FleSSR Project Deliverable

Sustainable business models for community cloud infrastructure in the UK higher education sector

Matt Johnson and Andy Powell, Eduserv

Introduction

Cloud-based infrastructure essentially comprises two offerings, cloud-based compute and cloud-based storage. These are perhaps best typified for most people by the two main components of the Amazon Web Services (AWS)[1] public cloud offer, the Elastic Compute Cloud (EC2)[2] and the Simple Storage Service (S3)[3], though, of course, there are many other related services offered by Amazon and many other providers of similar public cloud infrastructure across the Internet.

Typically, the defining characteristics of such services are:

- self-service, on-demand, highly scalable provision;
- a general (and global) target audience;
- usage-based charging with billing typically done by credit card;
- remote access via the network, typically with the ability to replicate data and compute across multiple sites.

This document looks at some of the issues around the development of a sustainable community cloud infrastructure for the UK higher education (HE) sector. In the context of this document, 'community cloud', means *a cloud infrastructure offer that is targeted specifically at the HE community and which is run by organisations within (or closely aligned with) that community* and 'sustainable' means *supported long term by a business model other than central government funding*. Community cloud infrastructure services in UK HE will need to be both sensitive to the needs of academics and administrators (partly in order to differentiate them from those of the more general public cloud offers typified by AWS) and sustainable in the context of a much narrower (i.e. smaller) target market than those typically associated with public cloud services.

The rest of this document briefly considers various aspects of this space including an analysis of the market, a look at the pricing and billing models typically adopted by external providers, consideration of the pricing and billing models that might be most appropriate in the HE sector, and some thoughts on the non-price differentiators that might apply to providers in the academic space.

Note that throughout, the term 'public cloud services' is used to refer to existing mainstream cloud infrastructure offerings such as those from Amazon, Microsoft, Rackspace and so on. These services tend to have a wide (usually global) target audience. This document focuses

[1] http://aws.amazon.com/
[2] http://aws.amazon.com/ec2/
[3] http://aws.amazon.com/s3/

solely on infrastructure. It does not consider SaaS services (email, calendaring, image-sharing, etc.) running in the cloud.

Analysis of the market

This section considers the HE market for cloud infrastructure services.

There appear to be two audiences in HE for community cloud infrastructure services: individual academics (which primarily means individual researchers (or research groups) but might also include lecturers and students); and the institution itself (usually in the form of the IT Services department or equivalent but potentially also including other administrative departments).

These two audiences have different concerns and, in offering services to them, one is competing against different parties in the market place.

Academics are often already users of public cloud infrastructure such as AWS, or are at least not averse to the use of such services to support scholarly activities. As such, any community cloud infrastructure targeted specifically at researchers will have to compete (on price, functionality, service levels and environmental impact) against AWS and similar offerings.

The institution (IT Services) will typically not already be a user of public cloud infrastructure, and will probably be disposed against such use for various reasons (including concerns around loss of control, service levels, data protection, privacy and their own job security). As such, any community cloud infrastructure targeted at the institution will probably have to compete (on price, functionality, service levels and environmental impact) against similar in-house provision of the equivalent service (which may or may not be cloud-based in any true sense). This situation is compounded because of the difficulty in assessing the full costs of in-house provision for such services (because of staff being used to perform a range of activities for example), making direct comparisons on price difficult or impossible.

Many institutions will be existing users of VMware[4] and/or Hyper-V[5] internally and some will be seeing the need for extending that provision to the cloud, either to meet demand that cannot be catered for internally or to provide resilience. It seems unlikely that such a need would be met acceptably through the use of a non-VMware and non-Hyper-V community cloud infrastructure.

In terms of attitudes to payment mechanisms, individuals will typically be happy (or will at least be prepared) to pay for cloud infrastructure using their credit card (often their personal credit card which they then claim back on expenses). Many institutions find it problematic to pay for services using a credit card.

