Leveraging e-Infrastructures for Urgent Computing

Siew Hoon Leong\textsuperscript{a,},* Anton Frank\textsuperscript{a}, Dieter Kranzlmüller\textsuperscript{a,b}

\textsuperscript{a}Leibniz Supercomputing Centre, Boltzmannstraße 1, Garching near Munich 85748, Germany
\textsuperscript{b}Ludwig-Maximilians-Universität München, Oettingenstraße 67, Munich 80538, Germany

Abstract

Urgent computing enables responsible authorities to make educated decisions by supporting the computations of simulated predictions of time critical events. Unfortunately, most domains of science cannot afford dedicated resources for their urgent computing problems. As a solution, exploiting existing e-Infrastructures is invaluable for many problems if the wide array of available resources in today’s e-Infrastructures can be utilised. In this paper, we focus on rarely occurring events that are best suited for urgent computations on existing HPC, Grid and Cloud e-Infrastructures. Since e-Infrastructures are meant to serve more than just one community of users, they have inherent characteristics that have to be modified or adapted in order to enable them effectively for urgent computing. We hope to demonstrate that there are many existing and on-going developments that can be leveraged to prepare existing e-Infrastructures for urgent computing.

Keywords: urgent computing; e-Infrastructures; evaluation; use cases; challenges; HPC; Grid; Cloud

1. Introduction

Civil protection, in particular disaster management, is very important to most nations and civilians in the world. When disasters like earthquakes, tsunamis, forest fires, storms, floods, radiological accident, epidemic outbreaks, etc., are expected or have taken place, it is of utmost importance to make timely decisions for managing the affected areas and reduce casualties. Computer simulations can generate information and provide predictions to facilitate this decision making process. When simulations are enabled and supported to complete within a (short) required timeframe to support decision making, this class of computing is referred to as urgent computing. Urgent computing enables responsible authorities to make educated decisions by providing simulated predictions of disasters, the impact and required evacuation zones, etc.

The term “urgent computing” is most known after its introduction in the context of TeraGrid, predecessor project of XSEDE, in 2006 under the SPRUCE (Special Priority and Urgent Computing Environment) science gateway project. The concepts behind urgent computing rely on a mixture of disciplines, services and policies.

* Corresponding author. Tel.: +49-89-35831-8761; fax: +49-89-35831-9700.
E mail address: siew.hoon.leong@lrz.de (S.H. Leong), anton.frank@lrz.de (A. Frank), kranzlmueller@ifi.lmu.de (D. Kranzlmüller)
There are production implementations where researchers and nations perform computations to support time
critical decisions, e.g. the earthquake early warning system, which earthquake-prone Japan invested $600
million to develop. This system enables that when seismic activities occur, the primary wave is used to
compute the destructive secondary wave in approximately two minutes. Early warnings, i.e. a few seconds to a
few minutes, can thus be issued to the civilians. In spite of the usefulness of urgent computing, existing e-
infrastructure providers with leading HPC (High Performance Computing), Grid and Cloud resources cannot
easily support this mode of operation. In this paper, we provide a survey and carry out an evaluation of the
technical challenges faced and the improvements required to support urgent computing on existing e-
infrastructures. This can serve as a feasibility study to see what is lacking and what is already available to
enable urgent computing on existing e-infrastructure.

This paper is organised as follows. In Section 2 and 3, related urgent computing work and the evaluation
methodology are shared respectively. Section 4 outlines the use cases and the challenges. The main evaluation
findings and results are discussed in Section 5. Finally, the conclusion is drawn and shared in Section 6.

2. Related work

The most widely adopted and reliable method for supporting urgent computing appears to be dedicated
resources as in the case for the Japanese earthquake early warning system [2], the German National
Meteorological Service Supercomputer and the North Carolina Forecast System at RENCI for storm surge [4].
Although it is tempting to suggest setting up dedicated resources for each domain specific urgent computations,
it is economically unviable to do so. This is especially so if the computations take place rarely and require huge
amount of resources. Despite the many advantages of using dedicated resources, there are also limitations. In
the aftermaths of Japan’s March 2011 earthquake, the damage to the power grid and rolling blackouts made it
impossible to operate the supercomputers to perform computations that can help to mitigate the damages of the
oil spill and predict the dispersion of the radioactive content in the ocean [5]. The Texas Advanced Computing
Center (TACC) stepped in to donate 500K compute hours to the research institutes. The ad-hoc solution that
helped to solve Japan’s predicament is a consequence of TACC’s readiness to support urgent computing.

