Using Clouds for Metagenomics: A Case Study

Jared Wilkening[1,2], Andreas Wilke[1,2], Narayan Desai[1], Folker Meyer[1,2]
[1]Mathematics and Computer Science Division – Argonne National Laboratory – Argonne, IL 60439
[2]Computation Institute – University of Chicago – Chicago, IL 60637
{jared, wilke, desai, folker}@mcs.anl.gov

***Abstract*—Cutting-edge sequencing systems produce data at a prodigious rate; and the analysis of these datasets requires significant computing resources. Cloud computing provides a tantalizing possibility for on-demand access to computing resources. However, many open questions remain. We present here a performance assessment of BLAST on real metagenomics data in a cloud setting, in order to determine the viability of this approach. BLAST is one of the premier applications in bioinformatics and computational biology and is assumed to consume the vast majority of resources in that area.**

I. Introduction

Genomics is one of the areas where biology and medical research meet high-performance computing. More or less complete genomic sequence data sets are rendered into digital objects in a complicated process involving significant laboratory and insilico work. Once established, the genomic sequence of an organism triggers a significant computational workload to decipher the protein content and form hypotheses on the lifestyle of this organism.

Gene sequencing systems are quickly growing in fidelity and detail, producing increasingly large data sets for analysis. Where sequencing machines produced 500 mega-basepairs (Mbp) of output last year, the current generation of devices produce 17 gigabases (Gbp) of output; and 95 Gbp devices are expected before the end of 2009. (Note that the bp unit used to measure the quantity of DNA data is equivalent to a byte in terms of data storage.) In order to use these outputs, massive analysis, typically using BLAST (Basic Local Alignment Search Tool) [1] is required. BLAST processing time scales linearly with input size. Hence, analysis of a 17 Gbp dataset takes 34 times longer then 500 Mbp dataset. While this application is embarrassingly parallel, its computational and I/O requirements are substantial.

Metagenomics is a relatively new technique that allows the analysis of DNA samples taken from a variety of environments: marine, terrestrial, and so forth. MG-RAST (Meta Genome Rapid Annotation using Subsystem Technology) [2] is currently the leading metagenomics analysis facility. It is growing quickly, with 700 new datasets added between January and April 2009 alone. Many of these datasets stem from previous-generation DNA sequencing technology and contain on average only 100 Mbp of data. However, the analysis of these smaller datasets still requires a formidable amount of computation.

Cloud computing is poised to change the economics of computation, bringing larger efficiencies of scale to system infrastructure costs. The Amazon Elastic Compute Cloud (EC2) is the current leader in this space, but several open-source cloud provisioning toolkits [3], [4], [5] have adopted the same interfaces. While the appeal of clouds may be limited for tightly coupled applications, the MG-RAST suite appears to be an ideal candidate.

We have performed a feasibility study on the use of cloud resources in the MG-RAST workflow. We have measured several factors, including cost, performance, and operational considerations. In Section II provides background on metagenomics and cloud computing background. In Section III, we describe our experiences both in adapting the MG-RAST workflow to the cloud and the resulting application performance. In Section IV, we discuss the implications for groups considering deploying parallel applications using clouds.

