SmashFS: A File System for OneBlox

Charles Hardin

Software Engineer (most of the time)

Company Background

- Start-up funded in 2011
- Enterprise class storage for broader customer set
  - Scalable storage
  - Easy to use (useable in 5 mins!)
  - Cloud managed storage
  - Primary and Backup use cases
- Innovative technology
  - File System
  - Cloud based management system
  - Embedded CPU based hardware
  - Many patents filed for IP developed
- Based in Sunnyvale, CA
- 50 employees

OneBlox is a Product and it is for sale

OneBlox 3308

- Industry's first Scale-Out storage appliance with
  - Inline de-dupe
  - Inline encryption
  - Real-time Replication
  - Disaster recovery
  - Less than 5 minutes to set up and use
- 8 x 3.5" HDD/SSD bays
  - Up to 48TB raw using (6TB disks)
  - Supports mix of drive technologies
- Clusters up to 6 nodes (now) & 200TB
  - 32 nodes and up to > 1PB (2014)

If We Could Reinvent Storage,

What Would It Do?

Storage should Scale Out – why doesn't it?

- Seem to always talk about those blocks
  - Where is that superblock?
  - Where is that block map?
  - Where is a free block?
  - Where is an inode block?
  - What is the block size – 512, 4096, 1984?
- Seem to always talk about protection
  - What RAID level is that again – 0, 1, 3, 5, 6, 10?
  - How do I resize that RAID stripe again?
  - Does this disk work in my RAID setup?
  - Is this a JBOD or a RAID?
- Disk is full – and?
  - Filled up a 24TB shelf - ok time to buy another?
  - Turn on that compression option and forget about it
  - Send out the email for everyone to "clean up" their home directories
- Can any of this be easy?

Object Based Storage is not new

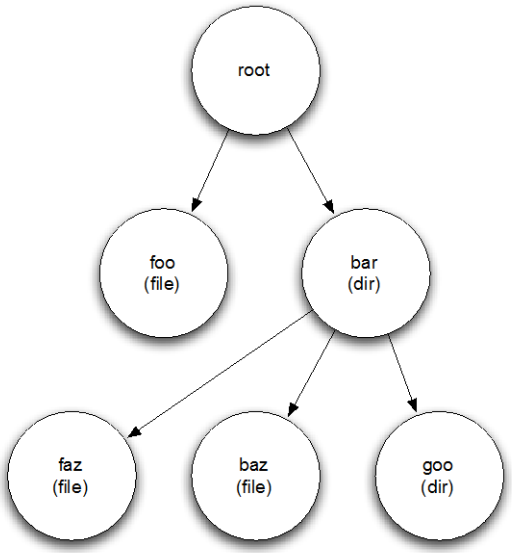
- If our content was an object then this is really about object based storage
  - Yes, because the key defines the identity to lookup the object
  - No, because the content is used to derive the key, so when the content changes the key changes for the object
- Instead of an object being “mutable”, an object is considered “immutable” because every content change invokes an identifier change that represents a new object.

Define an Object?

