Whitepaper
Urgent Computing Policy recommendations

Mirosław Kupczyk, Marcin Lawenda, PSNC

DOI: 10.5281/zenodo.10953116

This project has received funding from the European High-Performance Computing Joint Undertaking (JU) under grant agreement No 955558. The JU receives support from the European Union's Horizon 2020 research and innovation programme and Spain, Germany, France, Italy, Poland, Switzerland, Norway.

Table of Contents

1. Introduction

Urgent computing is a specialized type of computing that prioritizes and expedites the execution of time-critical tasks. Unlike traditional batch computing, where jobs are queued and executed in a sequential order, urgent computing provides immediate access to computational resources for critical applications that require timely analysis, decision-making, or response to real-world events.

The whitepaper is trying to answer the question regarding service's financing and propose how to organize the proper maintenance the Urgent Computing service. The protocol for urgent access to Tier-0 systems, automation rules , and an evaluation of the cost of different scenarios has been undertaken in the scope of the eFlows4HPC D6.6 "Protocol for urgent HPC" [5].

Policy serves also as a guideline for managing user access to the HPC infrastructure in a secure, efficient, and equitable manner, ensuring that computing resources are allocated effectively to support Urgent Computing needs while maintaining system integrity and security. The operational stage of the service must be the subject for further analysis. It requires the preliminary management methodology proposal as the minimal requirement of the prospecting policy.

2. Contractual considerations

So far , there is no official example of running contract of the usage of application in the Urgent Computing scenario in a relatively big computational environment. However the problem, from technical point of view, is quite well recognized and summarised e.g. in [1].

Providing immediate access to users and entities for time-critical decision support in reaction to unforeseen circumstances is not well-aligned with the typical access and usage protocols of HPC facilities. Additionally, these users depend on HPC workflows integrated into data-centric operations and may lack enduring, direct affiliations with supercomputing centers. Scenarios involving rapid insight extraction and decision support within tight timeframes necessitate resource scheduling in response to immediate needs. However, this process is complicated by the significant variability and diversity in the types of tasks to be handled. These encompass collections of jobs, data staging to facilitate workflows, as well as interactions with external services or facilities beyond the HPC systems or centers.

Facilitating model-driven simulations and real-time engagement with decision-makers demands interactive access to HPC assets within dynamic workflows handling extensive datasets. The eFlows4HPC Workflow middleware fulfil this requirement. Consequently, this presents both technical and organizational hurdles. As a pragmatic approach, we could contemplate the maintenance of traditional HPC clusters tailored for batch-style utilization of computational and storage resources, alongside clusters specifically configured for dynamic workflows and interactive usage scenarios. In situations necessitating urgent decision-making, such as coordinating evacuation, the demands on HPC resources can be considerable, rendering static partitioning potentially inefficient.

Numerous technical challenges emerge across various domains, including securing rapid access to HPC and Cloud resources, dynamically allocating and scheduling resources (including extensive utilization of accelerators), coordinating resource managers (encompassing elastic resource allocation – freeing up the available infrastructure, unified authentication and authorization, as well as usage accounting and monitoring), facilitating data-intensive workflows alongside data staging on node-local storage, and enhancing interactivity (such as near real-time simulation steering and resource optimization).

The DestinE initiative [6] outlines various scenarios necessitating immediate responses, notably in the context of disasters such as earthquakes, tsunamis, forest fires, and storms. However, it's crucial to monitor other significant events, including floods, radiological incidents, epidemic outbreaks, as well as astrophysical occurrences such as "space weather," involving particle eruptions (flares) originating from the Sun and potentially causing significant disruptions to satellite operations, telecommunications infrastructure, and even the internet. The project eFlows4HPC addresses the earthquakes and tsunamis scenarios.

It is very significant that all or almost all HPC centres, co-financed by public funds, are able and have the technical capabilities to handle Urgent Computing needs. The question is why there are no reports available about conducting this type of calculations in the production regime. The service is expensive, not only in terms of money, but also in influencing for users' environment. Due to the quality requirements, it is questionable to rely only on the public cloud environment available on the market. On premises HPC infrastructure must be the first choice of the operational service.

3. Access Policy

3.1. Purpose

This policy serves to establishing the guidelines for the fair and equitable use of urgent computing resources allocated within the HPC environment. Urgent computing involves the execution of time critical tasks that require immediate access to HPC resources.

The European Commission can play a vital role in promoting urgent computing and making it available to emergency responders across the European Union. By investing in urgent computing technologies, the Commission can help to ensure that Europe is better prepared to respond to crises. To do so, there is a need to have the financing model for of the Urgent Computing infrastructure providers.

