CPS-Remus: Eine Hochverfügbarkeitslösung für virtualisierte cyber-physische Systeme

Boguslaw Jablkowski
Olaf Spinczyk
Department of Computer Science
TU Dortmund

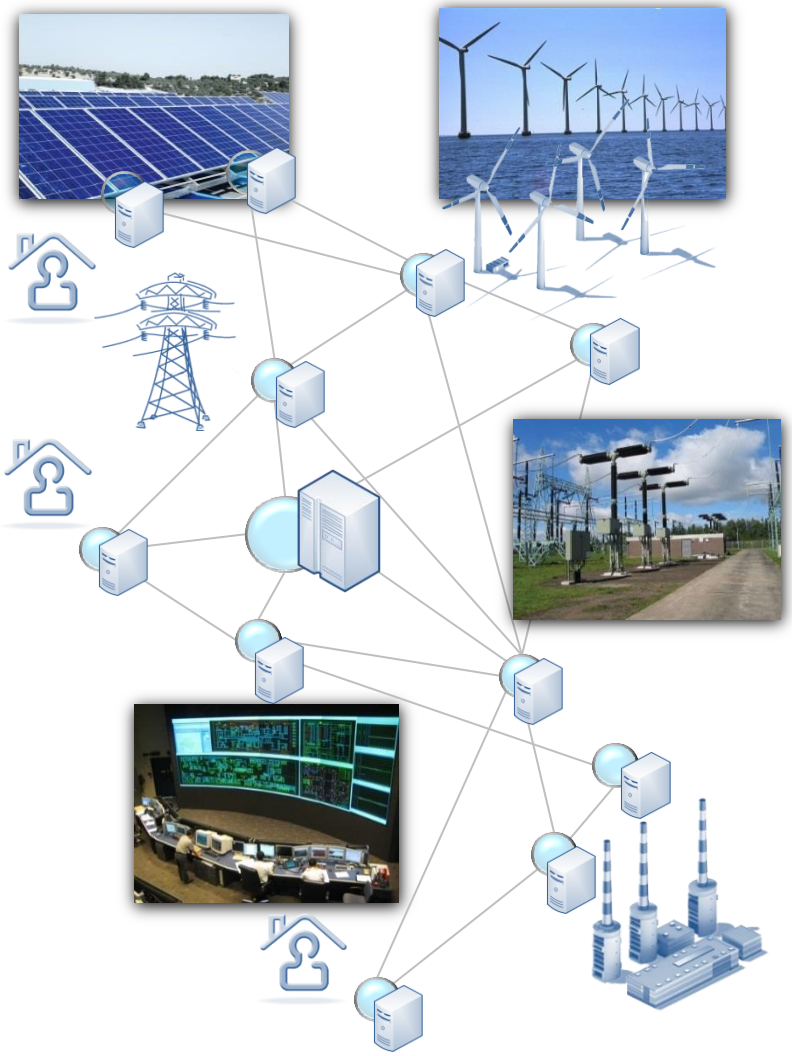
Agenda

- Motivation und Kontext
- CPS-Xen und die Architektur der Hochverfügbarkeitslösung
- Evaluation
- Fazit und Ausblick

