SINE NOMINE
ASSOCIATES

Intro to Ubik

Mark Vitale

20 June 2019

2019 OpenAFS Workshop

What is ubik?

- A software mechanism for maintaining a replicated distributed "database"
  - Elections
    - Establish and maintain a quorum of database servers with a single sync-site
  - Locking
    - Support distributed whole-file locking
  - Commits
    - Coordinate distributed, non-blocking atomic commits
  - Recovery
    - Coordinate distribution of replicated content after disruptions
- Not a true database, but supports simple database-like semantics
  - True relational database technology was $$$$ in the 20th century

Ubik design goals

- Available: database replicated among multiple servers for load sharing and resiliency
- Atomic: no partial or incomplete commits seen by users
- Non-blocking: allow reads and writes during network partitions or single-server outages – even a sync-site outage (unlike two-phase commit)
- Consistent: automatic distributed updates; automatic recovery from crashes and failed commits
- Simple: apps should be able to use a replicated, transactional server as easily as a traditional Unix file on a single-site Unix server.

Ubik limitations (K.I.S.S)

- Only one write transaction at a time
  - Simplifies logging and recovery
- No reads during write
- No deadlock detection or protection
  - Application writer is responsible for consistent lock order
- S.L.O.W.
  - Write latency is proportional to the sum of the RTTs from sync-site to each non-sync site
  - Synchronization (although rare) is … synchronous and serial

OpenAFS ubik (“DB”) servers

• vlserver	Volume Location server
• ptserver	Protection server
• buserver	Backup server
• kaserver	Kerberos 4 KDC - obsolete

Components

- Rx stack
  - Listener thread
  - Event thread (pthread only)
  - IOMGR thread (LWP only)
- Beacon thread (ubeacon_Interact)
- Recovery thread (urecovery_Interact)
- VOTE_* RPC service threads
- DISK_* RPC service threads
- Ubik disk buffer package

Ubik server roles

- Sync-site (“master”)
- Non-sync site (“clone”)
- Non-voting clone site

Role: sync-site

- Determined by winning an election
  - OR being the sole configured voting DB server
- Default sync-site is the DB server with the lowest IP address
  - Implemented by giving an extra vote to default server
- Accepts both reads and writes
- Coordinates
  - Elections
  - Writes and commits
  - DB version synchronization

Role: non-sync site

- Determined by losing an election and/or voting for someone else
- Will not vote for another for BIGTIME 75s
- May be elected sync-site in case of sync-site failure (crash, outage, network partition, etc.)
- Accepts only reads; writes fail with UNOTSYNC

Role: non-voting clone

- Specified by square brackets in cell configuration:
  - `bos addhost <server> [clonedb]`
  - `[cloned_ip]    #cloned_host` (in CellServDB)
- "Non-voting" is a misnomer – they vote, but their votes don't count!
- Can never be elected sync-site *
- Accepts only reads; writes fail with UNOTSYNC
- Provides a local database copy for remote locations
- Elections unaffected by network delays
- Network latency still counts for updates and synchronization

Election (beacon thread)

- Sync-site (or a wannabe) sends VOTE_Beacon to each server in CellServDB using multi_Rx
  - State=1 if sync-site, 0 if wannabe
- VOTE_Beacon reply:
  - 0 NO
  - <epoch> YES, and this is my local time
    - NOTE: Because Rx sees this as a non-zero return code, the reply is sent as an RX_PACKET_TYPE_ABORT
- Tally:
  - YES from a “non-voting” clone doesn’t count
  - YES from a voting clone counts for 2 votes
  - YES for self counts for 2 votes
  - YES from the lowest IP address counts 1 extra vote
- Results: if tally > number of servers, YOU WIN

Ubik election time constants

constant	value (s)	semantics
BIGTIME	75	each site MUST promise to vote for only one sync-site within BIGTIME interval; time to wait before presuming death of other server(s)
SMALLTIME	60	successful election term limit; a sync-site will resign when the last votes received are older than this
MAXSKEW	10	allowance for clock skew between DB servers; Implicit requirement for shared timebase
POLLTIME	15	period for elections (VOTE_Beacon requests) from sync-site (or wannabe)
RPCTIMEOUT	20	Time for VOTE_Beacon RPC timeout (original implementation – current default timeout is 12s)

