Research Summary

3. Februar 2026

Lehrstuhl Informatik 4 - System Software

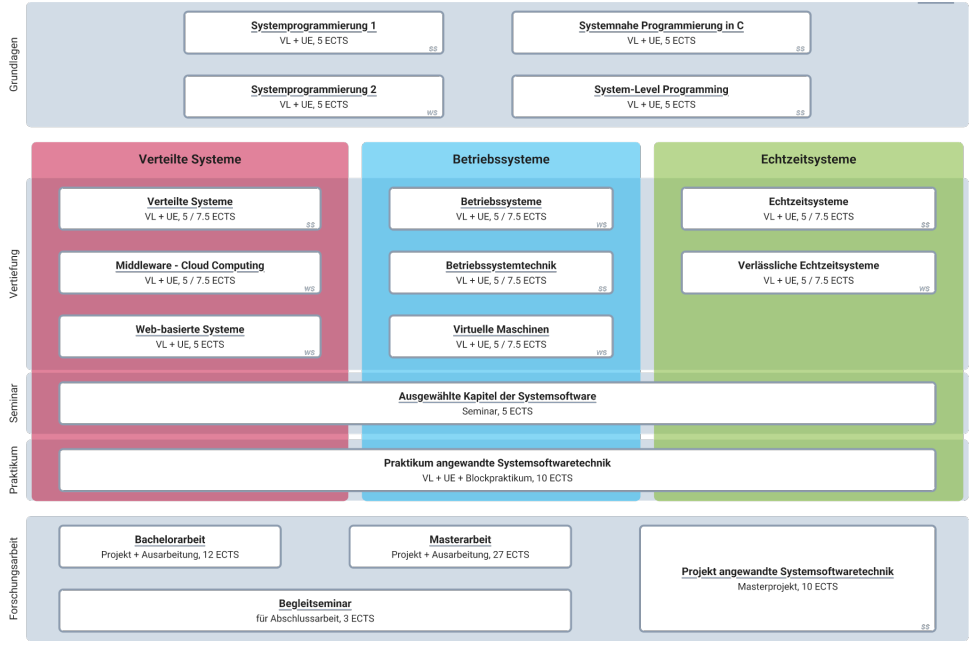
Friedrich-Alexander-Universität Erlangen-Nürnberg

Friedrich-Alexander-Universität
Technische Fakultät

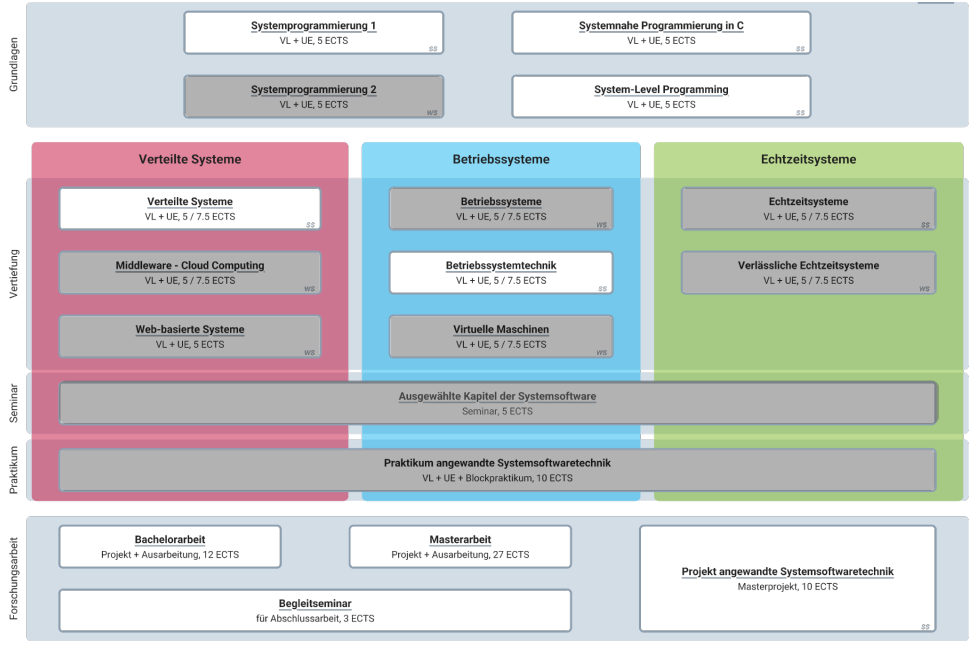
Lehrstuhl für Informatik 4
Systemsoftware