Motivation und Kontext

Moderne Energiesysteme

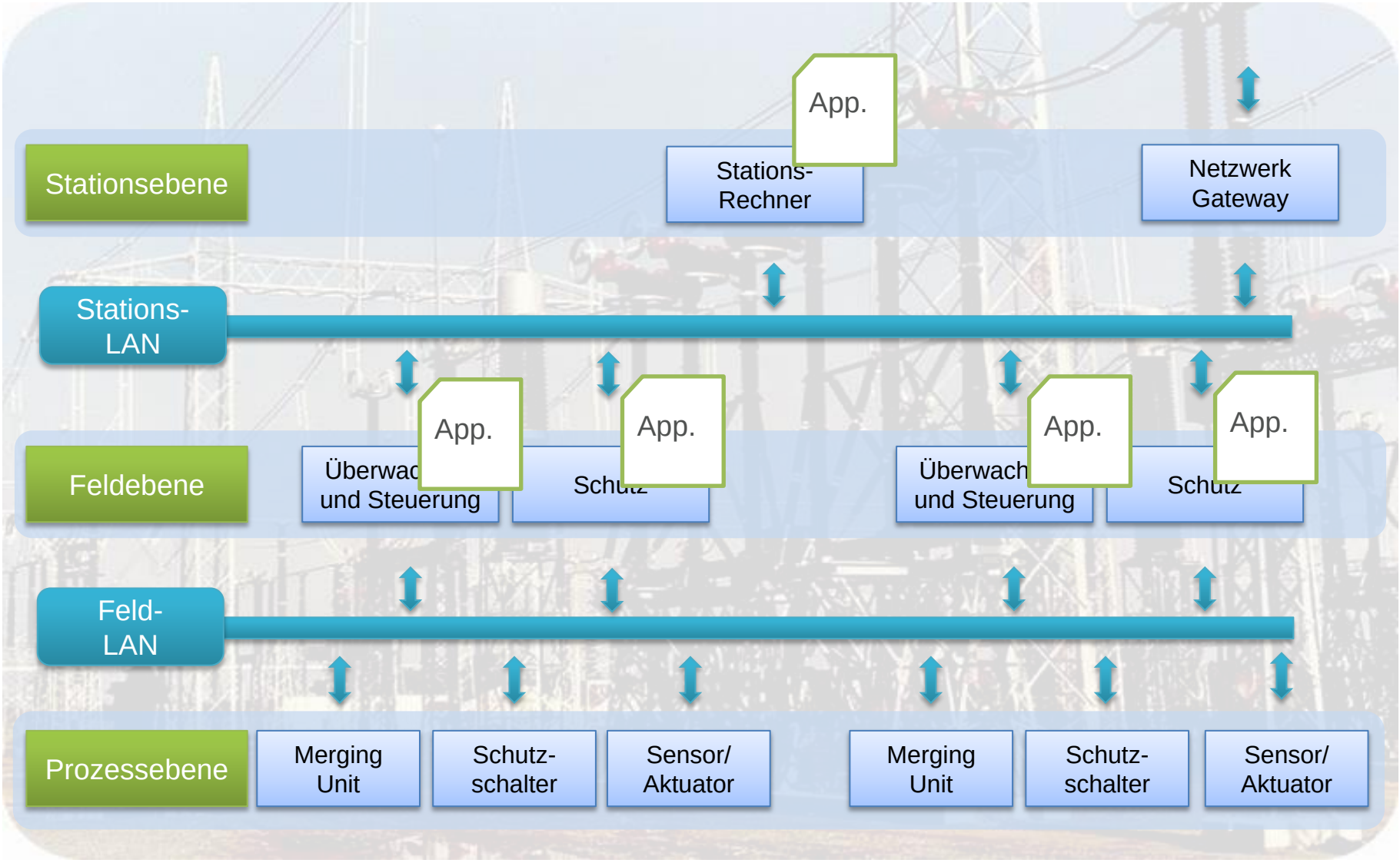
- verteilte und komplexe Systeme
- hochdynamisch
  - fluktuierendes Netz
  - volatile Einspeisung und Leistungsflüsse (Energiewende)
- hohes Maß an Überwachung und Steuerung notwendig
- steigende Rolle von Software
  - erhöht zusätzlich die Komplexität



