Bachelorarbeit im Studiengang Medieninformatik

Efficient Synchronization of Linux Memory Regions over a Network (Presentation Notes)

vorgelegt von **Felicitas Pojtinger**

an der **Hochschule der Medien Stuttgart**

am **03.08.2023**

zur Erlangung des akademischen Grades eines **Bachelor of Science**

Erstprüfer: **Prof. Dr. Martin Goik**

Zweitprüfer: **M.Sc. Philip Betzler**

Bachelor's Thesis

Efficient Synchronization of Linux Memory Regions over a Network (Presentation Notes)

Author: **Felicitas Pojtinger**

University: **Hochschule der Medien Stuttgart**

Course of Study: **Media Informatics**

Date: **2023-08-29**

Academic Degree: **Bachelor of Science**

Primary Supervisor: **Prof. Dr. Martin Goik**

Secondary Supervisor: **M.Sc. Philip Betzler**

Contents

List of Figures

List of Acronyms

API	Application Programming Interface
I/O	Input/Output
OS	Operating System
CPU	Central Processing Unit
RAM	Random Access Memory
SSD	Solid State Drive
HDD	Hard Disk Drive
CXL	Compute Express Link
VFS	Virtual File System
UUID	Universally Unique Identifier
CRC32	Cyclic Redundancy Check 32-Bit
LRU	Least Recently Used
WAN	Wide Area Network
LAN	Local Area Network
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
P2P	Peer-To-Peer
NATs	Network Address Translators
IPC	Inter-Process Communication
RTT	Round-Trip Time
SRP	SCSI RDMA Protocol
GNU	GNU's Not Unix
UNIX	UNIX Family of Operating Systems
macOS	Apple Macintosh Operating System
FreeBSD	Free Berkeley Software Distribution
NBD	Network Block Device
S3fs	S3 File System

NVMe Non-Volatile Memory Express

LTFS Linear Tape File System

LTO Linear Tape-Open

EXT4 Fourth Extended Filesystem

Btrfs B-Tree File System

LTFS Linear Tape File System

ELF Executable and Linkable Format

C C Programming Language

Rust Rust Programming Language

Go Go Programming Language

C++ C++ Programming Language

ARM ARM RISC Computer Processor Architecture

x86 x86 CISC Computer Processor Architecture

RISC-V RISC-V RISC Computer Processor Architecture

LPDDR5 Low-Power Double Data Rate 5

HTTP Hypertext Transfer Protocol

HTTPS HTTP Secure

HTTP/2 HTTP Version 2

QUIC Quick UDP Internet Connections

WebRTC Web Real-Time Communication

Wasm WebAssembly

WASI WebAssembly System Interface

IETF Internet Engineering Task Force

OIDC OpenID Connect

AWS Amazon Web Services

CNCF Cloud Native Computing Foundation

S3 Simple Storage Service

TLS Transport Layer Security

mTLS	Mutual TLS
SSH	Secure Shell
DoS	Denial of Service
JSON	JavaScript Object Notation
JSONL	JSON Lines
SQL	Structured Query Language
NoSQL	Not Only SQL
Protobuf	Protocol Buffers
IDL	Interface Definition Language
DSL	Domain-Specific Language
KV	Key-Value
Syscalls	System Calls
VM	Virtual Machine
RPC	Remote Procedure Call
REST	Representational State Transfer
FUSE	File Systems in Userspace

- Introduction
  - Title slide
  - ToC
  - About me
  - Abstract/introduction
- Methods
  - Pull-based synchronization with `userfaultfd`/Userfaults in Go with `userfaultfd`
    - Technology section: Memory organization & hierarchy
    - Technology section: Page faults
  - Push-based synchronization with `mmap` and hashing/file-based synchronization, discussion
    - Technology section: ‘mmap“
    - Technology section: Delta synchronization

  - Push-based synchronization with FUSE/FUSE implementation in Go, discussion
    - Technology section: FUSE
  - Mounts with NBD/NBD with go-nbd
    - Technology section: NBD
  - Push-Pull Synchronization with Mounts/managed mounts with r3map
    - Technology section: RTT, LAN and WAN
  - Pull-Based Synchronization with Migrations/Live migration
    - Technology section: Pre- and post-copy VM migration, workload analysis
- Optimizations
  - Pluggable Encryption, Authentication and Transport
  - Concurrent Backends
  - Remote Stores as Backends
  - Concurrrent RPC frameworks (dudirekta) and connection pooling (gRPC)
- Discussion and Results
  - Testing Environment
  - Access methods (userfaults vs. direct vs. managed mounts): Latency & Throughput, discussion
  - Initialization: Polling vs. udev
  - Chunking methods: Local vs. remote
  - RPC frameworks; discussion
  - Backends: Latency & throughput; discussion
  - General limitations of the r3map library (deadlocks etc.)
- Implemented Use Cases
  - Using mounts for remote swap with `ram-dl`
  - Mapping tape into memory with tapisk
- Future Use Cases
  - Improving cloud storage clients
  - Universal database, media and asset streaming
  - Universal app state mounts and migrations
- Conclusion
- Thanks