FUTURE PLATFORM WORKSHOP REPORT

1. Workshop Organization and Methodology

A facility for online data analysis to support ongoing experiments and other time-critical activities has long been on the wishlist of many sciences: 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 data. This need is becoming stronger as 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 an "instrument at large" and providing unprecedented insight into geophysical, environmental, and social phenomena. And finally, many scientific activities, such as "thought experiments", brainstorming sessions, and critical thinking have always required online data analysis support.

Recent technology trends, such as the increasing focus on data management technologies and the emergence of sustainable on-demand computing and commercial cloud facilities provide initial steps and potential building blocks for creating such compute facility. How do we fill the gap between what we have now and the capabilities we need to make the vision of an online data facility a reality? What research challenges need to be addressed – and can be addressed – in the coming 5-10 years? How should we adjust our vision if they are not solved? Is such facility compatible with the existing and evolving exascale resources and/or how should they evolve to be compatible? How do "beyond Moore's law" technologies enable advanced data analysis? Those are the challeges that the Future Platform workshop was organized to address.

The workshop was organized by Kate Keahey of Argonne National Laboratory and Jim Ahrens of Los Alamos National Laboratory and was held in the Hyatt Regency Crystal City in Arlington, VA on April 4-5, 2017.

The workshop agenda was organized around the following objectives: (1) defining requirements for "future platform" in several areas, (2) defining challenges and identifying opportunities in several technical areas of a future platform, and (3) defining the shape of a future platform, proposing ways to build it, and measuring progress around its construction. Each of those objectives was reflected in activities corresponding to a half day of the workshop. In order to define requirements for future platform, on the morning of the first day of the workshop, the attendees listened to four applications keynotes presented by speakers representing light sources, HEP, astrophysics, and smart cities, followed by a panel where attendees had the opportunity to ask questions and refine their understanding of the requirements. The afternoon of the first day was discussion on opportunities and challenges was keyed off by a round of lightning talks on proposed technical solutions, followed by two breakout sessions devoted to opportunities and challenges respectively. The morning of the second day started with two keynotes from representatives of open source (OpenStack) and industry (Amazon Web Services), followed by a panel on engagement with industry and open source community, and then by another round of breakout sessions discussing the shape of future platform and strategies to build it.

The workshop was attended by roughly ~60 attendees, most of them researchers/experts in the following seven key technical areas. To ensure comprehensive input the key technical areas were defined to cover the full stack of the future platform and included: systems, storage, networks, resource management, data, frameworks, and interaction. Attendance was partly by invitation to ensure participation of experts in the technical areas -- and partly by soliciting position statements from wider community to ensure inclusion of upcoming ideas (position papers were selected based on innovative ideas presented). We sought to ensure participation of outstanding researchers from academia as well as national laboratories.

On the first day, breakouts were organized by area (seven parallel breakouts of roughly eight participants each) as only area-specific challenges and opportunities were discussed. Each area was assigned a lead who was in charge of leading the discussion and writing the summary. The leads for the seven areas were as follows: systems (Jack Lange, University of Pittsburgh), storage (Brad Settlemyer, Los Alamos National Laboratory and Garth Gibson, Carnegie Mellon University), networks (Raj Kettimuthu, Argonne National Laboratory and Dimitrios Katramatos, Brookhaven National Laboratory), resource management (Shantenu Jha, Rutgers University), data (Kerstin Kleese van Dam, Brookhaven National Laboratory), frameworks (Laximikant Kale, University of Illinois in Urbana Champaign), and interactions (failed to converge). Their summaries of area breakouts constitute a major part of this report. On the second day another seven parallel breakouts took place, but this time they were structured such that each had at least one participant from each the seven areas. The second day breakouts discussed cross-cutting issues, specifically the shape of the future platform and strategies for its construction and evaluation. Self-elected leads were taking notes that were later summarized by workshop organizer.

All workshop presentations, list of participants, and the agenda are available at the workshop website: https://press3.mcs.anl.gov/futureplatform/. In addition, all the notes taken over the duration of the workshop are available in a generally accessible Google drive: https://drive.google.com/drive/folders/0B54oR7XAF7V9dU9mM1hyZmt4dVU.