TACC is an urgent computing resource in the scope of the SPRUCE project. SPRUCE provides perhaps the
most relevant work and complete solution on several TeraGrid resources to address the urgent computation
infrastructural challenges. It proposes and implements an elevated priority, preemption and/or next-to-run,
solution to handle urgent jobs by using the “Right-of-Way” tokens. It also considers finer details and have
procedures to prepare in advance the required scientific applications on the resources, easy to access web
portals for administrators and users, etc. Preemption is in particular hard to realise in the production mode since
existing policies make it impossible for resource providers to pause or delete running jobs from the system to
make way for urgent computations. To do so is in violation with the providers’ existing policies. This could
explain why there was only one provider in SPRUCE that supported preemption. The next-to-run strategies are
more widely adopted but could delay the urgent computations while waiting for running jobs to finish. There is
a known use case that did not successfully leverage on the SPRUCE implementation. The North Carolina storm
surge forecast started by using the SPRUCE software stack before switching to a dedicated computing system,
RENCI. Another development is that there is currently no resource in SPRUCE that supports preemption [4]. A
fundamental change in policies is thus the best solution to enable centres to effectively support urgent
computing via preemption. Such a change can possibly only be achieved by political influences that are beyond
that of most scientists and researchers. This is particularly the case in nations, which are relatively unaffected
by events that require urgent computing. In spite of the possible hurdles one has to overcome, the potential
benefits are too valuable not to investigate further. In the following sections, the statistics of existing or
proposed use cases for urgent computing are discussed and technical areas that would enable the realisation of
urgent computing on existing e-infrastructures are evaluated.
3. Evaluation methodology

The focus of this paper is on urgent computing use cases that do not have dedicated resources and have to exploit existing e-Infrastructures. Dedicated resource use cases can also benefit from these evaluation results, e.g. to increase their reliability by using e-Infrastructures as backup resources. The term e-Infrastructure refers to “a new way of conducting scientific research by the creation of a new environment for academic and industrial research in which virtual communities have shared access to unique or distributed scientific facilities (including data, instruments, computing and communications), regardless of their type and location in the world.” [6]. Existing HPC, Grid and Cloud infrastructures are the scientific facilities that we would be referring to. These resources are most commonly used by scientists and are typically freely available for research. In Europe, examples of such e-Infrastructures are PRACE (PaRtnership for Advanced Computing in Europe) [7] and EGI (European Grid Infrastructure) [8]. PRACE provides leading HPC resources for European scientists while EGI coordinates and manages a big number of distributed Grid and Cloud resources for various research communities. In the US, XSEDE provides an infrastructure supporting more than 2000 projects [9].

Use cases that have dedicated resources tend to be able to fulfil their urgent computation requirements by customising the resources as per their use cases. As such, the issues faced tend to be more trivial. For use cases that depend on existing e-Infrastructures, the issues encountered are more diverse and scientifically challenging. The needs of these use cases are many times in direct contradiction with the infrastructure setup or policies. Most e-Infrastructure providers are unable to properly support urgent computing in their current state without adjusting their operation model. Sometimes, technical solutions that are feasible cannot be applied as they conflict with the policies, setup, design, etc., of the infrastructures. The delicate balance, between what can be done and what not under the infrastructure restrictions, will be evaluated further on. Since the demand of urgent computing can be a big strain on the e-Infrastructures, the use cases that are considered particularly suitable to run on such infrastructures are narrowed down to rarely occurring use cases.

4. Urgent computing use cases and challenges

There are many scenarios that could benefit from urgent computing on existing e-Infrastructures. Table 1 shows a summary of use cases and challenges, and the solutions that are suggested or implemented. Some use cases and challenges require regular computations and thus have dedicated resources. Others exploit solutions, e.g. elevated priority, algorithmic strategies, decision support systems, etc. to help support urgent computing on existing infrastructures or resources. Clearly, there are a significant number of use cases and challenges that can leverage on existing e-Infrastructures if urgent computing is enabled.

