Time critical requirements and technical considerations for advanced support environments for data-intensive research

Zhao, Z.; Martin, P.; de Laat, C.; Jeffery, K.; Jones, A.; Taylor, I.; Hardisty, A.; Atkinson, M.; Zuiderwijk-van Eijk, A.; Yin, Y.; Chen, Y.

DOI
10.5281/zenodo.204756

Publication date
2016

Document Version
Final published version

Citation for published version (APA):
Zhao, Z., Martin, P., de Laat, C., Jeffery, K., Jones, A., Taylor, I., Hardisty, A., Atkinson, M., Zuiderwijk-van Eijk, A., Yin, Y., & Chen, Y. (2016). *Time critical requirements and technical considerations for advanced support environments for data-intensive research*. Paper presented at International workshop on Interoperable infrastructures for interdisciplinary big data sciences, Porto, Portugal. https://doi.org/10.5281/zenodo.204756

Time critical requirements and technical considerations for advanced support environments for data-intensive research

Zhiming Zhao*, Paul Martin, Cees de Laat
University of Amsterdam, the Netherlands
*Z.Zhao@uva.nl

Keith Jeffery
Natural Environment Research Council
United Kingdom

Andrew Jones, Ian Taylor, Alex Hardisty,
Cardiff University, United Kingdom

Malcolm Atkinson
Edinburgh University, United Kingdom

Anneke Zuiderwijk-van Eijk, Yi Yin
Delft University of Technology, the Netherlands

Yin Chen
EGI.eu, the Netherlands

Abstract— **Data-centric approaches play an increasing role in many scientific domains, but in turn rely increasingly heavily on advanced research support environments for coordinating research activities, providing access to research data, and choreographing complex experiments. Critical time constraints can be seen in several application scenarios e.g., event detection for disaster early warning, runtime execution steering, and failure recovery. Providing support for executing such time critical research applications is still a challenging issue in many current research support environments however. In this paper, we analyse time critical requirements in three key kinds of research support environment—Virtual Research Environments, Research Infrastructures, and e-Infrastructures—and review the current state of the art. An approach for dynamic infrastructure planning is discussed that may help to address some of these requirements. The work is based on requirements collection recently performed in three EU H2020 projects: SWITCH, ENVRI[PLUS] and VRE4EIC.**

Keywords—virtual research environment, research infrastructure, e-Infrastructure, real time system.

I. Introduction

In many scientific domains such as in the environmental and earth sciences, data-centric approaches play an increasing role. To study the development of earthquakes or volcanoes for example, one needs continuous observation of the surrounding geographic regions and their underlying strata in order to obtain the data necessary to model various seismological processes and their interactions. Depending on the problem scale, these observations can only be provided by sources distributed across different countries, institutions and data centres. Moreover, such research activities also often require advanced computing and storage infrastructure in order to analyse, process, model and simulate the data. Advanced environments to support research (*research support environments)* are clearly needed to enable researchers to access data, software tools and services from different sources, and to integrate them into cohesive experimental investigations with well-defined, replicable workflows for processing data and tracking the provenance of results.

Based on the types of functionality needed, several kinds of support environments can be identified that must work in tandem to support data-centric research: 1) computing, storage and network infrastructures, e.g., provided via EGI [2] and EUDAT [3], also called *e-Infrastructures*; 2) services for accessing, searching and processing research data within different scientific domains, called *Research Infrastructures* (RIs) [1], e.g., ICOS [4], EPOS [5] and EURO-ARGO [6] for the environmental and earth sciences; and 3) environments for providing user-centred support for discovering and selecting data and software services from different sources, and composing and executing application workflows based on them, called *Virtual Research Environments* (VREs), *Virtual Laboratories* [9] or *Science Gateways* [10], e.g., D4Science [8] and e-Labs [14].

In these research support environments, *timeliness* or *speed* can be seen as a crucial factor in several application scenarios. An example would be when generating early warning of potential disasters based on iterative simulation: the processing of environmental observations and system simulation have to meet certain time constraints for predicting environmental behaviour and for making decisions for handling possible consequences. Another example is in the context of data acquisition and management where information from sensors is collected continuously; they need to be processed nearly immediately in order to provide services for accessing 'nearly real-time' information about the environment. In the user interaction context, when interacting with distributed processes, attempts to steer the computation should be responded to within certain time constraints in order to deliver a sufficiently high quality of user experience. In this paper, we refer to these applications using the term *time critical applications*[1], and refer to the time requirements for such applications as *time critical requirements* or *time critical constraints*.

Historically, the requirements for timeliness and speed in research support environments are often interpreted differently: e.g., parallelising computing tasks and minimising execution time versus optimising the network communication and controlling the latency or transmission time within certain small boundaries, or simulating a physical system based on wall clock time. In some cases, they are mixed with the term '*real-time*', which has a more confined meaning in the context of embedded systems. Nowadays, research support environments are often constructed across different nations, and focus on different service abstractions, e.g., scientist-centred activities (VREs), data lifecycle management (RIs), or computing, storage and network services (e-Infrastructures). Having a consistent view on those time critical requirements, and taking them into consideration during development, have become important engineering demands.

In the next section, we characterise the forms of research support environment currently existing or in production. The work described in this paper is conducted in the context of a number of on-going EU funded projects for research support environment: ENVRI^{PLUS} for interoperable research infrastructures for environmental and earth sciences [11], VRE4EIC for virtual research environments for multi domain RIs [8], and SWITCH for time critical applications in Clouds [38]. The main goal of this paper therefore is to analyse the time critical requirements extracted from the more general requirements recently collected from those projects (Section 3), and so identify the gaps based on the state of the art (Section 4). In addition, a dynamic real-time infrastructure planner proposed originally in the SWITCH project is discussed as a possible means to address some of those requirements, drawing upon a real-world use case as justification (Section 5). Our conclusions are summarized in Section 6.

II. TIME CRITICAL SCENARIOS IN RESEARCH SUPPORT ENVIRONMENTS

In this paper we specifically focus on the three types of research support environment identified earlier: VREs, RIs and e-Infrastructures. Based on the specific focuses of those different environments, an abstract logical relation among them can be seen in the layers of Fig. 1.

In this section, we will analyse the time critical requirements in different research environments in the context of a number of on-going projects.