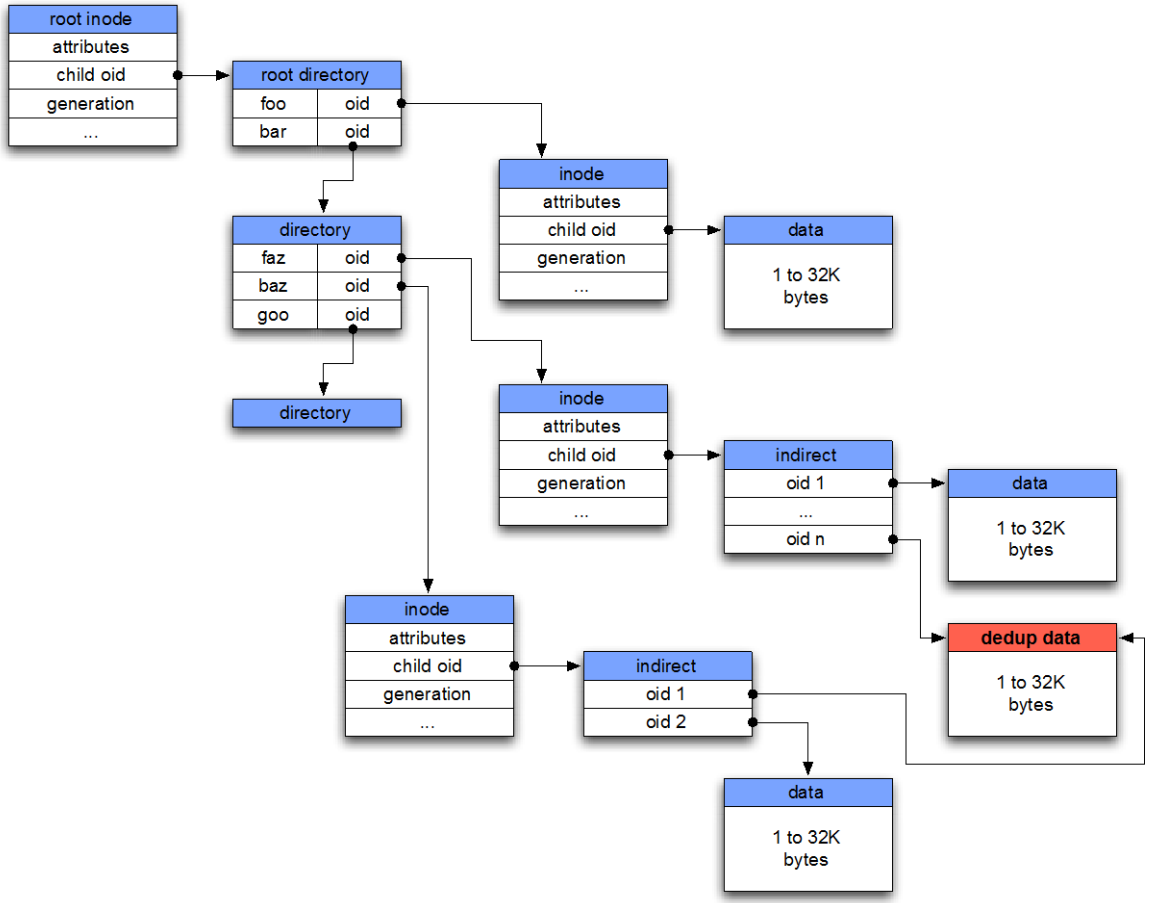
- Object
  - A tuple of a type, identifier, and content
- Object Type
  - Define an interpretation of the content as a particular file system entity (directory entry, inode, indirect block, data block, etc.)
- Object Identifer
  - The computed hash of the object content
  - obj.id = SHA1(obj.content)
- Object Content
  - A collection of arbitrary bytes that cannot exceed a maximum size of 32KB in the current implementation
- Object Attributes
  - Associated meta information about the object that is not part of the content (i.e. GC mark and sweep information)

Build a File System with Objects

FileSystem Tree



Relational Objects



Continuous Data Protection and Consistency

- CDP - If every update triggered a propagation to the root inode, then every operation would have a unique root hash
  - IO amplification based upon the depth of the tree
  - Distribution of a new root inode for every update
  - Technically doable, but is it really needed?
- Near CDP - Aggregate multiple inode updates in a window and during a “sync” operation propagate the inode(s) up the tree
  - Crash recovery is to the last “sync”
  - Propagation is deferred to amortize the cost based upon a sync time
- Can achieve CDP by the policy of propagation

