An interoperable runtime for distributed power management of large-scale HPC systems based on DDS.

Giacomo Madella

Alma Mater Studiorum, Università di Bologna
Research Topic: Piattaforme distribuite, sicure e interoperabili per l'efficientamento dei sistemi di calcolo larga scala

1 Introduction

In recent years, the rapid growth of supercomputing systems has led to an increased demand for efficient energy management strategies. As the landscape of computational capabilities continues its rapid expansion, the imperative for sustainable energy management solutions becomes even more pronounced. The formidable computational power exerted by modern supercomputers is often contrasted with the considerable energy consumption required to power their operations. This phenomenon is also evident in the growing interest in green computing, rather than pure and simply power[1][2][3]. In this context, the pursuit of innovative energy management strategies becomes not only an economic consideration but, more importantly a fundamental effort to reduce the ecological consequences of increasing energy consumption. Moreover, the performance of computing elements is inherently constrained by power consumption, implying that enhancing energy efficiency translates to achieving greater peak performance.

Modern architectures implement an integrated thermal and power controller on the die, the purpose of which is to provide maximum performance within physical and externally imposed limits. In addition to this, other tasks have been added to energy and thermal control, including: (i) exchanging messages with multiple agents (such as Node-manager, Board Management Controller (BMC), and operating systems) modifying its control action to satisfy all the constraints given from these agents; (ii) be aware of the security implications of its control actions, avoiding system-level fatal operating points and detecting and preventing security-based attacks about errors[4] or electrical noise virus[5].

In addition to this, there is the lack of complete, interoperable and open-source software capable of simultaneously managing and monitoring consumption without affecting the most crucial aspect of supercomputers: performance. Actually, there are several utilities [2.1] capable of solving a domain of problems, without however having the ability to interact with each other, in a direct way.

The ultimate goal would in fact be to connect the different tools available using a distributed approach, exploiting the potential of the Data Distribution Service (DDS) and the Real-Time Publish-Subscribe (RTPS).

1.1 DDS & RTPS

DDS (Data Distribution Service)[6] and RTPS (Real-Time Publish-Subscribe)[7] constitute two pivotal technologies in the realm of distributed and real-time communications. These technologies play a critical role in enabling efficient and reliable data transmission among interconnected devices and applications, holding particular significance in intricate scenarios such as embedded systems, the Internet of Things, and high-performance applications like HPC.

Specifically, DDS serves as a distributed communication framework that facilitates data exchange among software components distributed across heterogeneous network and allows to define the quality of service policy. On the other hand, RTPS serves as the underlying protocol employed by DDS to realize the publish-subscribe paradigm within real-time networks. RTPS focuses on the dependable delivery of real-time messages, ensuring that data reaches the appropriate recipients in the most efficient manner. This protocol also manages critical aspects such as data flow control and node synchronization.

2 State of the Art

The state of the art can be delineated into two key realms pertinent to the research study. Firstly, a partial overview of the existing landscape in supercomputing power management that aims to reveal noteworthy developments and challenges that lay the groundwork for the pursuit of interoperable solutions. On the other hand, in the DDS middleware part, the State of the Art is established by a widely recognized and extensively utilized implementation, which is ROS2 with its *rmw_dds_common*.

2.1 Power Management Demand in HPC

In the wake of the termination of Moore's Law and the cessation of Dennard scaling, the gradual reduction in semiconductor manufacturing processes has concomitantly yielded a marked augmentation in power density[8]. This factor, coupled with the escalating exigencies of performance enhancements, and the increase of carbon dioxide emissions associated, has forged an imperative to address and improve the power consumption quandary prevalent within High-Performance-Computing.

Various solutions have emerged as a result like: (i) Node-Level Energy Limitation[9] that aims to control the energy consumed by each computing node and it is based on dynamically adjusting the power or frequencies of hardware components such as CPUs, GPUs, and memory, based on the running workloads; (ii) Dynamic Voltage and Frequency Scaling[10] (DVFS) that involves dynamically adjusting the voltages and frequencies of hardware components based on the current workload. For example, when the computation demand is low, components can operate at lower frequencies and reduced voltages, thereby reducing energy consumption; (iii) Predictive Analysis and Machine Learning[11]: The use

of predictive analysis and machine learning can help optimize energy consumption. Predictive algorithms can forecast load spikes and take preventive measures to reduce consumption, while machine learning can optimize energy management strategies based on historical and real-time data; (iiii) Thermal/Power Capping[12] technologies used to keep operating temperatures and energy consumption in check. When a node or component reaches a certain temperature or power threshold, the system applies automatic restrictions, such as lowering clock frequency or limiting the use of specific components.
All these methods need to be coordinated and orchestrated by different actors such as system manager, node manager, job manager, task manager and monitors.