Table 1. Urgent computing use cases and challenges

| Use case                        | Origin                | Regular computation | Solution/Suggested solutions                                      |
|--------------------------------|-----------------------|---------------------|------------------------------------------------------------------|
| Flash floods [10]               | Italy (Genoa)         | No                  | HPC resources                                                    |
| Weather forecast on PLX [11]    | Italy/Germany         | Yes                 | Dedicated resource                                               |
| Earthquake early warning system | Japan                 | Yes                 | Dedicated resources                                              |
| Tracking real-time storms in    | USA                   | No                  | Elevated priority (non-dedicated resources)                      |
| project LEAD [12]               |                       |                     |                                                                  |
| Storm surge computation [4]     | USA                   | Yes                 | Dedicated resource                                               |
| Forest fire propagation [13]    | Spain                 | No                  | Generic algorithm (possible with both dedicated and non-dedicated resources) |
| Using smartphones on cloud [14]| USA                   | No                  | Mobile phone and/or cloud resources                               |
However, the current user policies and strategies adopted by these infrastructure providers imply many challenges. Here, we attempt to identify the common challenges that are critical and/or yet to be discussed or sufficiently looked into.

Table 2. Overview of solutions for each use case and challenge from Table 1

| Use case/Challenge                      | Frequency of Computation     | Access          | Algorithms        | Computing resource | Data resource | Policies/strategies | Setup |
|-----------------------------------------|------------------------------|-----------------|-------------------|--------------------|---------------|---------------------|-------|
| Flash floods [10]                       | Event-driven but daily monitoring | Registered users | X                 | X                  |               |                     |       |
| Weather forecast on PLX [11]            | Daily                        | Registered users | X                 | X                  | X             |                     |       |
| Japan’s earthquake early warning system [2] | Event-driven                | Registered users | X                 | X                  | X             |                     |       |
| Tracking real-time storms in project LEAD [12] | Event-driven                | Registered users | X                 | X                  | X             |                     |       |
| Storm surge computation [4]             | Event-driven but daily monitoring | Registered users | X                 | X                  |               |                     |       |
| Forest fire propagation [13]            | Event-driven                | Not Applicable  | X                 |                   |               |                     |       |
| Using smartphones on cloud [14]        | Event-driven                | All smartphone users | X                 | X                  | X             | X                   |       |
| Implementing computing on HPC system [15] | Event-driven                | Registered users | X                 | X                  | X             |                     |       |
| High-level knowledge based structures for Simulations [16] | Dependent on use case | Registered users | X                 | X                  | X             |                     |       |

The solutions proposed by various researchers or communities are summarised in Table 2. Most use cases and challenges require computing resources and have or plan to have computation resources available. Some solutions revolve around improving the policies/strategies, e.g. batch scheduler, and the setup of the resources to support their use cases. Most assume that the algorithms and data resources are available for their required urgent computation and thus did not propose any specific strategies to customise them for urgent computation purposes. Higher-level solution like a decision support system is also proposed to help facilitate decision making by providing the required data. In fact, all categories are important to ensure that urgent computing can be effectively carried out.

In the following section, algorithmic solutions, accessibility of e-Infrastructures, suitability of different computation resources, use case specific setup, policies/strategies of the infrastructure and the importance of reliability are discussed in more detail. The importance of data resources to support big data in many scientific domains is also very critical and is a big topic that must be looked into but will not be discussed in this paper.
5. Evaluation

Urgent events that happen rarely and/or require only huge computation resources during these rare circumstances, leveraging on existing e-Infrastructures can be more reasonable and economically viable. Since public infrastructures have been purchased or set up for different purposes, challenges have to be overcome before urgent computing can be supported. In order to address the current limitations due to policies and infrastructure requirements, suggestions and on-going development work in various critical areas are discussed and shared in the following sub-sections.

5.1. Algorithmic solutions

The top HPC resources offer petaflops computing power today and are expected to achieve exaflops before 2020. Urgent computing should take this into consideration when selecting algorithms to support their use cases. In addition to scalability, robustness and fault tolerance are also critical. This is especially relevant if the code is using hundreds of thousands of cores. In event of faults, e.g. mechanical failure of a single core, the traditional way of writing periodic checkpoints (restart files) is no longer practical or trivial [17]. A complete restart of a computation, especially from scratch, is not an option for urgent computation. P. Beckman puts it aptly, “Late results are useless” [18]. As such, algorithmic strategies should be considered to mitigate the risk of having to perform a complete rerun or to create time consuming restart files. Fault tolerant algorithms that are able to recover from faults are particularly interesting. Strategies proposed include the combination use of the chaotic relaxation and meshless methods [17], and the sparse grid combination technique [19]. When applied appropriately, these methods could potentially allow a computation to continue without compromising the result when a certain number of tasks/cores, limited by a threshold, fail during runtime.