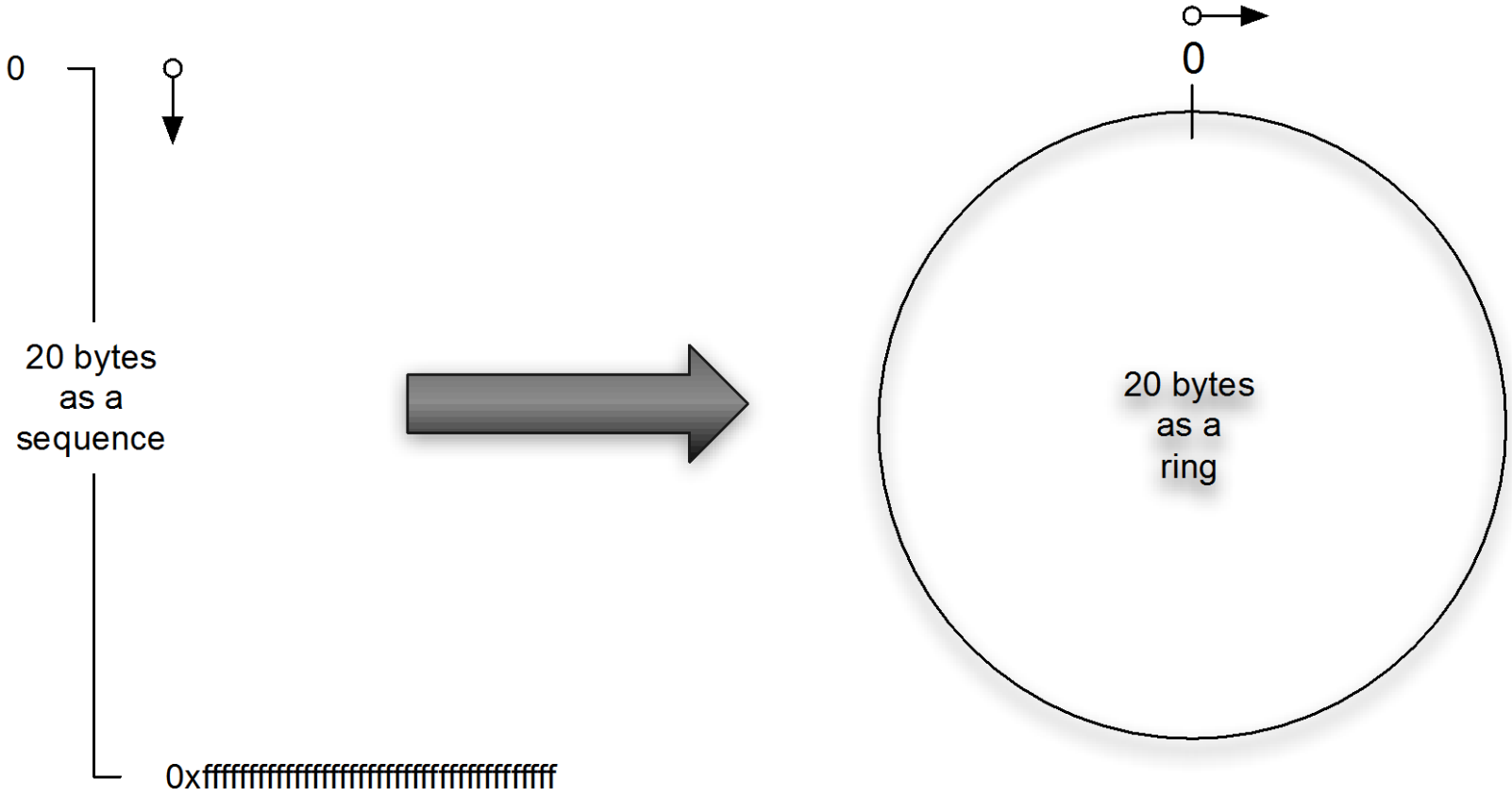
Garbage Collection

- Any object is “deleted” when it is no longer referred to by a metaroot
  - Snapshot is removed and has the last reference to an object
  - Accomplished by a mark and sweep
- Mark the “current” tree defined by the metaroot
  - Requires coordination and consensus
  - Tree traversal and mark all the objects across all snapshots
  - Share the “latest” mark
- Sweep any objects that have not been marked
  - Decentralized and deferred
  - Any object not at the “latest” mark can be reclaimed
- Feels “heavyweight” for work?

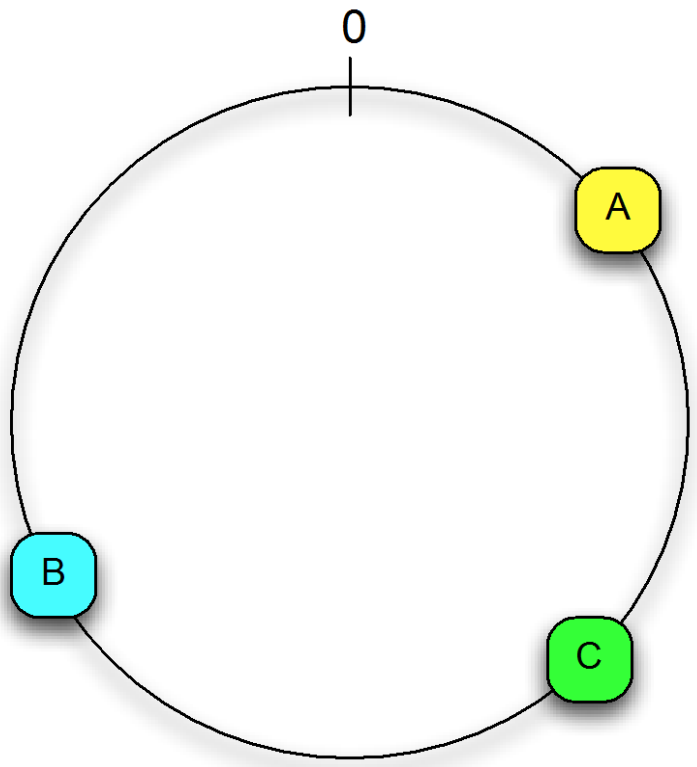
Distributed…
Right?

