Science in the Cloud: Lessons from Three Years of Research Projects on Microsoft Azure

Dennis Gannon*, Dan Fay, Daron Green, Wenming Ye, Kenji Takeda
Microsoft Research
One Microsoft Way, Redmond WA 98004
*corresponding author dennis.gannon@microsoft.com

ABSTRACT

Microsoft Research is now in its fourth year of awarding Windows Azure cloud resources to the academic community. As of April 2014, over 200 research projects have started. In this paper we review the results of this effort to date. We also characterize the computational paradigms that work well in public cloud environments and those that are usually disappointing. We also discuss many of the barriers to successfully using commercial cloud platforms in research and ways these problems can be overcome.

Categories and Subject Descriptors

[**Cloud Computing, Applied Computing**]: cloud architecture, distributed systems and applications, parallelism, scalable systems, bioinformatics, geoscience, data analytics, machine learning, web services.

Keywords

Cloud computing, map reduce, scalable systems, platform as a service, infrastructure as a service, cloud programming models.

1. INTRODUCTION

Four years ago Microsoft Research began a series of programs to allow researchers access to cloud computing on our Windows Azure platform. The first three years we worked closely with research funding agencies including the National Science Foundation, The European Commission, The Chinese National Academy of Science and others. This program resulted in over 80 awards to academic researchers around the world. In August of 2013 the Microsoft Research team began a new, more open program that allowed researchers to apply directly to Microsoft for the grants. The first deadline for proposals was Oct 15, 2013 and we review proposals every 2 months. As of April 2014 over 140 proposals have been selected for the program. We expect that 180 will be selected by the end of the project's first year.

In addition to the grant program, MSR has created an extensive training program to introduce researchers to the best practices of building applications on the Windows Azure cloud. The training consists on one and two-day events that are mostly held at university facilities around the world. To date, over 500 researchers have attended these sessions. So far we have held 18 of these training events in 10 countries. A partial list of the current projects is available at the project website [1] along with information on how to apply for one of the grants or the training program.

In this paper we describe the design patterns that have been most effective for applications on Windows Azure and illustrate these with examples from actual projects. The patters we discuss are

1. Ensemble computations as map-reduce.
2. Science portals and gateways
3. Community data collections
4. Shared VM science images
5. Streaming data to the cloud

For each we will enumerate the specific lessons learned through the experiences of our users.

2. DEFINING THE CLOUD

Seven years ago the topic of cloud computing was very new. Amazon, Google and Microsoft were beginning to realize their massive data center infrastructure could be used for more than their own internal business needs and they began to offer various parts of their capabilities to the public as services. The computer science research community also began building out infrastructure services through server virtualization, one of the key cloud concepts, and many of these experiments have matured into sophisticated, open source software stacks. Seeing a possible new tool for research, the scientific community has also been drawn to the cloud.

The earliest attempts to use cloud computing for scientific application were based on the assumption that "the cloud" may be a replacement for supercomputer. A careful study of this conjecture led to disappointing results when large-scale applications and standard supercomputing benchmarks were ported to cloud platforms [2]. (A follow up study showed more promise [3] and relates to approaches described here.) To avoid this confusion we will describe the cloud here in terms of the capabilities of Microsoft Azure and note that very similar capabilities exist on the other major public cloud platforms.

Microsoft Azure is best described as a layers of services for building large scale web-based applications. At the hardware level it is a global-scale collection of data centers with millions of compute and data servers and a content delivery network. The "fabric controller" that monitors and manages the hardware resources handles automatic OS and service patching and automatic data and service replication across fault domains. At the lowest software level for application development Azure is an Infrastructure-as-a-Service (IaaS) platform for hosting Windows or Linux virtual machines (VMs).

2.1 Cloud Services

The next levels up are cloud services. These are applications with web front ends and one or more levels of backend worker processes. These are designed as stateless distributed services that are well suited for hosting multi-user, scalable analysis tasks.

The cloud service programming model provides an abstraction for building applications as scalable collections of stateless communicating processes as illustrated in Figure 1 below.