Lehre

Lehrangebot



Lehrangebot *im kommenden Sommersemester*



Neue Veranstaltung: Betriebssystemsicherheit

- **Inhalte**
  - Sicherheitsmechanismen des Betriebssystems
  - Mikrokernel und verwandte Systemarchitekturen
  - Sicherheitsaspekte von Virtualisierungs- und Containermodellen
  - Trusted Computing (z. B. TPM und ARM TrustZone)
  - Confidential Computing (z. B. Intel SGX/TDX und AMD SEV-SNP)
- **Praktischer Anteil**
  - Analyse & Diskussion von aktuellen Forschungsarbeiten
  - Implementierung eines vereinfachten Containersystems
  - Einsatz von Confidential Computing
- **Aufbau & Umfang (5 ECTS)**
  - Vorlesung (Rüdiger Kapitza)
  - Tafel- & Rechnerübung (Ole, Dustin & Maxim)
- **Wann:** Erste Vorlesung 15.4., 12:15–13:45 Uhr, Raum 0.035
- **Mehr Infos bald unter:** https://sys.cs.fau.de/lehre/ss26/bss

Abschlussarbeiten

Directed Acyclic Graph-based BFT Protocols under Fire BA/MA/Projekt

- **Challenges:** *DAG Protocols are used in Distributed Ledger Systems*
  - e.g., in **Sui** and **Aptos**
  - for Atomic Broadcast of transactions
  - **but** their performance and behaviour is not sufficiently studied
- **Research:** *Analysis of DAG protocols in an event-driven simulation*
  - Modelling of **protocol logic**: broadcast layer, commitment rules, ...
  - **Evaluate** protocol behavior under various network conditions

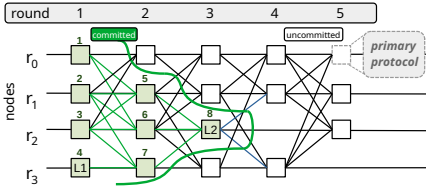


Abbildung 1: Atomic Broadcast: By interpreting the DAG, a deterministic rule can order all blocks after an anchor node is randomly chosen (here: L2).

Interesse?
Für nähere Infos, Mail an:
berger@cs.fau.de

Asynchronous Clients in PBFT & Execution Semantics

BA/MA/Projekt

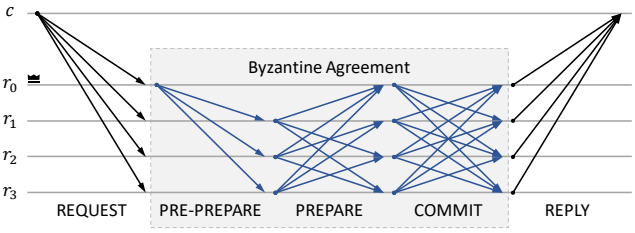


Abbildung 2: PBFT: Requests are totally ordered by a leader.

- **Challenges:** *PBFT is a widely used BFT state-machine replication protocol*
  - we maintain our own research-focused replication library THEMIS at our chair
  - written in **Rust**, it implements PBFT
  - we seek to **explore new features** (originally never described or implemented by PBFT)

- **Research:** *Implement and Evaluate techniques to handle **asynchronous clients** & network partitions*
  - Exactly-once-execution semantics
  - Single client sends multiple requests simultanously without blocking

Interesse?

Für nähere Infos, Mail an:

berger@cs.fau.de

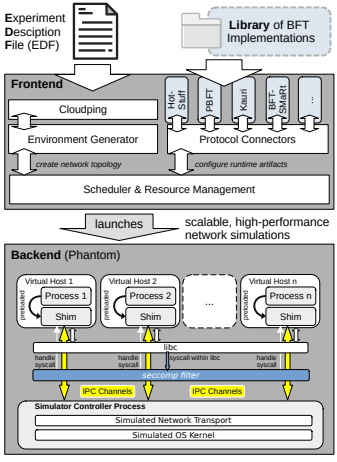
Validation of BFT Implementations through LLMs

BA/MA/Projekt

- **Challenges:** *BFT Protocol Implementations are used in Distributed Ledger Systems*
  - often, **billions of dollars** are at stake
  - **Correctness** of implementation is crucial
  - Can LLMs help **identify bugs** in existing implementations?
- **Research:** *Development of a LLM-based vulnerability-scanning method for BFT implementations*
  - How does **additional context** help? Paper specification, formal specification, Fuzzer outputs or protocol debug logs, ...
  - Can discovered bugs be **patched** out (semi-)automatically?

