HAST

Paweł Jakub Dawidek
<pjd@FreeBSD.org>

FreeBSD

FreeBSD Foundation

OMCnet Internet Service GmbH

TransIP BV

High Availability

services available all the time

how hard can it be? 

The challenge

The most common reason for HA cluster outages are problems in HA implementations

HAST

replicates data over the network

file system independent

(block-level)

recovers quickly

detects split-brain conditions

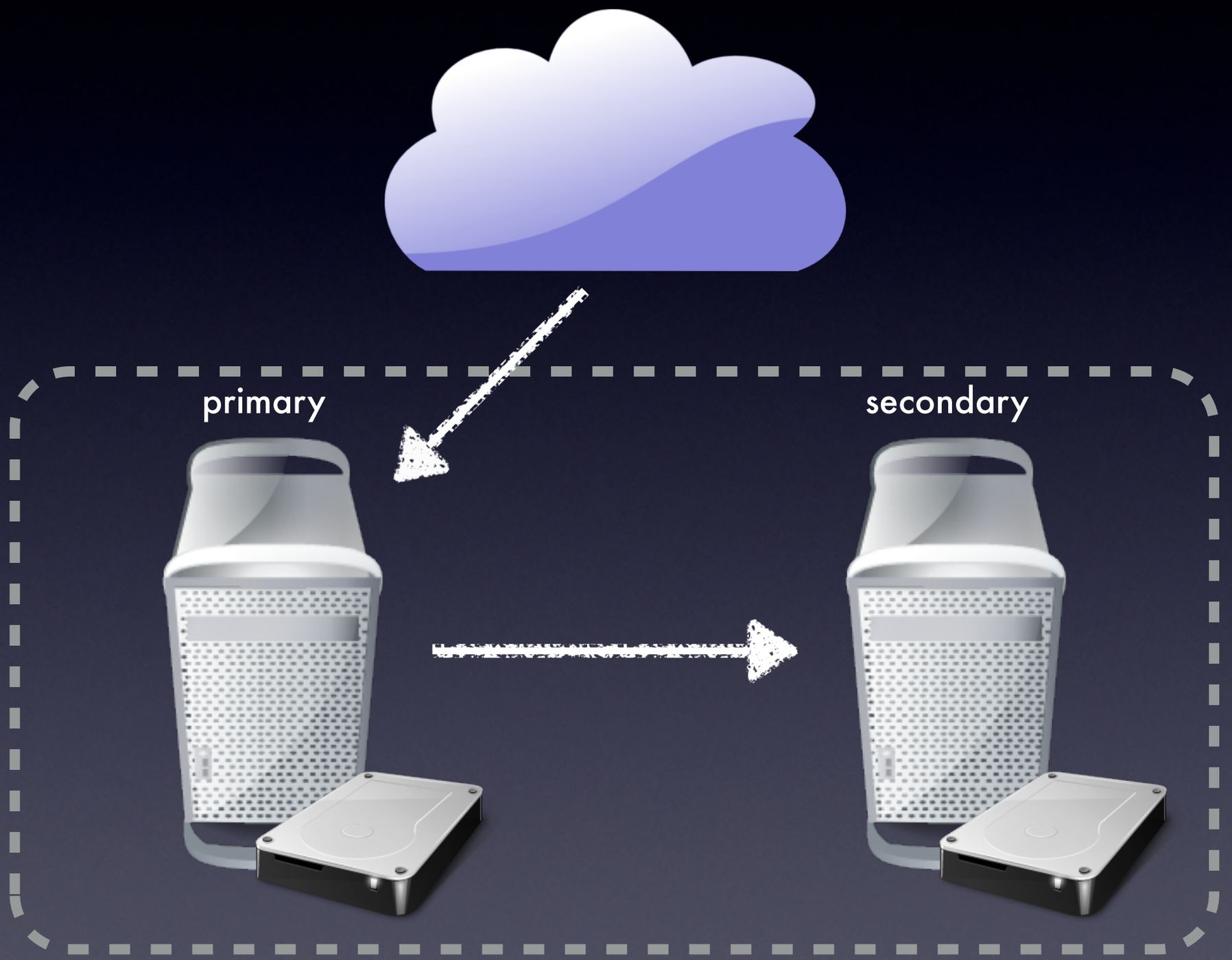
does not decide about its role

2

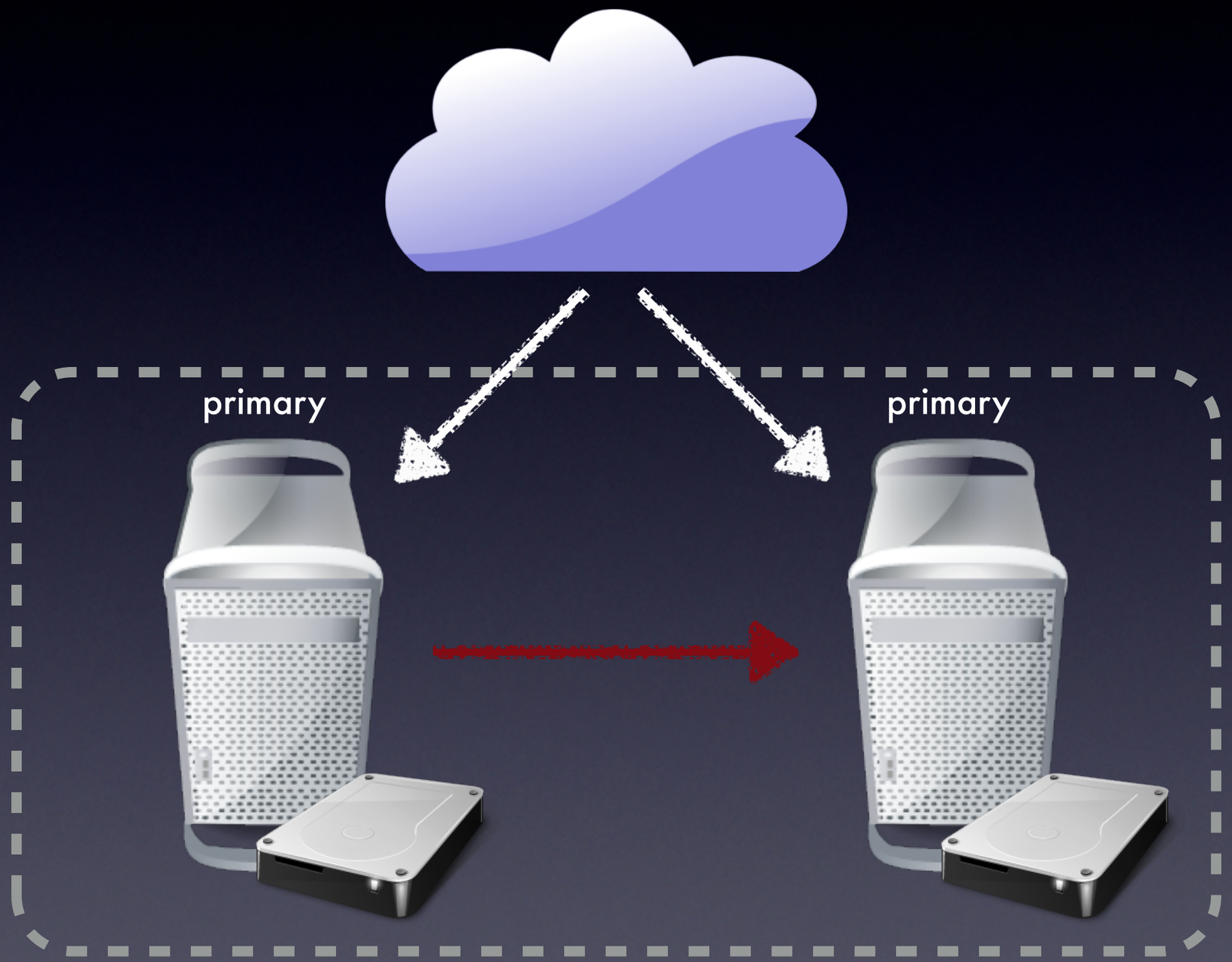
- primary
- secondary

2

HAST



split-brain



write

- comes from the kernel to userland hastd
- hastd has to write data to local disk and send it to secondary node

naive mode

- write data locally and send it to secondary
- report success

what could possibly go wrong? 

superslow mode

- mark extent as dirty
- write data locally and send it to secondary
- secondary ack on data write
- mark extent as clean
- report success

3

- fullsync
- memsync
- async

3

fullsync mode

- mark extent as dirty
- write data locally and send it to secondary
- secondary ack data write
- do not mark extent as clean
- report success

memsync mode

- mark extent as dirty
- write data locally and send it to secondary
- secondary ack data receive
- report success
- secondary ack data write
- do not mark extent as clean

async mode

- mark extent as dirty
- write data locally and send it to secondary
- report success
- secondary ack data write
- do not mark extent as clean

how to break memsync?

