International Journal of Science and Research (IJSR)
ISSN: 2319-7064
SJIF (2022): 7.942

A Cost-Benefit Analysis of Cloud Technology in Scientific Method of AWS

Jarman Nandi

Abstract: ***The exponential increase in the size of scientific datasets will make the existing method of transmitting data from data centres to workstations for processing obsolete very soon. Instead, processing will frequently occur on high-performance computers that are physically close to the data. There is an urgent need for assessments of how new technologies like cloud computing might enable a new distributed computing architecture. Using virtualization technologies, software that allows users to purchase computational and storage resources on demand. We provide here the findings of our research into the application of business cloud technology to scientific computing, with an emphasis on astronomy, including investigations into what sorts of applications can be run low cost and efficiently on the cloud, and an example of an application ideally suited to the cloud: processing a large-scale dataset to introduce a new research output. Here we are mainly focusing on Amazon Web Services. One of the key components of Cloud Computing is Elasticity and Scaling.***

Keywords: Virtualization Technology, Amazon Web Services, Cloud Computing, Scaling, Elasticity, Computing Architecture, Astronomy, Data Management

1.Introduction

New astronomical observatories expect to generate total data volumes of over 100 PB, a 100-fold increase [1]. The existing process of mining data from electronic archives and data centres and delivering it to PCs for integration requires a new computing architecture. Future archives must instead handle and analyse vast amounts of data using distributed high-performance technologies and platforms like grids and clouds. The astronomy community is working with computer scientists to develop the next generation of data-driven astronomical computing [2]. These include processing technology like GPUs, frameworks like MapReduce and Hadoop, and platforms like grids and clouds. What apps operate effectively and inexpensively on what platforms? Can the technology support 24/7 data centres? What are the hidden costs of these technologies? Where are the cost-efficiency trade-offs? They put what demands on apps. Is specific expertise required of end users and system developers to fully utilise them?

Several groups are exploring how apps function on these new technologies. One group [3] studies the use of GPUs in astronomy by examining performance improvements for input/output (I/0) and compute-intensive applications. They discover that 'arithmetically heavy' applications like radio telescope signal correlation and machine learning run 100 times quicker on GPUs than on CPU-based platforms. The Sloan Digital Sky Survey (http://wise.sdss.org/) has shown how MapReduce and Hadoop [5] can facilitate concurrent processing of pictures.

This research investigates the use of cloud computing in scientific process applications, focusing on astronomy. Cloud computing is a novel technique to provide and purchase computer and storage resources on demand for corporate customers. Most people are familiar with Amazon's Elastic Compute Cloud (EC2), but academic clouds like Magellan and FutureGrid are being developed for the scientific community and will be free to users. Workflow apps employ files to convey data between tasks and are data-driven. They are already widespread in astronomy and will become more so as the field's study becomes increasingly data-driven. Workflow applications include creating scientific datasets from raw satellite or ground-based sensor data. Tightly linked systems, whose processes interact directly via an internal high-performance network, are perhaps better suited to processing on computational grids [6]. An astronomer at the Infrared Processing and Analysis Center and a computer scientist at the University of Southern California's Information Sciences Institute (ISI) collaborated on this project.

2.Objectives of the Study

- To Develop a concept of Cloud Computing Technologies like-EC2, IAM, Virtualization etc of AWS
- To improve the potential threat to the company in terms of Cloud Technology
- To provide a one-stop-shop for cloud computing and IT needs
- To manage company’s IT infrastructure and operation in easy manner in detailed framework of AWS
- To Offer wide array of Cloud infrastructure and platform in affordable price as per the business need.
- To automate Security based practices to protect data in transit and rest.

3.Background of the Study

When it comes to building and running parallel applications, astronomers typically employ what is known as 'Infrastructure-as-a-Service' in the cloud environment. Root access to virtual machine (VM) instances is typically made available by cloud service providers to end users, but they typically do not provide system administrative assistance beyond ensuring that the VM instances work. The end user is responsible for configuring these instances, installing and testing applications, deploying tools for controlling and monitoring their performance, and basic system management. End users of commercial and academic clouds have been the focus of two studies [7, 8] that explore the implications of this business strategy. In general, astronomers do not have the training necessary to run their own systems or jobs, thus technologies that do this work for them are clearly needed. Wrangler [9] and the Pegasus Workflow Management System [10] were two such technologies employed in the research presented here.

- Wrangler is a cloud service that simplifies the deployment of complicated, distributed applications. For example, a Wrangler user may define their deployments in a simple extensible markup language (XML) style that describes the type and number of VMs to supply as well as the relationships between them. When the VMs are no longer required, Wrangler will provision and configure them according to their requirements, and monitoring them until they are removed.
- Pegasus has been in the works for a long time. Since its inception, it has been designed to be used by non-technical users who need to execute parallel programmes on high-performance systems. All that Pegasus needs of the end user is a simple abstract representation of the workflows (represented by an ordered DAG) that shows the processing flow and dependencies between tasks. Pegasus then takes over management and submission of jobs to the execution locations on their own initiative. Three parts make up the system.