A sustainable business model for community cloud infrastructure targeted at the institution may not hold up as a sustainable model for services to individuals (because it may be pitched at a price point above AWS for example). This may be OK in the short term but any such community cloud offer may suffer significant loss of credibility in the medium terms as individual uptake becomes more widespread. Conversely, a sustainable business model for

[4] http://www.vmware.com/
[5] http://www.microsoft.com/hyper-v-server/en/us/default.aspx

community cloud infrastructure targeted at individuals may prove problematic for institutions (because of an unacceptable billing model for example).

Of course, the academic market is not necessarily limited by geography and one route towards achieving the economies of scale realizable by the likes of AWS is to target academia on a European or worldwide scale. However, to do so may also compromise factors that would otherwise be seen as potentially positive to a national audience, for example not having data flow outside any national borders and the ability to forge very close working links with a particular community. Therefore, any broadening of the target market will need to be considered carefully.

Conclusions

Analysis of the HE market, and therefore any assessment of the viability of business models, is difficult because the two key target audiences for community cloud infrastructure services are very different in terms of their cloud attitudes, service level expectations, pricing analysis, and fundamental willingness to outsource. This is compounded because offers to each audience have to compete with separate (and very different) existing offers in the marketplace. Furthermore, in the case of institutional decision-making about the use of cloud infrastructure, the competition comes from existing in-house provision which may compromise the ability to make like-for-like judgments in a fair way.

Analysis of existing public offers

This section provides an analysis of the technologies used to deliver Amazons' Web Services (AWS) portfolio, in particular focusing on EC2 and S3 which typify the fundamental building blocks of any HE cloud infrastructure offer. It also looks at the indicative costs that would be incurred by a provider wishing to host similar services for the HE market.

This section does not look at software or proprietary IP that Amazon has developed over the last 7-8 years in delivering its AWS technology, but instead focuses on the presumed architecture and infrastructure needed to do so.

It is assumed that any competing offer, targeted at the education market, would be based on existing Cloud software, either OSS such as Eucalyptus[6] or OpenStack[7], or commercial such as Flexiant[8] or Nimbula[9].

Elastic Compute Cloud

Overview

Amazon EC2 offers infrastructure-as-a-service (IaaS) hosting of Linux and Windows servers, from multiple locations in the US, the Irish Republic and Hong Kong. Services are reported hourly and billed monthly in arrears according to usage of the following elements:

- Active instances – the amount of time a virtual machine (or instance) is running – the hourly rate depends on the amount of CPU time, memory and temporary disk storage (instance storage) allocated to the machine, as well as the physical location;

[6] http://en.wikipedia.org/wiki/Eucalyptus_(computing)
[7] http://www.openstack.org/
[8] http://www.flexiant.com/
[9] http://nimbula.com/

- Data transfer – the amount of network traffic, both inbound and outbound, measured in GB per month – as the amount of data used in a month increases, the per GB rate reduces;
- Elastic Block storage – the amount of non-volatile (EBS) disk storage allocated to the machine, charged as an average of provisioned (not used) storage per month as well as I/O requests from the instance to the block device;
- EBS Snapshots – storage of "snapshot" copies of EBS volumes in Amazon's S3 data store, charged per used GB per month as well as PUT/GET requests;
- Elastic Load Balancing – automated load balancing across instances, charged based on the amount of time in use as well as the amount of GB transferred through the load-balancer per month.

EC2 Instances

Instance sizing and CPU Benchmarking

Active instances are charged primarily based on the size of the instance that is requested – the more powerful the instance (in terms of CPU, memory and disk), the greater the hourly cost.

Due to the virtutalised nature of the infrastructure, the unit of measurement for CPU provisioning is based on something called an EC2 Compute Unit (ECU). The official definition is reasonably vague for modern comparison purposes:

> *An Amazon EC2 Compute Unit provides the equivalent CPU capacity of a 1.0-1.2 GHz 2007 Opteron or 2007 Xeon processor. This is also the equivalent to an early-2006 1.7 GHz Xeon processor.*

Given recent advances in processor technology, a more useful definition has been developed, based on research conducted by Huan Liu[10]. From an analysis of the AWS infrastructure Huan Liu has identified the following physical processors, together with their associated ECU values. Passmark CPU benchmarks have also been added:

Processor	Cores	ECU Estimate	CPU Mark
Intel X5550	4	13	5232
Intel E5430	4	11	4023
Intel E5410[11]	4	10	3448

This gives a rough estimate of between 350 and 400 CPU Marks per ECU – exact figures are difficult to calculate since it depends on the underlying efficiency of the virtualisation code.