- primary receives write from the kernel
- primary write data locally and send it to secondary
- secondary ack data receive
- primary reports success to the application
- secondary dies before storing the data on disk
- primary dies and never will be recovered
- slaves recovers, but is missing some writes already confirmed to the application

Configuration

/etc/hast.conf

```
resource data {
        local /dev/ada0
        on hosta {
                remote 10.0.0.1
        }
        on hostb {
                remote 10.0.0.2
        }
}
```

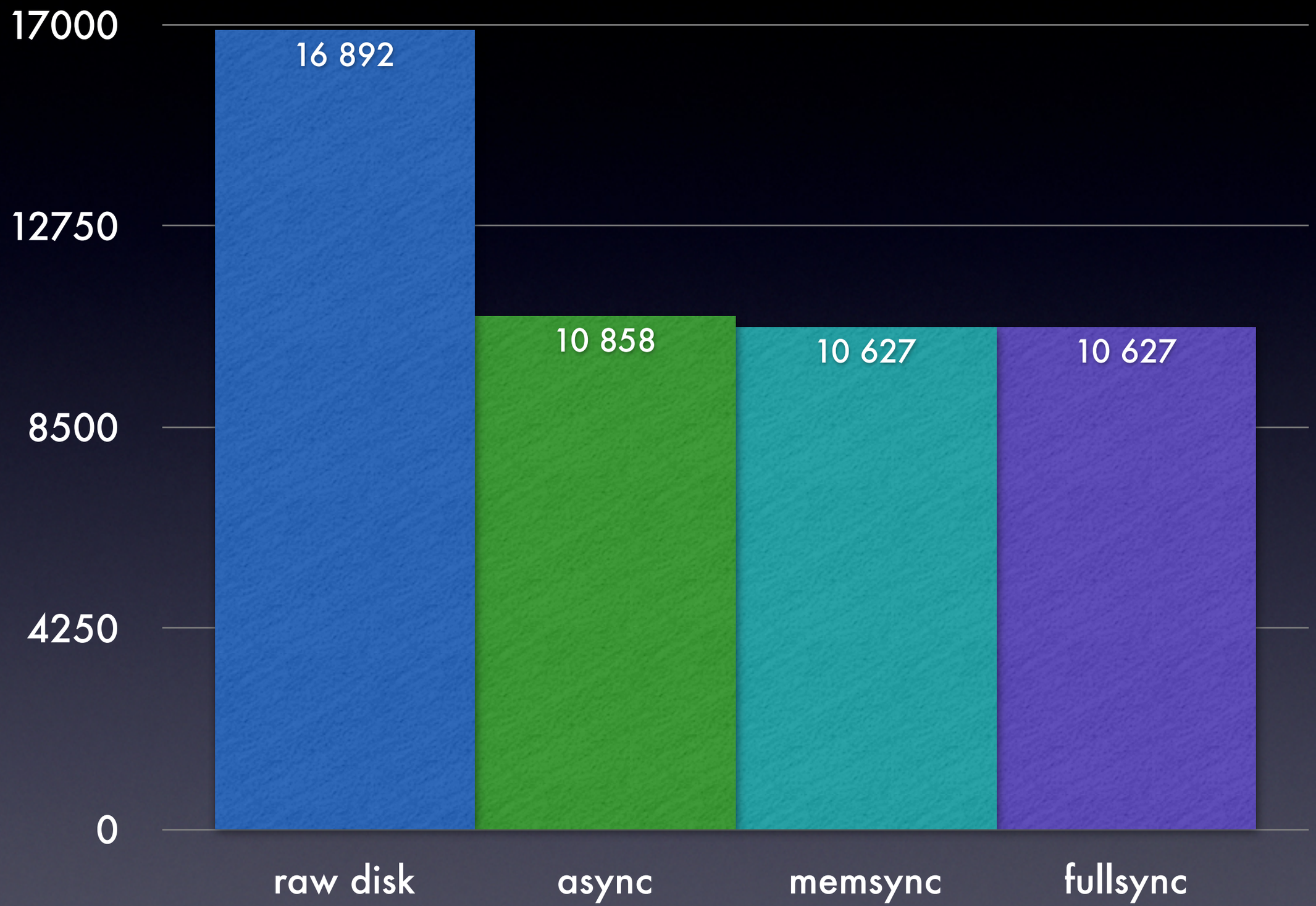
Quick start

```
hostb# hastctl create data

hostb# hastd

hostb# hastctl role secondary data


hosta# hastd

hosta# hastctl create data

hosta# hastctl role primary data
```

