dsync: Efficient Synchronization of Multi-Gigabyte Binary Data

Thomas Knauth, Christof Fetzer

November 7 @ LISA 2013

What's the problem?

- doing backups is important
- backup process should be fast and not waste resources
- just reading 4 TB of data (single disk) takes > 6 hours
- periodic, differential, state synchronization with minimal resource consumption

How do you do your backups?

Picking the right tool

dirvish

ZFS

copy

Time Machine

lightweight rsync

Dropbox

rsync

dsync

rdiff-backup

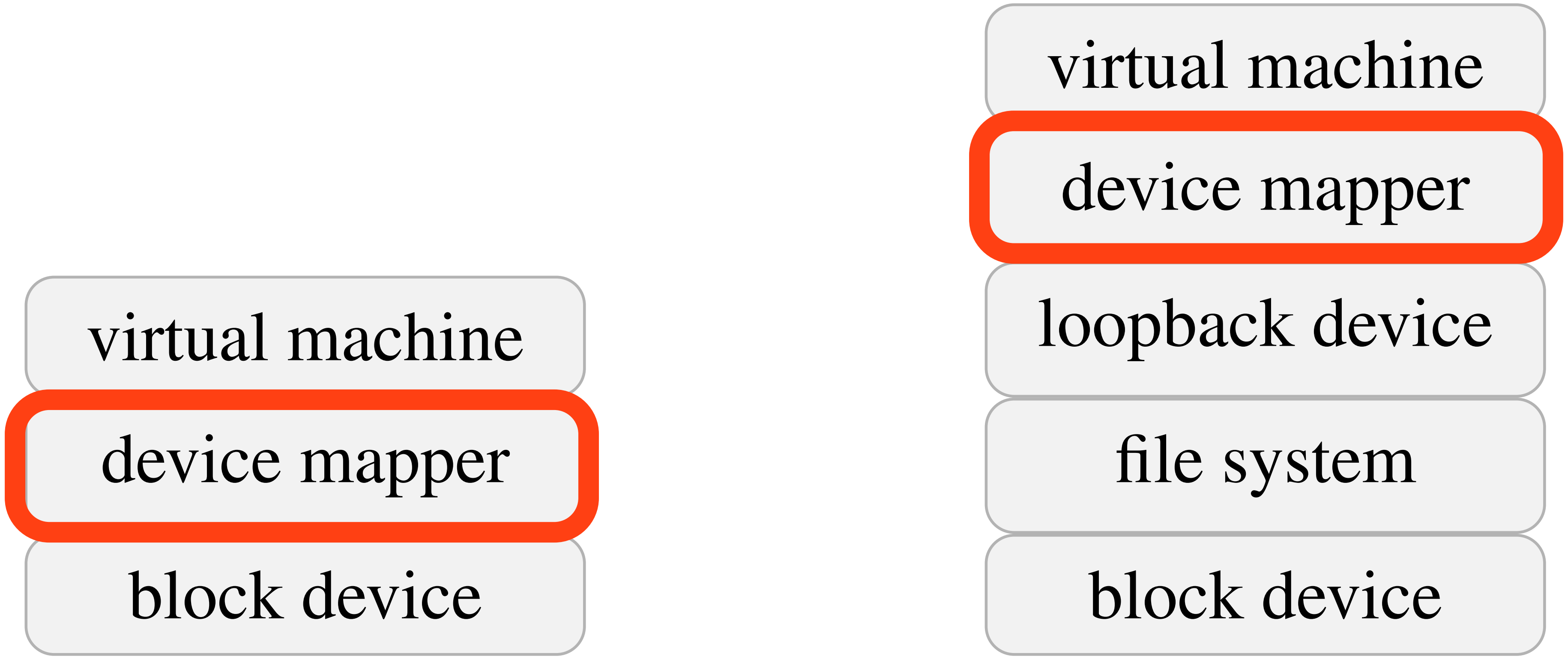
The generalist: rsync

- operates on file system level
- goal is to minimize data transfer
- has significant computational overhead for large (GB) files
- familiar to system administrators

The new guy: dsync

- kernel-space modification
- supplemented by user-space tools
- operates on block device level
  - independent of file system

Where does it fit in the stack?



virtual machine
device mapper
block device

virtual machine
device mapper
loopback device
file system
block device

How is dsync implemented?