A. Virtual research environments

Virtual Research Environments play a direct role in the *lifecycle of research activities* performed by scientists: e.g., for a) planning experiments, b) discovering resources from different sources, c) integrating them into a single application, d) executing the application and e) managing research data and collaborating with other scientists. Graphical environments, workflow management systems, and data analytics tools are typical components of such environments. In the lifecycle of research activities, time critical constraints apply to a number of scenarios, including:

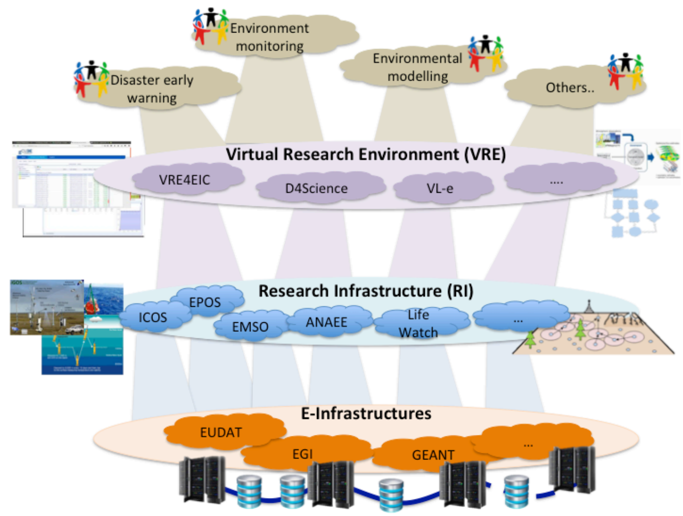

Figure 1. A layered view of the different kinds of research support environments and the role they play in ICT activities initiated by user communities.

1) *Data and service discovery.* Browsing catalogues of available data, services and other resources, and querying information from the catalogues are basic activities conducted when a researcher starts his research activities using a VRE. Response time is a key indicator of the performance of the catalogue service or the quality of user experience when using the system. Response time can also be important when the catalogues are invoked by software agents or application engines at runtime to retrieve additional information.

2) *Customisation of runtime environment and the scheduling of workflow applications.* These are important steps when planning the execution of an experiment, which may contain specific time constraints regarding the overall execution or specific parts of the application.

3) *Modelling and simulation of the evolution of physical systems.* Modelling and simulation are the main activities of many researchers studying complex systems, e.g., species migration in LifeWatch [14] or ecosystem dynamics in ANAEE [15]. Synchronising time across different individual simulation models can be crucial to modelling the correct behaviour of the overall system.

4) *Application steering and control features.* Runtime control of applications can be invaluable to researchers executing long runs and especially complex experiments. The VRE needs to not only provide the real time status of the system during execution, but should also permit user interventions to pause or reconfigure operations.

[1] Based on the definition of *time-critical* in the Oxford English dictionary (http://www.oed.com).

Naturally, scientists always want their data-related activities to be performed efficiently and effectively—which implies the VRE should impose minimal requirements for quality of user experience. In the VRE requirement collection and analysis in the VRE4EIC project [8], quality of user experience (QoS) in using a VRE was explicitly included, and more than forty per-cent of researchers interviewed expressed particular concerns about user experience. Response delays to user interaction, e.g., search and execution control, should be as short as possible and within certain bounds.

Other time constraints, e.g., in simulating physical system behaviour, should not only be as fast as possible, but should also take synchronisation between distributed components and managing simulation clocks and events into account.

B. Research infrastructure

Research Infrastructures play a key role in the *lifecycle of research data, services and the other assets*, providing security and access policies for e.g., the acquisition, curation, publication, processing and other usage of research data. RIs are often characterised by the types of research data they acquire and offer to their users, such as observations and measurements of earth systems in the case of environmental science research infrastructures [11], gene sequences and samples in marine biology research infrastructures [12], and document collections in humanities research infrastructures [13]. Sensor networks, data catalogues, and services for curation, identification, citation and provenance are typical components.

In the ENVRI[PLUS] project [11], key features and services of environmental research infrastructures are modelled with explicit consideration for their place in the research data lifecycle[2]. Key phases in the lifecycle include data acquisition, curation, publication, processing and usage. Critical time constraints can be seen from:

1) *Acquiring real-time observations and measurements*. Most environmental research infrastructures have underlying observation and measurement subsystems to monitor certain aspects of the world, e.g., the ocean, the geosphere, the atmosphere or biological ecosystems. Processing the observation data in real time and making them available in a nearly real-time manner are critical for certain RIs when they provide services based on the real time information, such as weather prediction or disaster early warning [16].

2) *Identifying and citing data.* Persistent identifiers allow users and software agents to refer to specific datasets correctly during the data lifecycle[3]. Assigning identifiers to raw sensor data or nearly real-time data, becomes another important requirement for an RI such as ICOS to provide data services.

3) *Publishing and accessing data with critical time constraints.* This can be seen as an important QoS requirement for delivering data content to users. During this process, time for checking data quality, annotating data with meta information, assigning identifiers, updating the relevant data catalogues, and informing data subscribers, all have to be finished within a certain time to guarantee the quality of service and user experience of the data user.

4) *Providing interfaces for VRE environments.* VREs rely on RIs to provide data access and service catalogues, and to carry out data processing. The quality of the services offered by research infrastructure is crucial for those critical time constraints required by applications generated within VREs. Besides offering services to meet quality constraints, research infrastructures should also explicitly model those QoS attributes as part of the meta information provided in the resource catalogues.

In many cases, users can directly interact with RIs, and perform research activities. Some of the RIs also include VRE-like components in their development plan, e.g., in the case of EPOS. In this case, requirements we mentioned in 2.A are also valid here.

C. e-Infrastructure

e-Infrastructures focus on the management of the *service lifecycle* of computing, storage and network resources. e-Infrastructures provide services for RIs or other applications to provision dedicated infrastructure, to manage computing tasks, and to deploy services needed for storage, data processing and other purposes. e-Infrastructures offer computing, storage, and network related services to their customers based on Service Level Agreements (SLAs). Critical time constraints can be seen from:

1) *Scheduling and execution of tasks.* The scheduling and execution of tasks on computing platforms including super-computers, compute clusters, and Grid and Cloud services are important services offered by e-Infrastructure providers such as EGI. Parallel computing has strict requirements for message passing between tasks, and for customised architecture for computing models, e.g., exploiting multiple cores, while other applications have very high demands on the processing of high volumes of data. High Performance Computing (HPC) and High Throughput Computing (HTC) are often characterised as the two main paradigms for choosing resources. While different computing tasks share the same resources, the infrastructure has to apply scheduling intelligence to ensure the time constraints required by each task are satisfied, or reserve sufficient resources in advance for those tasks to ensure their required performance.

