Building a Scalable Storage Infrastructure for Geneological Research

Jordan A. Nielsen
Ancestry.com Storage Architect

General Outline

- Introducing Ancestry.com
- Content Acquisition & Pipeline
- Image Challenges
- Storage Requirements
- OpenStack Swift
  - What is OpenStack Swift?
  - OpenStack Swift components
  - What is the ring?
  - How is it helping Ancestry.com

General Outline Cont…

- Testing
- Lessons Learned
- Tools to manage Swift
- Current Swift Clusters
- Want to learn more?

Introducing Ancestry.com

- Ancestry.com is the world's largest online family history resource
- 1400 employees
- 2.7 million subscribers
- 14 billion records

Introducing Ancestry.com Cont…

- "Who Do You Think You Are?" TV series
  - NBC
  - TLC

- Ancestry.com + LDS FamilySearch = 1 billion new historical records online

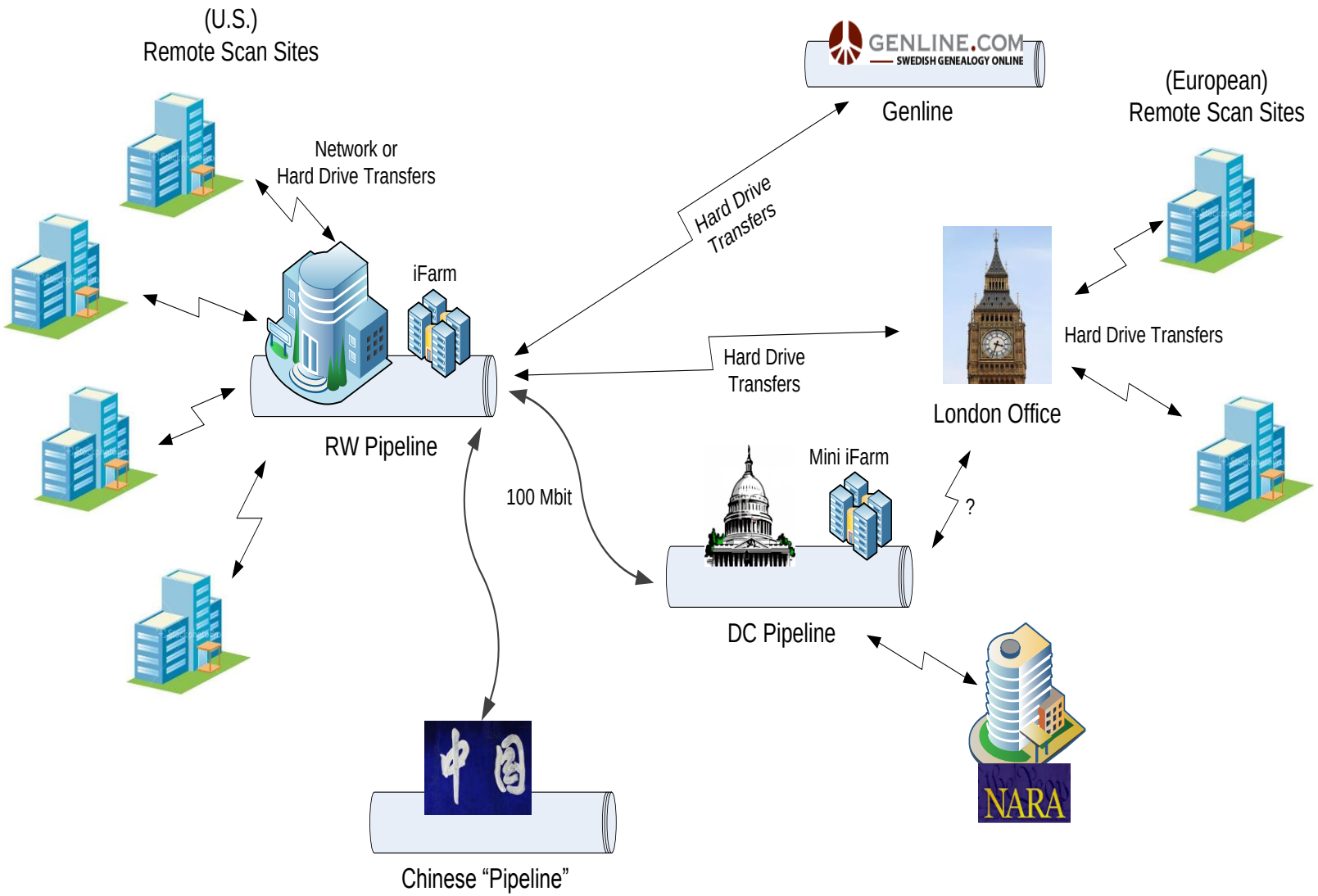
Introducing Ancestry.com Cont…

- 10PB file based storage
  - Images
  - User Contributed Content
- 2PB block storage
  - Databases
  - Virtualization