  - An executable workflow is generated based on an abstract workflow given by the user or a workflow composition system by the mapper (Pegasus mapper). It identifies the software, data, and computing resources that are necessary for the workflow's execution. The Mapper can also redesign the process in order to improve performance, and it can include transformations for data management and the creation of provenance information.
  - The execution engine (DAGMan) is responsible for carrying out the tasks indicated by the workflow in the sequence in which they are dependent. The operations required by the executable process are carried out by DAGMan using the resources (compute, storage, and network) indicated in the executable workflow.
  - Task manager (Condor Schedd): handles individual workflow tasks, ensuring that they are executed on both local and distant resources according to specifications.

Studies involving Pegasus have two primary advantages. Assuming the programme is developed for portability, it can be performed automatically on many execution locations. In addition to managing data for the user, Pegasus infers needed data transfers, registers data into catalogues, and records performance while keeping a consistent user interface for process submission. The end user is responsible for porting apps and installing dependencies. However, as noted by both Canon et al. [7] and the United States Department of Energy Advanced Scientific Computing Research Program [8], these expenditures must be considered when using a cloud platform. Costs associated with portability are not included in the findings shown here.

Research Methodology: Performance and cost of a commercial cloud for scientific computing:

Cloud platforms are created using the same commodity hardware as data centres. Providers often charge for processing, data input and output, data storage, disc activities, and storage of VM images and applications. As a result, the expenses of operating programmes vary greatly depending on resource use. Our objective was to determine which workflow apps operate best on a commercial cloud. The study's objectives were to:

- Compare the cloud's performance to a high-performance cluster with a high-speed network and a parallel file system;
- Assess the expenses involved with executing processes on a commercial cloud.

Table 1: Comparison of Workflow resource usage by application:

Application	I/O	Memory	CPU
Montage	High	Low	Low
Broadband	Medium	High	Medium
Epigenome	Low	Medium	High

Table 2: Data Transfer sizes per workflow on Amazon EC2

Application	Inputs	Output	Logs (MB)
Montage	4291	7970	40.0
Broadband	4109	159	5.5
Epigenome	1843	299	3.3

(a) The use of resources by the workflow apps:

As a result of the wide range of computing resources utilized by these three workflow applications, we selected them as our study subjects. To create mosaics of astronomical photos, Montage (http://montage.ipac.caltech.edu) gathers images in a format known as customizable image transportation system (FITS). It is possible to construct and compare a variety of synthetic seismograms using the Broadband (http://www.usc.edu/research/cme/) (geographical locations). Using high-throughput genetic sequencing devices and a previously generated reference genome, Epigenome (http://epigenome.usc.edu/) maps small DNA segments. Throughout the investigation, we built a single procedure for each application. As seen in Table 1, each has been classified as either medium or low in terms of resource use. The sizes of the input and output data are shown in Table 2 (below). Two micrometre all sky survey (2MASS) pictures were used to create an 8° square mosaic of the Galactic Nebula M16 in Montage. This process is classified I/O-bound since it spends more than 95% of its time waiting for data to be read from or written to disc. There are just four earthquake sources monitored at five sites in the Broadband workflow, which consumes more than 75% of its duration with physical memory-intensive operations. Epigenome's process is CPU-bound, since it uses 99.9% of CPU time and just 1% for I/O and other tasks.

(b) Setting up and running an experiment in a lab:

Two high-performance clusters were used in our studies, AmEC2 (http://aws.amazon.com/ec2/) and the National Center for Supercomputer Applications Abe high-performance cluster (http://www.ncsa.illinois.edu). Although it has been discontinued since these trials, Abe is typical of high-performance computing (HPC) systems in that it had a fast network and a parallel file system to enable high performance 1/0. AmEC2 is the most popular, feature-rich, and reliable commercial cloud. AmEC2 and Abe's performance was compared on a single node utilising a local disc on both platforms as well as a parallel file system on Abe, in order to ensure an objective comparison.

The workflow-management system was hosted outside the cloud, at ISI, and all workflow jobs were coordinated using two VM images, one for 32-bit and one for 64-bit AmEC2 instances. S3, AmEC2's object-based storage system, housed all of these photos. Five AmEC2 compute resources ('types, ' in table 3) are listed in column 1 to indicate the variety of resources available. Throughout the study, these instances will be referred to as AmEC2 instances. After being stored on EBS volumes, the input data was processed on local discs before being returned to the EBS volumes for long-term storage. With a storage capacity ranging from 1 GB to 1 TB, EBS is similar to a storage area network.

Table 3: Summery of Processing Resources on Amazon EC2

Type	Arch.	CPU	Cores	Memory (GB)	Storage	Network
ml. small	32 bit	2.0-2.6 GHz Opteron	1/2	1.7	local	1Gbps Ethernet
ml. large	64 bit	2.0-2.6 GHz Opteron	2	7.5	local	1Gbps Ethernet
ml. xlarge	64 bit	2.0-2.6 GHz Opteron	4	15.0	local	1Gbps Ethernet
cl. medium	32 bit	2.33-2.66 GHz Xeon	2	1.7	local	1Gbps Ethernet
cl. xlarge	64 bit	2.0-2.66 GHz Xeon	8	7.5	local	1Gbps Ethernet

Table 4: Summary of processing resources on the Abe high-performance cluster

Type	Arch.	CPU	Cores	Memory (GB)	Storage	Network
abe. lustre	64 bit	2.33 GHz Xeon	8	8	local	10 Gbps lnfiniBand
abe. lustre	64 bit	2.33 GHz Xeon	8	8	lustre	10 Gbps lnfiniBand