The processors listed above are generally 2-3 years old; new cloud infrastructure is likely to be built on low-profile rack servers or blade technology using the latest 6-core chips from Intel or the 12-core chips from AMD.

The respective CPU marks (and derived ECU estimates) are given below:

Processor	Cores	ECU Estimate	CPU Mark
Intel X5650[12]	6	20	7948

[10] http://huanliu.wordpress.com/2010/06/14/amazons-physical-hardware-and-ec2-compute-unit/
[11] http://www.cpubenchmark.net/cpu.php?cpu=Intel+Xeon+E5410+%40+2.33GHz
[12] http://www.cpubenchmark.net/cpu.php?cpu=Intel+Xeon+X5650+%40+2.67GHz

AMD 6176 SE[13]	12	21	8203

Derived physical server specifications

Using this information and assuming the Amazon EC2 Small Instance type as a baseline, we can infer:

- a standard physical cloud server with 2 x X5650 CPUs would support roughly 40 ECUs, or 40 Small Instances (each consuming 1 virtual core with 1 ECU);
- each Small Instance has access to 1.7 GB of memory; given a 0.3 GB overhead for virtualisation, this means that a fully-loaded physical server supporting just small instances would need 80 GB of memory allocated;
- each Small Instance has access to 160 GB of instance (local) storage, meaning that a fully-loaded server supporting just small instances would need 6.4TB of usable, or approximately 8 TB of RAID-0 raw storage.

Using a Large Instance type:

- a standard physical cloud server with 2 x X5650 CPUs would support roughly 40 ECUs, or 10 Large Instances (each consuming 2 virtual core with 2 ECUs each);
- each Large Instance has access to 7.5 GB of memory; given a 0.3 GB overhead for virtualisation, this means that a fully-loaded physical server supporting just small instances would need 78 GB of memory allocated;
- each Large Instance has access to 850 GB of instance (local) storage, meaning that a fully-loaded server supporting just small instances would need 8.5 TB of usable, or approximately 10 TB of RAID-0 raw storage.

Using an Extra Large Instance type:

- a standard physical cloud server with 2 x X5650 CPUs would support roughly 40 ECUs, or 5 Extra Large Instances (each consuming 4 virtual core with 2 ECUs each);
- each Large Instance has access to 15 GB of memory; given a 0.3 GB overhead for virtualisation, this means that a fully-loaded physical server supporting just small instances would need 77 GB of memory allocated;
- each Large Instance has access to 1700 GB of instance (local) storage, meaning that a fully-loaded server supporting just small instances would need 8.5 TB of usable, or approximately 10 TB of RAID-0 raw storage.

Since a Micro Instance type isn't based on ECU (which is variable) or instance storage (which is non-existent), the only way of deriving performance is based on memory usage:

- each Micro instance has access to 0.6 GB of memory; given a 0.3 GB overhead for virtualisation, this means that a fully-loaded physical server (with 80 GB memory) would be able to support around 80 Micro instances;
- a standard physical cloud server with 2 x X5650 CPUs would support roughly 40 ECUs, 80 Micro instances would therefore have access to 0.5 ECUs each;
- instance storage is not supported on AWS Micro instances.

[13] http://www.cpubenchmark.net/cpu.php?cpu=AMD+Opteron+6176+SE

Given the similarities between memory and CPU ratios across various instance sizes it can be inferred (and is also logical) that most EC2 instances are deployed on standardised servers with a common specification.

