Appendix D

Individual Reports from the 2009–2010 DOE Systems Biology Knowledgebase Workshops

DOE Workshop on Cloud Computing in Systems and Computational Biology: Workshop Report

November 16, 2009
Portland, Oregon

Convened by

The U.S. Department of Energy Office of Science
Office of Biological and Environmental Research

Organizers: Folker Meyer (Argonne National Laboratory), Susan Gregurick (U.S. Department of Energy), Peg Folta (Lawrence Livermore National Laboratory), Bob Cottingham (Oak Ridge National Laboratory), and Elizabeth Glass (Argonne National Laboratory)

Audience: 130+ people

Speakers: Folker Meyer (Argonne National Laboratory), Dawn Field (Oxford, UK), Eugene Kolker (Seattle's Children Hospital), David Haussler (University of California, Santa Cruz), Simon Twigger (Medical College of Wisconsin), Ananth Kalyanaraman (Washington State University), Michael Schatz (University of Maryland), Sam Angiuoli (University of Maryland), Narayan Desai (Argonne National Laboratory), Lavanya Ramakrishnan (National Energy Research Scientific Computing Center), Kate Keahey (Argonne National Laboratory), Bob Grossman (University of Illinois at Chicago), Judy Qiu, (Indiana University), Thomas Brettin (Oak Ridge National Laboratory), Owen White (University of Maryland), and Deepak Singh (Amazon)

Panelists: Susan Gregurick (DOE), Owen White (University of Maryland), Pete Beckman (Argonne National Laboratory), Kathy Yelick (Lawrence Berkeley National Laboratory), Dawn Field (Oxford Centre for Ecology and Hydrology), David Haussler (University of California, Santa Cruz), Jeff Grethe (University of California, San Diego), Folker Meyer (Argonne National Laboratory), Victor Markowitz (DOE Joint Genome Institute), Eugene Kolker (Seattle Children's Hospital), Bob Cottingham (Oak Ridge National Laboratory)

Introduction

According to a recent National Institute of Standards and Technology document, cloud computing is a model for enabling convenient, on-demand, network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction ("The NIST Definition of Cloud Computing," v15). This new approach is an evolving paradigm for providing and using computational services. Previously, scientists could either build local computing resources, where all aspects of the system hardware and software could be tuned to their application, or they could adapt their application to pre-existing computing resources. There was no middle ground between these two options. Now, with cloud computing, through the use of virtual machine images (VMs), scientists can carry a full computing environment with their application to new systems, in a simple and portable fashion. A cloud platform provisions both the hardware platforms and a means to install and run a given environment stored in a virtual machine (VM).

Unlike Grid or high performance computing, cloud architectures can have little to no centralized infrastructure and up to now have mostly been characterized as a third party computing service which is rendered under a utility model or a ‘pay as you compute’ model. The virtue of this model is that the end user does not necessarily have to invest in computational hardware, software, or administration to enable their particular application or science. However, the drawbacks include latency in data transferred to and between a cloud system, potential data security issues, and system resilience. Moreover, users need to learn how to best make use of cloud interfaces and capabilities.

Nevertheless, the ‘compute as you go’ properties that are inherent to cloud computing make this an interesting platform that may be well suited for the computational needs of the systems biology research community. While other communities have shared data, the large volume of data shared within those communities typically comes from a small number of data sources. In the ‘Omics’ disciplines of systems biology such as genomics, transcriptomics, proteomics, etc., data generated using a variety of instruments (DNA sequencers, mass spectrometers, micro array readers, etc.) are shared with the community at large and are used as the basis for a variety of integrated research projects, frequently involving large computations. However, the sharing process is not yet very effective, and as more data generators and data consumers enter this arena, more efficient ways of data sharing will be required. The scale of the deployments of next-generation instruments in the biology disciplines mirrors the number of research laboratories working in the Omics disciplines. The result is a striking democratization of Omics data generation and the subsequent need for collaborative research. As a consequence, many high volume data sources exist in biology and with the data volume rising exponentially, a platform that provides low-cost, flexible computational services, like that of the cloud, could be a good match for the needs of the systems biology community.