- modification to device mapper module (drivers/md/dm-linear.c)
- one bit per 4 KiB block
- for example, 4 TiB disk requires 128 MiB bit vector
- in-memory data structure

Interfacing with dsync

- virtual file in /proc
- user-space tools to extract and merge block from/into device
- can build shell pipeline:

```
# dmextract srcdev | ssh remote dmmerge targetdev
```

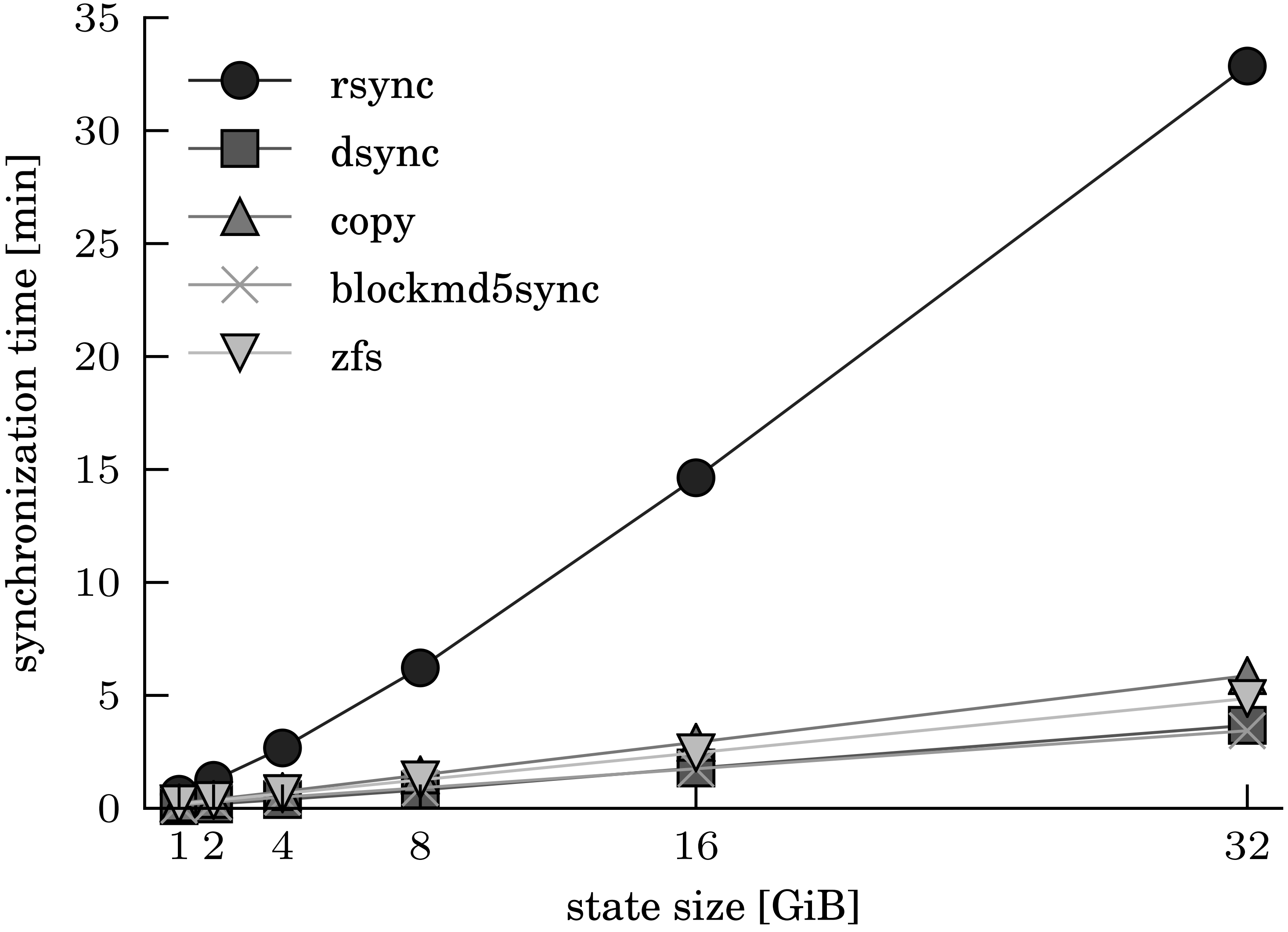
How was dsync evaluated?