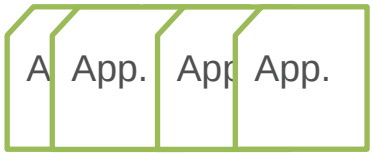
Ansatz Virtualisierung

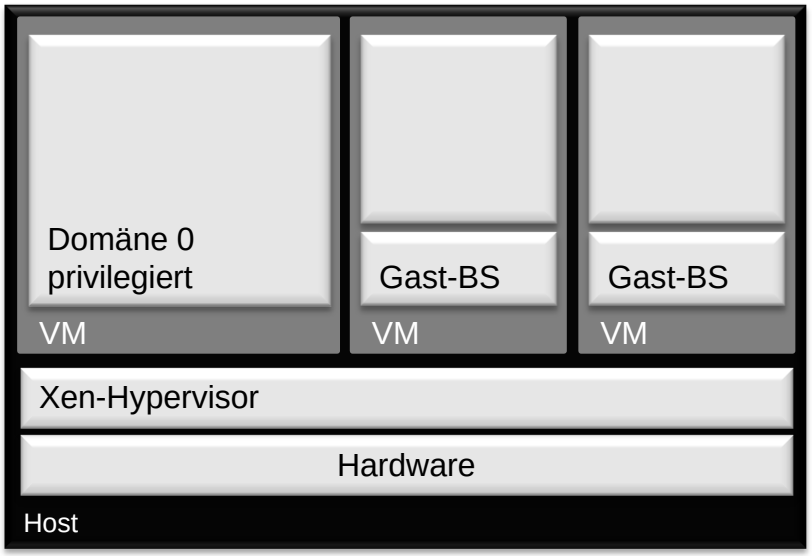
- bewährte Technologie für Systemintegration
- solide Fehlertoleranzeigenschaften
  - **Isolation** für Fehlereingrenzung
  - **Hochverfügbarkeit**
- gute Wartungs- und Administrationsmöglichkeiten
  - homogene Hardware
  - **Migration** von virtuellen Maschinen (VM)
- Entkopplung von dedizierter Hardware
- Kostensenkung (Energie, Wartung, Anschaffung etc.)

CPS Integration – Beispiel Schaltanlage



CPS Integration





Agenda

- Motivation und Kontext
- CPS-Xen und die Architektur der Hochverfügbarkeitslösung
- Evaluation
- Fazit und Ausblick

CPS-Xen Projekt

Ziel:

- **vorhersagbare und fehlertolerante Ausführungsplattform**

Grundlage:

- Xen Hypervisor [1]

CPS-Xen Erweiterungen

- drei Echtzeitscheduler
  - fixed-priority
  - rate-monotonic
  - deadline monotonic
- **ereignisbasierte Hochverfügbarkeit**

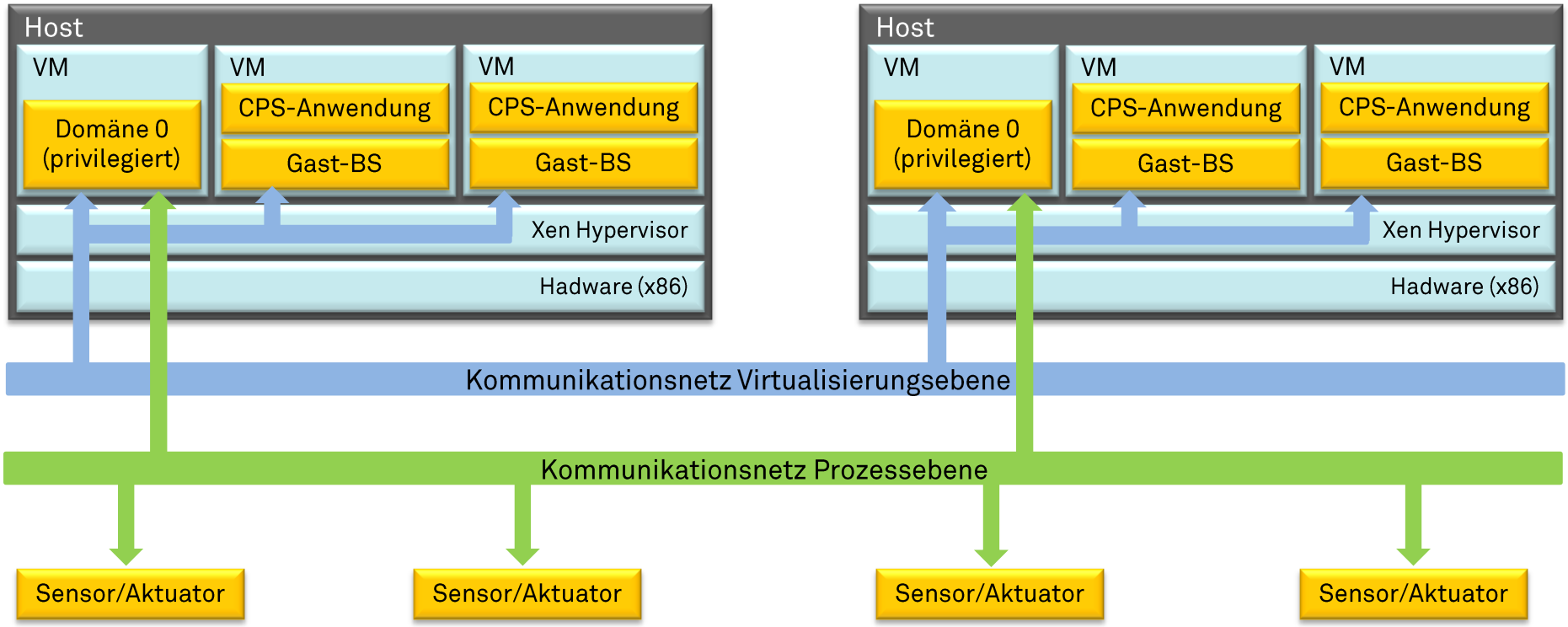
CPS-Xen Quelltext:

- auf GitHub [2]

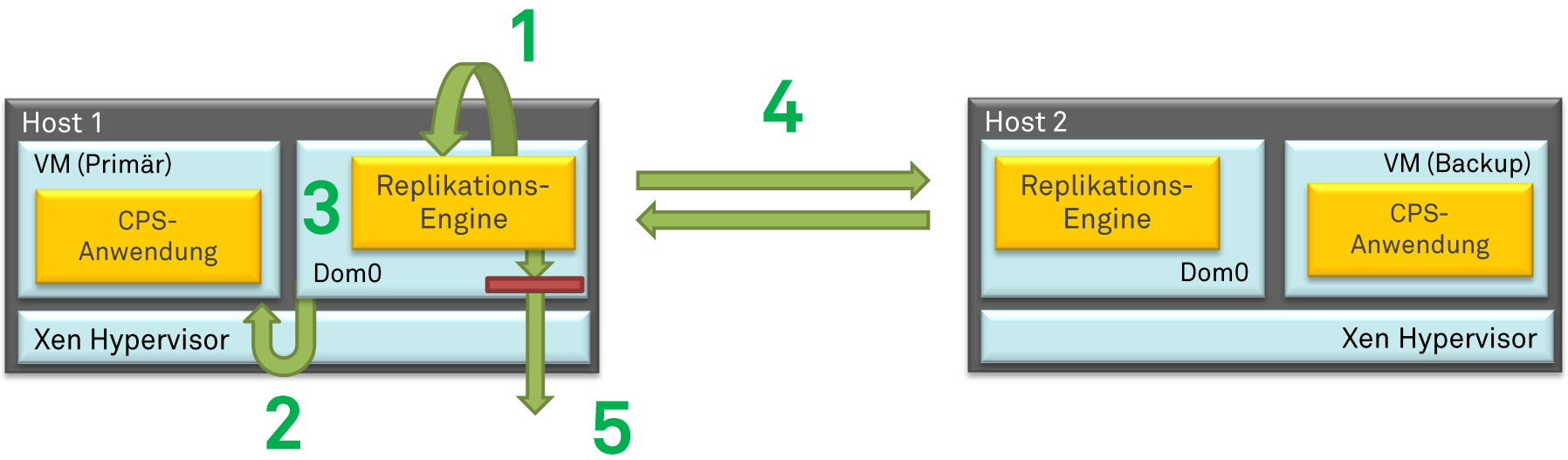
[1] http://www.xenproject.org/

[2] https://github.com/cpsxen/cps-xen

CPS-Xen Architektur

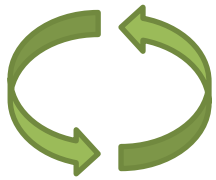


Remus – Periodische Hochverfügbarkeitslösung



1. Zyklisches Ereignis für die Absicherungsaktivierung tritt ein.
2. Die abzusichernde (primäre) VM wird suspendiert.
3. Die in der letzten Periode beschriebenen Speicherseiten werden zu einem Checkpoint zusammengefasst.
4. Die Zustandsdifferenz wird an die Sicherungskopie übertragen, die VM wird fortgesetzt, die Ausgaben werden zwischengepuffert.
5. Antwortfreigabe nach einer Bestätigung des Sicherungsservers.

