Journal of Network and Computer Applications 32 (2009) 1012–1022

Contents lists available at ScienceDirect

Journal of Network and Computer Applications

journal homepage: www.elsevier.com/locate/jnca

MetaCDN: Harnessing ‘Storage Clouds’ for high performance content delivery

James Broberg [a,*], Rajkumar Buyya [a], Zahir Tari [b]

[a] Department of Computer Science and Software Engineering, The University of Melbourne, Australia
[b] Department of Computer Science and Information Technology, RMIT University, Australia

ARTICLE INFO

Article history:
Received 30 October 2008
Received in revised form
12 March 2009
Accepted 21 March 2009

Keywords:
Content delivery networks
Cloud storage
Storage as a service
Overlay
Load balancing
Request redirection

ABSTRACT

Content delivery networks (CDNs) such as Akamai and Mirror Image place web server clusters in numerous geographical locations to improve the responsiveness and locality of the content it hosts for end-users. However, their services are priced out of reach for all but the largest enterprise customers. An alternative approach to content delivery could be achieved by leveraging existing infrastructure provided by ‘Storage Cloud’ providers, who offer internet accessible data storage and delivery at a fraction of the cost. In this paper, we introduce MetaCDN, a system that exploits ‘Storage Cloud’ resources, creating an integrated overlay network that provides a low cost, high performance CDN for content creators. MetaCDN removes the complexity of dealing with multiple storage providers, by intelligently matching and placing users’ content onto one or many storage providers based on their quality of service, coverage and budget preferences. MetaCDN makes it trivial for content creators and consumers to harness the performance and coverage of numerous ‘Storage Clouds’ by providing a single unified namespace that makes it easy to integrate into origin websites, and is transparent for end-users. We then demonstrate the utility of this new approach to content delivery by showing that the participating ‘Storage Clouds’ used by MetaCDN provide high performance (in terms of throughput and response time) and reliable content delivery for content consumers, whilst the MetaCDN system itself introduces minimal overhead compared to using these ‘Storage Clouds’ directly.

1. Introduction

Numerous ‘Storage Cloud’ providers (or ‘Storage as a Service’) have recently emerged that can provide Internet-enabled content storage and delivery capabilities in several continents, offering service level agreement (SLA) backed performance and uptime promises for their services. Customers are charged only for their utilisation of storage and transfer of content (i.e. a *utility computing* Broberg et al., 2008 model), which is typically in the order of cents per gigabyte. This represents a large paradigm shift away from typical hosting arrangements that were prevalent in the past, where average customers were locked into hosting contracts (with set monthly/yearly fees and excess data charges) on shared hosting services like DreamHost (New Dream Network,). Larger enterprise customers typically utilised pervasive and high performing content delivery networks (CDNs) like Akamai (Maggs and Technologies, 2001; Su et al., 2006) and Mirror Image, who operate extensive networks of ‘edge’ servers which deliver content across the globe. In recent years it has become increasingly difficult for competitors to build and maintain competing CDN infrastructure, and a once healthy landscape of CDN companies has been reduced to a handful via mergers, acquisitions and failed companies (Pathan and Buyya, 2008). However, far from democratising the delivery of content, the most pervasive remaining CDN provider (Akamai) is priced out of the reach of most small to medium enterprises (SMEs), government agencies, universities, and charities (Rayburn, 2008). As a result, the idea of utilising Storage Clouds as a poor man’s CDN is very enticing. At face value, these storage providers promise the ability to rapidly and cheaply ‘scale-out’ to meet both flash crowds (which is the dream and the nightmare of most web site operators) and anticipated increases in demand. Economies of scale, in terms of cost effectiveness and performance for both providers and end-users, could be achieved by leveraging existing ‘Storage Cloud’ infrastructure, instead of investing large amounts of money in their own content delivery platform or utilising one of the incumbent operators like Akamai. In Section 2, we analyse the services provided by these storage providers, as well as their respective cost structures, to ascertain if they are a good fit in our MetaCDN system.

These emerging services have reduced the cost of content storage and delivery by several orders of magnitude, but they can be difficult to use for non-developers, as each service is best utilised via unique web services or programmer API’s, and have their own unique quirks. Many websites have utilised individual

* Corresponding author. Tel.: +61 3 83441339.
E-mail addresses: brobergj@csse.unimelb.edu.au (J. Broberg), zahir.tari@rmit.edu.au (Z. Tari).

1084-8045/$ - see front matter
doi:10.1016/j.jnca.2009.03.004

Table 1
Cloud storage cost structure (<2 TB for Nirvanix, <10 TB for Amazon, <5 TB for Mosso).

Cost type	Nirvanix global SDN	Amazon S3 USA	Amazon S3 EU	Amazon CloudFront	Mosso Cloud Files
Incoming data ($/GB)	0.18	0.1	0.1	N/A	0.00[a]
Outgoing data ($/GB)	0.18	0.17	0.17	0.17 (US/EU), 0.21 (HK), 0.221 (JP)	0.22
Storage ($/GB)	0.25	0.15	0.18	N/A	0.15
Requests ($/1000 PUT)	0.00	0.01	0.012	N/A	0.02
Requests ($/10,000 GET)	0.00	0.01	0.012	0.01 (US), 0.013 (JP), 0.012 (EU/HK)	0.00

[a] Free until March 2009.

Storage Clouds to deliver some or all of their content (Elson and Howell, 2008), most notably the New York Times (Gottfrid, 2007) and SmugMug (MacAskill, 2007), however there is no general purpose, reusable framework to interact with multiple Storage Cloud providers and leverage their services as a Content Delivery Network. Most 'Storage Cloud' providers are merely basic file storage and delivery services, and do not offer the capabilities of a fully featured CDN such as automatic replication, fail-over, geographical load redirection, and load balancing. Furthermore, a customer may need coverage in more locations than offered by a single provider. To address this, in Section 3 we introduce MetaCDN, a system which utilises numerous storage providers in order to create an overlay network that can be used as a high performance, reliable and redundant geographically distributed CDN.

However, in order to utilise storage and file delivery from these providers in MetaCDN as a Content Delivery Network, we want to ensure they provide sufficient performance (i.e. predictable and sufficient response time and throughput) and reliability (i.e. redundancy, file consistency). Whilst individual Storage Clouds have been trialled successfully for application domains such as Science Grids (Palankar et al., 2007; Matei Ripeanu, 2007) and offsite file backup (Jungle Disk, Inc.,), their utility for general purpose content delivery, which requires low latency and high throughput, has not been evaluated rigourously. In Section 4 we rigourously evaluate the performance of replicas deployed by the MetaCDN system over a 24 h period, from six different client locations, as well as examining the overhead of the MetaCDN system itself. In Section 5 we consider the future directions of MetaCDN and identify potential enhancements for the service. Finally, in Section 6 we offer some concluding remarks and summarise our contribution.

2. Storage Clouds

In order to ascertain the feasibility of building a system such as MetaCDN, we need to evaluate whether the Storage Clouds used posses the necessary features, performance and reliability characteristics to act as CDN replica servers. Whilst performance is crucial for content delivery, we also need to examine the cost structures of the different providers. At face value these services may appear ludicrously cheap, however, they have subtle differences in pricing and the type of services billed to the end-user, and as a result a user could get a nasty surprise if they have not understood what they will be charged for.