This report is organized as follows. Section 2 summarizes data challenges and patterns related to the exemplar use cases presented during the first half day of the workshop. Section 3 summarizes the requirements presented by those use cases as well as a synthesis of those requirements into a Future Platform definition achieved on the 3rd half day of the workshop (morning of the 2rd day). Section 4 includes area breakout reports from the areas on the 2nd half day of the workshop (afternoon of the 1st day). Finally Section 5 has a report on recommendation for building methodology and metrics. The report concludes in Section 6.

2. Application Requirement Summaries

The first half day of the workshop was devoted to defining the requirements for the Future Platform. We selected four applications: two based on the input provided earlier by the workshop on experimental and observational sciences [1] and two based on more recent community work and more aggressively emerging application patterns such as might inform future features.

2.1. **HEPCloud**

The High Energy Physics (HEP) experiments are deployed on the Large Hadron Collider (LHC) are capable of recording millions of particle collisions per second; even after in-situ filtering the data produced adds up to many PBs. For the scientific impact of the experiment to be understood – and thus to make other experiments feasible – this data needs to be processed as soon as possible after it is produced. For the HEP CMS experiment, this processing is currently accomplished over a worldwide computing infrastructure including resources of Open Science Grid and Worldwide LHC Computing Grid, and consisting of a total of 150,000 cores, ~75 Petabyte of disk, and ~100 PB tape storage. Interactions between these resources are ensured by strong networks connecting individual sites, with weekly transfer volume between all sites: 4-6 Petabyte.

However, this aggregate processing power is soon likely to be insufficient. The High-Luminosity Large Hadron Collider (HL-LHC) will be a major upgrade of the current LHC, capable of increasing data production rates exponentially. It is currently estimated that with its coming online within the next decade High Energy Physics computing will need 10-100x current capacity. Thus, more computational power will be needed to process this data. While in the past

Moore's Law could be relied upon to provide increasingly more processing power as data production needs grew with successive computing infrastructure upgrades, this is no longer the case. However, additional computing power can be obtained from commercial clouds where the price of offerings continues to fall, as well as HPC infrastructure, where the number of cores per node continues to increase. Secondly, the experimental processing needs are not steady state, but come in bursts corresponding to active experiments yielding results – this means that computing power is needed only at certain times so that an arrangements where it can be provisioned elastically based on need is the most efficient for this use case.

Based on these observations, a pilot project (codenamed HEPCloud [2]) was developed that utilized resources dynamically provisioned in commercial clouds, specifically Amazon Web Services (AWS) and Google Cloud Engine (GCE) alongside local resource in Fermilab. The pilot project was significant in that demonstrated the viability of carrying out HEP analysis on this elastic testbed at large scales: hundreds of thousands of job slots in AWS and GCE with credits awarded on these systems in excess of $300K. The pilot project demonstrated the viability and efficiency of using large-scale on-demand resources (in this case available from commercial clouds – but that could also be provided by any on-demand datacenter)

2.2. **Advanced Photon Source**

The Advanced Photon Source (APS) is the Premier high-energy X-ray source in U.S providing experimental facilities to 5,700 researchers per year, in all 50 states plus Puerto Rico, 33 countries, 150 companies, and 250 universities. The instrument serves many diverse groups and projects, including condensed matter physics, chemistry, advanced materials, environmental and geo sciences, and life sciences and biology.

The APS is currently undergoing an upgrade which will increase its brightness and speed with which the images are produced by a factor of 100x or more compared to what it is today, revolutionizing scanning probe microscopies and making new insights possible. As a result APS-U will increase the data produced by 2-3 orders of magnitude. It is not only the data volume however that is increasing but the complexity of its processing: complex, multi-modal data needs advanced computation for interpretation. In addition, simulation is now increasingly required to help guide experiments along the lines of the "digital twin" model. And finally, new, increasingly diverse, user groups with new requirements now seek to make use of APS facilities.

All this defines new requirements for computational support of APS experiments. First, on-demand processing capability is needed to support ongoing experiments – the data needs to be looked at during the experiment in order to determine next steps. Due to data volume, complexity, and diversity of needs, HPC resources may be needed for this 'on demand' computation – though rough estimates of when the on-demand need may occur will be provided. Non-trivial requirements are also emerging in the ease of use and flexibility areas in particular as we try to accommodate new user demographics: Can we build modular analysis pipelines so that elements of analysis can be flexibly reused across different interactions? Can methods be developed whereby beamline and domain scientists contribute and share code effectively? Finally, there are also requirements for specific tool such as visualizing multi-modal, multi-scale data.