- mix of synthetic and real world workloads
- synthetic: random block modifications
- real world: virtual machine disks (RUBiS) and Microsoft Research traces
- two machines (source and target) connected via switched Gigabit Ethernet

Sync times for various tools

SSD, Figure 3

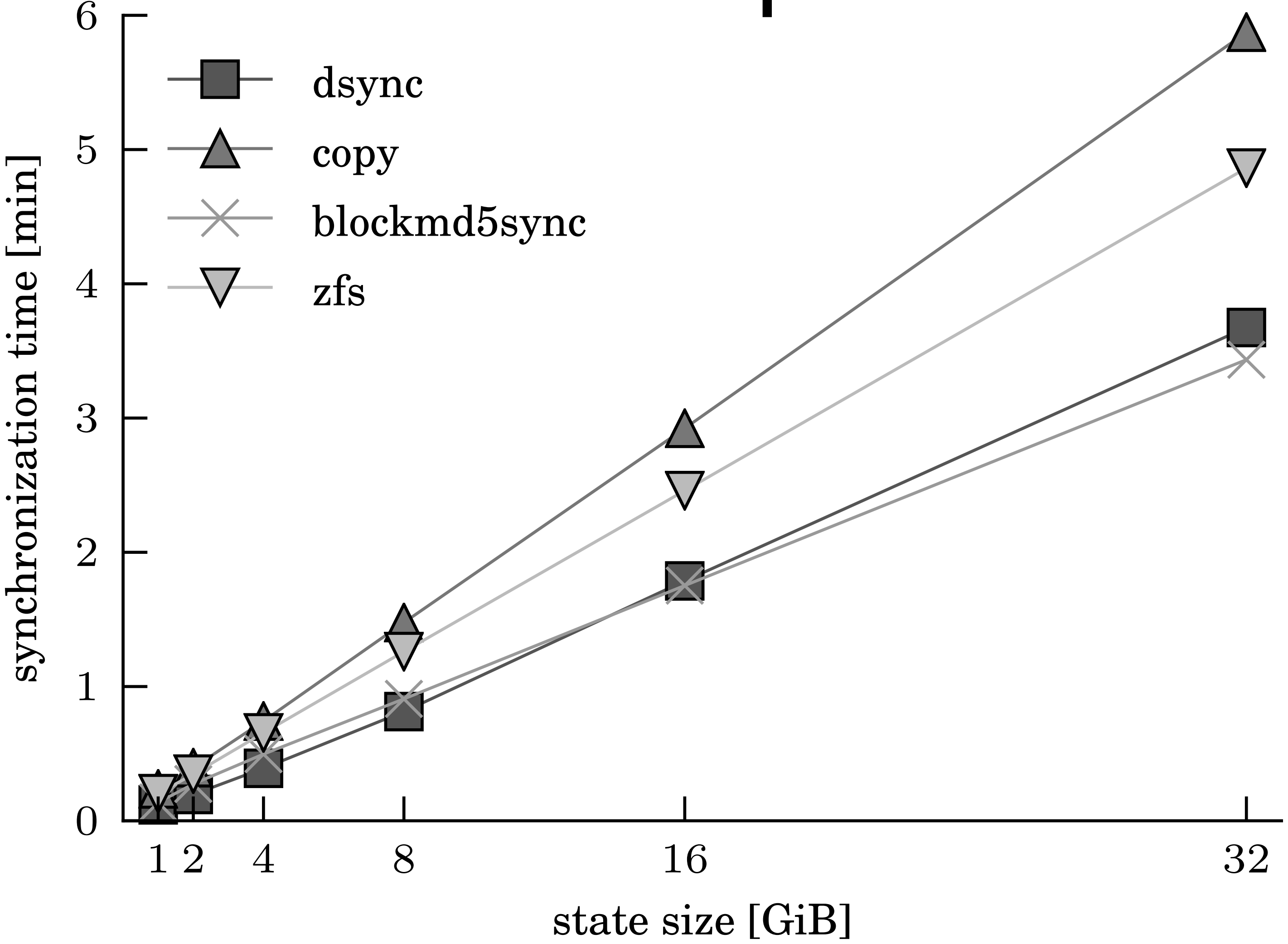
10% block updates



Sync times for various tools

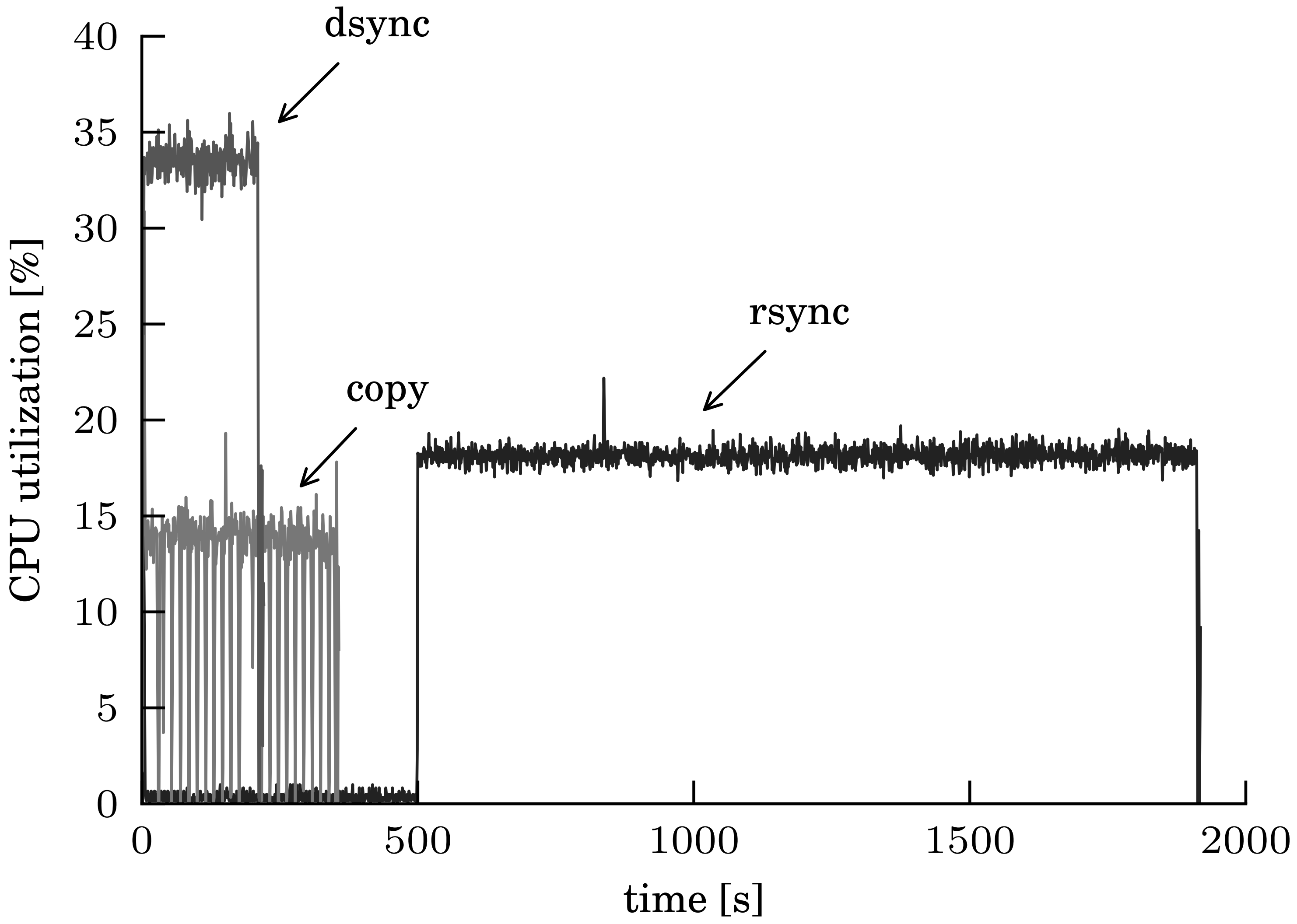
SSD,
Figure 3

10% block updates



