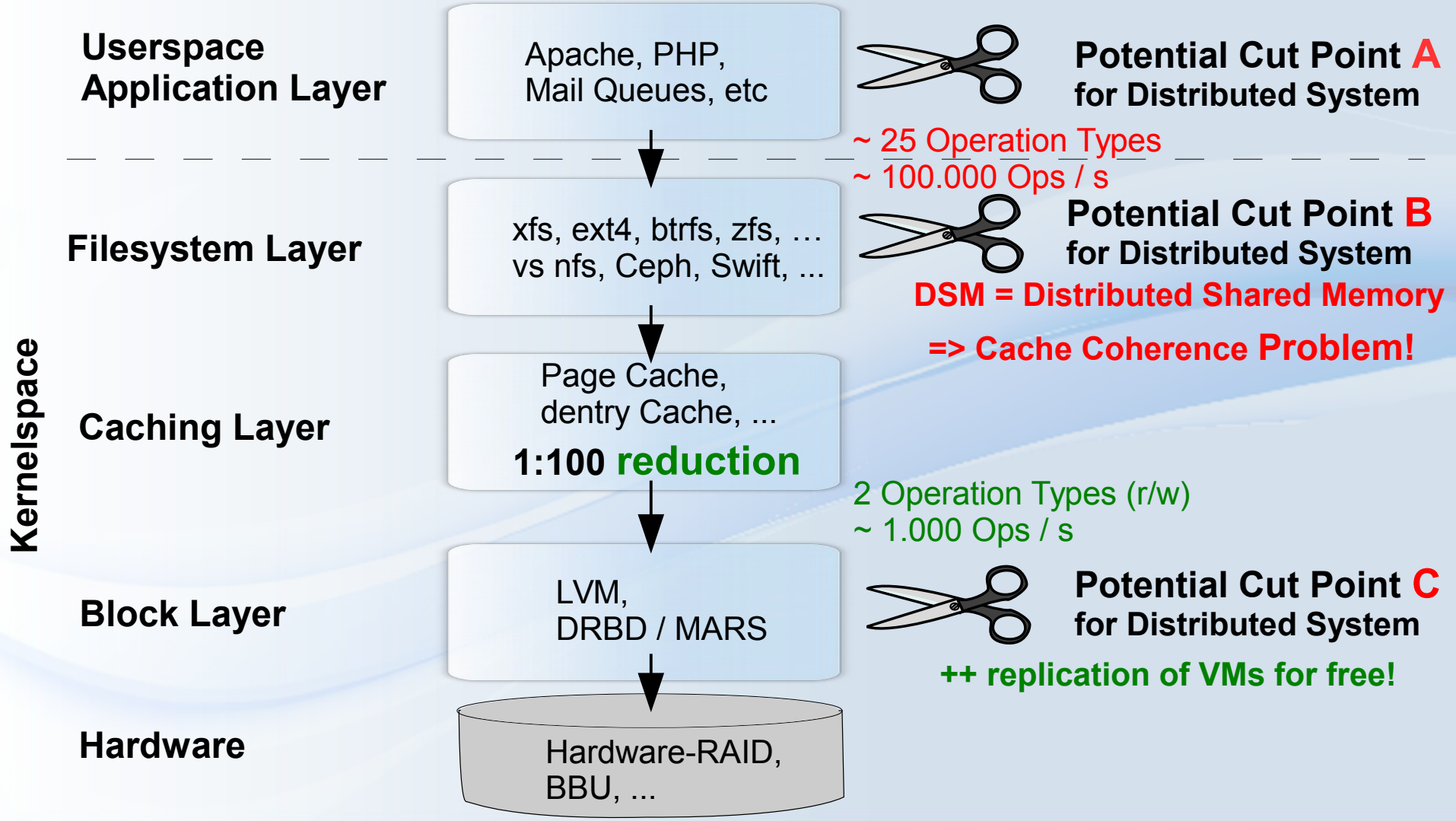
MARS: Replicating Petabytes
over Long Distances

GUUG 2016 Presentation by Thomas Schöbel-Theuer

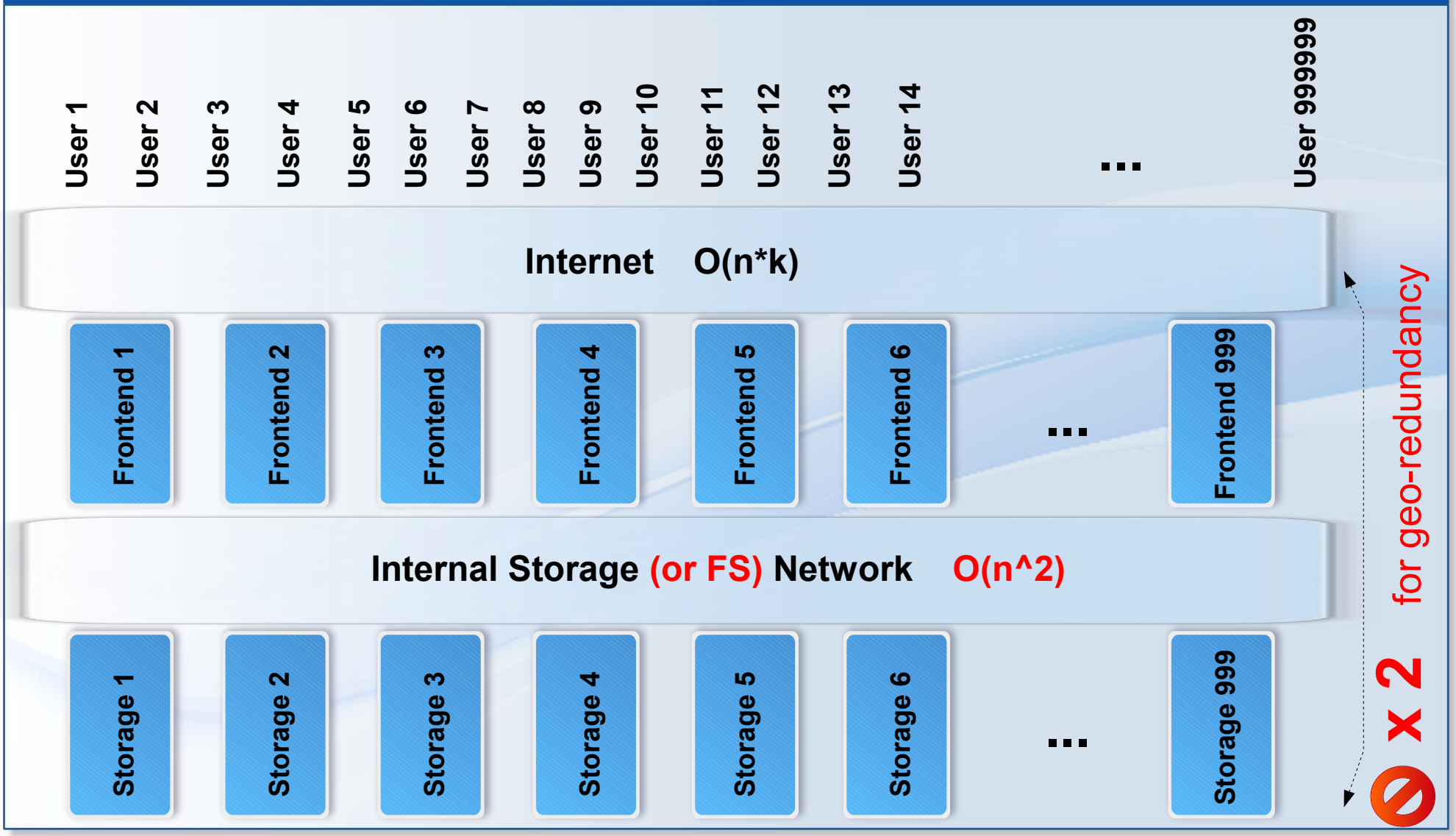
Replicating Petabytes: Agenda

- Long Distances: Block Level vs FS Level
- Long Distances: Big Cluster vs Sharding
- Use Cases DRBD vs MARS Light
- MARS Working Principle
- Behaviour at Network Bottlenecks
- Multinode Metadata Propagation (Lamport Clock)
- Example Scenario with 4 Nodes
- Current Status / Future Plans