2.3. **Exascale Numerical Laboratories**

High Performance Computing (HPC) resources support computations in wide variety of sciences including physics, materials science, a wide range of environmental and biological computations. Simulations relevant to those sciences currently produce large amounts of data. With the advent of exascale and memory footprints projected to exceed petabytes, the output of a simulation composed of a million timesteps may easily reach thousands of exabytes. HPC thus conforms to

the pattern of a large data-producing instrument whose data producing capability is about to improve dramatically -- and thus with the corresponding challenges.

Since we cannot easily store and manage such large amounts of data, we need to develop new methods for deciding what data to keep (i.e., what data contains the most valuable/dense information). In doing so we can deploy lessons learned from LHC which represents a large data producing instrument used by multiple experiments/beamlines, each working on different challenges. Since the data produced by LHC is overwhelming, each of the experiments working with it uses in-situ hardware filters (or "triggers") to reduce this data by a factor of 10M, rejecting valid but sparsely sampled data. This process yields a more manageable dataset (which still amounts to 100s of PB). Similarly, we could treat an exascale computer running a community simulation as a data-producing instrument and organize "exascale numerical laboratories" representing research groups that "tap into" its ouput and analyze it output interactively, keeping the most interesting data. The interactive analysis could use a range of techniques such as immersive virtual sensors, posterior feature tagging and localized re-simulations, machine, joins with user derived subsets, and data driven simulations.

To enable such interactions we need to evolve the current infrastructure approach to support real-time in-situ analytics "triggers" with simple APIs to ensure community buy-in. The intent of those triggers is to downselect simulation data so that the output can be kept at petabyte scale. To ensure efficiency in as much as possible those triggers should operate on in-memory data making use of burst buffers as needed and optimizing it on a global scale. Further the operations on data objects need to be improved to support low-overhead and fast fine grain exploration of specific object features exploring localized access patterns as possible. Finally, real-time user exploration should be supported by supporting high-level scripting, simple immersive services and visualization.

2.4. **Smart Cities**

With the improvement of sensors and wireless technologies it is now possible to create large ad hoc experimental instruments (as opposed to pre-built experimental devices such as LHC) that generate data streams that can be filtered, correlated and analyzed as needed to yield answers to specific questions. For example, we can use them to investigate city infrastructure operations, explore the correlations of factors such as weather, pollution, and noise -- and then explore their effect on city inhabitants and processes as well as investigate correlations between various types of relationships, economic activities, or mobility patterns. The data sources for these investigations can range from dynamic sources such social networks, existing traffic records (e.g., taxi fleet activity), or custom installed sensors (e.g., Array of Things sensing devices) to traditional and relatively static data sources (e.g., census data). The backbone of such an ad hoc "instrument" is provided by a complex computational framework ranging from just-in-time and often in-situ computational capabilities to supercomputers.

This mode of investigation is likely to grow in scale as new sources of data emerge or are deployed to become comparable to the amounts of data produced by the existing large scientific instruments. The computational backbone of those ad hoc instruments will almost certainly put a premium on support for time-controlled execution needed to provide timely feedback for experimental strategy adaptation – but also to support function of the instrument itself.

Experiences of the existing platforms show that a significant proportion of processing will be done in-situ (e.g., on the actual deployed sensors), partly to avoid large data transfers over relatively low bandwidth networks and partly to support faster/local decision making. The development of hardware and software systems supporting processing on such "edge devices" will thus be an important research direction. At the same time, much of the data will continue to

require powerful computational facilities to process; thus algorithms and systems that explore the interplay of capabilities at those two ends of the spectrum will be increasingly needed. Important requirements in this context include support for quality of service in networking, the development of deep learning, as well a the ability to process and visualize information.

3. Defining the Future Platform

3.1. **Overall summary of requirements**

The major requirements defined in the use case discussion were as follows:

1. Virtually all of the considered use cases emphasize timeliness of response as an important feature of the future platform. Some of the motivation for this includes using (potentially complex) computations in the process of experiment convergence, decision making, or even to steer the instrument itself, support for interactive components such as visualization that ultimately aid such decision making, and the general importance of producing scientific output in a timely fashion (every experiment can be seen as part of a "series").
2. That response is increasingly hard to achieve given the increase in data production on one hand, and the end of Moore's Law on the other; scientists have thus been looking beyond local resources and exploring the use of HPC and cloud computing resources. This trend places high emphasis on obtaining such resources in a timely fashion as per #1; while this is in principle possible for cloud computing resources, historically it has been hard on HPC resources.
3. The need to go beyond local resources as noted in #2 with qualities of service that still ensure a "timely response" gives rise to two types of requirements: (a) better understanding and balancing of global data placement versus local needs, and (b) the ability to provide quality of service in an end-to-end, potentially widely-distributed, system. This in turn, requires insight into managing QoS for multiple qualities including nodes, networking, and storage and managing system-wide variability.
4. Many of the communities noted the need to respond to new user demographics and the associated ease-of-use requirements, the need to work collaboratively via systems that are designed to easily integrate the contributions of others in the community, and the need to make the process in which data is produced more transparent as well as improve its reproducibility to aid in high-level understanding of data.

3.2. **Future Platform**

The overall shape of the future platform that emerged from the discussions was of a distributed environment that enables adaptive allocation and integration of compute, storage and network resources to support complex applications. Most breakout groups noted that different use cases will require different solutions and thus some problems may in practice require subsets of such "future platform" – nevertheless the general solution will require them all.

The features of the overall composite platform were seen to be as follows:

- Adaptive and flexible -- to applications, workloads, and communities. Given the need for timely response times in experimental and observational sciences (#1) and also the need for flexibly composing applications from general-purpose components required to support new user demographics as well as ease of use and contribution (#4), optimizing special-purpose applications for specific architectures is no longer a cost-effective approach. Instead, future facilities should strive to provide mechanisms whereby a reasonable resource offering can be tailored to an application need (this is e.g., the case

for HEPCloud where a resource offering is elastically expanded as needed to provide sufficient capacity)

- Programmable – to provide this adaptive quality the future platform will need to be able to provide mechanisms whereby quantities of resources, defined by suitable qualities of service, in particular timeliness (#1), will be combined to provide an end-to-end solution (#3).
- Defining and managing required and achievable qualities of service will be the basic building block and the most challenging element of the future platform. This may include introducing mechanisms for time-controlled execution and environment isolation in HPC datacenters (#2), providing reservations for different types of compute, storage, and network resources, managing performance isolation, and otherwise ensuring end-to-end quality of service (#3).
- In the most general case the future platform will be increasingly distributed (hybrid datacenter, superfacility), connecting instruments and resources from mid to high-range and from academic to commercial (#2). This is both a reflection of increasingly distributed applications relying on a range of resources from small in-situ processors to large datacenters (e.g., combining information from sensors in Smart Cities applications) and a continuing trend towards achieving economies of scale via globally sharing resources.
- Integrating mechanisms fostering reuse and repeatability is an increasingly important concern, both from the perspective of avoiding multiplying data and as a good academic practice (#4).

4. Future Platform Components

4.1. **Systems Report Summary**

The principal requirement in the systems area is the need to provide increased usability to support different classes of users, each with different tool requirements and environment expectations as brought up by the application keynote speakers. Thus a key capability for a future platform would be the ability to easily deploy custom environments that contain the tools and libraries that a given user would like to have available. This is not quite feasible on current platforms given the need for tools to be preinstalled by operations staff, which has been shown to not scale with the expanding set of available tools and libraries.

The mechanisms to enable this are already well established: containers and virtual machines. Both approaches allow varying degrees of customization as well as performance profiles. We felt that both technologies have a role to play in the system software stack for future platforms. The underlying technologies for both of these approaches are available and becoming increasingly capable, however we identified a number of key shortcomings that still need to be addressed by both containers and virtual machines.

First, file system access is currently still an unsolved problem. The primary challenge here is the fact that both containers and VMs allow the user to grant themselves root privileges inside the container/VM context. This prevents a global file system from being directly mapped into the environment, because users would easily be able to bypass file system access controls. The current solution is to operate on an in memory file system that is stored as an opaque set of files in the global FS, and preventing direct FS access from a container/VM. This configuration prevents cross user data sharing as well as optimized FS operations due to the loss of semantic information due to the higher level of abstraction.

Second, future hardware environments are showing much more diverse and heterogeneous hardware architectures. These differences in the underlying hardware often require the use of customized libraries and system software configurations for each individual system. Currently, there is no good way to provide a "universal" image inside either a VM or container that can easily be redeployed to a different system without significant hardware compatibility issues. A number of approaches to handling this problem are underway, but they (1) are still new and somewhat unproven and (2) do not completely solve the problem in its entirety. Some examples include the use of the Spack package manager to automatically and easily generate environment images that contain the requisite hardware support libraries, OpenACC allows targeting a set of specific hardware platforms, and the Chapel runtime that seeks to provide automatic targeting of diverse hardware at the language runtime.