Table 4 shows that both of the Abe nodes, abe.lustre and abe.local, employ the same resource type, a 64-bit Xeon computer, but they differ only in their 1/0 devices. A 10 Gbps InfiniBand network connects the two machines. Virtualization overhead on AmEC2 may be estimated based on the performance of abe.lustre vs cl.xlarge, which has a similar computational capacity. The Lustre file system was used to store all of the programme files and input files. Data were copied to a local drive before the process was performed for the abe.local trials, as well as intermediate and output data. The Lustre file system was used to store all intermediate and output data for abe. lustre. As a result, Condor glide-in jobs were deployed on Abe that established Condor daemons on the worker nodes, which in turn contacted the summit host and were used to perform workflow activities. Using grid protocols, Condor workers may be sent to a distant cluster as user tasks, which are then gliding in. For example, a Condor central manager may be utilised by the user to perform user's tasks on a distant resource via a glide-in. They reduce some of the wide-area system overheads, which helps workflow applications, function better.

(c) Amazon EQ vs. Abe EQ performance comparison:

Montage, Broadband, and Epigenome processes are shown side by side in Figure 1 for all of Amazon EC2 and Abe's platforms in Table 3 and Table 4. When we talk about runtimes, we're talking about the entire amount of time, measured in seconds, that elapses between the start of a process and the end of that workflow.

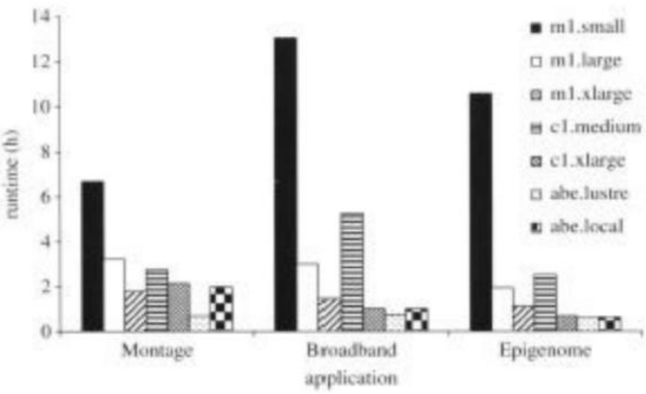


Figure 1: The runtimes in hours for the Montage, Broadband and Epigenome workflows on the Amazon EQ cloud and on Abe. The legend identifies the processor instances listed in tables 3 and 4

Commencing the VMs (usually between 70 and 90 seconds), data transmission and queue delays for starting glide-in jobs on Abe are not included in these estimates. The ml. xlarge resource provided the greatest results. For the file system buffer cache, it has twice as much memory as other machine types, and this extra capacity is utilised by the Linux kernel to speed up wait times for 1/0. All AmEC2 resource types except ml. small performed rather well, with the exception of ml. small, which is significantly less powerful than the others. Because the cl.

xlarge type is virtually identical to the abe. local type (within 8%) and delivers nearly identical performance, the virtualization overhead does not significantly decrease performance. Figure 1 illustrates the performance benefit of high-performance parallel file systems for a 1/0 bound application, which is the most relevant finding. The processing times on abe.lustre are roughly three times quicker than the fastest AmEC2 machines, even if the AmEC2 instances are not unreasonably sluggish. AmEC2 has started to provide high-performance choices after the end of this study, and redoing this experiment with them might be helpful.

Broadband Internet service (memory bound). For a memory-intensive programme like Broadband, the processing advantage of the parallel file system disappears: abe.lustre delivers just a somewhat higher performance than abe.local. To put it another way, the virtualization overhead is so little that it doesn't even make a difference. AmEC2 can match Abe's performance in memory-intensive applications like Broadband if each core has at least 1 GB of memory. If there are fewer, the system will be forced to idle certain cores in order to minimise memory or swapping shortages. Broadband has the lowest performance on machines with the smallest memory, such as the ml. small and cl. medium (1.7 GB). ml. small has a 50 percent share of one core, while cl. medium has a memory constraint that prevents it from using more than one core.

Inherited Genes (CPU bound). There is no advantage to using a parallel file system in Abe compared to using a local file system; processing times on abe.lustre were just 2% quicker than on abe.local. The processing time for cl. xlarge was 10% higher than the processing time for abe. local, which shows that virtualization overhead may be more relevant for CPU-bound applications. The greatest results for Epigenome were achieved on computers with the most cores, as predicted.

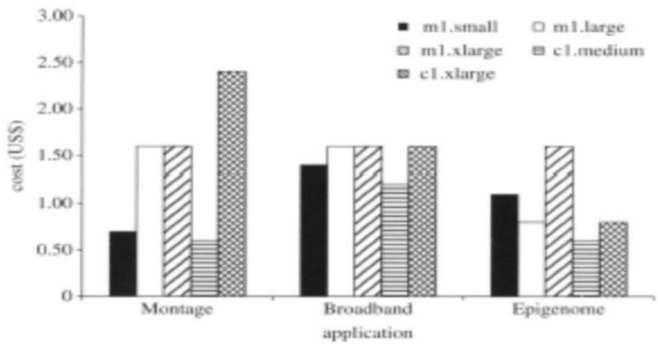


Figure 2: The processing cost for the Montage, Broadband and Epigenome workflows on the Amazon EQ cloud and on Abe. The legend identifies the processor instances listed in tables 3 and 4

(d) A cost-benefit study of the operation of workflow applications on Amazon EQ

Using AmEC2, you will be billed on an hourly basis for all of its resources, including computing resources (which includes the cost of running the VM), data storage (which includes the cost of VM images), and data transport in and out of the cloud.