2) *Customising, reserving and provisioning suitable infrastructures.* In a typical scenario for guaranteeing specific runtime system-level performance as needed by specific applications, the e-Infrastructure should take the critical time constraints required by the application, together with the cost of the resource and the total available resources into account and make an optimal

[2] As described in http://envri.eu/rm.

[3] Data lifecycle can be seen from ENVRI reference model, DCC curation lifecycle, http://www.dcc.ac.uk/resources/curation-lifecycle-model

plan for the user. This scenario is becoming increasingly important in contexts where the e-Infrastructure offers virtualised resources to its users.

3) *Monitoring runtime behaviour for the infrastructure, and providing controllability.* Monitoring and control are considered key services for high-level applications that want to benefit from the underlying programmability and controllability offered by the infrastructure, e.g., Cloud or software-defined networking.

4) *Failure recovery for deployed services and applications.* Failure recovery is often highlighted as a key service, in particular when supporting time or mission critical applications. Time constraints are not only imposed on failure detection, but also on decision-making and recovery actions.

The quality of e-Infrastructure services is crucial for their uptake by users. From the use cases we have analysed, we can clearly see these time constraints on scheduling and executing tasks, resource provisioning, monitoring and failure recovery.

D. *Requirements characterisation*

So far, we have only used the generic term “critical time constraints” to describe time-related quality requirements in research support environments. In this section, we shall first examine the different kinds of time critical requirement we have identified, and then align those requirements with the terms used by the real-time community. Based on this alignment, we will then proceed with the technology review in the following section.

Fig. 2 shows a proposed high-level taxonomy for classifying different time critical requirements. At an abstract level, service quality is the generic term to describe the different constraints of the application. We identify *timeliness and speed* related aspects from quality requirements and use them to identify time critical applications specifically. We then break our definition down further:

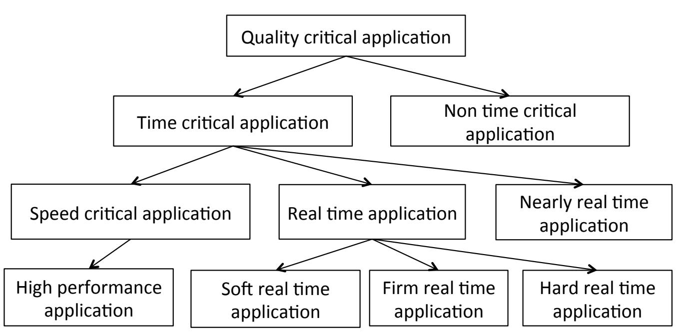

Figure 2. Terminologies related to time critical applications.

1) We have *speed critical applications*, where the objective is simply to minimise the completion time; these applications most suit the high-performance computing paradigm.

2) A *real-time application* is often characterised by bounded response time constraints, or *deadlines*, on the input to the system, with a risk of severe consequence on failure to meet these deadlines [17]. Based on the severity of the consequences if the system does not respond within the required time boundary, a real-time application is referred to as *hard real-time* when any deadline it misses leads to an effective failure of the application, *soft real-time* when missing deadlines only leads to a degradation of perceived performance, and *firm real-time* when no specific missed deadlines will lead to immediate application failure, but missing more than a few deadlines effectively results in failure anyway.

3) *Nearly real-time* (or *near real-time, or ‘NRT’ for short*) refers to a kind of delay introduced by data processing or transmission that is acceptable within certain bounds. Note that this is not the same as soft real-time, where failure to meet the real-time requirement may be acceptable in limited circumstances; nearly real-time applications can still impose a hard requirement for processing to fall within the permitted bounds, even those bounds are themselves quite broad.

These definitions are not mutually exclusive. The existence of deadlines is most significant within time critical applications. We can see such deadlines in a number of contexts from the examples of Section 2, as shown in Table 1 below.

Support environment	**Process**	**Critical time constraints**	**Type**
Virtual research environment	Data/service Search, query	Response time for getting results from queries	Soft
	Workflow orchestration / choreography	Execution deadlines for the entire workflow or specific processes	Soft
	Modelling and simulation	Simulation clock	Firm
	Application steering and runtime control	Response time for control operation	Soft
Research infrastructure	Data acquisition and quality control	QC time	Firm, NRT
	Data publication and access	Catalogue update,	Soft
		access response time	Soft
	Interface to VREs	Response time to specific service	Soft
	Data identification	Identifier assignment	Soft

e-Infrastructure	Task scheduling	Execution time of computing tasks	Firm
	Infrastructure reservation / provisioning	Provisioning delay	Soft
	Monitoring	Monitoring delay	Firm, NRT
	Failure recovery	Recovery	Firm

Table 1. Some selected time critical scenarios and time constraints in the research support environments.

Most processes in research support environments are 'soft' in in the sense that failure to meet deadlines is usually not disastrous. However processes that are continuously run in tandem with data acquisition can be seen to be firm real-time due to the 'knock-on' effect of repeated failure to process their inputs on time. Hard real time constraints are rare in the general sense, but may emerge in specific cases, e.g., where the process is made part of a safety-critical application such as disaster response.

In practical terms, the 'firmness' of a response time constraint dictates the degree of limited resource that should be allocated to ensuring the constraint. Isolated failures do not have the same impact as failures that beget further failures. It may be possible (and desirable) in specific research support environments to be able to assign a metric to constraints based on firmness that can be translated into concrete resource level requirements or adaptation strategies.

III. TECHNOLOGY REVIEW

We shall review the current technologies related to the processes we overviewed in Table 1, and then discuss the gaps between requirements and current technologies.

A. *Time critical information search and query*

Time critical data/service queries are generated when searching for information to address urgent needs or for mission critical business contexts, e.g., life saving or disaster decision making [18]. Typical technologies mainly include modelling the search activities of users, and their (future) interests [19], predicting future queries [20], optimising data catalogues [21], prioritising urgent tasks in search engines [18], and optimising information presentation of contextual information [22]. Such kinds of time critical queries are not currently treated differently in RIs or VREs, because many research activities are supported in current VREs only at a prototype level. We can see however several potential use case scenarios with mission critical characteristics, e.g., searching for urgent information in the context of real-time collaborative societal challenge studies.