High Performance Computing (HPC) can be used in various civil protection services to enhance disaster preparedness, response, and recovery efforts. Some common HPC applications include:

1. **Natural Disaster Modeling and Prediction:** HPC programs are used to simulate and forecast natural disasters such as hurricanes, earthquakes, floods, tsunami and wildfires. These simulations can help authorities in evacuation planning, resource allocation, and risk assessment.
2. **Emergency Response Coordination:** HPC systems enable real-time data analysis and communication during emergencies, facilitating efficient coordination among first responders, government agencies, and humanitarian organizations.
3. **Infrastructure Resilience Analysis:** HPC tools are employed to assess the resilience of critical infrastructure, such as bridges, dams, and power grids, to various hazards and to develop strategies for mitigating vulnerabilities.

4. **Remote Sensing and Monitoring:** HPC programs support the analysis of satellite imagery, aerial surveys, and sensor data for monitoring environmental conditions, detecting anomalies, and assessing the extent of disaster impacts.
5. **Social Media Analytics:** HPC tools are increasingly utilized to analyse social media data for early warning purposes, situational awareness, and understanding public sentiment during emergencies.

Some of the aforementioned use-cases are the subject to be run in the Urgent Computing mode. There are some technical guidelines, how to run such kind of applications but lack of long-term ideas how to support the sustainability of the selected service.

The goal of this work is to propose the generic recommendations for prospective creation of Policy of coordinating the scenario of Urgent Computing service.

3.2. The stakeholders

Proper organization of the Urgent Computing (UC) process requires understanding and defining the organizational structure by identifying entities, resources and processes that are closely related to each other.

The main entities involved in the organizational process of UC are the ordering institution and the implementing institution.

Commissioning institutions are institutions responsible for monitoring/supervising the phenomena that are the subject of running the simulation. In general, crisis management is the domain of public administration bodies constituting an element of managing national security, which consists of preventing crises, preparing to take control over them through planned actions, reacting in the event of crisis situations, removing their effects and restoring resources and critical infrastructure.

We can distinguish the following crisis management entities:

- crisis management authorities,
- opinion-giving and advisory bodies competent for initiating and coordinating actions undertaken in the field of crisis management,
- crisis management centres maintaining 24-hour readiness to take action.

These bodies are established both at the level of government administration (central or local) and local government. They have an interest in conducting advanced computer simulations by HPC centres to assess potential risks and define mitigation plans.

Computer centres have appropriate resources to conduct complex simulations to evaluate the impact of sudden phenomena on the life, health and property of the population and to propose measures necessary to control them. A selected HPC centre is prepared both organizationally and operationally to launch priority tasks.

The ordering organization signs a contract with the HPC centre committing to maintain computational readiness within a defined time frame. In addition to computing power, space is also provided for storing results and a defined reporting method is defined. Moreover, such preparation is confirmed by periodic checks, including test runs of the application.

3.3. Definition

Urgent computing refers to the execution of time-constrained tasks with a high priority for immediate execution. This may include tasks that require immediate analysis of data, tasks that are needed to respond to real-time events, or tasks that must be completed by a certain deadline.

Urgent computing application refers to the Crisis Management situation. If the application is not run within time constraints, it cannot be referred as “Urgent”. However, in such environment, regular jobs’ run can have valuable scientific results.

Urgent computing is characterized by the following key features:

- Real-time or near-real-time performance: Urgent computing tasks must be completed within tight deadlines, often within seconds or minutes. This requires the use of high-performance computing resources and efficient algorithms to minimize execution times.
- High priority and preemptability: Urgent computing tasks are given priority over non-urgent tasks. This means that urgent jobs can pre-empt lower-priority jobs if necessary to ensure that they are completed on time. The lower-priority jobs may lose their current results!
- During resources’ freeing stage, some jobs may be not properly pre-empted. It concerns mainly distributed applications (MPI, OpenMPI, GPU mix). Due to the
- Resource isolation and dedicated access: Urgent computing often requires dedicated access to exclusive resources to ensure that there is no interference from other tasks. This can be achieved through special partitions, pools, or even dedicated hardware accelerators. Each of these solution is expensive.
- Flexible job scheduling and management: Urgent computing systems must be able to handle unpredictable job arrivals and prioritize tasks effectively. This requires flexible job scheduling mechanisms and real-time monitoring capabilities to dynamically adjust resource allocation.

3.4. Requirements

Technical and organisational statements must be included in the proposal of the contract. According to the eFlows4HPC analysis of the existing landscape, the undertaking must focus on the particular application workflow and the services utilising it. The one contract template cannot be proposed since each service and each application have got different goals concerning different stakeholders with their expectations. However, we gather some common point which must be addressed in the policy.

3.4.1. Technical requirements

All required application software and datasets must be available on the selected system.

Data sets suitable for validation must be pre-installed.