Runtimes & Interoperability Within this context, certain solutions have emerged, each with their own strengths and shortcomings. These power-stack utilities have been proposed targeting specific optimizations such as:

- Countdown[13]: an open-source runtime library that is able to identify and automatically reduce the power consumption of the computing elements during communication and synchronization of MPI-based applications;
- EAR[14]: an open-source software that provides monitoring, as well as job accounting focused on power and application performance;
- OAR[15]: versatile resource and task manager, designed to be flexible and distributed;
- Examon[16]: a lightweight monitoring framework for supporting accurate monitoring of power/energy/thermal and architectural parameters in distributed and large-scale high-performance computing installations.

Also some of the most well-known software include *Variorum* (LLNL), *GEOPM* (Intel) [17], and *HDEEM* (Atos)[18], to name a few. All these solutions represent attempts to tackle power management challenges, yet they display differing levels of compatibility. This lack highlights the urgent need for a comprehensive interoperability framework. These tools, though efficacious for specific use cases, often fail to cohesively integrate and cooperate due to varying interfaces and implementations.

2.2 DDS & ROS

Ros Middleware interface[19], developed within the context of the Robot Operating System (ROS), stands as a pinnacle of innovation in the realm of creating middleware for Distributed Data Systems. The middleware leverages advanced data serialization techniques, optimized for real-time performance and efficient data exchange. Its incorporation of Quality of Service (QoS) parameters and support for both publish-subscribe and request-response communication patterns underscores its sophistication, making it a state-of-the-art choice for building DDS middleware in distributed environments.

In fact ROS capitalizes on the Data Distribution Service as a foundational communication middleware protocol[20], which plays a pivotal role in enabling seamless and efficient data exchange and interaction between the myriad components that constitute a robotic ecosystem.

The structure of the middleware-ROS, exemplified by *rmw_dds_common*[20], combined with various middleware specific to each DDS service provider, also allows for easy switching of the DDS implementation using a simple variable change, invoked before launching the ROS-node, among the many available options (FastDDS, CycloneDDS, ConnextDDS, etc.). This approach permits choosing the most suitable DDS implementation for specific use case. For instance, if we intend to use FastDDS (most suitable one for real-time communications) as the primary DDS service, we can set *rmw_implementation=fastdds*, and ROS will be able to abstract all of its functions.

3 Project's Description

The central theme of my PhD project will primarily analyze the performance of various DDS implementations, along with different configurations and Quality of Services, in a quest to ascertain the most suitable approach. Once the optimal solution becomes apparent, the focus will shift towards leveraging of this implementation to enable interoperability among the different actors involved in the large scale power management problem. An intriguing approach will be to employ the state-of-the-art framework proposed by ROS2, along with its *Ros-Middleware*.

A second pivotal aspect of the project will encompass the actual implementation of the previously developed middleware across all components constituting the HPC power-stack. This implementation will shed light on an **"interoperable runtime for distributed power management of large-scale HPC systems based on DDS"**, facilitating the seamless management of power on a significant scale within a distributed environment. This endeavor aims to solidify the interconnection between various actors, enabling dynamic and collaborative resource management within the realm of high-performance systems.

Moreover, the final phase of the project involves studying and implementing a security layer for the previously developed implementation (both DDS and power-stack utilities). This step ensures the safeguarding of the intricate power management interactions and data exchanges, reinforcing the overall reliability and resilience of the system.

4 Expected Results

This project is expected to achieve three main contributions:

- The cost-performance analysis of different DDS implementations, within different QoS and configurations aimed at finding the best solutions that can be used to interconnect the various tools;

- The design of an open-source middleware able to satisfy the different needs of the different actors present in the HPC domain;
- The implementation and interconnection of middleware created inside each different tools in order to obtain an interoperable and complete power manager;

5 Proposed project timeline

- Year 1:
  - Literature overview on the implementation and state of the art of modern HPC architectures, and open-source software already available.
  - Creation of a simulation environment, to evaluate DDS implementation.
  - Modeling and characterization of power management libraries on real HPC system and processors.
- Year 2:
  - Analysis and comparison of the identified middleware structure designs.
  - Prototype implementation of an holistic power management solution.
- Year 3:
  - Explore the security side of the tools used and the middleware carried out.
  - Adapt the project to a broad range of scenarios and possible implementations.

