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
(GPL)

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
(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

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

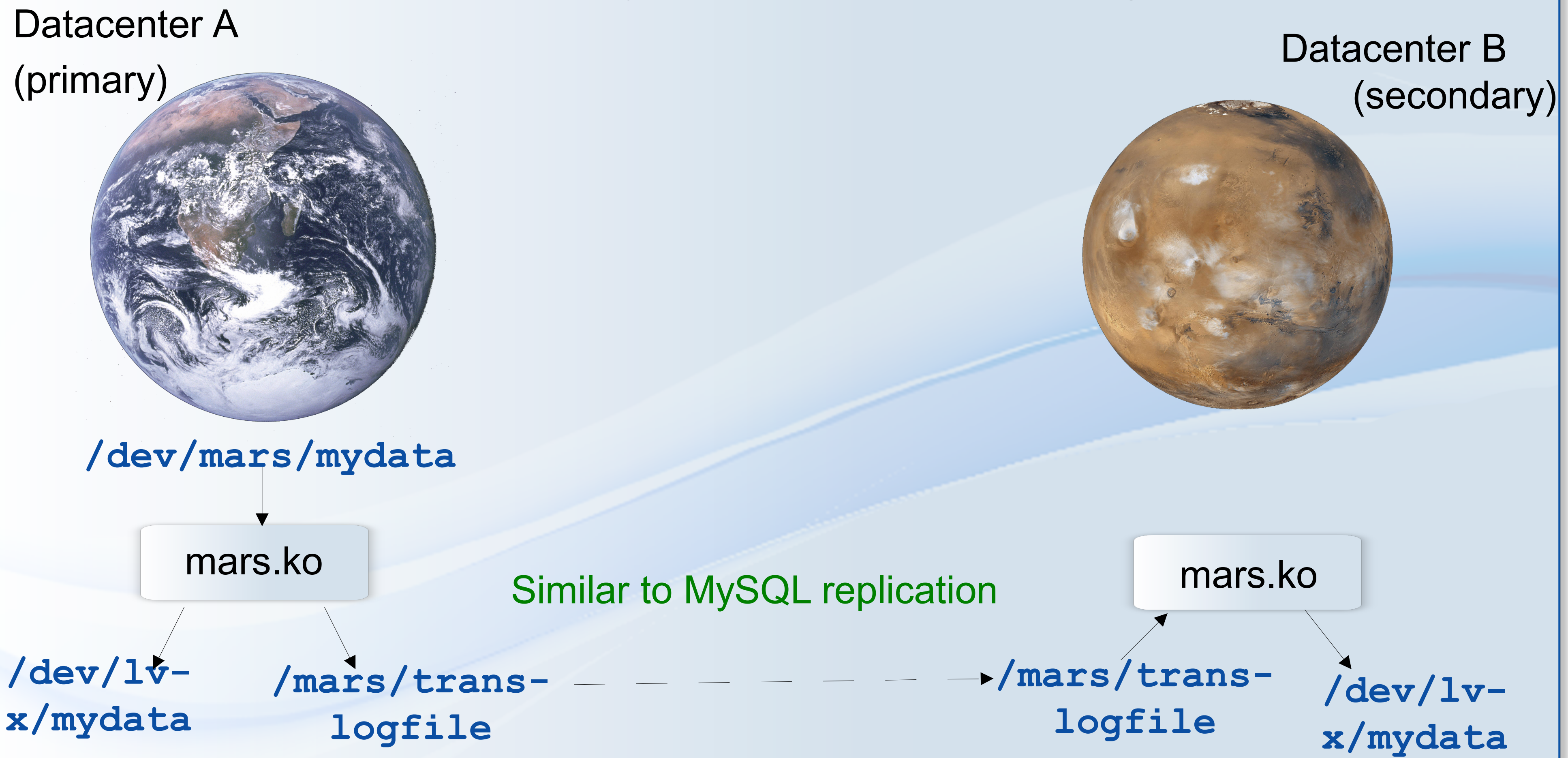
(GPL)

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

MARS Working Principle

Multiversion Asynchronous Replicated Storage



Datacenter A (primary)

/dev/mars/mydata

mars.ko

/dev/lv-x/mydata

/mars/trans-logfile

Similar to MySQL replication

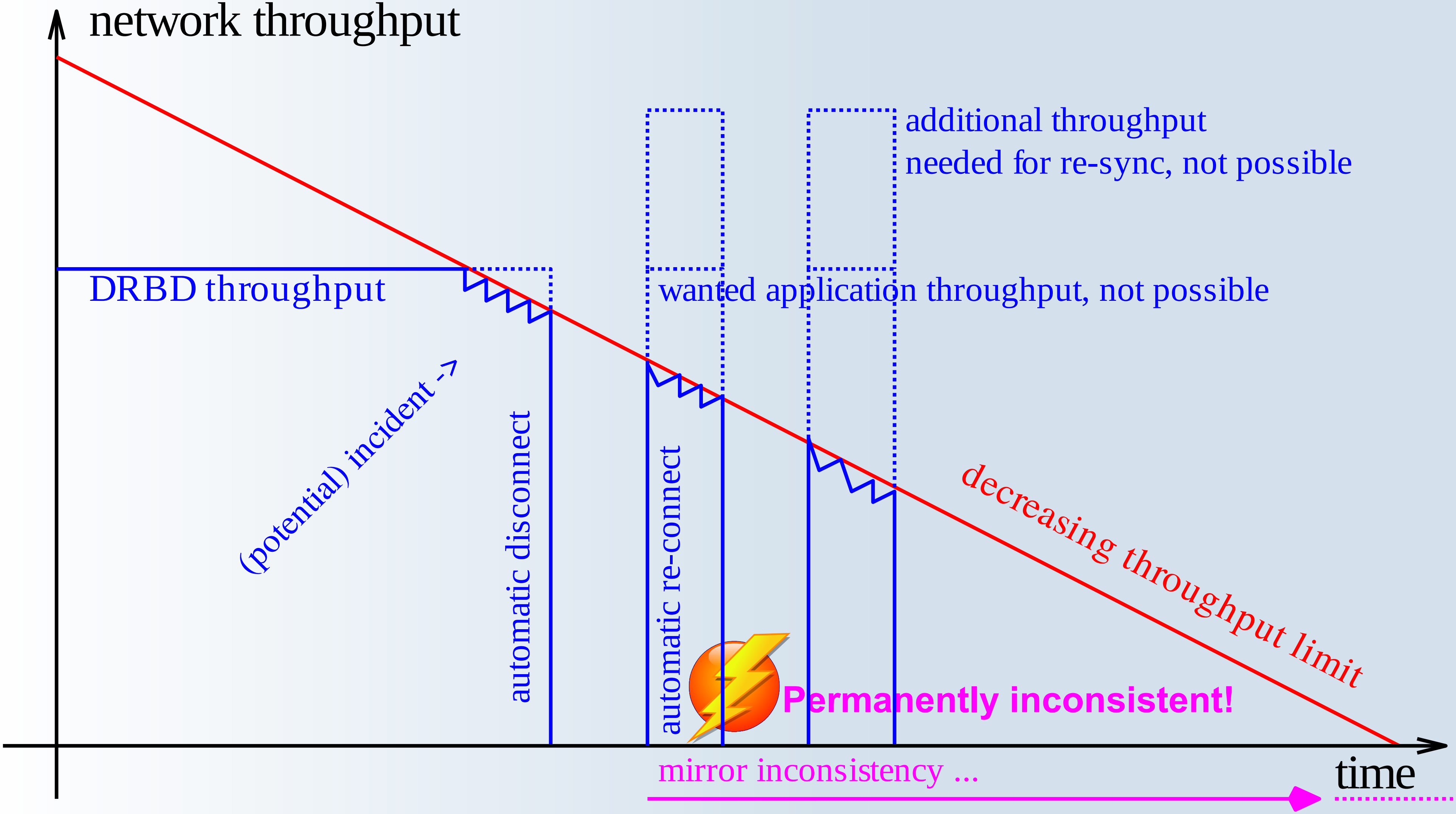
Datacenter B (secondary)

