BabuDB: Fast and Efficient File System Metadata Storage

Jan Stender, Björn Kolbeck,
Mikael Högqvist

Zuse Institute Berlin

Felix Hupfeld

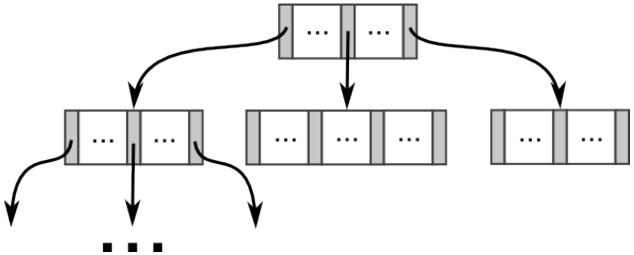
Google GmbH Zurich

Motivation

- Modern parallel / distributed file systems:
  - Huge numbers of files and directories
  - Many storage servers but few metadata servers
- Examples:
  - Lustre, Panasas Active Scale, Google File System
- Metadata access critical wrt. system performance
  - ~75% of all file system calls are metadata accesses
  - Metadata servers are bottlenecks

Motivation

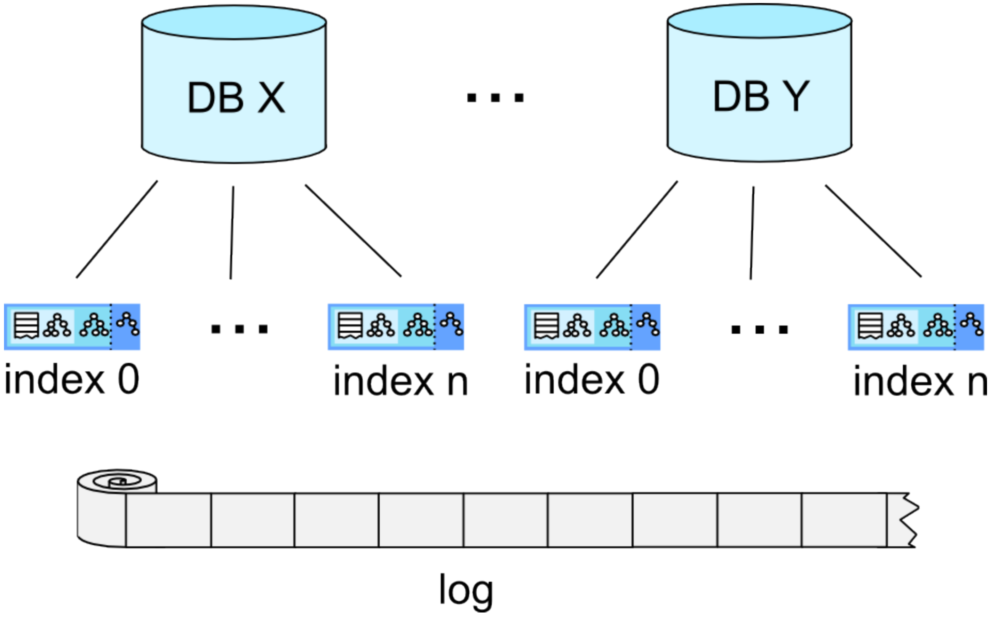
- B-tree-like data structures used for metadata storage
  - ZFS, btrfs, Lustre, PVFS2



- Downsides:
  - Hard to implement and test, high code complexity
  - Multi-version B-trees even more complex
  - On-disk re-balancing expensive