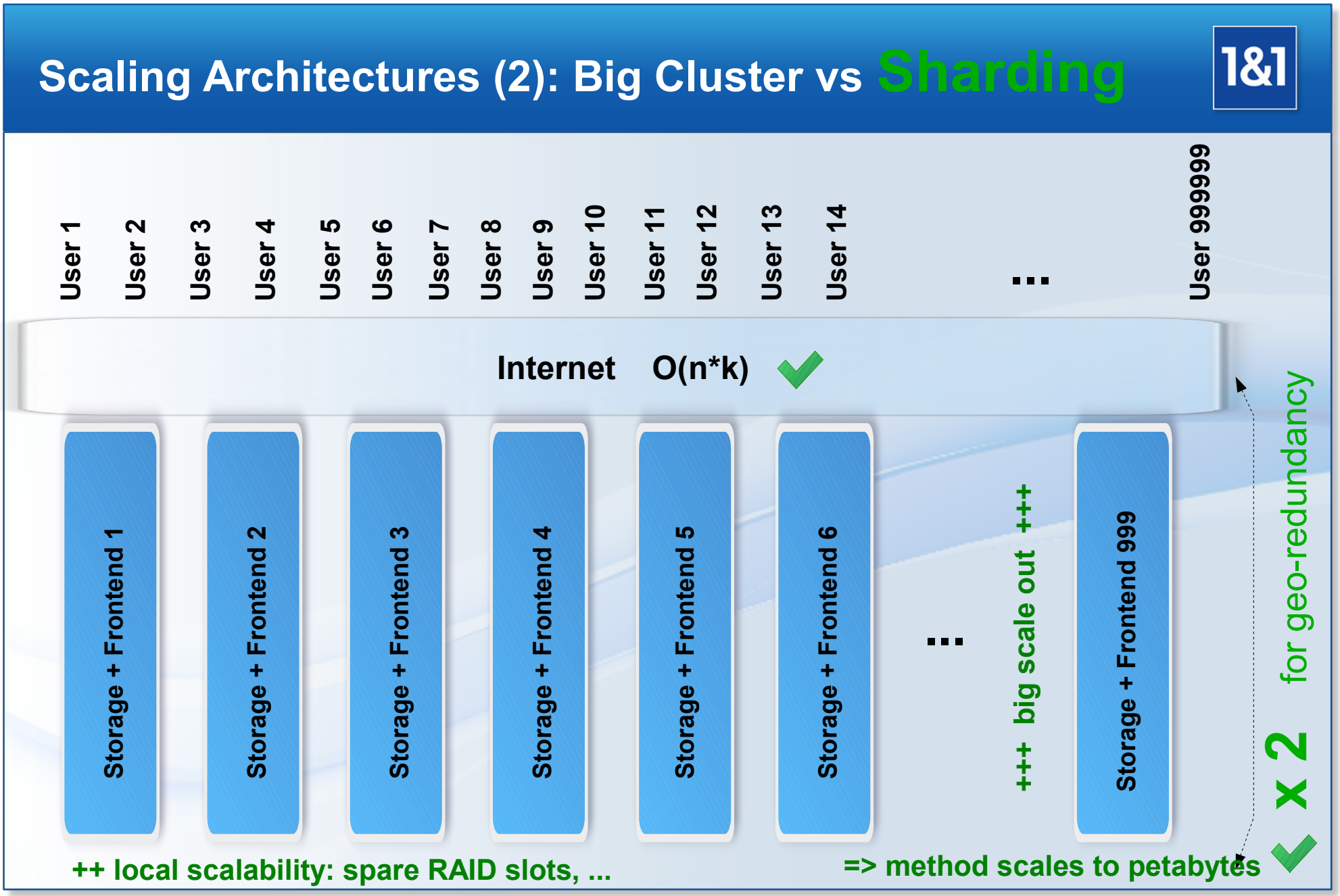
Replication at Block Level vs FS Level



Scaling Architectures (1): Big Cluster vs Sharding



Scaling Architectures (2): Big Cluster vs Sharding



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:

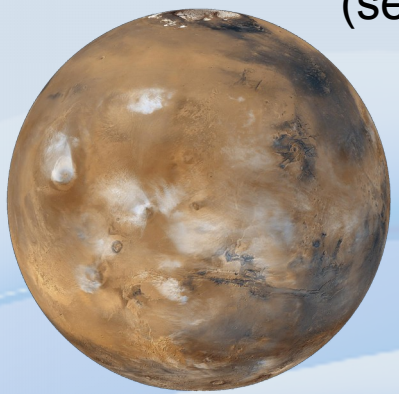
- Distances: **any** (>>50 km)
- Asynchronously
  - near-synchronous modes in preparation
- Tolerates **unreliable network**
- Anytime consistency
  - no re-sync
- Under pressure: no inconsistency
  - possibly at cost of actuality
- Needs >= 100GB in `/mars/` for transaction logfiles
  - dedicated spindle(s)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

MARS Working Principle

Multiversion Asynchronous Replicated Storage

Datacenter A
(primary)

Datacenter B
(secondary)