mars.ko

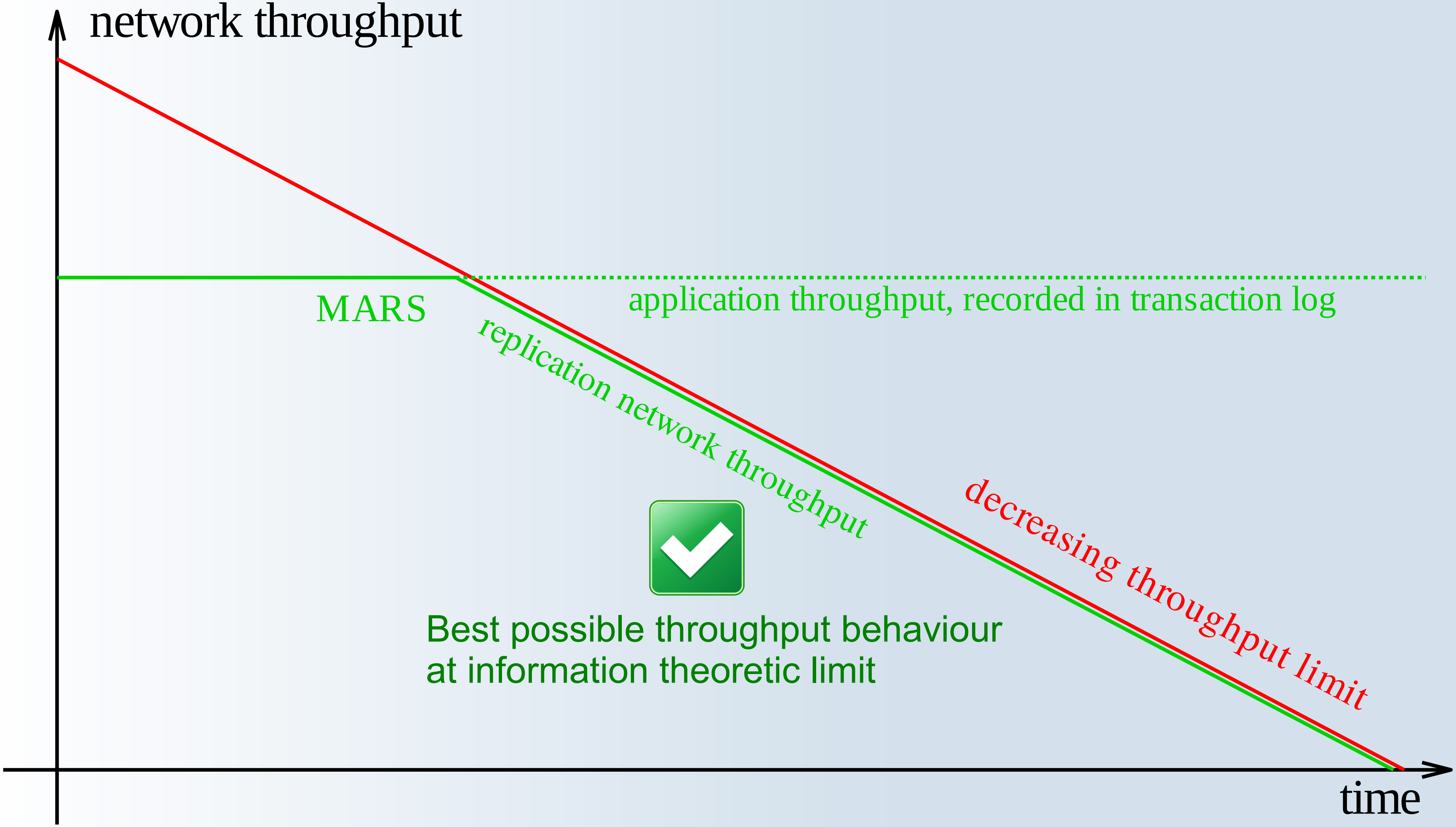
/mars/trans-logfile

/dev/lv-x/mydata

Network Bottlenecks (1) DRBD

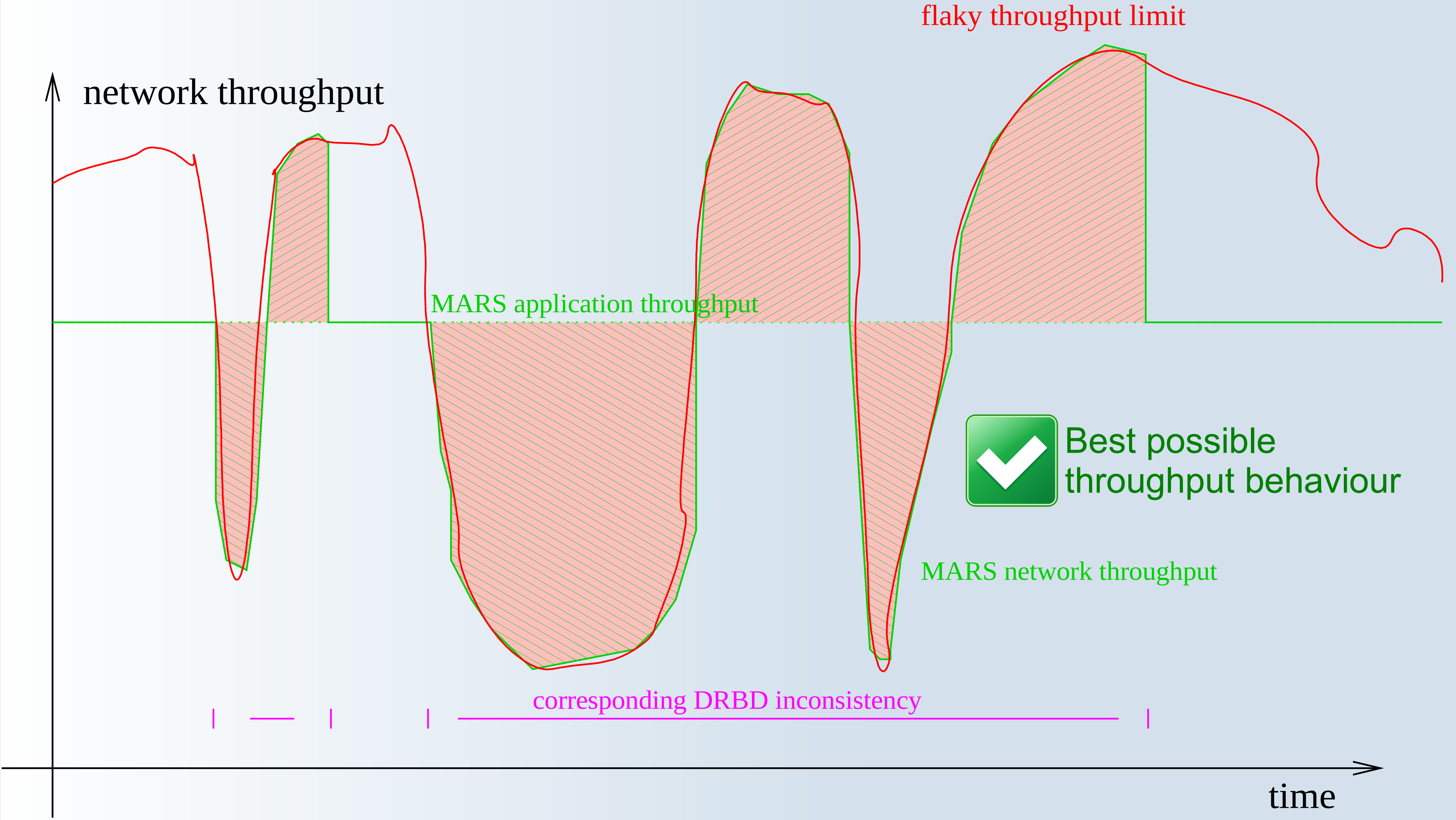


Network Bottlenecks (2) MARS



Best possible throughput behaviour
at information theoretic limit

Network Bottlenecks (3) MARS