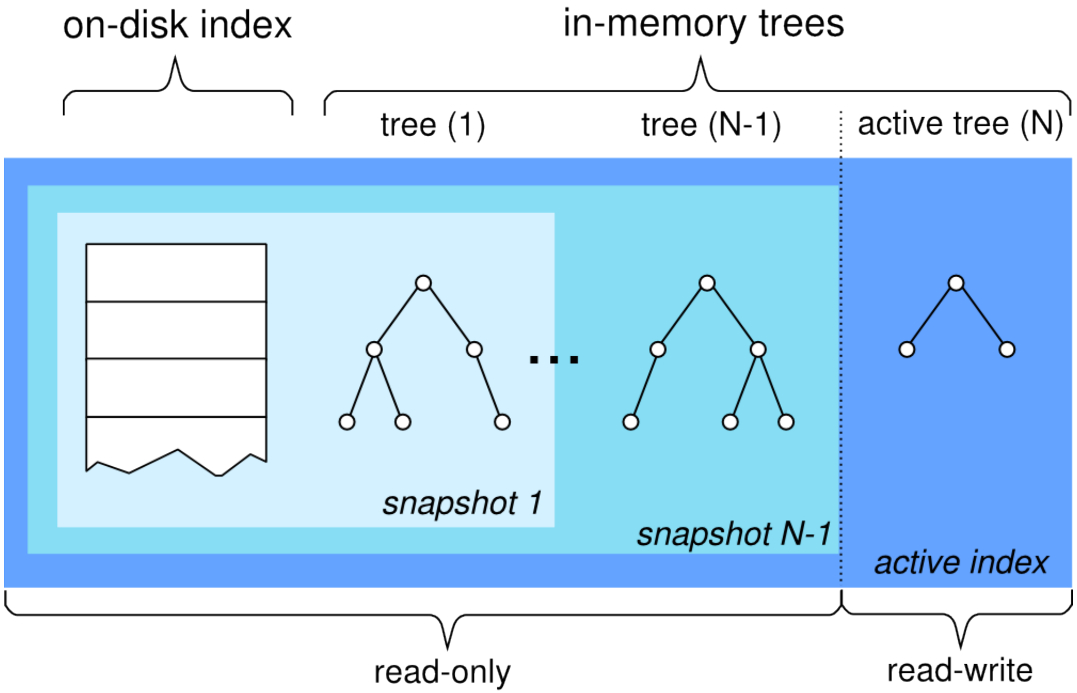
BabuDB

– Key-value store

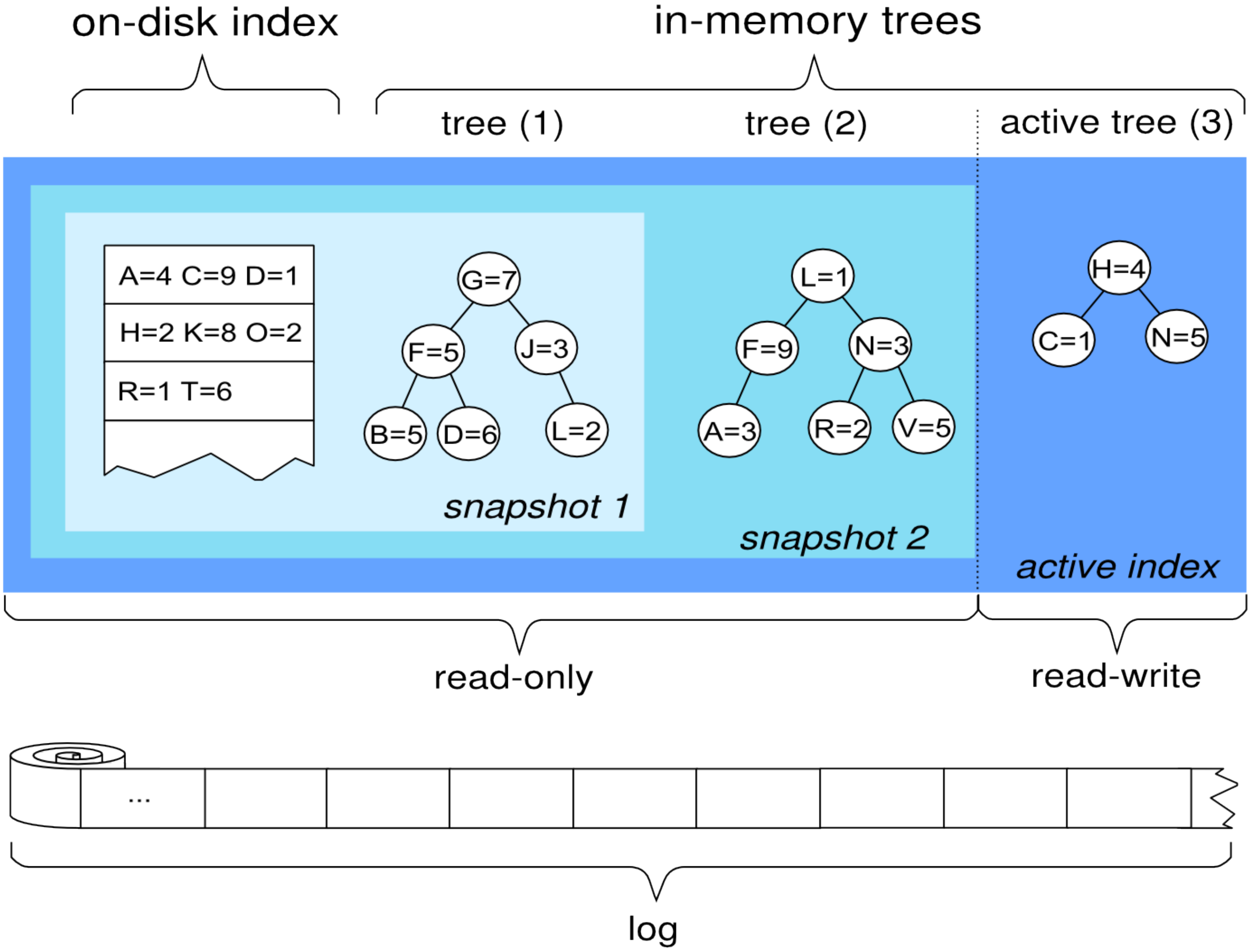


– FS metadata: key-value pairs stored in DB indices

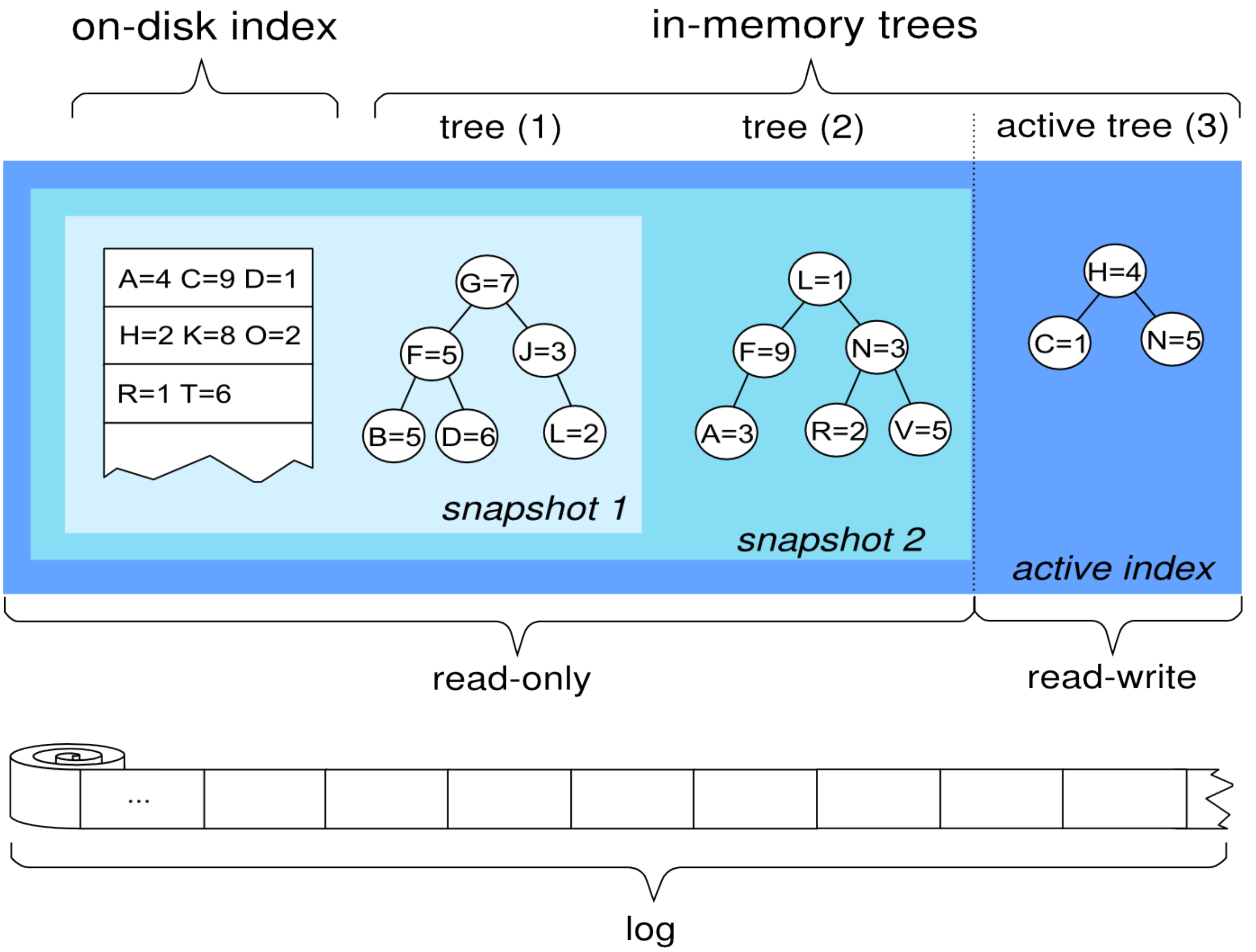
BabuDB: Index



Example

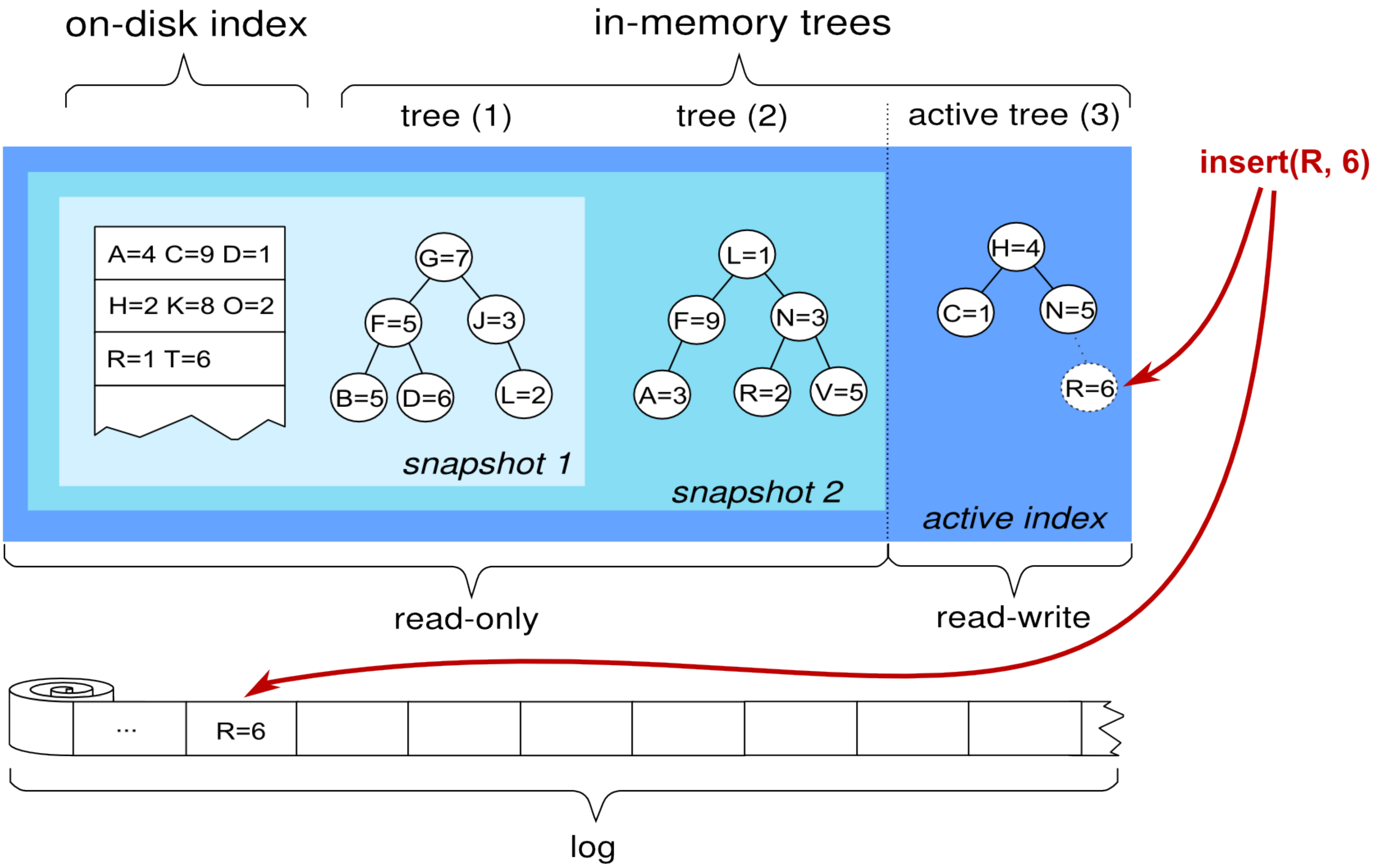

on-disk index
in-memory trees
tree (1)
tree (2)
active tree (3)
A=4 C=9 D=1
H=2 K=8 O=2
R=1 T=6
G=7
F=5
J=3
B=5
D=6
L=2
snapshot 1
L=1
F=9
N=3
A=3
R=2
V=5
snapshot 2
H=4
C=1
N=5
active index
read-only
read-write
...
log

Example: Insertions

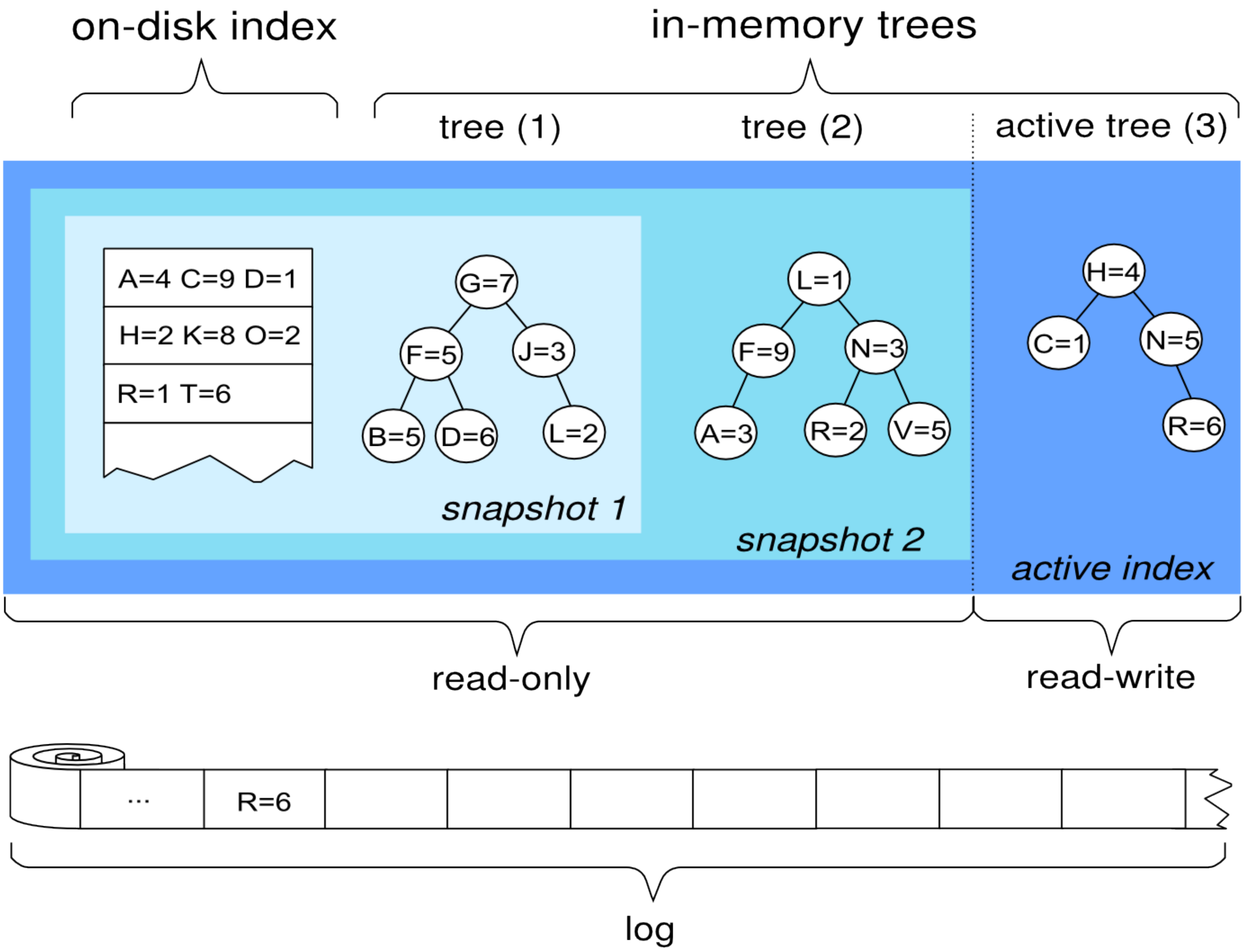


insert(R, 6)

