Rethinking streaming system construction for next-generation collaborative science

Matthew Wolf and Greg Eisenhauer
College of Computing
Georgia Institute of Technology
Atlanta, GA
{mwolf,eisen}@cc.gatech.edu

Patrick Widener
Center for Computing Research
Sandia National Laboratories Albuquerque, NM
pwidene@sandia.gov

I. Introduction: Streaming Middleware

Event- and stream-based systems for science are playing an increasingly important role. In high performance computing, the I/O bottleneck as we move into the near-exascale regime puts an increasing focus on in situ analytics pipelines, many of which share important characteristics with other streaming middleware solutions. Additionally, there is an increasing desire to dynamically process experimental and observational data streams, sometimes in concert with validation simulation runs. Altogether, this demonstrates a core need for further developments in high performance streaming services that can be shared and developed across communities.

However, streaming software systems have a mixed history within computational and experimental science. Although there are numerous examples from both high performance computing as well as experimental and observational data, most of these have been built as individual, bespoke infrastructures. One facet that we believe has helped to drive this diversity of solutions is the fact that streaming is in many cases tightly tied to some control or decision process within the experiment or run. Even within a single experiment, the scientist may need to use several different types of decision processes, each with their own time scale and impact, and this diversity of needs leads to a diversity of solutions. For instance, data quality issues (e.g. dirty images) may be addressed with one type of control loop, while data validity issues (e.g. picture of the right phenomenon) are addressed at a different scale.

Mapping streaming concepts to modern computational science involves a sort of "$M * N$" task of making multiple choices in multiple dimensions to address a given scenario. As noted above, there is a large diversity of control operations and their timeliness requirements that are needed across streaming scenarios (e.g. microsecond feedback times for dynamic load balancing). Similarly, there is a significant variety of data processing requirements so that actions are taken based on proper information (e.g. data fusion across several image streams to validate that a phenomenon is occurring). Finally, there is a wide variety in the distances and between end points, from shared memory to shared interconnect to wide area linkages.

Informed by our experience over many years in building middleware for high performance applications, we contend that efforts to build a single streaming service to meet these needs are misguided. It is not necessary, however, to reinvent the wheel for each new system or application. Instead, one should think about building a generalized toolkit – a palette of operators and high performance connections that allow one to easily build and validate your particular streaming system. Like a selection of pipettes, flasks, vials and assay reagents, the high performance streaming constructor approach aims to give the scientist tools to build and reason about the correct streaming solutions, without trying to dictate a "one-size-fits-all" approach. As a working example of the issues at hand, we consider next a case based on combustion science.

II. Motivation: Experimental Combustion Science

The typical workflow in the combustion community is highly collaborative. The collaborative structure often consists of one or more groups of experimentalists, experimental data analysts, modelers, computationalists, and project coordinators. These teams are typically composed of groups whose skills best complement each other for the problem at hand; they disband and reform with different participants as new projects arise. The scientific goals of such teams are to develop models of the combustor physics that human beings can understand, and often to construct physics-based engineering tools. These collaborations currently rely on a great deal of post-hoc processing, sneaker-net transmission of data, and grad-student mediated transfers of information. We have been evaluating how their existing tools and decision processes can be extended as appropriate, by removing "rough edges" from the streaming infrastructures and making processing decisions, control, and data management both more natural and more flexible for the scientists.

The first use case considers a collaborative program that features a stereo Particle Image Velocimetery (PIV) experiment. Velocities of calculated by injecting small particles into the flow, and then the changes in positions between closely-timed pairs of images are calculated. In a stereo PIV measurement, the flow is simultaneously imaged with two cameras instead of one. This allows the PIV algorithm to compute all three velocity vector components throughout the measurement plane. An experimental campaign usually consists of many (tens) of measurements (each consisting of