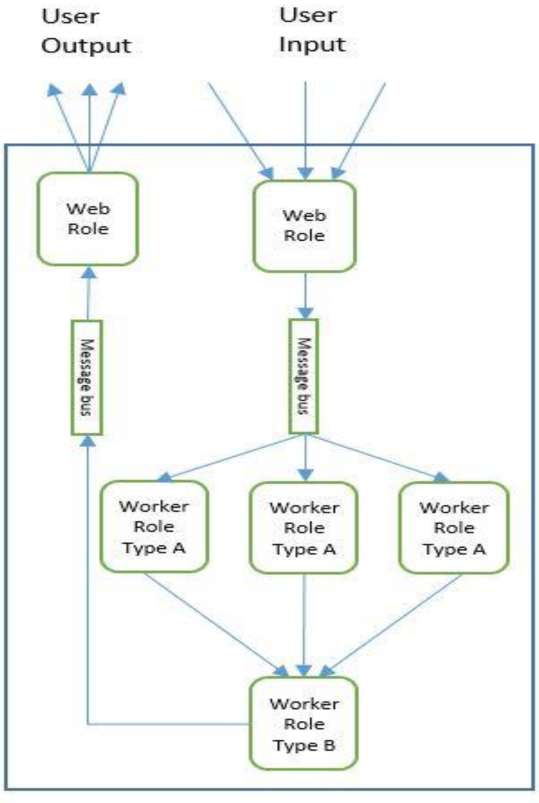

Figure 1. A cloud service composed of web roles that control input and output to remote users, a message queue for delivery of tasks to one type of worker role and a "reducer" worker role that summarizes output to return to the user.

In this model the programmer builds "web roles" and "worker roles" and defines the communication between them with message queues. Each role is realized as a continuously running process on one or more CPU cores in the data center. Web roles interact with the users by taking input which is packaged into tasks and handed to a pool of worker roles for execution. Upon completing a task a worker can send a message back to a web role that can provide a notification to the user about the result. Alternatively, the worker can send the result to a midlevel "reducer" worker that filters and accumulates a final result. While the roles are stateless they have full access to the blob and table storage systems. Statelessness is extremely important for assuring fault tolerance. If a worker crashed, the system will automatically restart it.

A number of important services exist to help building scalable cloud applications. These include

- Caching to help applications keep data closer to the application logic. Memcached protocols are supported as are multi-tenant shared caches.
- Messaging between application services in the cloud or to on premise servers is supported by the Service Bus. This allows a pub-sub topic-based subscription model as well as brokered messaging in the case that senders and receivers are not online at the same time.
- Media Services and the Content Delivery Network allow you to build high performance application that deliver media to clients running Xbox, Windows, MacOS, iOS and Android.
- Mobile application services are also supported with native client libraries for Windows Phone and iOS. The messaging services facilitate pushing notifications to mobile apps.
- Access Control Services let you easily include sophisticated authentication and authorization capabilities in your application. User authentication is via various identity providers including Facebook, Twitter, Google and Microsoft.

2.2 Data Services

The cloud is not really complete unless there is a rich collection of data services. Windows Azure has:

- Blob Storage in which each byte of data is replicated at least three times across fault domains or across multiple data centers. The replication policy is up to the application designer.
- Table Storage provides a distributed NoSQL capability for unstructured data.
- SQL Relational databases that allow replication to support multiple physical servers for high availability.
- Apache Hadoop is provided as a service that allow the user to create Hadoop clusters on demand that integrate with SQL databases and the other storage services.

All of these services are managed remotely by REST interfaces and programmatic APIs in languages such as Python, Java, PHP and C#.

3. SCIENCE IN THE CLOUD

3.1 Scalable Simulation

The extensive cloud services stack described above is a result of the evolution of Windows Azure over the last seven years. When we began providing Azure to the scientific community in 2009, the only capability that was available was the cloud services layer.

When deploying a cloud service the programmer need only tell Azure how many instances of each type of role to create. This allows applications to be very easily scaled from small number of compute cores to very large numbers. Cloud services are designed to run continuously and scale up or down based on the user load.