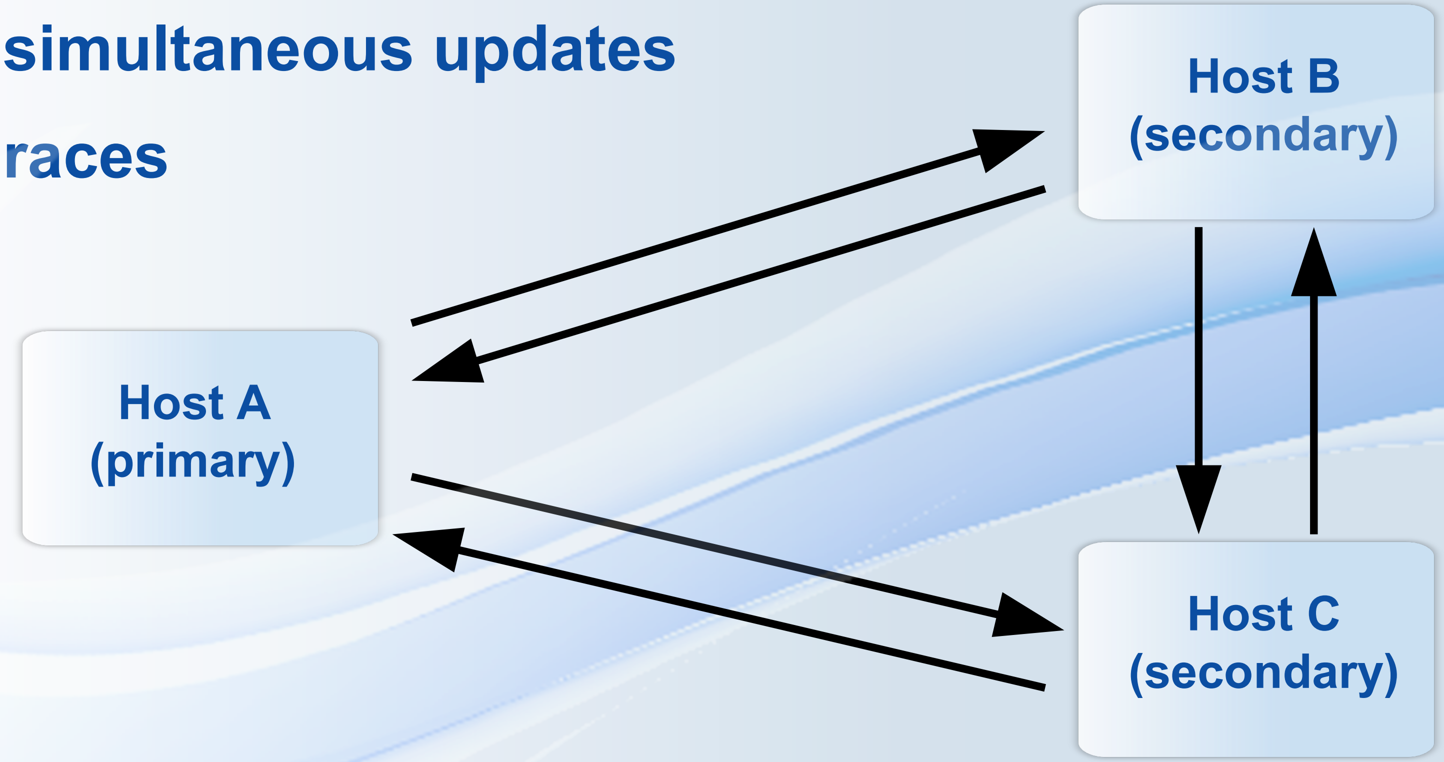


✓ Best possible throughput behaviour

Metadata Propagation (1)

Problems for ≥ 3 nodes:

- simultaneous updates
- races



Solution: symlink tree + Lamport Clock => next slides

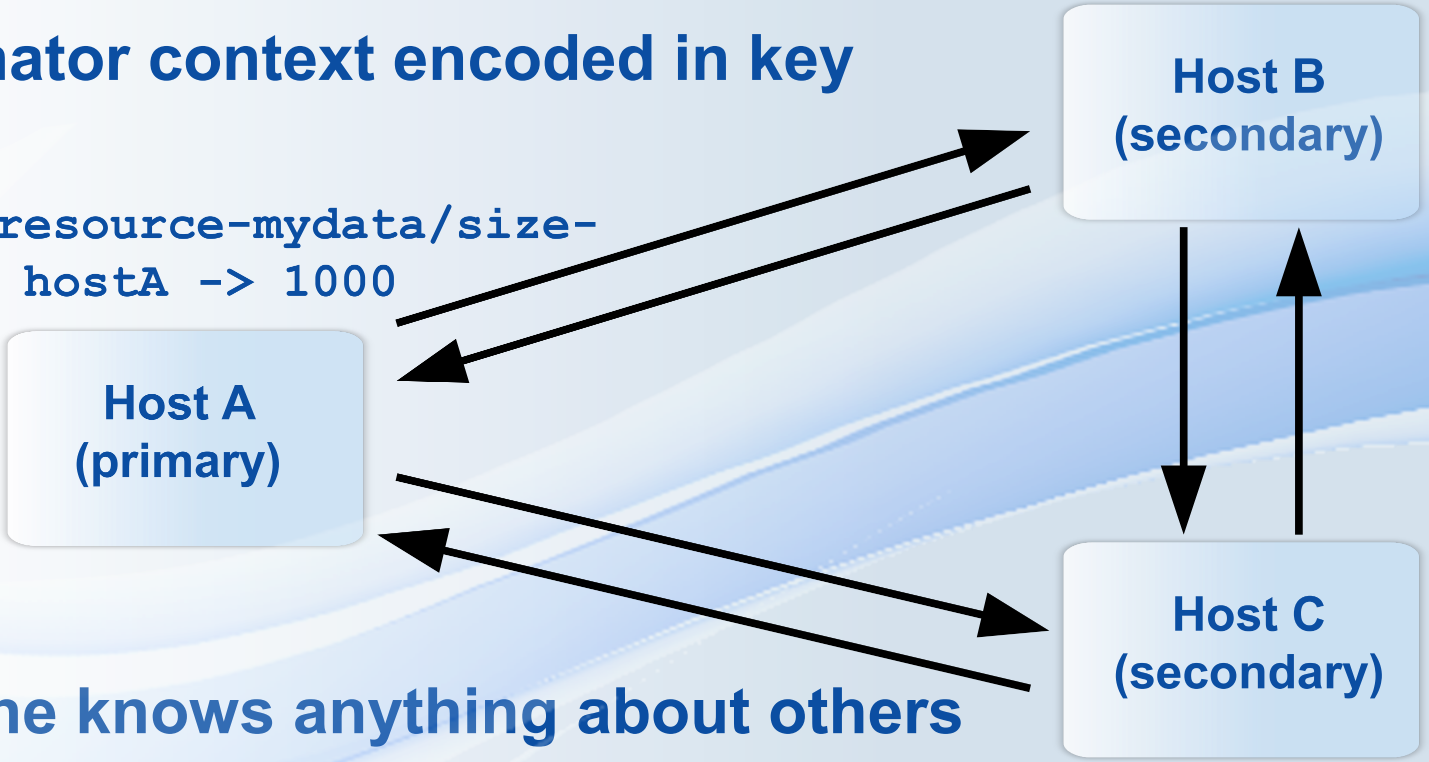
Metadata Propagation (2)

