CONNECTING THE DOTS

ISC High Performance

Monitoring Energy Consumption of Workloads on HPC Vega

6th ISC HPC International Workshop on
"Monitoring & Operational Data Analytics" (MODA25)

Teo Prica[1,2], Aleš Zamuda[2]

13 June 2025

[1]IZUM - Institute of Information Science , [2]UM - University of Maribor, Faculty of Electrical Engineering and Computer Science, Institute of Computer Science

Agenda

- Introduction
- Related Work and Methods
- Methodology
- Experimental Results
- Conclusion and Future Work
- References

VEGA HPC

Introduction

Introduction

- Insight of **first launched EuroHPC JU system** (setup, adjustments and mechanisms).
- Management and monitoring of energy consumption within **HPC AI workloads**.
- Interest from various **scientific domains**, **industry**, and **SMEs**.
- Necessary to introduce **effective** and **essential mechanisms** for **energy sustainability**.
- Differences in systems and architectures lead to different end solutions and policies[1].
- **A challenge for users**, lack of knowledge, no unified framework.
- **EuroHPC Federation Platform** project for integrating current and future systems[2].
- System administrators & Application Support Teams (AST) provide high-level support across multidisciplinary areas and **best practices and guidelines are available**.

VEGA

[1] Roman Iakymchuk et al. *Best Practice Guide – Harvesting energy consumption on European HPC systems: Sharing Experience from the CEEC project*. 2024. DOI: 10.5281/zenodo.13306639.

[2] Henrik Nortamo. "EuroHPC FP: A Federated Platform for HPC Infrastructure in Europe". Presented at FOSDEM 2025. 2025. URL: https://fosdem.org/.

Related Work and Methods

Related Work and Methods

Research on energy efficient HPC involves mechanisms with focus on pillars framework[3].

- **Infrastructure**,
- **hardware**,
- **software**,
- **optimization**.

Despite advances there are still challenges to consider:

- Scalability issues,
- infrastructure limitations and bottlenecks,
- and efficiency trade-offs.

VEGA HPC

[3] Torsten Wilde, Axel Auweter, and Hayk Shoukourian. "The 4 Pillar Framework for energy efficient HPC data centers". In: *Computer Science - Research and Development* 29 (2013).

Related Work and Methods

- Power Capping and Constraint
- Baseboard Management Controller (BMC)
- Intelligent Platform Management Interface (IPMI)
- Running Average Power Limit (RAPL)
- System Management Interface (SMI)
- NVIDIA Data Center GPU Manager (DCGM)
- Frequency Throttling
- Checkpoints and Restart (CR)
- Software in HPC
- Slurm Workload Manager
- Monitoring and Metrics Visualization
- Current Setup of HPC Vega

VEGA HPC

Power Capping and Constraint

Controlled and limited operation of the system environment (HW/SW)[4].

- Monitoring and extending consumption according to requirements,
- effective or preventive constraints (hardware failures),
- reducing high operating costs in higher peaks,
- lower priority tasks,
- partially turning off the compute nodes,
- distributed loads,
- in the last resort, jobs may still be killed.

VEGA HPC

[4] Alberto Cabrera et al. "Energy efficient power cap configurations through Pareto front analysis and machine learning categorization". In: *Cluster Computing* 27 (2023), pp. 3433–3449. ISSN: 1573-7543.

Baseboard Management Controller (BMC)

BMC is a microcontroller integrated into the motherboard of servers, reachable via **Out-of-Band** (**OOB**) network and enables:

- Remote power management,
- monitoring,
- troubleshooting,
- system administration of server systems,
- independent of the OS.

VEGA HPC

IPMI and RAPL

Intelligent Platform Management Interface (IPMI) is a set of standard specifications and components.

- Commonly used utility is **ipmitool**,
- intended for system administration via OOB network.
- Controlling computer subsystems **BIOS**, **CPU**, and **OS**,
- monitoring, logging, recovery, and retrieving data from sensors (SDR, energy, etc.).