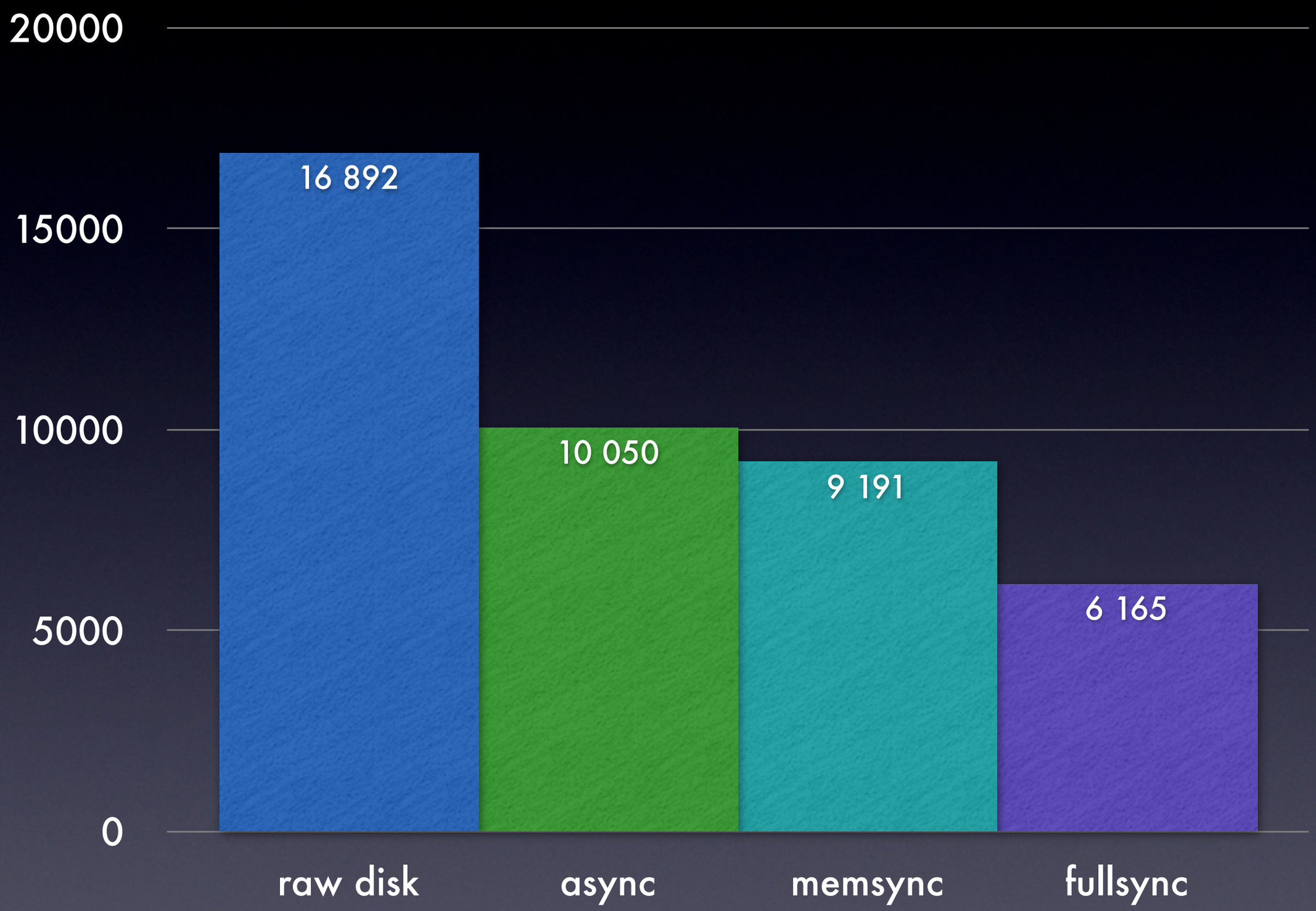
```
hosta# newfs /dev/hast/data

hosta# mount /dev/hast/data /data
```