This community is characterized by its growing creation and consumption of data as well as the inherent rise in computational demands. Two prominent features attracting solution providers in bioinformatics, computational biology, and systems biology to “the cloud” are the “seemingly endless supply of cycles” (P. Beckman) and the ability to “bring your own environment” (O. White).

Cloud Computing for the DOE Systems Biology Knowledgebase

The DOE Genomic Science program supports systems biology research to ultimately achieve a predictive understanding of microbial and plant systems for advancing DOE missions such as sustainably producing biofuels, investigating biological controls on carbon cycling, and cleaning up contaminated environments. To manage and effectively use the exponentially increasing volume and diversity of data resulting from its projects, the Genomic Science program is developing the DOE Systems Biology Knowledgebase (genomicscience.energy.gov/compbio/). Envisioned as an open cyberinfrastructure to integrate systems biology data, analytical software, and computational modeling tools that will be freely available to the scientific community, the Knowledgebase will drive two classes of work: (1) experimental design and (2) modeling and simulation.

A cloud-based computational platform presents a promising opportunity for the DOE Systems Biology Knowledgebase (Kbase). The objective of this workshop was to elicit community input on the feasibility of using the cloud paradigm as a component of Kbase. Specifically, the workshop participants evaluated the requirements for a cloud-enabled Kbase and presented ideas for engaging the High Performance Computing (HPC) community in this effort.

The three charges for this workshop were:

1. What are the characteristics of applications that would be appropriate for effective utilization of cloud architecture?
2. What are the hardware bottlenecks that prohibit cloud architectures from being easily adopted by high-throughput biological data analytics?
3. What are specific tools that need to be developed or enhanced in order to make cloud architectures easily adopted for biological data and bioinformatics algorithms?

This workshop brought together more than 130 computer scientists, bioinformaticists, and computational biologists to discuss the feasibility of using cloud computing for DOE's systems biology Kbase. This workshop was held in conjunction with Supercomputing '09 in Portland Oregon. The one-day workshop consisted of leaders in these fields presenting work in cloud computing for biological research as well as a two-hour round table discussion centered on the charge questions**.**

The recommendations from this workshop are summarized below:

1. Switching to clouds will require re-engineering, at first for scalability and then for fault tolerance.
2. A healthy research community requires open source reference implementations of algorithms and pipelines. While many algorithms are open source, the pipelines tend to be closed source due to tight integration with locally existing environments. Clouds provide a unique opportunity to open up existing pipelines to more scientific scrutiny.
3. There is a need for standards in computational workflows and data, including the sharing of intermediate computational results.
4. There is no consensus for the software appliance operational model in bioinformatics (as demonstrated by an informal vote on preferences to "bring your own VM" vs. "use a provided VM").
5. Wide area data movement is still an active issue in the bioinformatics community; these problems are amplified when dealing with clouds.

6. There is a need for a system enabling the flow of data across different VMs instantiated in a cloud, potentially for VMs from different appliance providers.

Clients for the Kbase Computing Platform

We discussed three types of potential users who would benefit from the Kbase computing platform: computational biologists, bench biologists, and integrators who aim to integrate smaller components into advanced workflows.

From an architectural perspective it is interesting to consider for whom the Kbase computing platform should be tailored. While any platform can support multiple types of users, having an understanding of which users one intends to support the most, or with highest priority, will influence the ultimate design of the hardware platform. We anticipate computational biologists and bioinformatics users being advanced users insofar as they would require more access and more control of their resources hosted by the computing platform.

Bench biologists, on the other hand, would be users of a wider variety of readily available third-party software applications, and therefore the ability to support key third-party applications would influence the architecture.

Need for Reference Data and Standards

Standards are a mechanism for capturing information in a form easily shared and integrated with other data or data types. Using data standards to capture data entities is the foundation for comparative analysis and integration of information.