To summarise, EC2 physical servers are likely to be built to the following configuration:

CPU	ECUs	Memory (GB)	Disk (TB)
2 x Intel 6-core (X5650)	40	80	10
2 x Intel 4-core (E5410)	20	40	5
Derived Ratio	1	2	0.25

Note that the optimum physical server specification in terms of price/performance is likely to be dependent not on the price of the CPU but on the optimum configuration of DRAM memory in the servers (memory tends to be the single biggest cost in a high-performance server).

Pricing

EC2 Instance pricing

The following table details EU pricing for AWS EC2 instances (as of 3rd February 2011), based on both standard pricing and 1 year reservations.

	Standard			Reserved		Saving
Instance Type	**Hourly Rate ($)**	**1 Year Total[14] ($)**	**Hourly Rate ($)**	**Setup Rate ($)**	**1 Year Total ($)**	**Res v Std**
Small	0.095	832.20	0.04	227.50	577.90	31%
Large	0.38	3,328.80	0.16	910.00	2,311.60	31%
Extra Large	0.76	6,657.60	0.32	1,820.00	4,623.20	31%
Micro	0.025	219.00	0.01	54.00	141.60	35%

EC2 income

Based on the EC2 pricing shown above, a single physical server with 2 x 6-core CPUs, 80 GB of memory and 10 TB of raw disk would generate the following annual income:

Server (standard) income based on utilisation (%)					
Instance Type	**100**	**80**	**60**	**40**	**20**
Small (40)	33,288.00	26,630.40	19,972.80	13,315.20	6,657.60
Large (10)	33,288.00	26,630.40	19,972.80	13,315.20	6,657.60
X. Large (5)	33,288.00	26,630.40	19,972.80	13,315.20	6,657.60
Micro (80)	17,520.00	14,016.00	10,512.00	7,008.00	3,504.00

Note that the Micro instance types generate almost half the income than the standard types, however this is because they do not offer instance storage and so do not have to have local disk available.

Server costs

List price for an HP DL380 G7 server, with 2 x X5650 CPUs, 96 GB of memory, 8 TB of SATA disk and 2 x 10GbE is around $18,000 + VAT. This would provide a specification that matches the CPU requirement, exceeds the memory requirement but is slightly short on disk

[14] Assumes 365 days x 24 hours = 8760 hours

space; however instance storage is not one of the compelling parts of the EC2 offer, and so a reduced allocation of instance storage is unlikely to affect demand. Assuming that the lifespan of a server is 3 years, this gives an annual hardware cost of $6,000.

Power costs for the year, based on 0.15$ per KW/hr, a PUE of 2.0 and a server drawing 450W of power at peak load is around $1,200.

Typical rackspace costs would be around $12,000 per annum and would accommodate 8 physical servers, giving an annual rack cost of around $1,500.

Given that this type of infrastructure is likely to be highly automated, the operator cost per physical server should be minimal – it is assumed an hour per month for swap-out of disks or faulty components, at a rate of $100 per hour, giving an annual total of around $1,200.

Combined, the estimated server component of the infrastructure would work out at around $10,000 per annum. Based on Standard sized instance pricing and a 60% server utilisation (income of $19,900 per server annum), this works out at around a 100% margin.

This cost excludes core infrastructure costs such as networking, firewalls and overall service development and management; nevertheless it demonstrates that based on server costs alone the breakeven point can be achieved at around 30% server utilisation.

EC2 EBS Storage

Overview

Based on the Amazon AWS website[15], EBS storage offers persistent storage for Amazon EC2 instances. These offer greater I/O performance than instance storage, and can also be snapshotted to the S3 datastore for long-term archiving. EBS volumes can be between 1 GB and 1 TB in size.

Durability and Availability

EBS volumes are designed to be highly available, but not to the scale of the S3 filestore; according to Amazon:

> *Amazon EBS volume data is replicated across multiple servers in an Availability Zone to prevent the loss of data from the failure of any single component.*

This suggests that the EBS storage service is based on either a distributed SAN device such as a P4000 solution, or on a replicated SAN environment. A further statement suggests that within the same availability zone data is likely to be concentrated onto a smaller number of storage devices:

> *Because Amazon EBS servers are replicated within a single Availability Zone, mirroring data across multiple Amazon EBS volumes in the same Availability Zone will not significantly improve volume durability.*

Somewhat unusually, durability seems to be dependent on the size and delta change of the volume, with larger volumes being more susceptible to data loss.