Ubik election invariants

- To ensure that only one site can be elected sync-site at a time, the election constants must obey these invariant relations:
  - BIGTIME > SMALLTIME
  - BIGTIME – SMALLTIME > MAXSKEW
  - SMALLTIME > RPCTIMEOUT + max(RPCTIMEOUT, POLLTIME)
  - BIGTIME > RPCTIMEOUT + max(RPCTIMEOUT, POLLTIME)

Quorum

- “quorum” is the minimum number of votes required to elect a sync-site.
- therefore, if a sync-site has been elected, we have quorum
- this is true EVEN if not all members of the quorum have the current DB yet
- reads require NEITHER quorum NOR current DB version
- writes require BOTH quorum AND current DB version

Synchonization (recovery thread)

- Maintains state of connections to other servers (all roles)
  - Every 30s, send DISK_Probe to any ”down” servers to reestablish contact
- Ensures that all sites have the same version of the database (sync-site only)
  - Every 4s, check recovery state; as needed, find latest version of database (DISK_GetVersion) and propagate it (DISK_GetFile, DISK_SendFile – NOT MULTI!)

Recovery state

- All states reflect sync-site’s viewpoint

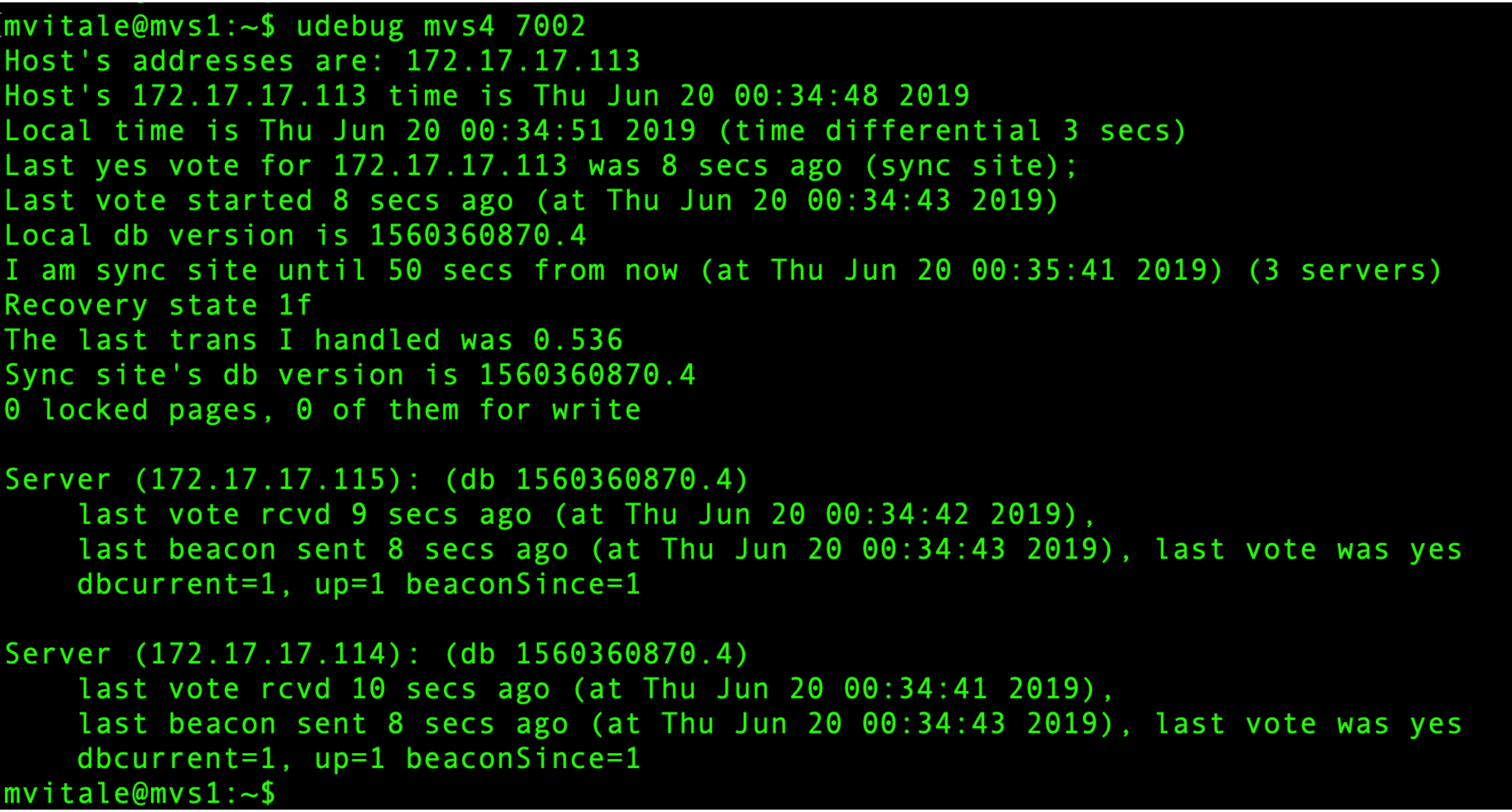
UBIK_RECSYNCSITE	0x01	I am sync site
UBIK_RECFOUNDDB	0x02	I know the best DB version
UBIK_RECHAVEDB	0x04	I have a local copy of best DB version
UBIK_RECLABELDB	0x08	I did first write commit to DB
UBIK_RECSENTDB	0x10	I have sent best DB to everyone

