Processing

Introduction defining context and scope

There are a great many requirements for processing at every stage of the data lifecycle from validating, error correcting and monitoring during data acquisition to transformations for comprehensible presentations of final results. Every step in between has major processing requirements. All forms of data preparation, filtering and transformation to achieve consistent input to subsequent stages in the data lifecycle or the next step in a scientific method. Analysis, pattern matching and statistical reduction to extract relevant signals from complex and noisy data. Large-scale simulations to generate the implications of current models, correlation of those results with well-prepared derivatives from observations and then refinement of the models.

A lot of technologies and approaches have been developed to support these tasks including:

- ***High Performance Computing solutions***, i.e., aggregated computing resources thus to realise an “high performance computer” (including processors, memory, disk and operating system);
- ***Distributed Computing Infrastructures***, i.e., distributed systems characterised by heterogeneous networked computers called to offer data processing facilities. This includes high-throughput computing and cloud computing;
- ***Scientific workflow management systems (SWMS)***, i.e., systems enacting the definition and execution of *scientific workflows* consisting of [Liew 2016]: a list of tasks and operations, the dependencies between the interconnected tasks, control-flow structures and the data resources to be processed;
- ***Data analytics frameworks and platforms***, i.e., platforms and workbenches enabling scientists to execute analytic tasks. Such platforms tend to provide their users with implementations of algorithms and (statistical) methods for the analytics tasks.

These classes of solutions and approaches are not isolated, rather they are expected to rely on each other to provide end users with easy to use, efficient and effective data processing facilities, e.g., SWMS rely on distributed computing infrastructures to actually execute their constituent tasks.

In Europe, PRACE definitely represents the major initiative for High Performance Computing. Similarly, EGI is a point of reference for distributed computing. Both these initiatives are discussed in detail in other parts of this deliverable (see Compute, storage and networking) and will not be further analysed in this section. In this section we will thus focus on Scientific Workflow Management Systems and Data Analytics frameworks and platforms.

Over the last two decades, many large-scale scientific experiments take advantage of scientific workflows to model data operations such as loading input data, data processing, data analysis, and aggregating output data.

The term workflow refers to the automation of a process, during which data is processed by different logical data processing activities according to a set of rules, along with the attendant tasks of, for example, moving data between workflow processing stages. Workflow management systems (WMS) [Bux 2013] aid in the automation of these processes, freeing the scientist from the details of the process, since WMS manage the execution of the application on a computational infrastructure.

Scientific workflows allow scientists to easily model and express the entire data processing steps and their dependencies, typically as directed Acyclic Graph (DAG), whose nodes represent workflow steps that are linked via dataflow edges, thus prescribing serial or parallel execution of nodes.

Scientific workflows have different levels of abstraction: abstract and concrete. An abstract workflow models data flow as a concatenation of conceptual processing steps. Assigning actual methods to abstract tasks results in a concrete workflow.

There are four key properties of scientific workflows, which are handled differently in each scientific workflow management:

- **Reusability**: Workflow management systems have to make it easier for workflow designer to reuse their previously developed workflows in their under development workflows. Many workflows provide mechanisms for tracing provenance and methodologies that foster reproducible science [Santana-Perez 2015].
- **Performance**: Workflow optimisation is not a trivial task, there are different methods can be applied on a workflow to reduce the execution time [Spinuso 2016].
- **Design**: Almost all the modern workflow management systems provide a rich graphical user interface for creating workflows. The aim of providing graphical composition mechanism is to ease the step of describing workflows for the workflow developers.
- **Collaboration**: Due to the increase in the number of workflows and collaborative nature of scientific research projects developing share and collaboration mechanisms through the network and Internet for workflows is a must. Some projects such myExperiment [De Roure 2009], Wf4Ever [Belhajjame 2015], and Neuroimaging workflow reuse [Garijo 2014], are specially focused on this approach.

Scientific workflows perform two basic functions. They manage (a) the execution of constituent codes and (b) information exchanged between them. Therefore, an instantiation of a workflow must represent both the operations and the data products associated with a particular scientific domain. It should be assumed that individual operations and data products were developed independently in an uncoordinated fashion. Workflows must be usable by the target audience (computational scientists) on target platforms (computing environments) while being represented by abstractions that can be reused across sciences and computing environments and whose performance and correctness can be modelled and verified.

In parallel with scientific workflows, a series of platforms and frameworks have been developed to simplify the execution of (scientific) distributed computations. This need is not new, it is actually rooted in high-throughput computing which is a well-consolidated approach to provide large amounts of computational resources over long periods of time. The advent of Big Data and Google MapReduce in the first half of 2000 brings new interests and solutions. Besides taking care of the smart execution of user-defined and steered processes, platforms and environments start offering ready to use implementations of algorithms and processes that benefits from a distributed computing infrastructure.