The cloud service model is clearly very different from the traditional batch supercomputing model. Cloud applications are designed to be fault tolerant, long running, dynamically scalable services that support many concurrent users and provide access to large data collections. Supercomputers are optimized to provide scalable computation in a batch execution environment. Consequently early attempts to port HPC MPI-based applications to Windows Azure were largely unsuccessful. However, by 2011 Windows HPC had been ported to Azure and it because possible to use Azure as a way to "burst" MPI computations from a local cluster to the cloud. This approach was used by Marty Humphrey and John Goodall in a study of watershed modeling using a cluster at the University of Virginia in combination with HPC on Microsoft Azure [4]

3.1.1 Map-Reduce and Pleasingly Parallel Computations.

The cloud service model has proved to be extremely powerful for simulation and data analysis with ensemble computations, parameter sweeps and basic map-reduce computations. In 2009 we created a cloud service for metagenomics based on the standard genomics Blast application [5]. Using this service a user could upload a collection of DNA samples to the web role and the workers would execute the Blast application on each sequence in parallel. The ensemble of results would then be gathered and returned to the user as a summary report.

This form of map-reduce in which a large number of computational tasks is mapped onto large number of workers and the result is reduced to a single result is extremely common. Radu Tudoran, Gabriel Antoniu, Bertrand Thirion and Alexandru Costan from INRIA investigated the use of the cloud for joint genetic and neuroimaging data analysis on large cohorts of subjects. The scientific goal of the project is to understand the link between genetic factors and certain brain diseases that can be detected through fMRI scans. To do this they needed to discover a significant correlation between a genetic SNP and a neuroimaging trait, or through a regression analysis discover some set of SNPs that predict a specific brain image characteristic. The team built a special-purpose map-reduce framework to tackle the data analysis. The key contribution of the project was a concurrency-optimized data storage system which federates the virtual disks associated to VMs [6]. The computations were distributed over 300 cores and they demonstrated that their map-reduce framework scaled well for this high dimensional data analysis challenge.

Many other research projects used the same approach for scalable simulation and data analysis on Azure. For example Sushil Prasad and Dinesh Agarwal used this approach for polygon overlay processing for geographic information systems [7]. Nikolas Sgourakis, while he was at the University of Washington ported the BOINC based Rosetta@home application to Azure. Using this application he was able to use 2000 Azure cores to do a major study of protein folding was to elucidate the structure of a molecular machine called the needle complex, which is involved in the transfer of infectious agents into cells from dangerous bacteria, such as salmonella, e-coli, and others [8]. We currently have over 20 bioinformatics research project running on Windows Azure.

Scalable Simulation Lessons

1. *It is possible to create a virtual HPC cluster in the cloud but unless there is a HPC scheduler and a high-bandwidth, low latency hardware network available, the performance of MPI-intensive codes will suffer. However, large scale ensemble computations and map-reduce tasks are very well suited to the cloud because network demands are very small.*
2. *The traditional commercial cloud infrastructure is designed to host long running web services. To support dynamic loads on web services it is common for a cloud to have dynamic scaling capability that allows new-VMs to be created on-demand and decommissioned when no longer needed. (This is essential if you want to keep the cost of using the cloud down.) However, deployment of new VMs takes much longer (seconds) than most scientific applications programmers expect. So applications that require a large number of VMs should have a total execution time that is long enough so that these startup delays are not significant fraction of the total time.*
3. *Building a dynamically scaled web service is not a trivial programing exercise. Most scientific programmers have not had this type of distributed systems training. There is an opportunity for systems builders to create efficient, easy to use frameworks for the scientific researcher that does not want to become a cloud computing specialist.*

Because this basic map-reduce model of computation is so common, we have developed a new tool that can be easily configured by researchers to run large ensemble computations involving compiled Matlab, R or Python applications. The input data and application code for the computations can be configured to come from Drop Box, oneDrive or Azure blog storage. We will release this to the research community in 2014.

Another approach to doing this type of map-reduce is to use Hadoop, which is now available on Azure as a service called HDInsight. Wuchun Feng and his team at Virginia Tech used HDInsight for doing genome analysis [9] and, in the process, helped the HDInsight team debug the early release of the system.

For many large data analysis tasks map-reduce is only a small component of the computation. It is often the case that the map-reduce component is part of an in iterative process. In this case there are many optimizations that can be made to improve locality and overall performance. Judy Qiu, Thilina Gunarathne, Geoffrey Fox and Xiaoming Gao from Indiana University have developed Twister4Azure as one such system for optimized iterative map reduce computations [10].