Running Average Power Limit (RAPL) is an interface feature introduced by Intel's Sandy Bridge architecture,

- Provides access to real-time measurements of hardware counters based on CPU sockets and DRAM,
- suitable for measurements of power consumption of various HW setups.[5]

VEGA HPC

[5] Howard David and et al. "RAPL: Memory power estimation and capping". In: *2010 ACM International Symposium on Low-Power Electronics and Design*. 2010, pp. 189–194. DOI: 10.1145/1840845.1840883.

NVIDIA

- **SMI** is a command (nvidia-smi) utility used for managing and monitoring NVIDIA devices, based on NVIDIA Management Library (NVML) C-based API.
- **DCGM** is managing and monitoring for GPUs and allows reducing the power consumption of GPUs clock speed in DCs.
- **NVIDIA Multi-Instance GPU (MIG)** provides up to seven independent slices per GPU (smaller workloads[6].
- Generic Resources (GRES) such as NVML are available within an additional configuration file (`gres.conf`)[7].

VEGA

[6] Tirth Vamja et al. *On the Partitioning of GPU Power among Multi-Instances*. 2025. DOI: 10.48550/arXiv.2501.17752. arXiv: 2501.17752 [cs.DC].

[7] Morris A. Jette and et al. "SLURM: Simple Linux Utility for Resource Management". In: *Proceedings of the 9th International Workshop on Job Scheduling Strategies for Parallel Processing (JSSPP)*. 2002. DOI: 10.1007/10968987_3.

Frequency Throttling

Adjustment of the workload **frequency** and **performance**, which reduces the number of instructions inside the processor, to save consumed energy while not affecting or compromising the performance.

- Is possible within a heterogeneous system (CPU/GPU),
- adjust the core or memory frequency,
- solutions have been developed to automatically adjust frequencies per workload,
- dynamic adjustment without affecting the performance (CPU and GPU),
- newer processors enable performance autonomous scalability on OS level[8].

[8] Kai Ma et al. "Energy conservation for GPU–CPU architectures with dynamic workload division and frequency scaling". In: *Sustainable Computing: Informatics and Systems* (2016), pp. 21–33. ISSN: 2210-5379.

Checkpoints and Restart (CR)

CR mechanisms save states of jobs as checkpoints that are restored upon restart[9].

- Adopted for fault tolerance in HPC workloads,
- several checkpoints could be established per workload,
- continuing the execution after interruption,
- used for power management to reduce cost,
- offloading a larger workload,
- interconnect errors caused by network congestion,
- various CR implementations i.e DMTCP, BLCR, and CRIU.

VEGA HPC

[9] Ifeanyi P. Egwutuoha and et al. "A survey of fault tolerance mechanisms and Checkpoint/Restart implementations for High performance computing Systems". In: *The Journal of Supercomputing* 65 (2013). ISSN: 1573-0484. DOI: 10.1007/s11227-013-0884-0.

Software in HPC

The various application domains resulting in different workloads, preperation and requirements.

- different package managers
- custom-builds from source,
- **containerization** is increasingly popular due to various beneficial advantages such as reproducibility, portability, consistent environment, scalability, and isolation[10].
- various repositories could be accessible via **CernVM-FS**[11],
- tools with power management capabilities i.e. APM, AMD ROCm, **LIKWID**, **Perf**, **PAPI**, EAR, LLview, and **MERIC**[12].
- importance of efficient code, code optimization, and performance analysis!

[10] Barbara Krašovec and Teo Prica. "Secure Usage of Containers in the HPC Environment". In: *Nordic e-Infrastructure Tomorrow*. Springer Nature, 2025, pp. 96–112. ISBN: 978-3-031-86240-3.

[11] Jakob Blomer and et al. "New directions in the CernVM file system". In: *Journal of Physics: Conference Series* 898 (2017). DOI: 10.1088/1742-6596/898/6/062031.