A fault tolerant parallel library is necessary to support fault tolerant parallel algorithms. There is also effort to enable the implementation of fault tolerance solutions for MPI based applications by a Fault Tolerance working group from the MPI forum [20]. Some proposed implementations are the options to enable a developer to define the error for a failure to allow an MPI application to continue and a quiescence interface to ensure that no communication occurs with a particular communicator when checkpointing is taking place.

In addition to fault tolerant algorithms, there are other classes of algorithms that can be useful. For events like storm, tsunami and flash floods, the data is getting updated as the events evolved. The accuracy of the computation can thus be greatly improved if the new/updated information can be used in the computation. However, restarting the computation with the new/updated data might not be an option in an urgent run due to time constraints. Adaptive algorithms that handle evolving data can thus be of interest. Numerous adaptive algorithms have been proposed for evolving data, e.g. performing only a partial update of the coefficients in a given iteration while using the nonlinear Volterra filter [21] and the use of CVFDT algorithm to perform data mining on time-changing data streams [22]. Such algorithms could potentially be very useful in assisting urgent computations to arrive at more reliable and accurate results and/or for the numerical solution to converge faster.

Another class of algorithms that can be applicable to urgent computing is genetic algorithms. There is always a level of uncertainty in the recorded data, which can contribute to wrong results, e.g. data from seismic stations can be affected by the noises from nearby construction sites. The use of genetic algorithms can help to reduce this uncertainty by allowing the best selection settings for a given prediction quality constraint [13].

As the scientific communities prepared themselves for big data and bigger computing facilities, interesting algorithmic developments are being carried out. Some of the effort is very applicable to urgent computing and can lead to an improvement in the accuracy, robustness, reliability, etc. of our urgent codes. The urgent computing community should thus be on a constant look out for such developments to exploit them further.
5.2. Accessibility of e-Infrastructures

Getting access to an e-Infrastructure typically involves strict and time-costly procedures. This is particularly true for expensive HPC infrastructures, e.g. PRACE. Typically, HPC resources have security policies that new users have to become accustomed to before they can access the resources with ease. One example is the SuperMUC, one of the biggest HPC resources in Europe hosted by the Leibniz Supercomputing Centre (LRZ) in Germany. The firewall on this PRACE machine allows only registered IPs to access it and only incoming traffic is allowed on most ports. Users thus have to ensure that e.g. their codes do not access external data resources during runtime and that data exchanges are initiated from outside this machine. Another PRACE resource, hosted at High Performance Computing Center Stuttgart, does not allow any shared resources to directly access it and has a unique setup for certain tools and services. The French PRACE resource at IDRIS has a similar firewall setup as LRZ and requires security clearance for non-European users. As such, when a user switches among resources within the same e-Infrastructure, some level of familiarisation can be required.

On the other end, public Cloud resources do not face the same access issues as more expensive HPC resources. One can configure the accessibility based on the requirements. Clouds are thus the ideal resources in terms of accessibility if one can afford the cost of using them. In the case of Grid infrastructure like EGI, accessibility is not as strict as the HPC infrastructure but is typically restricted by the Grid middleware, e.g. gLite or Globus, which is installed. When users switch from one e-Infrastructure to another, they have to get accustomed to the new setup. This is not effective especially for rarely occurring use cases.

To effectively support urgent computing on e-Infrastructures, a common set of access policies is recommended. In project like PRACE and EGI, this is already partially achieved. However, many sites specific restrictions still apply. To overcome this, policies have to be changed and this can typically be possible via political decisions, which can be influenced by societal needs. In fact, political changes can happen rapidly, e.g. the Japan’s nuclear power plant incident in 2011 prompted Germany to hasten the plan to shut down a number of nuclear power plants and to terminate the operation of the rest by 2022 [23]. In order to be able to quickly react to changes, the requirements to enable common access should be gathered now.