`/dev/mars/mydata`

mars.ko

Similar to MySQL replication

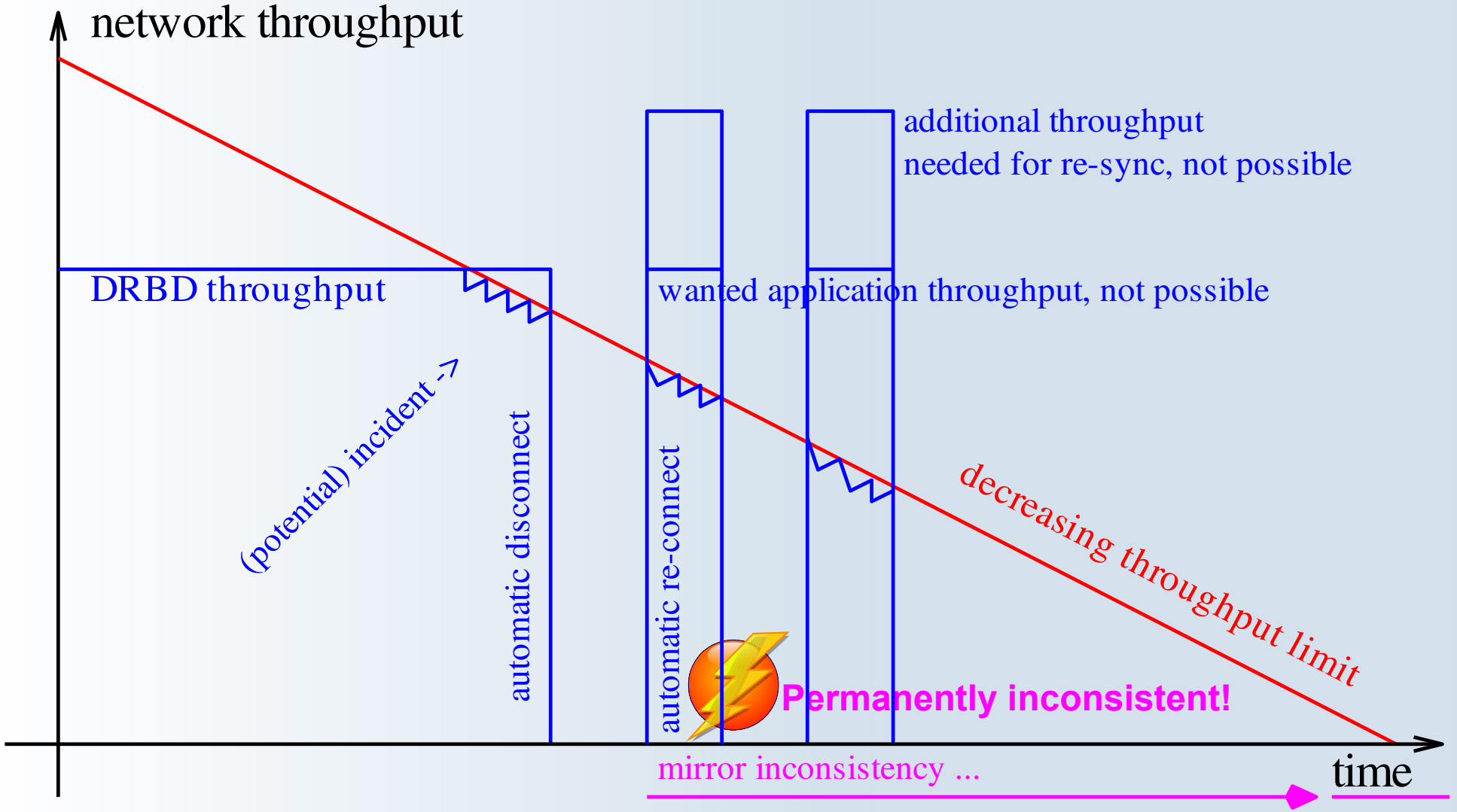
mars.ko

`/dev/lv-x/mydata`

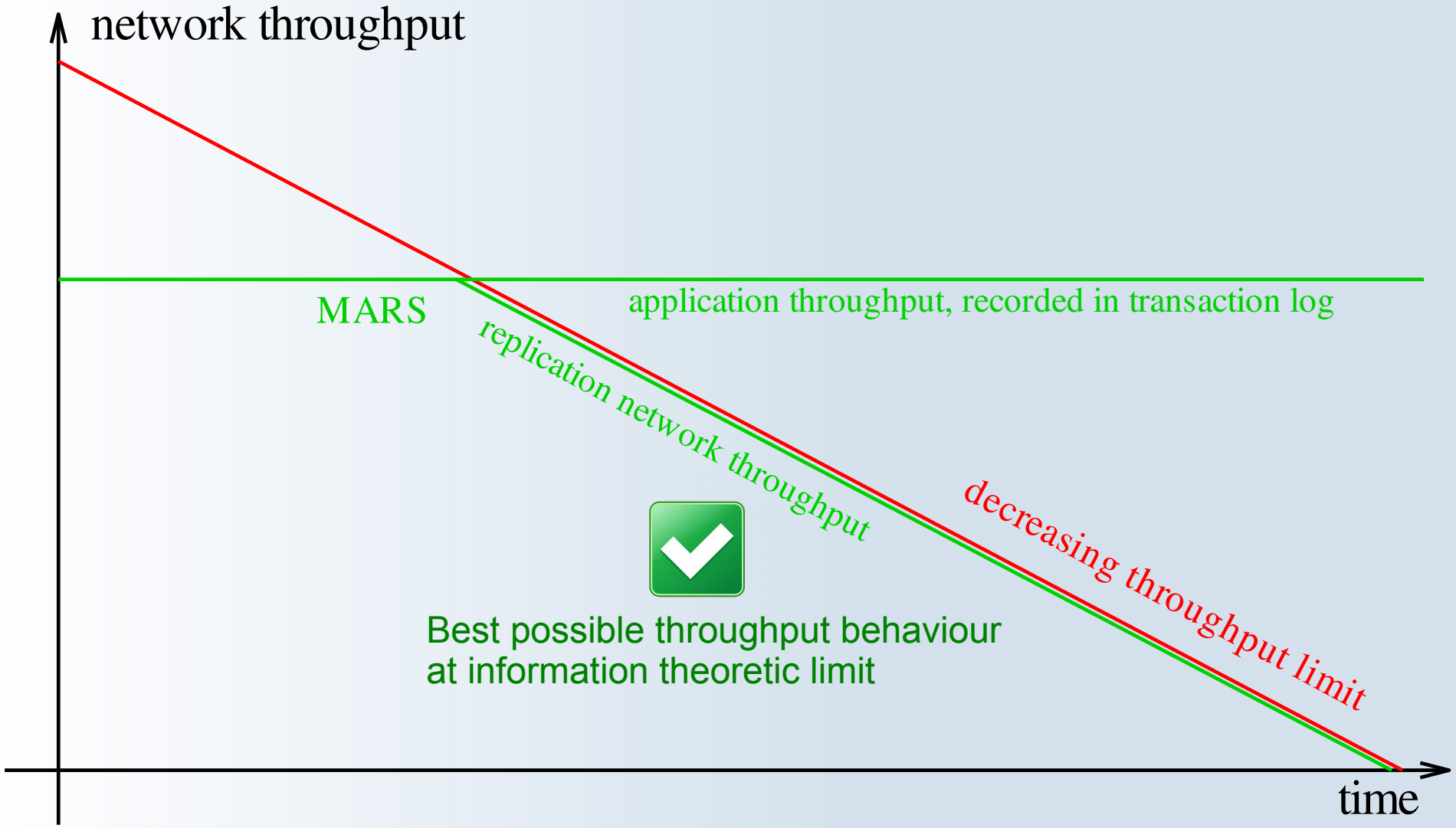
`/mars/trans-logfile` → `/mars/trans-logfile`