For the purposes of this paper, we chose to analyse the two most prominent Storage Cloud providers, Amazon Simple Storage Service (S3) and Nirvanix Storage Delivery Network (SDN). At the time of writing, Amazon offers storage nodes in the United States and Europe (specifically, Ireland) whilst Nirvanix has storage nodes in the United States (over two separate sites in California), Germany and Singapore. Another Storage Cloud provider of note is Mosso Cloud Files, located in Dallas, TX, which recently launched in late 2008 as a beta service. Microsoft have also announced their Cloud Storage offering, Azure Storage Service, which is currently available only as a limited community technology preview (CTP). Unfortunately, we were unable to thoroughly evaluate these providers at the time of writing. As an interesting counterpoint, we also evaluated the performance of Coral CDN (Freedman et al., 2004), a free peer-to-peer content distribution network, composed of a world-wide network of web proxies and name-servers that run on PlanetLab (Fiuczynski, 2006) nodes across the globe. Whilst Coral is not specifically a Storage Cloud, it is used by many websites to handle day-to-day traffic as well as to add on-demand capacity to deal with flash crowds. However, Coral CDN offers no specific SLA or performance guarantees.

Amazon S3 was launched in the United States in March 2006, and in Europe in November 2007, opening up the huge infrastructure that Amazon themselves utilise to run their highly successful e-commerce company, Amazon.com. In November 2008, Amazon launched CloudFront, a content delivery service that added 14 edge locations (8 in the United States, 4 in Europe, and 2 in Asia). However, unlike S3, CloudFront does not offer persistent storage. Rather, it is analogous to a proxy cache, with files deployed to the different CloudFront locations based on demand and removed automatically when no longer required. Amazon provides REST and SOAP interfaces to its storage resources, allowing users the ability to read, write or delete an unlimited amount of objects, with sizes ranging from 1 byte to 5 GB each. As noted in Table 1, Amazon S3 has a storage cost of $0.15 per GB/month in their USA data center, or $0.18 per GB/month in their EU data center. Incoming traffic (i.e uploads) are charged at $0.10 per GB/month, and outgoing traffic (i.e. downloads) are charged at $0.17 per GB/month, from the USA or EU site. For larger customers, Amazon S3 has a sliding scale pricing scheme, which is depicted in Fig. 1. Discounts for outgoing data occur after 10, 50, and 150 TB of data a month has been transferred, resulting in a subtly sub-linear pricing response that is depicted in the figure. An important facet of Amazon's pricing that should be noted by users is the additional cost per 1000 PUT/POST/LIST or 10,000 GET HTTP requests, which can add up depending on the type of content a user places on Amazon S3. Whilst these costs are negligible if a user is utilising Amazon S3 to primarily distribute very large files, if they are storing and serving smaller files, a user could see significant extra costs on their bill. In Fig. 1, we can see for users serving content with an average file size of 100 kB, a larger cost is incurred. As a point of comparison, we have included the 'average' cost of the top 4–5 major incumbant CDN providers.[1]

Nirvanix launched its Amazon S3 competitor, the Nirvanix Storage Delivery Network (SDN) on September 2007. The Nirvanix service was notable in that it had a SLA-backed uptime guarantee at a time when Amazon S3 was simply operated on a best-effort

[1] Information obtained from Rayburn (2008) and http://www.cdnpricing.com, part of a popular website and blog for CDN and streaming media professionals run by StreamingMedia.com.

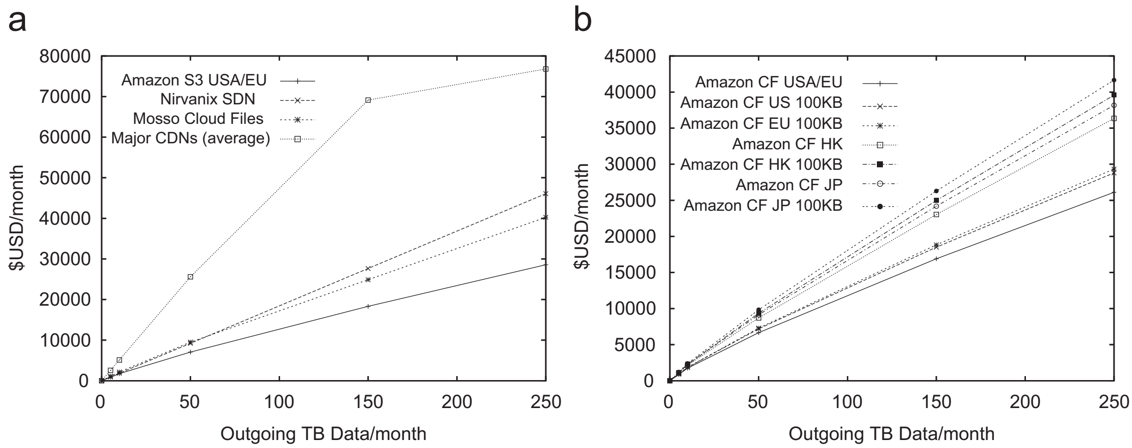

Fig. 1. Pricing comparison. (a) Cloud Storage pricing versus traditional CDNs. (b) Amazon CloudFront.

Table 2
Feature comparison.

Feature	Nirvanix SDN	Amazon S3	Amazon CF	Mosso Cloud Files	Azure Storage	Coral CDN
SLA	99.9	99-99.9	99-99.9	99.9	*	None
Max. file size	256 GB	5 GB	5 GB	5 GB	5 GB	50 MB
US PoP	Yes	Yes	Yes	Yes	Yes	Yes
EU PoP	Yes	Yes	Yes	Yes	No	Yes
Asia PoP	Yes	No	Yes	Yes	No	Yes
Australasia PoP	No	No	No	Yes	No	Yes
Per File ACL	Yes	Yes	Yes	Yes	Yes	No
Automatic replication	Yes	No	Yes	Yes	No	Yes
Developer API	Yes	Yes	Yes	Yes	Yes	No

The asterisk denotes that no SLA has been defined at this time for Azure Storage, as the service has not officially launched yet.

service basis. Unsurprisingly, shortly after Nirvanix launched its SDN, Amazon added their own SLA backed uptime guarantees. Nirvanix differentiates itself in several ways (depicted in Table 2), notably by having coverage in four locations, offering automatic file replication over sites in the SDN for performance and redundancy, and supporting file sizes up to 256 GB. Nirvanix is priced slightly higher than Amazon's service, and they do not publish their pricing rates for larger customers (>2 TB/month). Nirvanix provides access to their resources via SOAP or REST interfaces, as well as providing SDK's in Java, PHP Zend, Python, and C#.

The Coral CDN is not specifically a Storage Cloud, but it does have some level of control for site operators that wish to utilise the large network of Coral servers distributed over the globe. Coral is trivial to utilise and incorporate into websites, and can be triggered simply by appending .nyud.net onto a URL (e.g. http://www.cnn.com.nyud.net) that you wish to be cached by the Coral CDN. Coral supports the *Expires: header*, *Pragma: no-cache* and the *Cache-control: header*, allowing the site operator some level of control over what is cached, when it is cached and for how long. Coral is a free service, which is appealing to many, and is used frequently on websites that have been linked from Slashdot.org and Digg.com, to avoid their websites crumbling under the sudden increase in load. However, Coral has some limitations that might preclude it being utilised by site operators. Coral is a large network of web proxies, and it does not provide persistent storage. Files are cached for 12 h at a time, with a maximum file size of 50 MB, making it unsuitable to use to distribute large files. Furthermore, participating Coral nodes have limits placed on how much data they are willing to distribute. At the time of writing, Coral sites can distribute an upper-bound of 250 GB per day, but individual sites may have lower limits, as well as hourly and per host limits. Coral is operated on a best-effort basis, and due to its community driven, non-commercial nature, it offers no SLA or performance guarantees.