To ease accessibility, SPRUCE and many EGI communities set up science gateways. With the increasing use of mobile devices, there are additional advantages of a web-based solution. When a disaster strikes, scientists could initiate an urgent computation via the science gateways from virtually anywhere in the world over the Internet. Mobile apps specific to a use case can be a further step to improve usability.

Easy common accessibility to the required resources independent of the einfrastructures is an important criterion to ensure the success of urgent computing. This is especially so when a single use case uses multiple resources from multiple e-Infrastructures. Web-based solutions, i.e. science gateways and mobile apps, are good recommendations to facilitate accessibility and ensure common user interfaces. In combination with standardised access policies, the ease of using e-Infrastructures can be further enhanced.

5.3. Suitability of different computational resources

There are three categories of infrastructures, HPC, Grid and Cloud, which can be of interest to the field of urgent computing. For use cases with codes that require a huge amount of computation and/or data resources, the HPC resources are a better choice. HPC resources have large data storages, high-speed interconnects and are linked with fast networks. As such, it is useful for many use cases. In fact, most HPC resources can be used to run any kind of computations if required. Many public HPC resource providers, however, restrict the application codes allowed to run on their resources by certain criteria, e.g. parallel implementation, performance and scalability, to ensure that the expensive machines are used effectively. In addition, codes can have significantly varying performance on different HPC hardware. The resources selected for a particular code thus have to be carefully considered. Performance modelling can be a solution to help assess the suitability.
For use cases that can use less resources and/or compute in a massively parallel way, the high throughput Grid resources are a better choice. Although Grid infrastructure typically does not have data storages, interconnects and networks that are as big and fast as on a HPC infrastructure, it has many more resources and data storages that when combined together can be a force to contend with. Applications or codes that are suitable on the Grid infrastructures are those that are embarrassingly parallel or can be task farmed into many small parallel tasks and be run in parallel on different resources.

Public Clouds typically offer the midway solutions. Amazon cloud for instance provides both HPC and HTC (High Throughput Computing) resources. The available network bandwidth and storage space can be selected and configured as per requirements. The computational power might not be as impressive as what a HPC centre can offer but it can be sufficient for many use cases. It also has the advantage of being available on demand and thus is suitable for rarely occurring use cases when cost is taken into consideration.

In summary, the suitability of an e-Infrastructure is dependent on the use case. Often a number of e-infrastructures can be used. This is an advantage since backup resources should be planned to improve reliability (to be discussed in Section 5.6).

5.4. Use case specific setup

Although HPC application codes are expected to be scalable, some research codes that can be used for urgent computation might not scale. The reasons behind them vary, from the problem size, algorithmic implementation to inherent characteristics of the scientific problem. This does not imply that such codes are unsuitable for urgent computing. Instead, it is a characteristic that should be exploited. Requiring less resource offers the possibilities to use strategies that are normally not possible. One example is the Weather Research and Forecasting modelling system used in the DRIHM project where each run of the code for simulating a flash flood requires approximately 300 cores. Increasing the granularity of the mesh did not help to derive more detail information but triggered a slower convergence. The scientists found out that the strategy of performing multiple simulations while shifting the mesh actually derives more information than increasing the problem size. This particular use case can leverage on the huge HPC computation resources by performing smaller computations with shifted mesh in parallel. In fact, it opens up the possibility to exploit HPC Clouds and Grid solutions. Thus, each simulation can be run on multiple resources across different e-Infrastructures in parallel.

In Section 5.1, generic algorithms are proposed for urgent computing. One of the strategies that complements this algorithmic solution is to increase the population sizes used by performing parallel computations [13]. The recommendation is to use a bigger set of recorded data and thus reduce the uncertainty. Instead of feeding all data into a single computation, which will increase the total time needed, multiple smaller computations in parallel are proposed. Such a setup can allow the use case to leverage not only on HPC resources but also on Cloud and Grid resources.

In conclusion, the setup for each use case should be carefully considered and a use case specific setup should be exploited to take full advantage of the existing e-Infrastructures.

