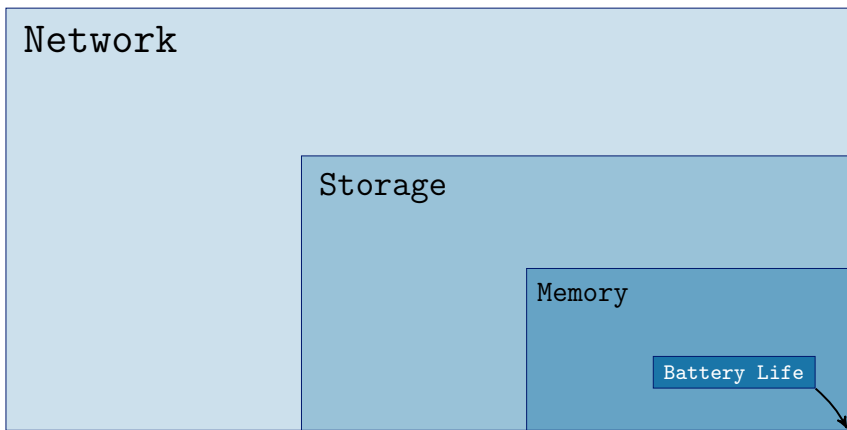


Forschung und studentische Arbeiten
Energiegewahre Programmierung
Verteilte energiegewahre Systeme
Sichere virtuelle Maschinen



- Fokus der Arbeiten des Lehrstuhls für Informatik 4 lässt sich am besten mit dem Begriff *Operating Systems Engineering* beschreiben
→ Verteilte Systeme, Betriebssysteme
- Forschungsgebiete
 - Energiegewahre Programmierung und energiegewahre Systeme
→ Timo
→ <https://www4.cs.fau.de/~thoenig/>
 - Verteilte energiegewahre Systeme
→ Christopher
→ <https://www4.cs.fau.de/~ceibel/>
 - Cloud-Computing, sichere Cloud-Computing-Komponenten
→ Klaus
→ <https://www4.cs.fau.de/~stengel/>
 - Byzantinische Fehlertoleranz (BFT), ressourceneffiziente BFT-Systeme
→ Johannes und Tobias
→ <https://www4.cs.fau.de/Research/REFIT/>





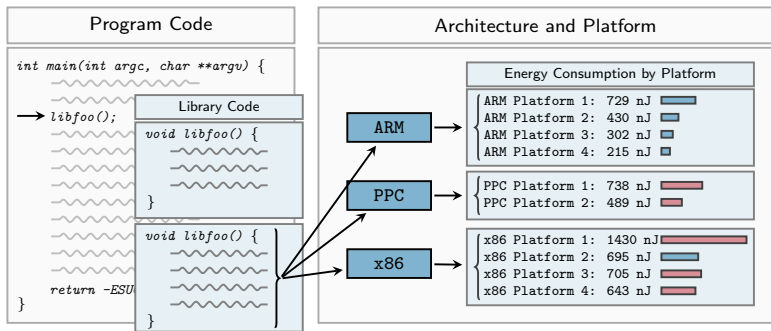
- Netzwerk: Faktor 3,3 Millionen → 300 Bit/s vs. 1 GBit/s
- Speicherkapazität: Faktor 1,4 Millionen → 360 kByte vs. 500 GByte
- Arbeitsspeicher: Faktor 0,5 Millionen → 4 kByte vs. 2 GByte
- Batterielaufzeit: Faktor 10 → 1 h vs. 10 h



- Fortschritt der letzten drei Jahrzehnte
 - Netzwerk: Faktor 3,3 Millionen
 - Speicherkapazität: Faktor 1,4 Millionen
 - Arbeitsspeicher: Faktor 0,5 Millionen
 - Batterielaufzeit → 1 h vs. 10 h (Faktor 10)
- Energiesparmechanismen
 - Dynamische Systemaspekte: aktive Beeinflussung des Systemverhaltens
 - Statische Systemaspekte: Compiler-Optimierungen, Speicherzugriffsmuster
- Energiegewahre Programmierung
 - Rückwärtsgerichteter Optimierungsprozess
 - Aufwendige Analysemethoden



Energiegewahre Programmierung mit SEEP



- Vorwärtsgerichteter Prozess, um Programmierer dabei zu unterstützen, energiegewahre Programme zu entwerfen
- Automatisierte Energieabschätzung für heterogene Plattformen



- System-Kontext
 - Klein(st)systeme (Stichwort: *Smart Dust*)
 - Eingebettete Systeme
 - Energy-Aware High-Performance-Computing (HPC), Exascale-Computing
- Aktuelle Forschungsarbeiten
 - Integrierte Ansätze für energiegewahre Programmierung
 - Energiebewertung einzelner *Software*-Systemkomponenten
 - Kombination mit Verfahren der statischen Codeanalyse
 - Bachelor- und Masterarbeiten
 - Studien- und Diplomarbeiten



T. Hönig, C. Eibel, R. Kapitzka, and W. Schröder-Preikschat
SEEP: Exploiting Symbolic Execution for Energy-Aware Programming
ACM Operating Systems Review (Best Papers from HotPower '11 Workshop on Power-Aware Computing and Systems) 45(3):58–62, 2012.



■ Masterarbeit:

Energieeffizienzsteigerung von Cloud-Computing-Anwendungen unter Ausnutzung heterogener Hardwarekomponenten

→ Herangehensweise: Feingranulare Partitionierung von Anwendungen und optimale Verteilung der Partitionen auf verschiedenartige Rechnersysteme

■ Hardware-Heterogenität

- Multiprozessor- ($\{SM, MP\}P/\dots$) u. Speicherarchitekturen ($\{N, \}UMA/\dots$)
- Hardwarebeschleuniger und Spezialprozessoren (DSPs, GPUs, NPUs, ...)
- Spezielle Instruktionen und Instruktionseinheiten
- ...

■ Bestimmung der jeweils energieeffizientesten Rechereinheit erfordert statische und dynamische Analysen aller verfügbaren Rechnersysteme

- Energieverbrauchsabschätzungen / Energiemodelle
- Betrachtung der Mehrkosten durch verteilte Ausführung

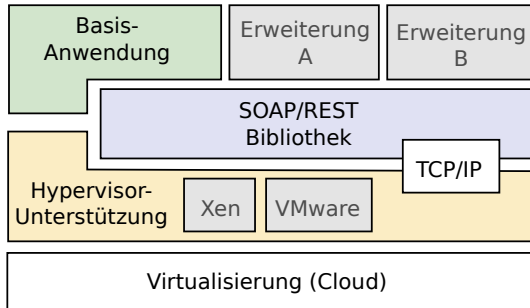
■ Weitere studentische Arbeiten (BA/MA/MP)

↔ Mail an Christopher <ceibel@cs.fau.de>



Sichere Dienste für Cloud-Computing

- Angepasste Architektur für Cloud-Umgebungen



- Reduzierte Angriffsfläche durch weniger Code
- Implementierung mit funktionaler Programmierung (Haskell)
- Partielle Verifikation



- Effective Aspects
 - memcached Prototyp
 - Flexiblere Rekonfiguration

- Intel SGX
 - Ausführungsmodus künftiger Intel CPUs
 - Verschlüsselung und Code-/Datenintegrität
 - Absicherung gegen böartigen Hypervisor/Betriebssystem

→ Master-Projekte