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

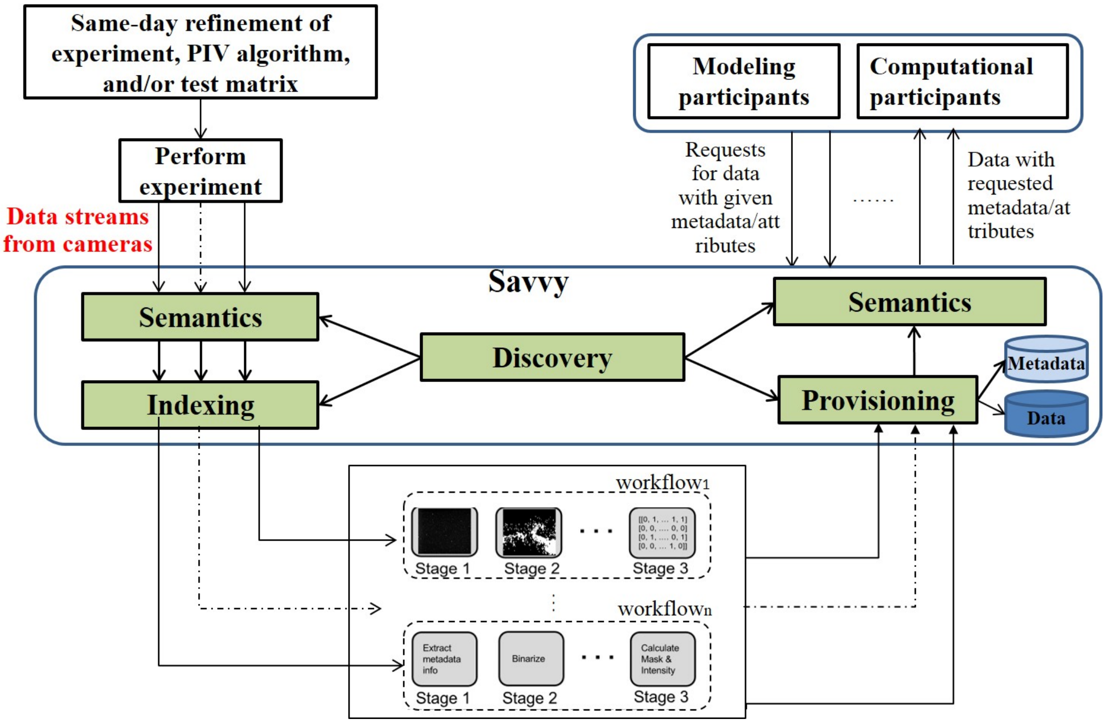

Fig. 1. Functional categories of data stream requirements for this combustion science use case. In addition to streams directly supporting computational science workflows, a streaming infrastructure must establish several publish/subscribe interfaces through which information about the service itself is communicated. A discovery stream will allow clients to subscribe and receive information about interrogate each instances manifest what indexes and what data are being managed by that instance. We envision a larger information environment where these manifests are advertised on well-known subscribable streams. As service instances receive changes to their manifests (a new data abstraction created, a new index made available or a new entry in an existing index), those changes will also be pushed to interested subscribers.

tens of thousands of digital photographs). Most experiments will implement four or more cameras, such that raw data from a 1 second measurement will occupy on the order of 100 GB of PIV and PLIF photographs. The output of the PIV algorithm is another 100 GB of velocity vector field data.

At this point, the data is algorithmically validated to check that it is a physically realistic result; if not, the parameters of the PIV algorithm must be tuned, or a higher quality set of PIV images must be captured, after which the PIV algorithm is re-run. In many labs, diagnostic equipment is shared between different groups, and is passed to the next group before the PIV algorithm is complete. If the PIV images must be re-captured, the experimentalists may have to wait several months for another turn with the diagnostic system. This iterative measurement/calculation/refinement process is repeated until it produces data of suitable quality, after which sharing and analysis of the data may begin. After the long validation/refinement stage discussed above, the data is ready for analysis. The analysis stage offers an opportunity to assess the initial design of experiments. This assessment answers questions such as ”are we exploring a range of parameters where the interesting physics occur?” Often, this analysis informs the next iteration of design of experiments, which is refined to focus on the interesting part of the parameter space. At this point, the current dataset is deemed preliminary, and the whole procedure is repeated to produce the final dataset.

Tracing through this example, therefore, are many opportunities for decisions and management of the streams of data and metadata throughout the collaboration:

- immediate control time scales, where experimentalists focus on trying to assess whether there are experimental errors, such as mistuned laser illumination or misaligned cameras;
- stream processing opportunities that might affect a parameter sweep during a day's run;
- cross-stream computations for feature extraction, to see if the evolution fits with previous runs and simulated results;
- streams between collaborators at multiple sites, requiring shared synchronization of multiple sources of data so that different expertise can be brought to bear.

III. A Way Forward

Inspired by the commonalities we have observed among these and other use cases, we seek to demonstrate that a flexible, expressive stream construction toolkit will provide significant leverage to end-users, improving individual computational and experimental data processes, making collaborations with other scientists easier to set up and maintain, and offering great potential for reuse of design metaphors, data, and code. Careful attention to the nature of the operations performed on data as well as to the structure of the control loops involved, in these and other use cases, can provide insights crucial to the adoption of a toolkit-based approach.

Some of the open questions in this space that we believe should motivate future work includes:

- Interactivity is not just human-in-the-loop. How can advanced middleware enable the delegation of control decisions?
- What is the right balance between functionality and ease-of-use for streaming solutions?
- Science occurs today, using existing ad hoc tooling. What sort of change-management is needed as processes change from a bulk processing approach to a streaming one?
- Decisions are frequently made not on timeliness of data, but on data quality. How can quality of information metrics achieve parity with quality of service ones?
- We need decision tree support for different accesses – workflow, web, storage, human, delegated/automated, etc. As the tool makers, how best to guide the science designer towards the right approach for their end goals?

Middleware to support stream computing has great potential for supporting science at the extreme scale, whether experimental, computational, or a hybrid. By rethinking current approaches to constructing the many overlapping types of streams that are needed, we hope to realize next-generation collaborative environments which support the multiple types and timescales of decisions required.