[12] Roman Iakymchuk et al. *Best Practice Guide – Harvesting energy consumption on European HPC systems: Sharing Experience from the CEEC project*. 2024. DOI: 10.5281/zenodo.13306639.

Slurm Workload Manager

Slurm[13] manages resources, job scheduling, measuring, and retrieving data from its database.

- Wide range of plugins, tailored solutions,
- power management is integrated through a plugin,
- either to prevent overload (prevent HW issues) or increase the utilization or consumption,
- commands **suspend**, **hold** a running job, or **resume** to resume a suspended job,
- accounting is done through a **Slurm Energy Accounting Plugin**,
- `GPU management library`, **IPMI**, `PM_Counters`, `RAPL`, and `XCC`.

VEGA HPC

[13] Morris A. Jette and et al. "SLURM: Simple Linux Utility for Resource Management". In: *Proceedings of the 9th International Workshop on Job Scheduling Strategies for Parallel Processing (JSSPP)*. 2002. DOI: 10.1007/10968987_3.

Monitoring and Metrics Visualization

In the facilities such as HPC, monitoring is done via **metrics** which are **collected**, **stored**, and **visualized**:

- Supply infrastructure,
- system-wide overview of the cluster,
- specific equipment, devices, hardware, and software being monitored,
- electricity consumption, which fluctuates between **700 kW** and **1 MW**.

VEGA HPC

Grafana, Prometheus, and Victoria Metrics

- **Grafana** is an open-source solution for the visualization of metrics obtained from various databases stored into Time Series Database, queries, visualized and monitored using prepared dashboards.
- **Prometheus** is open-source used to collect various metrics, monitoring and alerting and can be added as a source within Grafana.
- **Victoria Metrics** is an open-source high performance time series database with capabilities to monitor and control the system, added as an additional source to Grafana.
- **Node exporter** is a tool used to monitor, collect and export various metrics from a client.

VEGA
HPC

Current Setup of HPC Vega

- Nodes are divided into five partitions **cpu**, **gpu**, **dev**, **largemem**, and **longcpu** (1028).

- Energy consumption is retrieved through **IPMI** sensors,
- entries are assigned to parameters within **Slurm** configuration, and
- deployment, configuration distribution cluster-wide, is managed through **Ansible**.
- **Node Health Check (NHC)** on nodes.

- Two storages, the **Lustre** parallel file system (1 PB) for scratch and a large capacity storage based on **Ceph** (19 PB) for home, project directories, data storage, and beyond.

Current Setup of HPC Vega

- Power constraint is managed via various physical sensors in a DC, readings are retrieved from **heating, ventilation, air conditioning (HVAC)**.
- **Supervisory Control and Data Acquisition (SCADA)** and **Data Center Infrastructure Management (DCIM)** based system.
- An alert is triggered after deviations of events from given thresholds.
- Triggers upon fail (picked by the management system),
- A signal is sent to Slurm through **snmptrap**, to reduce power consumption via Slurm abilities with **suspend** command and an unsuspend via **resume** command.
- Tailored implementation executed synchronously rack-by-rack, maintained by the vendor.
- Reduction or increase of consumed power and sustaining the lifetime of the components.
- **BEO is currently disabled in HPC Vega!**[14]

VEGA HPC

[14] Roman Iakymchuk et al. *Best Practice Guide – Harvesting energy consumption on European HPC systems: Sharing Experience from the CEEC project*. 2024. DOI: 10.5281/zenodo.13306639.

