Virtual Observatories: A Facility for Online Data Analysis

Kate Keahey
keahey@mcs.anl.gov
Argonne National Laboratory

A facility for online data analysis to support ongoing experiments and other time-critical activities has long been on the wishlist of many scientists: large experimental instruments, equipped with millions of sensors, and producing hundreds of terabytes of data per experiment will be used more efficiently if extended with a computational facility providing the scientist with ongoing insight into the data. Further, recently these sensors have left the lab and started multiplying at large: inexpensive and increasingly sophisticated sensor devices now allow scientists to instrument forests, oceans or cities turning our planet into a "laboratory at large" and providing unprecedented insight into geophysical, environmental, and social phenomena. This new development is driven by proliferation of personal sensors, marketed at large scales, and driving progress on technological (such as e.g., battery life) as well as economic factors (i.e., price) in technologies that can be leveraged for scientific exploration. Driven by emerging mass-market infrastructure, this trend is likely to continue accelerating and offering new, revolutionary, opportunities to science.

Online analysis so far has typically been addressed primarily by acquiring dedicated compute facilities to ensure on-demand availability and thus timely computation for known needs. This approach is both expensive and limited as the computation can only scale within the dedicated compute facility. Further, the online analysis needs of "laboratories at large", composed of dynamic and often ad hoc sets of sensors, are more extensive. While contained instruments typically present demanding but known requirements for online processing, a "laboratory at large" is formed of a dynamic set of sensors that can become active or inactive at different times or produce data whose interest highly depends on their values so that the computational requirements for online analysis are rarely known a priori and often fluctuate significantly. Furthermore, an experiment in a such "laboratories at large" can go on indefinitely. This creates a need for not only on-demand but also dynamic and adaptable processing.

The availability of on-demand compute resources giving the user control over the execution environment, as popularized by infrastructure cloud computing and its extensions to make it suitable for operation in scientific settings [1, 2], offers a promising solution to these challenges. However on-demand resource availability by itself will not solve the problem fully. To solve it, we need an infrastructure that will leverage on-demand availability to provide resources in a way that is scalable, sensitive to demand, i.e., capable of anticipating and reacting to new resource demands, capable of rapid change as different types of processing become interesting or uninteresting, and parsimonious to accommodate the large processing scales required by today's computing needs. To maximize impact, such facility should also integrate collaborative elements, i.e., allow users to define and publish their algorithms and codes (potentially as Data Object Identifiers (DOIs)), such that they can be (a) shared easily and by users who are not necessarily experts on their operation -- as opposed to being tied to resources on which they have been configured, and (2) easily deployed close to the data as needed rather than moving the data itself.

We propose a data-centric virtual observatory architecture designed for scalable, fast-response, and efficient data stream processing. This architecture enables collaborative computing by allowing users to publish algorithms and codes such that they can be easily integrated within the framework, dynamically activating and dismissing data streams as focus of investigation shifts.

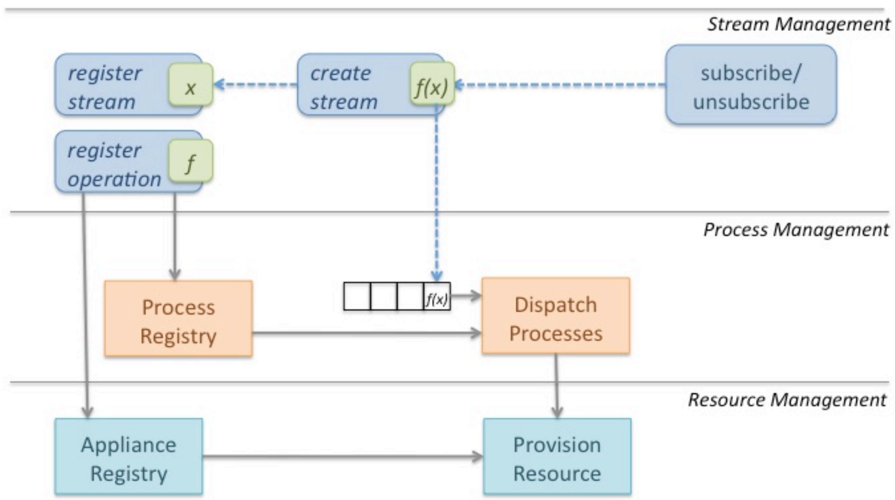

Figure 1: Architecture of the virtual observatory

A high-level conceptual diagram of the architecture is shown in Figure 1. The user can *register* an existing push or pull data stream from a variety of sources, including live experimental data or a previously archived stream. The user can also *register an operation* by (1) defining and registering an appliance, i.e., a rendition of an environment guaranteed to support the application (represented as e.g., a virtual machine) and (2) specifying and registering job/process execution parameters (such as e.g., information about the appliance required for job execution, how to run an application or transform data from the framework into input expected by the application or output going the other way). And finally the user can *create a new data stream* by applying the operation to an existing data stream. Such stream can be interacted with by subscribing or unsubscribing such that only the streams that have subscribers are allocated resources. Activating a subscription to a created stream triggers a request for the repeated execution of a corresponding operation and, if needed, a subscription to another stream or streams. The operation/process execution request causes the process manager to look up the operation's dependencies (e.g., the appliance type) and request their deployment as needed. This process can be monitored and fine-tuned on stream, process, and resource management levels to reduce response time, improve proximity of computation to data or balance overall cost as needed -- effectively transforming a logistical problem into a policy problem. A chained set of subscriptions in effect implements a workflow whose branches can be dynamically modified during the computation acquiring and releasing resources as needed.

This architecture has been implemented and deployed by projects such as Ocean Observatory Initiative using data large (e.g., satellite images) and small (e.g., salinity and temperature data) to provide on-going ocean observation and is currently extended to work with environmental sciences at ANL. While the experiences from those deployments have been generally positive they raise many challenging issues including cost-effective data consistency and distribution models models in an "elastic" resource environment, feasibility of HPC in virtualized/cloud environments, management and synchronization of networking, storage and compute resources for data-centric computations, and quality of service of the end-to-end system.

References

1. Marshall, P., K. Keahey, and T. Freeman. Improving Utilization of Infrastructure Clouds. in CCGrid 2011.

2. Sotomayor, B., K. Keahey, and I. Foster. Combining Batch Execution and Leasing Using Virtual Machines. in HPDC 2008. 2008. Boston, MA.