Example: Insertions

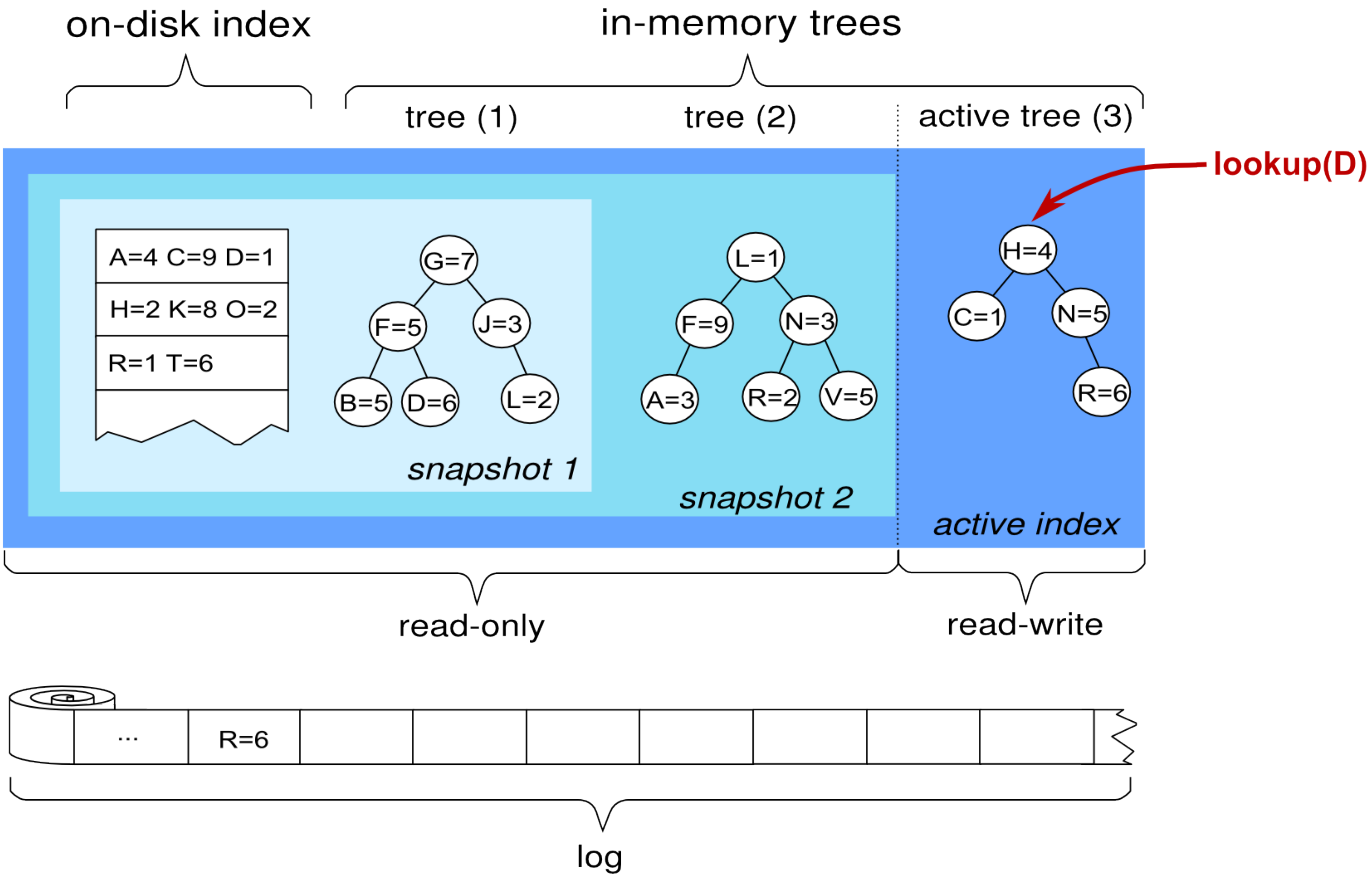


Example: Lookups

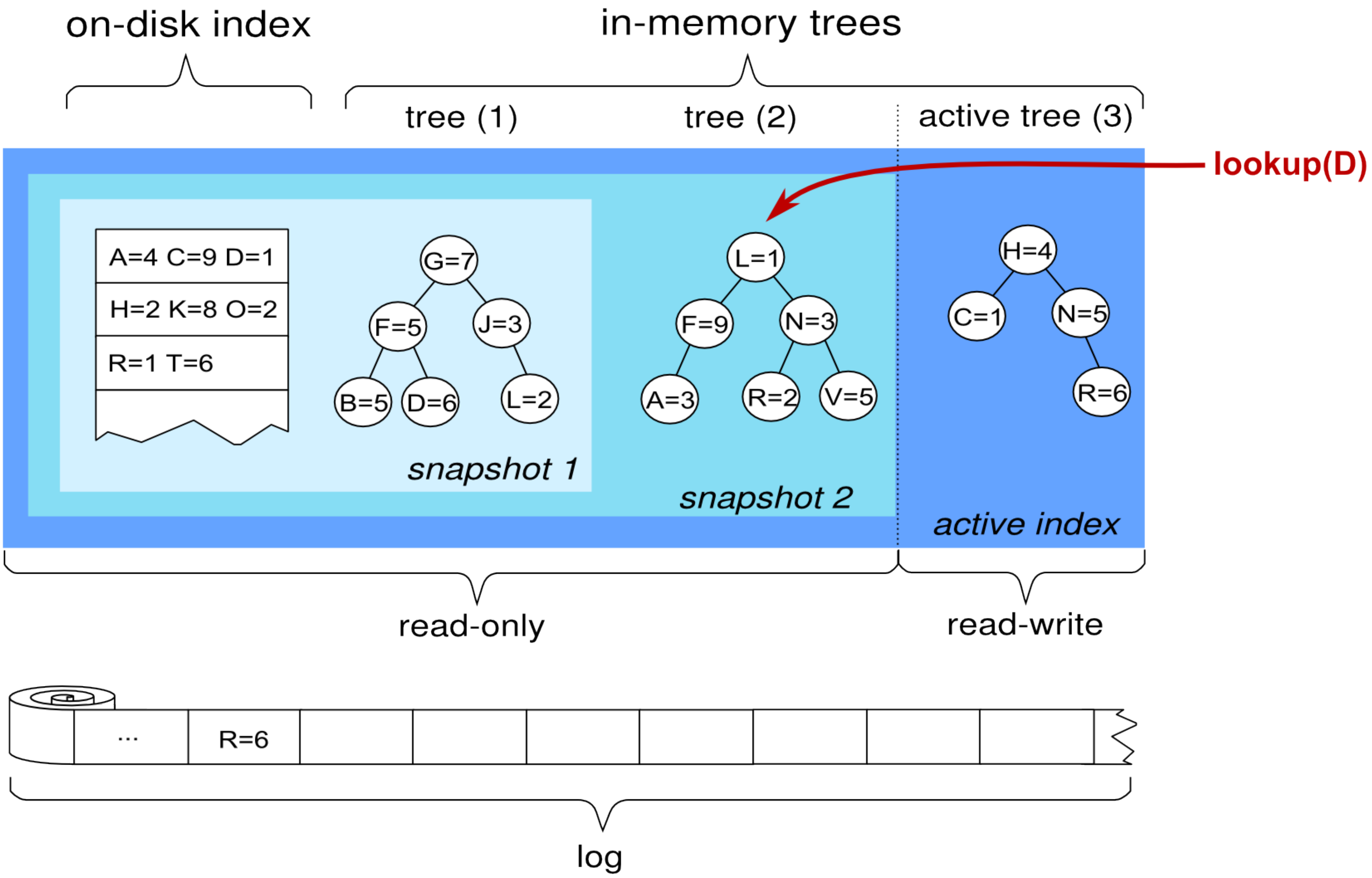


lookup(D)

