MARS Light: Replicating Block Devices over Long Distances

LinuxTag 2014 Presentation by Thomas Schöbel-Theuer
Agenda

- Use Cases DRBD/proxy vs MARS Light
- Working Principle
- Behaviour at Network Bottlenecks
- Multinode Metadata Propagation (Lamport Clock)
- Example Scenario with 4 Nodes
- Current Status / Future Plans
Use Cases DRBD vs MARS Light

**DRBD**

- Application area:
  - Distances: **short** ( <50 km )
  - Synchronously
  - Needs **reliable** network
    - “RAID-1 over network”
    - best with crossover cables
  - Short inconsistencies during re-sync
  - Under pressure: long or even permanent inconsistencies possible
  - Low space overhead

**MARS Light**

- Application area:
  - Distances: **any** ( >>50 km )
  - Asynchronously
    - near-synchronous modes in preparation
  - Tolerates **unreliable network**
  - Anytime consistency
    - no re-sync
  - Under pressure: no inconsistency
    - possibly at cost of actuality
  - Needs >= 100GB in `/mars/` for transaction logfiles
    - dedicated spindle(s) recommended
    - RAID with BBU recommended
Use Cases DRBD+proxy vs MARS Light

**DRBD+proxy**  
(proprietary)

**Application area:**
- Distances: any
- Aynchronously
  - **Buffering in RAM**
- Unreliable network leads to **frequent re-syncs**
  - RAM buffer gets lost
  - at cost of actuality
- **Long** inconsistencies during re-sync
- Under pressure: **permanent** inconsistency possible
- High memory overhead
- Difficult scaling to k>2 nodes

**MARS Light**  
(GPL)

**Application area:**
- Distances: any (>>50 km)
- Asynchronously
  - near-synchronous modes in preparation
- Tolerates **unreliable network**
- Anytime consistency
  - no re-sync
- Under pressure: no inconsistency
  - possibly at cost of actuality
- Needs >= 100GB in /mars/ for transaction logfiles
  - dedicated spindle(s) recommended
  - RAID with BBU recommended
- Easy scaling to k>2 nodes
Multiversion Asynchronous Replicated Storage

Datacenter A (primary)

/dev/mars/mydata

/mars.ko

/dev/lvx/mydata

/mars/translogfile

Datacenter B (secondary)

/mars/translogfile

/mars.ko

/dev/lvx/mydata

Similar to MySQL replication
Network Bottlenecks (1) DRBD

- network throughput

- DRBD throughput

- (potential) incident

- automatic disconnect

- automatic re-connect

- wanted application throughput, not possible

- additional throughput needed for re-sync, not possible

- decreasing throughput limit

- mirror inconsistency ...

- Permanently inconsistent!
Network Bottlenecks (2) MARS

Best possible throughput behaviour at information theoretic limit
Network Bottlenecks (3) MARS

Best possible throughput behaviour

network throughput

MARS application throughput

MARS network throughput

corresponding DRBD inconsistency

flaky throughput limit

time

MARS LinuxTag 2014 Presentation by Thomas Schöbel-Theuer
Problems for $\geq 3$ nodes:
- simultaneous updates
- races

Solution: symlink tree + Lamport Clock $\Rightarrow$ next slides
Symlink tree = key->value store

Originator context encoded in key

Host A (primary)

Host B (secondary)

Host C (secondary)

/mars/resource-mydata/size-hostA -> 1000

/mars/resource-mydata/size-hostA -> oldValue

Anyone knows anything about others

But later
Lamport Clock = virtual timestamp

Propagation goes never backwards!

Races are compensated

Propagation paths play no role
Productive Scenario since 03/2014 (1&1 eShop / ePages)

MARS LinuxTag 2014 Presentation by Thomas Schöbel-Theuer

AppCluster A1 (primary) → AppCluster B1 (secondary) via georedundancy (BGP)

AppCluster A2 (secondary) ← room-to-room

AppCluster B2 (secondary) ← room-to-room

potential data flow

actual data flow (in this scenario)
Current Status / Future Plans

- Source / docs at
  github.com/schoebel/mars

- Productive on customer data since 03/2014

- Database support / near-synchronous modes planned for end of 2014

- Further challenges:
  - community revision at LKML planned
  - split into 3 parts:
    - Generic brick framework
    - XIO/AIO personality (1st citizen)
    - MARS Light (1st application)
  - hopefully attractive for other developers!
### DRBD+proxy Architectural Challenge

- **DRBD Host A** (primary)
  - bitmap A
  - huge RAM buffer

- **Proxy A’**
  - A != A’ possible
  - data queue path (several GB buffered)
  - completion path (commit messages)

- **Proxy B’** (essentially unused)
- **DRBD Host B** (secondary)
  - bitmap B

- **Sector #8**: same sector #8 occurs n times in queue

- **n times**
  - => need \( \log(n) \) bits for counter
  - => but DRBD bitmap has only 1 bit/sector
  - => workarounds exist, but complicated
    (e.g. additional dynamic memory)
MARS Light Data Flow Principle

Host A (primary)

Transaction Logger

Temporary Memory Buffer

/dev/mars/mydata

/writeback in background

append

/dev/lv-x/mydata

/mars/resource-mydata/log-00001-hostA

Host B (secondary)

Logfile Replicator

Logfile Applicator

long-distance transfer

/mars/resource-mydata/log-00001-hostA

/dev/lv-x/mydata

MARS LinuxTag 2014 Presentation by Thomas Schöbel-Theuer
Framework Architecture for MARS + future projects

Framework Application Layer
- MARS Light, MARS Full, etc

Framework Personalities
- XIO = eXtended IO ≈ AIO

Generic Brick Layer
- IOP = Instance Oriented Programming
- AOP = Aspect Oriented Programming

External Software, Cluster Managers, etc

Userspace Interface marsadm

MARS Light

MARS Full

... future

XIO bricks

Strategy bricks

other future Personalities and their bricks

Generic Bricks

Generic Objects

Generic Aspects
Aspects are automatically attached on the fly
Global external bandwidth > 285 GBit/s
Peering with biggest internet exchanges on the world
Own metro networks (DWDM) at the 1&1 datacenter locations
IO Latencies over loaded Metro Network (1) DRBD

Load = ~30,000 IOPS on 50 spindles RAID-6 (7x shared-derived from blkreplay.org)

red = write latency
blue = read latency

Load = ~30,000 IOPS on 50 spindles RAID-6 (7x shared-derived from blkreplay.org)
IO Latencies over loaded Metro Network (2) MARS

red = write latency
blue = read latency

Same load as before, same conditions