5.5. Policies or strategies of the infrastructure

In order to support urgent computing, the proposal to support various batch scheduler policies have been recommended by SPRUCE and others. The priority queue system was proposed and implemented in SPRUCE. Other suggestions, especially those in the US, are based on the priority queue framework. In principle, the technical requirements to support this framework can be fulfilled by most infrastructures. However, there are no policies that encourage or allow the providers to do so. In fact, most SPRUCE providers support only next to run and not preemption. Apparently, there is currently no SPRUCE resource that supports preemption.
Under the current policy restrictions, one strategy to support preemption without significant policy change can be to suspend small jobs, in particular test jobs, and drain the machine until the urgent job can commence. This strategy can be fulfilled by a number of HPC centres without violating their existing policies. Small jobs are considered because of several inherent advantages. On a big HPC system like SuperMUC, a normal production job would typically use more than 1000 cores. In addition to highly scalable application codes, parallel applications that use fewer cores per application instance but can utilise the same order of resources by task farming are also acceptable. In the task farming case, the submitted job should be in the size of at least 1000 cores with each task utilising less cores. Thus it is highly probable that most jobs that are using less than 1000 cores are test runs. Suspending test runs do not have a significant impact and the time required to prepare restart files is significantly shorter for jobs using fewer cores.

Table 3. Usage pattern on SuperMUC from 1st November 2012 to 4th December 2012

| Job size (cores) | Core hours used | No. of jobs | No. of jobs (%) |
|------------------|-----------------|-------------|-----------------|
| 1-64             | 656070          | 6468        | 23.3%           |
| 65-128           | 1009276         | 1744        | 6.3%            |
| 129-256          | 1605662         | 6555        | 23.6%           |
| 257-512          | 7465720         | 6567        | 23.7%           |
| 513-1024         | 3910953         | 1418        | 5.1%            |
| 1025-2048        | 7071963         | 2144        | 7.7%            |
| 2049-4096        | 5490584         | 2101        | 7.6%            |
| 4097-8192        | 4172542         | 376         | 1.4%            |
| 8193-16384       | 5880974         | 208         | 0.7%            |
| 16385-32768      | 7642383         | 95          | 0.3%            |
| 32769-147456     | 2589273         | 64          | 0.2%            |

Table 3 shows the SuperMUC’s usage statistic from November to early December 2012. Since SuperMUC is in its initial operation phase, the number of smaller jobs is expected to be higher than usual. Currently, 29.6% of the jobs are using 128 or less cores and 76.9% of the jobs are using 512 cores or less. Although one can expect fewer smaller jobs in the future, if 25% of the machine at any one time is filled with smaller jobs (<1000 cores), 36,000 cores can potentially be available for urgent jobs. In combination with the draining strategy, more cores can be made available. This implies that suspending only small jobs and draining the machine could allow many urgent jobs to get resources within a relatively short time. The usage pattern over a longer period and the time required to suspend a job from each job class have to be further collected before a more conclusive presumption can be drawn. Another candidate that can help in this investigation is JUQUEEN in Jülich, Germany. JUQUEEN requires production job to use at least 2048 cores, thus any jobs with less than 2048 cores can possibly be suspended. By involving more resources, an improved investigation result can be expected.

5.6. Reliability

Reliability is a very important factor to be taken into consideration in all aspects, e.g. data and network, and algorithms, services and infrastructure, to properly support urgent computation. Data that arrives on the computation resource should be reliable. If data corruption or packet loss occurs while staging, recovery must be possible. The packet should be re-sent since restart is not an option for urgent computing, especially if big data is involved. A reliable network should also be available to ensure that access to the computation resource
and data staging is possible at anytime. Additionally, the wide area network linked to the infrastructures should be reliable and if possible an acceptable bandwidth and quality of service should be assured. Dedicated high-speed network between resources could be an option. An example of such network is the GÉANT project, which provides pan-European data network dedicated to the research and education community.

Algorithms and services on the computation facilities for pre-processing, computation and post-processing should be reliable. To avoid failures, tests have to be conducted beforehand to ensure the installations are correct and that reliable results and services are computed and provided respectively. Similar preparations are performed well in advance in SPRUCE. Procedures should be in place to install and test new versions of codes thoroughly on each urgent computing resource that might be used for that particular urgent computation. When there are related software updates, in particular compiler updates, and/or system updates, all urgent computing software and services have to be tested again, i.e. automatic regression and functional tests, to ensure that they are still functioning well. In fact, regular tests should be scheduled to catch unforeseen issues. Monitoring systems should also be in place to ensure that up-to-date information on each resource is known.