Increases in biological data production are dramatic. For example, DNA sequencing throughput has grown from 100-500 megabytes per run to 30-90 gigabytes per run in a 12-18 month time frame. As an increasing number of diverse data products (e.g., genome annotation, protein families, pathways, etc.) are transformed and consumed by many different parties, there is a strong need for versioning and controlling production of reference datasets and provenance information. The choice for data transformation also is important for downstream processing (e.g., missing genes impact metabolic pathway predictions, false gene starts impact protein family curation). There is limited sharing of results—especially for compute intensive intermediate results—because data formats have been designed for different purposes. Since using a few centralized, closed pipelines can not meet the communities' dynamic data and information needs, rapidly evolving data exchange standards and data provenance descriptions will be required.

Use of Workflows

The concept of "abstract workflows" seems to be the right level of abstraction for most bioinformatics and computational biology groups. Most HPC providers are focused on parallel implementation of workflows.

In bioinformatics, large-scale computations are often encoded in scripts that wrap and automate the use of standard tools and methods, typically making the pipelines opaque and hard to port between computational platforms. The international Genome Standards

Consortium (GSC) has promoted rich, transparent descriptions of analysis pipelines, such as Standard Operating Procedures (SOPs), which can be published in academic journals and help promote best practices. While these SOPs are fine for describing a process to colleagues, they are not readily interpretable by a computer and are often not rich enough to ensure that results are reproducible at another institution. As a complement to human-readable SOPs, the GSC and its M5 metagenomics working group seek a workflow description language for describing bioinformatics pipelines that can be executed on large computational resources. The format, such as an XML, should be exchangeable between institutes, platform independent, and compatible with existing workflow systems. Of particular interest is portability across grid and cloud architectures, as groups interested in running workflows may have access to a variety of resources. An ideal format would be relatively high level and would describe the process in terms of standard tools or algorithms, such as BLAST. An ideal format would also hide certain complexity required to improve performance, such as partitioning data into batches for batch processing, as these steps may significantly impact the process flow but not the analysis algorithm (such as an readily parallelizable search like BLAST). Also, ideally, such a format already exists within the workflow community and can be adopted and promoted for use by the GSC and M5, although it is not immediately clear which, if any, existing workflow descriptions fulfill the goals outlined here. We also highlight that in cloud computing environments, there are unique opportunities to integrate analysis pipelines and data as part of a single shared resource, the “cloud.”

Reference Virtual Machine Images

First introduced in the 1960s on IBM’s 370 platform, virtual machines (VMs) enable running multiple operating systems on the same hardware platform using a hardware abstraction layer (nowadays referred to as “hypervisor”). Using VM technology, the process of loading an operating system environment becomes similar to loading an application in today’s desktop environments. The VM-image is a single file not dissimilar to existing application programs (e.g., GNU Emacs).

The DOE FastOS program (run out of the DOE Office of Advanced Scientific Computing Research) has clearly demonstrated the value of OS customization for application performance. Clouds (VM instances in particular) provide an unprecedented opportunity to tailor runtime systems to bioinformatics applications. Thus, it seems clear that the ability to run arbitrary VMs is going to be a key component of the Kbase systems architecture.

The creation and maintenance of a set of reference VMs can add significant value, enabling a large number of bioinformatics scientists. A “VM marketplace” with open, maintained reference images and the ability for individual developers to add functions and/or programs to those images will be an important component.

The team providing the computational platform (cloud) should be charged with providing a set of reference VMs and maintaining them over time. Outside developers should be enabled to generate “appliances” using the VMs provided (or using user-provided VMs). The use of a configuration system (e.g., BCFG) should be encouraged to separate images (provided by the resource provider) and the semantics of value provided by third parties.

The provision of reference VMs as an open-maintenance model and a configuration management tool to enhance the VMs was addressed in a discussion about infrastructure as a service (IAAS) vs. software as a service (SAAS). It seems clear that a core team should provide infrastructure as a service enabling a distributed team to provide software services.

Evaluation of Use Paradigms for Clouds

Despite the availability of cloud computing IAAS and SAAS offerings over the last few years, adoption in the computational science community in general and bioinformatics community in particular has been slow. Recently, groups (several of whom presented at the workshop) have begun to evaluate and use cloud resources as a part of their computational platform. Cloud resources provide a set of capabilities and operational properties that are distinct from traditional computational resources, both in terms of computation as well as data storage. Effective use of these resources will likely require explicit adaptation of applications, use policies, and operational practices in order to accommodate these differences.