II. Background

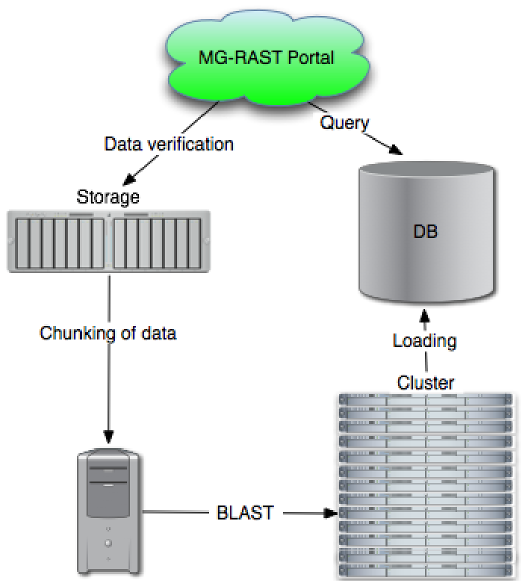
A. Metagenomics

The latest iteration of traditional genomics is metagenomics [6], the sequencing of DNA from the constituent organisms present in an environment. DNA isolation is relatively simple and inexpensive, and the cost of subsequent sequencing is decreasing relative to throughput, allowing for greater data acquisition. Therefore, researchers have been able to obtain metagenomic data from complex microbial assemblages found in marine, terrestrial, and human-derived and other mammalian-derived environments (e.g. , gastrointestinal tracts). The questions researchers are trying to answer with these datasets vary, but normally include which organisms are present (who is there), what metabolic functions are present in the DNA (what are they doing), and which variants of the genes are present and in what relative abundances.

Traditional (first-generation) genomics previously required months of upstream work (clone library creation) before DNA sequencing. Also, signicant effort was frequently required after sequencing (sequence assembly and annotation). Metagenomics datasets, particularly on the scale generated from so-called next-generation sequencing technologies, taxes the computational community even more heavily. Sequence data is isolated and sequenced within days and usually is ready for computational analysis within a week. For example application of metagenomics-oriented tools to medicine has the great potential to generate massive streams of datasets as the technology matures into a set of diagnostic tools. Published research has already highlighted the significant amount of interpersonal variation between individuals [7], [8], [9], [10]. Moreover, medical researchers are interested not only in disease states (i.e. , comparing healthy to diseased samples) but in monitoring the progression and treatment of a disease. A large set of samples can easily be generated from a single patient simply by introducing a temporal component to the sampling scheme.

B. MG-RAST

The MG-RAST portal has a straightforward workflow. Users upload DNA sequences, in the form of standard FASTA formated queries, to the MG-RAST portal. These queries are then "BLASTed" against the well curated SEED protein database [11] as well as other databases (e.g. ribosomal RNA). In the workflow, basic preprocessing is performed, chunking the query into smaller pieces and queuing for execution via Sun Grid Engine [12]. Each unit of this chunked query input is processed by a discrete BLAST instance against the various databases. This operation is the most computationally heavy portion of the workflow.



MG-RAST is currently supported by a cluster consisting of approximately 500 cores. These resources are unable to cope with the rapidly increasing demand caused by improvements in sequencing technology. Elastic computational resources, such as Amazon EC2, appear to be quite compelling for cases such as this. Performance in a local versus cloud enviroment is discussed in detail in Section III. The results are then loaded into a database for analysis by the users via the portal.

C. BLAST

BLAST is the primary tools used in bioinformatics for matching sequenced genomic sequence data to existing databases. The primary problem is that fragments of DNA data need to be matched against existing data to find similarities. Because DNA sequencing systems result in partial fragments of gene sequences encoding for proteins, each of these needs to be compared with all existing sequences in the database. This process is time consuming. BLAST implements several optimizations that allow for improved performance, at the cost of a small amount of precision. This accuracy level has been widely accepted in bioinformatics.

BLAST is a computationally bound process. It is embarrassingly parallel; the chunking process mentioned above scales single queries out into a series of work units that take approximately two hours each to process. In general, a BLAST run results in 15 times the input length in output.

D. Cloud Computing

EC2 [13] is a cloud computing platform allowing easy access to a scalable number of server instances with very little overhead. EC2 uses the Xen hypervisor[14], an open source virtualization platform, to provide these instances. Instances are classified based on the resources made available to them and the corresponding pricing is based on this. Instances are defined based on the instance type and an Amazon Machine Image (AMI).