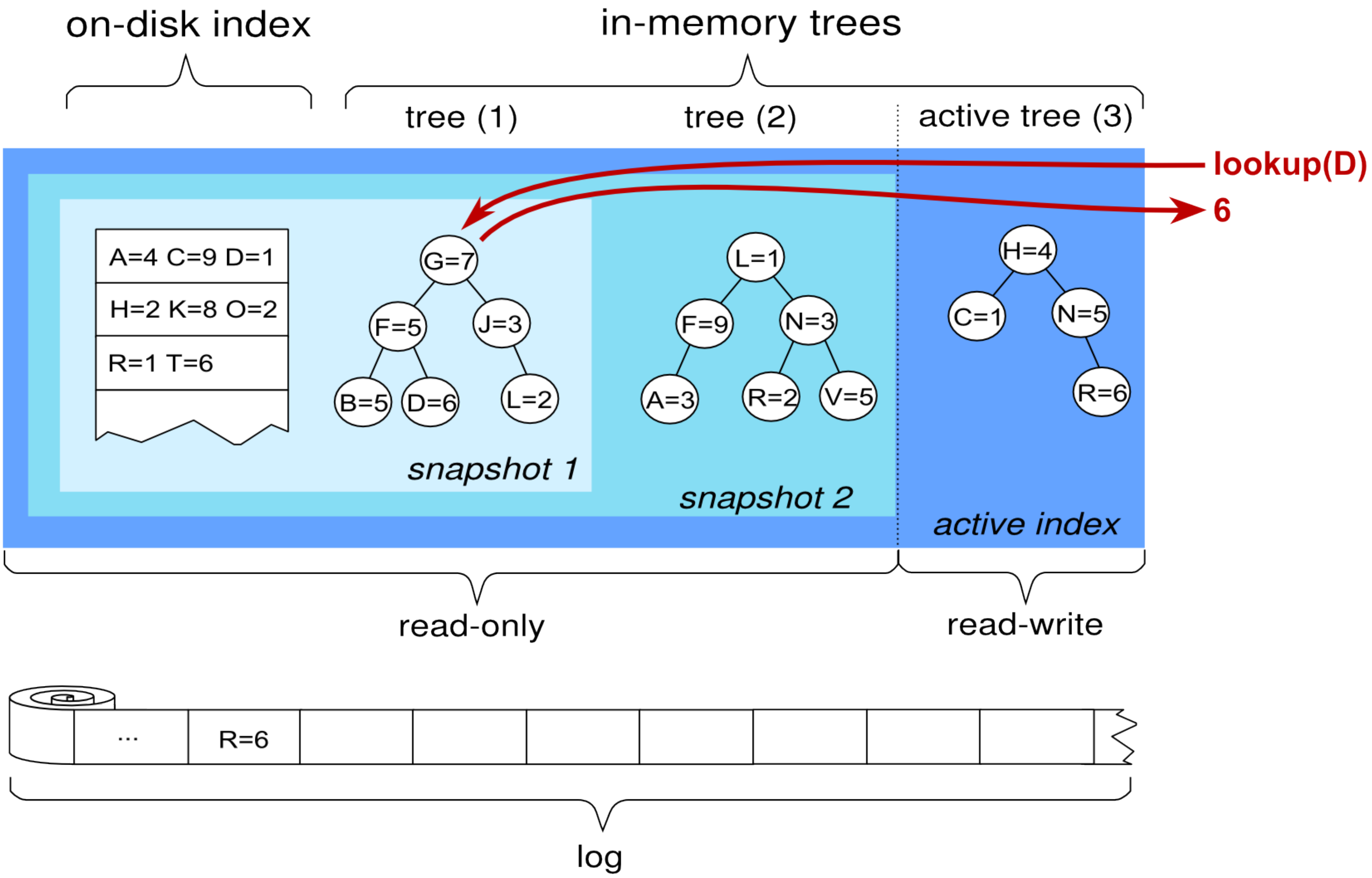
Example: Lookups



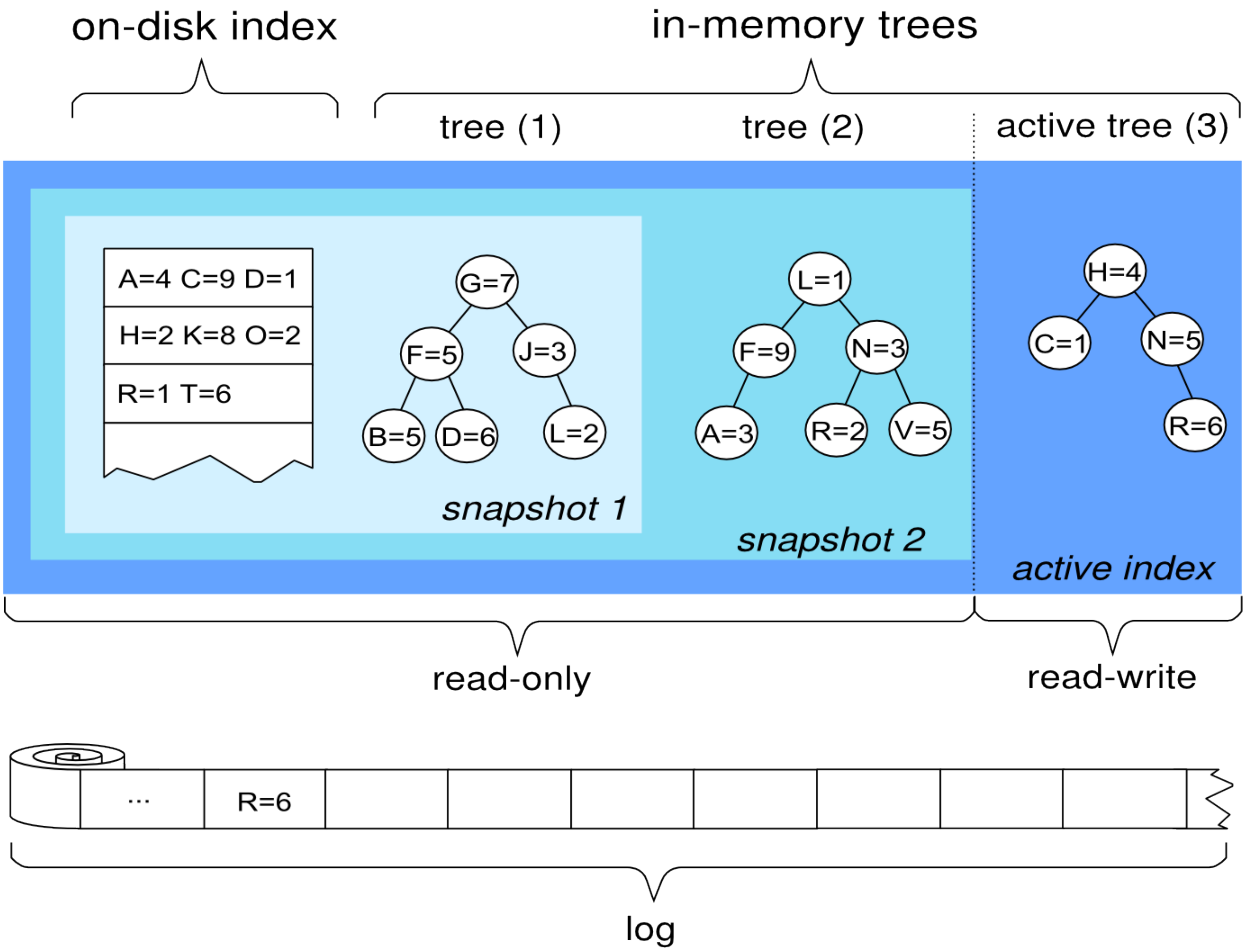
Example: Lookups



Example: Lookups

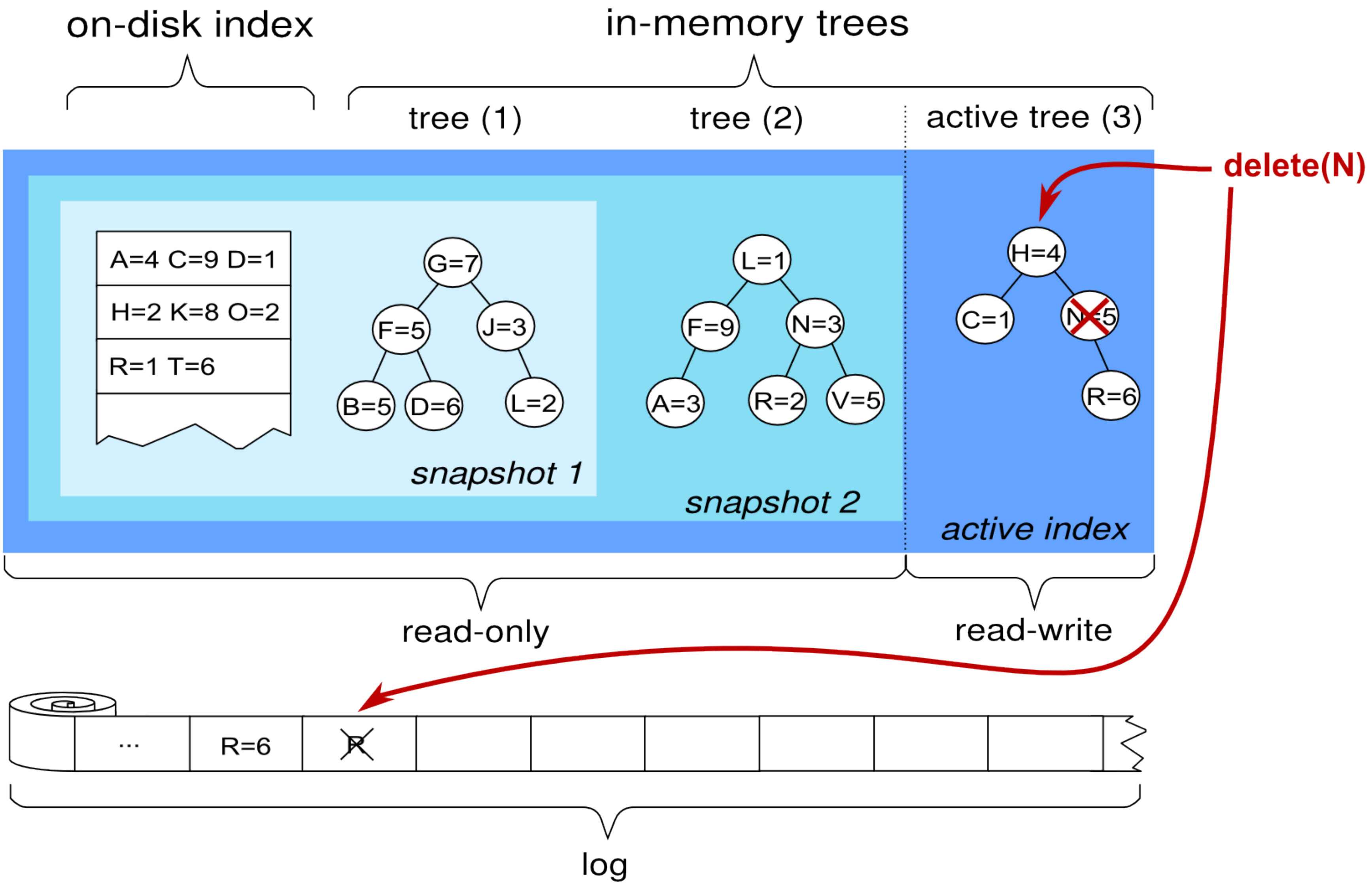


Example: Deletions

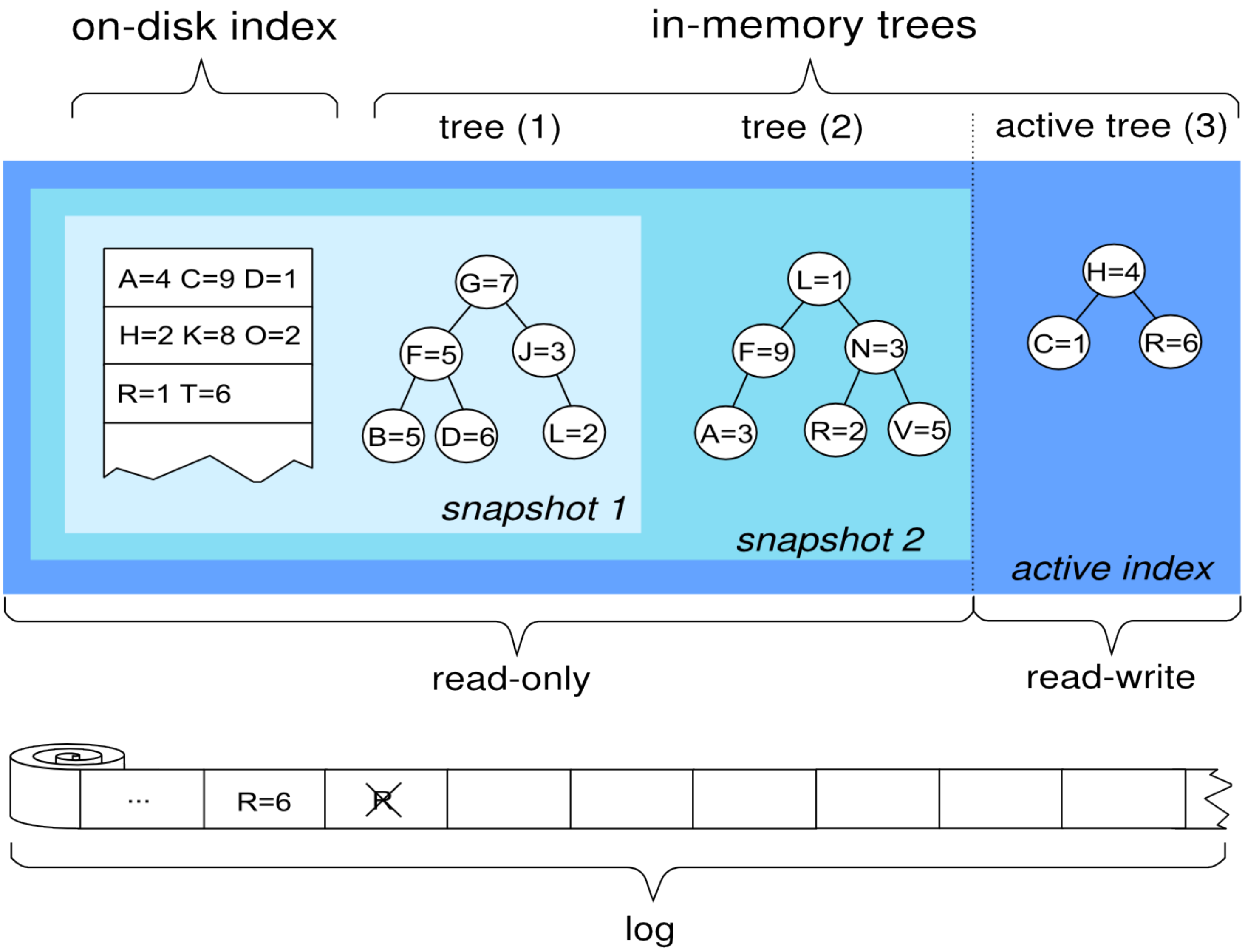


delete(N)

Example: Deletions

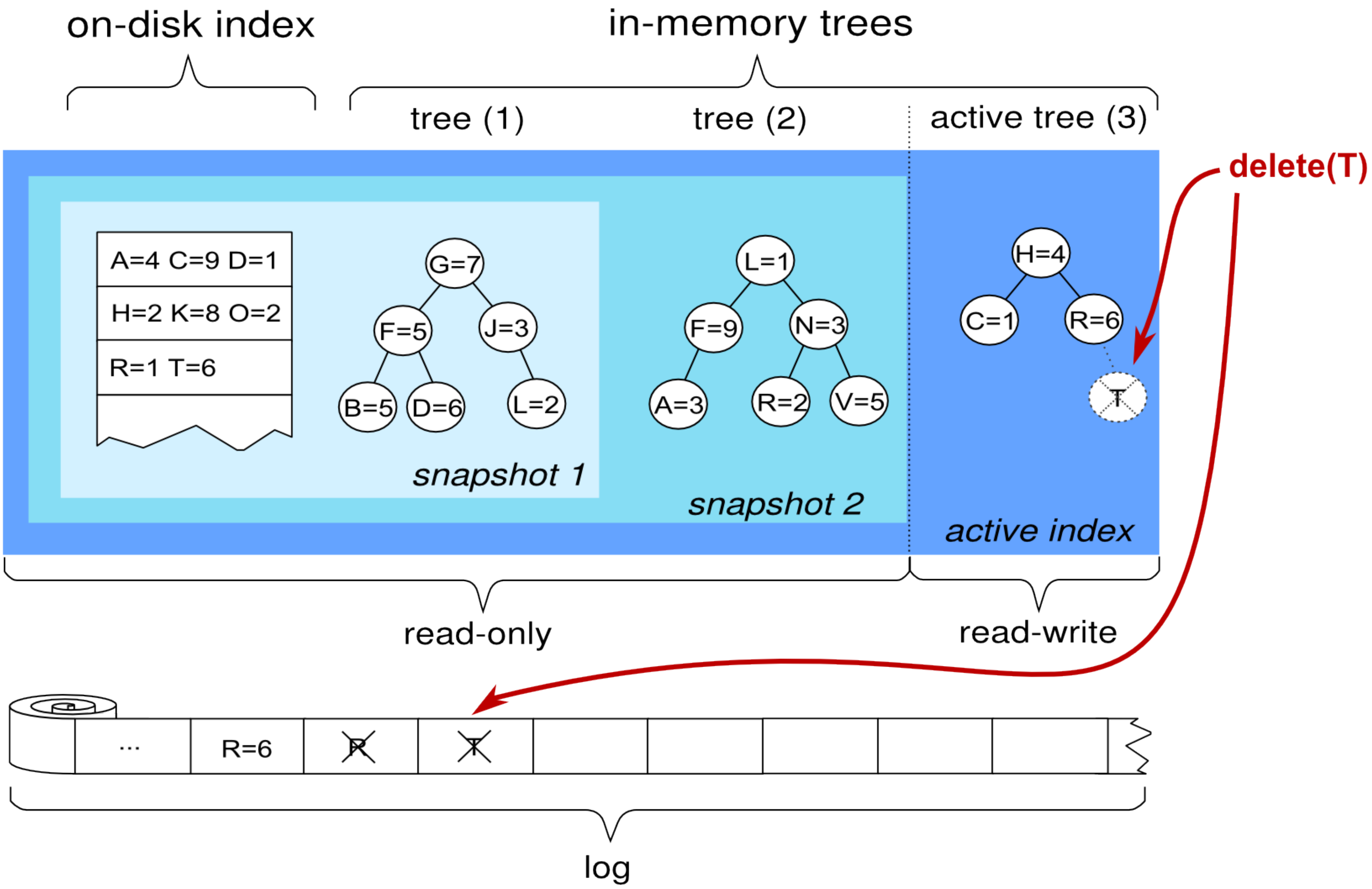


Example: Deletions

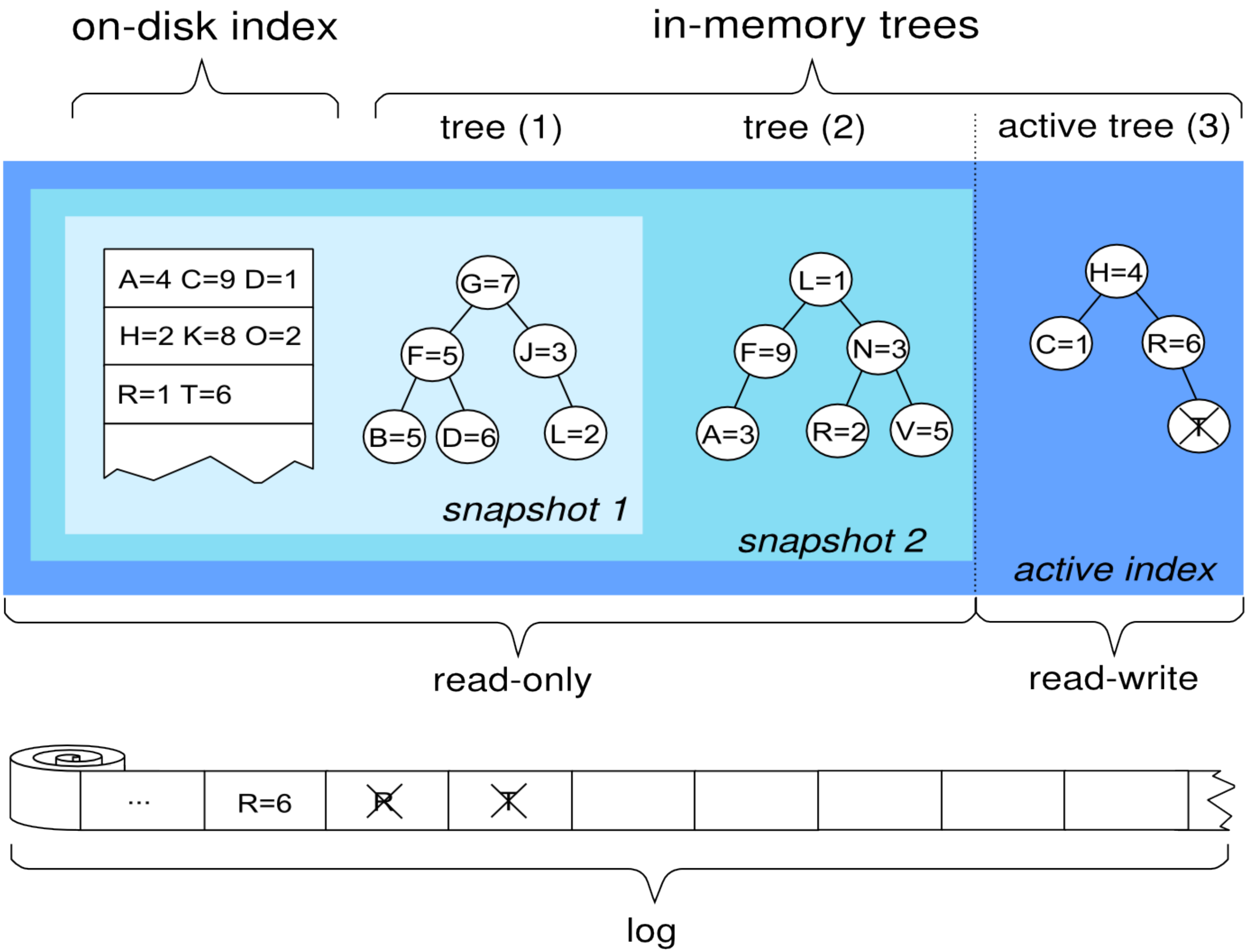


delete(T)