Periodische Hochverfügbarkeit



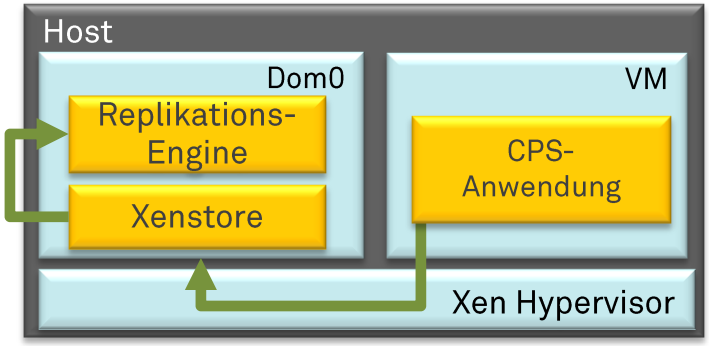
Ist dieses Modell geeignet für CPS?

- suboptimal
- unflexibel – VM-Absicherung nur mit fester Periode
- Ausführung kann in einem beliebigen Moment unterbrochen werden
- erhöht Latenz
- erzwingt eine konservative Parametrisierung des Systems
- erzeugt unnötigen Overhead

Zwischenspeicherung der Ausgaben erforderlich?

- eher nicht
- Sensoren und Aktuatoren sprechen kein TCP und benötigen keine konsistente Zustandshistorie.

Ereignisbasierte Hochverfügbarkeit

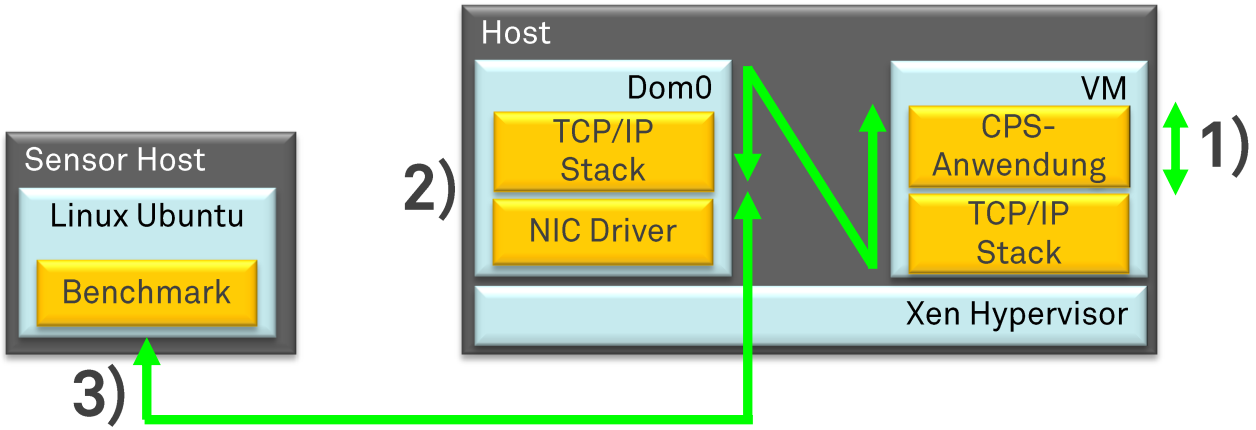


- VM entscheidet selbst über den Zeitpunkt der Absicherung
  - z.B. nach einem bestimmten Ausführungspfad im Programm
- Realisierung über den ***Xenstore***
  - hierarchisch aufgebauter Namensraum (ähnelt *sysfs*)
  - angesiedelt in Dom0
  - kann von allen VMs genutzt werden
  - Zustand jedes Eintrags kann von einem sog. ***watch*** überwacht werden
  - Ereigniserzeugung für Checkpoints durch Schreiben in Xenstore

Agenda

- Motivation und Kontext
- CPS-Xen und die Architektur der Hochverfügbarkeitslösung
- Evaluation
- Fazit und Ausblick

Latenztypen und Messstellen



Hardware:

- Dell PowerEdge R620
- Intel Xeon E5 (8 cores, 2.6GHz)
- Intel I350 1 Gbit Ethernet