1) Instance Types: Several flavors of instances are available, ranging from relatively small configurations to extremely large ones. Computational resources are described in terms of EC2 compute units (ECUs); each of these is roughly equivalent to a 1.0-1.2 GHz 2007 vintage Opteron or Xeon processor. Several instance types are available, which vary the RAM, number of cores, and ECUs available to an instance. The small configuration has one core, 2 ECUs, and 1.7 GB of memory, while the large instance has 8 cores, 16 ECUs, and 16 GB of memory. High-CPU flavors of these instance types are also available; these change the ratio of memory to compute power on the instance.

2) EC2 Pricing: Table I shows the instance types available from EC2 with pricing data.

TABLE I
INSTANCE TYPES

Name	ECUs	Memory	Reserved (1Y)	Demand
S	1	1.7 GB	\$325 \$0.03/hr	\$0.10/hr
L	4	7.5 GB	\$1300 \$0.12/hr	\$0.40/hr
High-CPU Large	5	1.7 GB	\$650 \$0.06/hr	\$0.20/hr
High-CPU Extra-Large	20	7 GB	\$2600 \$0.24/hr	\$0.80/hr

Pricing is determined based on a basic rate for the instance type and whether the instance was reserved or not. Instances can either be allocated on demand, or reserved in advance. Reserved instances have an initial fee but a reduced per-hour cost. Reservation costs range from \$325 to \$2600 for a single year reservation, depending on instance type. Three year reservations range in cost from \$500 to \$4000. Use of instance reservations does blunt the effectiveness of EC2's elasticity, as scaling beyond the number of reserved instances quickly causes the rate of cost to increase considerably. We performed our tests on several of these configurations.

Network transfers are also billed; Table II shows the cost structure for data transfers.

TABLE II
DATA TRANSFER COSTS

Direction	Cost
Inbound (World to Cloud)	$0.10/GB
Outbound (Cloud to World)	$0.17/GB

Data transfer costs will be modest. We expect the costs of computation to overshadow transfer costs by a large margin.

III. EXPERIENCES

The analysis phase is the most computationally expensive and time-consuming portion of the MG-RAST pipeline. Most of this phase consists of BLAST analysis. This portion is the one to which cloud computing resources are most readily applicable.

A naive implementation of cloud use is adequate for this pipeline. Once a properly configured AMI is available, it can be instituted on demand. This AMI contains the BLAST binaries and registers with SGE to receive work units.

In this section, we describe the basic setup and compare performance results between cloud instance types and local cluster nodes.

A. Setup

Instance setup was simple. We started by using a publicly available CentOS 5 x86_64 AMI. To this, we added a BLAST database. We also added a BLAST blastall binary built from NCBI v2.2.0 sources using the Intel 11 compilers. Futher, we added a FASTA sequence file containing 10 Kbp of metagenomic data. Several other minor software packages were also installed.

While this step was simple, it was not entirely turn key. Some expertise was needed in order to properly configure the system. A majority of pre-existing AMIs are built with a large common base of software; some customization is needed for all but the most basic of use cases. Also, patching is still needed. Overall, this shows that some amount of system maintenance effort is required, even for cloud-based systems.

On local system tests, the same BLAST blastall binary, FASTA sequence file, and BLAST database were used.

Table III describes the various configurations used for testing. We ran full tests using two EC2 instance types: the Large and High-CPU Extra-Large configurations. We also tried some EC2 smaller instance types, which gave poor performance because of lack of resources. Three local systems were tested as well. Each is a dual-processor quad-core system with 16 GB of RAM. These systems are of varying ages, with in-service dates from 2007, 2008, and 2009, respectively.