Current Setup of HPC Vega

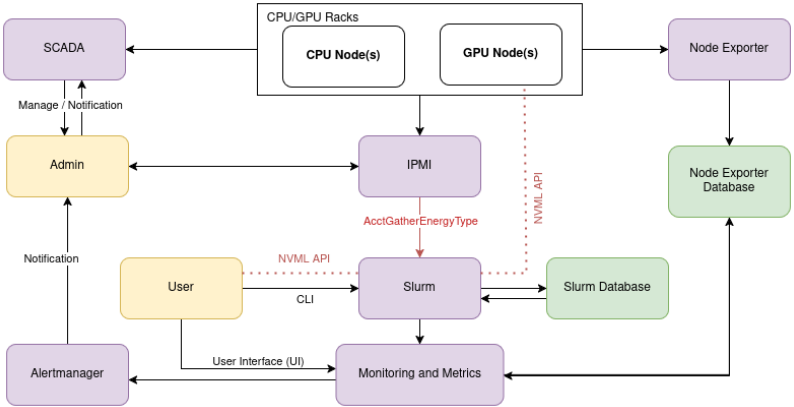

Figure: Overview of measurement, monitoring, and metrics.

Grafana, Prometheus, and Victoria Metrics

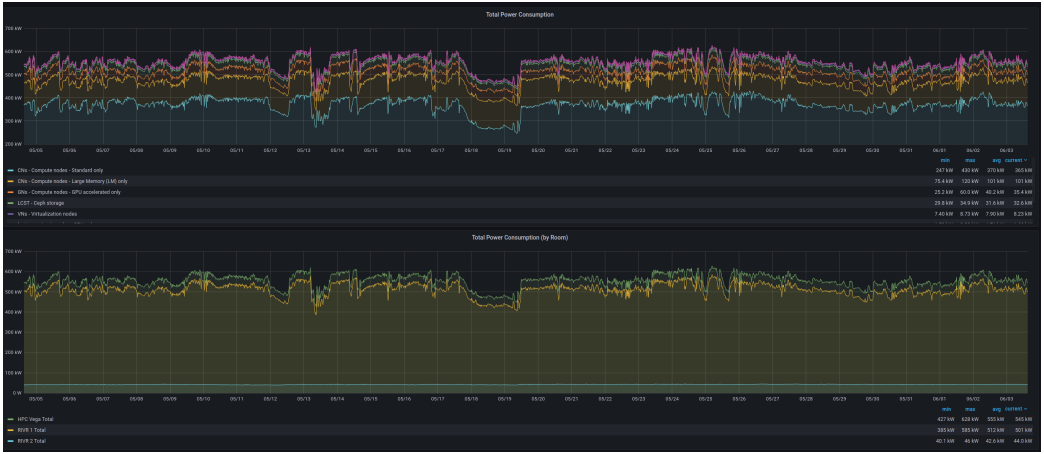

Figure: Grafana dashboard of total power consumption by equipment.

Differential-Evolution-Based Hyperparameter Otimization

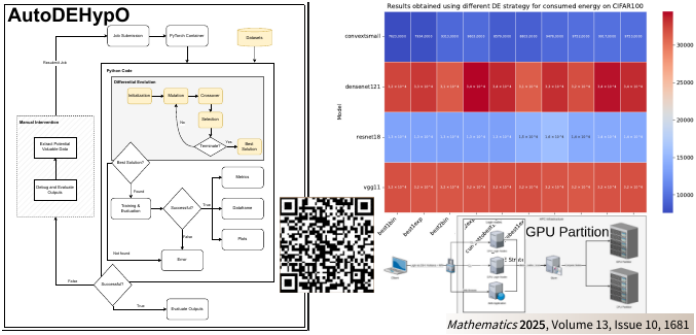

Figure: High-Performance Deployment Operational Data Analytics of Pre-Trained Multi-Label Classification Architectures with Differential-Evolution-Based Hyperparameter Optimization (AutoDEHypO)[15]

[15] Teo Prica and Aleš Zamuda. "High-Performance Deployment Operational Data Analytics of Pre-Trained Multi-Label Classification Architectures with Differential-Evolution-Based Hyperparameter Optimization (AutoDEHypO)". In: *Mathematics* 13.10 (2025). ISSN: 2227-7390. DOI: 10.3390/math13101681. URL: https://www.mdpi.com/2227-7390/13/10/1681.