B. *Time critical workflow execution*

Executing distributed workflow applications with critical time constraints has been studied for different types of tasks, whether short term tasks with fixed duration for computation, or persistent tasks (services) only performing computation when they are invoked or when certain events occur. The critical constraints in those applications often exist as deadlines for finishing the execution of the entire application or some sub-set of tasks, or for responding to a given invocation or event within a certain time. Scheduling the execution of such applications often needs consideration of not only the deadlines of individual tasks but also cost and occupation of resources [23]. Partial Critical Path based algorithms are the basic approach in solving such kinds of problem [24]. When customising virtual infrastructures based on IaaS provided by Cloud, a common approach will 1) select suitable virtual machines (VMs) based on certain task-VM performance matrices, 2) minimise communication costs between tasks by grouping tasks needing frequent communication in the same VM, and 3) refine the selection based on calculation of new critical paths. In this context, meta-heuristic approaches e.g., particle swarm optimisation [25] is often used to minimise the execution cost of the workflow within the overall deadline of the application. Multi-Objective Heterogeneous Earliest Finish Time (HEFT) is proposed to optimise the trade-off between monetary cost and the make-span of the application, although HEFT was originally designed for application scheduling on fixed rather than elastic infrastructures. Most existing work focuses on guaranteeing a single deadline encompassing the entire application, e.g., Critical Path-based Iterative (CPI) [26] and complete critical paths (CPIS) [27]. Those technologies have been widely investigated for applications modelled as Directed Acyclic Graphs (DAGs), e.g., data pipeline based applications. DAG-based methods are popular for building data-flows for data-intensive applications; the above technologies can be directly applied when those applications have partial or global deadlines.

C. *Real time modelling and simulation*

Simulating physical systems does not always require the simulation to run in real-time at wall clock rates [28], in particular if the goal is to understand evolutionary characteristics or to predict system behaviour. However, executing such simulations on distributed infrastructure does impose requirements on managing the simulation times of different sub-components, e.g., time management in High Level Architecture (HLA) for controlling the casual relation among events and time [29]. In data science, coupling different simulation models of individual systems can be performed to study the behaviours of complex systems, e.g., combining species distribution models with weather models to study how diseases are distributed via insects and species migration at different times.

D. *Real-time computing steering*

Real-time steering of the execution of a computing system requires not only monitoring and visualisation of the runtime status so that users can make decisions, but also execution in real-time to steering operations performed by users in response to that status. This has to be handled in an asynchronous manner, freeing up the user to perform other work while waiting for the execution to proceed to the next decision point.

The runtime status of an application can be obtained via monitoring of both the underlying platform and infrastructure, and the application itself. The infrastructure-level monitoring often takes place at network level, e.g., measuring throughput, round-trip time, packet loss and jitter [30], and on computing and storage nodes, e.g., measuring CPU usage, memory usage, and I/O [31]. In cloud environments, infrastructure monitoring can take place at both physical and virtual layers. Monitoring the service quality of cloud environments allow providers or users to evaluate compliance with SLAs, which requires mappings to be produced between resource metrics and SLA parameters, or the translation of Service Level Objectives (SLOs) to lower-level resource requirements [34]. At the application level, the application status monitoring often requires embedding probes inside application components. Any logging and provenance subsystems of distributed applications often captures the runtime status of the system as well [32]. To visualise the runtime status and to allow a user to make correct decisions regarding system control, different kinds of monitoring information together with the context of the system execution have to be harmonized based on the time stamp. Semantic technologies are often used to integrate such information and to offer query interfaces to link them [33].

Runtime steering of computing systems can take the form of adaptations of application logic at certain control points where the system actively provides time windows for users to intercede, or else the system can be interrupted by the user during execution [35]. The controllability of infrastructures e.g., dynamically configuring or scaling nodes [36], or controlling network flows [37], offer applications opportunities to refine the system performance. For these approaches to work, it is necessary to minimise the response time for any given control operation to meet critical time constraints. In recent projects such as SWITCH [38], automating application control based on performance diagnosis is highlighted as a key feature for time critical cloud applications. In research support environments, time critical computing system control can be seen in several on-going use cases in ENVRI^{PLUS} and VRE4EIC projects, e.g., adaptive data distribution for Euro-Argo data, and for quality control and processing of ICOS observation data [11].

E. *Real-time data acquisition and nearly real time data*

Acquiring real-time observations or monitoring data from devices or sensors are important processes in many research infrastructures [1]. The quality of the communication between sensors/devices and data processing units is crucial to control the delay in new data acquisition. Software-defined sensor networks can be used to optimise the communication between sensors [39], as can using edge nodes (in the context of edge computing) to enhance the communication between sensors and data processing nodes in Clouds [40].

To make sensor data available for users in near real-time in research infrastructures, partially automating data quality control and semantic annotation are important [41]. Currently, most data quality control is manually done by human experts; standardising part of the processing and using virtualised infrastructure to auto-scale that processing are mentioned in several use cases for environmental science RIs [42].

F. *Real-time data transfer*

Real-time data transfer between distributed components is a frequently seen task in data infrastructure applications. Technologies are being developed at both network and transfer service level. At network level, real-time data protocols [43], multi-path TCP and other protocols are used to optimise the throughput of data streaming. Software Defined Networking (SDN) [44] technologies are used to dynamically adapt the network flows between data sources and destinations, and traffic programming models such as co-flow [45] are also used to reschedule runtime data transfer between tasks. At transfer service level, dynamic schedule data transfer workers are used in LBSDER to handle the balance of data downloads [46].

G. *Infrastructure provision for time critical applications*

Efficient provisioning of networked virtual infrastructures essentially enables applications to adapt e-Infrastructure at runtime to meet time critical requirements. Container technologies can significantly reduce the bootstrap time of virtual nodes. Optimising VM image size, e.g., in FVD [47], or directly forking runtime images from memory e.g., in SnowFlock [48] and Twinkle [49] can reduce the provisioning time, but can only applied at the provider side. Using P2P or SDN technology to optimise VM image transfer among data centres [50] is also possible. As for dynamic provisioning, previous related research mainly includes network embedding and inter-cloud architecture to satisfy dynamic requirements. In the SWITCH project, a transparent network virtual infrastructure graph partitioning and parallel provisioning approach is being developed to map infrastructure over different data centres and to enable dynamic adaptation of the topology [51]. These IaaS-based provisioning mechanisms exhibit large potential for supporting large distributed applications over large e-Infrastructures.