The key challenges posed by cloud resources are scalability, fault tolerance, lack of locality, and pricing. One of the primary promises of clouds is elastic, on demand scaling. In order to take advantage of this capability, pipelines need to eliminate scalability bottlenecks and support the dynamic addition and release of resources. With the addition of scalable computational resources, fault tolerance quickly becomes an issue. Pipelines must be able to cope with frequent resource failures in a transparent and robust fashion.

Bioinformatics applications tend to be quite I/O intensive. This will be a key challenge with cloud systems, where locality and network topology are not well exposed. Traditional network file systems are not well positioned to solve this problem, as they are usually deployed in fixed configurations and cannot easily be migrated to follow compute resources as they move around inside the cloud. Data aware programming models (like MapReduce/Hadoop) are capable of solving this problem, but require substantial reworking of analysis pipelines.

The final challenge is pricing. Commercial clouds provide compelling services, but are not optimized for computationally bound workloads. Cloud pricing models include raw node hours consumed, as well as data transfer, storage, and use forecasting. These must be taken into consideration when deciding which tasks to run on commercial clouds instead of local resources. This issue may also drive adoption of cloud approaches (such as Eucalyptus, Nimbus, or Hadoop) on local resources.

Several approaches to adapt pipelines to cloud resources were presented at the workshop. Each of these demonstrated pros and cons of particular approaches, and none is likely to represent eventual production cloud architectures. Several groups demonstrated architectures that extended the local computing infrastructure directly into clouds without modification. This has the benefit of being relatively straightforward to implement, but suffers from a number of potential security issues, when exposing local infrastructure to cloud resources. The lack of clear locality information makes this approach suboptimal for highly I/O intensive analysis techniques. Another approach was to build a work management system (AWE) that has knowledge of the semantics of work units. This allows the re-use of intermediate results (that have already been computed), as well as the optimization of task placement based on data requirements. This approach has been more labor intensive than the previous approach, however, it should provide better scalability, security, and a better fault tolerance model. A third approach was to completely adapt applications to the Hadoop/MapReduce programming model. This programming model has explicit support for data parallel operations, hence it supports very aggressive data locality optimizations. This approach is the most labor intensive of the three discussed here, however, it supports the most aggressive data locality optimizations of the three.

Applications Suitable for Cloud Technology

Applications Suitable for Porting to "the Cloud"

As was demonstrated by a number of presentations during the workshop, not all computations are equally well suited for running on a distributed cloud platform. A cloud-based approach is only one of many conceivable technical choices. For specific tasks, using a large shared-memory machine might be more appropriate. In some instances, a local cluster might offer benefits over a cloud machine (e.g., when the amount of computation is small, and the amount of data is large, as is the case for the image analysis step of DNA sequencing pipelines).

Currently there is no fixed, well understood model for when an existing application is "a good fit" for a cloud-based solution. The weakly defined nature of "cloud computing" is an important contributing factor. Assuming that "cloud" is synonymous with "distributed VM image-based computing," the following factors seem to play a role in determining whether a cloud platform is a good target for a given application:

- What are the communication patterns of the application (client-server vs. intense client-client communication)? And in this context another important consideration is available network bandwidth.
- Does the application rely on a central/global high-performance file system with specific semantics or performance?

Example Applications

Various groups in bioinformatics have already gathered experiences with the use of cloud (or cloud like) platforms. Here we highlight some of the use-cases presented during the workshop.

While the scales of their computational requirements are different, several groups found aspects of cloud computing to be enabling for their data analysis needs. A group from the Medical College of Wisconsin used Amazon's EC2 product to make an existing internal pipeline available to more users (overcoming internal resource limitations that were throttling use of an instrument pipeline). This can be taken as a typical example where relatively small groups are enabled to provide their pipeline to outsiders, without the need for the group to invest in local hardware and/or charging users for computational services. Both the amount of data and the computational resources required for this type of approach are modest.