To ensure infrastructure reliability, there should always be more than one resource for each use case. This is also applicable to cloud resources. Cloud outages happened even to big cloud providers like Amazon and Microsoft. It is thus important to always prepare backup resources and ensure that resources are in different localities. In event of a natural disaster, there could be a chance that a particular resource in the affected region becomes unavailable. In the aftermaths of the tsunami in Japan as mentioned in the earlier section, the lack of power renders the machines non-operational. A Cloud outage caused by an electrical storm at Amazon resulted in the shut down of hardware in a region [24]. Infrastructure monitoring is thus critical. Urgent computing services, e.g. the user interface, should always have up-to-date information on resources availability.

6. Conclusion

Urgent computing enables responsible authorities to make educated decisions by providing simulated predictions of disasters, the impact and required evacuation zones, etc. We believe that many urgent computing use cases that do not have dedicated resources can benefit from using existing e-Infrastructures, making them valuable for society and probably increasing their usage efficiency. Today’s e-Infrastructures are particularly interesting since they open up a big range of resources where each resource should have common access, tools and services. However, a number of challenges have to be faced to allow such e-Infrastructures to effectively and efficiently support urgent computing. Existing e-Infrastructures have policies that are in contradiction with the requirements of urgent computing and have characteristics that have to be considered to prepare computations on them. Use cases have unique features and requirements that should be carefully examined before deciding the strategies to enable them on these e-Infrastructures. Most solutions or proposed implementations assume that resources are available and thus tackled only on resource specific issues. SPRUCE provides perhaps the most complete infrastructure solution. However, it does not address issues like suitable algorithms, reliability, data, etc., which were either investigated by others or not mentioned. In order to prepare existing e-Infrastructures for urgent computing, all aspects have to be carefully considered.

This paper provides an evaluation of the on-going development work and solutions in areas on algorithmic solutions, accessibility of e-Infrastructures, suitability of different computation resources, use case specific setup, policies and strategies of the infrastructure and reliability. With this evaluation, we stress the point that tackling only one or two aspects of urgent computing is insufficient. In order to advance the field of urgent computing, all related and necessary innovative ideas and existing solutions should be leveraged upon. Obviously more testing is required since the systems involved are complex and interdependencies are manifold. The ultimate aim is to have a set of best practices and requirements for the e-Infrastructure providers and urgent computing users, and a compulsory framework implementing the set. As a result, resource providers and scientists can cooperate efficiently and easily when doing urgent computing on existing e-Infrastructures.
Acknowledgements

The authors would like to thank Dr. M. Brehm and Dr. R. Bader, Leibniz Supercomputing Centre, for making available SuperMUC’s usage statistics and offering valuable suggestions, and Dr. A. Parodi, CIMA Research Foundation, for sharing his DRIMH’s flash flood urgent computing use case. A very big thanks to Asst. Prof. Dr. I.L. Muntean, Technical University of Cluj-Napoca, for his vital guidance, motivation, suggestions, feedbacks and support without which this paper would never be completed on time. This work has been partially funded by the European Commission under its 7th Framework Programme within the e-infrastructure project VERCE, project reference 283543.

References

[1] P. Beckman, S. Nadella, N. Trebon, I. Beschastnikh, 2006, SPRUCE: A System for Supporting Urgent High-Performance Computing, In: Proceedings of the IFIP WoCo9 Conference, USA: Springer; p. 295-316.

[2] E. Yamasaki, 2012, What We Can Learn From Japan’s Early Earthquake Warning System, Momentum: Volume 1: Issue 1, Article 2.

[3] C. Nyquist, 2012, The USGS and Partners Work to Develop an Earthquake Early Warning System for California, USGS Top Story, 17 April.

[4] B. Blanton, J. McGee, J. Fleming, C. Kaiser, H. Kaiser, H. Lander, R. Luetttich, K. Dresback, R. Kolar, 2012, Urgent Computing of Storm Surge for North Carolina's Coast, Procedia Computer Science, Volume 9, p. 1677-1686.