Methodology

Slurm Account Gather Configuration

Challenges had to be tackled when enabling the power measurement of consumption via Slurm. Frequencies were adjusted using **AcctGatherNodeFreq** parameter, which was set from **25** to **60** and **JobAcctGatherFrequency** parameter, from **10** to **60**.

```
AcctGatherEnergyType=acct_gather_energy/ipmi
AcctGatherNodeFreq=60
JobAcctGatherFrequency=60
JobAcctGatherType=jobacct_gather/cgroup
```

CPU and GPU are mapped to the IPMI sensor at different positions:

```
EnergyIPMIFrequency=60
EnergyIPMICalcAdjustment=yes
EnergyIPMITimeout=60
EnergyIPMIPowerSensors=Node=value     # CPU = 13 ; GPU = 20
```

VEGA HPC

Generic Resources Configuration

The modification was also made to the **GRES** configuration file.
The line commented below represents the state before switching to **NVML**:

```
AutoDetect=nvml
Name=gpu Type=ampere File=/dev/nvidia[0-3]

#NodeName=gn[01-60] Name=gpu Type=ampere File=/dev/nvidia[0-3]
```

VEGA HPC

Consumed Resources

To retrieve consumed resources of finished jobs from Slurm via **sacct** command:

```
sacct -u $USER -X -j $jobids --format=JobID,State,ConsumedEnergy, \
     Elapsed,TresUsageOutAve%40,NNodes --parsable2 --delimiter=";"
```

Similarly, to retrieve consumed resources of finished Slurm jobs with **sacct** command and parameter **TRESUsageInAve** with enabled **NVIDIA NVML**:

```
sacct -j $%jobid$ -u $%USER$ -Pno TRESUsageInAve
```

Monitoring of the GPU utilization, memory, and power consumption via **nvidia-smi** command:

```
nvidia-smi -l 1 --query-gpu=index,utilization.gpu, \
   utilization.memory,power.draw --format=csv,nounits > output &
```

VEGA HPC

Experimental Results

Obtained and Measured Results

Retrieved data from `ipmi-sensors` command was executed on GPU compute node.

ID	Name	Type	Reading	Units	Event
10	Watchdog	Watchdog 2	N/A	N/A	'OK'
11	SEL	Event Logging Disabled	N/A	N/A	'OK'
12	CWG_LM5066I_IIN	Current	24.11	A	'OK'
13	LM5066_IIN	Current	4.38	A	'OK'
14	RDST_INST_PWR1	Power Unit	290.00	W	'OK'
15	RDST_INST_PWR2	Power Unit	340.00	W	'OK'
16	RDST_INST_PWR3	Power Unit	260.00	W	'OK'
17	RDST_INST_PWR4	Power Unit	260.00	W	'OK'
18	LM5066_PIN	Power Unit	240.00	W	'OK'
19	AVG_PWR	Power Unit	240.00	W	'OK'
20	CWG_LM5066I_PIN	Power Unit	1335.00	W	'OK'
21	CWG_LM5066I_VIN	Voltage	54.98	V	'OK'
22	LM5066_VIN	Voltage	54.90	V	'OK'
23	CPU0_Status	Processor	N/A	N/A	'Processor Presence detected'
24	CPU1_Status	Processor	N/A	N/A	'Processor Presence detected'
25	CPU0_TEMP	Temperature	52.00	C	'OK'
26	CPU1_TEMP	Temperature	57.00	C	'OK'
27	DIMMG0_TEMP	Temperature	49.00	C	'OK'
28	DIMMG1_TEMP	Temperature	49.00	C	'OK'
29	DIMMG2_TEMP	Temperature	48.00	C	'OK'
30	DIMMG3_TEMP	Temperature	47.00	C	'OK'
31	CPU0_DTS	Temperature	48.00	C	'OK'
32	CPU1_DTS	Temperature	43.00	C	'OK'
33	GPU1_TEMP	Temperature	68.00	C	'OK'
34	GPU2_TEMP	Temperature	68.00	C	'OK'
35	GPU3_TEMP	Temperature	68.00	C	'OK'
36	GPU4_TEMP	Temperature	62.00	C	'OK'
37	CWG_LM5066I_TEMP	Temperature	51.00	C	'OK'
38	LM5066_TEMP	Temperature	51.00	C	'OK'
39	MB_TEMP1	Temperature	50.00	C	'OK'
40	MB_TEMP2	Temperature	53.00	C	'OK'
41	CWG_TEMP1	Temperature	54.00	C	'OK'