Pre-installing validation datasets, researchers and users can immediately access them without the need of doing additional setup or data transfer which ensure accessibility. It streamlines the workflow and saves the time, where quick access to validated data is essential. Pre-installed datasets ensures, that experiments and analyses can be easily utilised by different users. This is essential for verifying research findings, validating models, and enables the reliability of results. It is also essential for collaboration. Local availability of datasets facilitates collaboration among researchers working on similar projects or using the same datasets. It promotes consistency in data analysis and enables researchers to compare results more effectively.

Large datasets can be time-consuming to transfer to HPC systems, especially if they need to be transferred repeatedly for different experiments or analyses. Local data reduces the need for frequent data transfers, minimizing network congestion and improving overall system efficiency.

Transferring datasets to HPC systems carries inherent risks, including data corruption, loss, or unauthorized access during staging. Pre-installed data reduces these risks by ensuring that the data is securely stored on the system. Also, the data representation including metadata and data format can be considered by local system management service in order to their placement in more efficient way. This is important in premises systems equipped with the multi-layered or hierarchical storage. This early storage decision is important for minimizing data access times, maximizing IOPs and throughputs of physical and logical links in SDS networks.

Validation protocol using pre-installed software and pre-installed data.

There are several possible methodologies available and well described. We can highlight three important steps in order to maintain the service in good health. The standard IQ/OQ/PQ can be the entry point for understanding Installation Qualification (IQ), Operational Qualification (OQ), and Performance Qualification (PQ) derived from medical devices quality management systems ISO 13485:2016. It is important for stakeholders to understand, that the service must operate in highest availability regime to support human rescue and save lives. Other useful methodologies are e.g. PM^2 (EC Project Management Methodology) and ITIL4. ITIL4 refers to 'Service Validation and Testing' as a service management practice.

The ITIL Service Validation enumerates sub-processes and their process objectives. ITIL Test Model Definition specifies in detail how the Release will be tested and quality-assured. In particular, this process defines the testing concept and specific test cases to be used during Service Validation. Release Component Acquisition acquires the components of a Release and submit them to an initial assessment. This process ensures that only components which meet stringent quality criteria are allowed to enter the intensive testing phase. Release Test tests all Release Components and all tools and mechanisms required for deployment, migration and back out. This process ensures that only components which meet stringent quality criteria are deployed into the live production environment. Service Acceptance Testing verifies if all conditions are met for the service to be activated, and to obtain a binding consent from the Urgent Computing contracting party that the new service fulfils the agreed Service Level Requirements.

In accordance with agreed SLA, the validation procedure must concern entire system and application stack. In general, the e.g. daily probs can yield unsatisfactory overall results for the system as a whole. Disrupting of availability connected with the freeing of resources predeceases the running the normal users out of the HPC system. In eFlows4HPC, the system tests were performed when the scheduled maintenance of the system had been known in advance. The outcome of this procedure was the following:

- Pre-emption of all regular jobs was not satisfactory. The resident process' memory stored in the local node memory did not guaranteed the capabilities for urgent computing process being placed on the node. In this case the assessment of the potential free local memory and amount of memory required by urgent computing process is a hard task. With the difficult assumption that we know the exact amount of free memory on each node after removing tasks and also knowing the maximum memory requirements of UC processes, there is still the problem of scheduling nodes to appropriate processes. This is a very interesting, theoretical problem, unfortunately its implementation has been not discovered.
- Checkpoint/restart mechanism cannot guarantee the proper restart of all regular jobs. Most of the HPC jobs are distributed and parallel in type, utilising the inter-process communication, including GPU data exchanging.
- Removing the regular users' jobs from the system demonstrates the acceptable trade-off between loosing the time (jobs must be requeued) and the protection of human health or life.

3.4.2. Activating

While validation of the software can blend in with the regular production environment, the workflow of a real event will inevitably need more. Activating the emergency response typically involves implementing a designated mechanism within the system that can be triggered by authorized personnel. There must be created a specific feature or function within the computing platform that serves as the emergency response trigger. This could be a dedicated web portal, command line interface, or automated script designed to initiate the emergency response process.

3.4.3. Processes

The last organizational element that should be paid attention to ensure appropriate support of Urgent Computing mechanisms is the definition of data delivery and processing procedures.

One of the first stages of processing is to provide access to the input data necessary to perform calculations. This includes both rarely variable data (e.g. topology of the area, layout of streets and buildings in the city) and frequently variable data (e.g. weather conditions, sensor data). The owners of the latter are most often entities commissioning the simulation (crisis management authorities, local governments) and access to them should be specified in the contract.

Conducting simulations should present several possible scenarios for the development of the situation, taking into account the use of specific measures and actions aimed at minimizing the development of a dangerous phenomenon. Therefore, the procedures must provide for the possibility of running simulations in parallel with different data sets (so-called ensemble scenarios).