`/dev/lv-x/mydata`

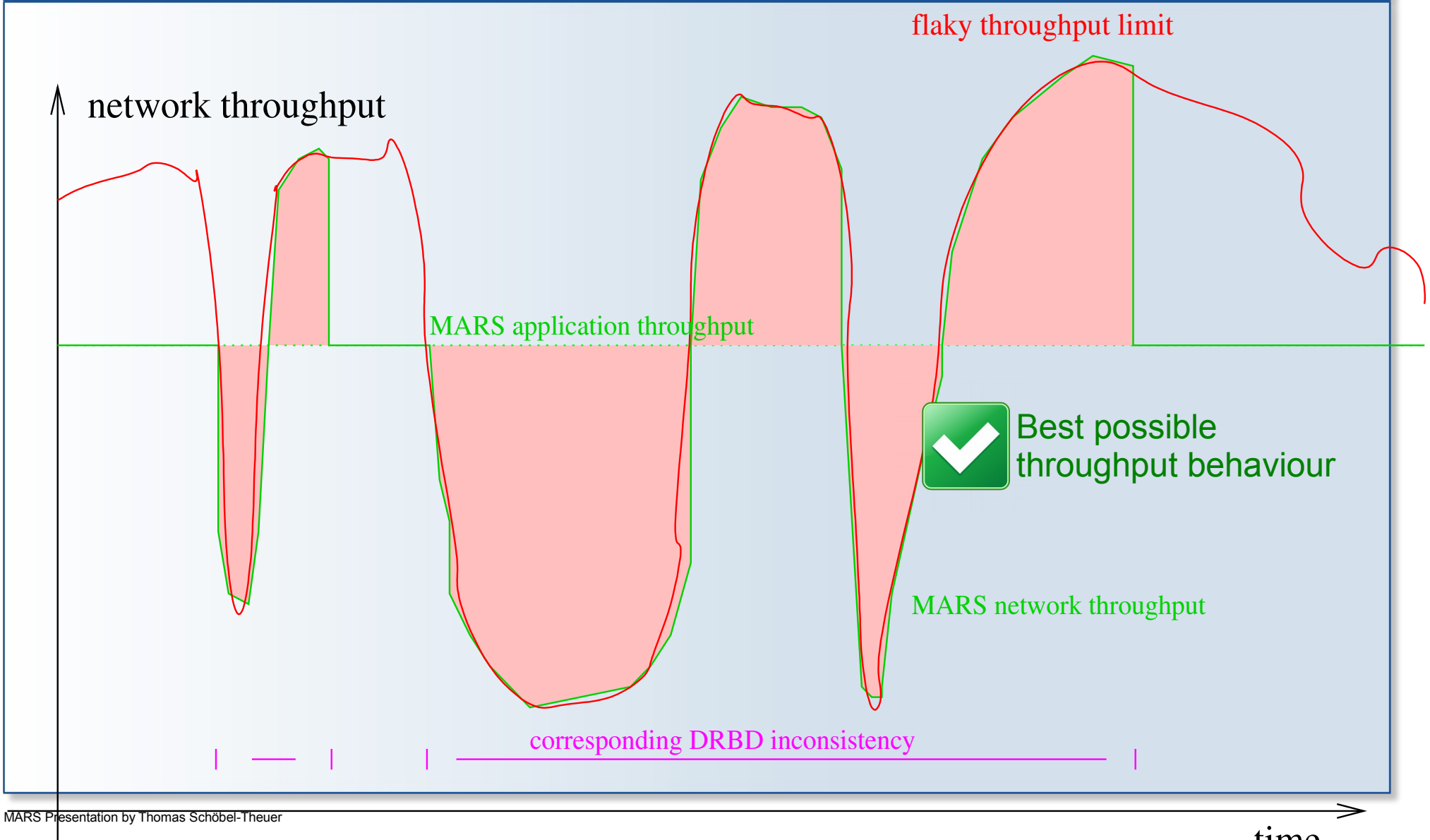
Network Bottlenecks (1) DRBD



Network Bottlenecks (2) MARS



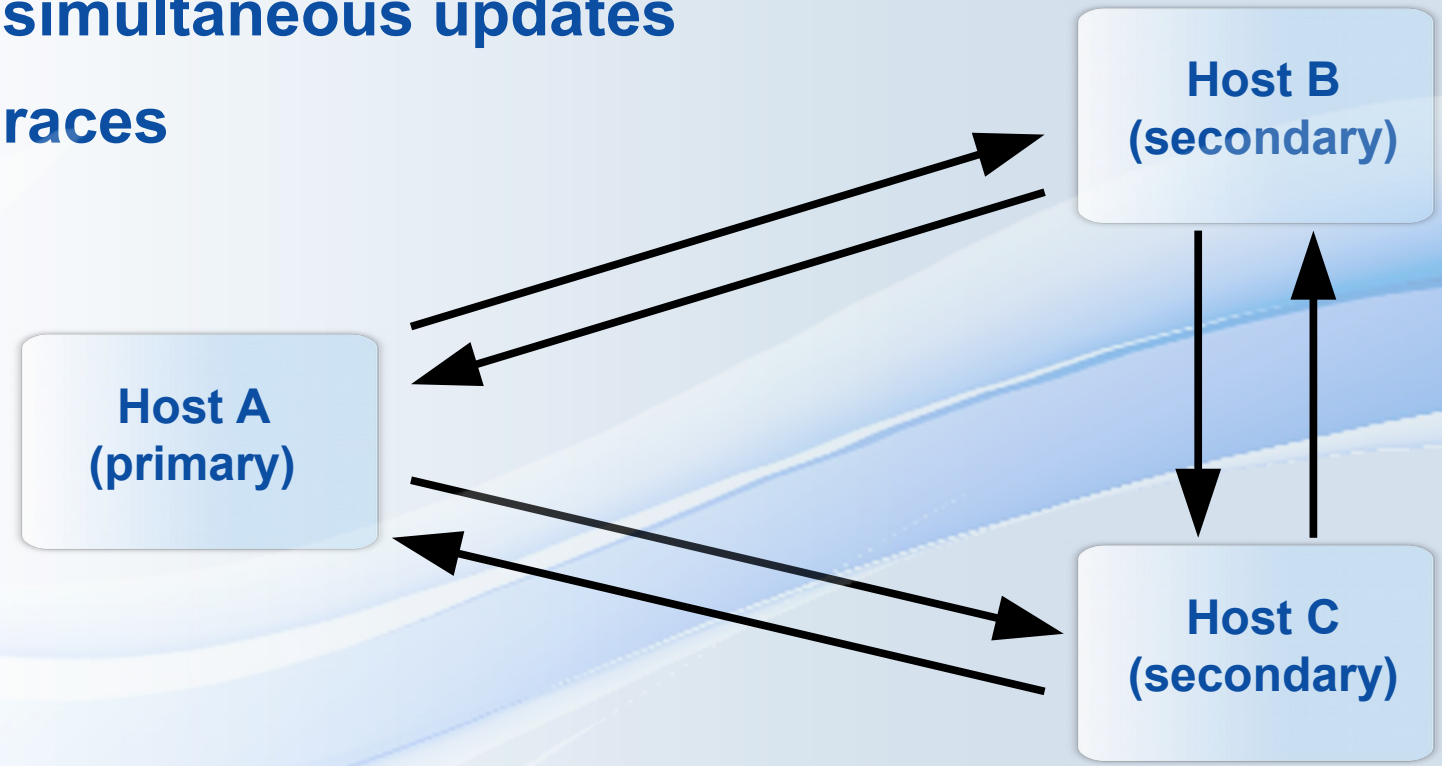
Network Bottlenecks (3) MARS



Metadata Propagation (1)

Problems for ≥ 3 nodes:

- simultaneous updates
- races

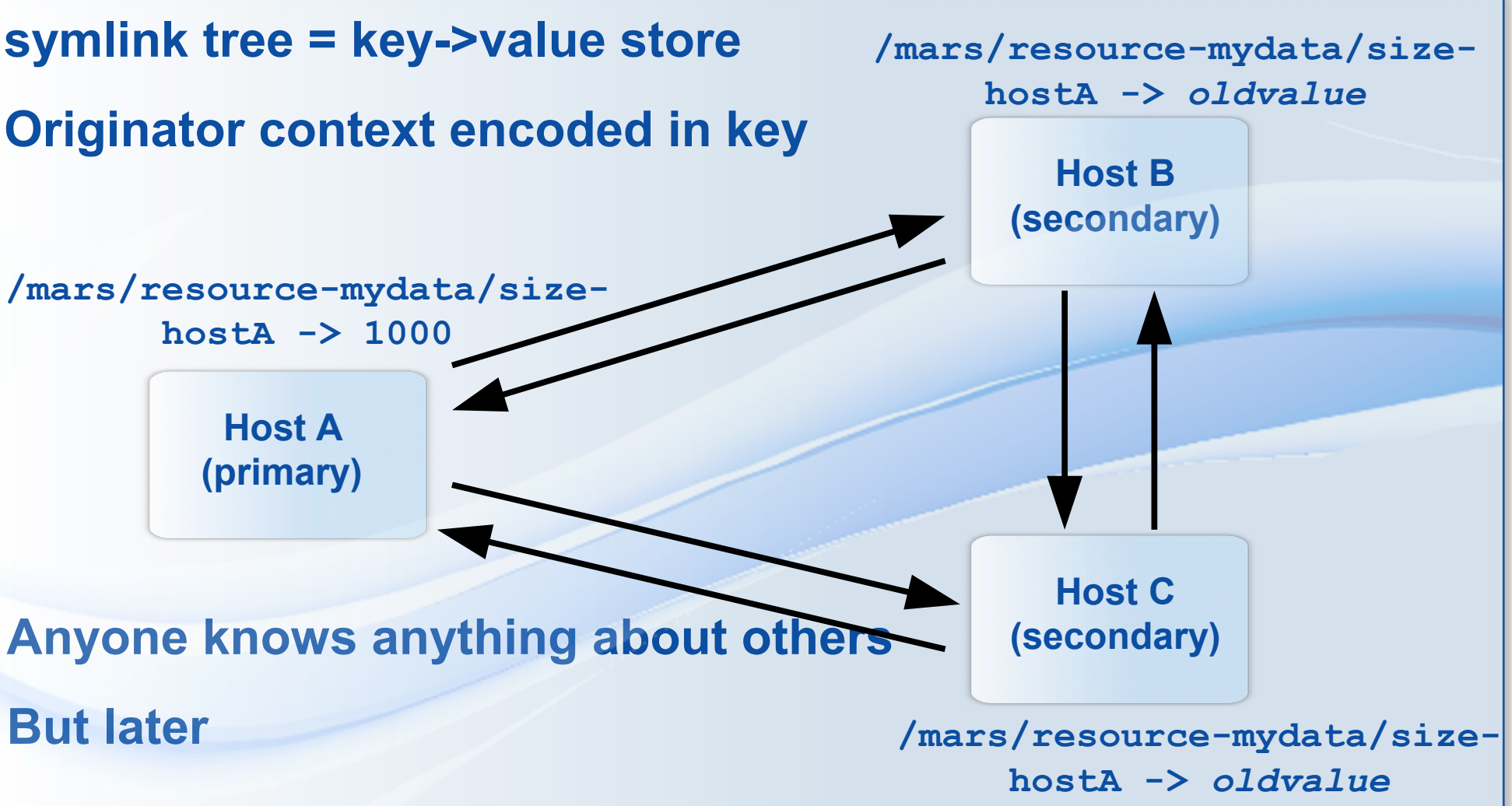


Solution: symlink tree + Lamport Clock => next slides

Metadata Propagation (2)