One major point of discussion was the underlying system philosophy should be, and in particular what sort of service model should the system export. For instance, should the platform resemble existing HPC capacity systems (batch scheduled job queues), cloud service based architectures (custom computing environments available on a tightly integrated centrally managed system), grid systems (systems that are loosely coupled and managed independently), or possibly even a capability supercomputing platform. Even the small set of applications presented in the keynotes, all had different system requirements that would be addressed by different platform architectures. For instance, the LHC is probably best addressed via a set of grid systems, the relatively localized national lab experiments (such as the photon sources) are better suited to local capacity systems or even dedicated clusters, while the big data analytics workloads are almost certainly better served via cloud systems.

4.2. **Storage Breakout Summary**

The Storage Area has identified several gaps in the manner in which storage systems will be deployed for future online analysis platforms and in how scientists will access future storage systems.

In terms of the deployment of future storage systems as storage experts we have a good understanding of the increased performance available with weaker consistency semantics, and the drawbacks of existing POSIX file systems. However, it is still unclear exactly which weaker semantics are most useful to scientific data analysis -- especially in combination with new emerging storage medias. While file systems researchers are typically able to achieve high degrees of performance with storage systems that provide very little coherence and consistency for both data and namespaces, it is still unclear to what degree scientific workloads can coexist and benefit from these weaker semantics. In support of these weaker storage semantics and emerging media, we recognize that storage extensions for advanced parallel programming models are often required. We also have identified opportunities for advancement in the scheduling of shared storage resources. In particular, understanding how to stage data into and out of the hierarchy of storage tiers.

Significant research opportunities exist in tailoring storage systems to science use cases. While HDF5, NetCDF, and ADIOS have established themselves as viable scientific data file formats, tools for easily describing and searching scientific provenance data are still in their infancy. Research into storage software and toolsets that improve the management of provenance data would significantly improve the ability of scientists to validate and reproduce results. In particular, we have identified that storing provenance information in a queryable format, and tools for querying provenance are in need of basic research.

It is also clear that significant efforts are required in improving scientist's understanding of storage system access. In particular, we have identified that user's need to understand the storage

system performance of their codes over time, and how their storage system performance compares to the other users at facilities. While libraries for characterizing performance for each job exist, efforts that collate those results and describe performance in relative terms are not currently available. This is especially useful due to the challenging nature of data analysis access patterns.

4.3. **Network Breakout Summary**

The Network Area has identified a number of issues to be addressed in the development of the Future Platform in the area of networks. As essential for the establishment of stability in the system, one needs to define a number of feedback loops, mechanisms to provide information about the operation of key system aspects. Such mechanisms should, for example, expose the implications of utilizing Quality of Service (QoS) and the consequences of its abuse or failure to provide it; indicate the easiness of discovering services and their APIs; characterize policies for authentication, authorization, auditing, and resource sharing; provide monitoring of operations, provider feedback with regard to participation costs, and user feedback with regard to expectations met. Overall, we believe it is critical to monitor the "Quality of Experience" (QoE), i.e., human ratings for the experience that the system provided to users, which can be used to influence future (possibly automated) decisions and choices – as well as the QoS.

Several other questions offer multiple research opportunities in the networking area. For example, how can one gain access to resources owned by multiple different parties/domains? What is the least amount of information required from users to service them? Can we provide alternatives if a request cannot be met? A problem that has been around for years, the "last mile" problem is something that must be addressed if high-performance network resources are to be accessed efficiently and effectively. It is unlikely that such resources can be utilized by default all the way to the actual data source or destination. While the Science DMZ was invented as a solution to this problem, one has still to respect all local policies and restricting to get their data to or from the Science DMZ. Finally, one must take into account the heterogeneity of capabilities across network domains. For example, provisioning of resources may not be supported along the entirety of a network path, which would call for adaptive solutions to maintain QoS priorities.