Symlink tree = key->value store

Originator context encoded in key

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

`/mars/resource-mydata/size-hostA ->` *`oldvalue`*



`/mars/resource-mydata/size-hostA ->` *`oldvalue`*

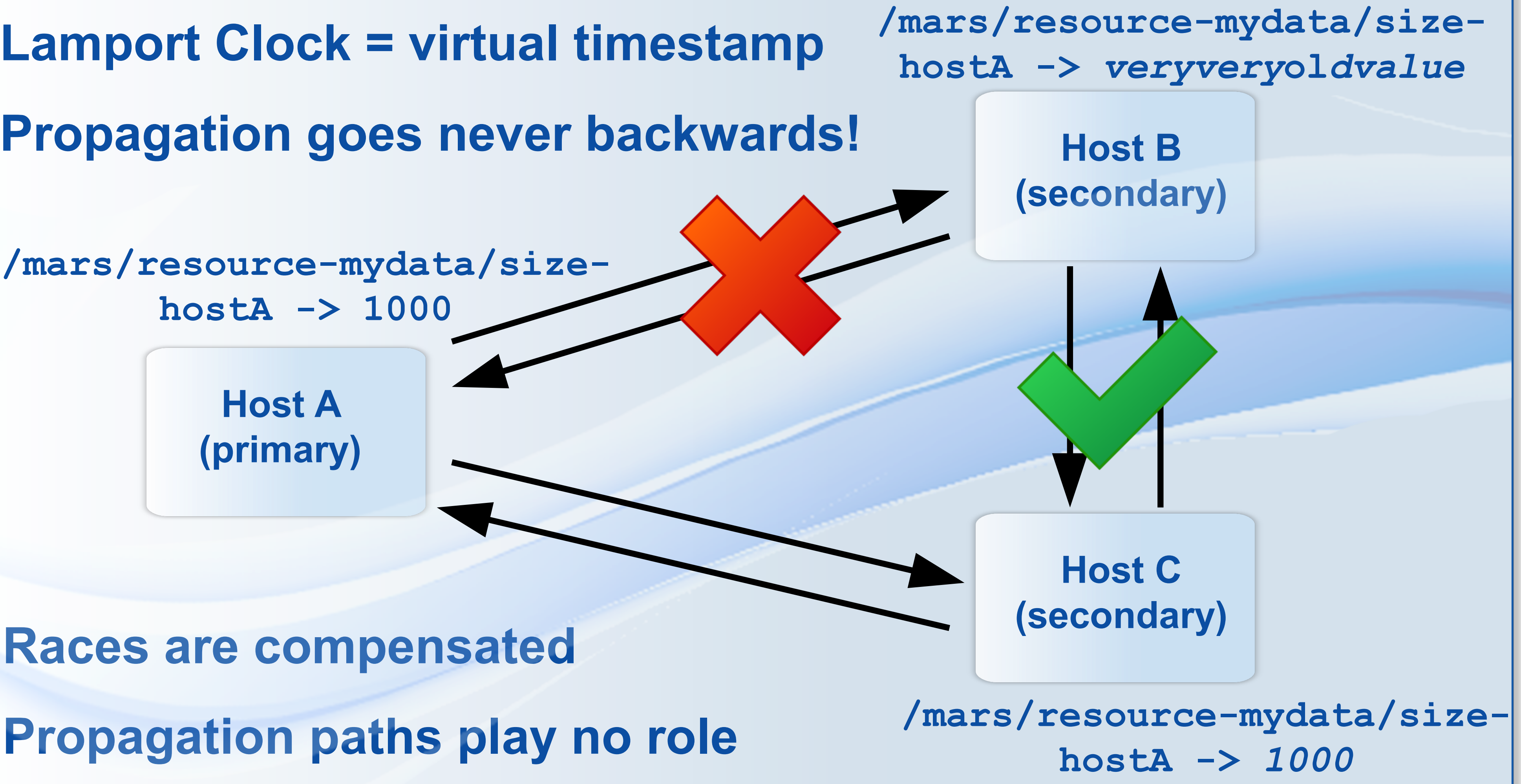
Anyone knows anything about others

But later

Metadata Propagation (3)

