Pond – The Ocean Store Prototype

Overview

– Goals
– Features
– Design
– Implementation
– Experimental Results

Goals – A Distributed File System Offering

– Incremental Scalability
  • More servers translates to more available data
– Secure Sharing
  • Access Control
– Long term durability
  • With high probability data should not be able to leave the system

Key Features

– Location Independent Routing
  • Tapestry
– Byzantine Update Agreement
  • For management of the inner ring
– Push based cache correction
  • Overlay locality aware multi-cast network
– Continuous archiving
  • Erasure codes
Design

- Two tier network
  - Upper tier composed of well connected powerful servers
    - Serialize changes to data
  - Lower tier composed of user workstations
    - Cache data
    - Archive data
    - Read / Write data

The *Data Object*

- Can be thought of as corresponding to a File
- Is composed of immutable versions
- Each version is broken into B-tree of blocks
- Is referenced by an AGUID
  - Versions by VGUID
  - Blocks by BGUID
- Can be conditionally operated on
• Retrieving Data
  – AGUID: secure hash of name and public key
  – Contact primary replica to find VGUID
  – From the VGUID retrieve BGUID’s
  – Copy the block data to the local system
  – Join the *dissemination tree*
    • Act as a cached copy

• Controlling Data
  – Primary Replica
    • Publishes AGUID to VGUID mappings
      – Digitally signs
    • Enforces access control
    • Serializes writes
    • Pushes cache updates
    • Archives data

• Writing data
  – Send a request to the primary replica
  – Replica verifies credentials
  – Checks predicates
  – Creates new VGUID and then associates data
  – Pushes update down *dissemination tree*
• Archiving Data With Erasure Codes
  – Divides data into N chunks
  – Encodes chunks to M erasure blocks
  – M > N
  – Any N of the M blocks is sufficient for reconstruction
  – Located by erasure block number and BGUID.
  – How does one know the BGUID?
    • The AGUID is unavailable?

• Primary Replica – The Inner Ring
  – Byzantine internal decisions
  – Decisions published with by public key
    • Each node has a fraction of the private key
    • Enough fractions to prove a Byzantine agreement was reached are required to sign a decision

• Inner Ring – Changing Nodes
  – Byzantine decision
    • Decides to elect
    • Decides Who to elect
    • Chooses the key set
  – Old keys are deleted
    • By Byzantine assumption, conspiring nodes do not have enough keys to publish

• The Responsible Party
  – Publishes node statistics
  – Used to nominate nodes to inner ring
  – Has no say over the actions of the inner rings
  – There could be many of them
  – Being compromised would not destroy the network
• Implementation of the Pond Prototype
  – Pros
    • 50,000 lines of Java
    • Event based between modules
    • Some modules are pluggable
    • Highly portable
  – Cons
    • Garbage collector ‘Stops The World’

Storage Overhead
  – B-Tree dominates cost of small files
  – Convergence at 32KB
  – Erasure Codes add 4.8x storage penalty

Write Latency Components
  – For small updates
    • Computing the signature dominates
  – For large updates
    • Computing the erasure fragments dominate

<table>
<thead>
<tr>
<th>Phase</th>
<th>4 kB Update</th>
<th>2 MB Update</th>
</tr>
</thead>
<tbody>
<tr>
<td>Check Validity</td>
<td>0.3</td>
<td>0.4</td>
</tr>
<tr>
<td>Serialize</td>
<td>6.1</td>
<td>26.6</td>
</tr>
<tr>
<td>Update</td>
<td>1.5</td>
<td>113.0</td>
</tr>
<tr>
<td>Archive</td>
<td>4.5</td>
<td>566.9</td>
</tr>
<tr>
<td>Sign Result</td>
<td>77.8</td>
<td>75.8</td>
</tr>
</tbody>
</table>

Tests are local to minimize network’s effect

Write Throughput
  – Increasing data size amortizes signature time
  – Approaches 8MB/s as block size grows
  – With archiving enabled
    • Performance peaks at 2.6MB/s
Propagation Efficiency
- As Replicas Increase
  - Network economy becomes more efficient
  - Less high RTT links are used
- Tests are with 10, 20, and 50 replicas
  - This is 2%, 4% and 10% of the network
  - Are these numbers likely to occur in practice?

Andrew Benchmark
- **WAN**
  - Read Performance
    - Up to 4.6x better
  - Write Performance
    - Up to 7.3x worse
- **LAN**
  - Read Performance
    - From 2x to 3x worse
  - Write Performance
    - From 8x to 80x worse

Are these tradeoffs acceptable?

Questions?