- udebug to the sync-site to examine the current recovery state

0x1f	Normal
0x17	Normal after new DB, before first write

udebug utility

- Useful for determining sync-site, diagnosing quorum issues:
  - udebug <server> <port>
- Specify the -long option to a non-sync server in order to obtain some additional information about the other servers (implicit default for sync-site)



```
mvitale@mvs1:~$ udebug mvs4 7002
Host's addresses are: 172.17.17.113
Host's 172.17.17.113 time is Thu Jun 20 00:34:48 2019
Local time is Thu Jun 20 00:34:51 2019 (time differential 3 secs)
Last yes vote for 172.17.17.113 was 8 secs ago (sync site);
Last vote started 8 secs ago (at Thu Jun 20 00:34:43 2019)
Local db version is 1560360870.4
I am sync site until 50 secs from now (at Thu Jun 20 00:35:41 2019) (3 servers)
Recovery state 1f
The last trans I handled was 0.536
Sync site's db version is 1560360870.4
0 locked pages, 0 of them for write

Server (172.17.17.115): (db 1560360870.4)
    last vote rcvd 9 secs ago (at Thu Jun 20 00:34:42 2019),
    last beacon sent 8 secs ago (at Thu Jun 20 00:34:43 2019), last vote was yes
    dbcurrent=1, up=1 beaconSince=1

Server (172.17.17.114): (db 1560360870.4)
    last vote rcvd 10 secs ago (at Thu Jun 20 00:34:41 2019),
    last beacon sent 8 secs ago (at Thu Jun 20 00:34:43 2019), last vote was yes
    dbcurrent=1, up=1 beaconSince=1
mvitale@mvs1:~$
```

Best practices

- Avoid connecting voting servers over (slow, relatively unreliable) WAN links.
  - Consider non-voting clones
- Use an odd integer for quorum set size.
  - Use non-voting clones if you need an even number
- Make backup copies of your databases.
  - `bos stop` does _not_ shutdown ubik servers gracefully (no signal handlers)
- Run `prdb_check` and `vldb_check` occasionally.
- The `udebug` utility is valuable for checking configuration and operation.

Further reading

- By Michael Leon Kazar:
  - Quorum Completion
    - CMU ITCID, Pittsburgh, PA, 1988
  - Ubik – A Library for Managing Ubiquitous Data
    - CMU ITCID, Pittsburgh, PA, 1988
  - Ubik: Replicated Servers Made Easy
    - IEEE *Proc. of the Second Workshop on Workstation Operating Systems*, pages 60–67, September 1989
- By Jeff Hutzelman:
  - Ubik threading analysis
    - https://lists.openafs.org/pipermail/openafs-devel/2011-February/018329.html
    - OpenAFS source tree: doc/txt/ubik.txt

This slide intentionally left blank