Performance (latency)

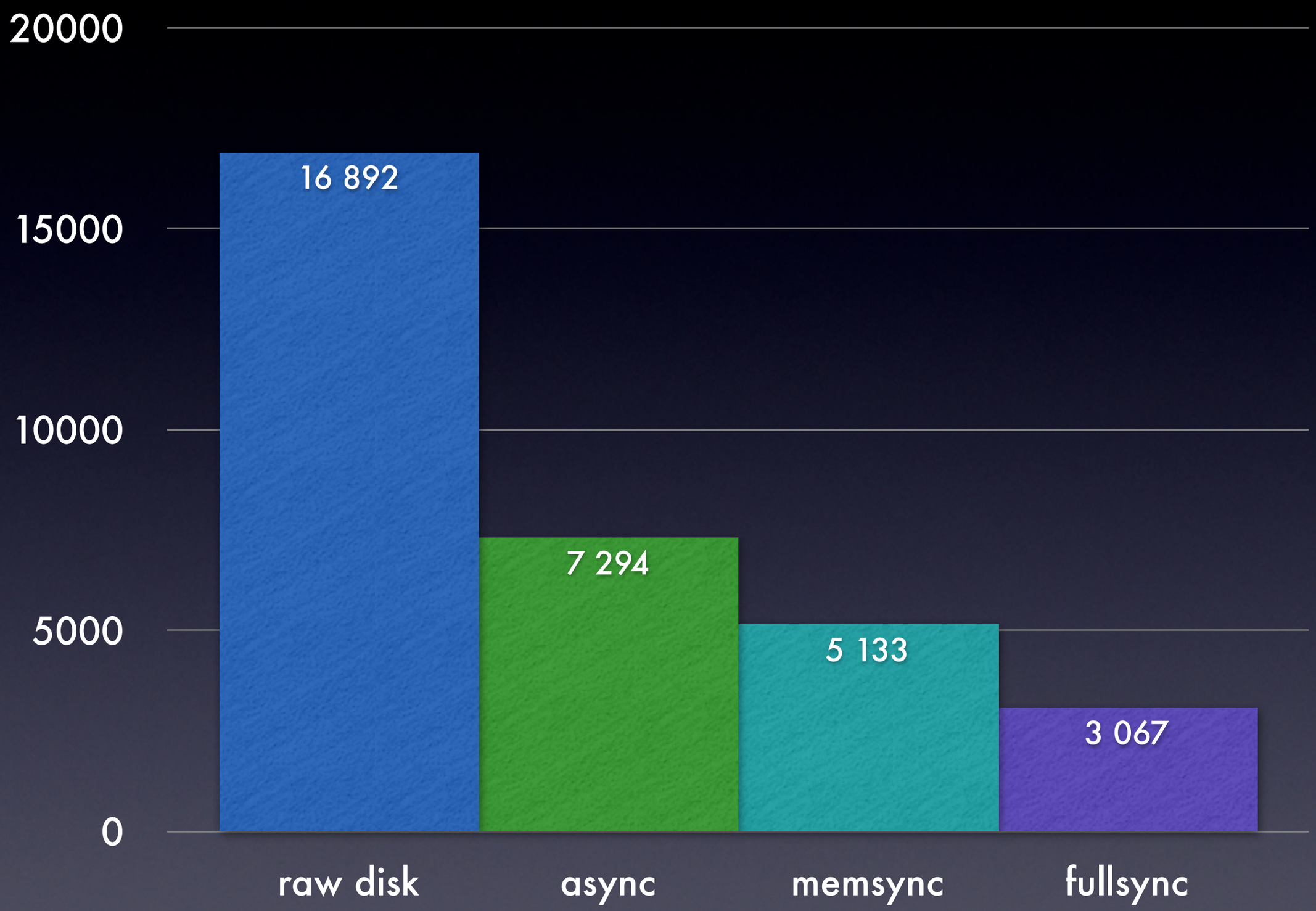
IOPS, latency, no secondary



IOPS, latency, secondary over lo0



IOPS, latency, secondary over 1Gb

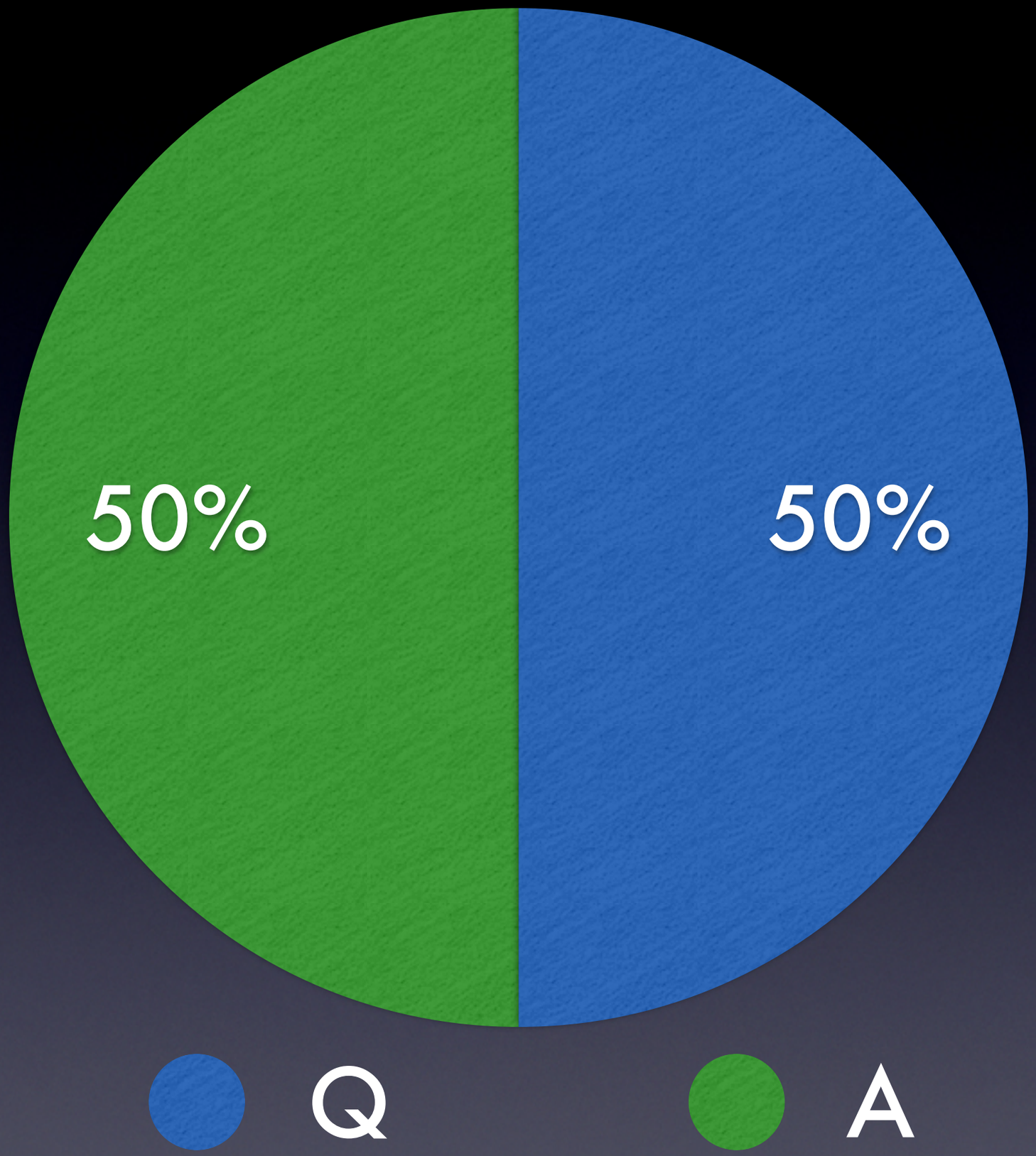


Recent work

>300

- many bug fixes
- SIGHUP handling (in the least intrusive way)
- synchronization avoidance where possible
- compression support (hold and lzf)
- checksum support (crc32 and sha256)
- hooks (run external program on various events)

- async, memsync modes
- direct reads
- sandboxing (capsicum+)jail+setuid+setgid
- internal keepalive
- use of printfs extensions
- possibility to define source connection address
- pidfile path in config
- metaflush


50%
50%
Q
A