MetaCDN: Harnessing Storage Clouds for High Performance Content Delivery

James Broberg[1] and Zahir Tari[2]

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

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 harnessing existing infrastructure provided by 'storage cloud' providers, at a fraction of the cost. MetaCDN is a system that leverages several existing 'storage clouds', creating an integrated overlay network that provides a low cost, high performance content delivery network for content creators. MetaCDN intelligently places content onto one or many storage providers based on the quality of service, coverage and budget preferences of participants.

The MetaCDN System

Numerous 'storage cloud' providers (or 'Storage as a Service') exist that can provide coverage in several continents, offering Service Level Agreement backed performance and uptime promises for their services. Customers are charged based on their utilisation of storage and transfer of content, which is typically in the order of cents per gigabyte. Whilst these emerging services have reduced the cost of content storage and delivery by several orders of magnitude, they can be difficult to use for non-developers, as each service is best utilised via unique web services or programmer API's. Furthermore, a customer may need coverage in more locations than offered by a single provider. To overcome this, MetaCDN utilises numerous storage providers in order to create an overlay network that can be used as a high performance, reliable and geographically distributed CDN.

The MetaCDN system integrates with each storage provider via a *connector* that provides an abstraction to hide the complexity arising from the differing provider interfaces. An abstract class, *DefaultConnector*, is defined that prescribes the basic functionality that each provider could be expected to support, that must be implemented for all existing and future connectors. These include basic operations like creation, deletion and renaming of files and folders. If an operation is not supported on a particular service, then the connector for that service should throw a *FeatureNotSupportedException*.
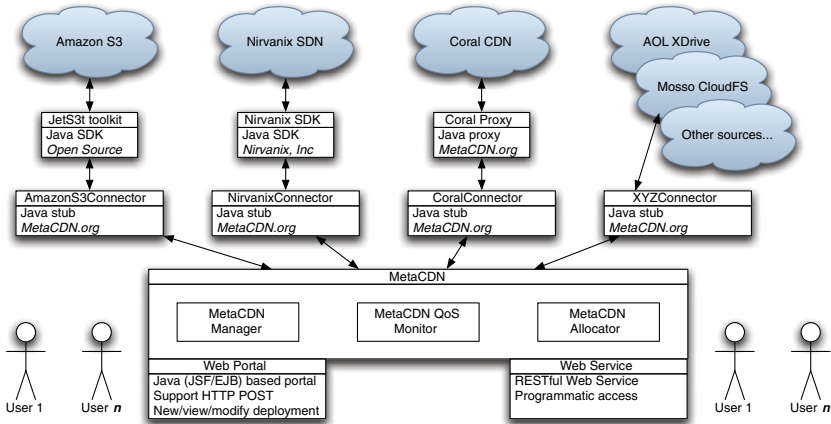
Fig. 1. MetaCDN

The service (depicted in Figure 1) is presented to end-users as a web portal for small or ad-hoc deployments (which is the focus of this paper) or as SOAP and RESTful Web Services (currently under development) for integration of customers with more complex and frequently changing content delivery needs. The web portal was developed using Java Enterprise and Java Server Faces (JSF) technologies, with a MySQL back-end to store user accounts, deployments, and the capabilities and pricing of service providers.

The MetaCDN system offers a number of functions via the web portal interface[1], including: the creation of an account in the MetaCDN system, where a user registers their details, as well as credentials for any service providers they wish to utilise; manual deployment of content to geographical regions of the user's choice; deployment of file replicas to numerous geographically distributed locations based on a user's storage and transfer budget; viewing and modifying existing content deployment; and viewing the physical location of deployed content replicas as a Google Maps overlay.

A number of features are currently under active development, including: matching and deployment of file replicas to storage providers based on quality of service parameters like uptime, average throughput and average response time for end-users located in specific geographical areas; a single URL and namespace for uploaded files, with automatic client redirection to optimal replicas; deploying large files using Bittorrent as well as HTTP; and making all MetaCDN functionality available via SOAP and RESTful Web Services.

---

[1] A screencast of the web interface is available at http://www.metacdn.org