumption in high-performance computing systems, they can have a negative impact on system performance. Improving energy in cluster systems is a challenging task, as it requires balancing operational costs with system reliability. Thus, measures need to be taken to ensure system reliability while improving energy efficiency. A study [11] emphasizes the importance of industrial development in energy systems and the role of artificial intelligence (AI) in the energy market. The authors describe various AI technologies such as fuzzy logic systems, artificial neural networks, genetic algorithms, and expert systems that can improve energy efficiency, management, transparency, and utilization of renewable energies. The study discusses research opportunities and policy recommendations to improve sustainability in the energy industry and highlights the need for continued investment in global research on AI and data-driven models. Several authors have proposed a framework comprising pillars to enhance energy efficiency in HPC data centers. Although the specific details of the framework are not outlined in

the summary, it is expected to encompass critical aspects such as hardware design, cooling infrastructure, power management, and workload scheduling. These articles make valuable contributions to the field of energy efficiency in HPC systems by offering a systematic approach to optimize energy consumption within data centers [12],[13]. In this paper, failure is assumed to be the consequence of erroneous sequences or traces of events, not as a single fault event as in many approaches. For instance, a failure occurs when: events happen in a different order, if some of them are repeated, or when they happen simultaneously. The supervision system will have a problem if the temporal pattern recognizes (as good sequences) traces such as *aabc*, *bac* or *abbcc*, and many other combinations; furthermore, if simultaneous occurrences of these events are not detected. Using a tool for diagnosing energy efficiency in HPC systems could offer substantial cost savings potential. These savings can have a significant impact on monthly and annual expenses, allowing for investments in research endeavors or upgrades to the supercomputer infrastructure.

3 "V-nets," a new formalism used to diagnose energy efficiency in large-scale systems

In their work, [14] [15] proposed a new formalism known as V-nets. It serves as a tool for modeling and analyzing complex systems that evolve over time and are characterized by specific events. V-nets employ a graphical representation that differs from Petri nets, for instance, V-nets use squares instead of circles to represent events (no places), and arcs connecting them including time restrictions with frequency occurrences. This formalism comes with a set of properties defining the system's behavior, including rules for firing transitions, rules for simultaneous events, and false-positive analysis. V-nets offer a more accurate and efficient means of modeling complex systems and diagnosing faults than other formalism, particularly in the context of Cyber-Physical Systems and industrial applications. Furthermore, V-nets have potential applications in energy efficiency diagnosis by graphically representing the system's discrete events and their relationships. Analyzing the V-net model enables the identification of abnormal energy consumption patterns caused by specific event sequences, leading to energy optimization by repairing or replacing faulty components. Thus, V-nets are powerful for detecting energy efficiency issues in complex systems.

3.1 Formalism description

Definition 1: A V-net (V_N) is defined as the tuple $\langle \xi, \mathcal{T}, \mathcal{G}$, INIT, END, Frec, R $\rangle$ such that:

- $\xi \subseteq E$, in which ξ is the set of events involved in the V_N,
- $\mathcal{T}$ is the set of temporal constraints of the V-net,
- $\mathcal{G} = (\mathcal{V}, \mathcal{A}, \mathbf{Ev})$ is a directed graph in which:
  - $\mathcal{V}$ represents the events of ξ

  - Arcs $\mathcal{A}$ represents the different time constraints between events.
  - **Ev**:(True/false) represents the activation or deactivation of arcs $\mathcal{A}$ according to the evolution of the timed observations.
- INIT corresponds to the event that initiates the event sequence, see Fig. 1
- END corresponds to the event that ends the event sequence, see Fig. 1. In a V-net, the initial event and the final event could be the same.
- **Frec**: corresponds to the frequencies for each event in the event sequence. Fox example, in the interval or temporal constraint $a^{f_a}[I^-, I^+]^{f_b}b$; $f_{a(b)}$ is the frequency of the event a (b) on the interval $[I^-, I^+]$.
- R contains the set of logical predicates that correspond to the restrictions and warnings that confirm the V-net recognition. Examples:
  - **Frec**(a):5
  - **Frec**(a):1 $\wedge$ **Frec**(b):1
  - **Warning:** a=b

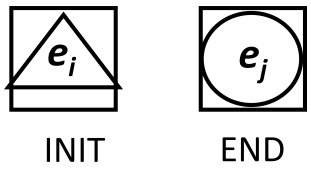

Fig. 1. Graphical representation of INIT and END

Similarly to language theory, in the V-nets (V_N) the alphabet corresponds to ξ. The timed language $L^T(\xi, \mathcal{T})$ of each V_N will be referenced in the strings w that can be generated according to the configuration of the events in the graph. The V-net and the concatenation of the elements in the strings will be restricted by the set of temporal constraints $\mathcal{T}$.