3.2 Science Gateways and Community Data Collections.

A science gateway [11] is a web portal that provides registered users with access to tools and data collections specific to some discipline. This concept was first introduced in the NSF TeraGrid project and continued with the XSEDE project. The cloud is an ideal host for science gateways and several of our projects have supported them. In fact many of the architectural elements of the cloud services described above fit this definition however in those examples the gateway users were restricted to the small research teams that built and deployed the service. A true science gateway should be designed to concurrently support a large number of users.

There are many examples of these services that are intended for broader communities of users. Ignacio Blanquer has built a gateway for support next generation genomic sequencing on Windows Azure [12]. The portal provides a web client for accessing bioinformatics tools such as BLAST, BWA, FASTA bowtie, BLAT, and SSAHA that can be configured into pipelines and run on Azure. Additional support for more complex workflow is provided by Jacek Cala and Paul Watson from the University of Newcastle based on their e-Science Central system running on Azure [13].

Jennifer Dunne of the Santa Fe Institute and Sanghyuk Yoon and Neo Martinez from Pacific Ecoinformatics and Computational Ecology Lab have developed a web portal for ecological network simulations and analysis. This one uses Network3D to provide a game-like environment for simulating ecological modeling [14]

FetchClimate [15,16] is a science gateway for retrieving climate data for any geographical region, at any grid resolution: from global, through continental, to a few kilometers, and for any range of years, days within the year, and/or hours within the day. FetchClimate can also return information on the uncertainty associated with the climate data and data sources used to fulfil the request. When multiple sources of data could potentially provide data on the same environmental variable FetchClimate automatically selects the most appropriate data sources. Finally, the entire query can be shared as a single URL, enabling others to retrieve the identical data. FetchClimate was developed by Drew Purves and the Computational Science Lab at Microsoft Research Cambridge, in collaboration with Microsoft Research Redmond and the MSTLab at Moscow State University.

3.2.1 Community Data Collections

Community data collections such as those supported by FetchClimate are critical resource for scientific communities. A big component of presenting a community data collection involves active curation: making sure that the metadata for elements of the collection is available and the data can be indexed and searched. A project from the California Digital Library and Microsoft Research is DataUp [17]. An open-source tool to help researchers document, manage, and archive their data. DataUp assists with data management and preservation, supports archiving and publishing of tabular data among scientists, allows repository administrators to upload or create required and optional metadata fields using preferred standards and ensures that data is valid for downstream data processing. It can be deployed as a web gateway on Azure with a SQL server backend.

Another interesting example comes from Harris Wu, Kurt Maly and Mohammad Zubair of Old Dominion University. They created a web-based system (FACET [18]) that allows users to collaboratively organize and classify multimedia collections on Azure.

To simplify the challenge of managing research data, Bill Howe, Garret Cole, Alicia Key, Nodira Khoussainova, and Leilani Battle of the University of Washington e-Science Institute has built a cloud-based relational data sharing and analysis platform called SQLShare [19] that allows users to upload their spreadsheet data and immediately query it by using SQL—no schema design, no reformatting, and no database administrators are required.

The British Library has published one million illustrations from 17th, 18th, and 19th century books scanned from their historic collections. These are made available on the Flickr photo sharing service, to provide public access to the images [20]. Copies of the images are also stored on Microsoft Azure, using the Flickr API to provide user interaction. Hosting the images on Azure allows them to run analytics for quality assurance, computer vision algorithms using OpenCV, and to choose related images to rotate on the Flickr web page. They use Tumblr hosted on Azure to publish updates and tweets about the illustrations every hour [21].

Science Gateway and community data collection lessons.