Lamport Clock = virtual timestamp

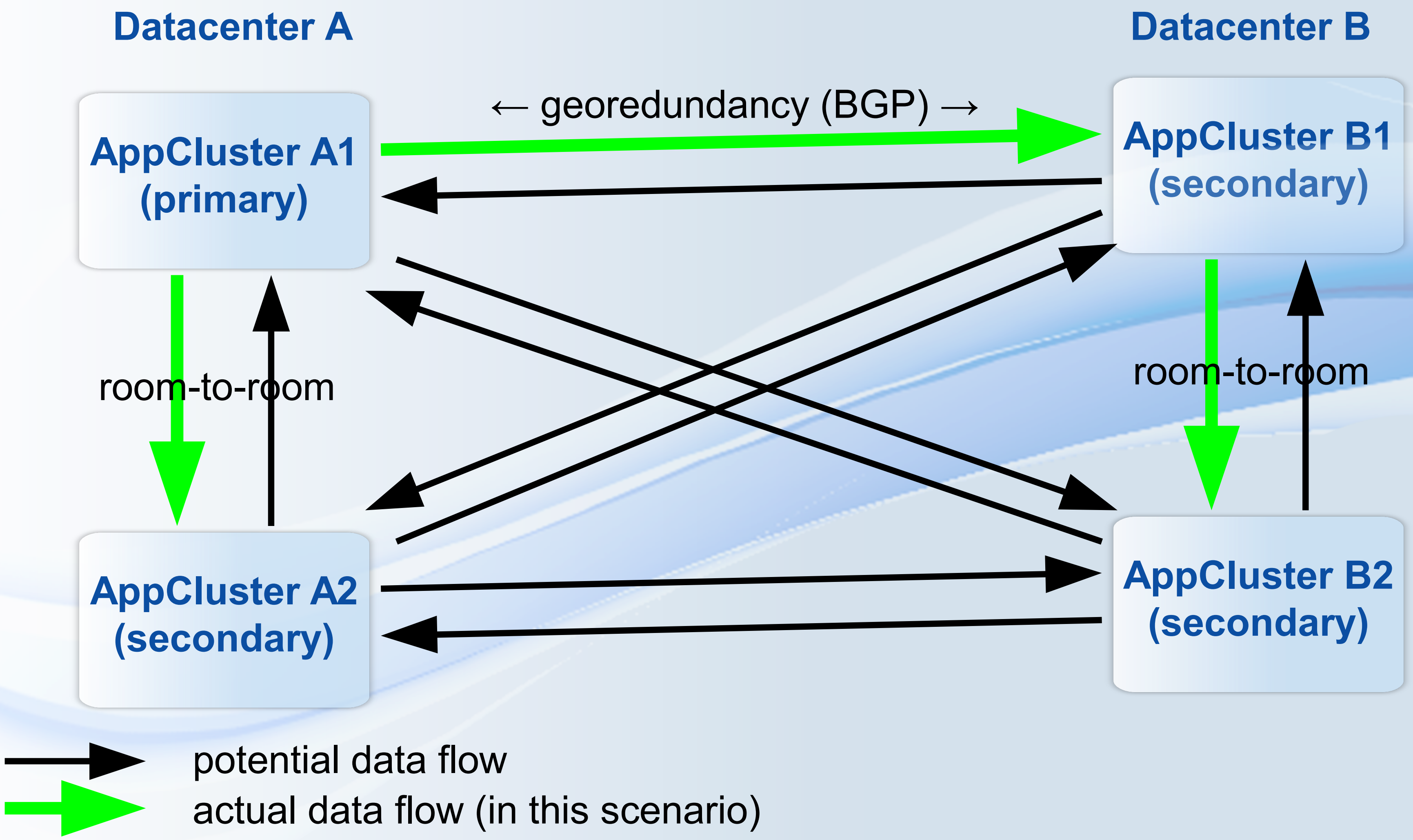
Propagation goes never backwards!


/mars/resource-mydata/size-
hostA -> *veryveryoldvalue*
Host B
(secondary)
/mars/resource-mydata/size-
hostA -> 1000
Host A
(primary)
Host C
(secondary)
/mars/resource-mydata/size-
hostA -> *1000*

Races are compensated

Propagation paths play no role

Productive Scenario since 03/2014 (1&1 eShop / ePages)



Datacenter A

Datacenter B

AppCluster A1 (primary)

← georedundancy (BGP) →

AppCluster B1 (secondary)

room-to-room

room-to-room

AppCluster A2 (secondary)

AppCluster B2 (secondary)

potential data flow

actual data flow (in this scenario)

Current Status / Future Plans

- Source / docs at

  `github.com/schoebel/mars`

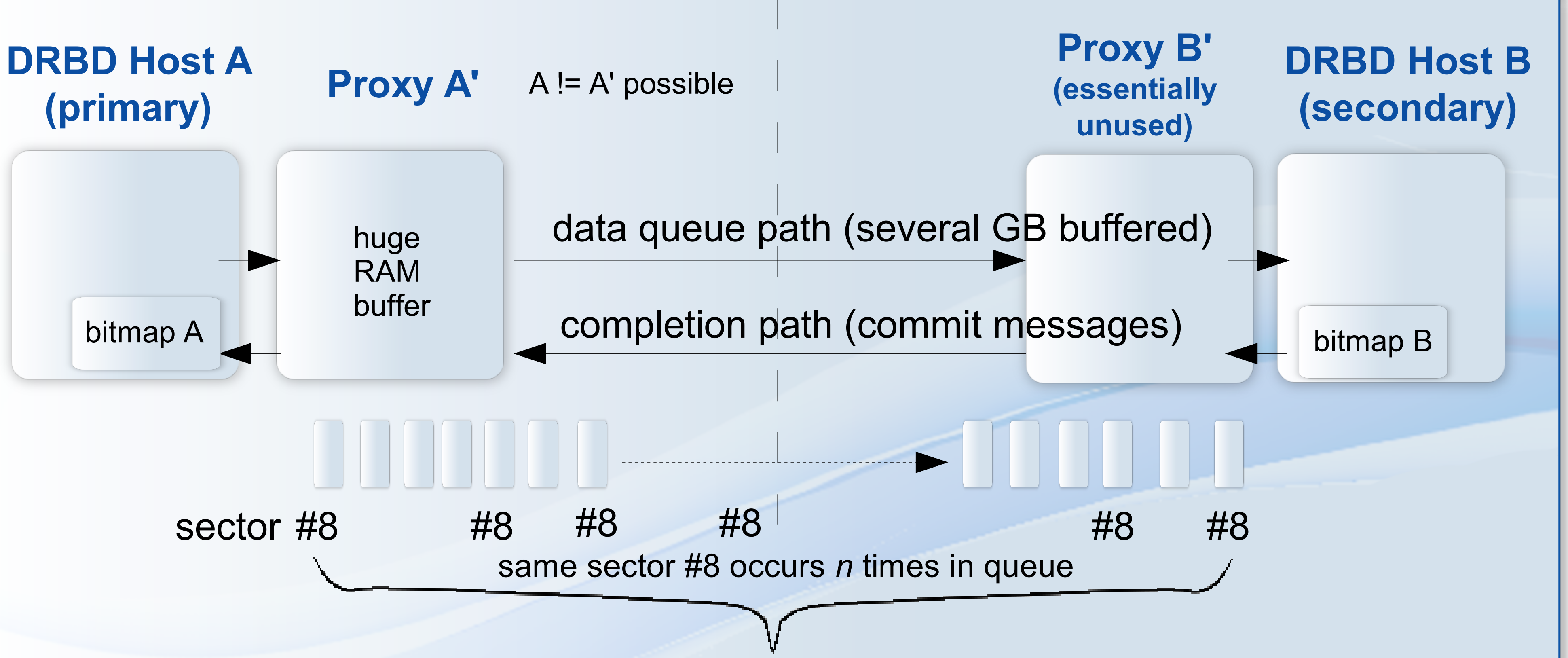
- Productive on customer data since 03/2014

- 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!