The paper [16] describes the importance of fault detection and diagnosis (FDD) in ensuring the reliable operation of systems, and how FDD approaches such as Analytical Redundancy (AR), Principal Component Analysis (PCA), and Discrete Event System (DES) can be used. The authors highlight the challenge of measurement inconsistency in FDD schemes, which can result in a weak diagnosis, and propose a solution in the form of a Measurement Inconsistent Discrete Event System (MIDES) framework that uses MI parameters for FDD. The efficacy of the proposed method is illustrated using a pump valve system and a MIDES-based intrusion detection system. From this paper, we took for the next example two discrete events, referenced with the dynamic host configuration protocol (DHCP). Event a: DHCPDISCOVER, and event b: DHCPOFFER. The example presented in Fig. 2 has $\xi = \{a, b\}$; and the timed language is $L^T = \{(a,t_1), (a,t_2), (a,t_3), (b,t_4)\}$ with $t_1 < t_2 < t_3 < t_4$. This V-net represents the behavior of the dynamic host configuration protocol (DHCP), where the first event that occurs is a, and it is repeated three times with a time restriction of

1 to 2 time units between each pair of events. The final event that occurs in this V-net is the event b which happens between 1 to 2-time units after that the third occurrence of a. The frequency of an event e_i in one event sequence is finite. If f^{max} is the maximum number of timed occurrences in a sequence, then the maximal frequency of an event e_i will be $\mathbf{Frec}(e_i) \leq f^{max}$. For the example in Fig. 2, we have $f_a^{max} = 3$ and $f_b^{max} = 1$.

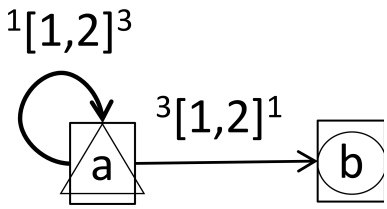

Fig. 2. Example of a V-net (V_N)

4 GUANE efficiency energy

The Universidad Industrial de Santander (UIS) has a supercomputing cluster named Guane SC3. Guane is a heterogeneous cluster consisting of 26 computing nodes, each equipped with dual Intel Xeon processors and up to 256 GB of RAM. In addition, there are two NVIDIA Tesla GPUs and one Intel Xeon Phi coprocessor per node, making a total of 52 GPUs and 26 Xeon Phi coprocessors. This cluster has a peak performance of 292.8 Teraflops and is used for scientific simulations and research in different fields, such as physics, chemistry, biology, and engineering. This cluster is connected to a high-speed Infinity-band network, providing low-latency and high-bandwidth communication between nodes. They also have a shared parallel file system with a capacity of 1.2 Petabytes, allowing researchers to store and access large amounts of data. The UIS supercomputing facilities are managed by the High-Performance Computing (HPC) group, which provides technical support and training to researchers and students from different departments and institutions. The HPC group also collaborates with national and international research projects, promoting scientific and technological development in Colombia and the Latin American region. Therefore, they could use V-nets, a formalism for modeling and analyzing discrete event dynamic systems, to diagnose energy efficiency problems in complex systems. Representing the system's discrete events and their relationships in a graphical way allows for a more visual and intuitive understanding of the system's behavior; V-nets can be used to identify specific sequences of events that lead to higher energy consumption and optimize the system accordingly. Using V-nets, the HPC group can detect and diagnose abnormal energy consumption patterns. Then, by identifying the specific sequences of events that lead to higher energy consumption, they can

optimize the system to reduce energy consumption. In addition, V-nets can be used to detect faulty system components that can contribute to higher energy consumption. Through the diagnosis of these faulty components, they can be repaired or replaced, leading to a more energy-efficient system. Using V-nets for the diagnosis of energy efficiency in complex systems provides a powerful tool for the HPC group to optimize their systems and reduce energy consumption, ultimately leading to cost savings and a more sustainable approach to computing. Let us consider the following representative discrete events in HPC operation:

- Event A (ST: Start-up of the supercomputer): The HPC system is started up, and all necessary hardware and software components are initialized. The cooling system is activated to ensure that the system temperature remains within a safe operating range.
- Event B (ID: Idle state of the supercomputer): The HPC system is in an idle state, waiting for a job to be submitted. During this state, the system consumes minimal power and the cooling system maintains the temperature within an acceptable range.
- Event C (JS: Job submission): A job is submitted to the HPC system. The system receives the job and verifies that all necessary resources are available to complete the job. If resources are available, the system transitions to the next state.
- Event D (EX: Job execution): The job is executed on the HPC system. The system assigns the necessary resources to the job, and the cooling system works to maintain the temperature within an acceptable range. The system monitors the job to ensure that it is executing correctly.
- Event E (CD: Completion of the job and shutdown of the supercomputer): The job is completed and the system is completely shut down. The cooling system is turned off, and the system's resources are released. The system is then ready to receive another job submission.
- Event G (JC: Job cancellation): The job is canceled before completion. The system must release the assigned resources and transition back to the idle state. This event may occur due to user cancellation, hardware failure, or other reasons.
- Event H (IS: System idle due to lack of jobs): The HPC system remains idle for an extended period due to a lack of submitted jobs. During this state, the system may consume more power than in the idle state, as some components may remain active.
- Event J (HE: High energy consumption state): This event represents a state of high energy consumption, such as when the HPC system is running at full capacity, executing many jobs, or using all available resources. During this state, the cooling system must work harder to maintain the temperature within an acceptable range, and the system may consume significantly more power than in idle state.

This process ensures that the HPC system is used efficiently and safely. By properly managing the system's resources and monitoring its performance, the

system can process jobs quickly and accurately while maintaining a stable temperature and consuming minimal power. Therefore, the behavior of the system can be described as a sequence of events that occur during its operation. These events include starting up the system, submitting jobs to the system, executing jobs, and shutting down the system. Understanding the normal behavior of an HPC system is essential to optimize its energy consumption and ensure its reliable operation. In one event sequence, Event B (ID) can be repeated multiple times, as the system may remain in an idle state waiting for new job submissions. Event C (JS) and Event D (EX) cannot occur simultaneously, as the system must verify that all necessary resources are available before executing the job. The event G (JC) can occur at any time during job execution, and the system must release the assigned resources and transition back to idle state. Event H (IS) and Event J (HE) may occur simultaneously, as the system may consume more power than in the idle state when waiting for new job submissions, and high energy consumption may occur during job execution. However, if the system remains in a high energy consumption state for an extended period without executing any jobs, it could be considered a fault and waste of energy. Now, the following event sequence will describe the normal behavior of an HPC system using the events mentioned earlier, including time restrictions between events:

The following description has been structured according to the expertise and experiences of the HPC group at UIS. The sequence of events begins with Event A (ST: Start-up of the supercomputer), the HPC system starts, and all necessary hardware and software components are initialized. The cooling system is activated to ensure that the system temperature remains within a safe operating range. Between 4 and 5 time units after, Event B (ID: Idle state of the supercomputer) occurs, then, the HPC system is in an idle state, waiting for a job to be submitted. During this state, the system consumes minimal power and the cooling system maintains the temperature within an acceptable range. This event can be repeated multiple times as the system waits for job submissions - Duration: indefinite until Event C occurs, they may have simultaneous occurrence. The next event is Event C (JS: Job submission) with a duration between 2 and 3 time units. A job is submitted to the HPC system. The system receives the job and verifies that all necessary resources are available to complete the job; if the resources are available, the system transitions to the next state. After the Event C, Event D (EX: Job execution) indicates that the job is executed on the HPC system. The system assigns the necessary resources to the job, and the cooling system works to maintain the temperature within an acceptable range. The system monitors the job to ensure that it is executing correctly, the duration of this process could be between 5 and 10-time units. Event E (CD: Job completion and shutdown of the supercomputer) - Duration: between 4 and 5-time units. Currently, the job is completed, and the system is safely shut down. The cooling system is turned off, and the system's resources are released. The system is then ready to receive another job submission. Again, the event B (ID: Idle state of the supercomputer) - Duration: indefinite is activated. Therefore