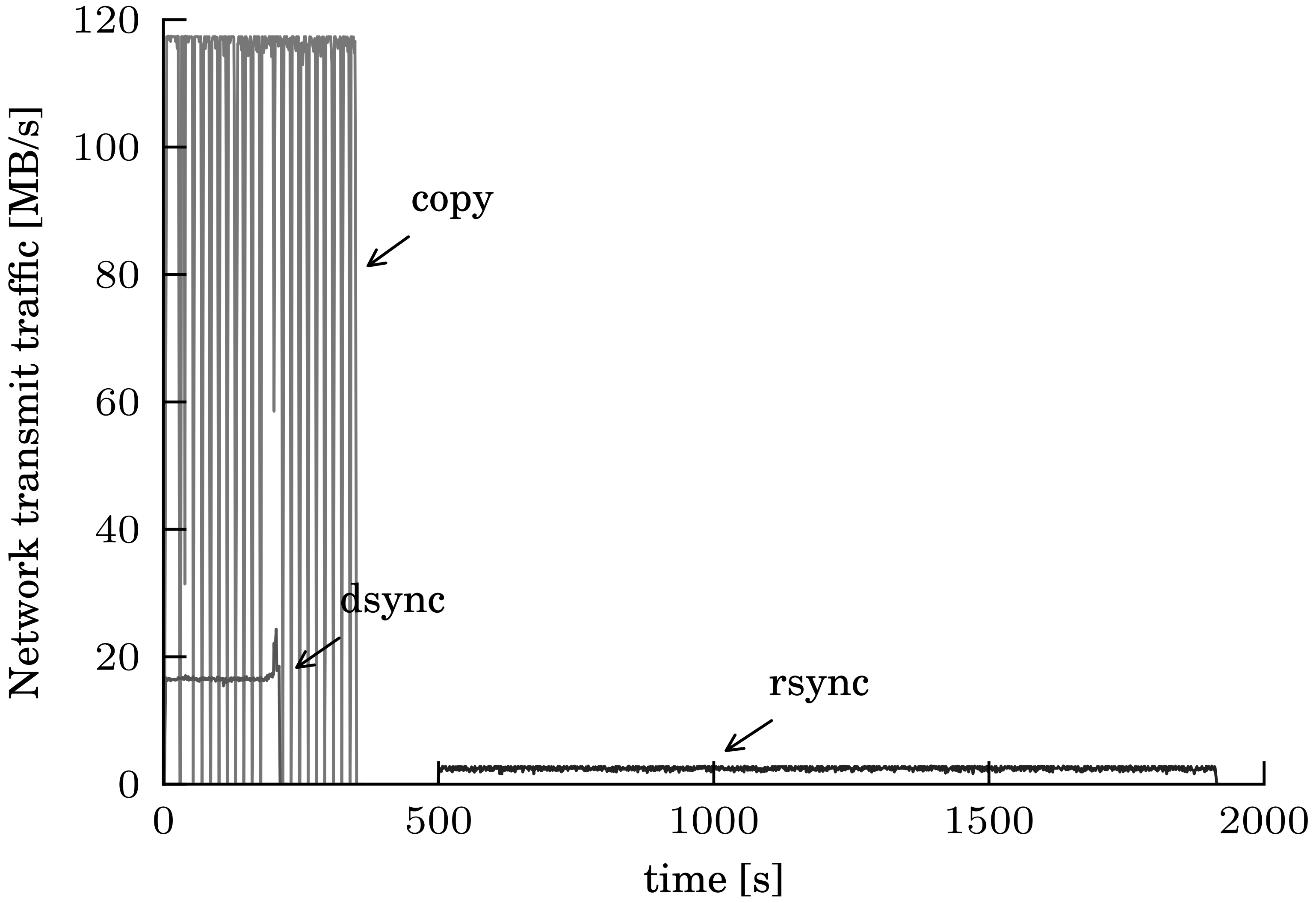
CPU utilization at the source

32 GiB,
SSD,
Figure 4



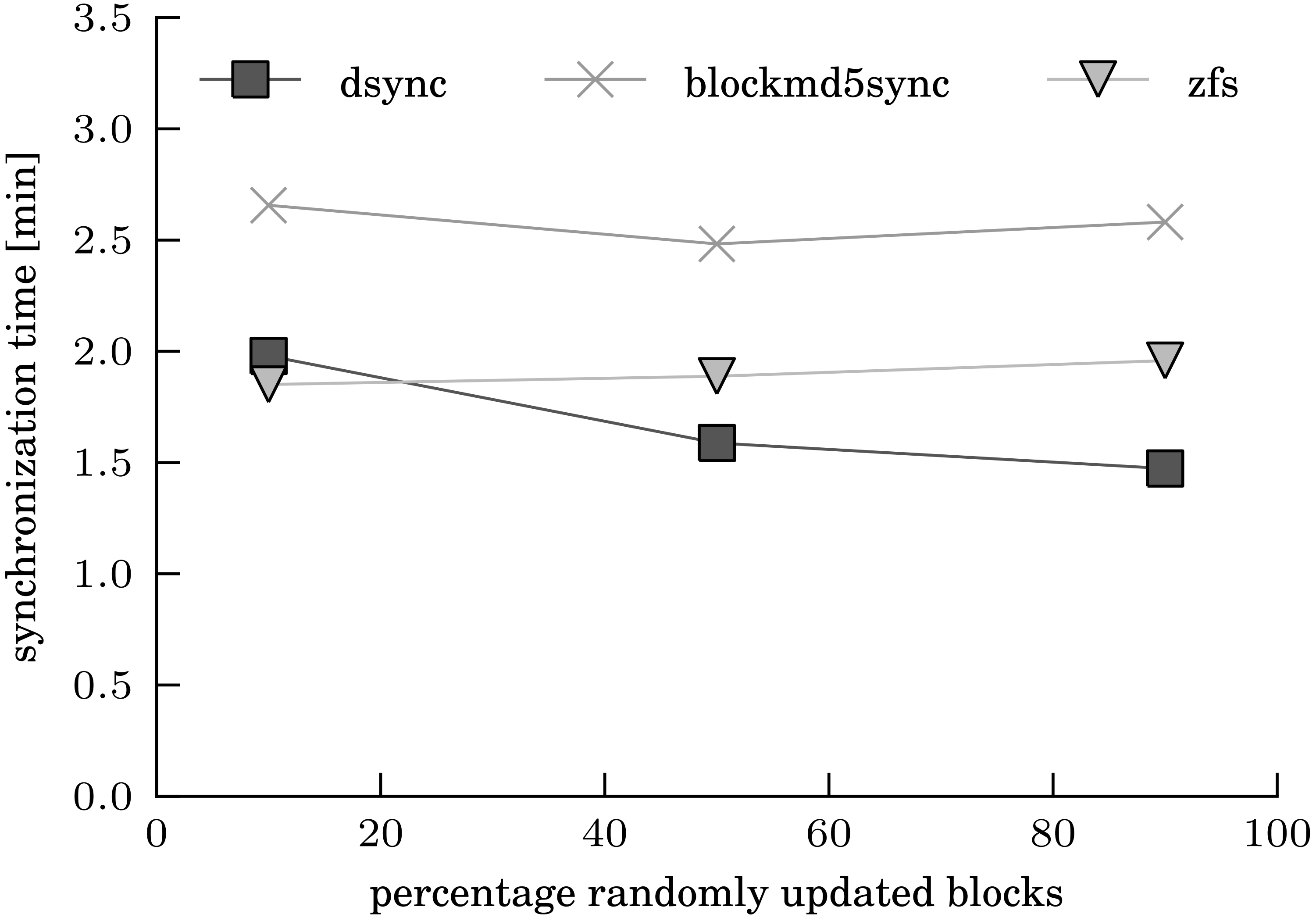
Network utilization at source

32 GiB,
SSD,
Figure 5



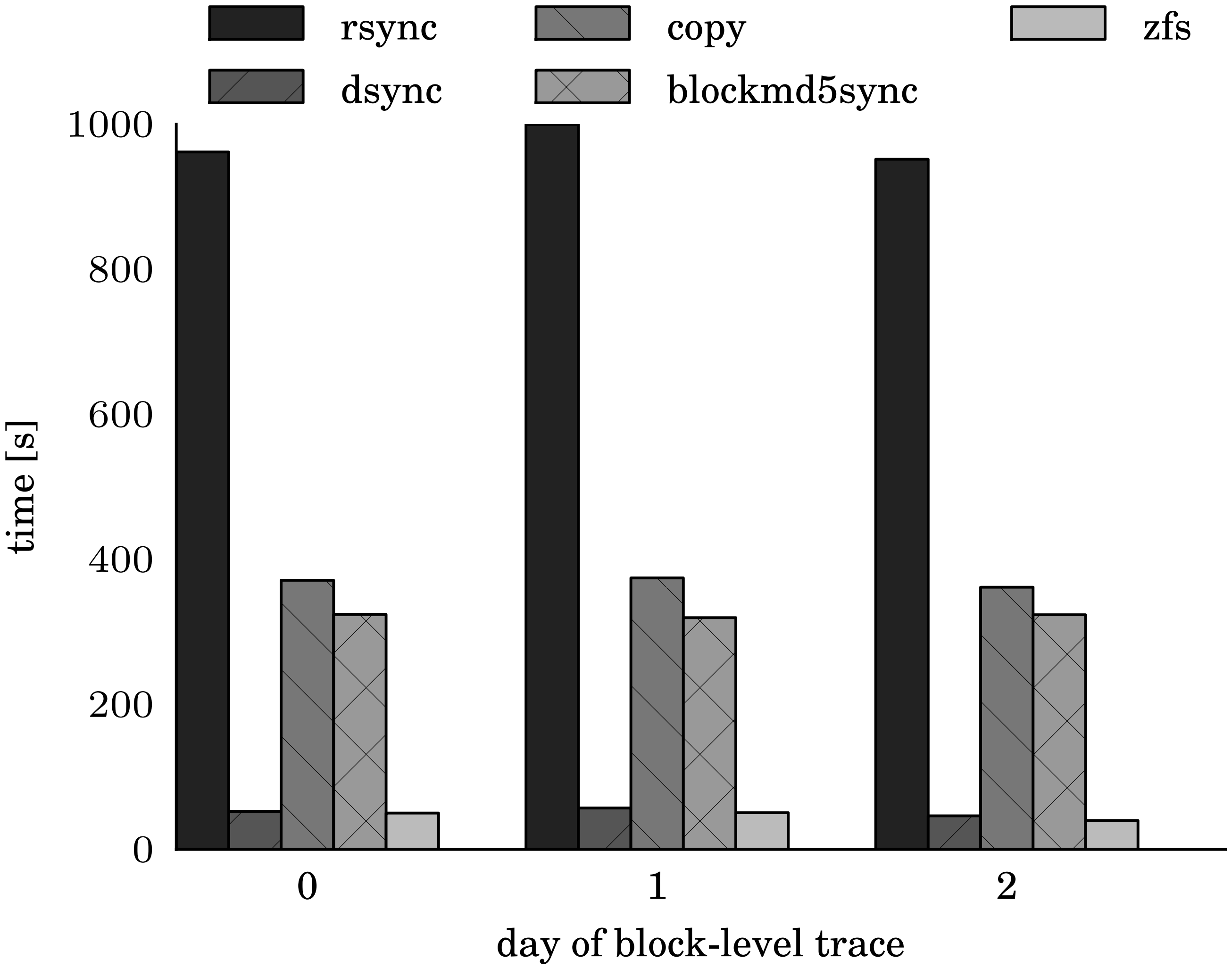
More updates decrease sync time slightly

8 GiB,
HDD,
Figure 6



Sync time on real-world traces

32 GiB,
HDD,
Figure 10



Summary

- tool to synchronize data at the block device level
- file system agnostic
- trades space for CPU and disk I/O bandwidth: track modifications instead of computing checksums

Open Science

- http://bitbucket.org/tknauth/devicemapper/

Help!

Work for PLX Technology or know anyone who works for them? Please come and talk to me!