H. *Real-time Service Level Agreement*

Real-time support of virtualised infrastructure has attracted more and more interest [52]. SLAs for real-time applications, and the negotiation of such agreements at runtime, will be crucial for supporting real-time applications in Cloud. In this context, real-time task schedulers at operating system, network, hypervisor and virtualised infrastructure levels should all be taken into account by developers. SLA issuing and real-time negotiation technologies depend heavily on the complexity of the mapping between application requirements and the available resources, and the matching between quality requirements at different service layers. Most mapping approaches are based on graph mapping using key quality parameters such as execution time; however limited association between the application and infrastructure during application development makes the searching procedure over large resource graph very time consuming. In this context, the main approach currently taken to improve the search procedure is to include different types of heuristics and optimisation technologies, for instance parallelising the searching procedure for matching resources and applications [53], pre-processing the resource information by clustering the resource information based on the SLA request, and multi-objective optimisation for searching out alternative solutions [54]. However, SLA negotiations in e-Infrastructures mainly rely on human dialogue. The SLA attributes for time critical applications also hardly include.

I. *Real-time task schedulers*

In time critical systems, scheduling tasks in an optimal execution order is crucial to ensure that the system meets its deadlines. Depending on the type of tasks, different scheduling strategies can be developed. A typical way to determine the order of execution is based on the priorities of tasks. These priorities can be statically assigned or dynamically adapted based on the temporal status of the system. Algorithms like least completion time, earliest data deadline first, and least slack time are typical examples.

IV. CHALLENGES AND GAP ANALYSIS

In the above section, we can see a number of time critical research scenarios and technical requirements for different research support activities.

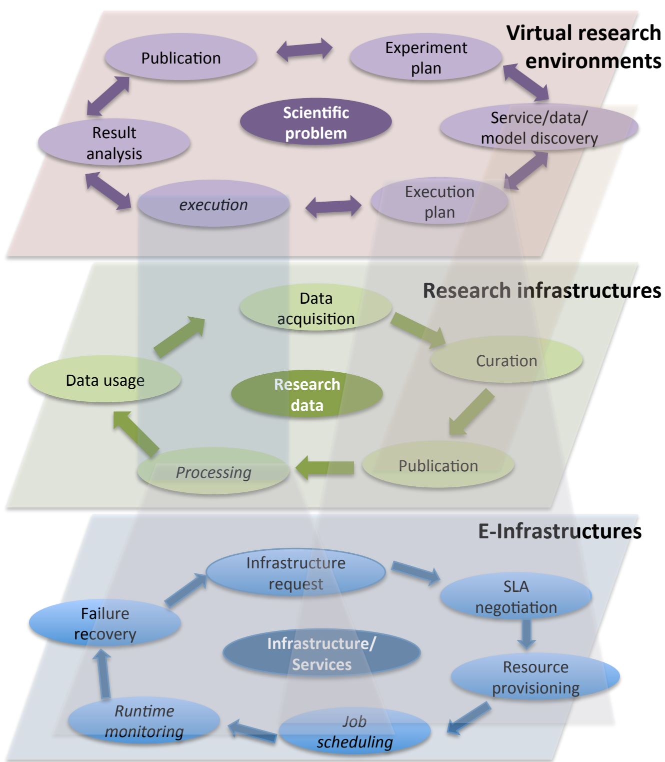

Figure 3: Activities require real-time support

Fig. 3 highlights what has been discussed in the previous two sections. In some cases, activities can cross different services provided by different support environments; supporting time critical requirements in these cases becomes very challenging.

1) To meet requirements for searching data and software services in virtual research environments, the underlying data query and access services from different research infrastructures and the storage services and virtual infrastructure where those RI services are deployed are all required to meet certain time critical constraints.

2) To develop a time critical application in a VRE, the developer not only needs to describe time constraints at application level, but also compose an application workflow—by choosing suitable services, data sources from research infrastructures, and by customising runtime virtual infrastructures for the application and provisioning it in an optimal e-Infrastructure.

3) During the execution of time critical applications in VREs, data sources, software components, and the execution engines of some parts of the application will have to be handled by different underlying research infrastructures or e-Infrastructures. The scheduler in low-level infrastructures has to guarantee the application-level time constraints.

From the literature and recent projects, we can see that engineering approaches and scheduling intelligence for time critical constraints have been hot topics for control systems, embedded systems and many other application fields. However, we can still see many technical gaps when applying these technologies to developing research support environments for time critical applications, including:

1) A lack of low-level support from e-Infrastructures and RIs. In current e-Infrastructures, e.g., EGI, high performance and high throughput are two key computing service models for applications. The real-time support at operating system, hypervisor and virtual machine levels are currently not yet fully realized by e-Infrastructures.

2) Moreover, the Service Level Agreements provided by e-Infrastructures do not yet support real time applications with strict response time constraints. In fact, even among commercial Cloud providers the support for time critical applications is in a very early phase.

3) In addition, most current development and execution of research applications focus on handling data flows and on integrating distributed computing tasks. The support for verifying time constraints at application level, and for scheduling and controlling the application at the underlying infrastructure level are currently very limited.

V. A PROPOSED APPROACH IN SWITCH

In this section we introduce a solution being developed in the SWITCH project for supporting time critical applications in Cloud, and then discuss a technical path to apply this technology in research support environments for time critical research activities.

A. *The SWITCH approach*

The SWITCH project tackles software engineering challenges for developing time critical applications in Clouds. It aims at handling time critical applications at three key phases in their lifespan: application development, virtual infrastructure customisation and provisioning, and runtime control, as shown in Fig. 4.