the system returns to the idle state, waiting for another job submission. If there are no job submissions for an extended period between 50 and 60-time units, this will lead to event H (IS). Event H (IS: System idle due to lack of jobs) - Duration: indefinite is activated. The HPC system remains idle for an extended period due to a lack of submitted jobs. During this state, the system may consume more power than in the idle state, as some components may remain active until Event C occurs. Event G (JC: Job cancellation) - Duration: between 4 and 5-time units. The job is canceled before it is complete. The system must release the assigned resources and transition back to the idle state. This event can occur simultaneously with Event D (EX) if a job is canceled during execution. Event J (HE: High energy consumption state): This event represents a state of high energy consumption, such as when the HPC system is running at full capacity, executing many jobs, or using all available resources. During this state, the cooling system must work harder to maintain the temperature within an acceptable range, and the system may consume significantly more power than in the idle state. This event can occur simultaneously with Event D (EX) if there are multiple jobs that are executed on the system. Let us assume event J as the final event for an event sequence in this analysis.

A temporal pattern to detect the behavior of this example is presented in Fig. 3 using the new formalism V-net $\langle \xi, \mathcal{T}, \mathcal{G},$ INIT, END, Frec, R $\rangle$ in which the events from which timed observations are taken are given by $\xi = \{A, B, C, D, E, G, H, J, \beta^*, \mu^*, \pi^*, \sigma^*\}$, and R: **Frec**$(J)$:1, **Warning**: β^* because if the system is idle (H) cannot exist high energy consumption (J) at time, **Warning**: μ^* because the system requires a minimal time for pass from a job submission C to a job execution D.

Possible simultaneous situations:

β^* corresponds to the simultaneous occurrence of H and J.

μ^* corresponds to the simultaneous occurrence of C and D.

π^* corresponds to the simultaneous occurrence of B and C.

σ^* corresponds to the simultaneous occurrence of D with G.

With this model, the number of timed observations for each event, their order, and simultaneity are detected. There are many possibilities of event sequences for the correct behavior of the system; from the totality, two possible matrices of occurrences are presented in Fig. 4. For example, Matrix 1 with 8 occurrences (including INIT and END) because in this matrix is assumed the simultaneous occurrence of Events B and C in Oc1, and two computational tasks were developed. In Matrix 2, these events do not occur simultaneously, they happen in Oc1 and Oc2, with 11 occurrences in total.

Three evaluation event sequences were used to test the V-net, and the results are exposed in Fig. 5. The first event sequence corresponds to S_1. These data come from the process, which recognizes the V-net as a good trace in 100%. For this example, the sequence of events S_2 is recognized at 73%. The last sequence of events analyzed is S_3, which was recognized at 43%, and it had a warning β^* at (β^*, 13).

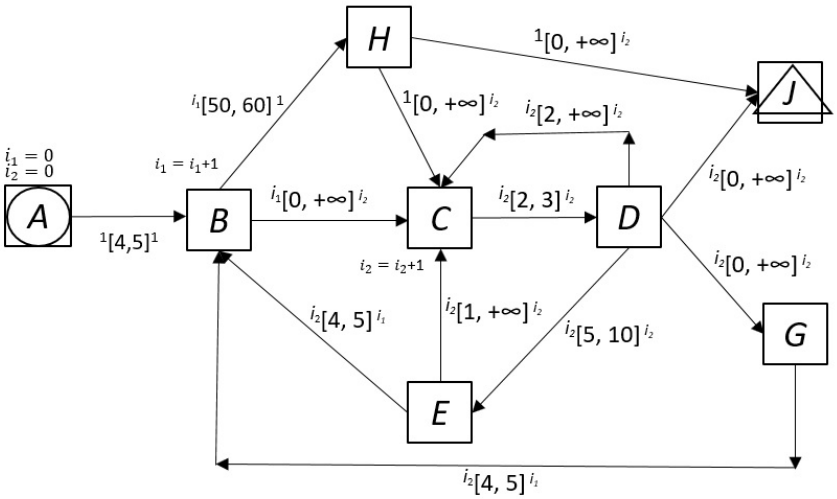

Fig. 3. Solution using V-net

Occurrence Matrix 1

	A	B	C	D	E	G	H	J	β*	μ*	π*	σ*
INIT	1	0	0	0	0	0	0	0	0	0	0	0
Oc1	0	0	0	0	0	0	0	0	0	0	1	0
Oc2	0	0	0	1	0	0	0	0	0	0	0	0
Oc3	0	0	0	0	1	0	0	0	0	0	0	0
Oc4	0	1	0	0	0	0	0	0	0	0	0	0
Oc5	0	0	1	0	0	0	0	0	0	0	0	0
Oc6	0	0	0	1	0	0	0	0	0	0	0	0
END	0	0	0	0	0	0	0	1	0	0	0	0

