SINE NOMINE ASSOCIATES

Ubik: Failure Modes & Best Practices

Tom Keiser <tkeiser@sinenomine.net>
Steven Jenkins <sjenkins@sinenomine.net>
Sine Nomine Associates

Agenda

- What is Ubik?
- Why Ubik?
- Ubik at Scale
- Common Failure Modes
- Best Practices
- Moving Forward…
- Q&A

What is Ubik, anyway?

SINE NOMINE ASSOCIATES

- What Ubik *is*:
  - A master/slave flat file replication technology
  - A master election protocol
  - A distributed whole-file locking system
- What Ubik *isn't*:
  - A database
  - A means of ensuring referential integrity
  - Fast

But don't we call them databases?

SINE NOMINE ASSOCIATES

- So if Ubik isn't a database, what are VL & PT?
  - Flat files
    - A *stream* of C data structures
    - A collection of routines for querying and updating the data structure stream
  - *Hierarchical-model* databases
    - Use hand-coded hash table "indices" (*unclustered*)
    - Records not aligned to pages
  - Use Ubik to propagate changes from master to slaves/clones

SINE NOMINE ASSOCIATES

Why Ubik?

Ubik: Failure Modes & Best Practices

Wait. Is OS/360 state-of-the-art?

SINE NOMINE ASSOCIATES

- Aren't hierarchical databases outdated?
  - For the most part, yes
    - E.F. Codd's Relational model displaced network and hierarchical for most applications
  - The hierarchical model lacks clear distinction between logical and physical layers, and thus lost
- Why the hierarchical model?
  - Simplicity of implementation
  - RDBMS technology was expensive in the 80s

The Anachronistic Database

SINE NOMINE ASSOCIATES

- What has OpenAFS gained by going with the hierarchical model?
  - Low-cost implementation
  - Simplicity
  - Lack of dependencies on large software systems
  - Deployments don't require DBA talent
  - Ubik as a clustering technology
    - And it "just works" (well, most of the time)

The Anachronistic Database

SINE NOMINE ASSOCIATES

- What has OpenAFS lost?
  - Write Performance
  - Transactions (and thus concurrency)
  - Referential Integrity
  - Flexibility to alter schema at low cost
  - Ability to add annotations and custom metadata
  - Ability to vertically scale by purchasing a commercial DBMS
  - Time and effort spent maintaining Ubik

The Anachronistic Database

SINE NOMINE ASSOCIATES

- So what bars OpenAFS from moving forward?
  - Cost
  - Ubik works “well enough” for average sites
- Does that imply Ubik doesn’t work for large sites?
  - It works, for the most part.
  - Unfortunately, there are failure modes at scale…

SINE NOMINE ASSOCIATES

Ubik at Scale

Ubik: Failure Modes & Best Practices

How does Ubik break at scale?

SINE NOMINE ASSOCIATES

- Synchronous Architecture
  - High-latency connections to satellite offices limit write performance of entire cluster
    - Write transaction latency is proportional to sum of RTTs from sync site to each other replication site
    - This includes non-voting clones
  - No read transactions permitted while write is executing
    - Significant QoS problem…

How does Ubik break at scale?

SINE NOMINE ASSOCIATES

- File distribution and quorum election can't occur concurrently (until pthreaded ubik)
  - Leads to loss of quorum
- Limited number of replication sites due to synchronous discipline
  - Replication sites updated *serially*
  - Difficult to support extremely large cells

How does Ubik break at scale?

SINE NOMINE ASSOCIATES

- Only one write transaction at a time
  - Because we have exactly one whole-file lock
- File update topology is a star
  - Want flexible RTT/bandwidth-aware topology
  - No support for asynchronous replicas
- Database files limited to 2GB in size
  - VL: ~14.5M volume groups
  - PT: ~11.2M entries (limit is lower when membership lists are large)

VL at Scale

SINE NOMINE ASSOCIATES

- Limited to 255 servers
- Limited to 16 (IPv4 only!) addresses per server
- pthreaded Ubik project uncovered *many* bugs
- File format difficult to augment
- Data indexed by <type, volume id>
  - Less useful than indexing by <volume id>, since volume type not always known a priori
- When things break, vos syncvldb is slow

PT at Scale

SINE NOMINE ASSOCIATES

- Predates X.500, LDAP, etc.
  - A wonderful idea at the time…
  - Now it's yet another directory to populate
    - Demand for it to go away in favor of industry standards
- Many sites automatically populate PT from their core enterprise directory
- Work ongoing to improve PT
  - Others making it obsolete by developing PT proxy servers that query an enterprise directory

SINE NOMINE ASSOCIATES

Common Failure Modes

Ubik: Failure Modes & Best Practices

How do things break?

- Failure to elect a master
  - Your database stays read-only
- Database corruption
  - Sadly, we have neither transaction history logs, nor referential integrity constraints to help us
    - Restore from a backup, or vos syncvldb
- Miscellaneous bugs
  - We do occasionally find them, as many of you can attest

Master Election

SINE NOMINE ASSOCIATES

- Ensure your CellServDB files reflect reality!
  - Be *very* careful with multihomed ubik servers
    - NetInfo/NetRestrict will not save you!
  - Ubik and DHCP aren't a good combination, unless you have control over the DHCP server
- Transient network failures will break ubik
  - Ubik is sensitive to routing problems
    - Your network must be a fully connected graph!

Referential Integrity

SINE NOMINE ASSOCIATES

- Bugs/crashes can result in RI failures
- Asynchronous Referential Integrity Verifiers
  - vldb_check
  - prdb_check
- While we don't have real RI, you can (should!) use these database auditing tools to detect problems
  - Hopefully before they cause a major outage…

SINE NOMINE ASSOCIATES

Best Practices

Ubik: Failure Modes & Best Practices

Best Practices

- Don’t place quorum set members behind unreliable WAN links
  - Non-voting clones are your friends!
- Use an odd integer for quorum set size
  - Non-voting clones are your friends!
- Make backup copies of your databases
- Run prdb_check and vldb_check occasionally
- udebug is also your friend

udebug

- What does recovery state 1f really mean?
  - Hexadecimal flag for recovery status on master:
    - 0x01 – this is the sync site (master)
    - 0x02 – Cluster database version scan complete
    - 0x04 – Our local database copy is the most recent known version in the cluster
    - 0x08 – database version label is up-to-date
    - 0x10 – all ubik servers have an up-to-date copy of the database

SINE NOMINE ASSOCIATES

Moving Forward…

Ubik: Failure Modes & Best Practices

Pthreaded Ubik Project

SINE NOMINE ASSOCIATES

- Currently in 1.5 branch:
  - Ported code to POSIX threads (add locks)
  - Beacon (vote) algorithm can now run in parallel with file transfers, resolving failure mode
  - One known bug left, which we hope to have resolved by start of conference
- Work under consideration/research:
  - Allow read transactions to occur in parallel with Ubik disk writes

Is Ubik Strategic?

SINE NOMINE ASSOCIATES

- This is a tough question:
  - We have a lot of time/energy/money invested in Ubik and our hierarchical-model databases
  - However, making our current implementation scale will be quite costly
- Ultimately, we are limited by lacking abstractions provided by a modern DBMS
  - For PT, the solution may be disruptive: proxies
  - For VL, things are less clear…

SINE NOMINE ASSOCIATES

Questions?

Ubik: Failure Modes & Best Practices