A major question in the development of new features and capabilities of the platform is how to proceed towards more intelligent networking. Is there a common language between networking and computation? For example, would it be possible to smartly compute while in transit to save cycles from the end sites? Clearly, it will be necessary to consider what lies at the end of the network, such as Internet of Things (IoT) devices, and/or applications, such as simulation or visualization, and follow a co-design process. Concluding, it is abundantly clear that scientists (users) need to be educated to have at least minimal network awareness. Better understanding the involvement of the network in getting data from A to B by time T will help in easing the experience. Also helpful will be having methods to translate user feedback, expressed using a QoE vocabulary, into resource provisioning plans.

4.4. **Resource Management Breakout Summary**

Resource management was both a specific topic of discussion as well as an undercurrent across many of the tables and topics. The discussion, analysis and resource management requirements of future online analytic platforms are best presented along conceptual and a practical dimensions.

Conceptual: Science is increasingly distributed, data sources are often not where compute is, and sometime are in many places; resources available to a scientist or project are also often distributed. Federation of compute and data resources across different scales and levels is needed. For example, future platforms/supercomputers must be part of a distributed whole. Dynamism will be a fundamental property of distributed resources at increasing scales. This is true for

multiple distributed resources and large-scale single resources with data. Without fundamental advances that address heterogeneity and dynamism, we will remain condemned to point and non-extensible solutions for distributed applications and systems. One of the primary drivers for distributed and dynamic resource management is the desire to overcome the limitations of a single resource, which in turn is driven by the need to overcome scale limits of a single resource, or overcome the functional limitations of a single platform, and thereby aggregate multiple heterogeneous platforms. However, the need for resource federation is often where the agreement ends, for there is little agreement on how to federate resources either qualitatively and quantitatively. This is partly because there are several considerations that need to be distinctly evaluated, as opposed to being lumped into a single large problem. Resource federation requires several smaller problems to be addressed: these range from multi-level scheduling, managing heterogeneous workload distribution, data-compute co-location and movement, logical resource overlays and obviously scheduling policies (providers) and incentives (users). A discussion of these problems, with an emphasis on the the models and methods of resource federation is necessary in order to advance the principles and practice of resource federation which is often mired in which software to use.

Practical: An important recurring topic related to resource management is the need to move away from the static, if not rigid resource management that existing high-performance computing platforms demand. The need for more agile resource management capabilities on future platforms is driven by the need to support emerging class of applications (e.g., streaming applications), co-scheduling coupled by distinct applications, as well as large-scale workflows. The need for more agile resource management in high-performance computers has been varyingly referred to as elastic computing, on-demand computing, bursty computing, scale-out etc. The workshop participants carefully distinguished these similar but distinct terms that are interchangeably used. (Detailed definitions and distinctions can be found in the notes/proceedings). Federation of resources as discussed above, is an important precondition to support bursty computing across physical resource boundaries. Independent of the specific solutions, there was agreement across participants that greater flexibility is required to support the range of use cases and advanced applications, and that any practical solution must involve resource providers, as a fresh approach to resource utilization policies is required to manage the tension between on-demand and batch models of computing. Technological advances -- ranging from containers and software-defined systems provide new opportunities to support flexible resource management on future platforms. Middleware building blocks that utilize these advances to provide advanced resource management capabilities, and that are composable and extensible to provide specific frameworks are needed.

4.5. **Data Breakout Summary**

The Data Area identified the following challenges requiring research investment:

White box machine learning - how do we make it understandable and verifiable what goes on in a machine learning applications - new explanatory model, a new language to describe the analysis in a mathematical, testable and reproducible form - necessary to build trust in the methods we use to discard data permanently.

New Data Representations - to combat cognitive overload from data complexity - Combine data compression, feature detection, multiscale, spatio-temporal (temporal ripe for development in terms of compression), knowledge extraction - interaction with domain scientists - their knowledge.

New interaction/engagement paradigms - what is the most effective way for users to engage with large, complex results either for decision making or exploratory purposes. Are multi-scale

representations enough or should we force incremental data representation, rather than all data in one, would the inclusion of other senses help or increase the cognitive overload? How can humans and computer better collaborate - perception models

Support hypothesis creation - between human and computer driven by the data and analytical methods

Decision making environment versus scientific discovery environment - what are the characteristics needed by either, which ones are in common, which ones have to be different.