symlink tree = key->value store

Originator context encoded in key



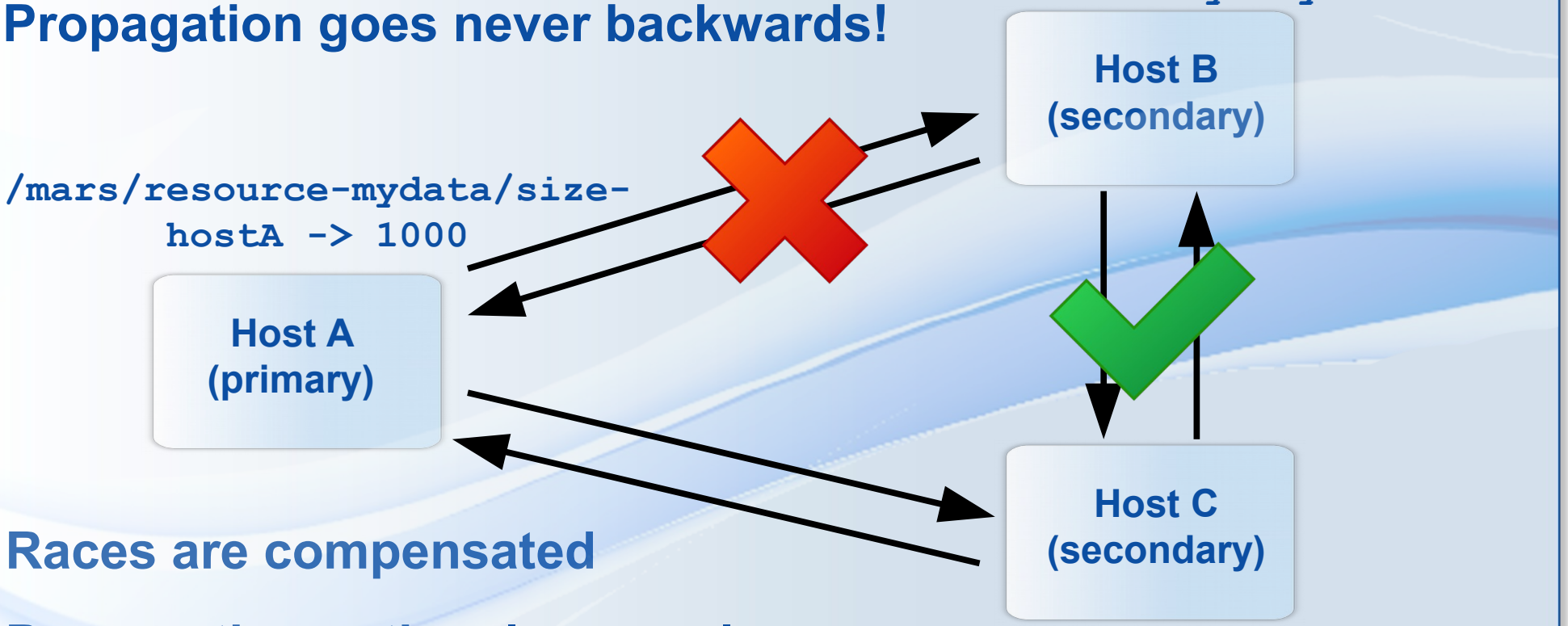
Anyone knows anything about others

But later

Metadata Propagation (3)

Lamport Clock = virtual timestamp

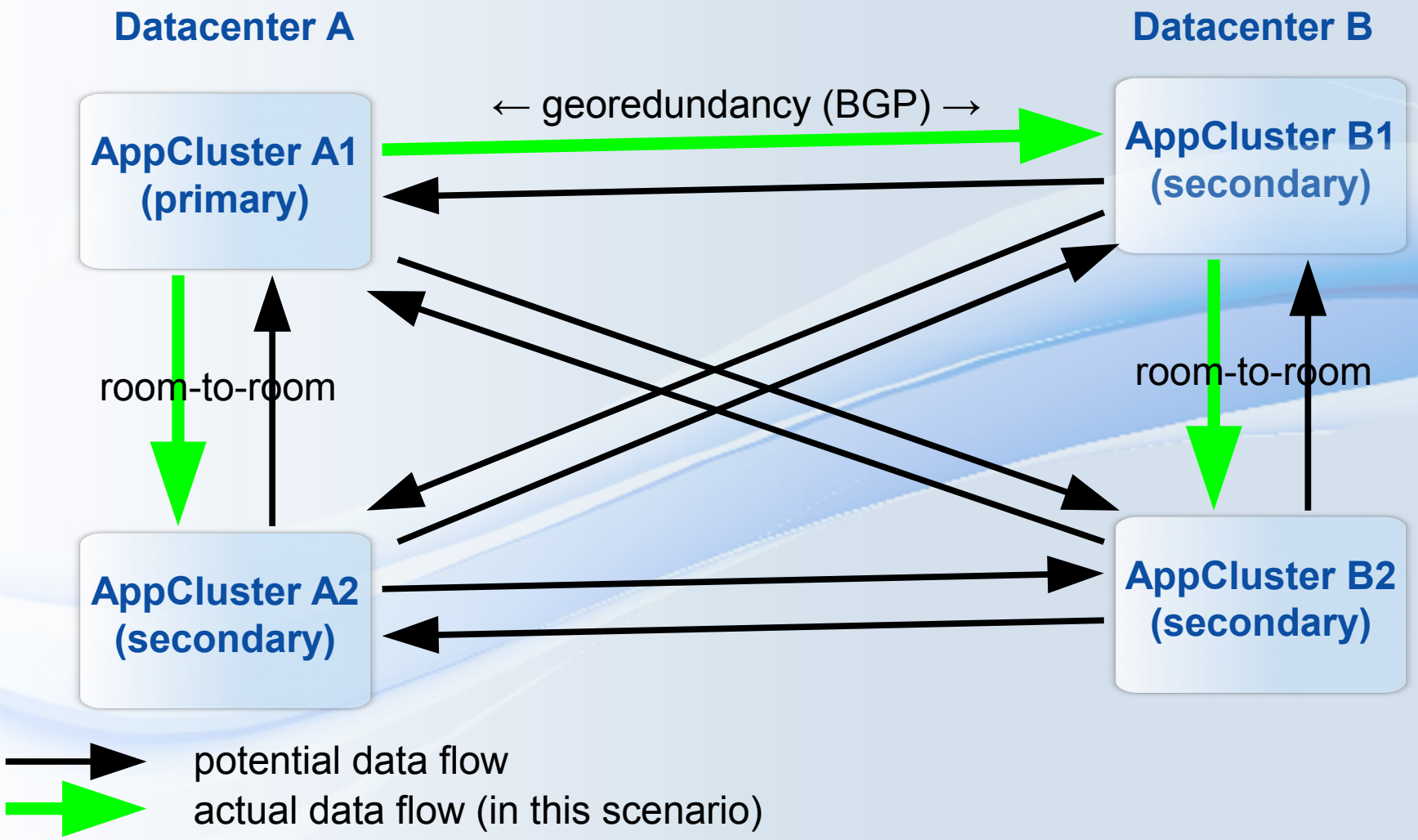
Propagation goes never backwards!