Example: Deletions

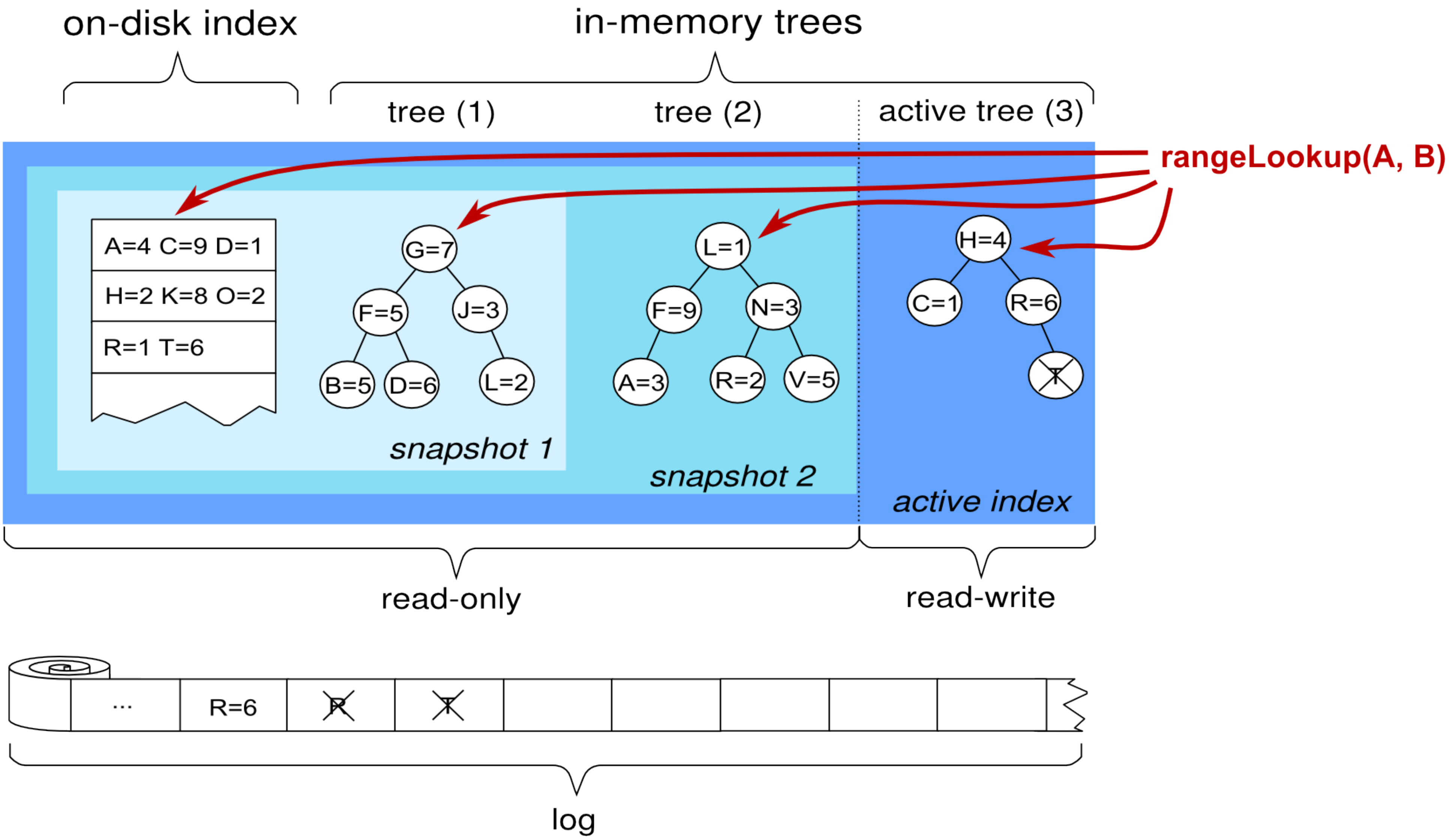


Example: Range Lookups

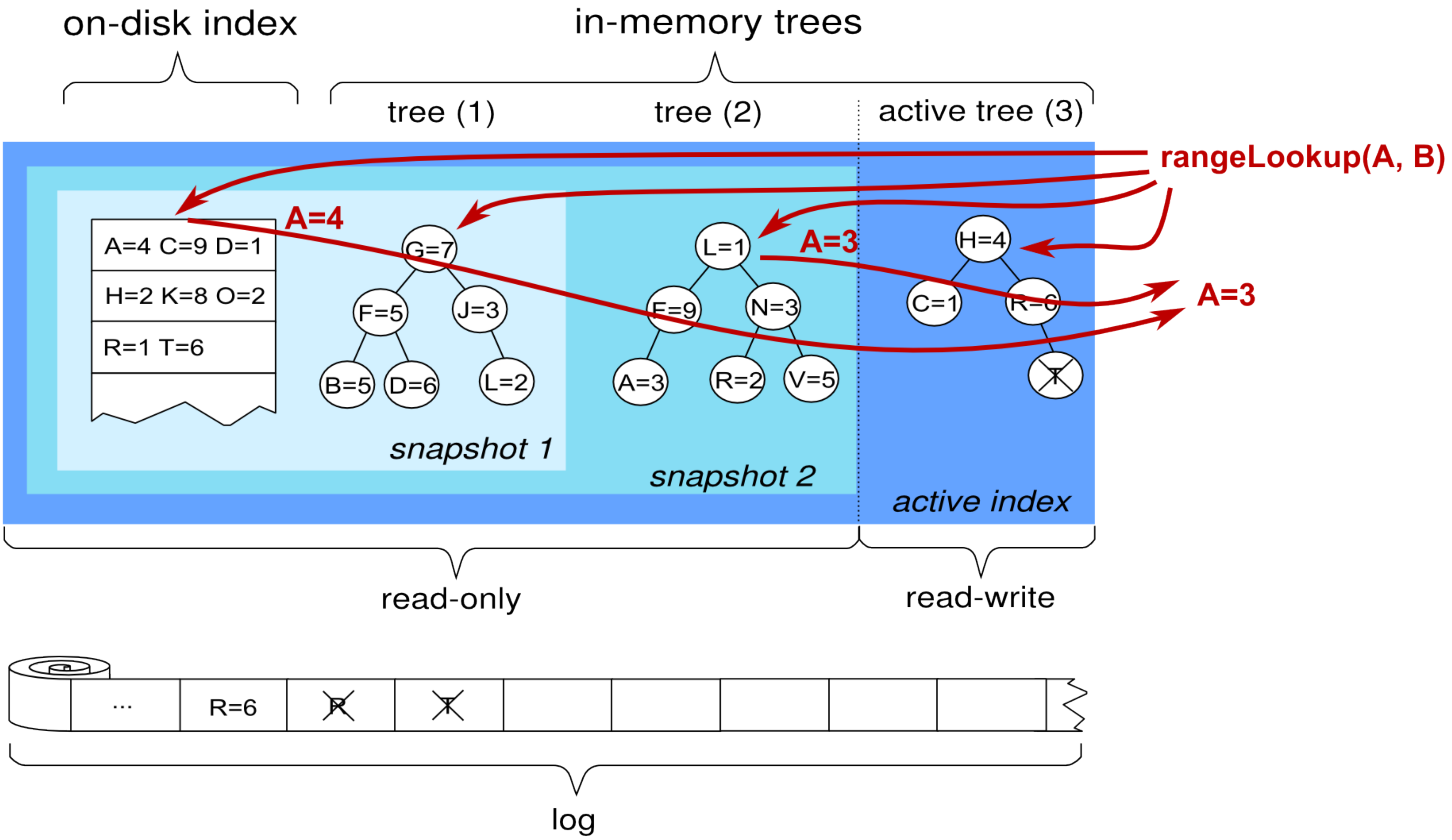


rangeLookup(A, C)