[5] The TeraGrid Community Steps Up to Help Japan in Crisis, National Science Foundation, USA, viewed 9 November 2012, <https://www.nsf.gov/discoveries/disc_summ.jsp?cntn_id=119412&org=NSF%20Oil%20spill%20simulation>.

[6] About e-Infrastructures, ERINA+ 2012, viewed 9th November 2012, http://www.erinaplus.eu/index.php/about-e-infrastructures.

[7] Partnership for Advanced Computing in Europe 2010, PRACE AISBL, Bruxelles, Belgium, viewed 9 November 2012, <http://www.prace-ri.eu/>.

[8] European Grid Infrastructure, EGI.eu, Amsterdam, The Netherlands, viewed 9 November 2012, <http://www.egi.eu/>.

[9] The Extreme Science and Engineering Discovery Environment 2011-2012 Annual Highlights, 2012, USA.

[10] A. Parodi, G. Boni, L. Ferraris, F. Siccardi, P. Parilara, E. Traverato, E. Foufoula-Georgiou, D. Kranzlmüller, 2012, The “Perfect Storm”: From Across the Atlantic to the Hills of Genoa, EOS Transactions, AGU, Volume 93, Number 24, p. 225-226.

[11] M. Rivi, A. Emerson, Using the IBM iDataPlex (PLX), 21st Summer School of Parallel Computing, IN. 3 May 2012. Lecture.

[12] S. Marru, D. Gannon, S. Nadella, P. Beckman, D.B. Weber, K.A. Brewster, K.K. Droegemeier. LEAD Cyberinfrastructure to Track Real-Time Storms Using SPRUCE Urgent Computing, CTWatch Quarterly, Volume 4, Number 1.

[13] A. Cencerrado, A. Cortés, T. Margalef, 2012, On the Way of Applying Urgent Computing Solutions to Forest Fire Propagation Prediction, Procedia Computer Science, Volume 9, p. 1657-1666.

[14] N. Palmer, R. Kemp, T. Kiellmann, H. Bal, 2012, The Case for Smartphones as an Urgent Computing Client Platform, Procedia Computer Science, Volume 9, p. 1667-1676.

[15] K.K. Yoshimoto, D.J. Choi, R.L. Moore, A. Majumdar, E. Hocks, 2012, Implementations of Urgent Computing on Production HPC Systems, Procedia Computer Science, Volume 9, p. 1687-1693.

[16] S.V. Kovalchuk, A.V. Boukhanovsky, 2012, High-Level Knowledge-Based Structures for Simulation within Urgent Computing Tasks, Procedia Computer Science, Volume 9, p. 1694-1703.

[17] A. Geist, C. Engelmann, 2002, Development of Naturally Fault Tolerant Algorithms, Oak Ridge National Laboratory.

[18] P. Beckman, I. Beschastnikh, S. Nadella, N. Trebon, 2006, Building an Infrastructure for Urgent Computing, High Performance Computing and Grids in Action, IOS Press, Amsterdam, Volume 16, p. 25-95.

[19] M. Hegland, 2012, Multidimensional problems and fault tolerance, International Symposium on Parallel and Distributed Computing (ISPDIC 2012), IN. 26 June 2012. Invited talk.

[20] MPI Forum’s Fault Tolerance working group’s home page, MPI Forum, Bloomington, IN, USA, viewed 10 December 2012, <https://svn.mpi-forum.org/tracmpi-forum-web/wiki/FaultToleranceWikiPage>.

[21] T. Aboulnasr, Q. Pan, 2005, Data-dependent partial update adaptive algorithms for linear and nonlinear systems, European Signal Processing Conference (EUSIPCO2005), Invited talk.

[22] G. Hulten, L. Spencer, P. Domingos, 2001, Mining Time-Changing Data Streams, ACM KDD Conference, San Francisco, California, USA.

[23] A. Breidhardt, ‘German government wants nuclear exit by 2022 at latest’, Reuters UK Edition, 30 May 2011, viewed 10 December 2012, <http://uk.reuters.com/article/2011/05/30/us-germany-nuclear-idUKTRE74Q2P120110530>.

[24] J. Clark, ‘Amazon Web Services: The hidden bugs that made AWS’ outage worse’, ZDNet, 3 July 2012, viewed 10 December 2012, <http://www.zdnet.com/amazon-web-services-the-hidden-bugs-that-made-aws-outage-worse-7000000186/>.