Occurrence Matrix 2

	A	B	C	D	E	G	H	J	β*	μ*	π*	σ*
INIT	1	0	0	0	0	0	0	0	0	0	0	0
Oc1	0	1	0	0	0	0	0	0	0	0	0	0
Oc2	0	0	1	0	0	0	0	0	0	0	0	0
Oc3	0	0	0	1	0	0	0	0	0	0	0	0
Oc4	0	0	0	0	1	0	0	0	0	0	0	0
Oc5	0	0	1	0	0	0	0	0	0	0	0	0
Oc6	0	0	0	1	0	0	0	0	0	0	0	0
Oc7	0	0	0	0	1	0	0	0	0	0	0	0
Oc8	0	1	0	0	0	0	0	0	0	0	0	0
Oc9	0	0	0	0	0	0	1	0	0	0	0	0
END	0	0	0	0	0	0	0	1	0	0	0	0

Fig. 4. Occurrence matrices

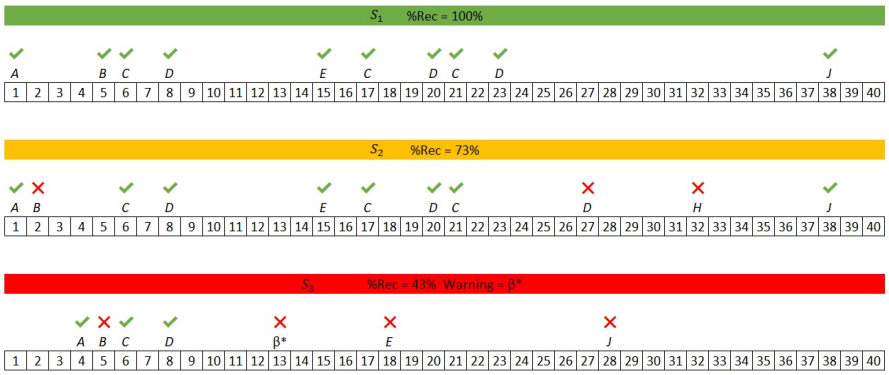

Fig. 5. Evaluation of the V-net with the event sequences S_1, S_2, and S_3

V-Net is a diagnosis tool that can improve the energy operation of an HPC system by detecting abnormal behavior in the system's event sequence. By analyzing the system's event logs, V-Net can identify patterns of behavior that deviate from the normal event sequence and indicate potential energy inefficiencies or faults in the system. For example, V-Net can detect if the system remains in a high energy consumption state for an extended period without executing any jobs, indicating a waste of energy. Moreover, V-Net can also provide real-time recommendations to optimize the system's energy consumption based on its current state and if the system is in a high-energy consumption state, V-Net can recommend reducing the number of jobs being executed simultaneously or prioritizing low-energy consumption jobs. By optimizing the system's energy consumption, V-Net can improve the HPC system efficiency and reduce energy costs. In summary, V-Net can improve the energy operation of an HPC system by detecting abnormal behavior in the system's event sequence and providing real-time recommendations to optimize the system's energy consumption.

The HPC cluster Guane SC3 at the Universidad Industrial de Santander has a maximum power consumption of 173.4 kW, according to the manufacturer's specifications. Assuming 24/7 operation with 8 hours of maintenance per month, the estimated total power consumption would be around 120,576 kWh per month, based on the maximum power consumption. However, actual power consumption can vary depending on factors such as workload, cooling efficiency, and power management strategies. Considering an average power consumption of 500 kW, the estimated monthly energy consumption for running the supercomputer for 24 hours a day with 8 hours of maintenance would be approximately 360,000 kWh, resulting in an electricity bill of around 72,000 USD per month based on Colombia's energy prices of 20 cents per kWh. It's important to note that this is an estimate and that the actual energy consumption and cost can vary depending on several factors. Nonetheless, it underscores the fact that running