TABLE III
TESTED CONFIGURATIONS

Name	# ECUs/CPUs	CPU/ECU type	Memory
L	4/2	Opteron 2218 HE	7.5 GB
H-CPU XL	20/8	Xeon E5345 2.33 GHz	7 GB
N07 2007 node	8	Xeon E5430 2.66 GHz	16 GB
N08 2008 node	8	Xeon X7350 2.99 GHz	16GB
N09 2009 node	8	Xeon E5540 2.53 GHz	16 GB

B. Comparison

We compared BLAST processing rates for each node evaluated. We determined these by finding the best BLAST configuration for each node experimentally and then running a dozen iterations of the timing tests. The timing tests consisted of enough instances of BLAST in parallel to saturate the system, each processing a 10 kbp input query. Each node was otherwise idle during the test. In each of the testing configurations, all BLAST runs exhibited consistent runtimes.

BLAST has very consistent runtimes for a given input set and database. Hense, these tests clearly demonstrate the differences in performance between all of the evaluated configurations.

TABLE IV
MBP PER DAY

Instance	Mbp per day
L	1.55
H-CPU XL	8.19
N07	9.56
N08	10.485
N09	11.657

Table IV shows the BLAST times of each node class. These show an expected trend; BLAST is easily able to benefit from improved performance on new systems. In terms of raw performance, the High-CPU Extra-Large EC2 instance appear to be comparable to a 2006 or 2007 vintage system. The Large EC2 instance has insufficient resources to be able to compete effectively with all of the other platforms we evaluated.

These measurements assume that each instance or node is fully utilized over the course of the measurement interval. This is the normal operational mode for the current MG-RAST backend, so we expect it to be similarly feasible on cloud instances.

Table V shows the expected completion times of sequencing datasets for each configuration tested and input size based on the performance observed in our timing tests. These figures are the most important, as these datasets are the quanta of work for users of the MG-RAST portal. Because of the parallel nature of BLAST, one can reduce these times directly by adding more

TABLE V
PERFORMANCE TIMES IN HOURS

Instance	95 Gbp	17 Gbp	.5 Gbp	.1 Gbp
L	1,470,943	263,221	7,742	1,548
H-CPU XL	278,403	49,819	1,465	293
N07	238,503	42,680	1,255	251
N08	217,444	38,912	1,144	229
N09	195,594	33,584	988	198

nodes to the computation.

IV. DISCUSSION

Use of clouds for computation-bound applications has much potential. However, several complex aspects remain. In particular, clouds are not a panacea; there appear to be several cases where their use may not always be beneficial. In this section, we discuss three major issues regarding the use of cloud computing; cost, data security, and some of the new capabilities that clouds can bring to application portals.

A. Cost Comparison

Cost savings are one of the primary motivations for cloud computing. It is expected that large cloud providers will be able to leverage considerable economies of scale when providing resources to customers. The open question is whether this expectation holds true when the applications heavily dependent on absolute CPU performance.

To properly evaluate the relative costs of cloud computing versus local resources, we compared a single cloud node with a comparable performancing local node. Our assumption is that the local node is operated as a part of a several hundred node cluster and that the facility space is already available. We compare node costs, plus power, cooling, system administration, and maintenance costs. Note that we use conservative estimates in this analysis.

1) Local Cost Analysis: A standard rack mount server with a 500-watt power supply will consume no more than 3180 KWh per year. Using an average rate of $0.0775/KWh [15] for industrial power in the state of Illinois as of December 2008, the power costs for a single server is not exceed $246.45 annually. Cooling costs are comparable. While facility costs vary widely, we assume an annual cost of $300 per node as machine room rent. Note that large up-front costs may occur when no pre-existing facilities are available.

An experienced system administrator can easily manage a 128-node Linux cluster. A conservative estimate of staffing costs, including overhead, at $175,000/year results in costs just shy of $1367 per node per year for management. This cost typically reduces with increasing system scale. Single administrators for even larger systems are common. For example, an average administrator at Argonne National Laboratory manages both a 256-node cluster and a 32-node cluster.

Taken together, these costs result in an overall infrastructure cost of approximately $2,160 per year per cluster node. Adding an additional $3,000 for a moderately priced cluster node increases the annual cost to $5,160.