The cost of resources. As demonstrated in the last column of Table 3, AmEC2 normally charges larger rates as the processor speed, the number of cores, and the amount of the memory rise, as indicated by the first column of Table 3. Graph 2 depicts the resource costs associated with the processes whose performances were represented in graph 1. The image clearly illustrates the trade-off that Montage must make between performance and expense. Even while the most powerful processor, cl.xlarge, provides a three-fold performance advantage over the least powerful processor, ml. small, it comes at a five-fold price premium. The most cost-effective approach is cl. medium, which provides performance that is just 20% less than that of ml.xlarge but at a cost that is five times less expensive.

Table 5: Storage costs for three processes on a monthly basis

Application	Input Volume (GB)	Monthly Cost
Montage	4.3	0.66
Broadband	4.1	0.66
Epigenome	1.8	0.26

Table 6: The Costs of Transferring data into and out of the Amazon EC2 Cloud

Application	Inputs (US $)	Output (US $)	Logs (US $)	Total Costs (US $)
Montage	0.42	1.32	<0.01	1.75
Broadband	0.40	0.03	<0.01	0.43
Epigenome	0.18	0.05	<0.01	0.23

Table 7: File System Investigated on Amazon EC2. [refer: Deelman et al. (10) for descriptions and references]

File System	Brief Description
Amazon S3	Distributed, object based distributed storage system
NFS	Centralized Node acts as a file server for a group of servers
GlusterFS (NUFA mode)	non-uniform file access (NUFA): write to new files always on local disk
GlusterFS (distribute mode) PVFS	distribute: files distributed among nodes
PVFS	intended for Linux clusters

When it comes to broadband, the situation is somewhat different. Processing costs do not differ significantly from machine to machine, so there is no compelling reason to use anything other than the most powerful machines available. In the case of Epigenome, the same findings are obtained: the machine with the highest performance, cl. xlarge, is also the second most affordable machine.

The expense of storing items. The cost of storing virtual machine images in S3 and the cost of storing input data in EBS are included in the storage cost. Both S3 and EBS charge a set monthly fee for data storage, as well as a fee for accessing the data, which differs depending on the application. The fixed prices for S3 are US$0.15 per GB per month, while the fixed charges for EBS are US$0.10 per GB per month. The variable costs for S3 are US$0.01 every 1000 PUT operations and US$0.01 per 10000 GET operations, whereas the variable charges for EBS are

US$0.10 per million 1/0 operations and US$0.01 per 1000 PUT operations, respectively. The 32 bit picture utilised for the trials in this study was 773 MB in size, compressed, and the 64 bit image was 729 MB in size, compressed, for a total fixed cost of US$0.22 per month for the 32 bit image and US$0.22 per month for the 64 bit image. In addition, 4616 GET operations and 2560 PUT operations were performed, resulting in a total variable cost of about US$0.03 per operation. Table 5 depicts the fixed monthly cost of storing input data for the three apps over a 12-month period. In addition, 3.18 million 1/0 operations were performed at a total variable cost of US$0.30 per operation.

The expense of the transfer. Additionally, AmEC2 charged US$0.10 per GB for data movement into the cloud and US$0.17 per GB1 for data transfer out of the cloud, in addition to its resource and storage fees. The transfer sizes and prices for the three processes are depicted in Tables 2 and 6. In Table 2, input data refers to the amount of data that is fed into the workflow, output data refers to the amount of data that is fed out of the workflow, and logs refers to the amount of logging data that is logged for workflow tasks and transmitted back to the submit host, respectively. Even though it is not included, the cost of the protocol used by Condor to communicate between the submit host and the workers is projected to be significantly less than US$0.01 per process, according to the company.

Table 6 outlines the input and output sizes, as well as their associated costs. While data transmission costs for Epigenome and Broadband are minimal, data transfer costs for Montage are significantly higher than the processing and storage expenses incurred while employing the most cost-effective resource type. Given that scientists would almost likely need to transfer items out of the cloud, transfer charges for high-volume products may prove prohibitively expensive for the scientists involved. Juve and colleagues [11] have demonstrated that the costs of data storage are significantly greater in the long run than the expenses spent if the data were housed locally. In their example, they say that hosting the 12 TB volume of the 2MASS survey on S3 would cost US$12 000 per year, the same cost as outright purchasing a disc farm, which includes hardware acquisition, support, and facility and energy fees for three years, would be the most expensive option.

e) Data sharing's cost and efficiency

To execute the processes, we employed the AmEC2 EBS storage system; however the data was copied to local drives before being used in the experiments. Different workflows, despite their similarity, differ in their ability to communicate efficiently across jobs due to a variety of factors, including storage system designs, as well as the method in which the workflow application itself consumes and saves files.