Introducing Ancestry.com Cont…

- 10-15PB archive storage
  - Original Images
  - New project

Content Acquisition



Content Pipeline

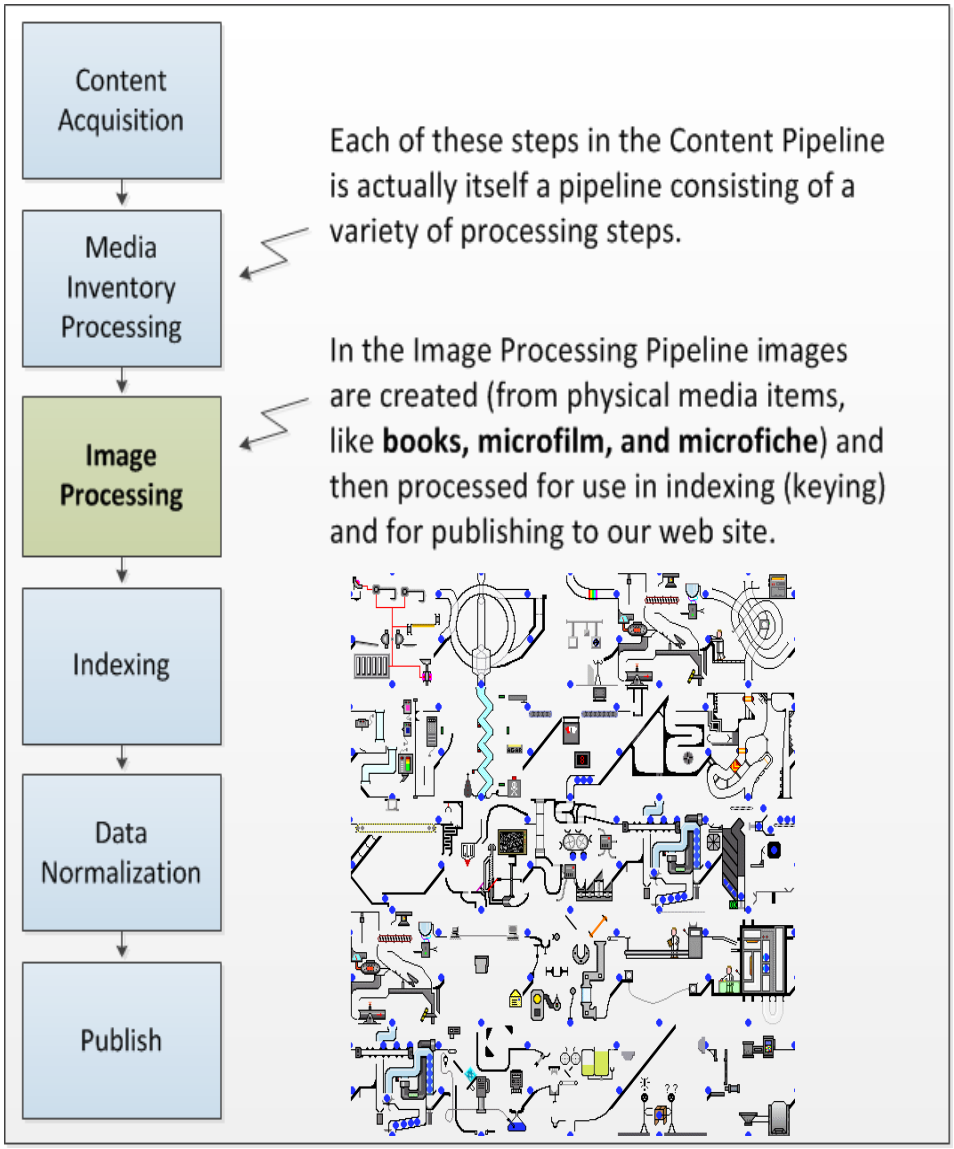


Image Processing Challenges

Image Processing Technology Challenges:

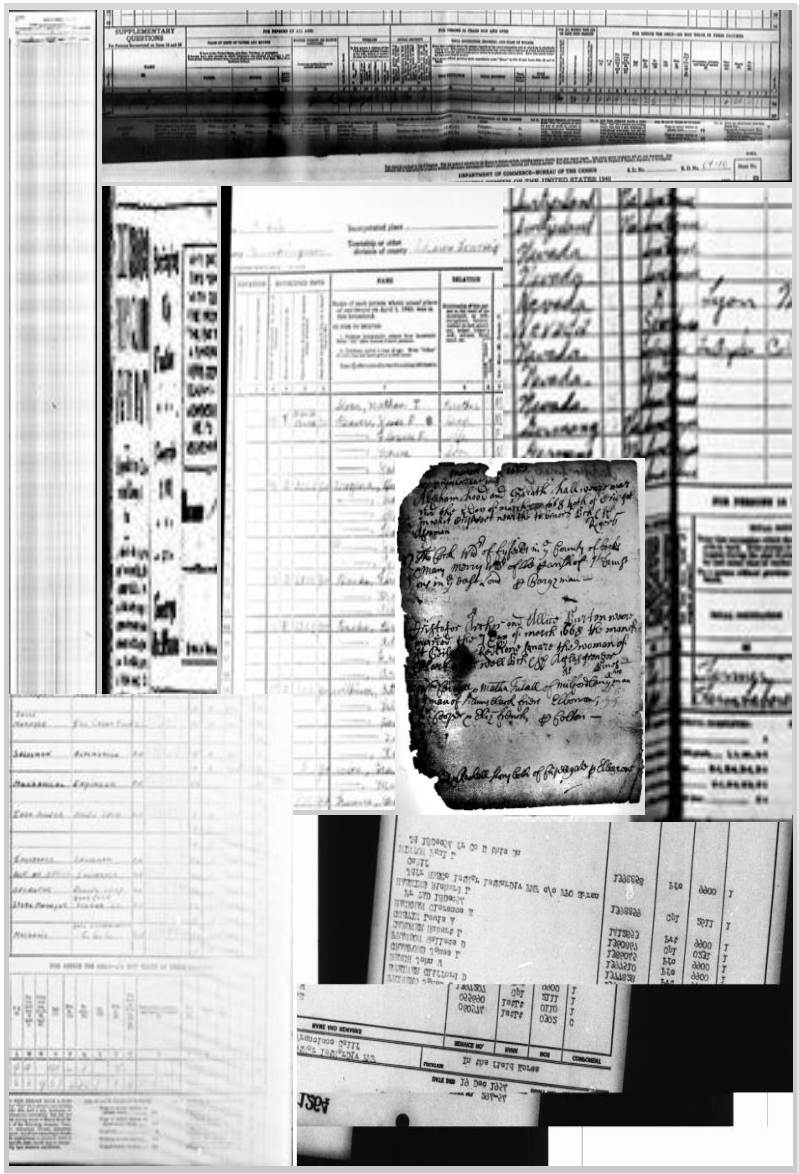
- *Volume*: Number of images we process
- *Variety*: Different kinds of forms we process
- *Quality*: Degraded and damaged source material

Variety of Material

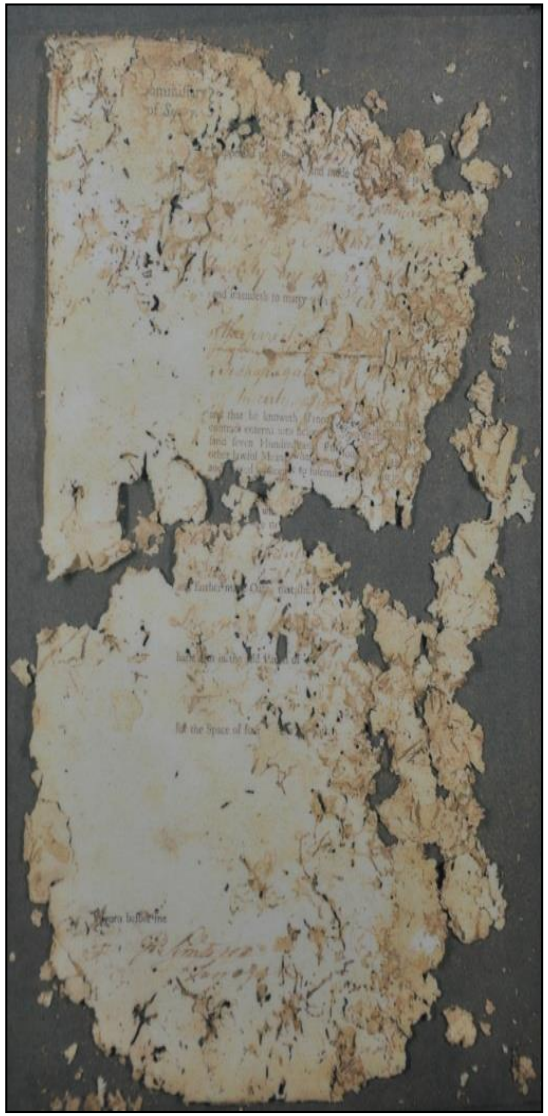
Our content comes in a variety of forms …