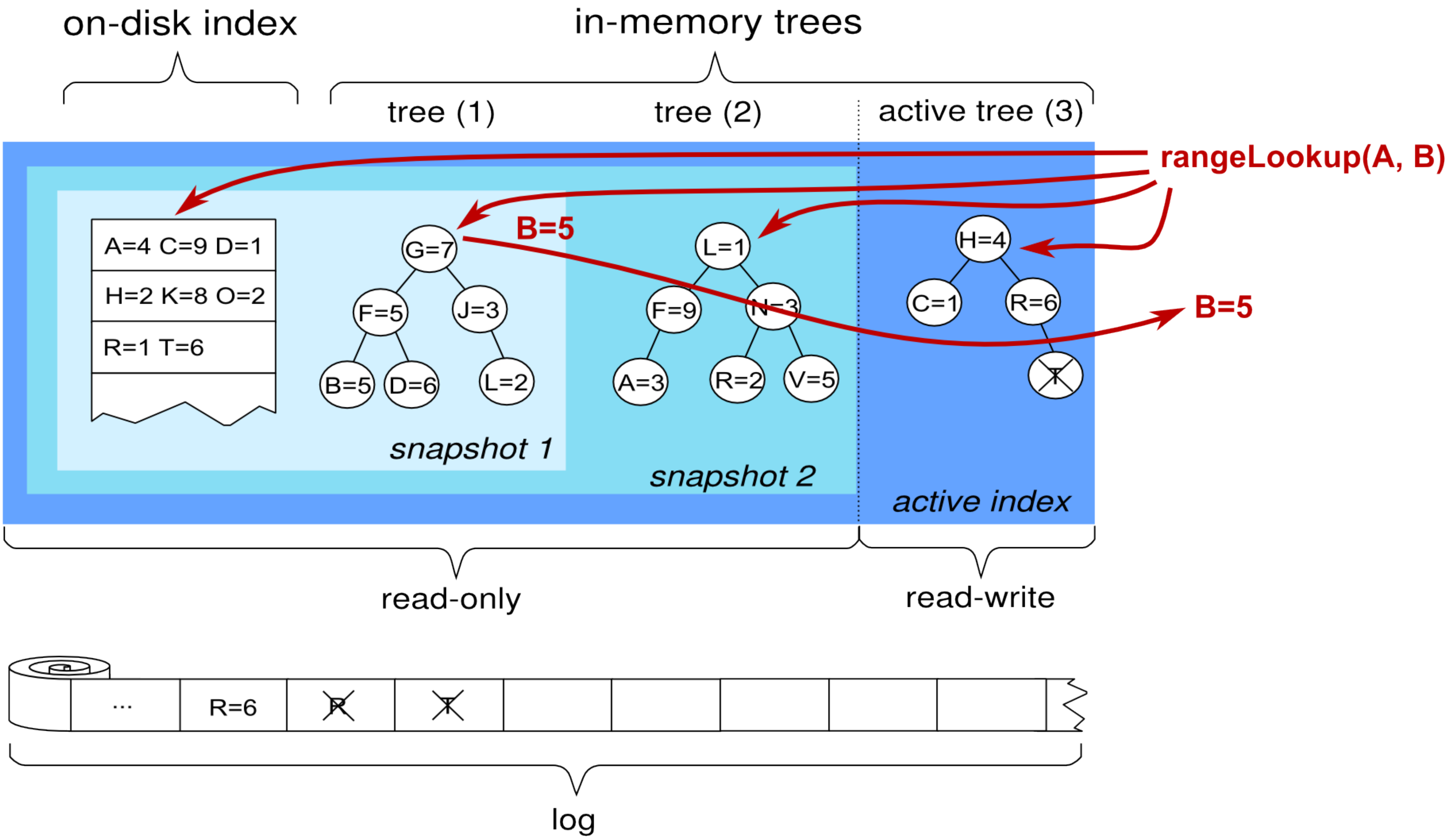
Example: Range Lookups



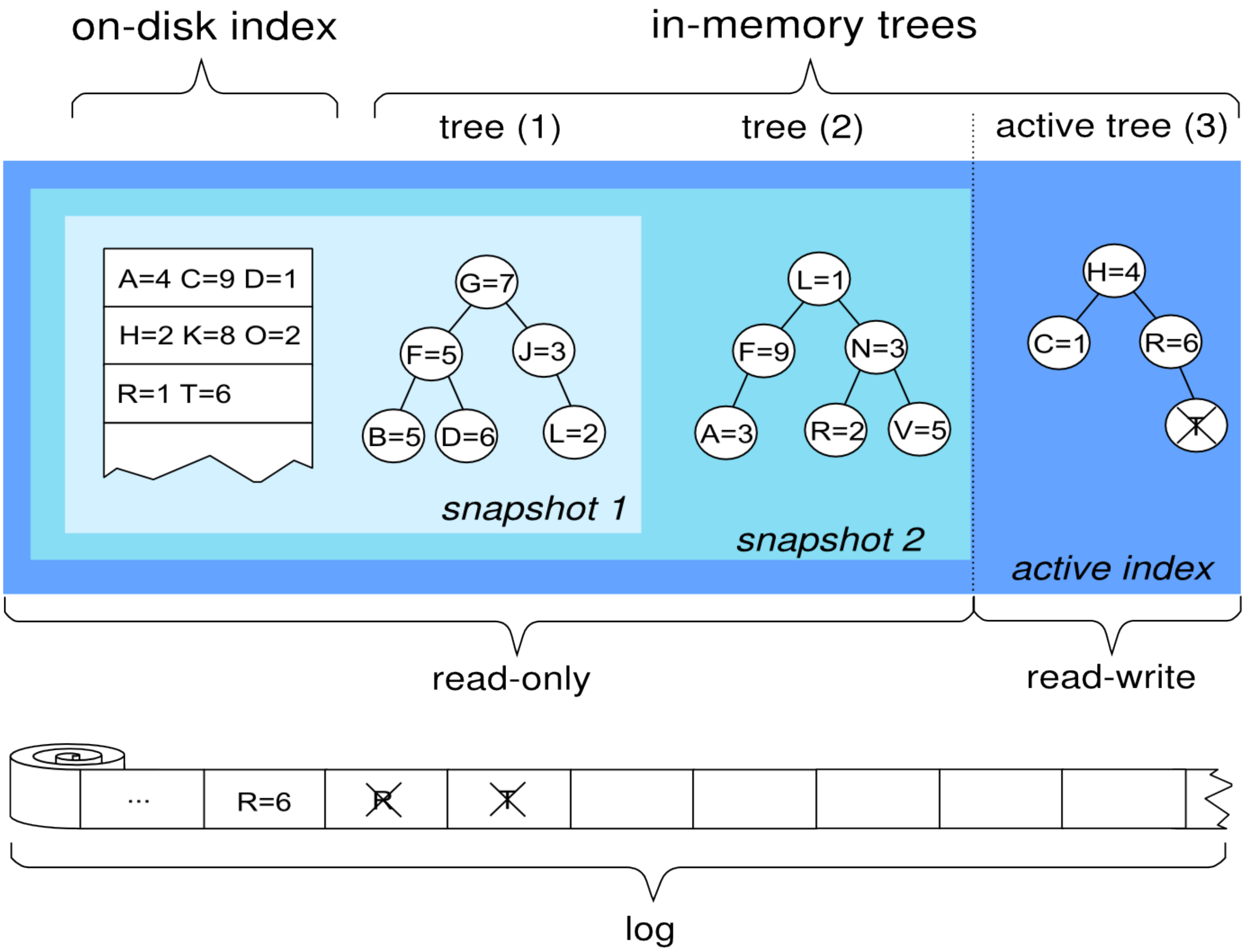
Example: Range Lookups



Example: Range Lookups

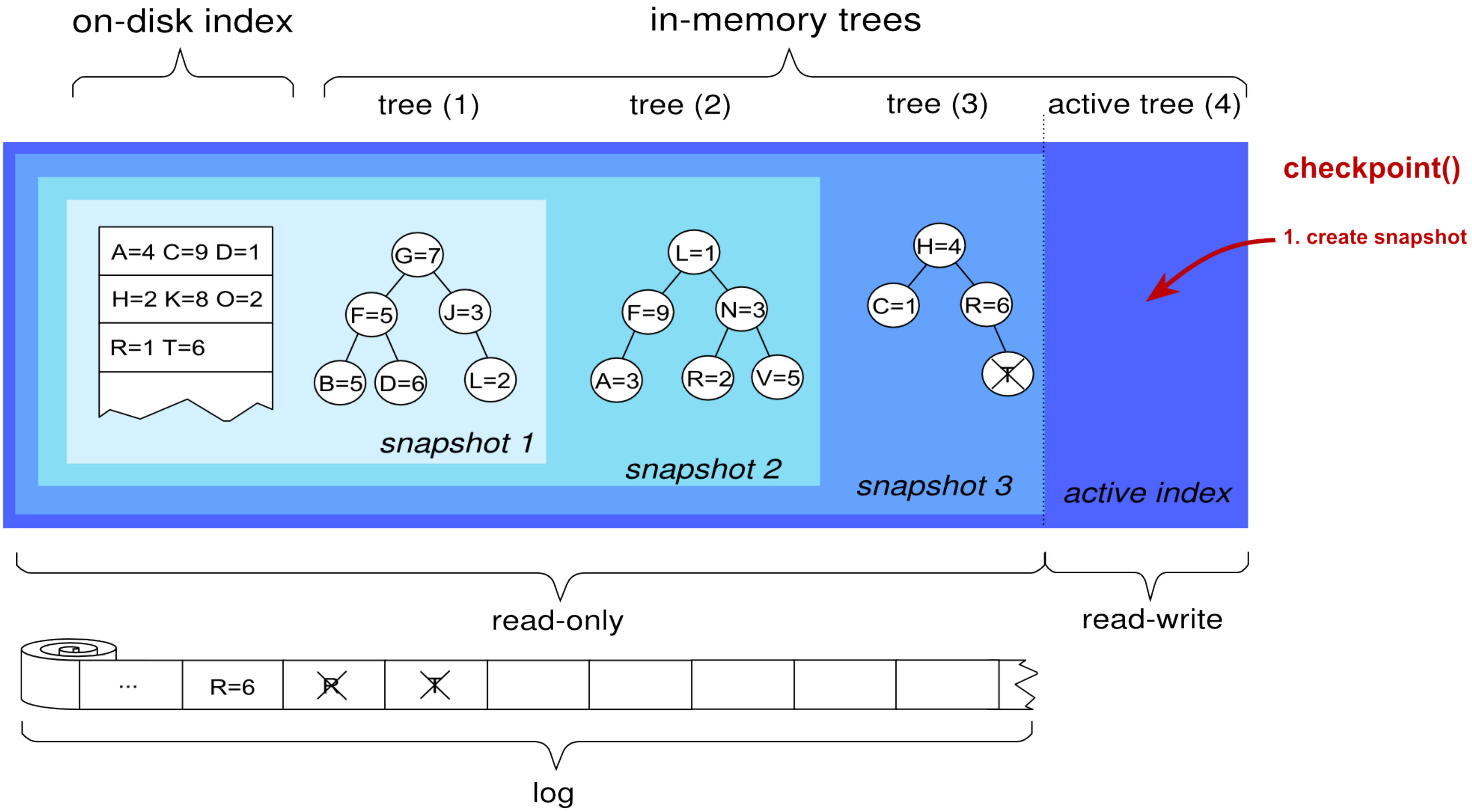


Example: Checkpoints

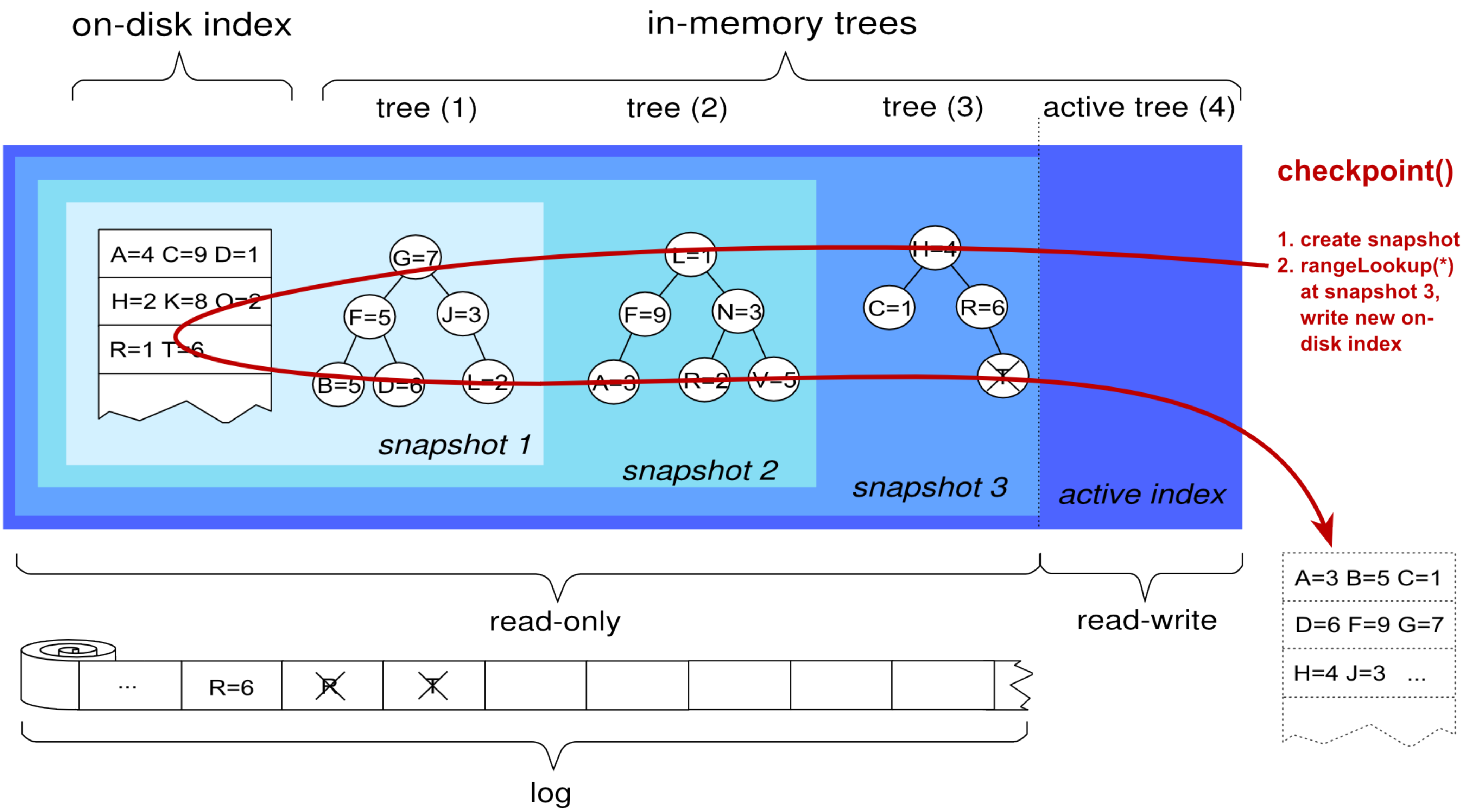


checkpoint()

