Nimbus: A Personal Cloud Storage Manager

Malte Schwarzkopf
supervised by Steven Hand

Cloud Storage
(and why it doesn’t currently work)

Increasingly, computation is happening in “the cloud”. At the same time, we are keeping more and more of our personal data online only.

• Ubiquitously accessible
• Secure storage (no worries about backups)
• Ability to give others access

Motivating Measurements

A multi-level cloud cache using an EC2 instance

Write-through Cache accessed using HTTP (squid [2])

• using LRU replacement policy
• memory and disk cache
• spawned on demand
• supports any backend

Important results:

• Variance in performance can be contained and diminished.
• Performance for a working set can be improved through simple, low-cost measures.
• The key: give more optimization control to the user.

Underlying problem: stuck between generality and specificity
(by Brewer’s conjecture [1], we cannot have a system that is optimal with regards to all of availability, consistency and partition tolerance)

Lack of Trust

Users worry about giving all their data to a cloud provider as they fear lock-in and abuse.

A solution: Nimbus

Personal Active Cloud Storage Manager

• Ideally, we would like a unified way of accessing data stored with a variety of federated services.
• Preliminary architecture:

The Nucleus is a small piece of meta data, stored in the cloud.
• holds service mappings and credentials (secure & encrypted)
• downloaded to bootstrap a Nimbus client

On demand, a Nimbus controller can be launched, which arbitrates between backends and performs active storage management.

Gives us generality while maintaining control!

Completely optional and compatible with existing services

Active Cloud Storage
and its many benefits

Backend Federation

• Multiple heterogeneous backends accessed through a unified interface

Extensible Architecture

• New backends can be added simply by writing a Nimbus driver using their existing APIs
• Platform to implement new concepts, e.g. parasitic storage.

User Control

• Data is held by many different services
• Aggregation is controlled by the user

Encryption, Compression, Transcoding

• Computational power of active storage controllers means that we can process data as it is retrieved.
• Could implement an incremental versioning system, or transcode media files on-the-fly for different clients.

Motivating Measurements

A multi-level cloud cache using an EC2 instance

Write-through Cache accessed using HTTP (squid [2])

• using LRU replacement policy
• memory and disk cache
• spawned on demand
• supports any backend

Important results:

• Variance in performance can be contained and diminished.
• Performance for a working set can be improved through simple, low-cost measures.
• The key: give more optimization control to the user.

http://www.cl.cam.ac.uk/~ms705/