- Microfilm (16mm and 35mm)
- Microfiche
- Bound books
- Loose sheets
- Digital

Quality of Source Material

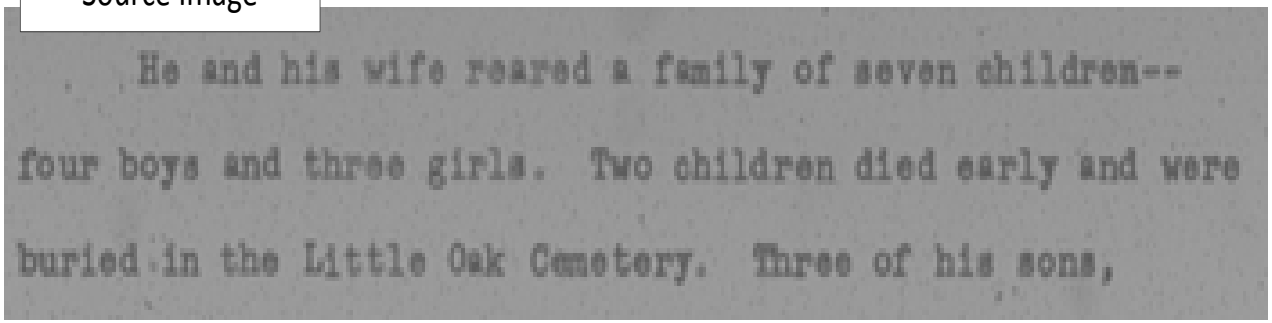


Quality of Source Material

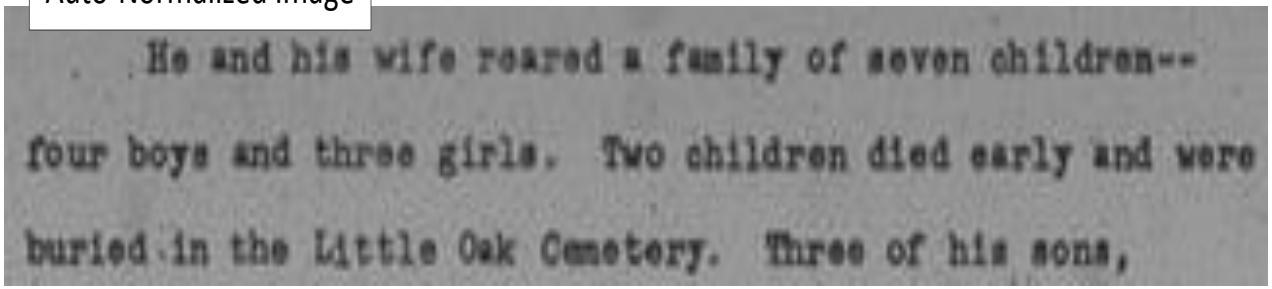


Normalize, Sharpened Image

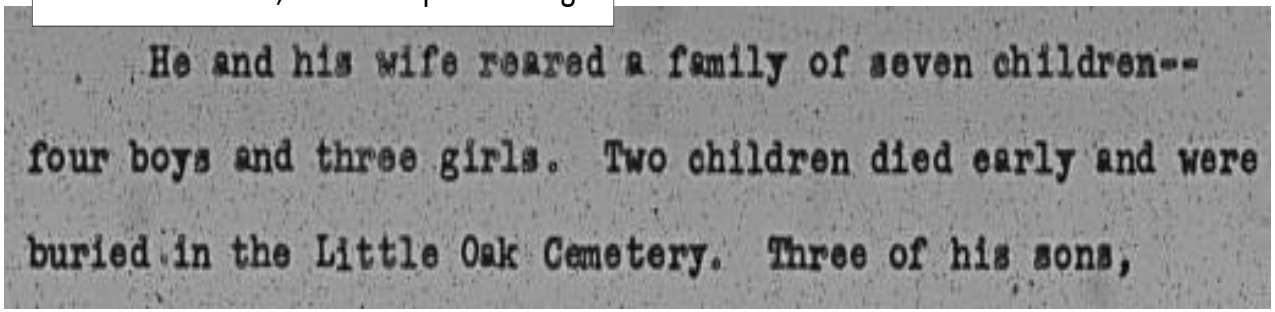
Source Image



Auto-Normalized Image

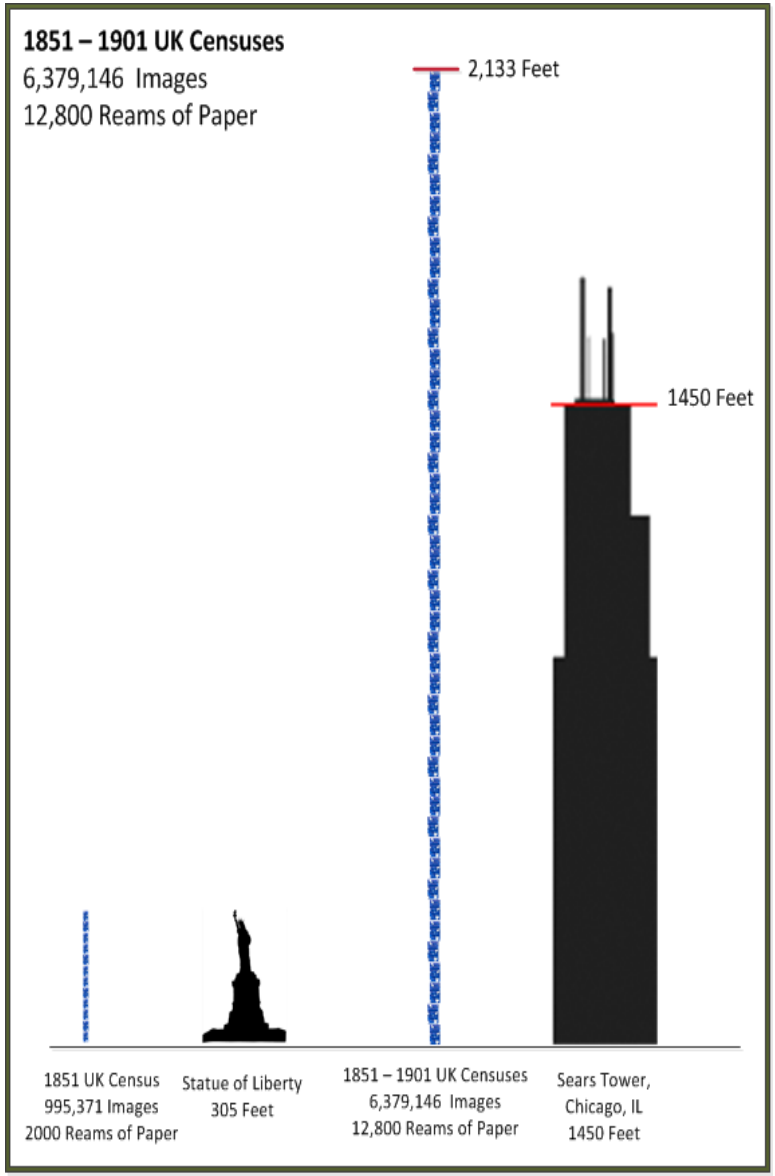


Auto-Normalized, Auto-Sharpened Image

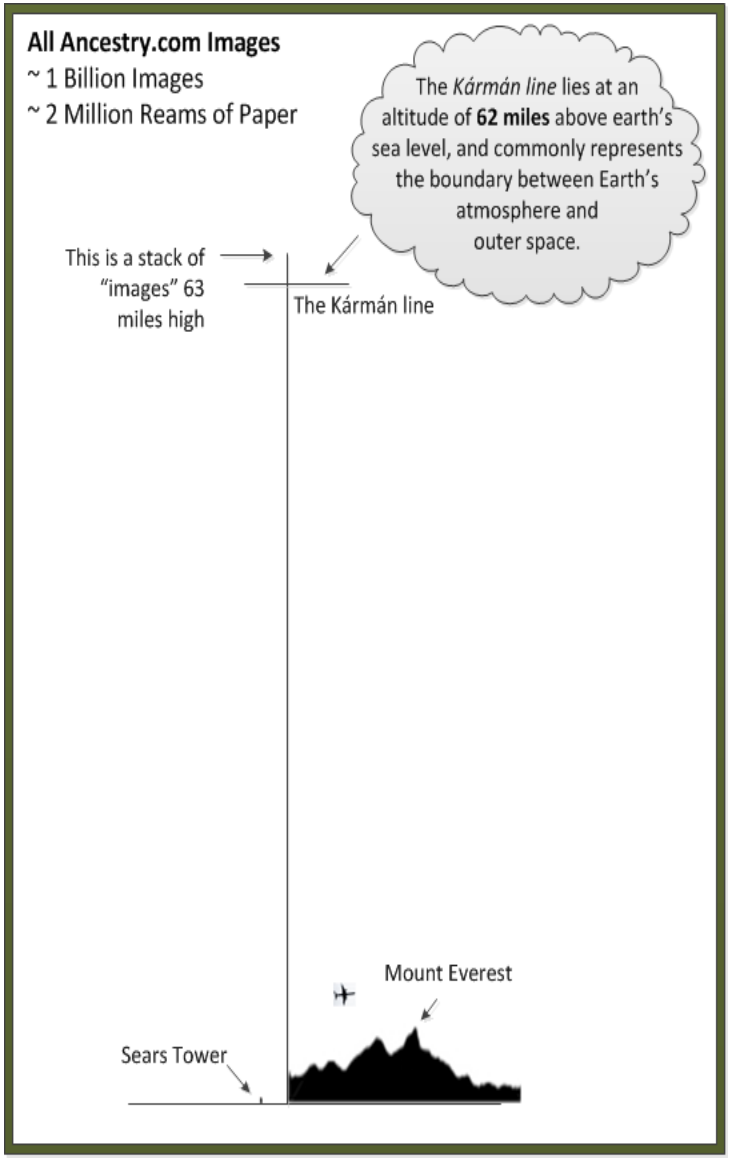


Volume of Material

Volume of Material Cont…



Volume of Material Cont…



How should we store all of this data?

Ancestry's Storage Requirements

- Use Open-source software
- Minimize vendor lock-in
- Build a shared nothing architecture
- Build an Infrastructure-as-a-Service (IAAS)
- Offer on-demand storage resources
- Leverage commodity based hardware
- Stay flexible and innovative

Ancestry's Storage Requirements Cont…

- Unstructured data stored at low cost
- Masks the differences between heterogeneous devices
- A solution that can scale to hundreds of petabytes
- Handles file management tasks such as replication, availability, and versioning
- Active-Active/HA

Ancestry's Primary Objective

Implement a robust, highly scalable, highly available, multipedibyte, multi-datacenter, software-defined open storage system that utilizes a nothing-shared architecture and runs on commodity hardware.

The answer is OpenStack Swift!

What is OpenStack Swift?