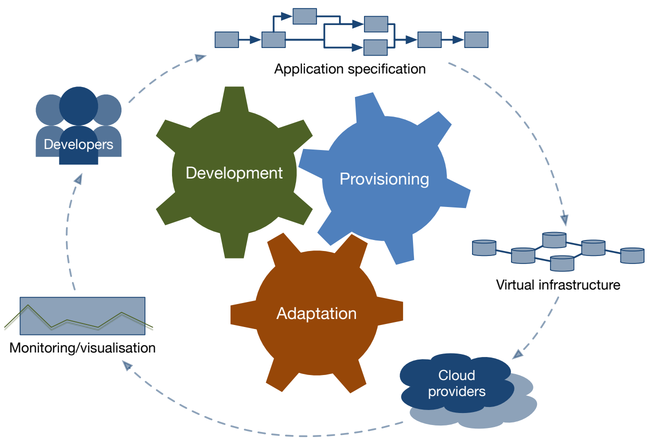

Figure 4. Application lifecycle and software workbench.

The core idea of the SWITCH environment is a new development and execution model, the application-infrastructure co-programming and control model, which will be developed for time critical cloud applications. The new model brings together application composition, execution environment customisation, and runtime control, which are normally treated as separate processes, into one optimisation loop based on time critical requirements. In this model:

1) The application logic will be programmed with consideration for critical time constraints together with the programmability and controllability of the cloud environment such that both the application and the virtual runtime environment for executing the application can be programmed and optimised together during the design phase.

2) A virtual runtime environment (a runtime environment created in the cloud for executing the application) can be customised to address critical application requirements, and can then be provisioned in the cloud with SLAs oriented towards time critical application requirements.

3) The application can autonomously adapt its own behaviour and that of the virtual runtime environment when performance drops at runtime. The SWITCH environment employs formal performance reasoning mechanisms to guide each step in the development, and tools are delivered to the users via each of three subsystems. The three sub-systems proposed in SWITCH are as follows:

The SWITCH Interactive Development Environment (SIDE) subsystem provides interfaces for all of the user- and programmer-facing tools, by exposing a collection of graphical interfaces and APIs that tie in SWITCH's services to a Web-based environment. The time constraints of applications can be described together with an initial plan of the virtual infrastructure. Standardised language is used to describe application constraints and infrastructure design.

The Dynamic Real-time Infrastructure Planner (DRIP) subsystem uses an extended critical path algorithm to select optimal virtual machines and to plan a customised virtual infrastructure for the application. A provisioning agent maps the infrastructure graph onto available data centres and provisions the infrastructure. Control agents are provided for application to manipulate the underlying infrastructures.

The Autonomous System Adaptation Platform (ASAP) monitors the real time runtime status of the application from both service level and the infrastructure level. Based on the monitoring information, the ASAP subsystem diagnoses the application's status and makes decisions on adapting the application's behaviour by calling control agents deployed alongside the application by DRIP.

B. A RI use case using SWITCH

SWITCH can contribute partially to the technologies needed by research support environments from different aspects. We shall discuss these aspects using a typical use case scenario in environmental RIs.

The use case is about cataloguing near real-time data and distributing data to different data subscribers. The basic scenario includes 1) collecting data from sensors, 2) updating the catalogues of near real-time data, and 3) distributing data to the remote data subscribers. In order to provide such data services to data subscribers with guaranteed service quality; several time constraints have to be considered, including 1) the data delivery time between sensor stations and the data repository, 2) the time cost of annotating metadata and updating the relevant data catalogues, 3) the time for storing data into repositories, 4) the time for retrieving data from repository, and 5) the time for delivering data content to distributed subscribers. We use this example to explain how an RI can plan and provision a customised virtual infrastructure for such service.

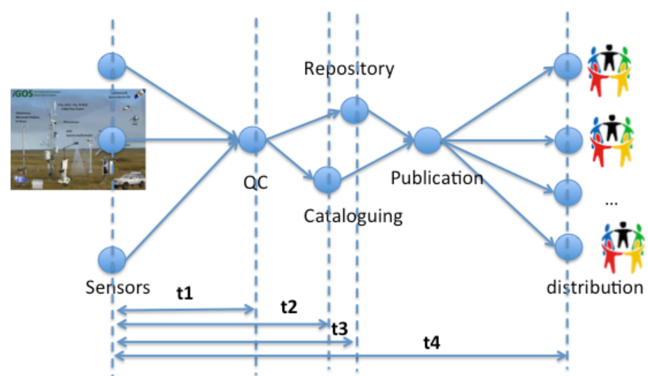

Figure 5. Time constraints in the data distribution use case.

The SWITCH can support research-support environments in the following aspects:

1) *Infrastructure description and planning based on the time constraints derived from the use case.* The planner selects VMs based on the performance of each task, as shown in Fig. 6.

2) *Provisioning virtual infrastructure on e-Infrastructure via standardised interface.* The provisioning component selects data centres and network connections based on the location of sensors and subscribers, and provision the infrastructure (decompose them into different parts if necessary) on one or more data centres.

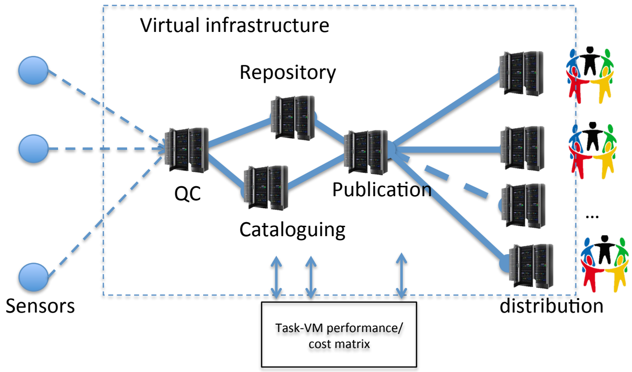

Figure 6. Planned virtual infrastructure for the data distribution use case.

3) *Scale-out of the distribution services based on the location of the subscribers.* The SWITCH is able to monitor the runtime performance and to dynamically adapt the infrastructure based on the runtime situation.

C. Discussion

The development of time critical solutions in research support environments is still in a very early phase. The original motivation of SWITCH is for industrial applications. Conceptually, we have demonstrated feasibility of using such technology in RI or e-Infrastructures. Currently, this activity is still on-going. The SWITCH provisioning is being tested with the EGI e-Infrastructure. The detailed development of the use case is on-going. Detailed implementation will be presented in a separate paper. An important consideration is the run-time overhead of monitoring and adaptation. The SWITCH approach is to use lightweight monitoring probes alongside application components and moving the analysis and decision logic to dedicated resources provisioned alongside applications; this should minimise overhead, but requires further study to prove that this is the case. In the meantime, applying the solutions from real-time communities, e.g., the scheduling algorithms at OS kernel, hypervisor, and virtualised resources, are attracting more and more interest.