[15] http://aws.amazon.com/ec2/#details

> *As an example, volumes that operate with 20 GB or less of modified data since their most recent Amazon EBS snapshot can expect an annual failure rate (AFR) of between 0.1% – 0.5%, where failure refers to a complete loss of the volume.*

This suggests that data is striped across a RAID partition with no more than 1 or 2 parity disks and with large RAID groups.

Performance

Given the claims of increased I/O performance compared with instance storage, is it assumed that EBS volumes are stored on a SAN-type device on SAS or FC disks, whereas instance storage is likely to be based on striped SATA disks local to the physical servers.

Based on the comment:

> *Because Amazon EBS volumes require network access, you will see faster and more consistent throughput performance with larger instances*[16]

It is assumed that EBS volumes are mounted using iSCSI or FCoE devices and that the larger compute instances (which have a greater share of physical resource) will therefore have more performant access to these SAN devices.

Regarding I/O throughput, Amazon estimates the following:

> *As an example, a medium sized website database might be 100 GB in size and expect to average 100 I/Os per second over the course of a month.*

Using the assumptions of 1 GB usable requiring 3 GB of raw storage, each GB of usable data would generate 1 I/O request per second, spread across two SAN devices. Assuming 100 TB raw storage SAN's delivering 70 TB usable, this would mean approximately 35,000 I/O requests per second on each device.

Assumptions

The following assumptions have been made about EBS storage:

- Each EBS volume is stored across 2 SAN devices (separated controllers)
- Each SAN holds data in a RAID-6 configuration with approximately 70% usable capacity
  - Therefore the overall efficiency of the storage is around 34%
  - Therefore 1 GB of usable space = 3 GB raw disk required.
- Each GB of storage will result in an average of 1 I/O request per second
  - Therefore each TB of usable storage uses ~1000 I/O requests per second

Pricing

Amazon charges a flat rate of $0.10 per GB of storage provisioned (not necessarily used) per month, as well as a charge of $0.10 per 1m I/O requests to the volume.

The first pricing factor is storage capacity; a charge of $0.10 per GB equates to $1.20 per usable GB per year, therefore the costs of raw storage must be $0.40 per GB per year. For 200 TB of raw SAN storage, this would work out at around $80,000 per annum.

[16] http://aws.amazon.com/ebs/

This assumes 100% utilisation of available disk space, which is unrealistic – however this is partly offset by the fact that EBS pricing is for provisioned, rather than used space, and that actual disk usage may be further reduced by compression and de-duplication.

The second pricing factor is I/O throughput. Based on the assumptions detailed above, 200 TB of raw SAN storage would use 70,000 I/O requests per second, or around 180bn per month. Using AWS pricing this would be around $18,000 per month or $216,000 per annum.

Taken over a 3 year period this equates to an EBS income of around $880,000, comprising $240,000 for 3 years of 200 TB raw SAN storage (split across two devices) and $648,000 for the associated I/O throughput.

Based on Amazon's 100% margin for Compute instances, this suggests that the cost of storage should be around $440,000 for 2 x 100 TB (raw) SAN devices.

Regarding usage, accurate assumptions on the actual take-up and occupancy of data storage and associated throughput is almost impossible, and so it would be recommended to build a 30-40% margin into any cost calculations for raw storage provision.

EC2 Data Transfer

Overview

Amazon monitors network traffic from the edge of the Amazon Cloud, and charge for both inbound and outbound data usage. Private network traffic within a single availability zone (i.e. within a data centre) is free of charge. Data transfer is calculated as an aggregate across the entire range of AWS services (EC2, S3, RDS, etc.).

Modelling bandwidth utilisation is perhaps the most complicated part of this analysis, since Amazon is able to achieve economies of scale far in excess of most SME hosting providers. Without historical data on cloud network utilisation is it almost impossible to develop an evidence-based usage model for bandwidth.