Power Managment Signal

Power management signal is received within Slurm per rack (GPU/CPU):

```
NTR:_received_power_management_signal_at_Thu_Jun__5_09:25:54_CEST cn[0193-0199,0201-0202,0204-0212,0214-0228,0230-0231,0234-0245,0247-0253,0256-0257,0259-0265,0268-0282,0286-0288]
NTR:_received_power_management_signal_at_Thu_Jun__5_09:30:58_CEST cn[0289-0296,0298-0300,0302-0309,0313-0314,0316,0318-0321,0326-0330,0333-0336,0338-0356,0360-0363,0366-0372,0374-0384]
NHC: Check Secu-hotpatch returned 1                               cn[0387,0399,0409,0419]
NTR:_received_power_management_signal_at_Thu_Jun__5_09:26:19_CEST cn[0577-0594,0596,0598-0609,0611-0623,0625-0627,0631-0634,0636-0645,0647-0660,0662,0665,0667-0668,0670-0672]
NTR:_received_power_management_signal_at_Thu_Jun__5_09:26:19_CEST cn0635
NTR:_received_power_management_signal_at_Thu_Jun__5_09:31:21_CEST cn[0673-0680,0682-0687,0689,0691-0698,0700-0711,0714-0715,0717,0719-0729,0734-0736,0740,0742-0764,0766-0768]
NTR:_received_power_management_signal_at_Thu_Jun__5_09:31:21_CEST cn0716
```

Figure: Received power management signal for power constraint CPU racks.

```
== Availability ==
Total drained 528 (518 cn nodes ; 10 gn nodes)
Total available 492 (442 cn nodes ; 50 gn nodes)
Overall 48.23% ; CN 46.04% ; GN 83.33%
```

Figure: Availability of drained and availiable compute nodes (319 power constrained).

```
== Availability ==
Total drained 209 (198 cn nodes ; 11 gn nodes)
Total available 811 (762 cn nodes ; 49 gn nodes)
Overall 79.50% ; CN 79.37% ; GN 81.66%
```

Figure: Availability of drained and availiable compute nodes (0 power constrained).

VEGA HPC

NVML

To retrieve consumed resources of finished jobs from Slurm with **disabled** NVML:

```
cpu=10:00,energy=47340,fs/disk=158664, \
    mem=1136584K,pages=0,vmem=1179592K
```

To retrieve consumed resources of finished jobs from Slurm with **enabled** NVML:

```
cpu=10:00,energy=74032,fs/disk=158791,gres/gpumem=19882M,\
      gres/gpuutil=100,mem=1136576K,pages=0,vmem=1146816K
```

VEGA HPC

Obtained and Measured Results

- **TensorFlow**, **PyTorch**, and **NVIDIA NeMo** frameworks were used.
- Official and optimized **Singularity** containers[16].
- Containers built from the public **NVIDIA NGC catalog**.
- Evaluation benchmarks **GLUE**, **SuperGLUE** and **wikitext-2**.
- Executed on single node with up to four GPUs per each workload.
- Limitation of available amount of memory per GPU (20 epochs, batch size 32).
- Models and datasets are available within the **Transformers** and **Datasets** package.
- Results obtained from **Slurm** and **IPMI**.

VEGA HPC

[16] Barbara Krašovec and Teo Prica. "Secure Usage of Containers in the HPC Environment". In: *Nordic e-Infrastructure Tomorrow*. Springer Nature, 2025, pp. 96–112. ISBN: 978-3-031-86240-3.