3. The MetaCDN system

In this section we introduce MetaCDN, a system that leverages several existing Storage Clouds, creating an integrated overlay network that aims to provide a low cost, high performance, easy to use content delivery network for content creators and consumers.

The MetaCDN service (depicted in Fig. 2) is presented to end-users in two ways. First, as a web portal,[2] that was developed using Java Enterprise and Java Server Faces (JSF) technologies, with a MySQL back-end to store user accounts and deployments, and the capabilities, pricing and historical performance of service providers. The web portal acts as the entry point to the system and also functions as an application-level load balancer for end-users that wish to download content that has been deployed by MetaCDN. Using the web portal, users can sign up for an account on the MetaCDN system, and enter credentials for any cloud storage or other provider they have an account with. Once this simple step has been performed, they can utilise the MetaCDN system to intelligently deploy content onto storage providers according to their performance requirements and budget limitations. The web portal is most suited for small or ad-hoc

[2] A screen-cast of the MetaCDN web portal interface is available at http://www.metacdn.org

deployments, and is especially useful for less technically inclined content creators.

The second method of accessing the MetaCDN service is via RESTful Web Services. These Web Services expose all of the functionality of the MetaCDN system. This access method is most suited for customers with more complex and frequently changing content delivery needs, allowing them to integrate the MetaCDN service in their own origin web sites and content creation workflows.

3.1. *Critical functionality of the MetaCDN platform*

The MetaCDN system works by integrating with each storage provider via *connectors* (shown in Figs. 2 and 3) that provides an abstraction to hide the complexity arising from the differences in how each provider allows access to their systems. An abstract class, *DefaultConnector*, prescribes the basic functionality that each provider could be expected to support, and *must* be implemented for all existing and future connectors. These include basic operations like creation, deletion and renaming of replicated files and folders. If an operation is not supported on a particular service, then the connector for that service throws a *FeatureNotSupportedException*. This is crucial, as whilst the providers themselves have very similar functionality, there are some key differences, such as the largest allowable file size or the coverage footprint. Fig. 3 shows two connectors (for Amazon S3 and Nirvanix SDN respectively), highlighting one of Amazon's most well-known limitations—that you cannot rename a file, which should result in a *FeatureNotSupportedException* if called. Instead, you must delete the file and re-upload it. The Nirvanix connector throws a *FeatureNotSupportedException* when you try and create a Bittorrent deployment, as it does not support this functionality, unlike Amazon S3. Connectors are also available for shared or private hosts via connectors for commonly available FTP-accessible shared web hosting (shown in Fig. 2), and privately operated web hosting that may be available via SSH/SCP or WebDAV protocols.

The MetaCDN service has a number of core components that contain the logic and management layers required to encapsulate the functionality of different upstream storage providers and present a consistent, unified view of the services available to end-users. These components include the *MetaCDN Allocator*, which selects the optimal providers to deploy content to, and performs the actual physical deployment. The *MetaCDN QoS monitor* tracks the current and historical performance of participating storage providers, and the *MetaCDN Manager* tracks each user's current deployment and performs various housekeeping tasks. The *MetaCDN Database* stores crucial information needed by the MetaCDN portal, ensuring reliable and persistent operation of the system. The *MetaCDN Load Redirector* is responsible for directing MetaCDN end-users (i.e. content consumers) to the most appropriate file replica, ensuring good performance at all times.

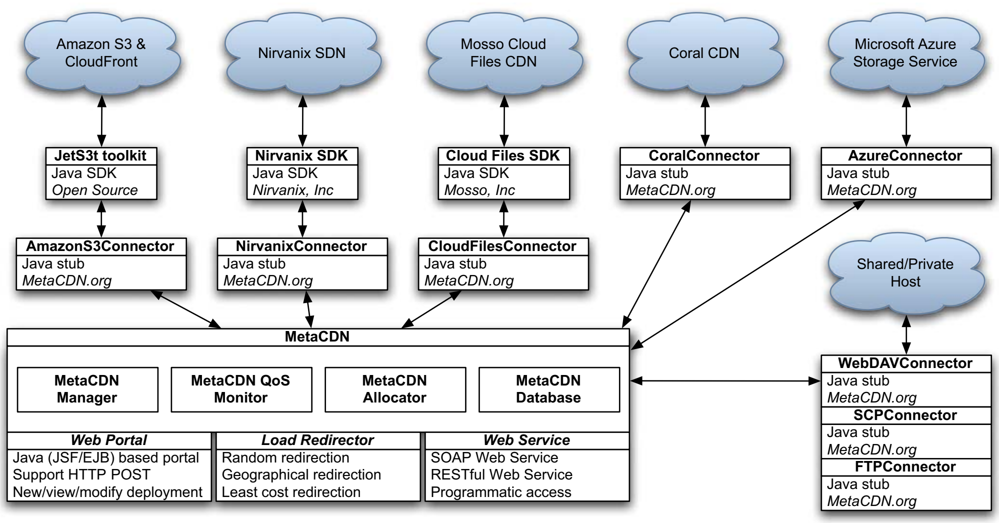

Fig. 2. MetaCDN.

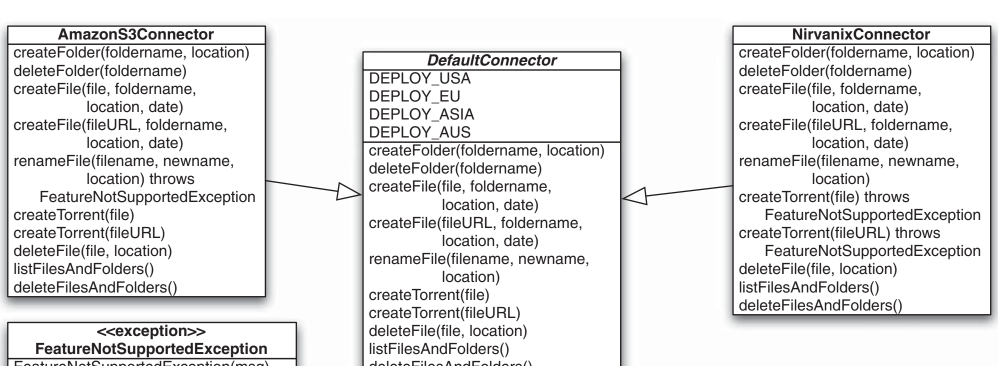

Fig. 3. MetaCDN connectors.

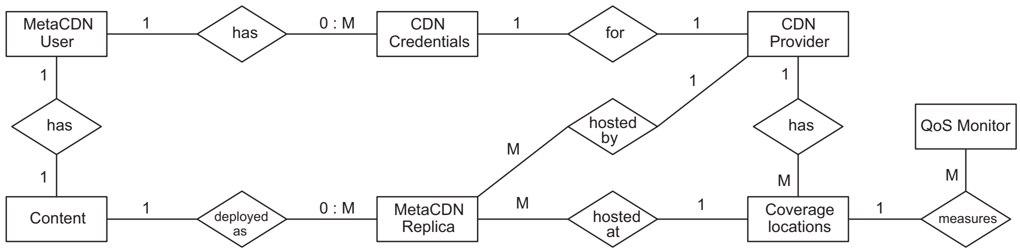

Fig. 4. MetaCDN database entity relationship.