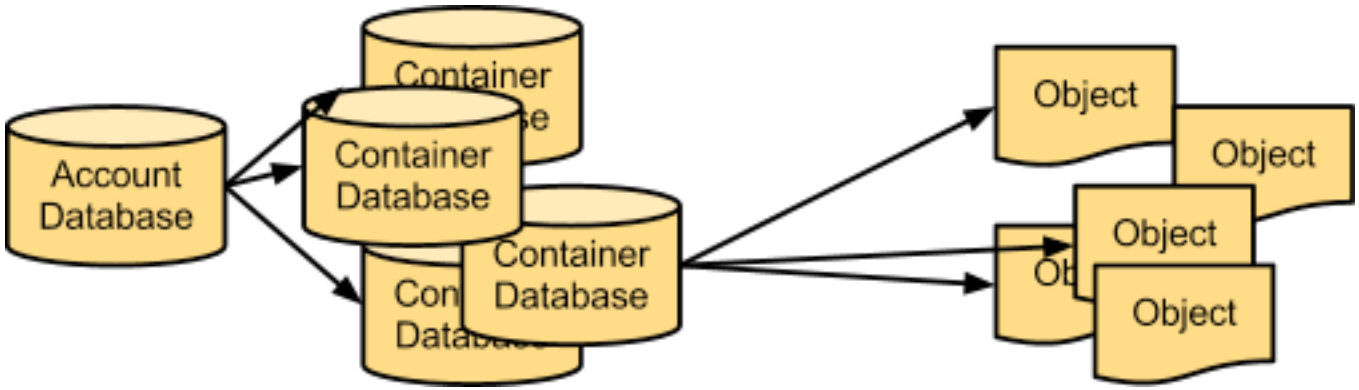
- OpenStack Swift is a highly available, distributed, eventually consistent object/blob store
- Objects are stored multiple times
  - Usually 3 replicas to protect the data
- Allows users to define failure domains

OpenStack Swift Components

- Proxy Server
  - RESTful HTTP API
  - Accepts HTTP verbs (PUT, GET, DELETE, POST)
- Account server
  - Keeps track of containers belonging to that account
  - Users have roles in accounts

OpenStack Swift Components

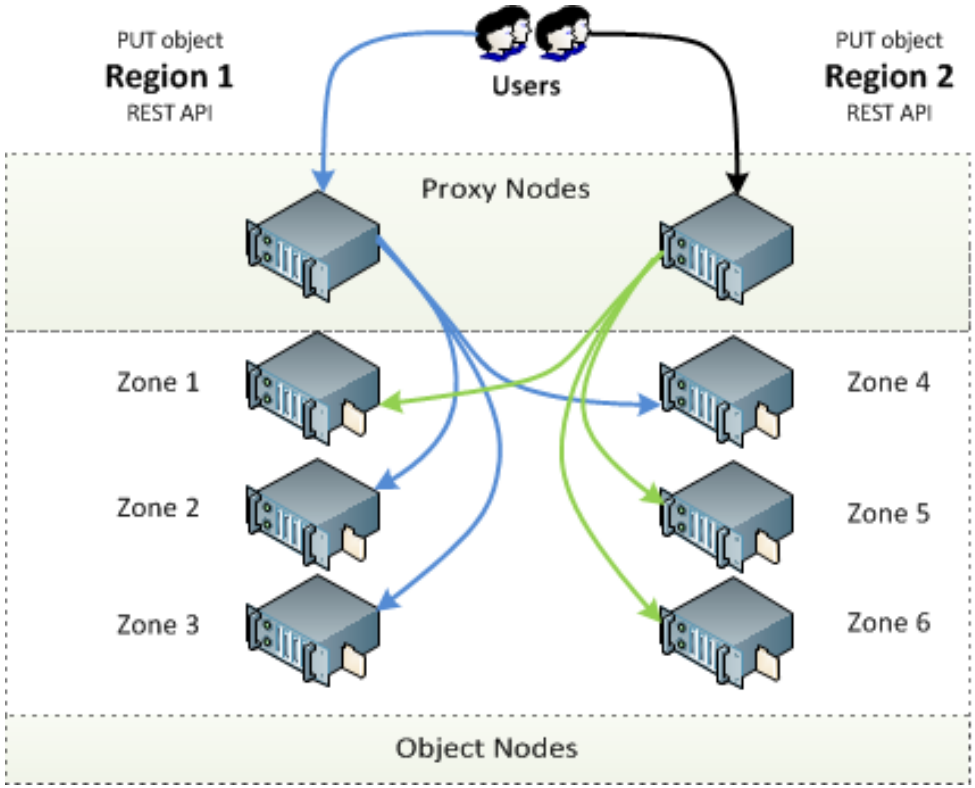
- Container server
  - Keeps track of objects in the container
- Object servers
  - Store the actual objects/data



What is the ring?

- Rings determine where data should reside in the cluster
- Separate ring for accounts, containers, and objects
- Modified consistent hashing ring

How Swift works?



Proxy Nodes tie together the rest of the Swift architecture. For each request, it looks up the location of the account, container, or object and routes the request accordingly. The public REST API is exposed through the Proxy Nodes.

The Object Node is a very simple blob storage node that can store, retrieve and delete objects stored on local devices. Objects are stored as binary files on the filesystem with metadata stored in the file's extended attributes (xattrs).

Testing Info

- Gravity SearchClient.exe
  - Simulates load from one or many clients
- SSbench
  - Swift benchmarking tool
- Swift-bench
  - Swift benchmarking tool

Testing - Significant findings

(Swift configuration)

- Throttle the Swift object auditor
  - Dropped the response time from 75+ms to 50ms. (First byte)

- Made configuration changes to throttle the auditor
  - [object-auditor]
  - files_per_second = 10
  - bytes_per_second = 1000000
  - zero_byte_files_per_second = 5

Testing - Significant findings Cont…

(Linux Tweaks)

- Linux Kernel IO scheduler modification.
  - This dropped response time from 50ms to 35ms.
- How to configure?
  - echo deadline > /sys/block/sd<a-l>/queue/scheduler (not persistent) – Good for testing!
  - Set the IO elevator to deadline in the /etc/grub.conf file

Testing - Significant findings Cont…

(Facebook Flashcache)

- Tested Facebook Flashcache on the object nodes.
- Facebook FlashCache is a general purpose writeback block cache for Linux.
- Increased performance from 35ms – 20ms.