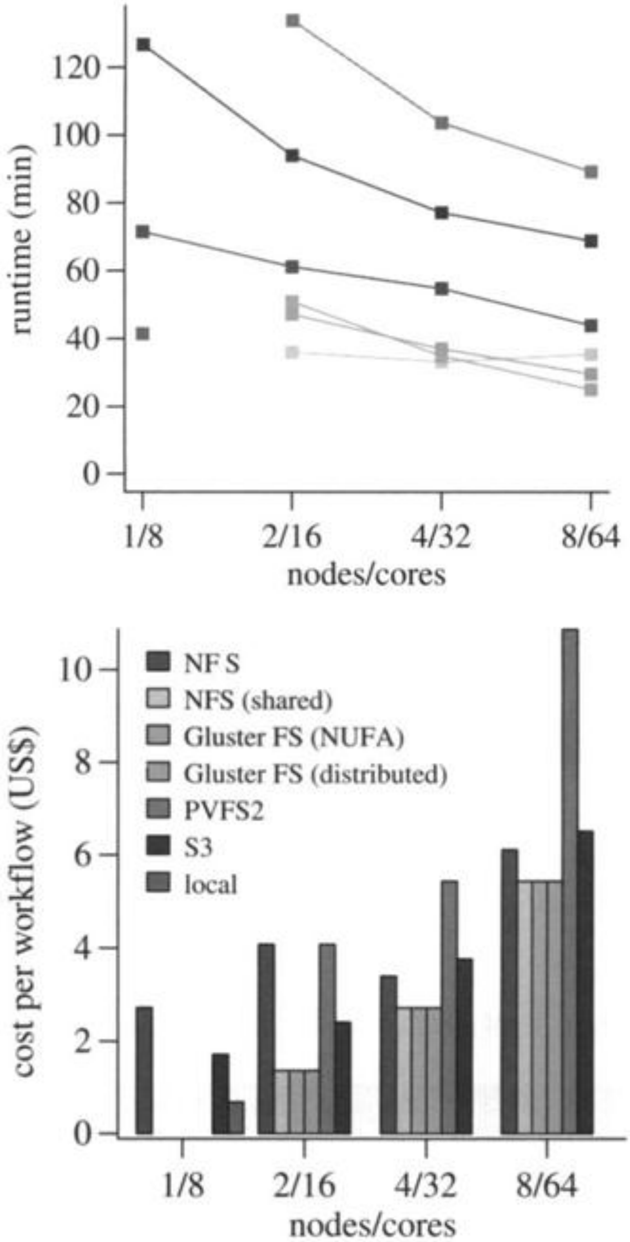
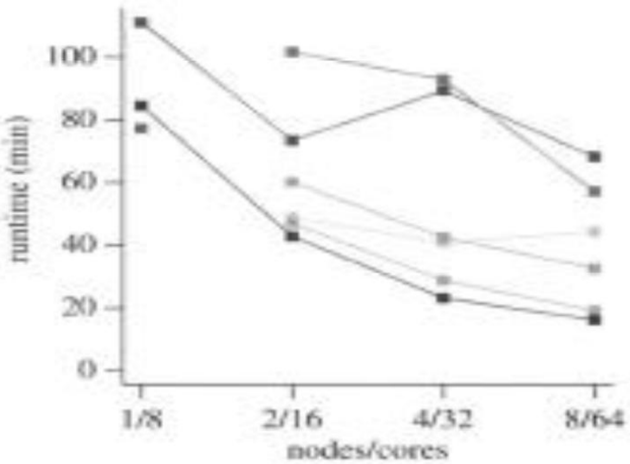


Figure 3: Variation with the number of cores of the runtime and data-sharing costs for the Montage workflow for the data storage options identified in table 7



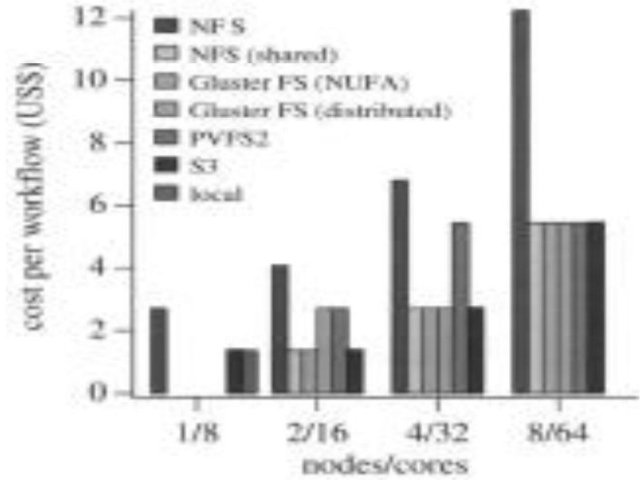


Figure 4: Variation with the number of cores of the runtime and data-sharing costs for the Broadband workflow for the data storage options identified in table 7