Example: Checkpoints



Example: Checkpoints



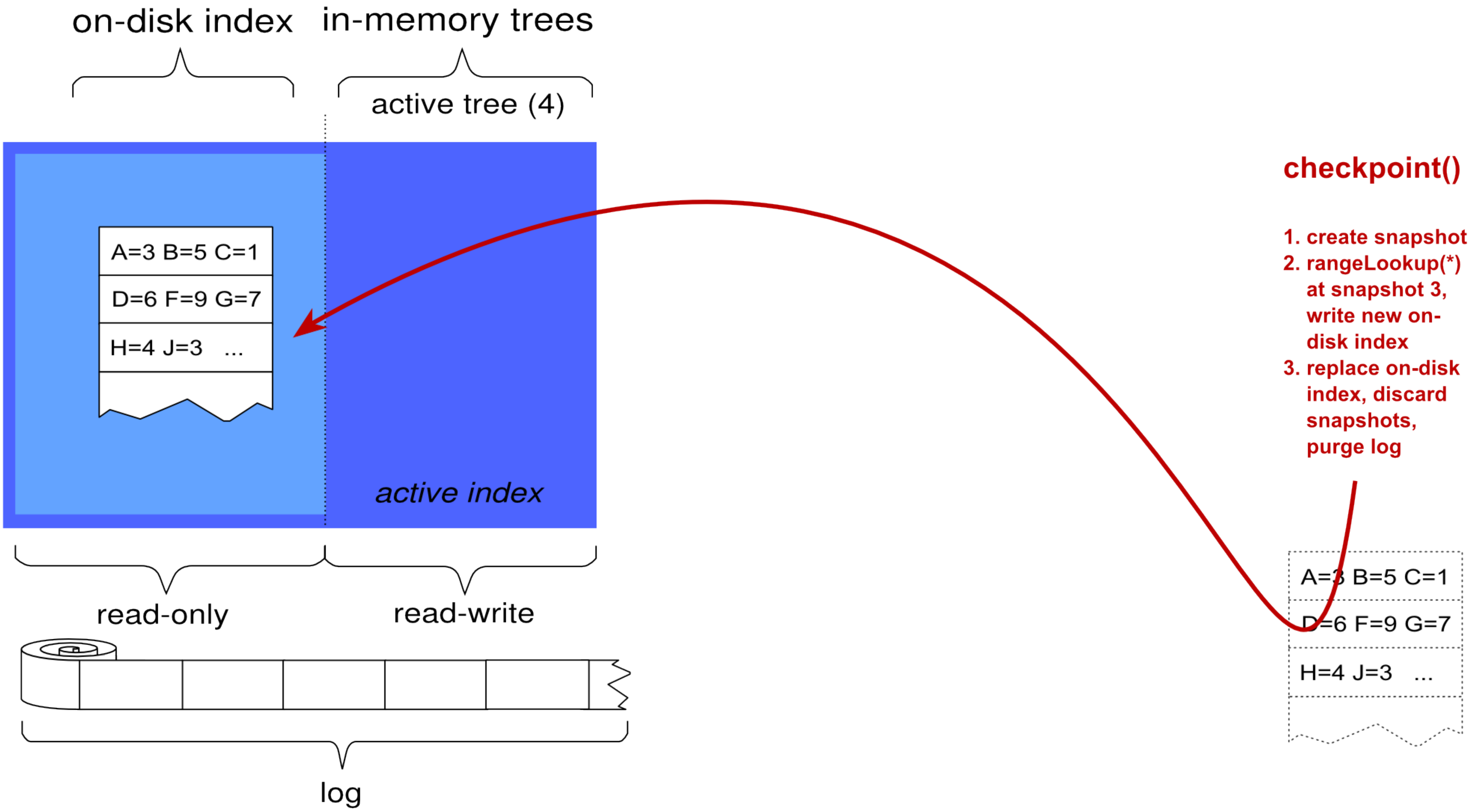
A=4 C=9 D=1
H=2 K=8 O=2
R=1 T=6

checkpoint()

1. **create snapshot**
2. **rangeLookup(*) at snapshot 3, write new on-disk index**

A=3 B=5 C=1
D=6 F=9 G=7
H=4 J=3 ...

Example: Checkpoints



on-disk index

in-memory trees

active tree (4)

A=3 B=5 C=1

D=6 F=9 G=7

H=4 J=3 ...

active index

read-only

read-write

log

checkpoint()

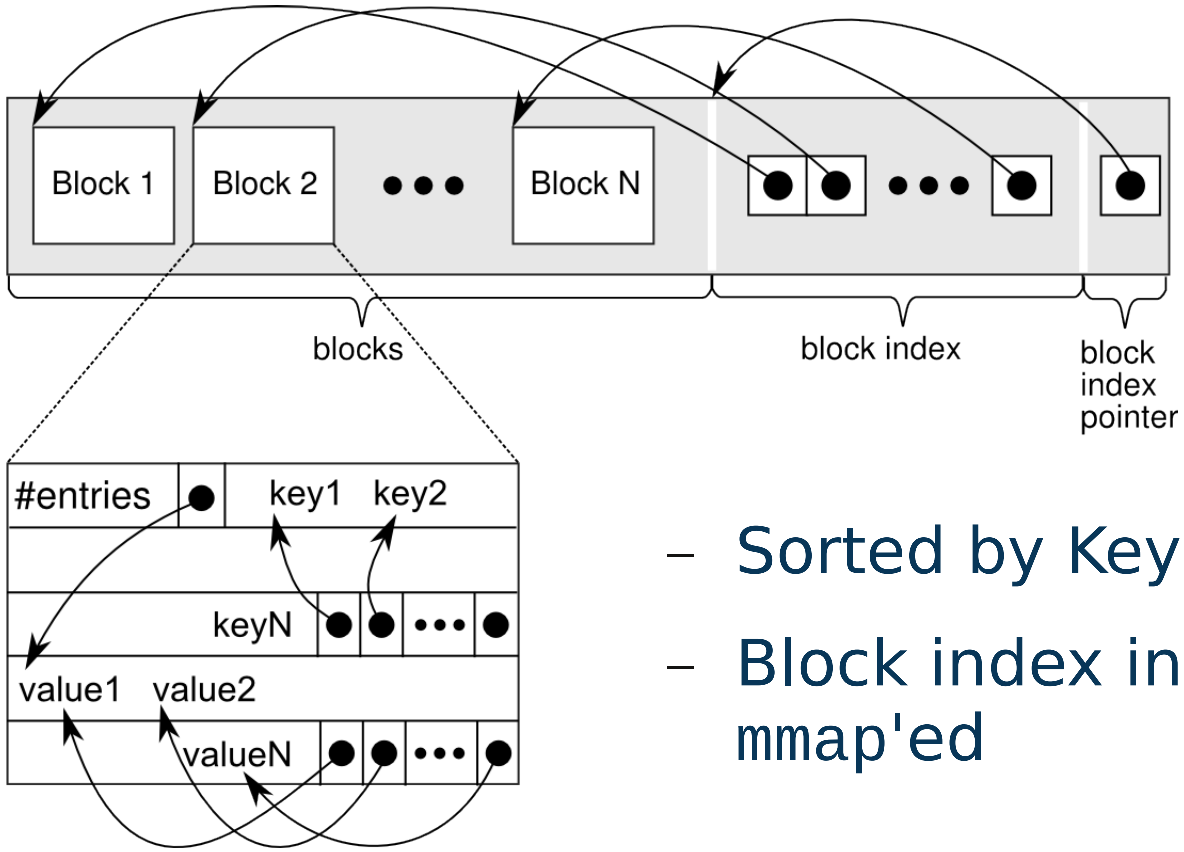
1. **create snapshot**
2. **rangeLookup(*) at snapshot 3, write new on-disk index**
3. **replace on-disk index, discard snapshots, purge log**

A=3 B=5 C=1

D=6 F=9 G=7

H=4 J=3 ...

On-disk Index



- Sorted by Keys
- Block index in RAM, blocks `mmap'ed`

BabuDB: Related Work

- Inspired by log-structured merge trees (LSM-trees)
  - Only one on-disk index
  - No „rolling merge“
- Made popular by Google Bigtable
  - Insert/lookup/merge similar as in Bigtable's Tablets

BabuDB: Metadata Mapping

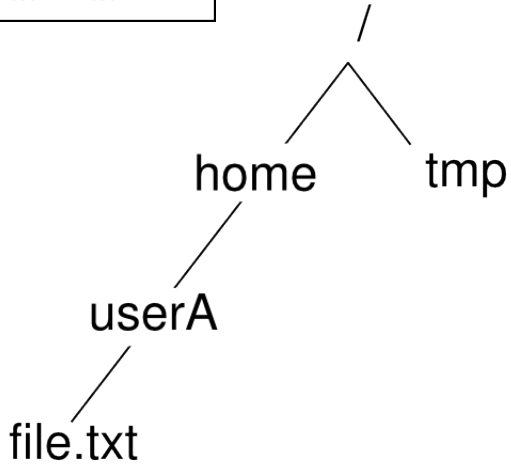
– Mapping a hierarchical directory tree to a flat database index:

key:	parentID	filename	type

value (type 1):	atime	ctime	mtime	file size

value (type 2):	ID	content locations	perm.	owner	link count	...

value (type 3):	xattr1.. xattrN	acl1.. aclN



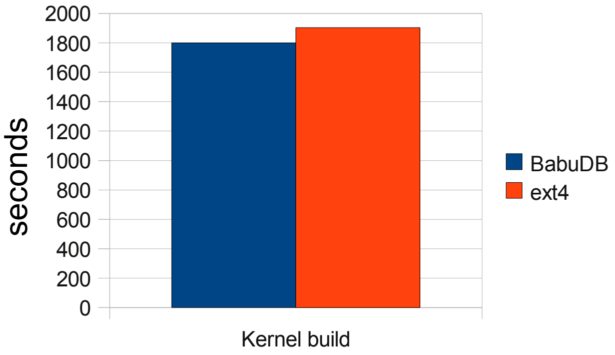
key	value
0,/,1	atime=2009-01-01 12:00 CET...
0,/,2	ID=1,perm=rwxr-x---...
0,/,3	empty
1,home,1	atime=2009-01-01 12:00 CET...
1,home,2	ID=2,perm=rwxr-x---...
1,home,3	*empty*
1,tmp,1	atime=2008-10-21 05:21 CET...
1,tmp,2	ID=3,perm=rwxrwx---...
1,tmp,3	*empty*
2,userA,1	atime=2009-01-01 12:00 CET...
2,userA,2	ID=4,perm=rwx------...
2,userA,3	*empty*
4,file.txt,1	atime=2008-10-05 23:49 CET...
4,file.txt,2	ID=5,perm=rwx------...
4,file.txt,3	*empty*

BabuDB: Advantages

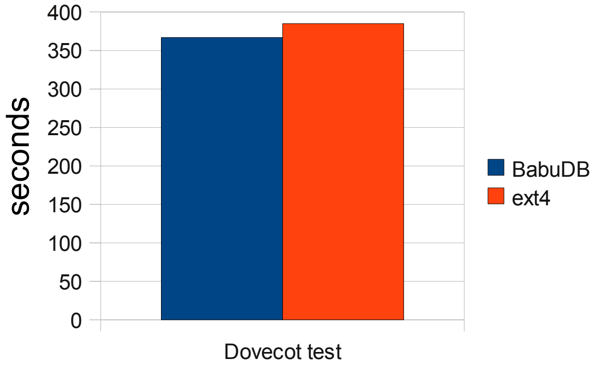
- Why BabuDB for File System Metadata?
  - Short-lived files
    - 50% of all files deleted within 5 minutes
  - Atomic file system operations w/o locking or transactions
    - e.g. `rename`
  - Directory content in contiguous disk regions
    - Efficient `readdir` + `stat`
  - Snapshots
    - No need for multi-version data structures

BabuDB: Evaluation

- Linux kernel build
  - ~10M calls: 44% `stat`, 40% `open`, 15% `readlink`, 1% others