Change history and amendment procedure

The review of this topic will be organised by Leonardo Candela in consultation with the following volunteers: @rosa. They will partition the exploration and gathering of information and collaborate on the analysis and formulation of the initial report. Record details of the major steps in the change history table below.For further details of the complete procedure see item 4 on the Getting Started page.

Note: Do not record editorial / typographical changes. Only record significant changes of content.

Date	Name	Institution	Nature of the information added / changed

Sources of information used

Two major sources of information have been used, literature available discovered by the web and technologies web sites. In particular, the following websites have been source of information:

- Apache Airavata website airavata.apache.org
- Apache Spark website spark.apache.org
- dispel4py website dispel4y.org
- Galaxy website galaxyproject.org
- gCube website www.gcube-system.org
- Kepler website kepler-project.org
- KNIME website www.knime.org
- Pegasus website pegasus.isi.edu
- Taverna website www.taverna.org.uk
- Triana website www.trianacode.org
- Wf4Ever website www.wf4ever-project.org
- WINGS website www.wings-workflows.org

Two-to-five year analysis

State of the art

*** A snapshot by Aleksi Kallio (CSC) ***

The hype was big data technologies started with Google MapReduce, which soon was implemented in open source by Apache Hadoop. Hadoop consists of two major components, the Hadoop Filesystem for storing data in replicated and distributed manner, and the map-reduce execution engine for batch processing data. Hadoop remains to be the mostly widely used system for production workloads, but many alternative technologies have been introduced. Most notably Apache Spark has quickly gained a wide user base. It provides efficient ad hoc processing, in-memory computing and convenient programming interfaces. Apache Spark is typically used in conjunction with the Hadoop Filesystem. Database-like solutions include e.g. Hive, the original and robust system for heavy queries, Apache Cassandra for scalable and highly available workloads and Cloudera Impala for extreme performance. Apacha Spark also provides Spark SQL for queries.

The most used programming languages for data science tasks are Python and R. Python is widely used for manipulating and pre-processing, but via popular libraries such as Pandas it also support rich variety of data analysis methods. The R language is the de facto tool for statistical data analysis, boasting the most comprehensive collection of statistical methods freely available. Both languages have bindings to big data systems such as Apache Spark.

Trends

Subsequent headings for each trend (if appropriate in this HL3 style)

Problems to be overcome

Sub-headings as appropriate in HL3 style (one per problem)

Details underpinning above analysis

Sketch of a longer-term horizon

Data processing is strongly characterised by the “one size does not fit all” philosophy, it does not exist and will never exist a single solution that is powerful and flexible enough to satisfy the needs arising in diverse contexts and scenarios.

The tremendous velocity characterising technology evolution calls for implementing data sustainable processing solutions that are not going to require radical revision by specialists whenever the supporting technologies evolve. Whenever a new platform capable of achieving better performance than existing ones becomes available, users are enticed to move to the new platform. However, such a move does not come without pain and costs.

Data analytics tasks tend to be complex pipelines that might require combining multiple processing platforms and solutions. Exposing users to the interoperability challenges resulting from the need to integrate and combine such heterogeneous systems strongly reduce their productivity.

There is a need to develop data processing technologies that tend to solve the problem by abstracting from (and virtualising) the platform(s) that take care of executing the processing pipeline. Such technologies should go in tandem with optimisation technologies (see Optimisation) and should provide the data processing designer with fine-grained processing directives and facilities enabling to specify in detail the processing algorithm.

Relationships with requirements and use cases

Most of the RIs that participate in ENVRIplus have computer-based scientific experiments, which need to handle massive amounts of data being some of them generated every day by different sensors/instruments or observatories. In most cases, they have to handle primary data streams as well as data from institutional and global archives. Their live data flows from global and local networks of digital sensors, and streams from many other digital instruments. Often, they employ the two-stage handling of data – established initial collection with quality monitoring, then an open ended exploration of data and simulation models where researchers are responsible for the design of methods and the interpretation of results. These researchers may want to 're-cook' relevant primary data according to their own needs. Their research context has the added complexity of delivering services, such as hazard assessments and event, e.g., earthquake, detection and categorisation, which may trigger support actions for emergency responders. They therefore have the aspiration to move innovative methods into service contexts easily.

Data streaming is essential to enable users such scientists from Atmosphere, Biosphere, Marine and Solid Earth domains, to move developed methods between live and archived data applications, and to address long-term performance goals. The growing volumes of scientific data, the increased focus on data-driven science and the areal storage density doubling annually (Kryder's Law), several stress the available disk I/O – or more generally the bandwidth between RAM and external devices. This is driving increased adoption of data-streaming interconnections between workflow stages, as these avoid a write out to disk followed by reading in, or double that I/O load if files have to be moved. Therefore, data-streaming workflows are gaining more and more attention in the scientific communities.