6 Outline of the proposed findings assessment criteria

The criteria to asses the proposed findings will be:

- An open-source middleware capable of using different implementation of DDS able to allow the interoperability of the entire power stack;
- An exhaustive analysis of different DDS configurations and different implementations relevant to the power management;
- The development of a simulation environment able to provide reliable results;
- The possibility to integrate the complete power manager into a real supercomputer;

References

[1] Edward Curry et al. "Developing a Sustainable IT Capability: Lessons From Intel's Journey". In: *MIS Q. Executive* 11 (2012), p. 3. URL: https://api.semanticscholar.org/CorpusID:1690698.

[2] Katja Biedenkopf, Ellen Vanderschueren, and Franziska Petri. "Riding the Green Wave? Green Electoral Success and the European Green Deal". In: *The EU Political System After the 2019 European Elections*. Ed. by Olivier Costa and Steven Van Hecke. Cham: Springer International Publishing, 2023. ISBN: 978-3-031-12338-2. DOI: 10.1007/978-3-031-12338-2_17. URL: https://doi.org/10.1007/978-3-031-12338-2_17.

[3] The Hardvard Gazette. *Smaller, faster, greener*. 2021. URL: https://news.harvard.edu/gazette/story/2021/03/what-will-green-computing-look-like-in-the-future/.

[4] Pengfei Qiu et al. "VoltJockey: Breaching TrustZone by Software-Controlled Voltage Manipulation over Multi-Core Frequencies". In: CCS '19. London, United Kingdom: Association for Computing Machinery, 2019, pp. 195–209. ISBN: 9781450367479. DOI: 10.1145/3319535.3354201. URL: https://doi.org/10.1145/3319535.3354201.

[5] Vasileios Tenentes et al. "Run-time Detection and Mitigation of Power-Noise Viruses". In: *2019 IEEE 25th International Symposium on On-Line Testing and Robust System Design (IOLTS)*. 2019, pp. 275–280. DOI: 10.1109/IOLTS.2019.8854375.

[6] Object Management Group. *Data Distribution Service*. 2004. URL: https://www.omg.org/spec/DDS/1.0.

[7] Object Management Group. *DDS Interoperability Wire Protocol*. 2008. URL: https://www.omg.org/spec/DDSI-RTPS/2.0.

[8] Mark Bohr. "A 30 Year Retrospective on Dennard's MOSFET Scaling Paper". In: *IEEE Solid-State Circuits Society Newsletter* 12.1 (2007), pp. 11–13. DOI: 10.1109/N-SSC.2007.4785534.

[9] Sherin M A et al. "Node level Power Profiling and Thermal Management in HPC system". In: *2016 2nd International Conference on Green High Performance Computing (ICGHPC)*. 2016, pp. 1–6. DOI: 10.1109/ICGHPC.2016.7508064.

[10] Zhiquan Lai et al. "Latency-aware DVFS for efficient power state transitions on many-core architectures". In: *The Journal of Supercomputing* 71 (2015), pp. 2720–2747.

[11] Amir Mosavi and Abdullah Bahmani. *Energy Consumption Prediction Using Machine Learning*. Mar. 2019.

[12] Intel. *Reducing Energy Consumption and Costs*. https://www.intel.com/content/dam/www/public/us/en/documents/case-studies/reducing-energy-consumption-and-costs.pdf. 2018.

[13] Daniele Cesarini et al. *COUNTDOWN: a Run-time Library for Performance-Neutral Energy Saving in MPI Applications*. 2019. arXiv: 1806.07258 [cs.DC].

[14] Julita Corbalan and Luigi Brochard. *EAR: Energy Management Framework for High-Performance Computing*. Barcelona Supercomputing Center (BSC-CNS). URL: https://www.bsc.es/research-and-development/software-and-apps/software-list/ear.

[15] OAR Team. *OAR: Resource and Task Manager for Clusters and other Computing Infrastructures*. GitHub repository. 2011. URL: https://github.com/oar-team/oar.

[16] *Examon HPC Monitoring*. URL: https://github.com/EEESlab/examon (visited on 05/16/2023).

[17] INTEL. *GEOPM*. 2017. URL: https://sc17.supercomputing.org/SC17%20Archive/tech_poster/poster_files/post176s2-file3.pd.

[18] Daniel Hackenberg et al. "HDEEM: High Definition Energy Efficiency Monitoring". In: *2014 Energy Efficient Supercomputing Workshop*. 2014, pp. 1–10. DOI: 10.1109/E2SC.2014.13.

[19] Dirk Thomas. *ROS 2 middleware interface*. https://design.ros2.org/articles/ros_middleware_interface.html. 2017.

[20] Open Robotics. *ROS 2 Documentation: Iron documentation*. https://docs.ros.org/en/iron/index.html. 2023.