New data products that include reproducible analysis etc. If there are features missed in this data, either there is enough data to do more analysis or the data needs to be regenerated with new expectations. I can then give you my data product with included analysis and you can see the results I get, check its correct, use it to do further analysis, or decide the analysis is wrong and create an entirely new data product (with analysis)

4.6. **Frameworks Breakout Report**

One striking aspect of online analysis applications of the future is their wide diversity. Data sources might be real-time, external to the computer processing the data – or, the data might arise from an ongoing massive parallel simulation. The relentless shower of data generates a deluge that must be processed before it washes away. The processing itself may range from simple sampling and accumulation to most complex machine learning algorithms. Data generated during a simulation, being processed concurrently with it, may free you from the real-time constraints, but engenders newer challenges of processing data without slowing down the underlying simulation. Most significantly, different application domains generate a diversity of online analysis tasks.

In this context, designing and building a new system from scratch for each online analysis application will be a daunting task. We must develop frameworks that provide commonly needed support and simplify creation of new applications.

Some of the technology drivers for frameworks include the sophistication and complexity of emerging hardware in HPC platforms. The processors, accelerators, and FPGAs create an evolving computational landscape. Static and dynamic variability in speeds and capabilities of computational cores is a challenge that must be increasingly dealt with. Similarly, there is a wide variety of memory technologies, including NVM's, burst buffers, and scratch pads at different levels of the hierarchy.

There is a need for different advanced programming models to express individual analysis tasks, as well as the overall workflow. Some of the existing programming models developed in the HPC context have characteristics that are of use for this purpose. For example, Charm++'s interacting-objects model allows its runtime system to shrink and expand the sets of processors used by a job. Such programming models need to be developed further in the context of online analysis frameworks.

Job schedulers, which operate at the level of the whole machine, must be aware of workflows, and data sources. Further they must take advantage of the shrink-expand capabilities provided by adaptive run-time systems for individual jobs.

Challenges for development of future online analysis frameworks involve sociological challenges, programming model challenges and whole machine optimization challenges.

Sociological changes have to do with resistance to adoption of novel frameworks and programming models developed for supporting online analysis. In recent years, the HPC community, faced with its own challenges of sophisticated CSE applications and increasingly

complex hardware, has shown some willingness to adoption of newer programming models. Will the online analysis community be able to develop newer frameworks of demonstrable utility within this window of opportunity? Will the application community adopt such frameworks? What is needed to generate a sense of ownership towards frameworks by application developers? Another challenge is quantifiable metrics for user productivity, in order to ensure that frameworks research is accountable and then to demonstrate to the user community the benefits and utility of individual frameworks. This may help influence the sociological factors positively. At the same time, it must be noted that practical use-cases that showcase successful applications developed using frameworks go a long way in attracting application developers to the frameworks.

Some of the challenges for programming models include: will MPI, which has served the HPC community well, evolve sufficiently rapidly to allow easy expression of online analysis tasks? Alternatively, will new programming models designed specifically for online analysis be able to interoperate efficiently with MPI? Are task-based or data-driven programming models of today adequate with small extensions for online analysis, or will we require brand-new programming models?

Elastic control and management of resources at the level of individual jobs, elements of workflows, and the entire machine presents a new set of challenges. Frameworks may exist at two or three levels: whole-machine frameworks (which generalize job scheduler/resource manager/provisioning), per-workflow and per-job frameworks. The latter 2 must allow for elasticity of resources, and there should be bi-directional communication between the whole-machine frameworks and the latter two, so jobs/workflows can request more/fewer resources , and machine-level frameworks can grant or withdraw resources based on global demand and global optimization. Synergistic development of frameworks at these multiple levels is a key challenge here.

5. Building the Future Platform

5.1. **Development Strategies**

The consensus that emerged from the second day breakouts was that building the future platform should proceed in the following stages:

Stage 1: Exemplars, demonstrators, etc., i.e., focus on the development of applications that provide concrete instantiations of Future Platform features. This stage is essential to provide a refinement of the understanding of the requirements of data applications. This understanding is already underway with large-scale demonstrations such as the HEPCloud proof of concept already ongoing. An important part of this phase would thus be to understand and digest the ongoing work.

Stage 2: Pilots, i.e., building small-scale but generalized implementations (with onboarding process) that support a selected set of “future platform” features. The intent of pilots is to attempt to build solutions that will serve more than a single community to assess how features of demonstrators generalize over different sets of requirements, i.e., are they specific to one community or do they (currently) capture a cross-community underserved need.

Stage 3: Building infrastructure/solutions -- following up on pilots to build specific production-quality solutions serving broad needs. This is the stage at which involvement of open source community and industry becomes important.