Testing - Significant findings Cont…

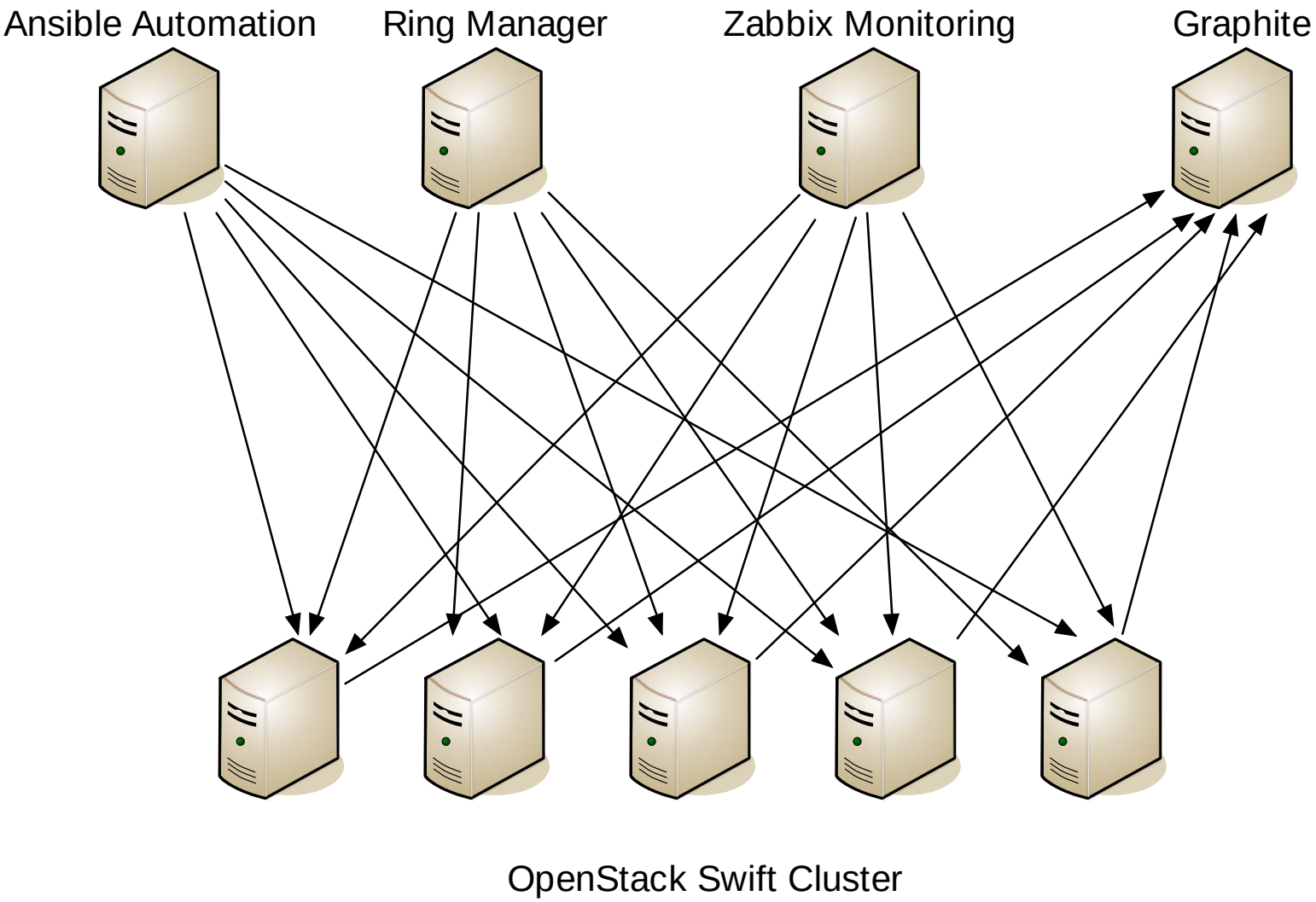
(Varnish Cache)

- Implemented Varnish Cache on Proxy Nodes
- Varnish is a very fast HTTP accelerator. It transparently caches images, CSS / Javascript files and content pages, and delivers them quickly without much overhead.
- Increased performance from 20ms – 2-3ms if data was found in cache.
- Very beneficial if a lot of the same data is accessed over and over again.