Races are compensated

Propagation paths play no role

Productive Scenario since 02/2014 (1&1 eShop / ePages)



Current Status

- Source / docs at
  `github.com/schoebel/mars`
  `mars-manual.pdf` ~ 100 pages
- light0.1stable productive on customer data since 02/2014
- MARS status Feb 2016:
  > 1700 servers (shared hosting + databases)

  > 2x8 Petabyte total

  ~ 10 billions of inodes in > 3000 xfs instances

  > 8 millions of operating hours
- Socket Bundling (light0.2beta)
  Up to 8 parallel TCP connections per resource

  easily saturates 1GBit uplink between Karlsruhe/Europe and Lenexa/USA
- WIP-remote-device
  `/dev/mars/mydata` can appear anywhere
- WIP-compatibility:
  no kernel prepatch needed anymore

  currently tested with vanilla kernels 3.2 … 4.4

Future Plans

- `md5` checksums on underlying disks
- Mass-scale clustering
- Database support / near-synchronous modes

- Further challenges:
  - community revision at LKML planned
  - replace symlink tree with better representation
  - split into 3 parts:
    - Generic `brick` framework
    - `XIO` / `AIO` personality (1st citizen)
    - MARS Light (1st application)
  - hopefully attractive for other developers!

Appendix

Use Cases DRBD+proxy vs MARS Light

DRBD+proxy
(proprietary)

Application area:

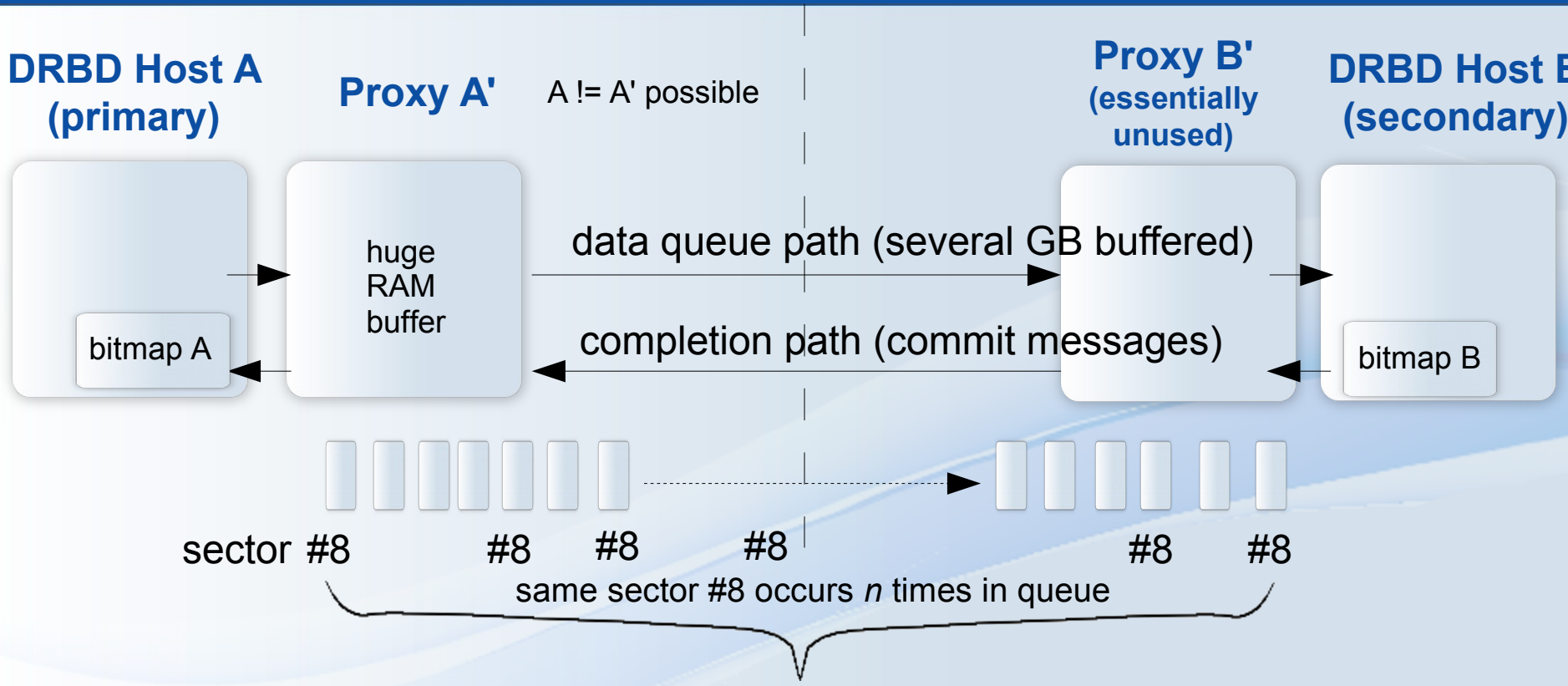
- Distances: any
- Aynchronously
  - **Buffering in RAM**
- Unreliable network leads to **frequent re-syncs**
  - RAM buffer gets lost
  - at cost of actuality
- **Long** inconsistencies during re-sync
- Under pressure: **permanent** inconsistency possible
- High memory overhead
- Difficult scaling to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

MARS Light
(GPL)

Application area:

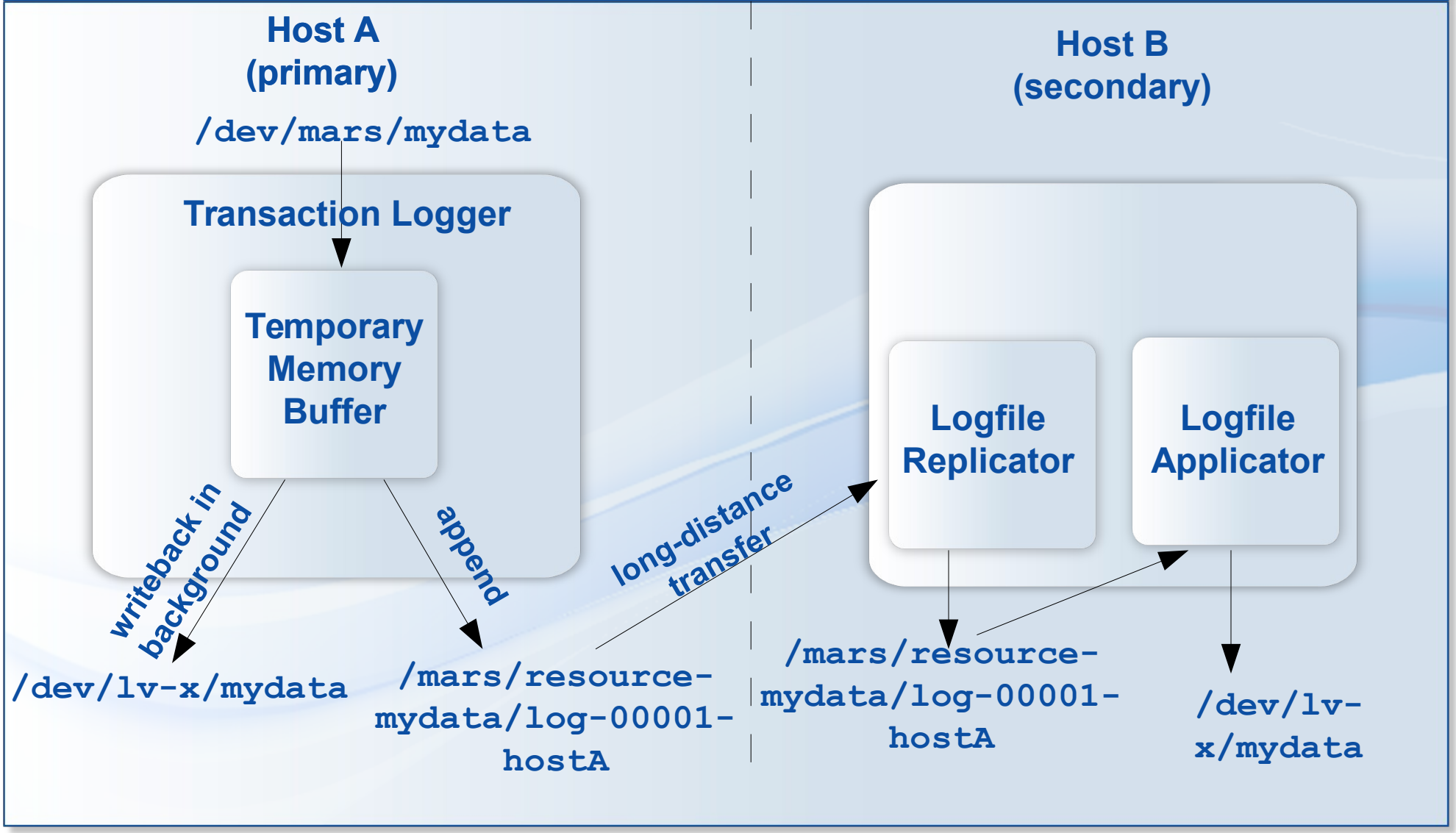
- Distances: **any** (>>50 km)
- Asynchronously
  - near-synchronous modes in preparation
- Tolerates **unreliable network**
- Anytime consistency
  - no re-sync
- Under pressure: no inconsistency
  - possibly at cost of actuality
- Needs >= 100GB in `/mars/` for transaction logfiles
  - dedicated spindle(s)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