Objects Everywhere

Define our Object Namespace based on SHA1



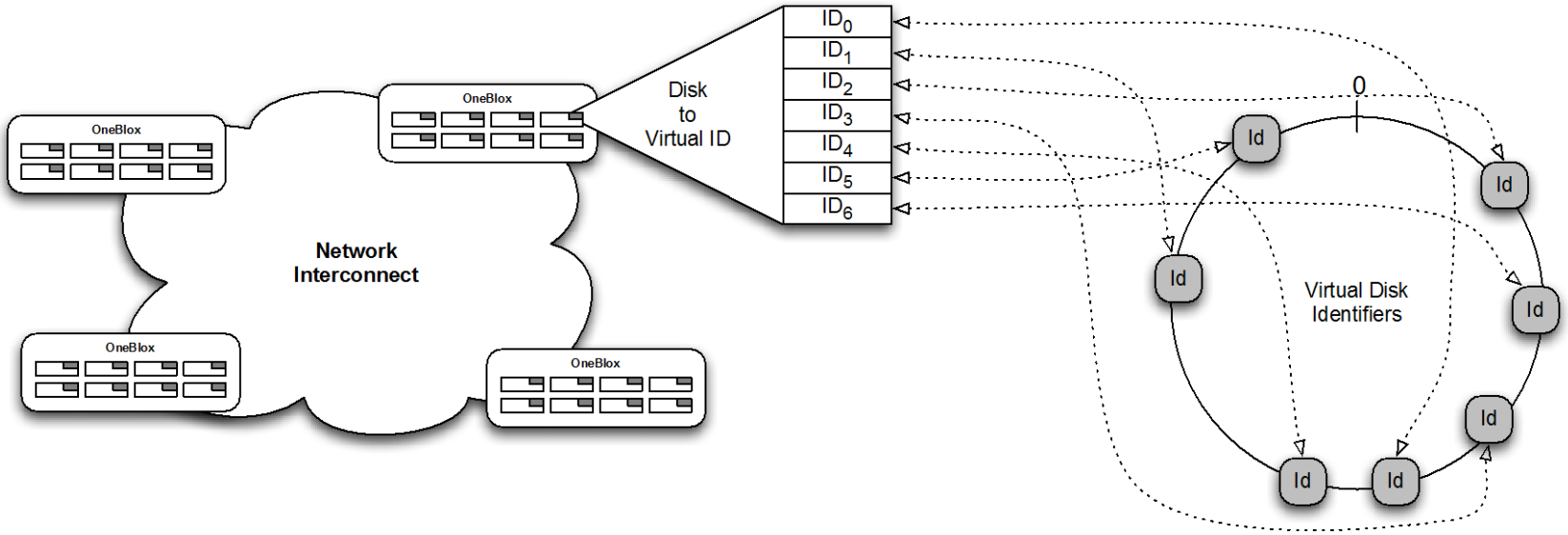
Disk Identifiers



Disk Physical Identifiers (ring location is random)

- disk.id = SHA1(disk.mfg + disk.serial)
  - "pie" is sliced randomly
- So, should there only be one identifier?
  - Not all disks are the same
  - Statistically smooth the ring

Cluster Map “Disks” to the Ring



Physical Cluster Nodes to Ring Namespace

Software Stack Simplified

Distributed File System

- POSIX Semantics
- Lease Model for Consistency

Distributed Object Store

- Routing and Ownership for Objects

Messaging Layer (MESH of TCP Connections)

Key Value Store

- Content Addressable

Disks (Reliable Storage)

Distributed File System

- Filesystem layer in each node “acquires leases” for sub-trees of the namespace
  - Based on activity (file open, dir ops)
  - Each node has dirty state of its files/dirs
  - Reads/writes directed through leasing node (only if one already exists)
  - Each node responsible for flushing on sync events
- Global Metadata Operation component (MOP) that hands out leases
  - Crash-recoverable using a distributed “whiteboard” for in-memory state
  - Serializes the File System Updates to preserve ordering on syncs across nodes
  - Owner of the “metaroot” and propagation to all the nodes of the last agreement

Late Night Reading List

- FAWN - http://www.cs.cmu.edu/~fawnproj/papers/fawn-sosp2009.pdf
- NASD - http://www.pdl.cmu.edu/ftp/NASD/Sigmetrics97.pdf
- Venti - http://en.wikipedia.org/wiki/Venti
- Chord - http://en.wikipedia.org/wiki/Chord_(peer-to-peer)
- Scatter - http://homes.cs.washington.edu/~arvind/papers/scatter.pdf
- Spanner - http://research.google.com/archive/spanner.html

Take the above and your own insights – add to a blender and puree to get your own take on SmashFS

Conclusion

- Engage in deeper relationship with CMU PDL
- Looking to hire top candidates in the following areas:
  - File Systems
  - Cloud Management
  - QA
  - Support
- Contacts:
  - Ramesh Balan, VP, Engineering
    - ramesh@exablox.com
  - Charles Hardin
    - ckhardin@exablox.com

Thank You

ExaBlox Engineering

 @exablox