Assumptions

For the purposes of this analysis, it is assumed that the core networking infrastructure (including leased-line circuits, firewalls, routers and switches) are part of the general overhead – only internet transit costs will be done on a marginal basis.

Other assumptions include:

- 1 Mbit/s = 300 GB data transfer per total month;
- All data transfers happen during office hours only (8am – 6pm, Mon – Fri), resulting in approximately 200 hours per month of activity (out of around 720 total hours in a month); this gives the following: 1 Mbit/s = 80 GB of data transfer per working month;

Pricing

Inbound traffic is charged at $0.10 per GB; outbound pricing is $0.15 per GB for the first 10 TB, dropping to $0.08 per GB once transfers are more than 150 TB per month.

S3 Storage

Overview

Amazon's Simple Storage Service (S3) is a highly durable storage infrastructure designed for mission-critical and primary data storage. The service offers a web-based interface, and is object-based rather than block or file level access. S3 offers versioning capability on all objects stored in the system.

Amazon also offers a Reduced Redundancy Storage (RRS) capability, but this is outside the scope of this analysis.

Durability and Availability

S3 storage is designed to provide 99.999999999% durability and 99.99% availability of objects over a given year. It provides the following information[17] to demonstrate how this is achieved:

> *Objects are redundantly stored on multiple devices across multiple facilities in an Amazon S3 Region. To help ensure durability, Amazon S3 PUT and COPY operations synchronously store your data across multiple facilities before returning SUCCESS. Once stored, Amazon S3 maintains the durability of your objects by quickly detecting and repairing any lost redundancy. Amazon S3 also regularly verifies the integrity of data stored using checksums.*

It provides more information about resiliency in the storage facilities:

> *Designed to sustain the concurrent loss of data in two facilities.*

From this information we can derive the following capabilities:

- Data is stored on at least 3 separate physical devices;
- Writes are handled synchronously to all devices before acknowledging success;
- Background integrity checks are run regularly to ensure integrity;

The need for background checks suggests that this infrastructure is built using commodity disks, and that a custom application is used to manage the data integrity; if the data was stored in a normal SAN then this functionality would be delivered by the storage device itself.

In order to maximise storage capacity it is assumed that each S3 storage device will be configured with minimal redundancy and that data durability is gained from multi-device replication. Therefore it is likely that each S3 storage device uses RAID-5 across a large RAID group, and achieves around 80% usable storage.

Assuming a device with 100 TB raw storage (80 TB usable), 300 TB raw storage is needed to deliver 80 TB of usable capacity (N+2 redundancy), giving a ration of raw to usable storage ratio of approximately 4:1.

Performance

Amazon do not comment on specific performance metrics for S3 storage, however based on an analysis provided by HostedFTP[18], the following S3 performance metrics have been identified:

[17] http://aws.amazon.com/s3/

Size	Latency	Throughput
100KB files	120ms	0.36 MB/s
1 MB files	327ms	3 MB/s
10 MB files	1132ms	7.5 MB/s
100 MB files	6741ms	10.7 MB/s

According to the AWS blog the S3 service peaks at around 200,000 requests per second[19]. To give some indication of scale, if a typical Apache web server is able to handle around 500 concurrent connections, and so would require a front-end server farm of around 400 web servers to handle that level of demand.

Assumptions

The following assumptions have been made regarding this service:

- Average load is 10 times lower than peak load, hence around 20,000 requests per second;
- The ration of GET to PUT requests is 10:1;

S3 pricing

Amazon has a sliding scale model for S3 pricing, based on three components – storage, data transferred and requests, each of which is considered in more detail below.

Storage pricing

Charging for storage is based on the amount of data that is stored within S3, according to the following scale:

- 0 – 1 TB of data $0.14 per GB
- 1 – 50 TB of data: $0.125 per GB
- 50 – 500 TB of data: $0.110 per GB
- 500 – 1000 TB of data: $0.095 per GB

This means that 1 TB of usable data storage works out at roughly $150 per month, or $1,800 per annum; given the assumptions detailed above this would require 4 TB of raw storage, and so 1 TB of raw storage must be priced at no more than $450 per annum

Assuming a standard Amazon margin of 100%, this means that the cost of S3 storage should be no more than around $225 per TB per annum; given a 3 year lifespan of the storage, this means that the total cost per TB should be around $675.