an HPC system can be a significant expense. Thus, implementing energy-efficient strategies, such as those discussed earlier, can help reduce energy consumption and cost. The V-net diagnosis model can help reduce energy consumption in an HPC system by identifying energy inefficiencies and providing recommendations to optimize the system's energy usage. Using V-net, it is possible to detect abnormal behavior in the system and identify specific areas that need improvement. For example, the V-net model can identify patterns of energy consumption during idle states and job execution phases. If the model detects a long period of idle time with high energy consumption, it could recommend adjusting the cooling system's settings to reduce energy usage during these periods. Similarly, if the model identifies that the system is consuming excessive energy during job execution phases, it could recommend optimizing the system's resource allocation to reduce unnecessary energy consumption. Therefore, by identifying areas of inefficiency and providing targeted optimization recommendations, the V-net model could help reduce energy consumption in an HPC system by up to 15-18%. This can result in significant cost savings over time while ensuring reliable and efficient system operation. Through the implementation of these measures and taking advantage of the V-nets tool, it is possible to achieve a reduction in energy consumption and cost savings in the range of 15 to 18%. Nevertheless, it is important to note that these are approximate values, and the actual reduction may vary depending on the specific system and its operating conditions. If we assume that using V-nets can lead to a 15-18% reduction in energy consumption, then the monthly electricity bill for running the supercomputer would be reduced by approximately 10,800-12,960 USD. This translates to annual cost savings of approximately 129,600-155,520 USD. These cost savings are significant and can be used to invest in other areas of research or to upgrade the supercomputer infrastructure. In addition, reducing energy consumption can also have a positive impact on the environment by reducing carbon emissions and promoting sustainability. It is important to note that the actual cost savings may vary depending on various factors such as the specific usage patterns of the supercomputer and the efficiency of the cooling system. However, the implementation of V-nets as a diagnostic tool can still provide valuable information and help identify areas where energy consumption can be optimized.

5 Conclusion and future work

In conclusion, the literature review highlights the importance of energy efficiency in HPC systems due to the high energy consumption and associated costs. The use of V-nets as a diagnostic tool to formalize the problem as a temporal pattern recognition task has been shown to be an effective approach for identifying abnormal behavior and optimizing energy consumption. The construction of the V-net temporal pattern involves identifying discrete events during the operation of the supercomputer and using them to create a model that can classify the system's behavior. The V-nets can then be used to detect abnormal behavior, such as unnecessary use of cooling systems or idling, which can be optimized

to reduce energy consumption and costs. The potential cost savings from using V-nets to diagnose energy efficiency in HPC systems could be significant, with estimates ranging from 15 to 18 percent. This translates to significant monthly and annual cost savings, which can be invested in other areas of research or used to upgrade the supercomputer infrastructure. Future work in this area could involve further refining the V-net model to improve its accuracy and extending it to other types of HPC systems or other energy-intensive applications. Additionally, the implementation of energy-saving measures based on the V-net diagnosis could be evaluated and optimized for further efficiency improvements. The use of V-nets as a diagnostic tool for improving energy efficiency in HPC systems has great potential for reducing energy consumption, saving costs, and promoting sustainability.