1. *Gateway longevity depends upon community support. A science gateway is like other online services. Without users it will die. And to keep users it will need to be refreshed and maintained.*
2. *Getting data to the cloud. One of the most common concerns about starting a cloud projects is the challenge of uploading a large data collection. While Internet2 has a 100Gb backbone network, it does not mean a researcher can move a terabyte in 10 seconds from a machine is a university lab. Because of the way TCP works it is better to use a tool that moves the data on many parallel channels simultaneously. Shipping disks is another solution. The most successful community data collections are the ones where many community members add data over an extended period of time or when existing cloud data collections can be integrated and shared.*
3. *Data Curation is an essential component of any community data collection. This fact is well understood now in many scientific disciplines and well curated collections are appearing.*
4. *A challenge for those building science gateways and community data collections is financial sustainability. Cloud data storage is expensive because it is replicated and on-line. In addition to good curation, large scale data collection need to manage multi-level storage strategies where "hot" data is kept near servers in the cloud and the rest stored in archival services. Subscription-based business models are needed to allow broad access and still pay the bills. Academic research projects can be given free access, especially if they contribute to the data quality and curation and commercial use can pay commercial rates.*

3.3 Shared VM Science Images

It is standard practice for scientific communities to share important open-source, domain-specific software tools. However, using these tools often involves complex installation procedures or the resolution of library conflicts. Cloud computing obviates such impediments by enabling communities to share a complete operating system image, pre-installed with all the tools needed by specialized groups of users. Thus, a newcomer to the community can install the image in the cloud and be doing productive work very quickly. Moreover, the community can keep the cloud-based VM image updated with the latest version of the software.

Microsoft Open Technologies operates VM Depot [22], a community-driven catalog of preconfigured operating systems, applications, and development stacks—VM images that can installed in minutes by anyone with a Microsoft Azure account. Several VM Depot images have proven popular with the scientific community. For example, Elastacloud has donated an image called Azure Data Analysis, which includes R, IPython, and a number of high quality open-source, data analysis tools.

Several other domain-specific VMs are in the works. One is an instance of the Dataverse platform for Harvard University [23]. Dataverse is designed as a web of data repositories. Having a VM image for Dataverse will make it very easy for anybody to create their one instance that can be linked into the Dataverse network. Another available VM image is BioLinux. This version has been enhanced with additional support for scripting from Python. Several other examples of science VMs will be announced soon.

We are currently soliciting proposals for Microsoft Azure resources to develop other science VM images. More information is available on the project website [1].

Challenges and lessons for Science VMs.

1. *A limitation of the Science VM is the problem of updates. If a user installs additional software in a science VM, the user will need to reinstall that software when the new version of the VM image is available. This problem goes away if the original VM hosts services and the additional user software accesses those services from a different VM.*
2. *There is an interesting tradeoff between Science VMs and Science Gateways. Software updates to a Science Gateway are largely invisible to the user. But unless the gateway owner has a way to bill the user, the owner must subsidize the cost of the user's gateway computations. By providing a VM image, the user is responsible for his or her own account and there is no additional cost for the provider of the VM image. This makes the VM image a very scalable solution.*

3.4 Streaming Data to the Cloud

One of the most common uses of public cloud resources is data streaming. The public clearly appreciates the data that streams out of the cloud in the form of movies and music. But the cloud is also a place that can be the sync of rich collections of data streams. There are obvious examples of data streaming into the cloud: E-mail, Twitter streams, images from cell phones and the vast quantities of data from the "Internet of Things" consisting of billions of on-line instruments. While we all are currently dealing with the policy challenges that arise from the pernicious use of this data to invade our privacy, there are many exciting and beneficial applications.

A great example is the work we supported at the University of the Aegean in Greece. They developed the VENUS-C Fire app [24] featuring Bing Maps, Microsoft Silverlight, and Microsoft Azure to determine the daily wildfire risk and fire propagation in the vulnerable island of Lesvos during its dry season. The data gathered involves weather data, ground instruments that monitor the dryness of ground vegetation, GPS data of the firefighting resources and satellite imagery. The university team, led by Nikos Athanasis generates a visualization of environmental factors each morning for the island's fire management team, who then use the app to determine optimal resource allocation across the island for the day. This is another excellent example of a science gateway where the users are the firefighters themselves.

Cloud computing is well-suited to periodic processing of data from instruments. Johnston *et al* [25] created a space situational awareness cloud system for processing data from the Department of Defense Space Surveillance Network to predict space debris collisions and near-earth object events. This space data is published twice daily online, and is processed by the Azure service using hundreds of Monte-Carlo simulations, with the end-to-end pipeline managed through the Azure Service Bus. This architecture is designed to accommodate computing asteroid surface impact prediction on Earth. Cloud computing is ideal for this scenario, as in the event of a potential earth impact, hundreds or thousands of cores would be required on-demand as there would be a diminishing job time to completion for such a crisis scenario.