The *MetaCDN Database* stores crucial information needed by the MetaCDN system, such as MetaCDN user details, their credentials for various Storage Cloud and other providers, and information tracking their (origin) content and any replicas made of such content. Usage information for each replica (e.g. download count and last access) is recorded in order to track the cost incurred for specific content, ensuring it remains within budget if one has been specified. The database also tracks logistical details regarding the content storage and delivery providers utilised in MetaCDN, such as their pricing, SLA offered, historical performance and their coverage locations. The MetaCDN Database Entity Relationship is depicted in Fig. 4, giving a high-level semantic data model of the MetaCDN system.

The *MetaCDN Allocator* allows users to deploy files either directly (*uploading* a file from their local file system) or from an already publicly accessible origin website (*sideloading* the file, where the backend storage provider pulls the file). It is important to note that not all backend providers support sideloading, and this is naturally indicated to users as appropriate. MetaCDN users are given a number of different deployment options depending on their needs, regardless of whether they access the service via the web portal or via web services. It is important to note that the deployment option chosen also dictates the load redirection policy that directs end-users (consumers) to a specific replica. The available deployment options include:

1. Maximise coverage and performance, where MetaCDN deploys as many replicas as possible to all available locations. The replicas used for the experiments in Section 4 were deployed by MetaCDN using this option. *The MetaCDN Load Redirector directs end-users to the closest physical replica.*
2. Deploy content in specific locations, where a user nominates regions and MetaCDN matches the requested regions with providers that service those areas. *The MetaCDN Load Redirector directs end-users to the closest physical replica.*
3. Cost optimised deployment, where MetaCDN deploys as many replicas in the locations requested by the user as their storage and transfer budget will allow, keeping them active until that budget is exhausted. *The MetaCDN Load Redirector directs end-users to the cheapest replica to minimise cost and maximise the lifetime of the deployment.*
4. Quality of service (QoS) optimised deployment, where MetaCDN deploys to providers that match specific QoS targets that a user specifies, such as average throughput or response time from a particular location, which is tracked by persistent probing from the *MetaCDN QoS monitor. The MetaCDN Load Redirector directs end-users to the best performing replica for their specific region based on historical measurements from the QoS Monitor.*[3]

[3] This load redirection policy is still under development.

After MetaCDN deploys replicas using one of the above options, it stores pertinent details such as the provider used, the URL of the replica, the desired lifetime of the replica, and the physical location (latitude and longitude) of that deployment in the *MetaCDN Database*. A geolocation service (either free[4] or commercial[5]) is used to find the latitude and longitude of where the file is stored.

The *MetaCDN QoS Monitor* tracks the performance of participating providers (and their available storage and delivery locations) periodically, monitoring and recording performance and reliability metrics from a variety of locations, which is used for QoS optimised deployment matching. Specifically, this component tracks the historical response time, throughput, hops and HTTP response codes (e.g. 2*XX*, 3*XX*, 4*XX*, or 5*XX* which denotes success, redirection/proxying, client error or server error) of replicas located at each coverage location. This information is utilised when performing a quality of service optimised deployment (described previously), and in the near future this information will be harnessed for QoS-aware load redirection.

This component also ensures that upstream providers are meeting their service level agreements, and provides a logging audit trail to allow end-users to claim credit in the event the SLA is broken. This is crucial, as you cannot depend on the backend service providers themselves to voluntarily provide credit or admit fault in the event of an outage. In effect, this keeps the providers 'honest', and due to the agile and fluid nature of the system, MetaCDN can redeploy content with minimal effort to alternative providers that can satisfy the QoS constraints, if available.

The *MetaCDN Manager* has a number of housekeeping responsibilities. First, it ensures that all current deployments are meeting QoS targets of users that have made QoS optimised deployments. Second, it ensures that replicas are removed when no longer required (i.e. the 'deploy until' date set by the user has expired), ensuring that storage costs are minimised at all times. Third, for users that have made cost optimised deployments, it ensures a user's budget has not been exceeded, by tracking usage (i.e. storage and downloads) from auditing information provided by upstream providers.

The *MetaCDN Load Redirector* is responsible for directing MetaCDN end-users (i.e. content consumers) to the most appropriate file replica. When a MetaCDN user deploys content, they are given a single URL, in the format http://www.metacdn.org/FileMapper?itemid=XX, where XX is a unique key associated with the deployed content. This provides a single namespace which is more convenient for both MetaCDN users (content deployers) and end-users (content consumers), and offers automatic and totally transparent load balancing for the latter. Different load balancing

[4] Hostip.info is a community-based project to geolocate IP addresses, and makes the database freely available.

[5] MaxMind GeoIP is a commercial IP geolocation service that can determine information such as country, region, city, postal code, area code and longitude/latitude.

and redirection policies can be utilised, including simple Random allocation, where end-users are redirected to a random replica; geographically aware redirection, where end-users are redirected to their physically closest replica; least-cost redirection, where end-users are directed to the cheapest replica from the content deployers' perspective; and QoS aware redirection, where end-users are directed to replicas that meet certain performance criteria, such as response time and throughput. The load balancing and redirection mechanism is depicted in Fig. 5, for an example scenario where a end-user in the East Coast of the United States wishes to download a file. The user requests a *MetaCDN URL* such as http://www.metacdn.org/FileMapper?itemid=1, and the browser attempts to resolve the base hostname, www.metacdn.org. The authoritative DNS (A-DNS) server for this domain resolves this request to the IP address of the closest MetaCDN portal—in this case www-na.metacdn.org. The user (or more typically their web browser) then makes a HTTP GET request for the desired content on the MetaCDN gateway. In the case of geographically aware redirection, the MetaCDN Load Redirector is triggered to select the closest replica for the end-user, in an effort to maximise performance and minimise latency. MetaCDN utilises a geolocation service (mentioned previously) to find the geographical location (latitude and longitude) of the end-user, and measures their distance from each matching replica using a simple spherical law of cosines, or a more accurate approach such as the Vincenty (1975) formula for distance between two latitude/longitude points, in order to find the closest replica. Whilst there is a strong correlation between the performance experienced by the end-user and their locality to replicas (which is demonstrated in Section 4), there is no guarantee that the closest replica is always the best choice, due to cyclical and transient fluctuations in load on the network path. As such, we intend to investigate the effectiveness of more sophisticated active measurement approaches such as CDN-based relative network positioning (CRP) (Su et al., 2008), IDMaps (Francis et al., 2001), or OASIS (Freedman et al., 2006) to ensure end-users are always directed to the best performing replica.

4. Performance evaluation

In order to evaluate the performance of the MetaCDN system (and the storage providers it utilises), we deployed test files of size 1 KB, 1, 10, and 100 MB on all nodes currently available to us, in order to test the throughput and response time of these data sources. These files were deployed by the *MetaCDN Allocator* which was instructed to maximise coverage and performance, and consequently the test files were deployed on all available nodes. As noted in the previous section, the default MetaCDN load redirection policy for this deployment option is to redirect end-users to the physically closest replica. For the first two experiments, the replicas are accessed directly, but for illustrative purposes the closest replica is denoted by Φ in each case. On Amazon, we could utilise one node in the United States (Seattle, WA) and one node in Ireland (Dublin). Nirvanix provides two nodes in the United States (both in CA), one node in Singapore and one node in Germany. The test files were also cached where possible using Coral CDN. The file is replicated by Coral to participating Coral proxy nodes on an as-needed basis, depending on where the file is accessed from.