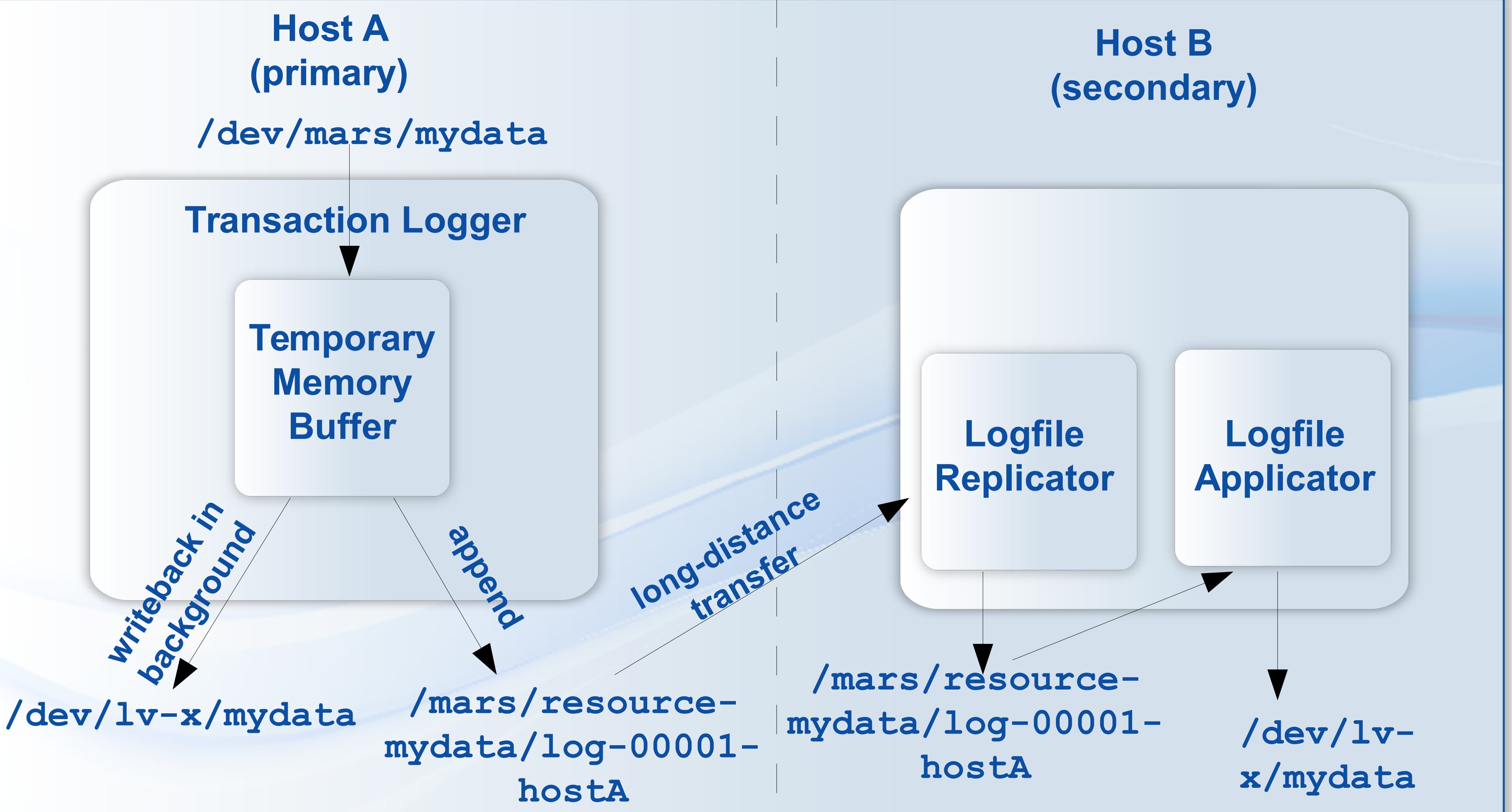
Appendix

DRBD+proxy Architectural Challenge



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



Framework Architecture

for MARS + future projects

Framework Application Layer

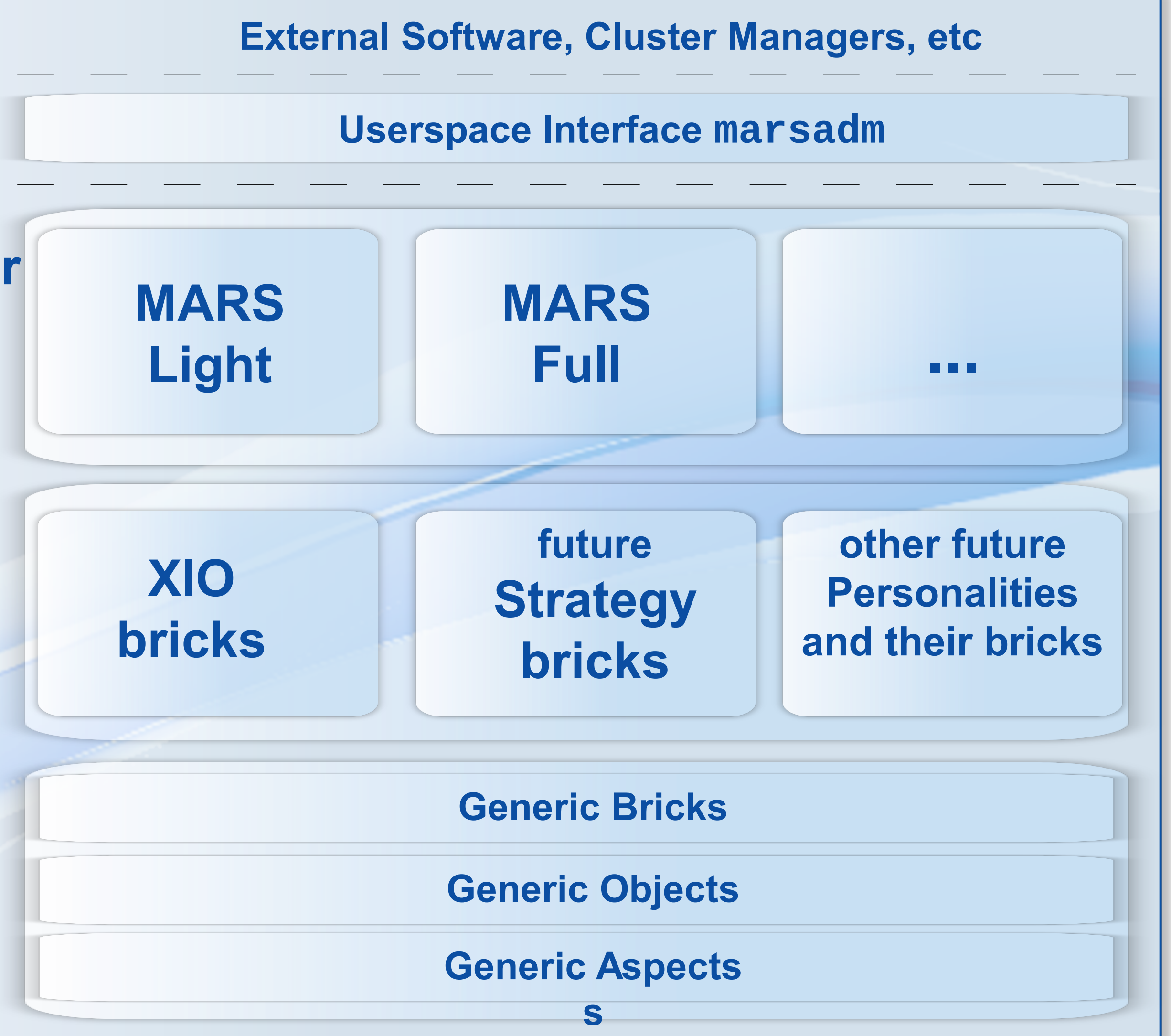
MARS Light, MARS Full, etc

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

...