Data transfer pricing

Data transfer pricing is based on the same model as most other AWS services, with a flat-rate for inbound data and a tiered outbound data usage model.

- Inbound: $0.100 per GB
- Outbound:
  - 0 – 1 GB: $0.00 per GB
  - 1 GB – 10 TB: $0.15 per GB
  - 10 – 50 TB: $0.110 per GB
  - 50 – 150 TB: $0.090 per GB

[18] http://hostedftp.files.wordpress.com/2009/03/s3ec2_0209.pdf
[19] http://aws.typepad.com/aws/2011/01/amazon-s3-bigger-and-busier-than-ever.html

- 150+ TB: $0.080 per GB

Data transfer pricing is subject to the same overheads and costs as detailed in the EC2 Data Transfer section.

Data request pricing

Data request pricing is based on the number of requests made through the S3 web service, and is broken down into two main types:

- PUT, COPY, POST and LIST: $0.01 per 1,000 requests
- GET: $0.01 per 10,000 requests

Given an average of 20,000 requests per second, this works out at around 51bn per month; assuming a 10:1 PUT:GET ratio, this means that around 47bn are GET requests and 4bn are PUT requests, giving a monthly income from data request pricing of:

If a web server can handle 200 requests per second then this works out at around 518m per month; assuming a 10:1 PUT:GET ratio, this means that around 470m are GET requests and 48m are PUT requests.

For each front-end web server handling 200 requests per second this gives a monthly income from data request pricing of $470 for GET requests and $480 for PUT requests, or around $11,400 per annum. Based on 100% margin, this gives a cost for the web service of around $5,700 per annum.

It is assumed that the majority of this income is spent on managing the web infrastructure and administration overhead needed to drive the S3 service, and that all storage costs (including I/O throughput) are covered by the storage costs.

Conclusions

Clearly, to be sustainable any community cloud infrastructure provision will have to be priced at approximately the same level as the major public cloud providers such as AWS. The HE market is unlikely to see any significantly higher priced offer as being attractive. The possible exception to this is where a community cloud offer that is more expensive than similar public cloud offers is still cheaper than in-house provision. However, given the difficulties in properly pricing in-house provision, this seems an unlikely scenario.

The analysis above indicates that Amazon's raw margins (excluding overheads) are in excess of 100% for the core AWS services and that it should be possible to offer broadly equivalent services to the HE community at broadly similar prices on a sustainable not-for-profit basis (despite that infrastructure being a significantly smaller scale than that of Amazon). Note, however, that in presentations given at the Eduserv Symposium 2011[20] at least two speakers noted that Amazon's current high margins give it significant breathing room to reduce prices such the need arise in the face of real competition.

Pricing and billing models

This section considers the typical pricing and billing models used by existing public cloud offers, and their appropriateness for the HE community.

[20] http://www.eduserv.org.uk/newsandevents/events/eduserv-symposium-2011

On pricing, there seem to be two options:

Pay as you use (essentially the Amazon model[21] where you are charged under various different categories for the amount of resource that you use) vs. *Plan* (where you commit to a certain amount of resource, paying extra, probably at a premium, if you go over your allowance - GoGrid[22] provide an example of this kind of pricing model).

Of course, there are some subtleties around these options. Amazon, for example, offers both Reserved Instances (where hourly rates are lowered by committing up-front to using the resource for either 1 or 3 years) and Spot Instances (where you bid for unused resource against a fluctuating hourly price). One might refer to these more generically as *Short-term commitment* vs. *Long-term commitment* and *Fixed-pricing* vs. *Bid-pricing*.

Long-term commitment allows the provider to plan their capacity more accurately (hence the lower prices. Bid-pricing soaks up otherwise unused capacity, again meaning that the provider can offer better value for money.