We deployed clients in Australia (Melbourne), France (Sophia Antipolis), Austria (Vienna), United States (New York and San Diego), and South Korea (Seoul). Each location has a high speed connection to major internet backbones to minimise the chance of the client being the bottleneck during this experiment. The experiment was run simultaneously at each client location over a 24 h period, during the middle of the week. As the test spans 24 h it experiences localised peak times in each of the geographical regions. Each hour, the client sequentially downloads each test file from each available node a total of 30 times, for statistical significance. The file is downloaded using the unix utility, `wget`, with the `–no-cache` and `–no-dns-cache` options to ensure that for each download, a fresh file is always downloaded (and not sourced from any intermediary cache) and that the DNS lookup is not cached either.

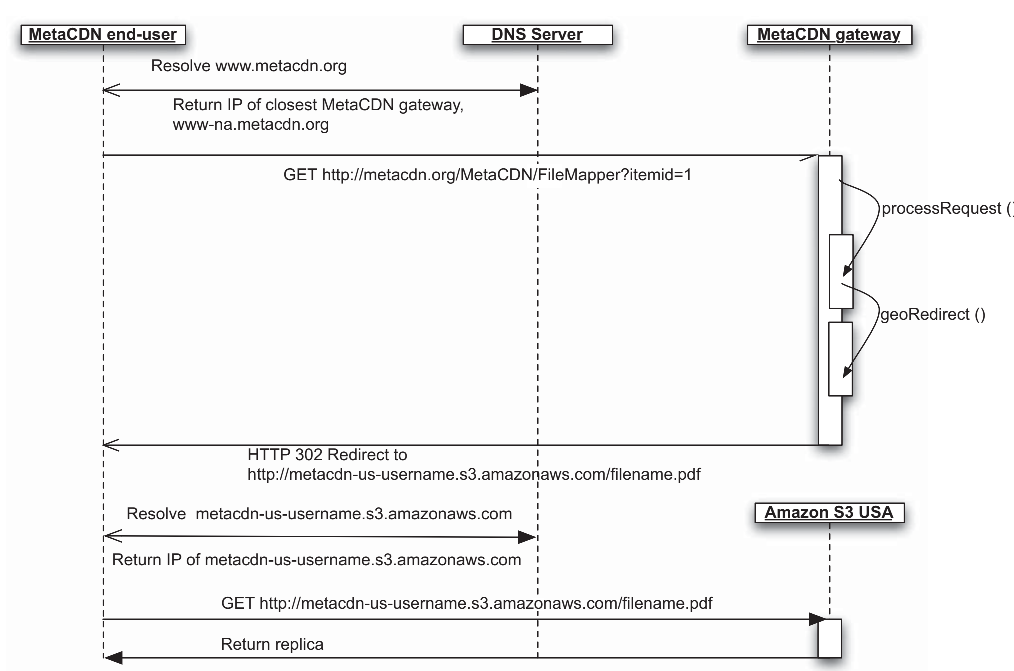

Fig. 5. MetaCDN load redirector.

In the interests of brevity, we present three sets of results. The first set of results shows the transfer speed to download each replicated 10 MB file from all client locations. The file is large enough to have some confidence that a steady-state transfer rate has been achieved. The second set of results capture the end-to-end response time when downloading each replica of a 1 KB file from all client locations. Due to the size of the file being negligible, the response time is dominated by the time taken to lookup the DNS record, and establish the HTTP connection. The third set of results compares the end-to-end response time when accessing a 1 KB MetaCDN replica file directly or via the MetaCDN Load Redirection facility (triggered by using a MetaCDN URL).

4.1. *Comparing throughput of Storage Clouds utilised by MetaCDN*

In Fig. 6, we show the throughput obtained when downloading the 10 MB test file, replicated on 7 sites, from 6 different client

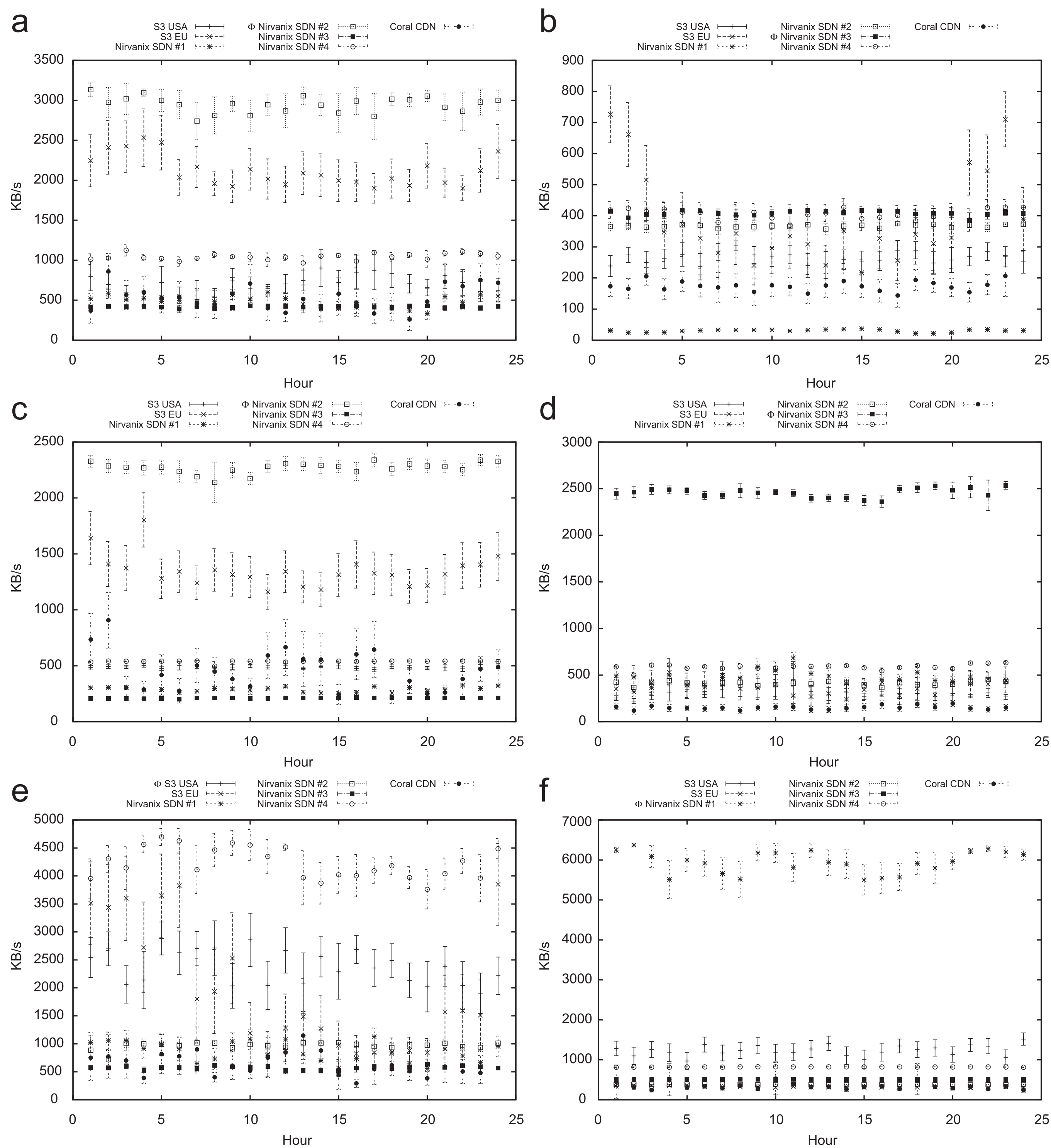

Fig. 6. Average throughput over 24 h from 6 client locations (*Φ* denotes closest replica): (a) Sophia Antipolis, France, (b) Melbourne, Australia, (c) Austria, (d) Seoul, Korea, (e) New Jersey, United States, and (f) San Deigo, United States.