XIO bricks

future Strategy bricks

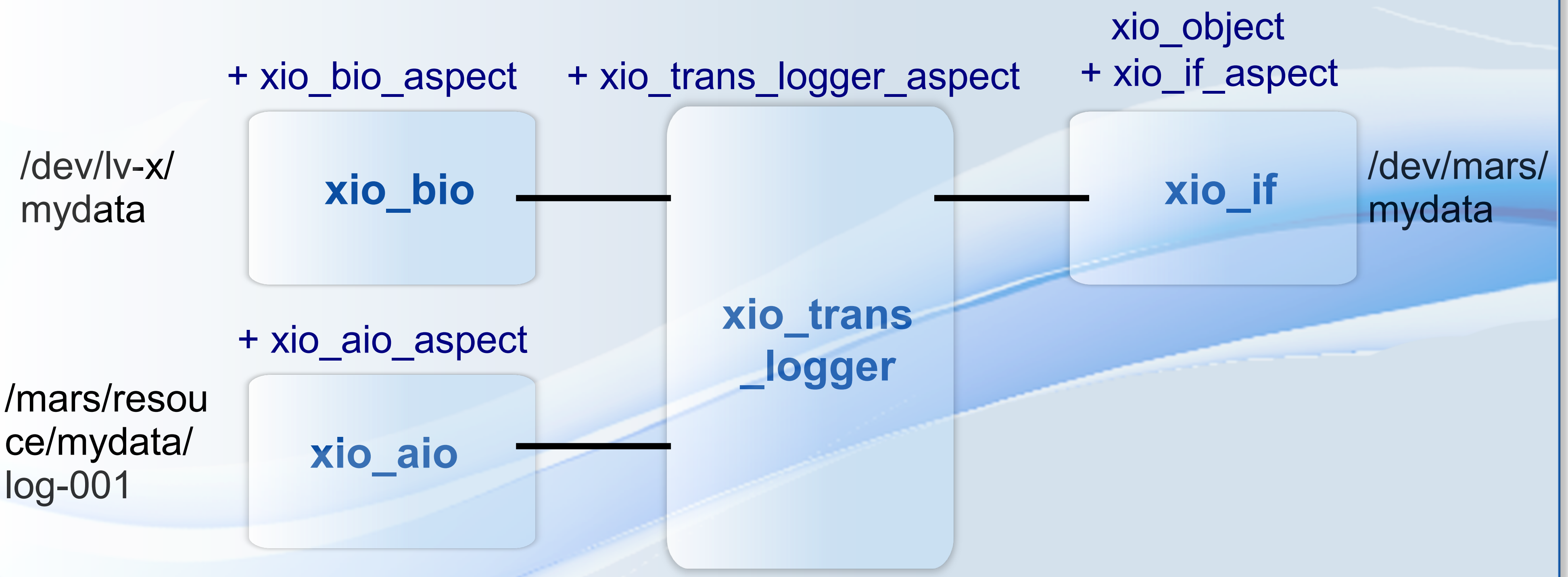
other future Personalities and their bricks

Generic Bricks

Generic Objects

Generic Aspects
s

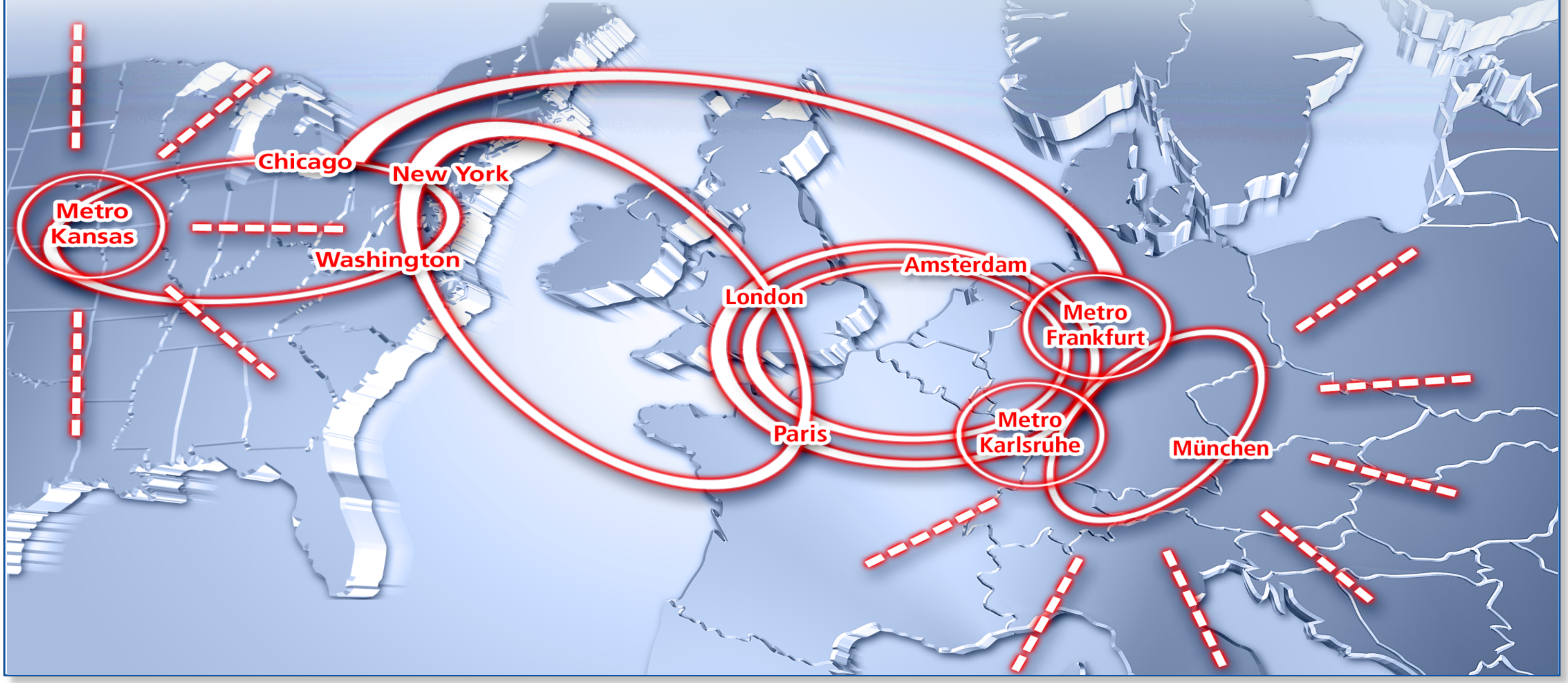
Bricks, Objects + Aspects (Example)