Interesse?
Für nähere Infos, Mail an:
berger@cs.fau.de
arne.vogel@fau.de

Progressing a Hybrid Simulation-Emulation Frontend for BFT Projekt



Interesse?
Für nähere Infos, Mail an:
berger@cs.fau.de

- **Challenges: *Shadow*** *is a hybrid simulation-emulation framework*
  - it allows to plug-in application binaries into a high-performance network and kernel **simulation engine**
  - We developed a **frontend** for **BFT systems experimentation**, which needs to be maintained and extended by new features
- **Implementation:**
  - Rewrite code to support newest simulation engine
  - Add new features, i.e. **live-monitoring** of system performance during simulations

PBFT Implementation Optimization

- BFT Protocols typically optimize for network
- In LAN: Compute Bottleneck
- Your Task:
  - Setup profiling tools in **Rust**
  - **Identify** bottlenecks in PBFT implementation
  - **Optimize** & Repeat :)

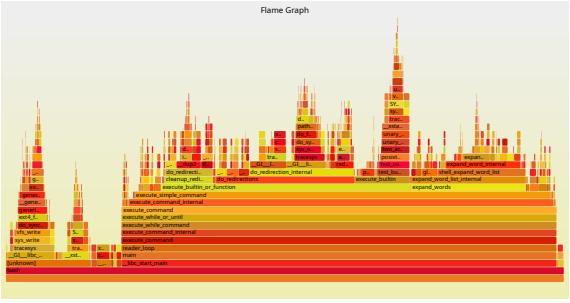


Abbildung 3: Flame graph (visualize profiled code)

Interested?
Mail to: paul.bergmann@fau.de

High Fidelity Latency Simulation

- Network Simulator simulates static latency
- **Reality:** Large latency spread across Wide Area Networks
- **Implement** configurable distribution in simulator
- Open source contribution

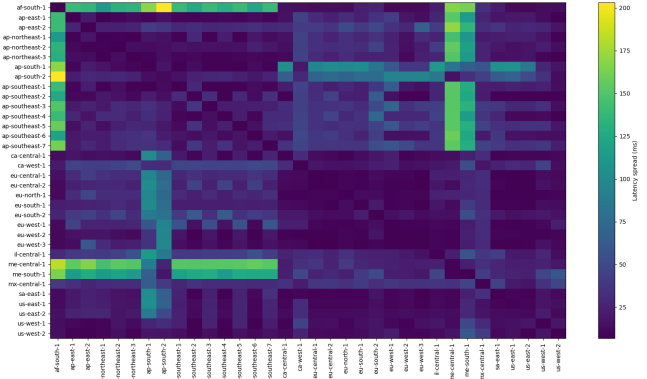


Abbildung 4: Latency Spread across AWS Regions

Interested?
Mail to: paul.bergmann@fau.de

Compartmentalization of Rust code

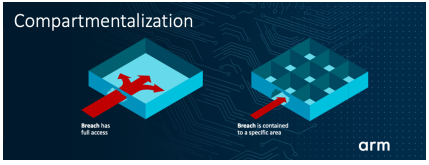


Abbildung 5: Concept[1]

Interested?

- in: Rust, SGX, compilers
- required: taken a course about compilers or OSes
- onciul@cs.fau.de

open topics

- Other isolation mechanisms (MMU,WASM,CHERI,..?)

→ generalize beyond SGX

- Dynamic dispatch across boundaries

→ extend existing compiler

- Benchmark real-world rust projects

→ measure performance implications

[1] https://www.eetimes.com/wp-content/uploads/media-1314103-armchericompartmentalization.png

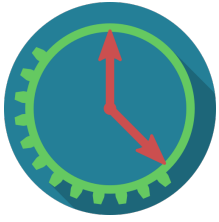
LLM Interface Generation

BFT Code Diversification
Diversified Byzantine Fault Tolerant implementations from mixed languages

- Small interfaces
- WebAssembly component model as shared platform
- LLM interface generation
- Verification against reference implementation