locations. The average throughput per hour is shown, as well as the 95% confidence intervals in each instance, to give us an indication of the consistency and variability of the throughput over time. Unsurprisingly, we note in nearly all instances that the highest throughput is achieved from replicas located close to the client location, in both physical distance and network distance.

From Sophia Antipolis, France, the client is able to achieve sustained high throughput from the Nirvanix SDN node 2 (located in Germany) and the Amazon S3 node (located in Ireland), achieving speeds of approximately 3.0 and 2.25 MB/s, respectively. In comparison to the speeds achieved from the nodes located in the USA, it is obvious that clients in France benefit significantly by having a nearby replica. The next best result is from another Nirvanix node (located in California) at approximately 1.0 MB/s, with the remainder achieving around 500 KB/s.

The results from Melbourne, Australia are of interest given that Australia is not as highly connected as Europe or North America, depending on a small number of expensive international links to

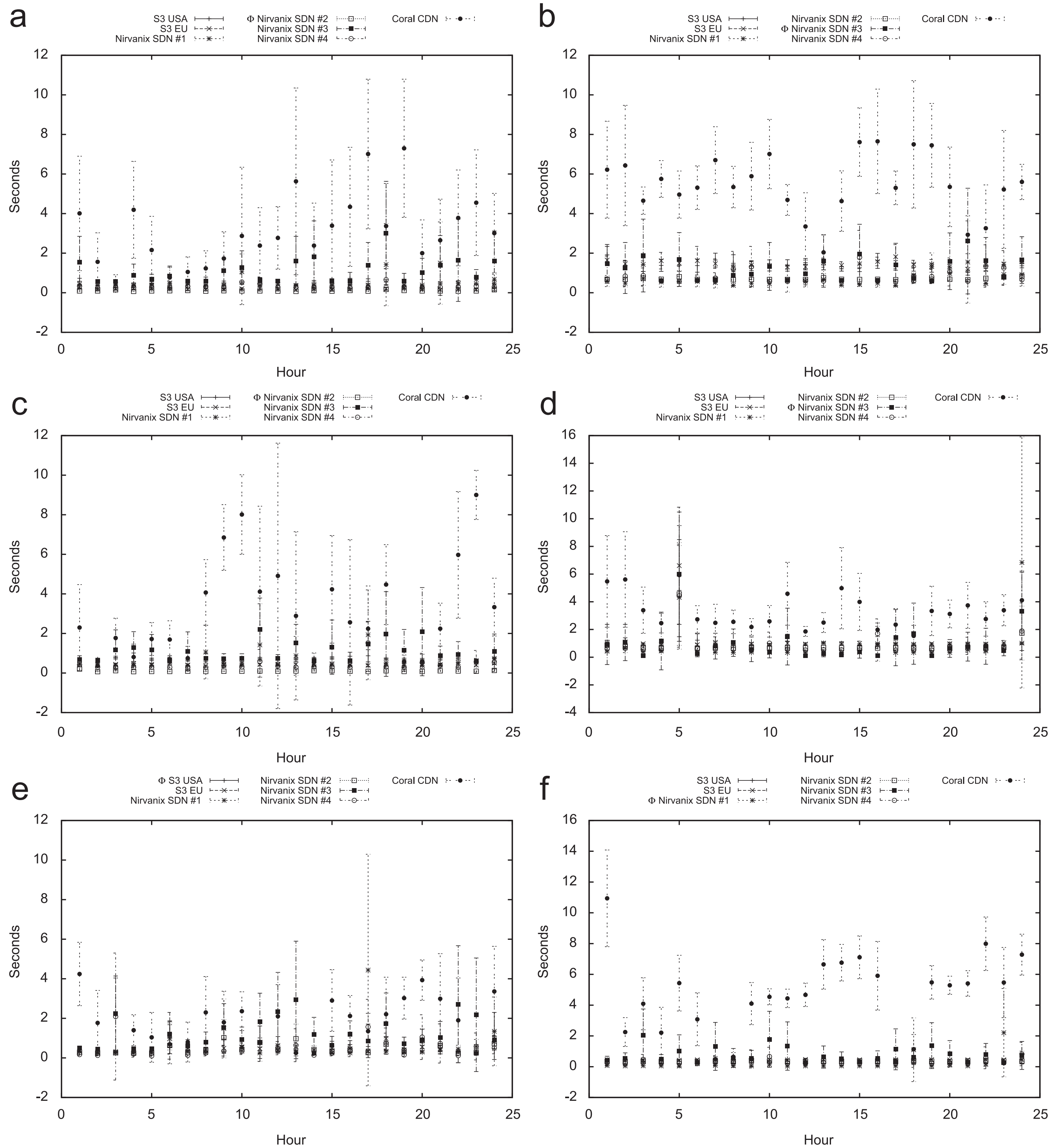

Fig. 7. Average response time over 24 h from 6 client locations (*Φ* denotes closest replica): (a) Sophia Antipolis, France, (b) Melbourne, Australia, (c) Austria, (d) Seoul, Korea, (e) New Jersey, United States, and (f) San Deigo, United States.

major data centers in Europe and the USA. We can see that for the most part, the client in Melbourne was best serviced by the node closest to it (Nirvanix node 3 in Singapore), with throughput consistently around 400 KB/s. Interestingly, in the first and last 4 h of the test, the best throughput is obtained from the replica located at Amazon's EU datacenter.

In Vienna, Austria, the best performance is obtained from the nearest replica (Nirvanix node in Germany), with the throughput consistently reaching around 2.25 MB/s, followed by the replica located at Amazon EU, with 1.5 MB/s and considerably higher variance in the download speed. The remainder of the replicas achieved download speeds between 250–500 KB/s.

From Seoul, Korea, the nearest replica in Singapore achieves a high, tightly bound throughput of 2.5 MB/s. The remainder of the replicas located in North America and Europe result in throughputs of approximately 100–500 KB/s.

High throughput was achieved from the two clients located in the United States. From New Jersey, the client achieves a throughput of approximately 4.2 MB/s from Nirvanix node 1 and 2.5 MB/s from Amazon S3, although there is some variation in the throughput, likely resulting from the fact that the replicas are located on the West Coast of the United States. During the same time period, from San Diego the throughput from the replica on Nirvanix node 1 achieves a steady, tightly bound throughput of around 6.0 MB/s, which is unsurprising given that the replica is located in the same state. The replica located on Amazon S3 USA achieves around 1.0 MB/s, with the remainder of the replicas below this point.