VI. Summary

In this paper, we analysed requirements for time critical constraints in research support environments from different scenarios, and identified gaps between requirements and the current technologies based on technology review. The work is performed in the context of three on-going projects: SWITCH, ENVRI^{PLUS} and VRE4EIC.

We highlight a number of challenging open issues in the development of research support environments. We investigate the requirements across different abstraction of the research support environments, and aims to provide guidelines for the development of different platforms to choose technologies in order to enable those time critical scenarios.

Acknowledgement

This work was supported by the European Union's Horizon 2020 research and innovation program under grant agreements No. 654182 (ENVRI^{PLUS} project), No. 676247 (VRE4EIC project) and No. 643963 (SWITCH project).

References

[1] Legal framework for a European Research Infrastructure Consortium- ERIC: https://ec.europa.eu/research/infrastructures/pdf/eric_en.pdf

[2] European Grid Initiative, egi.eu, www.egi.eu

[3] European Data Infrastructure (EUDAT) http://www.eudat.eu/

[4] ICOS, a European distributed infrastructure dedicated to the monitoring of greenhouse gases (GHG) through its atmospheric, ecosystem and ocean, http://www.icos-infrastructure.eu/.

[5] EPOS, a European Research Infrastructure on earthquakes, volcanoes, surface dynamics and tectonics, http://www.epos-eu.org/.

[6] EURO-Argo, the European contribution to Argo, which is a global ocean observing system, http://www.euro-argo.eu/.

[7] L. Candela, D. Castelli, A. Manzi, P. Pagano, Realising Virtual Research Environments by Hybrid Data Infrastructures: the D4Science Experience. International Symposium on Grids and Clouds (ISGC) 2014, Proceedings of Science PoS(ISGC2014)

[8] VRE4EIC, H2020 project www.vre4eic.eu

[9] Belloum, A.S.Z., Inda, M. A., Vasunin, D., Korkhov, V., Zhao, Z., Rauwerda, H., Breit, T., Bubak, M.T. & Hertzberger, B. (2011). *Collaborative e-Science Experiments and Scientific Workflows*. IEEE Internet Computing, 15(4), 39-47.

[10] Mark A. Miller, Wayne Pfeiffer, and Terri Schwartz. 2012. The CIPRES science gateway: enabling high-impact science for phylogenetics researchers with limited resources. In *Proceedings of the 1st Conference of the Extreme Science and Engineering Discovery Environment: Bridging from the eXtreme to the campus and beyond* (XSEDE '12). ACM, New York, NY, USA

[11] Environmental Research Infrastructure Cluster, EU H2020 project, www.envriplus.eu

[12] European Marine Biology Resource Centre (EMBRC): http://www.embrc.eu/

[13] CLARIAH: http://www.clariah.nl/

[14] LifeWatch, an e-science Infrastructure for biodiversity and ecosystem research, http://www.lifewatch.eu/.

[15] Anaee, analysis and experimentation on eco-systems, http://www.anaee.com/

[16] A Semantic IoT Early Warning System for Natural Environment Crisis Management", *IEEE Transactions on Emerging Topics in Computing*, vol.3, no. 2, pp. 246-257, June 2015

[17] Phillip A. Laplante, 2005, Real time system design and analysis, WileyInterscience.

[18] Nina Mishra, Ryen W. White, Samuel Ieong, and Eric Horvitz. 2014. Time-critical search. In*Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval* (SIGIR '14). ACM, New York, NY, USA

[19] R. White and S. Drucker. Investigating behavioural variability in web search. In WWW, pages 21–30, 2007

[20] D. Downey, S. Dumais, and E. Horvitz. Models of searching and browsing: Languages, studies, and application. In Proc. IJCAI, pages 2740–2747, 2007.

[21] E. Agichtein, E. Brill, and S. Dumais. Improving web search ranking by incorporating user behaviour information. In Proc. SIGIR, pages 19–26, 2006.

[22] P. Ingwersen and K. J¨arvelin. The turn: Integration of information seeking and retrieval in context, volume 18. Springer, 2005.

[23] J. Yu, R. Buyya, C. K. Tham, Cost-based scheduling of scientific workflow applications on utility grids, in: e-Science and Grid Computing, 2005. First International Conference on, IEEE, 2005, pp. 8–pp.

[24] S. Abrishami, M. Naghibzadeh, D. H. Epema, Deadline-constrained workflow scheduling algorithms for infrastructure as a service clouds, Future Generation Computer Systems 29 (1) (2013) 158–169.

[25] M. A. Rodriguez, R. Buyya, Deadline based resource provisioningand scheduling algorithm for scientific workflows on clouds, Cloud Computing, IEEE Transactions on 2 (2) (2014) 222–235.

[26] Z. Cai, X. Li, J. N. Gupta, Critical path-based iterative heuristic for workflow scheduling in utility and cloud computing, in: Service-Oriented Computing, Springer, 2013, pp. 207–221.

[27] Z. Cai, X. Li, J. N. Gupta, Heuristics for provisioning services to workflows in xaas clouds.

[28] Shadi Alawneh, Roelof Dragt, Dennis Peters, Claude Daley, Stephen Bruneau, "Hyper-Real-Time Ice Simulation and Modeling Using GPGPU", *IEEE Transactions on Computers*, vol.64, no. 12, pp. 3475-3487, Dec. 2015

[29] Zhao, Z., Albada, G.D. van & Sloot, P.M.A. (2005). *Agent-based flow control for HLA components*. International Journal of Simulation Transaction. 81(7), 487-501

[30] Ahmet Bulut, Nick Koudas, Anand Meka, Ambuj K. Singh, Divesh Srivastava, "Optimization Techniques for Reactive Network Monitoring", *IEEE Transactions on Knowledge & Data Engineering*, vol.21, no. 9, pp. 1343-1357, September 2009

[31] Jimeng Sun, Evan Hoke, John D. Strunk, Gregory R. Ganger, and Christos Faloutsos. 2006. Intelligent system monitoring on large clusters. In *Proceedings of the 3rd workshop on Data management for sensor networks: in conjunction with VLDB 2006* (DMSN '06)