One major benefit of the cloud model is that operational costs are completely elastic. While Amazon can leverage economies of scale regardless of which customers are using a set of instances, this is not true with local management. For example, costs do not scale locally down to a single node.

2) Cloud Cost Analysis: Cloud costs are much simpler to calculate. A single high-CPU XL node costs $4,700 for a full year, considering both reservation and recurring costs. Such a configuration is capable of analyzing 3.0 Gbp. The inbound and outbound data transfer costs of these results total $9. The large node costs $2,350 for a full year, for both reservation and recurring costs. Such a configuration is capable of analyzing 565 Mbp. The inbound and outbound data transfer costs of these results total $1.80. This total provides a lower bound to costs, as the cost of software management and administration is not included.

TABLE VI
COST

Cost per Mbp		
Instance	Reserve / Owned	On demand
L	1 yr $4.15 / 3 yr $3.03	$6.20
H-CPU XL	1 yr $1.56 / 3yr $1.15	$2.34
N07	1 yr $1.47 / 3 yr $0.91	N/A
N08	1 yr $1.35 / 3 yr $0.83	N/A
N09	1 yr $1.21 / 3 yr $0.74	N/A

Note these reserved and purchased node costs assume full usage. If either becomes under-utilized, the cost per unit of data obviously increases because of decrease in throughput. On-demand costs are only linked to the actual amount of time used and remain constant. None of the EC2 node pricing includes data transfer cost to or from the cloud, since they are negligible in comparison to the computation cost. The cost of $9 for the data transfer over an entire year's worth of usage on the highest throughput EC2 node configuration is hardly a serious price consideration.

Configuration choice is an important consideration when using cloud instances, particularly in terms of cost. This is a key limitation of cloud services. Amazon offers a small number of fixed instance types. Where slight hardware adaptation can provide large performance benefits, traditional systems can provide a large advantage. For example, some BLAST databases would greatly benefit from 24 GB of RAM, but such a configuration is not currently available.

3) Costs Discussion: The above analysis clearly demonstrates that users of cloud services still, as of spring 2009, pay a premium for cloud services when compared with local hosting. This premium stems from the fact that currently available cloud node hardware lags a few years behind currently available hardware. Unless regular cloud hardware refreshes keep pace with general hardware and pricing remains constant, local hardware will continue to have the advantage in terms of cost. Cloud providers need to consider these improvements if they wish to make clouds attractive for computationally bound applications. This is not to say that cloud computing is not a cost-effective solution under some conditions. In particular, there are several situations where this is the case:

- Occasional computation – In the case of infrequent computation, paying for cloud resources can be far more effective than building all of the infrastructure required to house and operate systems.
- Computation of unpredictable scale – In the case that computations occur with widely varying scale, augmenting local resources with cloud resources can be productive and cost effective.
- Time-sensitive computations – When results are time-sensitive, the elastic quality of clouds can be appealing. This capability does come at a higher cost, however, since elasticity is lost with reservations.

In any of these situations, HPC clusters tend to be either underutilized or overcommitted. Neither condition is desirable: the former is a waste of money, and the latter is not able to deliver timely results. When HPC clusters are effectively utilized, they operate at a lower cost then do cloud resources. Especially in cases where results are not overly time-sensitive and the computational needs remain fairly stable.

B. Security

Security is a key factor when considering cloud computing for computation. As opposed to a local configuration where no data leaves the trusted local environment, data must be staged to and from the cloud. Further exposure occurs if data is stored either in the cloud or inside of AMIs. Amazon's EC2 security policy is clearly stated online [16]. It is a comprehensive policy that covers the important aspects of security in detail. In this document, Amazon states that customers have built HIPAA compliant applications using cloud services as a back end. It should be noted that even though EC2's security policies are comprehensive, additional care beyond what is required for local clusters must be taken in order to ensure data integrity.