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

External Software, Cluster Managers, etc

Userspace Interface `marsadm`

Framework Application Layer

MARS Light, MARS Full, etc

- MARS Light
- MARS Full
- ...

Framework Personalities

XIO = eXtended IO ≈ AIO

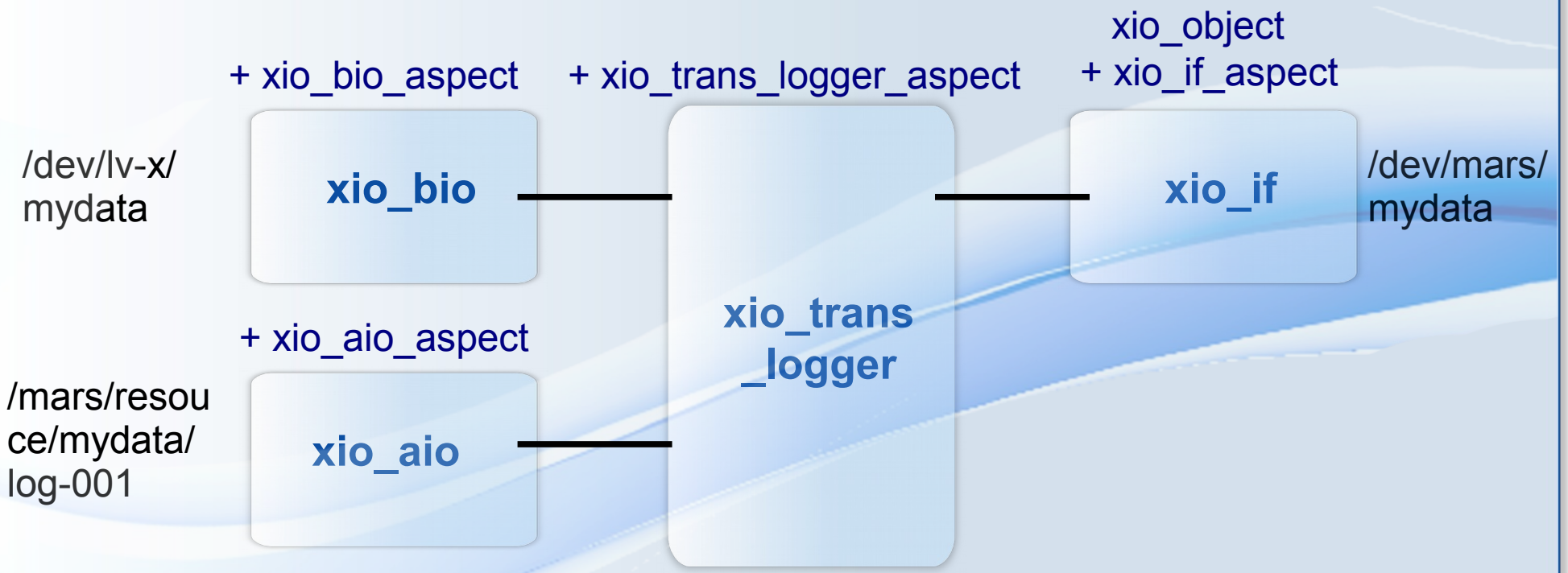
- XIO bricks
- future Strategy bricks
- other future Personalities and their bricks

Generic Brick Layer

IOP = Instance Oriented Programming
+ AOP = Aspect Oriented Programming

- Generic Bricks
- Generic Objects
- Generic Aspects

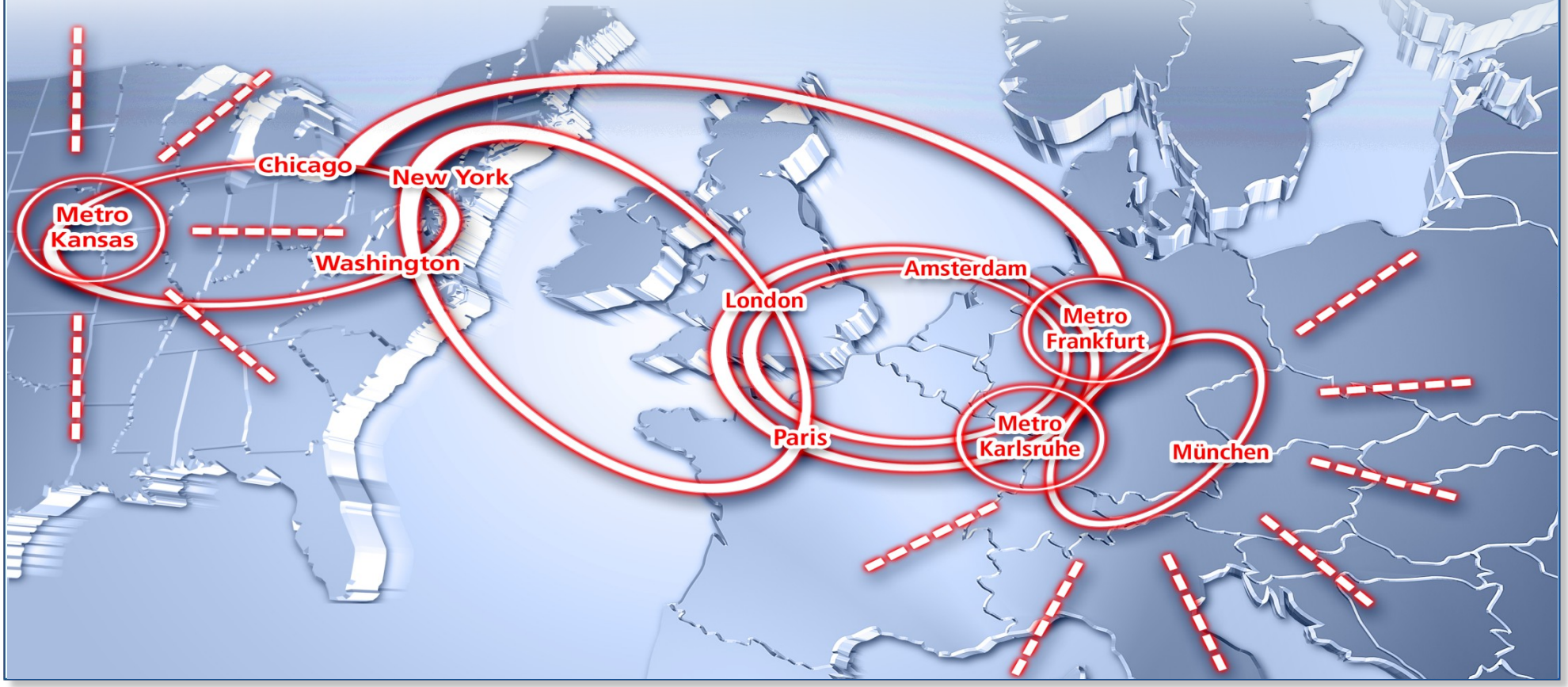
Bricks, Objects + Aspects (Example)