Examples two and three both showcase metagenomics applications that consume large quantities of resources with "larger" gigabyte sized datasets. Both examples are pipelines that are in constant use and are currently resource limited.

Example four from the University of Maryland is showcasing how the novel computational metaphors coming available within the cloud context alter application development.

Example 1: A Cloud-Enabled Proteomics Workflow at Medical College of Wisconsin. Modern mass spectrometers are capable of generating data many times faster than a typical single desktop computer is able to analyze it. We have brought together two recent developments, open source proteomics search programs and distributed on-demand or "cloud" computing, to allow for the construction of a highly flexible, scalable, and very low cost solution to proteomics data analysis: the Virtual Proteomics Data Analysis Cluster (ViPDAC). On boot, the application sets up the databases, links launch scripts, executes worker daemons, and starts monitoring the running processes. Access to the application is via a web browser to a server name provided by EC2 on startup. Users create a new search job and upload their datafile, which is split into independent chunks that are stored on S3 and distributed to waiting worker nodes. Each worker searches the datafile against a database specified in the job, storing the search results back on S3. When the job is complete, the head node downloads and assembles the result files into an archive suitable for use with other analysis tools.

Example 2: Argonne's MG-RAST Server. Metagenomics applications were among the first to explore the use of cloud computing. These large resource consumers are traditionally implemented as distributed applications, requiring a complex software stack and a central file system. They are also very similar to many of the existing genome analysis pipelines.

Argonne National Laboratory's metagenomics RAST server (MG-RAST) is one example for a recent development in that type of application. More than 120 gigabases of DNA have been analyzed via MG-RAST using a local cluster, TeraGrid, and cloud like resources. While the integration of TeraGrid happened by manually moving datasets and computations to TeraGrid, the integration of cloud resources was facilitated by using a novel workflow system: AWE. AWE (Argonne Workflow Engine) was initially used to run the similarity computation step of the pipeline on a variety of cloud-like resources.

AWE relies on a set of appliances that connect to a scalable fault tolerant server infrastructure for coordination. Both client and servers are lightweight and highly scalable. The server assigns work to clients based on the current workload and client capabilities. Work units are typically a small fraction of the full similarity comparison. AWE understands the structure and semantics of the work that is to be done, and hence can reuse intermediate results as well as scale the size of the work units depending on the speed and capabilities of the client execution environment. Similarly, AWE can use work unit data requirements to route work to locations where needed data is already present. Finally, AWE uses a lease mechanism in work assignment that allows automatic detection and re-routing of failure work units.

AWE provides a lightweight mechanism for distributing work across heterogenous resources, including HPC clusters, clouds, Blue Gene systems, and systems with accelerators (GPUs or FPGAs). Effectively harnessing these resources is a key challenge in order to maximize the analysis progress we can make.

Example 3: JGI's IMG/M. The DOE Joint Genome Institute (JGI) is one of the major sources of microbial genome and metagenome sequence data, currently conducting about 21% of the reported bacterial genome projects worldwide. Genome and metagenome sequence datasets from JGI and other centers are processed using annotation pipelines and then included in the Integrated Microbial Genome (IMG) system and its metagenome counterpart, IMG/M for comparative analysis. The JGI annotation pipelines support the annotation of genome and metagenome datasets sequenced using any kind of sequencing technology (Sanger, 454 GS0 – 454 Titanium, Illumina). Thus, in the past two years, the metagenome pipeline has processed more than 330 datasets, with variable sizes and distribution of sequence length. This pipeline employs a cluster of 280 CPUs with a processing rate of 24 hours for an average 454Titanium dataset of approximately .5 M reads. For datasets generated by new sequencing platforms (e.g., Illumina) the processing time is estimated to increase several fold with the current computing infrastructure. The integration of such datasets into IMG and IMG/M is also expected to require substantially larger computing capabilities. In order to determine the best solution for meeting this increasing demand for computing resources, a collaboration of researchers from the JGI's Genome Biology Program, Lawrence Berkeley National Laboratory's Biological Data Management and Technology Center, Advanced Computing for Science Department, and the National Energy Research Scientific Computing Center (NERSC) explored the performance and scalability of BLAST on a variety of platforms including a traditional HPC Platform (NERSC's Cray XT4 "Franklin" system), a commercial "Infrastructure as a Service" Cloud (Amazon's EC2), and a shared research "Platform as a Service" Cloud (Yahoo's M45).