C. Operational Model

The operational models of clouds and local resources are greatly different. In order to fully adapt MG-RAST to EC2, much more work would need to be done. More sophisticated scheduling and bookkeeping mechanism would have to be developed to keep track of the considerable amount of data that flows through the MG-RAST.

Currently system maintenance mechanisms would undoubtedly need to be changed. It is unclear at this point what the long-term reliability of EC2 instances would be under a heavy computational workload like BLAST.

A majority of the data housed within the MG-RAST is private. As with local resources, the AMI would be built from the ground up, and much more emphasis would have to be placed on security. Given Amazon's strong stance on security this would not be an insurmountable task, but it does require extra care.

System software maintenance is one area that is not vastly changed, regardless of the use of cloud computing. Cloud computing does not remove the need to understand and perform systems maintenance. Just as local machines need to be updated from time to time, so will cloud nodes if they are being used for long periods of time. For cycles shorter than a few days, maintaining AMIs should be sufficient. While shutting down and reinstantiating an instance isn't a hugely expensive process from the stand point of time, it will cumulatively affect throughput.

V. Conclusions

In this paper, we have presented a feasibility study of the use of cloud resources in the MG-RAST pipeline. Overall, the cloud, specifically Amazon's EC2, has appropriately configured resources to provide reasonable BLAST performance. Three main issues remain. First, costs are slightly higher to perform computations in the cloud, when compared with local costs. Second, the pricing of on-demand resources blunts much of the benefit of EC2's elasticity. Third, some security concerns remain to be completely addressed. Given more appealing pricing, cloud services could grow to provide the bulk of computation on public datasets. At this point however, the difference in costs is significant.

Moreover, costs in EC2 for computational capacity are calibrated against hardware that is several years old. This is reasonable for many applications, particularly IT-style ones. Computationally bound applications such as BLAST, however benefit greatly from the increased performance of each new generation of hardware. Considering the performance differences shown in Table IV, the performance gap between EC2 ECUs and real CPUs will continue to grow. Because new hardware is usually introduced at a similar price point to last year's model, this price-performance gap will continue to grow. We hope that Amazon will revisit their pricing policies to reflect this.

While our analysis does not suggest that moving the bulk of the computation stage of the MG-RAST to the cloud is prudent at this point, in several areas a more restricted use of cloud computing could be useful.

- By using cloud resources as a scale-out pool for high-priority jobs, time to solution could be greatly reduced for important jobs. These improvements would come at a substantial cost, however, as the use of on-demand resources incur a large cost penalty.
- If a looser federation mechanism were used for cloud computational instances, users could associate their cloud instances with MG-RAST, providing direct support for their computations. Users would benefit from increased priority for their jobs.

Acknowledgments

This work was supported in part by the Office of Advanced Scientific Computing Research, Office of Science, U.S. Department of Energy under contract DE-AC02-06CH11357 and in part by the Department of Energy award DE-FG02-08ER25835.

Part of this work was also funded by the National Institute of Allergy and Infectious Diseases, National Institutes of Health, Department of Health and Human Services, under Contract HHSN266200400042C.

This work was supported in part by the Office of Advanced Scientific Computing Research, Office of Science, U.S. Department of Energy, under Contract DE-AC02-06CH11357.

REFERENCES

[1] S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman, "Basic local alignment search tool." *J Mol Biol*, vol. 215, no. 3, pp. 403–410, October 1990. [Online]. Available: http://dx.doi.org/10.1006/jmbi.1990.9999

[2] F. Meyer, D. Paarmann, M. D'Souza, R. D. Olson, E. M. Glass, M. Kubal, T. Paczian, A. Rodriguez, R. Stevens, A. Wilke, J. Wilkening, and R. A. Edwards, "The metagenomics rast server - a public resource for the automatic phylogenetic and functional analysis of metagenomes," *BMC Bioinformatics*, vol. 9, pp. 386+, September 2008. [Online]. Available: http://dx.doi.org/10.1186/1471-2105-9-386

[3] Nimbus web site. http://workspace.globus.org/. [Online]. Available: http://workspace.globus.org/

[4] (2009, April) Eucalyptus web site. [Online]. Available: http://eucalyptus.cs.ucsb.edu/

[5] D. Nurmi, R. Wolski, C. Grzegorczyk, G. Obertelli, S. Soman, L. Youseff, and D. Zagorodnov, "Eucalyptus: A technical report on an elastic utility computing architecture linking your programs to useful systems," UCSB, Computer Science Technical Report 2008-10, August 2008.