4.2. Comparing response time of Storage Clouds utilised by MetaCDN

In Fig. 7, we show the end-to-end throughput obtained when downloading the 1 KB test file, which captures the response time of the replica servers. The average response time per hour is shown, as well as the 95% confidence intervals in each instance, to give us an indication of responsiveness of the replica servers and the user experience when accessing these replicas.

The response times from Sophia Antipolis, France and Vienna, Austria both consistently averaged under 0.15 s for the Nirvanix replica in Germany and under 0.40 s for the Amazon replica in Ireland. In this instance (and from all other client locations), the Coral nodes generally have the worst, and the most highly variable response time. This is due to the nature of the Coral service, which at the DNS resolution stage returns a set of servers for the client to choose from. This set generally changes each time a DNS request has been made, and appears to be an attempt to spread the load rather than return the fittest server each time. The remaining nodes in the US still give a reasonable response time of around 0.5 s from both France and Austria.

In Melbourne, Australia, the closest node (Nirvanix node 3) has no appreciable improvement in response time over the Amazon and Nirvanix nodes located in the US, and is in fact worse in some instances (despite having higher throughput). A similar situation occurred in with the client located in Seoul, Korea, who experienced quite variable response time from the Nirvanix node in Singapore (which gave it the highest throughput). The response time from the replicas located in the United States and Europe gave consistent response times in comparison.

From the two client locations in the US, response time was generally good and tightly bound, averaging well under a second response time. There were some notable exceptions from the New Jersey client but these were likely temporary network perturbations as they weren't closely correlated with results seen from the San Diego or any other clients.

4.3. Evaluating the performance of MetaCDN load redirection

Fig. 8 shows a comparison of the response time experienced by a client located in New Jersey, United States, and Melbourne, Australia, when accessing a MetaCDN replica (of size 1 KB) directly or via the MetaCDN Load Redirection facility (using a MetaCDN URL, described in Section 3.1). As described previously, as the size of the file is negligible, the response time is dominated by the time taken to lookup the DNS record, and establish the HTTP connection(s). In both Figs. 8(a) and (b) the replica accessed directly or via redirection is the closest physical replica. The intent of the experiment is to determine the delay incurred by the MetaCDN Load Redirection facility, which adds an additional DNS lookup and HTTP connection, on top of the time taken to compute the best replica URL to return. It should be stressed that the client is highly unlikely to know the address of the closest replica *a priori*, so the intent of this experiment is to evaluate the overhead of the load redirection facility rather than being a fair 'apples-to-apples' comparison. During this experiment there was only one MetaCDN gateway running (in Melbourne, Australia).

From Fig. 8(a) we can see that when accessing a MetaCDN URL (and subsequently using the MetaCDN Redirection facility), the response time for a client in New Jersey (United States) is approximately twice that of accessing the replica directly. Accessing the closest replica directly took 0.41 s on average over the 24 h period measured, as opposed to 0.9 s on average when using the MetaCDN Redirection facility. This delay would be

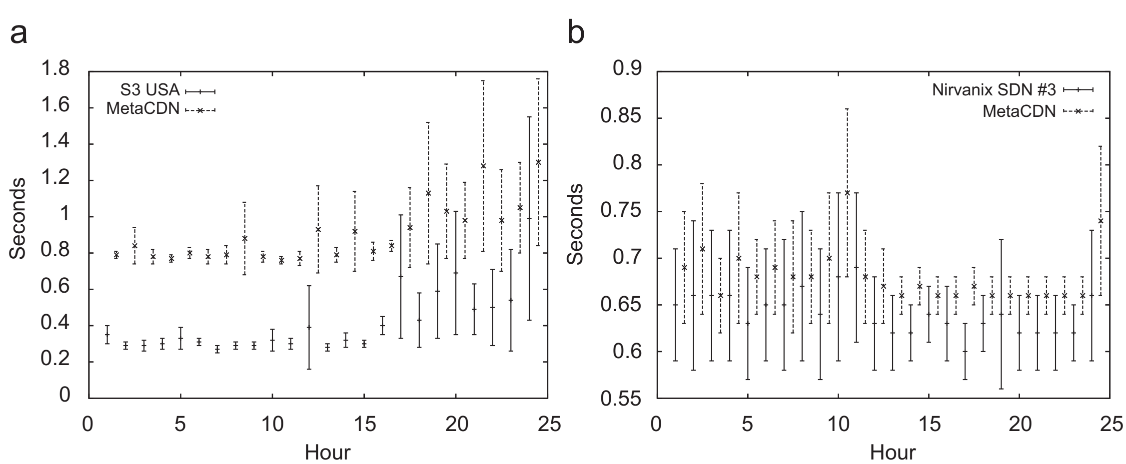

Fig. 8. Direct replica access versus MetaCDN load redirection: (a) New Jersey, United States and (b) Melbourne, Australia.

inconsequential when downloading large files (as it is more important to be directed to the best replica, and any delay would be amortised) but would be noticeable when downloading small files, such as web pages or embedded images.

A client located in Melbourne, Australia is depicted in Fig. 8(b). In this instance the client is located in close proximity to a MetaCDN gateway. This comparison is of significant interest to us as we can isolate the overhead caused by general network latency (which is incurred due to the additional DNS and HTTP connections required during redirection) from the overhead caused by the actual MetaCDN Load Redirection logic computing the best replica. Accessing the closest replica directly took 0.64 s on average over a 24 h period, whilst accessing the same replica via the MetaCDN Load Redirection facility took 0.68 s. The difference of 0.04 s suggests that the majority of the delay is incurred due to network latency from the additional DNS lookup and HTTP connection required (depicted in Fig. 5), and not the actual load redirection logic itself. Results obtained from the remaining experimental client locations in France, Austria, Korea and the United States (which are omitted here) are consistent with this observation. The obvious solution is to deploy more MetaCDN gateways to alleviate this problem, by ensuring their are gateways deployed in all major continents at the very least. This aspect is discussed in the next section.

4.4. Summary of results

Now that we have surveyed the performance of different Storage Cloud providers and the MetaCDN gateway, we are confident they provide the necessary performance to be utilised for reliable content delivery. Performance was especially good when there was a high degree of locality between the client and the replica servers, which was evident from client nodes in Europe, United States and Korea. The client in Australia had reasonable throughput and response time but would certainly benefit from more localised storage resources. In all, we found the results to be consistent (and in some cases better) in terms of response time and throughput with previous studies of dedicated (and costly) content delivery networks (Johnson et al., 2001; Su et al., 2006; Su and Kuzmanovic, 2008). However, further and longer term evaluation is needed before we can make any categoric claims.

Currently, a single MetaCDN gateway is running in Melbourne, Australia, whilst the system is still under active development. From the experiments conducted in Section 4.3, it is obvious that more gateways are needed to improve the performance and responsiveness for MetaCDN end-users (consumers). Additional MetaCDN gateways are planned for deployment in Europe, North America and Asia before MetaCDN launches as a public service, in order to ensure that users in all major continents have a responsive local gateway, and to allow the MetaCDN system to effectively scale-out, removing any single point of failure or bottleneck.