The pricing model for cloud services rewards long-term subscription to resources, as shown by the JGI/NERSC group and Wilkening et al. (IEEE Cluster 2009).This aspect of the pricing model limits the ability to dynamically scale their computation in resource limited environments. Additional overhead costs of scientific computing on a commercial cloud include boot up time, data transfer, and loading time.

Hadoop provides an alternative programming model for data intensive computing. It has a number of interesting capabilities including moving computation to the data in distributed environments as well as fault tolerance capabilities not present in traditional HPC systems. While Hadoop can be used with pre-existing applications, replacing workflow systems, its real potential is in new fault tolerant, scalable bioinformatics algorithms. An example of this is shown in the next section.

Example 4: Using Hadoop for Genome Assembly. Michael Schatz from the University of Maryland presented Crossbow, a novel Hadoop-enabled pipeline for quick and accurate analysis of resequencing data for large eukaryotic genomes using clouds.

It combines one of the fastest sequence alignment algorithms, Bowtie, with a very accurate genotyping algorithm, SoapSNP, within Hadoop to distribute and accelerate the computation. The pipeline can accurately analyze an entire genome in one day on a 10-node local cluster or in about three hours for less than $100 using a 40-node, 320-core cluster rented from Amazon's EC2 cloud computing service.

In addition, Schatz presented a new assembly program Contrail (contrail-bio.sf.net), which uses Hadoop for de novo assembly of large genomes from short sequencing reads. Contrail relies on the graph-theoretic framework of deBruijn graphs, similar to other leading short read assemblers (Velvet, Euler-USR, and ABySS). Preliminary results show Contrail's contigs are similar to those generated by other leading assemblers when applied to small bacterial genomes, but provides superior scaling capabilities when applied to large genomes.

Architecting the Cloud Machine

A number of preferences were clearly visible in the audience comments and in the presentations during the workshop regarding the layout of a possible cloud machine to support Kbase work.

VMs. A cloud machine for the Kbase should support both predefined virtual machine images (VMs) and the capability to run user-provided VMs. This feature will enable easy porting of existing software environments adapted to pre-existing local installations. In addition, however, a number of groups will require help with creating the runtime environments required for their work and will benefit greatly from a set of predefined images. The model suggested by Kate Keahey (Argonne/University of Chicago) was to provide a VM marketplace for the machine, with user-provided (non-supported) and supported VMs.

File System. Since a lot of the computation in genomics will be data driven, a fast parallel file system providing storage for all active datasets is another requirement that was implicit in many of the presentations and became abundantly clear in the discussions during the breaks in the workshop. Because of the large amount of existing code and data, this file system needs to support the typical Linux file system semantics. Hadoop Distributed File System (HDFS) is not an option because it does not integrate with existing code and binaries used in bioinformatics and genomics.

Nodes. As application requirements are vastly different between among machine types, the cloud machine should support a variety of node types, similar to EC2. A user performing BLAST analysis will need many (16) cores with a modest amount of memory (8-16 GB), whereas a single-core sequence assembly program (velvet) benefits from maximizing the available memory (e.g., 256 GB RAM).

Node Interconnect. While many HPC machines use MPI and rely on fast internal interconnects, the vast majority of bioinformatics applications do not benefit from fast interconnects. Instead, most communication is between a node and one or more data servers, not between nodes. For this reason a fast parallel file system will add more value to a Kbase cloud machine than any fast interconnect.

Support Model. An operational model like that for EC2 is well suited for a Kbase cloud machine, as its primary technical customers will be bioinformatics groups providing solutions.

Governance Model for Kbase

Governance. The Kbase cloud infrastructure must include management and oversight as appropriate to serve the needs of the scientific community. The Kbase administrators will be expected to be aware of the user communities accessing the system. They also will continuously address issues of user satisfaction, engage in frequent use-case development, and provide resource allocation mechanisms. Several methods should be considered to ensure appropriate utilization and governance of the system.