[6] C. on Metagenomics, *The New Science of Metagenomics: Revealing the Secrets of Our Microbial Planet*. Natl Academy Pr, 2007.

[7] P. B. Eckburg, E. M. Bik, C. N. Bernstein, E. Purdom, L. Dethlefsen, M. Sargent, S. R. Gill, K. E. Nelson, and D. A. Relman, "Diversity of the human intestinal microbial flora," *Science*, vol. 308, no. 5728, pp. 1635–1638, June 2005. [Online]. Available: http://dx.doi.org/10.1126/science.1110591

[8] R. E. Ley, D. A. Peterson, and J. I. Gordon, "Ecological and evolutionary forces shaping microbial diversity in the human intestine." *Cell*, vol. 124, no. 4, pp. 837–848, February 2006. [Online]. Available: http://dx.doi.org/10.1016/j.cell.2006.02.017

[9] R. E. Ley, P. J. Turnbaugh, S. Klein, and J. I. Gordon, "Microbial ecology: Human gut microbes associated with obesity," *Nature*, vol. 444, no. 7122, pp. 1022–1023, December 2006. [Online]. Available: http://dx.doi.org/10.1038/4441022a

[10] D. Frank, A. S. Amand, R. Feldman, E. Boedeker, N. Harpaz, and N. Pace., "Molecular-phylogenetic characterization of microbial community imbalances in human inflammatory bowel diseases," *Proceedings of the National Academy of Sciences*, vol. 104, no. 34, pp. 13 780–13 785, 2007.

[11] R. Overbeek, T. Begley, R. M. Butler, J. V. Choudhuri, H.-Y. Chuang, M. Cohoon, V. de Crécy-Lagard, N. Diaz, T. Disz, R. Edwards, M. Fonstein, E. D. Frank, S. Gerdes, E. M. Glass, A. Goesmann, A. Hanson, D. Iwata-Reuyl, R. Jensen, N. Jamshidi, L. Krause, M. Kubal, N. Larsen, B. Linke, A. C. McHardy, F. Meyer, H. Neuweger, G. Olsen, R. Olson, A. Osterman, V. Portnoy, G. D. Pusch, D. A. Rodionov, C. Rückert, J. Steiner, R. Stevens, I. Thiele, O. Vassieva, Y. Ye, O. Zagnitko, and V. Vonstein, "The subsystems approach to genome annotation and its use in the project to annotate 1000 genomes." *Nucleic Acids Res*, vol. 33, pp. 5691–5702, Oct 2005.

[12] (2009, April) Sun grid engine. [Online]. Available: http://www.sun.com/software/sge/

[13] Amazon ec2. [Online]. Available: http://aws.amazon.com/ec2/

[14] K. Fraser, H. Steven, R. Neugebauer, I. Pratt, A. Warfield, and M. Williamson, "Safe hardware access with the xen virtual machine monitor," in *In Proceedings of the 1st Workshop on Operating System and Architectural Support for the on demand IT InfraStructure (OASIS*, 2004. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.103.6391

[15] "Average retail price of electricity to ultimate customers by end-use sector, by state," http://www.eia.doe.gov/cneaf/electricity/epm/table5_6_a.html.

[16] Amazon ec2 overview of security processes. [Online]. Available: http://developer.amazonwebservices.com/connect/entry.jspa?externalID=1697