Obtained and Measured Results

Table: Obtained and measured results from HPC AI workloads.

Framework	Model	Dataset	Nodes / GPUs	Counter	Elapsed Time (M)	Consumed Energy (J)
TensorFlow	`distilbert-base-uncased`	`glue (mrpc)`	1/1	IPMI	00:36:04	360.48K
TensorFlow	`distilbert-base-uncased`	`glue (mrpc)`	1/4	IPMI	00:17:28	284.36K
PyTorch	`gpt2-medium`	`wikitext-2`	1/1	IPMI	00:40:45	1.63M
PyTorch	`gpt2-medium`	`wikitext-2`	1/4	IPMI	00:25:11	1.43M
NVIDIA NeMo	`distilbert-base-uncased`	`super_glue (rte)`	1/1	IPMI	00:13:04	**18.45? ***
NVIDIA NeMo	`distilbert-base-uncased`	`super_glue (rte)`	1/1	IPMI	00:12:50	226.84K
NVIDIA NeMo	`distilbert-base-uncased`	`super_glue (rte)`	1/4	IPMI	00:10:52	123.59K

* Incorrect reporting on consumed energy was detected.

Obtained results are not directly comparable due to the different models and datasets used, the results demonstrate a successful analytical data monitoring on HPC Vega.

Conclusion and Future Work

Conclusion

We presented management and monitoring of the consumed energy of **HPC Vega** workloads.

- Initially, power management was configured through **BEO** and **RAPL** (both disabled).
- A cluster-specific solution is used to manage power constraints.
- Resource consumption is recorded and logged through Slurm, IPMI, and NVML (GPUs).

This setup gained the capability to monitor and track resources and energy consumption of workloads.

VEGA HPC

Future Work

Activities towards emphasizing energy efficiency and sustainability uncovered several possibilities for further evaluation of these essential mechanisms.

- Transition from **BEO** to **Smart Energy Monitors (SEMS)**.
- **RAPL** (and MERIC) assessment is already in progress.
- A calibration with a professional power meter to cross-check and validate the data.
- Colledted data will be made available for users through a **portal**.

VEGA HPC

Acknowledgement

Mentor, assoc. prof. dr. Aleš Zamuda.

The authors acknowledge the **EuroHPC JU**, **HPC-RIVR** *and* **SLING** *consortium for providing allocation of computing resources on the national share of EuroHPC Vega within Development project (S24R08-01), which is hosted at* **Institute of Information Science (IZUM)**. *This research was conducted within* **Individual Research Work 2 (55D007)** *unit of doctoral programme Computer Science and Informatics at University of Maribor.*
The tuition for study enrolment is financed by **IZUM** *(17-2375-2024/01-ab). Authors also acknowledge the project* **DAPHNE** *funded by the European Union's Horizon 2020 research and innovation program under grant agreement No 957407.*

Questions?

The acquisition and operation of HPC Vega is funded jointly by the EuroHPC Joint Undertaking, through the European Union's Connecting Europe Facility and the Horizon 2020 research and innovation program, as well as the Participating State Slovenia. The operation HPC RIVR is partly co-funded by the European Union through the European Regional Development Fund and by the Ministry for Science, Education and Sport of the Republic of Slovenia. The operation is carried out within main priority axis no. 1: »International competitiveness of research, innovation and technological development in line with smart specialization for enhanced competitiveness and greening of the economy«, priority investment 1.1 »Enhancing research and innovation (R&I) infrastructure and capacities to develop R&I excellence, and promoting Centres of competence, in particular those of European interest«, specific objective 1.1.1 »Efficient use of the research infrastructure and development of knowledge/competences to improve national and international collaboration in the knowledge triangle« within the Operational Program for the Implementation of the EU Cohesion Policy 2014-2020.

VEGA VIRTUAL TOUR