Another aspect to be considered is that, scientific communities tend to use wide range e-Infrastructures for running their data-intensive applications, e.g., HPC clusters, supercomputers, and cloud resources. Therefore, workflow systems that are able to run them at scale on different DCIs without users making changes to their codes are currently in trend.

It is also necessary to provide facilities to run data-intensive applications across platforms on heterogeneous systems, because data can be streamed to and from several DCIs for performing various analyses. For these DCIs, it is not feasible to store all data since new data constantly arrive and consumes local store space. Therefore, after data are processed and become obsolete, they need to be removed for newly arrival data. So, data-stream workflow systems should be combined with traditional SWMS systems, which effectively coordinate multiple DCIs and provide functions like data transfers, data clean-up, data location and transfer scheduling.

All in all, the requirements for data processing are very heterogeneous, evolving and varied simply because diverse are the needs when moving across communities and practitioners. Moreover, even within the same community there are diverse actors having different perceptions, ranging from data managers that are requested to perform basic data processing tasks to (data) scientists willing to explore and analyse available data in innovative ways. When analysed from the perspective of (data) scientists the problem tends to become even more challenging because data are heterogeneous and spread across a number of diverse data sources, thus before being analysed for the sake of the scientific investigation, the data need to be acquired and "prepared" for the specific need. Steps will be needed to refine the understanding of these requirements to identify consistent and significant groups where the supplied toolkit for eInfrastructures may offer common, sharable solutions. Developing that clarity may be another focus for a think tank.

Summary of analysis highlighting implications and issues

Scientific workflows have emerged as a flexible representation to declaratively express complex applications with data and control dependences. A wide range of scientific communities are already developing and using scientific workflows to conduct their science campaigns. However, managing science workflows for synergistic distributed and extreme scale use cases is extremely challenging on several fronts workflow management system design, interaction of workflow management with OS/R and provisioning/scheduling systems, data movement and management for workflows, programming and usability, advanced models, provenance capture and validation to name a few.

A major challenge for ENVRIplus RIs applications is the integration of instruments into the scientist's workflow. Many scientists retrieve the data from a (web and/or archive) facility provided by their RIs and then realise some post analyses. Not many RIs offer the possibility to work with life data streamed directly from their instruments/sensors. Therefore, how the ICT workflows community can enable a seamless integration of live experimentation with analysis in a way that increases the overall turnaround time and improves scientific productivity can be identified as one of the mayor challenges, which involve:

- **Provisioning:** Models, algorithms, and mechanisms for resource provisioning: compute, data storage, and network. This includes open questions like How to efficiently determine the resources necessary for workflow execution over time? What information needs to be exchanged between the WMS and resource provisioning systems? How does the WMS adapt to the changes in resource availability?
- **Execution**: Examining the interplay between the WMS and system-side services (data movers, schedulers, etc.), WMS and the operating system or hardware present on the HPC platform. Issues of not only performance but also energy efficiency need to be taken into account. Support streaming data models and manage trade-offs of performance, persistence and resilience of data movements.
- **Adaptation**: Novel approaches to workflow resilience and adaptation. This includes how does the WMS discover hard and soft failures? There are several open questions that need to be addressed: Can provenance information help in detecting some of these anomalies, and the corresponding root causes? How does the WMS adapt to changes in the environment, to failures, to performance degradations? How is the resource provisioning, workflow scheduling, etc. impacted? How do we steer and reschedule workflows when there are failures?
- **Provenance**: What information needs to be collected during execution to support provisioning, execution, and adaptation. What metrics and metadata need to be in the provenance store and what should be the level of detail? Frequency of provenance information collection. Identification and interaction with all the system layers to collect the provenance data. Best strategy to store the provenance data. Development of provenance analysis models to analyse large and complex provenance information.
- **Analytical Modelling**: Exploration of more complex hardware and workflow designs, including novel memory architectures with in situ analysis and co-processing.
- **Collaboration**: another important aspect of the problem is the ability to support workflows within a scientific collaborator and related to that how to support the execution of a set of workflows (a workflow ensemble) on behalf of the user of collaboration, and how to describe and map collaboration workflows.

Besides complex scientific workflows, a lot of scientists are willing to specify their data processing algorithms by realising what falls under the "research software" umbrella. This represents a valuable research asset that is gaining momentum thanks to the open science movement. A lot of such a software is actually implemented by people having limited programming skills and computing resources. In these scenarios, environments conceived to use the software as-is and – with minor directives/annotations – enact its execution by relying on a distributed computing infrastructure are of great help [Coro 2014], e.g., this might enable the scientist to easily execute the code on a number of machines greater than the one he/she usually use, this might enable to expose the algorithm "as-a-Service" and thus to include it in scientific workflows.

Bibliography and references to sources

[Liew 2016] Chee Sun Liew, Malcolm P. Atkinson, Michelle Galea, Paul Martin,et al. Scientific Workflow Management Systems: Moving Across Paradigms, ACM Surveys, 2016.

TBC