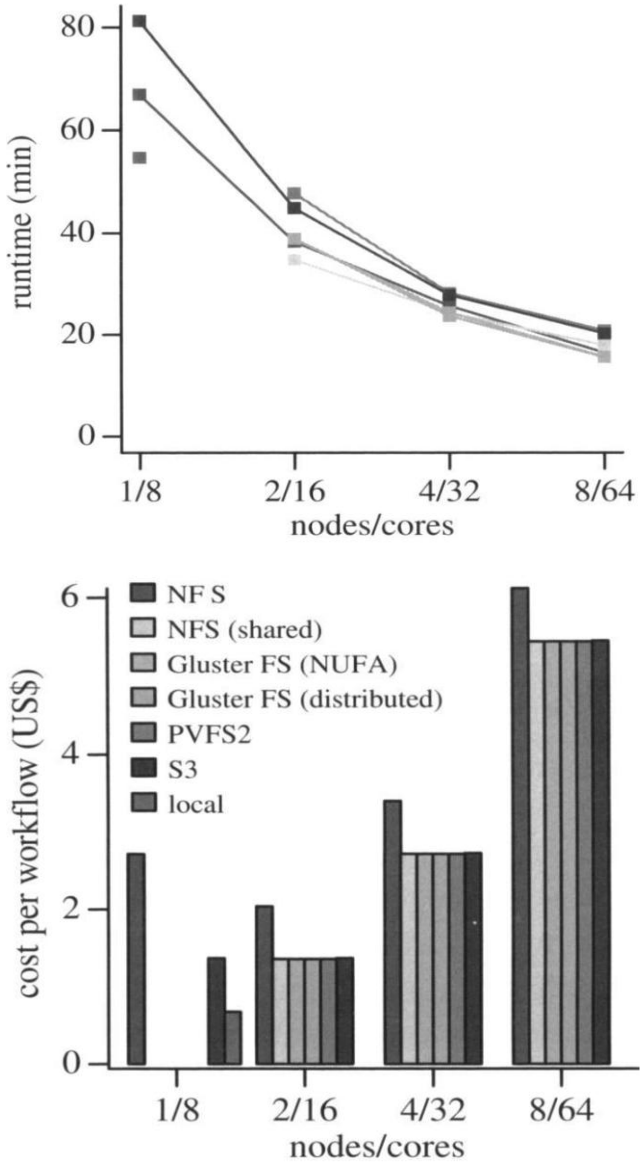


Figure 5: Variation with the number of cores of the runtime and data-sharing costs for the Epigenome workflow for the data storage options identified in table 7

Typical grids and clusters rely on a network or parallel file system for data storage The problem with replicating or replacing these file systems with storage systems with similar performance in the cloud is that it is difficult to do so. On AmEC2, in addition to Amazon S3, which the provider maintains, clients may use configuration tools like Wrangler to coordinate the creation of huge virtual clusters using standard file systems like the network file system (NFS), GlusterFS and the parallel virtual file system (PVFS). Using the storage systems indicated in table 7, we calculated the costs and performance of three different work processes. Figures 3 through 5 demonstrate how the three processes operated with these file systems as the number of worker nodes rose from 1 to 8 (the left-hand panels). Work flow runtime is strongly influenced by the storage system used. For a given number of nodes, as shown in Figure 3, the performance variance for Montage can be greater than a factor of three. This is due to Amazon S3's inability to handle the enormous number of little files generated by these procedures. PVFS's low performance is most likely due to a bug in the current release's tiny file optimization, which was not present when the test was conducted. This sort of operation is better handled by GlusterFS deployments.

However, because to its reliance on the central processing unit (CPU), Epigenome exhibits far less variety than Montage. PVFS is expected to function badly because of the enormous quantity of little files generated by broadband. The S3 client cache works better since many files are reused in the work flow, which results in better S3 performance. Most of the apps that we tested ran smoothly on GlusterFS, regardless of the size of the files or the number of clients. Possibly due to the usage of caching in our implementation of the S3 client, S3 performed well for one application. NFS functioned effectively in situations where there were few clients or if the application's 1/0 requirements were minimal. Tiny file processes worked badly on both PVFS and S3, although the PVFS version we tested did not have improvements for small files that were incorporated in future releases.

Figures 3 through 5 indicate the costs of operating the processes as a result of the changes in performance. Overall, the most cost-effective storage systems were those that offered the highest workflow performance. A dedicated node was required to host the NFS file system, but overloading a computer server with NFS server operations did not appreciably lower costs. As a result, Amazon S3 is at a disadvantage when it comes to processes involving large numbers of files. I/O heavy applications (Montage and Epigenome) were best served by GlusterFS; for memory intensive applications (Broadband), S3 was a better fit.

4.Data Analysis & Interpretation

Academic clouds for scientific purposes

(a) Academic cloud application development:

Academic clouds are being built to test technologies and facilitate research in on-demand computing. Magellan, for example, uses Eucalyptus technology (http://open.eucalyptus.com/) to create private clouds within the US Department of Energy's National Energy Research Scientific Computing Center. It is meant to research computer science difficulties relating to cloud computing systems such as authentication and authorisation, interface design, and optimization of grid- and cloud-enabled scientific applications [13]. Because AmEC2 might be prohibitively expensive for long-term processing and storage demands, we have investigated the usefulness of academic clouds in astronomy, comparing their performance to commercial clouds in the process.