Many groups in their presentation noted the lack of suitable experimental and gradual adoption platforms. This is particularly acute in technologies that are both new and require a significant investment for a viable platform, such as e.g., networking though it has been addressed to a certain extent by the existing Computer Science testbeds such as ESnet, GENI, or Chameleon – to

be effective they need to be more accessible. A significant factor emerging from the discussions was the importance of creating incentives – in resource management as well as in adoption.

5.2. **Milestones and Metrics**

The objective of the "Milestones and Metrics" breakout was to suggest milestones and metrics that can be used to evaluate progress in building the Future Platform.

The consensus was that different metrics apply at different stages of the Future Platform development. While the ultimate goal of scientific output may be publications and discoveries made using specific tools, in the short term other metrics may be used to measure incremental progress. The specific metric suggested were as follows:

Adoption (of intermediate or complete solutions): number of users, number of distinct domains (using a specific solution), number of cooperating facilities, geographic breadth, etc.

Level of integration: How many different sub-communities/silos have formed? (this is an inverse measure of progress)

Rate of adoption: What is the difficulty of migration to next use case? What is the rate of uptake by new users and communities? How long/difficult is onboarding?

6. Conclusions

The big data phenomenon is driven primarily by improvements to experimental devices and instruments (including supercomputers) and the emergence of new ones – and thus affecting primarily the experimental and observational communities. While the need to manage large data volumes better is of course a straightforward response to this challenge, a more complex and urgent need is represented by the requirement to support of the *computing patterns* required by those communities. These include the need for a timely response required for decision-making in operating instruments, converging experiments more efficiently, and leveraging efficiencies (such as operating on raw data or data in memory) that are not available outside of a narrow time window. When combined with the need to reach beyond local resources this translates into the ability to provide quality of service of individual resources – such as compute, storage and networking – as well as orchestrate the availability of these resources to provide end-to-end qualities of service.

In addition, the vital concerns of ease of use essential for the support of new user demographics, support for collaboration in the sense of providing a vehicle where multiple users can contribute data processing operations, and better management of scientific data as regards reproducibility have been brought up. While these are not specific to the "Big Data" phenomenon, addressing them will certainly make it easier to deal with them and foster productive scientific practices.

While these requirements are not new, two factors contribute to making them particularly important at the present time. First, with the growing data – and thus opportunity – the needs of experimental and observational sciences grew beyond the point where they could be satisfied by dedicated local resources. Second, with the advent of cloud computing and associated technologies, software-defined networking, and opportunities in both storage hardware and consistency models there is now significant technological potential that we can leverage. We thus have both the need and the opportunity to address for science.

The topics of investigation that emerged as particularly urgent in the context of supporting new programming patterns are understanding and resolving challenges associated with broadly defined "containers" (VMs, or container technologies) as they can provide the critical ease-of-use portability and packaging platform, exploring new -- both hardware and software (relaxed consistency, burst buffers) -- opportunities in storage, leveraging the SDN opportunity into

closing the gap between the requirements and solutions in data movement, and integrating the new resource management solutions introduced by the cloud computing innovation. In addition, new data management methods as detailed in Section 4 need to be addressed. Derivative challenges such as offering users viable programming models in the changed environment and methods allowing them to combat the complexity of dealing with a "programmable platform" also need to be addressed.

Discussions on the approach to build the platform highlighted the need for smaller and diverse pilot projects (which have already emerged) that will gradually adopt common mechanisms and tools. Two specific challenges to making progress were highlighted: the lack of generally available testbeds and the lack of incentives. The former is partially overcome by the availability of testbeds such as ESnet, GENI, or Chameleon (each with different policies of use and a different set of supported experiments however), and frequently the lack of incentives for a community to e.g., try a different programming model or tool whose advantages sometimes may not be immediately obvious. Creating such testbeds, adoption pathways, and general incentives where appropriate is difficult – but would catalyze progress.

Last but not least a lasting benefit of the workshop was as a "socializing venue" of the application scientists, Computer Science practitioners, open source and industry and created lasting connections between some of those groups.

7. References

[1] *Report of the DOE Workshop on Management, Analysis, and Visualization of Experimental and Observational Data*. LBNL-1005155. Bethesda, Maryland. September 29–October 1, 2015

[2] *HEPCloud, a New Paradigm for HEP Facilities: CMS Amazon Web Services Investigation*. Burt Holzman *et al. Computing and Software for Big Science* **1.** 2017