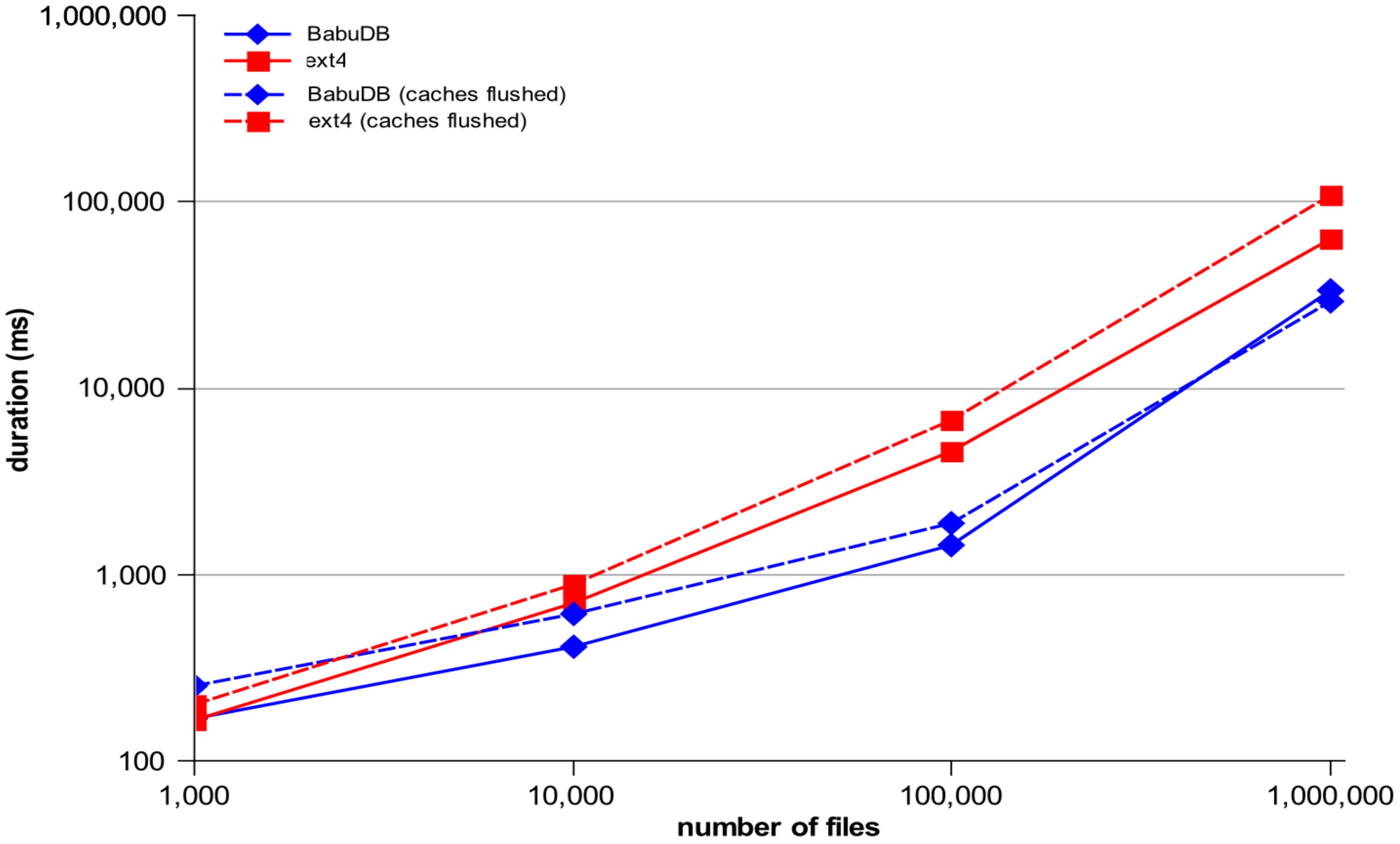


- Dovecot mail server + **`imaptest`**
  - ~2M calls: 51% `stat`, 48% `open`, 1% others



BabuDB: Evaluation

– Listing directory content



Summary

- BabuDB is ...
  - an efficient key-value store
  - optimized for file system metadata but also suitable for other purposes
  - suitable for large-scale databases
  - available for Java and C++ under BSD license
  - used in the XtreemFS metadata server

BABUDB

http://babudb.googlecode.com

XTREEMFS

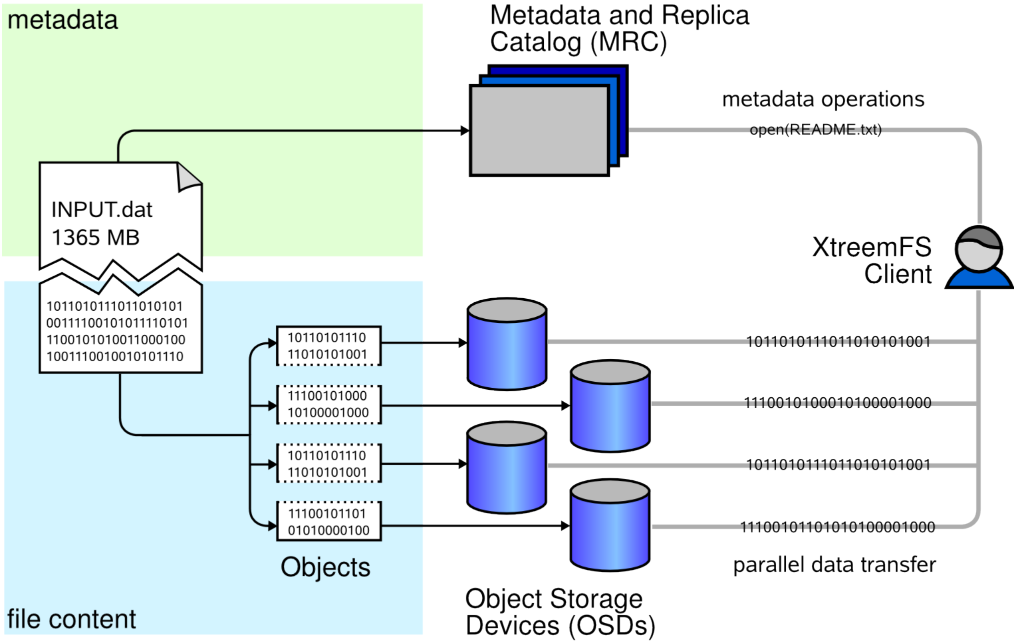
http://www.xtreemfs.org

Thank you for your attention!

Background: XtreemFS

- **XtreemFS: a distributed replicated Internet file system**
  - part of the XtreemOS research project
  - developed since 2006 by partners from Germany, Spain and Italy

XtreemOS
Enabling Linux for the Grid



- **Object-based architecture:**
  - **MRC** stores metadata
  - **OSD**s store pure file content as objects
  - **Client**s provide POSIX file system interface

www.xtreemfs.org

The XtreemOS Project

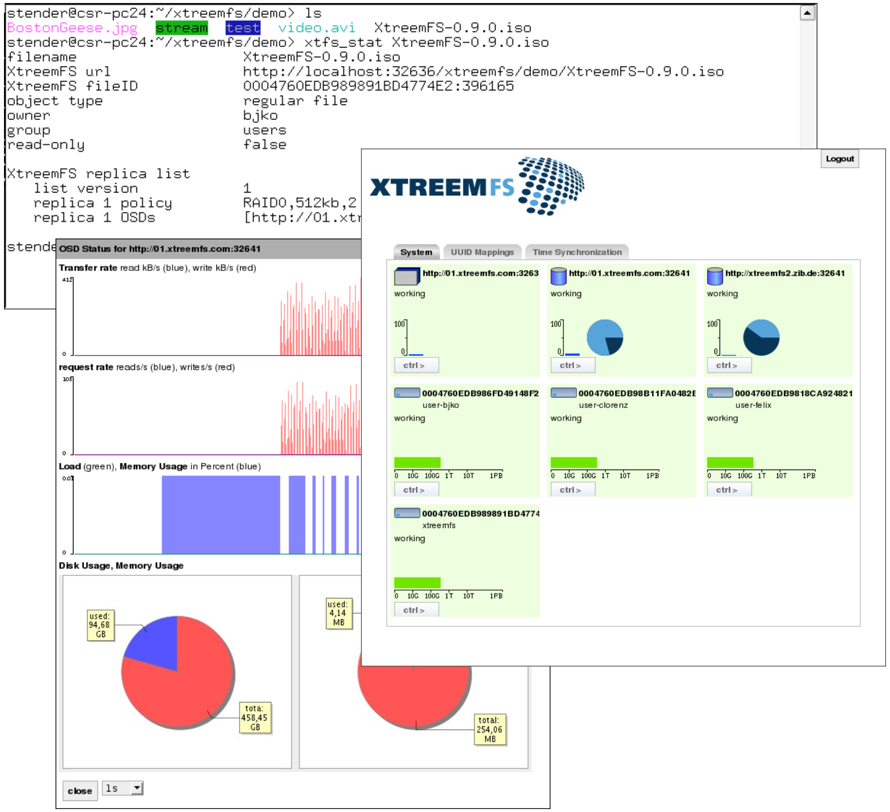
- Research project funded by the European Commission
- 19 partners from Europe and China
- XtreemFS is the data management component
  - developed by ZIB, NEC HPC Europe, Barcelona Supercomputing Center and ICAR-CNR Italy
  - ~ 3 years of development
  - first public release in August 2008

XtreemOS

Enabling Linux for the Grid

XtreemFS: Overview

- What is XtreemFS?
  - a **distributed** and **replicated** **POSIX** compliant file system
  - **off-the-shelve** Servers – no expensive hardware
  - servers in **Java**, runs on Linux / OS X / Solaris
  - client in **C**, runs on Linux / OS X / Windows
  - **secure** (X.509 and SSL)
  - **easy** to install and maintain
  - open source (GPL)

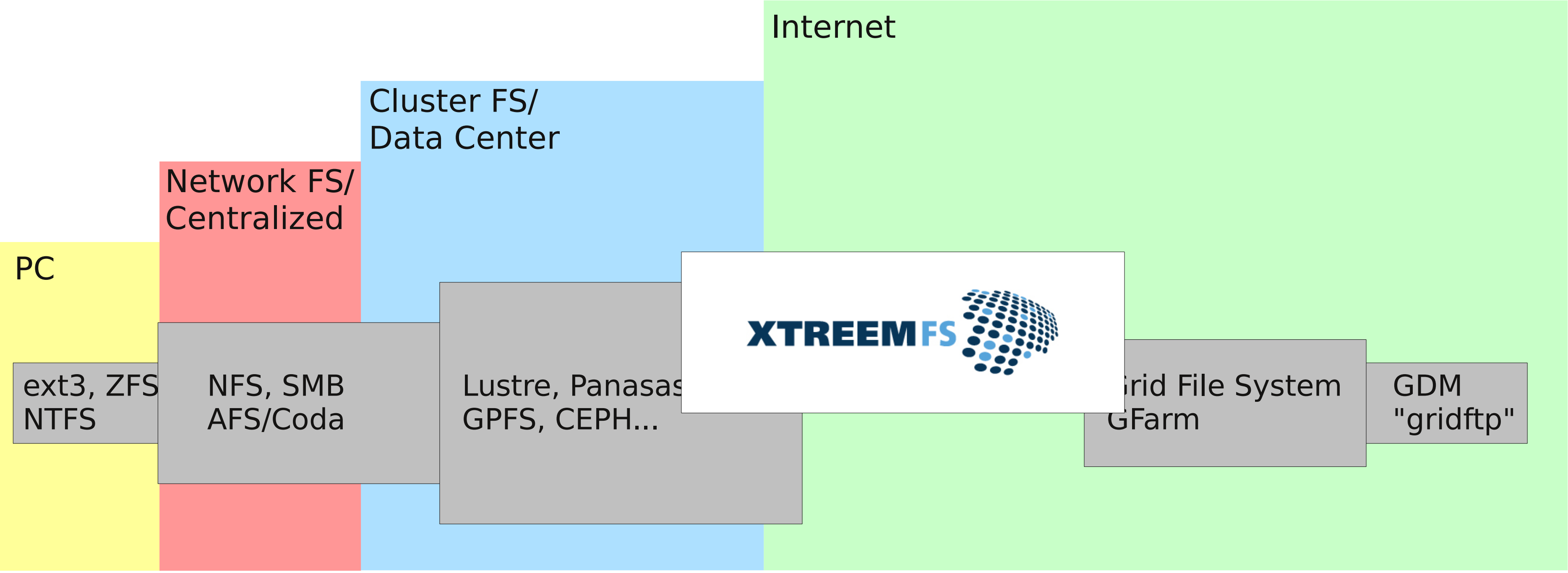


```
stender@csr-pc24:~/xtreemfs/demo> ls
BostonGeese.jpg  stream  test  video.avi  XtreemFS-0.9.0.iso
stender@csr-pc24:~/xtreemfs/demo> xtfs_stat XtreemFS-0.9.0.iso
filename                    XtreemFS-0.9.0.iso
XtreemFS url                http://localhost:32636/xtreemfs/demo/XtreemFS-0.9.0.iso
XtreemFS fileID             0004760EDB989891BD4774E2:396165
object type                 regular file
owner                       bjko
group                       users
read-only                   false

XtreemFS replica list
   list version             1
   replica 1 policy         RAID0,512kb,2
   replica 1 OSDs           [http://01.xtr

stende
```

File System Landscape



Internet

Cluster FS/
Data Center

Network FS/
Centralized

PC

ext3, ZFS
NTFS

NFS, SMB
AFS/Coda

Lustre, Panasas
GPFS, CEPH...

XTREEMFS

Grid File System
GFarm

GDM
"gridftp"