(b) Academic cloud experiments

To find exoplanets transiting stars in a 105° square region in Cygnus, we calculated periodograms for the time series data published by the Kepler mission (http://kepler.nasa.gov/). A total of almost 400 000 time series datasets have been provided so far, and this figure will expand significantly by 2014. Periodograms identify periodic signals in time series data, such as transiting planets and stellar variability. They are computationally costly yet easy to parallelize since each frequency is processed independently of the others. Our research relied on the Exoplanet Archive's periodogram service [13]. It uses three techniques to determine periodicities based on their structure and underlying data sample rates. It is a CPU-intensive programme that spends 90% of its time processing data, therefore the transport and storage costs are low [13]. Our initial tests used publicly available Kepler datasets. In all, we ran two sets of short processing runs on Amazon's cloud and one huge test on the US Cyber infrastructure TeraGrid We calculated the overall workflow execution time, input/output requirements, and expenditures.

The Condor pool was created using the Wrangler provisioning and configuration tool [14]. The user may indicate how many resources (file systems, job schedulers, etc.) should be provisioned from a cloud provider using Wrangler.

Table 8: Performance and costs associated with the execution of periodograms of the Kepler datasets on Amazon and the NSF TeraGrid

Resources	RUN 1 (AmEC2)	RUN2 (AmEC2)	RUN 3 (TeraGrid)
tasks	631 992	631992	631992
mean task runtime (s)	7.44	6.34	285
jobs	25 401	25401	25401
mean job runtime (min)	3.08	2.62	118
total CPU time	1304	1113	50019
total wall time (h)	16.5	26.8	448
inputs			
input files	210 664	210 664	210664
mean input size (MB)	0.084	0.084	0.084
total input size (GB)	17.3	17.3	17.3
output files	1263 984	1263 984	1263 984
mean output size (MB)	0.171	0.124	5.019
total output size (GB)	105.3	76.52	3097.87
cost (US$)			
compute cost	179.52	291.58	4874.24 (estimated)
output cost	15.80	11.48	464.68 (estimated)
total cost	195.32	303.06	5338.92 (estimated)

Table 9: FutureGrid available Nimbus and Eucalyptus cores in November 2010. IU, Indiana University; UofC, University of Chicago; UCSD, University of California San Diego; UFI, University of Florida

Resource	CPUs	Eucalyptus	Nimbus
IU India	1024 x 2.9 Ghz Xeon	400	--
UofC Hotel	512 x 2.9 Ghz Xeon	--	336
UCSD Sierra	672 X2.5 Ghz Xeon	144	160
UFI Foxtrot	256 x 2.3 Ghz Xeon	--	248
Total	3136	544	744

Table 10: Performance of periodograms on three different clouds

Site	CPU (Ghz)	RAM (GB)	WallTime (h)	Cumulative duration (h)	Speed up
Megellan	8 x 2.6	19	5.2	226.6	43.6
Amazon	8 x 2.3	7	7.2	295.8	41.1
FutureGrid	8 x2.5	29	5.7	248.0	43.5

Results of processing 210 000 Kepler time-series datasets on AmEC2 with 128 cores (16 nodes) of the cl. xlarge instance type (Runs 1 and 2) and on NSF TeraGrid with 128 cores (8 nodes) from the Ranger cluster are shown in Table 8. (Run 3). The methods employed in Runs 1 and 2 were quite comparable, however Run 3 utilised an algorithm that was far more computationally costly. The TeraGrid and Amazon nodes have similar CPU, memory, and bandwidth. The study illustrates that commercial clouds perform well for modest calculations at acceptable costs. However, when calculations expand in size, the expenses of computing increase. A 448-hour Kepler analytical run on AmEC2 would cost over $5, 000.

We examined academic and commercial cloud performance using the Kepler workflow. We utilised the academic clouds FutureGrid and Magellan.

In addition to heterogeneous computer platforms, the FutureGrid testbed features a data management system and dedicated network. To minimise overheads and maximise performance, it supports VM-based environments as well as native operating systems. Participants in the project combine open-source software components to build an easy-to-use software environment for grid and cloud computing research.

In November 2010, five clusters were located at four FutureGrid sites across the US (Table 9). We utilised Eucalyptus and Nimbus to manage and configure resources, and to keep our resource utilisation to a minimum so that other users might benefit.

We utilised Pegasus for workflow and Wrangler for cloud resource management. We utilised Amazon EC2, FutureGrid, and Magellan to construct periodograms for 33000 Kepler datasets. These periodograms used the Plavchan algorithm [13], the periodogram code's most computationally costly method. Tentative cloud installations and their results are shown in Table 10. The cumulative duration is the total of the execution timings of all jobs in the process.

On the three clouds, the performance is comparable, with a speedup of about 43 on 48 cores. This workflow on Amazon costs around $31, plus $2 for data transmission.

These early results are quite encouraging. Academic clouds may be a better option for large-scale processing than commercial clouds.