In our new program of grants we have several projects that are addressing streaming data topics. For example, Yung-Hsiang Lu of Purdue University is working on Cloud-Based System for Continuous Analysis of Many Cameras. Yuejie Chi from Ohio State University is looking at online distributed inference of large-scale data streams in the cloud. Blesson Varghese from the University of St Andrews is working on real-time financial catastrophic risk management on Microsoft Azure. And Victor O.K. Li from the University of Hong Kong has a project on data stream analysis for hidden causality detection in urban informatics.

Lessons from streaming data to the cloud.

Most of the early experience with streaming has been positive, but it is still early and more research and experimentation is needed.

4. CONCLUSION

In this paper we have described the ways the Microsoft Azure public cloud has been used by researchers in the grant program run by Microsoft Research. While this is not the only such grant program and there are scientists that are using the cloud who work for private research laboratories, we feel that the cross section of project mentioned here are typical. We have organized them into four categories: large scale simulation, web-based science gateways and community data collections, shared virtual machine image for science and streaming data collection and analysis. With the exception of streaming, where our experience is still in its early stages, we have provided a summary of outcomes and best practices.

The examples we have chosen are, for the most part, those that were begun early enough to have published scientific outcomes. In one respect the sample presented here is not representative of many of the newer projects. In the past year we have seen a greater emphasis on "Big Data" analysis and machine learning.

We are also seeing an explosion of interest in the topic of urban informatics. This has arisen from the need of cities to better plan and manage the challenges of growth, traffic, pollution, crime, emergency services and general social welfare. New York City, Beijing, Chicago, Singapore have all initiated major programs in this area and they are actively involving the research community.

Examples of recent Microsoft Azure award projects in urban informatics include: Yanmin Zhu, Shanghai Jiao Tong University, China. "NoiseSense: Crowdsourcing-based Urban Noise Mapping with Smartphones", Peng Gong, Tsinghua University, China. "Satellite Remote Sensing for Urban Computing—40 Year Dynamic Information on Land Use for Beijing City from Time Series Landsat Data and Computer Simulation", Hojung Cha, Yonsei University, Korea. "Development of a Crowd Sensing Framework for Inducing User Participation in Urban Environments", Vassilis Glenis, Newcastle University, United Kingdom, "Modelling Flood Risk in Urban Areas" and our favorite project title "Does 'Gangnam Style' really exist? - Answers from data science perspective" a study by Joon Heo from Yonsei University of open data sets from the city of Seoul to understand the governing factors for differentiating between Gangnam and other districts in the city.

5. REFERENCES

[1] http://www.reasearch.microsoft.com/azure

[2] Keith R. Jackson, Lavanya Ramakrishnan, Krishna Muriki, Shane Canon, Shreyas Cholia, John Shalf, Harvey J. Wasserman, and Nicholas J. Wright. 2010. Performance Analysis of High Performance Computing Applications on the Amazon Web Services Cloud. In Proceedings of the 2010 IEEE Second International Conference on Cloud Computing Technology and Science (CLOUDCOM '10). IEEE Computer Society, Washington, DC, USA, 159-168. DOI=10.1109/CloudCom.2010.69.

[3] Lavanya Ramakrishnan, Piotr T. Zbiegel, Scott Campbell, Rick Bradshaw, Richard Shane Canon, Susan Coghlan, Iwona Sakrejda, Narayan Desai, Tina Declerck, and Anping Liu. 2011. Magellan: experiences from a science cloud. In Proceedings of the 2nd international workshop on Scientific cloud computing (ScienceCloud '11). ACM, New York, NY, USA, 49-58. DOI=10.1145/1996109.1996119.

[4] M. Humphrey, N. Beekwilder, J. Goodall, and M. Ercan. Calibration of Watershed Models using Cloud Computing. Proceedings of the 8th IEEE International Conference on eScience (eScience 2012). Oct 8-12 2012.

[5] AzureBlast: a case study of developing science applications on the cloud., Wei Lu, Jared Jackson, Roger S. Barga. 01/2010; DOI:10.1145/1851476.1851537 In proceeding of: Proceedings of the 19th ACM International Symposium on High Performance Distributed Computing, HPDC 2010, Chicago, Illinois, USA, June 21-25, 2010.