Aspects are automatically attached on the fly

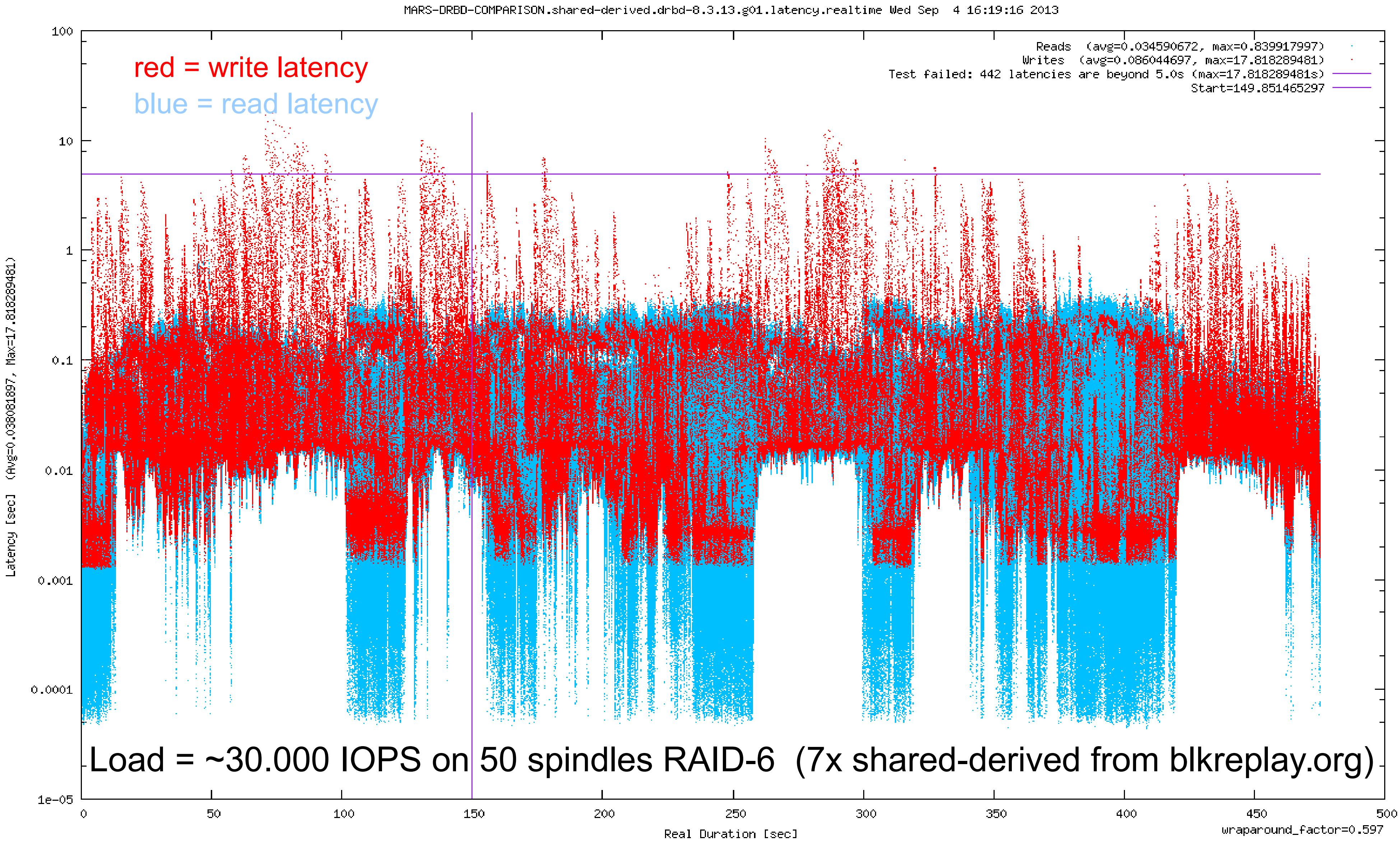
Appendix: 1&1 Wide Area Network Infrastructure

- Global external bandwidth > 285 GBit/s
- Peering with biggest internet exchanges on the world
- Own metro networks (DWDM) at the 1&1 datacenter locations



® 1&1 Internet AG 2012

IO Latencies over loaded Metro Network (1) DRBD

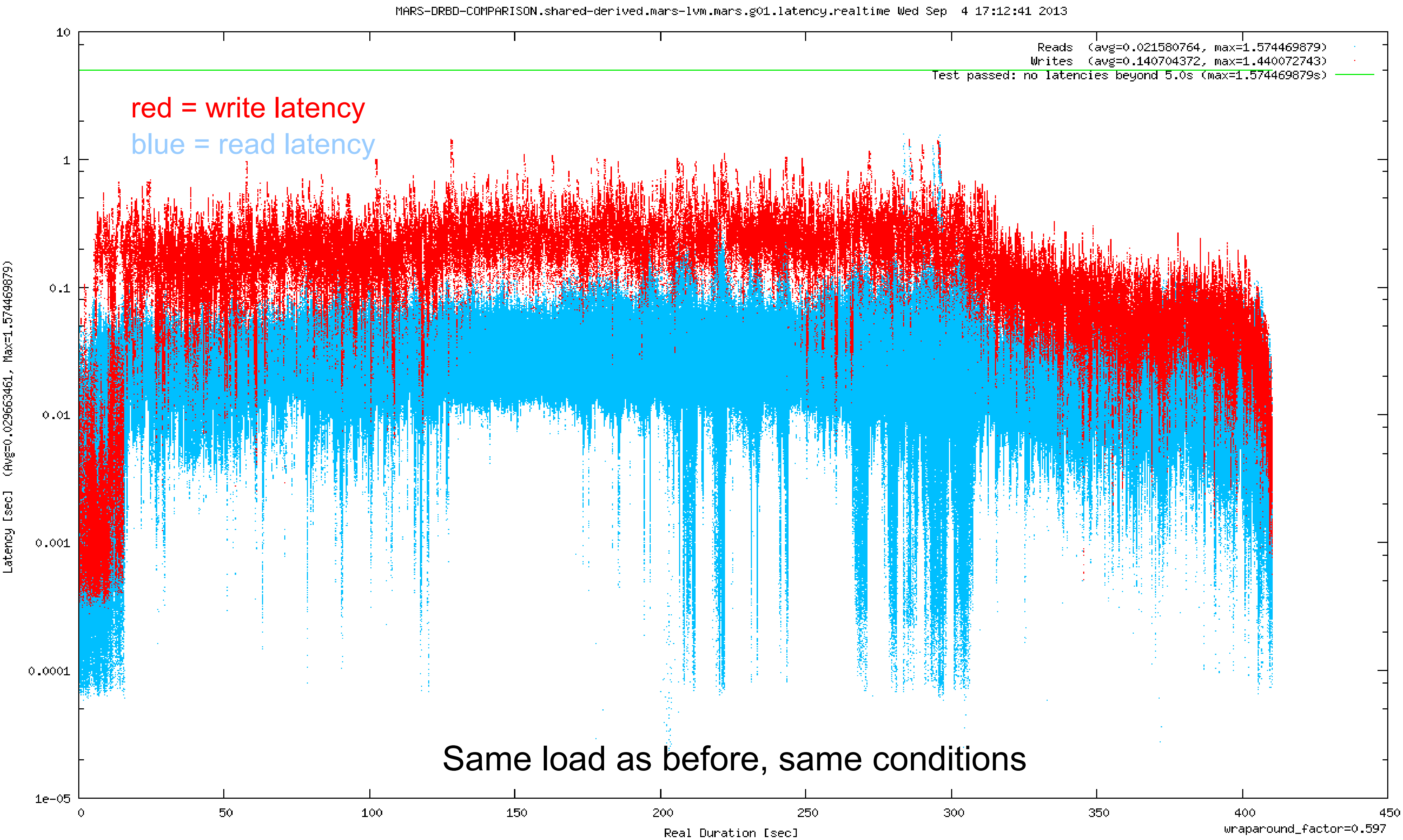


red = write latency

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