Testing & lesson

- Keystone or SWAUTH?
  - No central database like KeyStone
  - Stores authentication info in Swift
  - Implemented on each Proxy server

Tools to manage Swift



Tools to manage Swift cont…

- Grafana
  - Feature rich metrics dashboard and graph editor for Graphite
- Collectd
  - Collectd is a daemon which collects system performance statistics
  - Graphite Plugin available and default in version 5.4

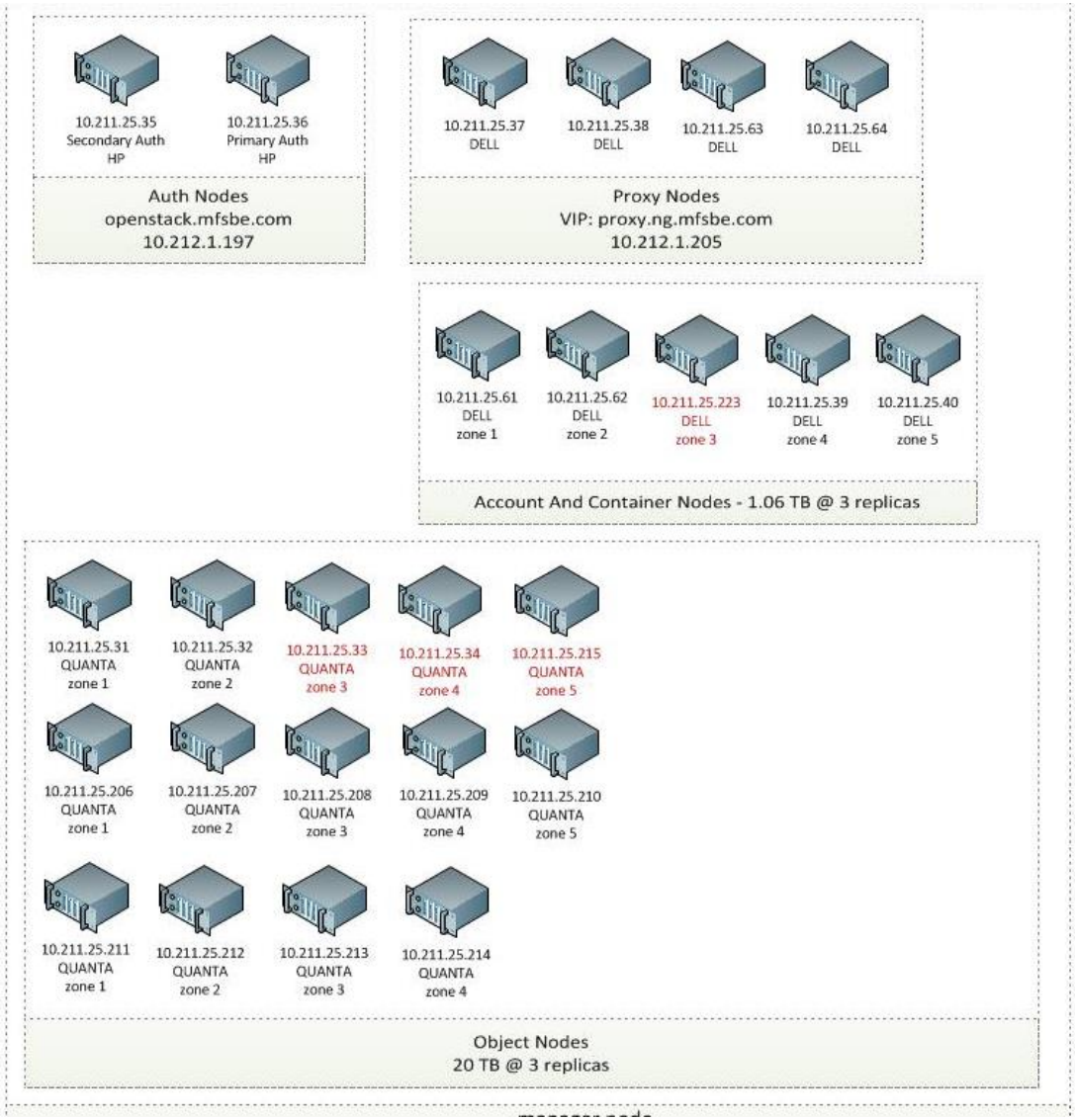
Tools to manage Swift cont…

- Collectl
  - Very nice feature rich command-line utility that can be used to collect performance data
  - Graphite plugin built in
- Statsd

Archive Cluster

- 2 – Proxy Servers
  - 2 Socket, 10 core
  - 128GB RAM
  - Bonded 10Gb Connections
- 2 – Account & Container Servers
  - 8 x 450GB SSD's
  - 2 x socket, 10 core
  - Bonded 10Gb Connections
- 4 – Object Servers
  - 1 x SD280 attached to one server
  - 84 LFF drives in 5U
  - 336TB per SC280 JBOD's
  - 1.3PB RAW per rack
  - 433TB useable (3 copies)

Tier1 Cluster



Enterprise Production Implementation

Rack Description: 45U ToR Switchs(4), Auth Nodes (1), Proxy Nodes (3), SSD Ring Nodes (4), Object Nodes (17)
Estimated Usable Swift Storage per Rack: 612TB RAW, 204TB useable (612TB/3)
Architecture will be driven by empirical data results from testing and recommendations from RackSpace.

Want to learn more?

- Jnielsen@ancestry.com
  - Email me if you have any questions!
- SwiftStack
  - Joe Arnold
  - John Dickenson (PTL Swift)
- Mirantis
  - Great OpenStack and Swift expertise
- Dell
- Redapt

Questions?