Usage Advisory Committee. A usage advisory committee should be established to ensure fair use of the Kbase cloud system. The committee should comprise Kbase IT staff, senior Kbase personnel, and possibly experienced service providers from the external community. The advisory committee should be prepared to meet on a rapid ad hoc basis to resolve contention issues and to review the scientific merit of projects creating a high demand for the system. Other committee issues may include verification of the user's credentials, appropriateness of applications run on the system, general availability of cloud resources, and rapid development of hardware or software solutions. Mechanisms for resource requests for the general scientific community should also be developed and reviewed by the usage committee. The Kbase cloud should establish robust configuration mechanisms for all hardware, software and storage utilization. Resource allocation of cluster size, performance, and scheduling will also be reviewed by the usage committee. This committee will address issues such as the need for service level agreements and will provide recommendations on the type of service provided by the Kbase cloud.

Performance metrics. The Kbase administrators will establish performance metrics for the cloud resource. The metrics will monitor reliability, usage, and general utilization of the system. The Kbase cloud resource should also consider the use of surveys to monitor its utilization by the research and bioinformatics community. Surveys should address whether researchers are aware of the cloud resource, whether users utilize the system for their own research, whether any publications have resulted from the resource, and whether use of the system has contributed to the DOE mission. Usability studies should also be performed during the course of the cloud project. Users and workflow developers should be contacted to address the quality of services, documentation, and APIs (Application Programming Interface) of the cloud system.

Scientific advisory board. The Kbase cloud resource should include a scientific advisory board (SAB) to ensure the successful deployment of this facility and help achieve the scientific goals of the users. The SAB will meet on a regular basis and establish overall policies for resource allocation, accounting, and monitoring. The SAB should include representation from the general scientific community, individuals with technical expertise in cloud systems, Kbase staff, and IT administrators. All SAB meetings should be attended by DOE Program Officers and possibly other funding agency representatives. All survey results, performance metrics, usability studies, and newly developed procedures will be reviewed by the SAB and DOE Program Officers. The SAB will also review long-term strategic objectives of the cloud resource and specify new objectives when necessary during the course of the project.

Outreach and communication. The Kbase cloud should also dedicate resources to education, training, and outreach. Staff should provide training materials, on-line presentations, and documentation so users can fully utilize all aspects of the system. Training should be directed to users and developers with a broad range of experience levels. Topics on how to perform genome annotation, assembly, and expression analysis as well as general workflow development techniques should be addressed. Incorporation of the Kbase cloud into existing class curricula in university courses should also be considered. The Kbase should also establish multiple modes of communication such as an online wiki, error-reporting systems, electronic newsletters, and an email contact list. Kbase staff should regularly present on the Kbase cloud at workshops and scientific conferences. Other forms of outreach such as posters, promotional materials, and advertising should be used.

Glossary*

Appliance (or Software appliance)

A software appliance is a software application that might be combined with just enough operating system (JeOS) for it to run optimally on industry standard hardware (typically a server) or in a virtual machine.

IAAS

Infrastructure as a Service (IaaS) is the delivery of computer infrastructure (typically a platform virtualization environment) as a service.

Omics

Informally refers to genomics, transcriptomics, proteomics, metabolomics, and other global molecular analyses that identify and measure the abundance and fluxes of key molecular species indicative of organism or community activity under defined environmental conditions at specific points in time.

SAAS

Software as a service (SaaS, typically pronounced 'sass') is a model of software deployment whereby a provider licenses an application to customers for use as a service on demand. SaaS software vendors may host the application on their own web servers or download the application to the consumer device, disabling it after use or after the on-demand contract expires.

In the Kbase context SAAS is often used to refer to a setup where the cloud service provider provides a fixed set of VM images that users can choose to run on the machine. The other option (open set of VM images) is often referred to as IAAS.

VM (virtual machine)

System virtual machines (sometimes called hardware virtual machines) allow the sharing of the underlying physical machine resources between different virtual machines, each running its own operating system. The software layer providing the virtualization is called a virtual machine monitor or hypervisor.

*sources include Wikipedia