References

1. e. a. F. Mantovani, Performance and energy consumption of hpc workloads on a cluster based on arm thunderx2 cpu, Future Generation Computer Systems 112 (2020) 800–818. doi:https://doi.org/10.1016/j.future.2020.06.033.
2. C. J. B. Hernandez, D. A. Sierra, S. Varrette, D. L. Pacheco, Energy efficiency on scalable computing architectures, in: 2011 IEEE 11th International Conference on Computer and Information Technology, 2011, pp. 635–640. doi:10.1109/CIT.2011.108.
3. N. V. Martyushev, B. V. Malozyomov, I. H. Khalikov, V. A. Kukartsev, V. V. Kukartsev, V. S. Tynchenko, Y. A. Tynchenko, M. Qi, Review of methods for improving the energy efficiency of electrified ground transport by optimizing battery consumption, Energies 16 (2) (2023) 729.
4. R. Calinescu, S. Kikuchi, Formal methods@ runtime, in: Foundations of Computer Software. Modeling, Development, and Verification of Adaptive Systems: 16th Monterey Workshop 2010, Redmond, WA, USA, March 31-April 2, 2010, Revised Selected Papers 16, Springer, 2011, pp. 122–135.
5. S. Petridou, S. Basagiannis, L. Mamatas, Formal methods for energy-efficient epons, IEEE Transactions on Green Communications and Networking 2 (1) (2017) 246–259.
6. R. Debouk, S. Lafortune, D. Teneketzis, Coordinated decentralized protocols for failure diagnosis of discrete event systems, Discrete Event Dynamic Systems 10 (1–2) (2000) 33–86. doi:10.1023/A:1008335115538.
   URL https://doi.org/10.1023/A:1008335115538
7. S. Irani, G. Singh, S. Shukla, R. Gupta, An overview of the competitive and adversarial approaches to designing dynamic power management strategies, IEEE Transactions on Very Large Scale Integration (VLSI) Systems 13 (12) (2005) 1349–1361. doi:10.1109/TVLSI.2005.862725.
8. D. Abdurachmanov, P. Elmer, G. Eulisse, R. Knight, T. Niemi, J. K. Nurminen, F. Nyback, G. Pestana, Z. Ou, K. Khan, Techniques and tools for measuring energy efficiency of scientific software applications, Journal of Physics: Conference Series 608 (1) (2015) 012032. doi:10.1088/1742-6596/608/1/012032.
   URL https://dx.doi.org/10.1088/1742-6596/608/1/012032
9. R. Schöne, J. Treibig, M. F. Dolz, C. Guillén, C. B. Navarrete, M. Knobloch, B. Rountree, Tools and methods for measuring and tuning the energy efficiency of hpc systems, Sci. Program. 22 (2014) 273–283.

10. A. H. Kelechi, M. H. Alsharif, O. J. Bameyi, P. J. Ezra, I. K. Joseph, A.-A. Atayero, Z. W. Geem, J. Hong, Artificial intelligence: An energy efficiency tool for enhanced high performance computing, Symmetry 12 (6) (2020). doi:10.3390/sym12061029.
URL https://www.mdpi.com/2073-8994/12/6/1029
11. T. Ahmad, H. Zhu, D. Zhang, R. Tariq, A. Bassam, F. Ullah, A. S. AlGhamdi, S. S. Alshamrani, Energetics systems and artificial intelligence: Applications of industry 4.0, Energy Reports 8 (2022) 334–361. doi:https://doi.org/10.1016/j.egyr.2021.11.256.
URL https://www.sciencedirect.com/science/article/pii/S2352484721014037
12. T. Wilde, A. Auweter, H. Shoukourian, The 4 pillar framework for energy efficient hpc data centers, Comput Sci Res Dev 29 (2014) 241–251. doi:10.1007/s00450-013-0244-6.
URL https://doi.org/10.1007/s00450-013-0244-6
13. S. Hussain, et al., Seven Pillars to Achieve Energy Efficiency in High-Performance Computing Data Centers, Springer, Cham, 2019. doi:10.1007/978-3-319-99966-1$_9$.
$URLhttps://doi.org/10.1007/978-3-319-99966-1_9$
14. J. W. Vasquez Capacho, C. G. Perez Zuñiga, Y. A. Muñoz Maldonado, A. Ospino Castro, Simultaneous occurrences and false-positives analysis in discrete event dynamic systems, Journal of Computational Science 44 (2020) 101162. doi:https://doi.org/10.1016/j.jocs.2020.101162.
URL https://www.sciencedirect.com/science/article/pii/S1877750320304634
15. J. Vasquez-Capacho, V-nets, new formalism to manage diagnosis problems in cyber-physical systems (cps) and industrial applications, IFAC-PapersOnLine 53 (5) (2020) 197–202, 3rd IFAC Workshop on Cyber-Physical & Human Systems CPHS 2020. doi:https://doi.org/10.1016/j.ifacol.2021.04.224.
URL https://www.sciencedirect.com/science/article/pii/S2405896321004080
16. M. Agarwal, S. Biswas, S. Nandi, Discrete event system framework for fault diagnosis with measurement inconsistency: case study of rogue dhcp attack, IEEE/CAA Journal of Automatica Sinica 6 (3) (2019) 789–806. doi:10.1109/JAS.2017.7510379.