There's also the issue of whether providers offer any kind of free-tier, as Amazon currently do, meaning that consumers can get going at little or no cost but, ultimately, driving up consumption for the provider. Again, one might refer to this generically as *Introductory tier pricing* vs. *Flat pricing*.

On billing models, there seem to be two aspects to consider:

Credit card vs. *Invoice* (for the payment mechanism) and *Up-front payment* vs. *Pay in arrears* (for the payment model).

To briefly draw an analogy with mobile phone tariffs, pay-as-you-go is (typically) a combination of what I'm calling here Pay as you use and Up-front payment while contract is (typically) a combination of Plan and Pay in arrears.

Conclusions

It seems to be the case, from discussions at conferences and so on, that institutions are typically more keen to pay by Invoice rather than Credit card, that they like to Pay in arrears and that they lean towards the predictability of a Plan rather than the open-ended nature of Pay as you use. On the other hand individuals are more likely to be willing to pay by Credit card, using Up-front payment on a Pay as you use basis.

Offering discounts for Long term commitment and Bid-pricing seem unlikely to be a critical factor in any decision to use the community cloud or not, and so can be ignored (at least in the short term). In terms of payment mechanisms, allowing Payment in arrears based on Invoice is probably necessary to provide a compelling offer at the institutional level.

Introductory tier pricing is likely to be problematic for any relatively small-scale community cloud provider because it is difficult to manage. Furthermore, offering an introductory tier carries the danger that it encourages a mindset in the consumer that stuff can be made

[21] http://aws.amazon.com/ec2/#pricing
[22] http://www.gogrid.com/cloud-hosting/cloud-hosting-pricing.php

available for free when the reality is that it cannot. This is not to say that introductory tier pricing can't be done - rather that its use would need to be carefully managed.

Overall, this seems to imply that sustainability is likely to be achieved only by supporting a range of pricing and billing models, appealing to both individual and institutional consumers.

Differentiating factors

This section considers the factors that might differentiate a community cloud infrastructure service offer for HE from existing solutions. In doing so, it is necessary to consider both those factors that distinguish a community cloud offer from existing public cloud offers and those that distinguish a community cloud offer from other in-house solutions. As noted above, individuals are more likely to be interested in the first of these, while institutions are more likely to be interested in the second.

For the former, the key differentiating factors are network connectivity, both in terms of the cost of that connectivity and in terms of latency and bandwidth (note that it is assumed here that any community cloud infrastructure offer is located directly on the JANET network), and closeness to the community (in terms of understanding community need, holding a position of trust, and so on).

For the latter, the key differentiators lie in price (though, as noted above, this may be difficult to compare to in-house solutions), service levels, 24/7 operation, efficiency, environmental impact and resilience.

Community cloud providers are highly likely to be UK-based and be able to offer significant reassurance that data will not leave UK borders and be managed in line with the DPA and related legislation. Furthermore, community cloud providers are likely to be wholly owned by universities or have charitable status, operating in line with a broadly ‘public good’ and ‘educational’ mission. This means that they will not need to be responsive to the short term demands of shareholders, nor likely to be able to sold off in ways that compromise their charitable mission.

The cloud infrastructure environment is evolving rapidly with technical and business developments happening on a regular basis. A potential negative differentiating factor of any community cloud infrastructure offer in the HE space is that the service is unable to keep pace with the rate of change of other public offers. This may manifest itself in various ways including an inability to match new business models (new pricing or billing models for example), core technical capability (compute power, storage capacity or whatever) or mechanisms for self-provisioning of resources (the functionality and usability of the Web interfaces to the service). Some of these issues may be mitigated through the use of open source software such as OpenStack, however these solutions, on their own, are unlikely to provide a complete solution given the current state of development, nor are they likely to be particularly agile in response to market forces.

Conclusion

The biggest factor against there being sustainable business models for community cloud infrastructure providers in the HE sector seems unlikely to be price. Rather it is the potential lack of both functionality and agility against the big public cloud providers.