[32] Artem Chebotko, Shiyong Lu, Seunghan Chang, Farshad Fotouhi, Ping Yang, "Secure Abstraction Views for Scientific Workflow Provenance Querying", *IEEE Transactions on Services Computing*, vol.3, no. 4, pp. 322-337, October-December 2010

[33] Taherizadeh, S., Jones, A., Taylor, I., Zhao,Z., Martin, P., Stankovski, V, (2016) *Runtime Network-level Monitoring Framework in the Adaptation of Distributed Time-critical Cloud Applications*, The 22nd International Conference on Parallel and Distributed Processing Techniques and Applications, 2016

[34] Saadeldin Moustafa, Khalid Elgazzar, Patrick Martin, Marwa Elsayed, "SLAM: SLA Monitoring Framework for Federated Cloud Services", *UCC*, 2015, 2015 IEEE/ACM 8th International Conference on Utility and Cloud Computing (UCC), 2015

[35] Angeliki Kritikakou, Claire Pagetti, Olivier Baldellon, Matthieu Roy, Christine Rochange, "Run-Time Control to Increase Task Parallelism In Mixed-Critical Systems", *ECRTS*, 2014, 2014 26th Euromicro Conference on Real-Time Systems (ECRTS), 2014

[36] Nicolas Serrano, Gorka Gallardo, Josune Hernantes, "Infrastructure as a Service and Cloud Technologies", *IEEE Software*, vol.32, no. 2, pp. 30-36, Mar.-Apr. 2015

[37] Alan Nussbaum, Shwetha Mathangi Chandra Choodamani, Karsten Schwan, "ObsCon: Integrated Monitoring and Control for Parallel, Real-Time Applications", *CLUSTER*, 2015, 2015 IEEE International Conference on Cluster Computing (CLUSTER)

[38] EU H2020 SWITCH www.switch-project.eu

[39] Angelos-Christos G. Anadiotis, Laura Galluccio, Sebastiano Milardo, Giacomo Morabito, Sergio Palazzo, "Towards a software-defined Network Operating System for the IoT", *WF-IOT*, 2015, 2015 IEEE 2nd World Forum on Internet of Things (WF-IoT)

[40] Apostolos Papageorgiou, Bin Cheng, Erno Kovacs, "Real-time data reduction at the network edge of Internet-of-Things systems",*CNSM*, 2015, 2015 11th International Conference on Network and Service Management (CNSM)

[41] Shaohan Hu, Shuochao Yao, Haiming Jin, Yiran Zhao, Yitao Hu, Xiaochen Liu, Nooreddin Naghibolhosseini, Shen Li, Akash Kapoor, William Dron, Lu Su, Amotz Bar-Noy, Pedro Szekely, Ramesh Govindan, Reginald Hobbs, Tarek F. Abdelzaher, "Data Acquisition for Real-Time Decision-Making under Freshness Constraints", *RTSS*, 2015, 2015 IEEE Real-Time Systems Symposium (RTSS), 2015

[42] Nuno Laranjeiro, Seyma Nur Soydemir, Jorge Bernardino, "A Survey on Data Quality: Classifying Poor Data", *PRDC*, 2015, 2015 IEEE 21st Pacific Rim International Symposium on Dependable Computing (PRDC)

[43] Mohammad Javad Shamani, Weiping Zhu, Vahid Naghshin, "TMPTCP: Tailless Multi-path TCP", *BWCCA*, 2015, 2015 10th International Conference on Broadband and Wireless Computing, Communication and Applications (BWCCA), 2015

[44] Zhao, Z., Grosso, P., Ham, J. van der, Koning, R. & Laat, C.T.A.M. de (2011). *An agent based network resource planner for workflow applications*. International journal of Multiagent and Grid Systems, 7(6), 187-202

[45] Zhouwang Fu, Tao Song, Sheng Wang, Fuzong Wang, Zhengwei Qi, "Seagull -- A Real-Time Coflow Scheduling System", *CSCLOUD*, 2015, 2015 IEEE 2nd International Conference on Cyber Security and Cloud Computing (CSCloud)

[46] Koulouzis, S., Belloum, A., Bubak, M., Zhao, Z., Zivkovic, M., de Laat, C. (2015) *SDN-Aware Federation of Distributed Data*, International journal of Future Generation Computer System

[47] C. Tang, "Fvd: A high-performance virtual machine image format for cloud." in USENIX Annual Technical Conference, 2011.

[48] H. A. Lagar-Cavilla, J. A. Whitney, A. M. Scannell, P. Patchin, S. M. Rumble, E. De Lara, M. Brudno, and M. Satyanarayanan, "Snowflock: rapid virtual machine cloning for cloud computing," in Proceedings of the 4th ACM European conference on Computer systems. ACM, 2009, pp. 1–12.

[49] J. Zhu, Z. Jiang, and Z. Xiao, "Twinkle: A fast resource provisioning mechanism for internet services," in INFOCOM, 2011 Proceedings IEEE. IEEE, 2011, pp. 802–810.

[50] R. Wartel, T. Cass, B. Moreira, E. Roche, M. Guijarro, S. Goasguen, and U. Schwickerath, "Image distribution mechanisms in large scale cloud providers," in Cloud Computing Technology and Science, 2010 IEEE 2nd International Conference on. IEEE, 2010, pp. 112–117.

[51] Zhou, H., Hu Y., Wang, J., Martin, P., de Laat, C. and Zhao, Z., (2016) *Fast and Dynamic Resource Provisioning for Quality Critical Cloud Applications*, IEEE International Symposium On Realtime Computing (ISORC) 2016

[52] Yu Hua, Hong Jiang, Dan Feng, "FAST: Near Real-Time Searchable Data Analytics for the Cloud", *SC*, 2014, SC14: International Conference for High Performance Computing, Networking, Storage and Analysis, SC14: International Conference for High Performance Computing, Networking, Storage and Analysis 2014, pp. 754-765

[53] Muler, C., Oriol, C., Franch, X., Marco, J., Resinas, M., Ruiz-Cort, A., Rodruez, A., (10`3) *Comprehensive Explanation of SLA Violations at Runtime*, IEEE Transactions on Services Computing, vol. 99, no. PrePrints, p. 1,

[54] Antonescu, A., Oprescu, A., Demchenko, Y., de Laat, C., Braun T., (2013) *Dynamic Optimization of SLA-Based Services Scaling Rules*, IEEE 5th International Conference on Cloud Computing Technology and Science, pp. 282-289