5. Future work

MetaCDN is currently under active testing and development and is rapidly evolving. Additional Storage Cloud resources are expected to come online now and in the near future, improving performance and expanding the coverage footprint of MetaCDN further. Mosso's Storage Cloud offering, Cloud Files, has recently launched, whilst Amazon have expanded their content delivery footprint to additional locations in the United States, Europe and Asia via their CloudFront service. Microsoft have also announced their Cloud Storage offering, Azure Storage Service, which is currently available only as a limited community technology preview. MetaCDN has recently been updated to support both the CloudFront and Azure Storage services. Due to the flexible and adaptable nature of MetaCDN, it is well poised to support any changes in existing Storage Cloud services as well as incorporating support for new providers as they appear.

However, it is likely that many locations on the so-called 'edges' of the internet may not have local Storage Cloud facilities available to them for some time, or any time in the foreseeable future. So far, most Storage Cloud infrastructure has been located in Europe, North America and Asia. However, MetaCDN users can supplement these 'black spots' by adding storage for commercial shared hosting providers (available in most countries) as well as privately run web hosting facilities thanks to the MetaCDN connectors for FTP, SCP/SSH and WebDAV accessible web hosting providers. These non-cloud providers can be seamlessly integrated into a MetaCDN user's resource pool and utilised by the MetaCDN system, increasing the footprint of the MetaCDN service and improving the experience of end-users via increased locality of file replicas in these areas.

In future work we intend to better harness the usage and quality of service metrics that the system records in order to make the MetaCDN system truly autonomic, improving the utility for content deployers and end-users. MetaCDN tracks the usage of content deployed using the service at the content and replica level, tracking the number of times replicas are downloaded and the last access time of each replica. We intend to harness this information to optimise the management of deployed content, expanding the deployment when and where it is needed to meet increases in demand (which are tracked by MetaCDN). Conversely, we can remove under utilised replicas during quiet periods in order to minimise cost whilst still meeting a baseline quality of service level. From the end-users (consumers) perspective, we are expanding the quality of service tracking to include data gathered from probes or agents deployed across the Internet to improve end-users' experience. These agents operate at a variety of geographically disparate locations, tracking the performance (response time, throughput, reliability) they experienced from their locale when downloading replicas from each available coverage location. This information will be reported back to their closest MetaCDN gateway. Such information can assist the MetaCDN Load Redirector in making QoS aware redirections, as the client's position can be mapped to that of a nearby agent in order to approximate the performance they will experience when downloading from specific coverage locations. As mentioned in Section 3, we are also investigating other active measurement approaches for QoS-aware client redirection.

6. Conclusion

The recent emergence of 'Storage Cloud' providers has tantalised content creators with content storage and delivery capabilities that were previously only obtainable by those who could afford expensive content delivery networks, such as Akamai and Mirror Image. However, they can be daunting to use for non-developers, as each service is best utilised via specific web services or programmer API's, and have their own unique quirks. Furthermore, these 'Storage Cloud' providers are merely basic storage services, and do not offer the capabilities of a fully featured CDN such as intelligent replica placement, automatic replication, failover, load redirection and load balancing. In this paper we presented MetaCDN, a simple, general purpose, reusable service that allows content creators to leverage the services of multiple 'Storage Cloud' providers as a unified CDN. MetaCDN makes it trivial for content creators and consumers to harness the performance and coverage of such providers by offering a single unified namespace that makes it easy to integrate into origin websites, and is transparent for end-users. We demonstrated that the performance of the MetaCDN service (and the 'Storage Clouds' it utilises) is compelling enough to utilise as a platform for high performance, low cost content delivery for content producers and consumers. Up-to-date information on MetaCDN can be found at http://www.metacdn.org.

Acknowledgements

This work is supported by Australian Research Council (ARC) as part of the Discovery Grant 'Coordinated and Cooperative Load Sharing between Content Delivery Networks' (DP0881742, 2008–2010). Thanks goes to Kenneth Chiu (SUNY Binghamton), Alexandre di Costanzo, SungJin Choi, Suraj Pandey (University of Melbourne), Sriram Krishnan (San Diego Supercomputing Centre), and Hyunsik Choi (Korea University) for their assistance with providing client nodes to run the experiments from.

References

Broberg J, Venugopal S, Buyya R. Market-oriented grids and utility computing: the state-of-the-art and future directions. Journal of Grid Computing 2008;6(3):255–76.

Elson J, Howell J. Handling flash crowds from your garage. In: USENIX 08: 2008 USENIX annual technical conference, June 2008.

Fiuczynski M. PlanetLab: overview, history, and future directions. ACM SIGOPS Operating Systems Review 2006;40(1):6–10.

Francis P, Jamin S, Jin C, Jin Y, Raz D, Shavitt Y, et al. Idmaps: a global internet host distance estimation service. IEEE/ACM Transactions on Networking 2001;9(5): 525–40.

Freedman M, Freudenthal E, et al. Democratizing content publication with coral. In: ACM symposium on networked systems design and implementation. ACM Press; 2004.

Freedman MJ, Lakshminarayanan K, Mazières D. Oasis: anycast for any service. In: NSDI'06: Proceedings of the third conference on networked systems design & implementation, USENIX Association, Berkeley, CA, USA; 2006. p. 129–42.

Gottfrid D. Self-service, prorated super computing fun! OPEN: all the code that is fit to printf(); 2007. Accessible from ⟨http://open.nytimes.com/2007/11/01/self-service-prorated-super-computing-fun/⟩.

Johnson K, Carr J, Day M, Kaashoek M. The measured performance of content distribution networks. Computer Communications 2001;24(2):202–6.

Jungle Disk, Inc. Jungle Disk: reliable online storage powered by Amazon S3. Accessible from ⟨http://www.jungledisk.com⟩.

MacAskill D. Scalability: set Amazon's servers on fire, not yours. In: ETech 2007: O'Reilly emerging technology conference; 2007. Accessible from ⟨http://blogs.smugmug.com/don/files/ETech-SmugMug-Amazon-2007.pdf⟩.

Maggs B, Technologies A. Global internet content delivery. In: First IEEE/ACM international symposium on cluster computing and the grid; 2001. p. 12.

Matei Ripeanu AI. S4: a simple storage service for sciences. In: 16th IEEE international symposium on high performance distributed computing (HPDC)—hot topics track; June 2007.

New Dream Network, LLC. Web hosting by DreamHost. Accessible from ⟨http://www.dreamhost.com⟩.

Palankar M, Onibokun A, Iamnitchi A, Ripeanu M. Amazon S3 for science grids: a viable solution? In: Fourth USENIX symposium on networked systems design & implementation (NSDI'07); April 2007.

Pathan M, Buyya R. A taxonomy of CDNs. Content Delivery Networks 2008: 33–78.

Rayburn D. Cdn pricing: costs for outsourced video delivery. In: Streaming media west: the business and technology of online video, September 2008. Available at ⟨http://www.streamingmedia.com/west/presentations/SMWest2008-CDN-Pricing.ppt⟩.

Su A, Choffnes D, Kuzmanovic A, Bustamante F. Drafting behind Akamai (travelocity-based detouring). ACM SIGCOMM Computer Communication Review 2006;36(4):435–46.

Su A-J, Choffnes D, Bustamante FE, Kuzmanovic A. Relative network positioning via cdn redirections. In: International conference on distributed computing systems; 2008. p. 377–86.

Su A-J, Kuzmanovic A. Thinning Akamai. In: IMC '08: Proceedings of the eighth ACM SIGCOMM conference on internet measurement. New York, NY, USA: ACM; 2008. p. 29–42.

Vincenty T. Direct and inverse solutions of geodesics on the ellipsoid with application of nested equations. Survey Review 1975;22(176):88–93.