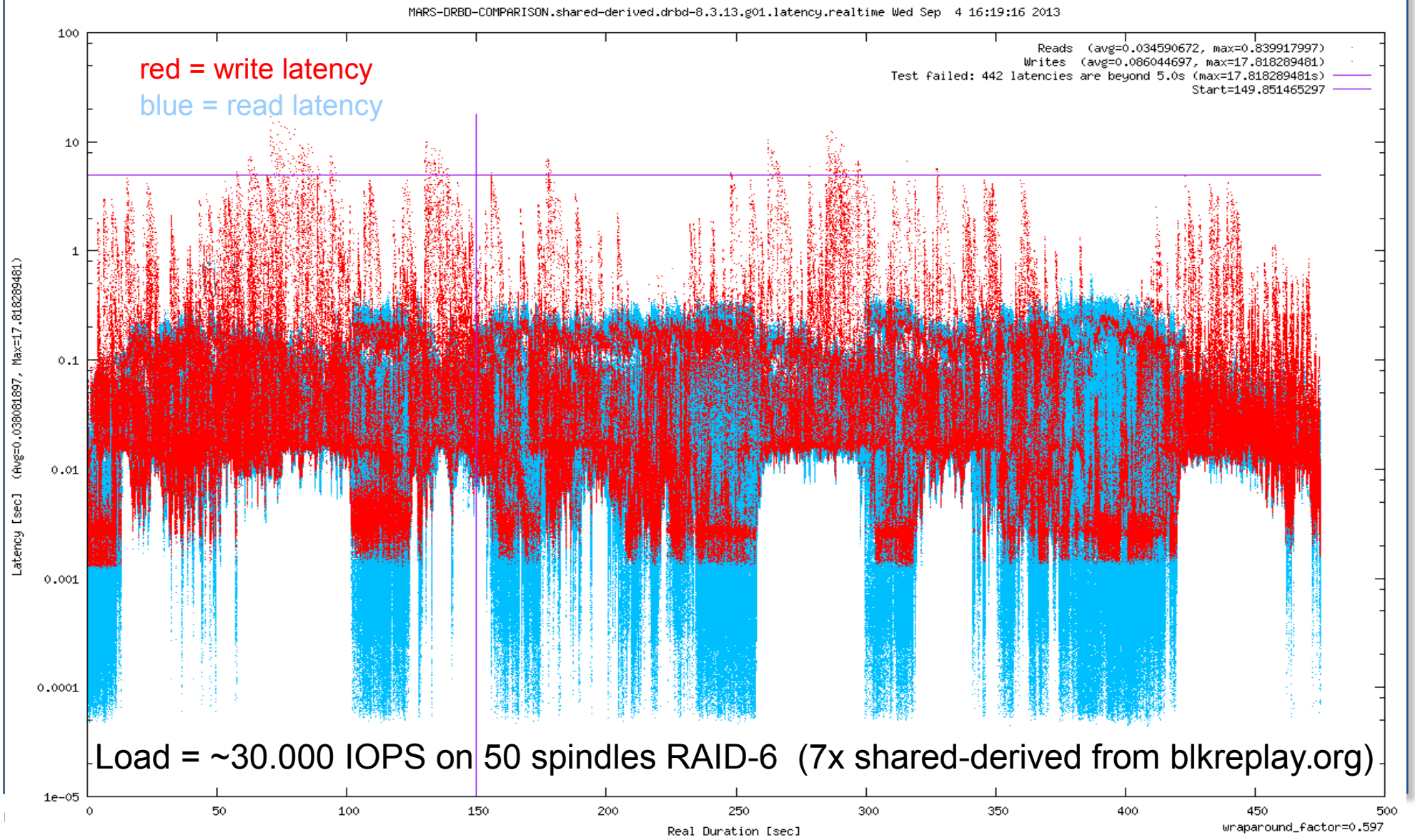
Aspects are automatically attached on the fly

Appendix: 1&1 Wide Area Network Infrastructure

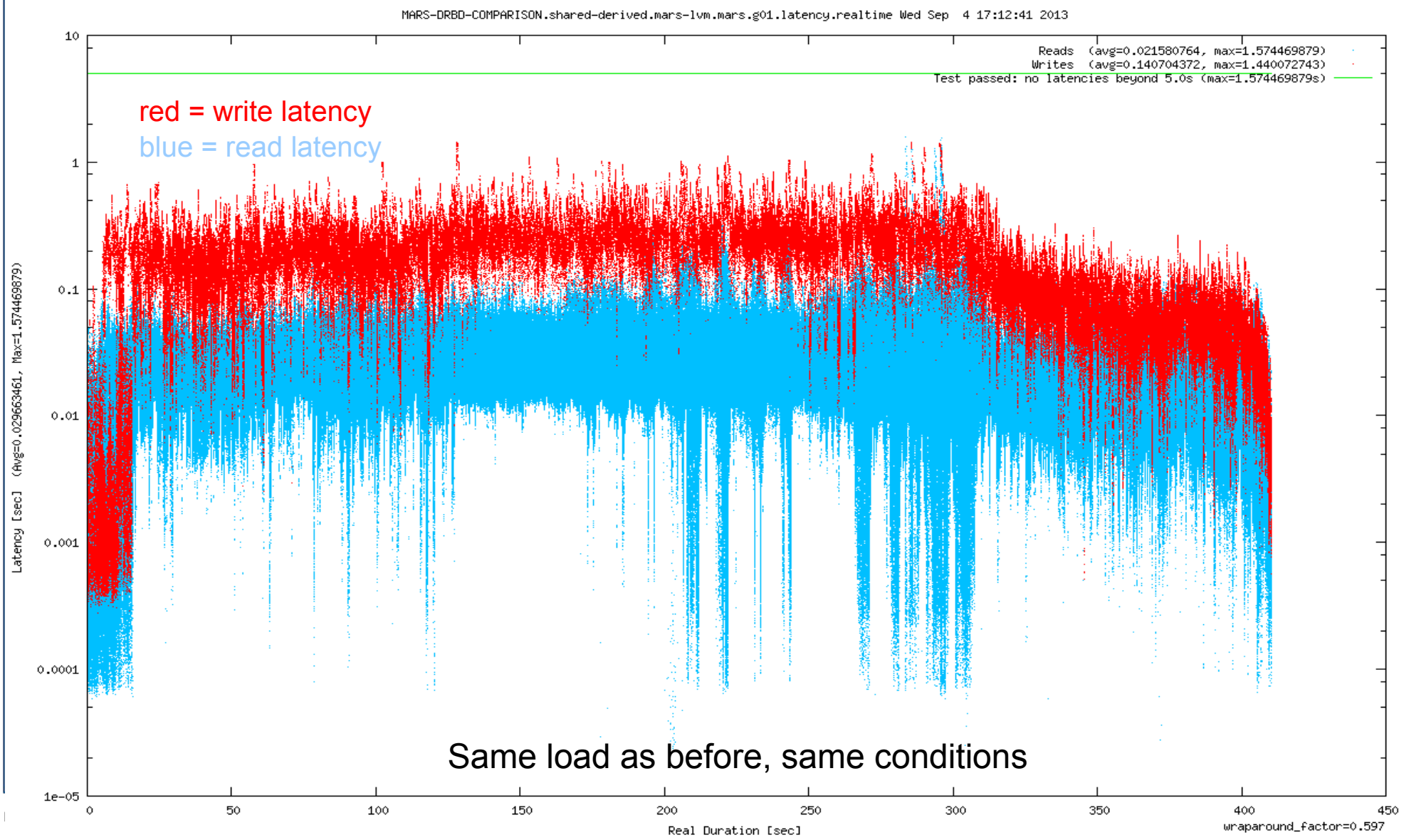
- Global external bandwidth > 285 GBit/s
- Peering with biggest internet exchanges on the world
- Own metro networks (DWDM) at the 1&1 datacenter locations



IO Latencies over loaded Metro Network (1) DRBD



IO Latencies over loaded Metro Network (2) MARS



Performance of Socket Bundling Europe↔USA