Interested?
Mail to: arne.vogel@fau.de

Virtio Treiber für JITTY OS



JITTY OS
Binärkompatibeles Unix-/Linux-ähnliches
Forschungsbetriebssystem unseres Lehrstuhls

- Virtio: *Straightforward*, *Efficient*, *Standard* & *Extensible*
- Implementierung von paravirtualisierten Treiber für Netzwerk, Blockgeräte & Konsole
- Vergleich mit existierenden (emulierten) Gerätetreibern

Interesse?
Für nähere Infos, Mail an: ott@cs.fau.de

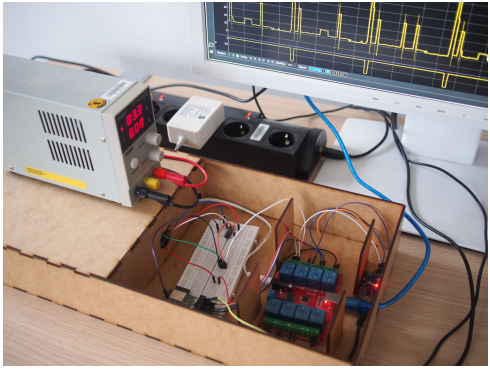
Energieoptimierung in Eingebetteten Systemen

BA, MA

Eingebettete Systeme

- Unterschiedliches Energieverhalten zwischen CPU und I/O-lastigen Operationen

→ Optimierungspotential durch geschickte Taktfrequenzwechsel



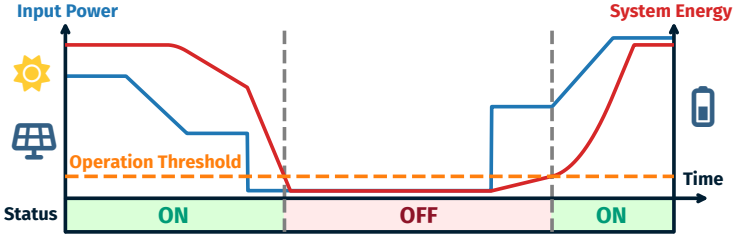
Thema

- Verwendung von Deep Reinforcement Learning, um optimale Frequenzwechsel zu bestimmen

Interesse?

Mail an: tobias.haeberlein@fau.de

Virtuell persistente Koroutinen in intermittierenden Systemen



Intermittierende Systeme

- Unvorhersehbare Energieversorgung von Sonnenenergie, etc.

→ Dadurch unregelmäßige Betriebsunterbrechungen

- Aber: Konsistenz und Fortschritt des Systems muss gewahrt werden

⇒ Ausführung mit Hilfe von (virtuell) *persistenten* Koroutinen

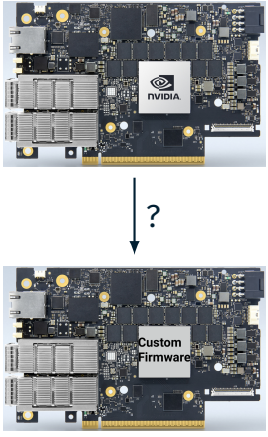
Interesse?
Mail an: preisner@cs.fau.de

Alternative Firmware für DPUs

MA

Thema

- Intelligente Netzwerkkarten (DPUs) bieten viele Möglichkeiten zur Auslagerung von sensitiven Operationen (z.B. Crypto) $\rightarrow$ Ziel von Angriffen
- ? Können wir die Firmware anpassen oder gar austauschen um mehr Vertrauen zu gewinnen?
- $\Rightarrow$ Möglichkeiten des SDKs und des Bootprozesses evaluieren um herauszufinden, was wir ändern können und was nicht



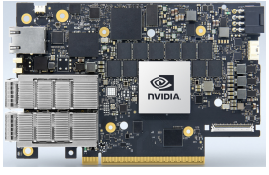
Interesse?
wiedemann@cs.fau.de

Crypto-Offloading von DPUs in OpenBSD

MA

Thema

- DPUs können die symmetrische Ver- und Entschlüsselung von IPSec und TLS übernehmen (Offloading).
- ? Aktuell gibt es keine Unterstützung in OpenBSD für Auslagerung an NVIDIA Bluefield DPUs
- ⇒ Erweiterung der Kernel- und Treiber-Infrastrukturen
- ⋆ MA erfolgt in Kooperation mit genua



Interesse?
wiedemann@cs.fau.de