5.Results / Discussion

Investigations into Amazon EQ in brief

- ✓ AmEC2's virtualization overhead is minimal in most cases, but it becomes more noticeable when running CPU-intensive applications.
- ✓ Resources provided by AmEC2 tend to be of lower quality and performance compared to those found in high-performance computing systems (HPCs). This is especially true for applications that rely heavily on I/O, such as those that use parallel file systems. For CPU and memory-bound applications, this advantage is essentially gone.
- ✓ End users should conduct a cost-benefit analysis of cloud resources in order to establish a usage strategy for their applications. There will always be a rise in the expenses of conducting research, but this work illustrates that the costs of resources, data transport, and storage must be included in the calculations. Data transmission costs can surpass processing costs in I/O-bound applications like Montage, as seen by this example. Expensive resources are not always the most cost effective.
- ✓ Locally hosted storage has the advantage of reducing operational costs, but AmEC2 eliminates those costs while also providing high-quality storage products.
- ✓ The disc storage technology utilised may have a significant impact on both performance and cost.
- ✓ For scientific applications, it will be beneficial to compare the costs and performance of several commercial clouds. However, such a research would require significant resources, which this article does not have.

6.Conclusions

The studies detailed here show how cloud computing may be useful in data-intensive professions like astronomy. Long-term data storage is prohibitively costly with AmEC2. For CPU-and memory-bound applications, the cloud is certainly an attractive option, especially when bulk processing is required and data quantities are small. Although AmEC2 hardware cannot match HPC systems for 1/O-bound applications, its cost and performance should be examined. End-users should always undertake a comprehensive cost-benefit analysis before using a commercial cloud to operate workflow applications, and should do it whenever prices change. While academic clouds cannot currently match AmEC2's service range, their performance on one product is comparable, and when fully developed, may be a viable alternative to commercial clouds.

References

[1] Hanisch RJ.2011 Data discovery and access for the next decade. In *Building on New Worlds, New Horizons: New Science from Sub-millimeter to Meter Wavelengths Conf, Santa Fe, New Mexico, 7-10 March 2011.* National Radio Astronomy Observatory. See https:/ /science.nrao.edu/newscience.
[2] Herriman GB, Groom SL.2011 How will astronomy archives survive the data tsunami? *Assoc. comput. Mach. Queue* **9,** 10. See http://queue.acm.org/detail.cfm?id=2047483.
[3] Barsdell BR, Barnes DG, Fluke CJ.2010 Analysing astronomy algorithms for GPUs and beyond. *Mont. Not. R. Astron. Soc.*4**08,** 1936-1944. (doi: 10.llll/j.1365-2966.2010.17257. x)
[4] Wiley K, Connolly A, Krughoff S, Gardner J, Balazinska M, Howe B, Kwon Y, Bu Y.2011 Astronomical image processing with Hadoop. In *Astronomical data analysis and software systems* XX, *ASP Conf Ser., Boston, MA, 7-11 November 2010,* vol.442 (eds IN Evans, A Accomazzi, DJ Mink, AH Rots), p.93. San Francisco, CA: Astronomical Society of the Pacific.
[5] White T.2009 *Hadoop: the definitive guide,* 1st edn. Sebabstopol, CA: O'Reilly.
[6] Juve G, Deelman E, Vahi K, Mehta G, Berriman B,

Berman BP, Maechling P.2009 Scientific workflow applications on Amazon EC2. In *Workshop on Cloud-based Services and Applications* in conjunction with *5th IEEE International Conference one-Science (e-Science 2009), Oxford, UK, 9-11 December* 2009. IEEE.

[7] Canon RS, Ramakrishnan L, Sakrejda I, Declerck T, Jackson K, Wright N, Shalf J, Muriki K.2011 Debunking some common misconceptions of science in the cloud. In *2nd Workshop on Scientific Cloud Computing (ScienceCloud 2011), San Jose, CA, June 2011*. ACM. See http://datasys.cs.iit.edu/events/ScienceCloud2011/.

[8] United States Department of Energy Advanced Scientific Computing Research (ASCR) Program.2011 Magellan final report. See http://science. energy.gov/~/media/ascr/pdf/ program-documents/docs/Magellan_Final_Report.pdf.

[9] Juve G, Deelman E.2011 Automating application deployment in infrastructure clouds. In *3rd IEEE Int. Conf Cloud Computing Technology and Science (CloudCom* 2011), *Athens*, *Greece*, 29 *November-1 December 2011*, pp.658-665. IEEE.

[10] Deelman E, Singh G, Su M-H, Blythe J, Gil Y.2005 Pegasus: a framework for mapping complex scientific workflows onto distributed systems. *Sci. Program.1*3, 219.

[11] Juve G, Deelman E, Vahi K, Mehta G, Berriman B, Berman BP, Maechling P.2010 Data sharing options for scientific workflows on Amazon EC2. In *Proc.2010 ACM/IEEE Int. Conf forHigh Performance Computing, Networking, Storage and Analysis (*SC'10), *New Orleans, LA, 13-19 November 2010.* IEEE.

[12] Juve G, Deelman E, Vahi K, Mehta G.2010 Experiences with resource provisioning for scientific workflows using corral. *Sci. Program.1***8,** 77-92.

[13] Berriman GB, Juve G, Deelman E, Regelson **M,** Plavchan P.2010 The application of cloud computing to astronomy: a study of cost and performance. In *Proc.6th IEEE Int. Conf e-Science Workshops, 7-10 December 2010,* pp.1-7. IEEE. (doi: 10.1109/eScienceW.2010.10)

[14] Laszewski G. *et al.*2010 Design of the futuregrid experiment management framework. In *Int. Conf for High Performance Computing, Networking, Storage and Analysis (*SC10), p.1. New Orleans, LA: IEEE