It is extremely important to deliver results with the required accuracy. This imposes additional requirements on the quality (e.g. granularity) of the input data, the quality of the computational model and ensuring effective data transfer. This last aspect is particularly important when data is provided from external sources such as sensors or cameras. To improve this process, urgent computing is often combined with edge computing to reduce the amount of transported data.

The data obtained in the simulation process should be subjected to evaluation and further processing. Most often, their quantity precludes their manual analysis. To obtain the desired information, it must be further processed using data analysis, often combined with AI.

Decision makers ordering calculations expect the outcomes to be presented in an understandable form that allows them to make the best decisions. Therefore, at the end of the processing process, visualisation tools should be used to properly visualise the obtained results, which, on the one hand, allow for the effective processing of large data sets and, on the other hand, present features that are of key importance from an analytical point of view.

Service Operation Process makes sure that IT services are delivered effectively and efficiently. The Service Operation process includes fulfilling user requests, resolving service failures, fixing problems, as well as carrying out routine operational tasks. Continual Service Improvement – CSI uses methods from quality management in order to learn from past successes and failures. The Continual Service Improvement process aims to continually improve the effectiveness and efficiency of IT processes and services, in line with the concept of continual improvement adopted in ISO 20000.

3.5. Financing model

Over past years many good strategies to run Urgent Computing jobs have been presented. From technical point of view, the organisation of the Urgent Computing workload is known. Several HPC centers demonstrated the piloting configurations for critical jobs. However , the financing model has not been introduced. The general problem with these jobs is, that the running ones spoils the current workload of the system, interfering with the jobs belonging to another users. We assume, that data center willing to host such service is eligible to freely change the machine workload policy – this phenomena concerns public HPC centers, serving the infrastructure for public founded universities, academia and national scientific institutes.

The observation one: The cost of running urgent computing jobs should be shared by the users who submit the jobs and the organization that owns the infrastructure resources. This is because urgent computing jobs use a disproportionate amount of resources, and they can disrupt the work of other users. The 'lost' computing time should be also cleared. Most often, the largest OPEX cost is related to electricity.

The observation two: The data center which implemented the Urgent Computing scenario on their system, must cover the costs not only during critical job executions but throughout the duration of the contract on different levels. Urgent Computing jobs may require additional operational support, such as monitoring and managing resource allocation and providing technical assistance to users. This additional overhead can strain the data center's operations team and increase its operating costs.

Data center may consider offering tiered pricing models for Urgent Computing services, **where Urgent Computing users pay a premium for the ability to pre-empt lower-priority jobs**. This approach can help balance the costs of Urgent Computing across the user base and ensure that the data center can recover its investments while providing valuable services to its all users including restarting.

3.6. Training

All parties and users involved in the processing the crisis situation are required to attend training sessions on urgent computing policies and procedures. This training, which must be elaborated, shall help people to understand the requirements and guidelines for working with urgent computing jobs. Since there are different stakeholders with different roles, each training package must ensure the adequate materials for each of those groups: Advisory bodies, application operators, application and data administrators.

The training must be conducted regularly in order to develop certain patterns of behaviour in a crisis situation. The training component is the most important, apart from the proper functioning of the computing environment.

4. Acknowledgemnt

This project has received funding from the European High-Performance Computing Joint Undertaking (JU) under grant agreement No 955558. The JU receives support from the European Union’s Horizon 2020 research and innovation programme and Spain, Germany, France, Italy, Poland, Switzerland, Norway. The National Centre for Research and Development in Poland partially supported this work by the decision DWM/EuroHPCJU/4/2021.

5. References

1. Manolis Marazakis, et al., HPC for Urgent Decision-Making, White Paper, etp4hpc.eu, 15/02/2022
2. Gordon Gibb, Rupert Nash, Nick Brown, Bianca Prodan, The Technologies Required for Fusing HPC and Real-Time Data to Support Urgent Computing, arXiv:2010.01543v1 [cs.DC] 4 Oct 2020
3. K.K. Yoshimotoa, et. All., Implementations of Urgent Computing on Production HPC Systems, Procedia Computer Science, Volume 9, 2012, Pages 1687-1693
4. M. Agung, Y. Watanabe, H. Weber, R. Egawa and H. Takizawa, "Preemptive Parallel Job Scheduling for Heterogeneous Systems Supporting Urgent Computing," in *IEEE Access*, vol. 9, pp. 17557-17571, 2021, doi: 10.1109/ACCESS.2021.3053162.

5. D6.6 Protocol for urgent HPC, eFlows4HPC, 2024, https://eflows4hpc.eu/deliverables/
6. Destination Earth: Use Cases Analysis, JRC Technical Report JRC122456, 2020. https://publications.jrc.ec.europa.eu/repository/handle/JRC122456