[6] Radu Tudoran, Alexandru Costan, Gabriel Antoniu, Hakan Soncu. "TomusBlobs: Towards Communication-Efficient Storage for MapReduce Applications in Azure." In Proc. 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid'2012), May 2012, Ottawa, Canada.

[7] Puri, S., Agarwal D., He, X., and Prasad, S.K. MapReduce algorithms for GIS Polygonal Overlay Processing, in: IEEE International Parallel and Distributed Processing Symposium workshops, to appear, Boston, USA, May 2013

[8] Thompson, J.M., N.G. Sgourakis, G. Liu, P. Rossi, Y. Tang, J.L. Mills, T. Szyperski, G.T. Montelione, and D. Baker, Accurate protein structure modeling using sparse NMR data and homologous structure information. Proceedings of the National Academy of Sciences, 2012. 109(25): p. 9875-9880.

[9] Nabeel M. Mohamed, Heshan Lin, Wuchun Feng. Accelerating Data-Intensive Genome Analysis in the Cloud. In Proceedings of the 5th International Conference on Bioinformatics and Computational Biology (BICoB), Honolulu, Hawaii, USA, March 2013.

[10] Thilina Gunarathne, Judy Qiu, and Geoffrey Fox, Iterative MapReduce for Azure Cloud in CCA11 Cloud Computing and Its Applications. April 12-13, 2011. Chicago, ILL.

[11] Wilkins-Diehr, N.; Gannon, D.; Klimeck, G.; Oster, S.; Pamidighantam, S., "TeraGrid Science Gateways and Their Impact on Science," *Computer* , vol.41, no.11, pp.32,41, Nov. 2008 doi: 10.1109/MC.2008.470

[12] Ignacio Blanquer, Goetz Brasche, Jacek Cala, Fabrizio Gagliardi, Dennis Gannon, Hugo Hiden, Hakan Soncu, Kenji Takeda, Andrés Tomás, Simon Woodman, Supporting NGS pipelines in the cloud, 2013 - journal.embnet.org

[13] J Cała, H Hiden, S Woodman, P Watson, Fast Exploration of the QSAR Model Space with e--Science Central and Windows Azure. 2012 - esciencecentral.co.uk

[14] Neo D. Martinez, Perrine Tonin, Barbara Bauer, Rosalyn C. Rael, Rahul Singh, Sangyuk Yoon, Ilmi Yoon , and Jennifer A. Dunne, Sustaining Economic Exploitation of Complex Ecosystems in Computational Models of Coupled Human-Natural Networks, Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence, 2013.

[15] http://fetchclimate.cloudapp.net/

[16] Matthew J. Smith, Paul I. Palmer, Drew W. Purves, Mark C. Vanderwel, Vassily Lyutsarev, Ben Calderhead, Lucas N. Joppa, Christopher M. Bishop, and Stephen Emmott, Changing how Earth System Modelling is done to provide more useful information for decision making, science and society., in Bulletin of the American Meteorological Society, American Meteorological Society, February 2014

[17] DataUp: Describe, Manage and Share Your Data, http://dataup.cdlib.org and http://research.microsoft.com/en-us/projects/dataup/

[18] Dazhi Chong, Kurt Maly, Elizabeth Rasnick, Harris Wu and Mohammad Zubair, "Social Curation of large multimedia collections on the cloud", Digital Humanities 2012, Hamburg, Germany, July 16-22, 2012

[19] B Howe, F Ribalet, S Chitnis, G Armbrust, D Halperin, SQLShare: Scientific Workflow Management via Relational View Sharing - 2013 Computing in Science and Engineering 15:22-31.

[20] http://www.flickr.com/photos/britishlibrary/

[21] http://mechanicalcurator.tumblr.com/

[22] http://vmdepot.msopentech.com/List/Index

[23] http://thedata.org/

[24] http://research.microsoft.com/apps/video/default.aspx?id=175587

[25] Clouds in Space: Scientific Computing using Windows Azure, Steven J Johnston*, Neil S O'Brien, Hugh G Lewis, Elizabeth E Hart, Adam White and Simon J Cox, Journal of Cloud Computing: Advances, Systems and Applications 2013, 2:2