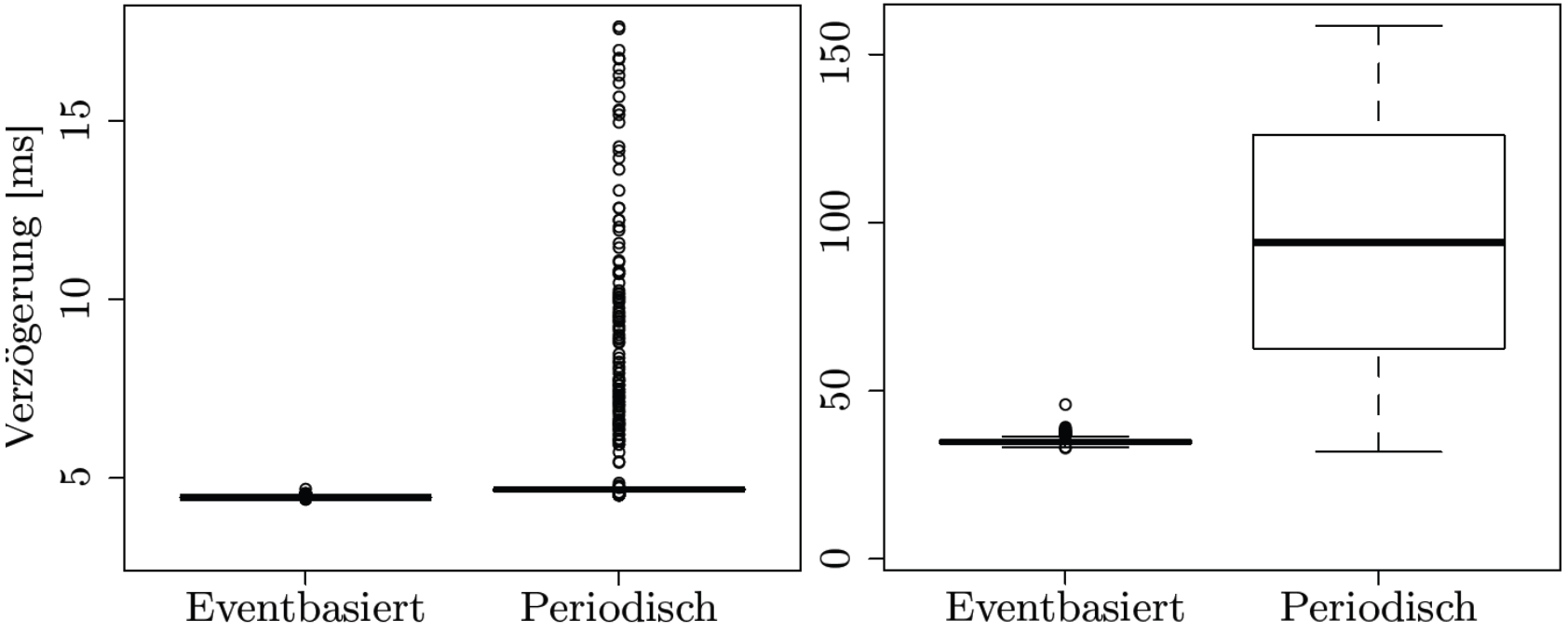
Operating system:

- Ubuntu 14 Server (Kernel 3.13)
- CPS-Xen (Xen version 4.5)

#	Latenztypen	Methode
1	Ausführungszeit	zyklengenaue Messung
2	Systemantwortszeit	systemtap – TCP/IP Stack – Sicherungsschicht (2)
3	Umlaufszeit	POSIX – clock_gettime()

Ergebnisse - Systemantwortszeiten

Periodisches vs. ereignisbasiertes Checkpointing:



a) Ohne Netzwerkpufferung

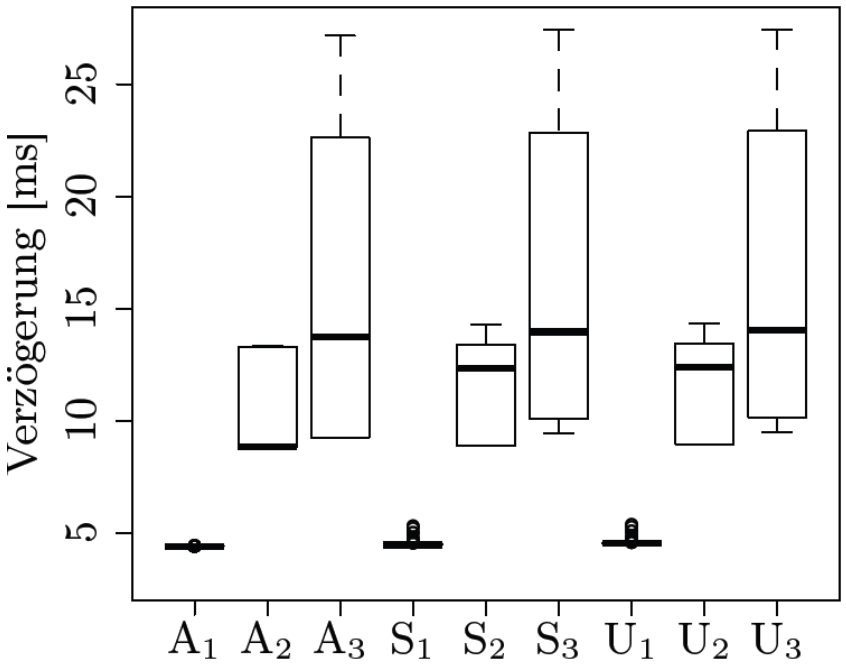
b) Mit Netzwerkpufferung

CPS-Anwendung mit:

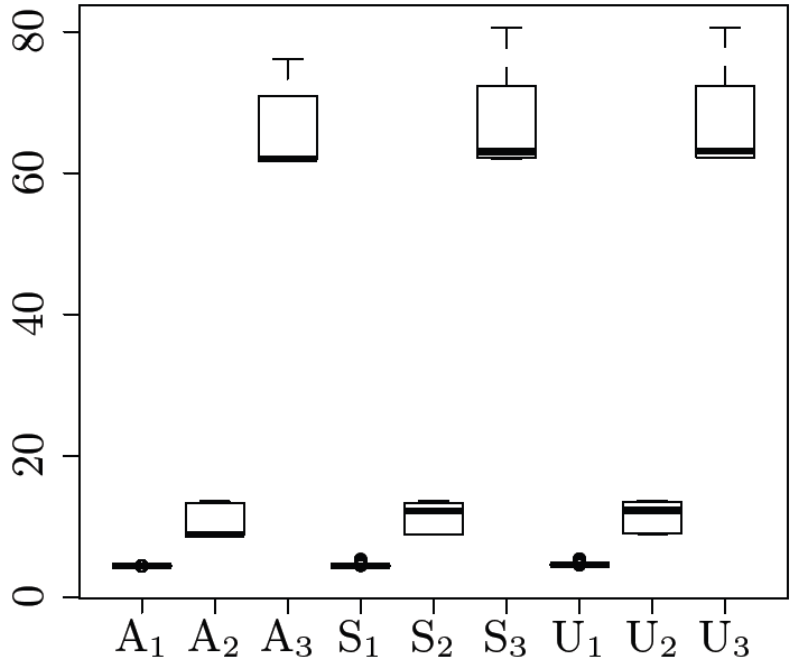
- Periode 100 ms, Bearbeitugnsbedarf (WCET) 4,5 ms
- Checkpointgröße ca. 2,8MB -> Netzwerkübertragungslatenz (1 Gbit-Netz) ca. 22ms

Ergebnisse – Skalierbarkeit - Prozessorlast

Ereignisbasiertes Checkpointing ohne Netzwerkpufferung:



a) Prozessorlast 50 %



b) Prozessorlast 80 %

#	VM	Periode/Deadline	WCET
1	VM1	20 ms	4,5 ms
2	VM2	50 ms	9,0 ms
3	VM3	100 ms	10 – 41,9 ms

Fazit und Ausblick

Ereignisbasierte Hochverfügbarkeit:

- deutliche Verbesserung der nichtfunktionalen Eigenschaften
- erweitert den Konfigurationsraum
  - geeignet für sporadische und aperiodische Modelle

Ausblick

- Leichtgewichtige Fehlertoleranz mittels maßgeschneideter Betriebssysteme (*unikernels*)
  - OpenSuse für Virtual Appliance
    - im Leerlauf 500-687 Speicherseiten in 100 ms
    - bis zu 2.81 MB -> Netzwerklatenz ca. 22 ms
  - vKratos, CIAO, MiniOS
    - im Leerlauf 4-7 Specherseiten in 100 ms
    - bis zu 28 kB -> Netzwerklatenz ca. 0.22 ms