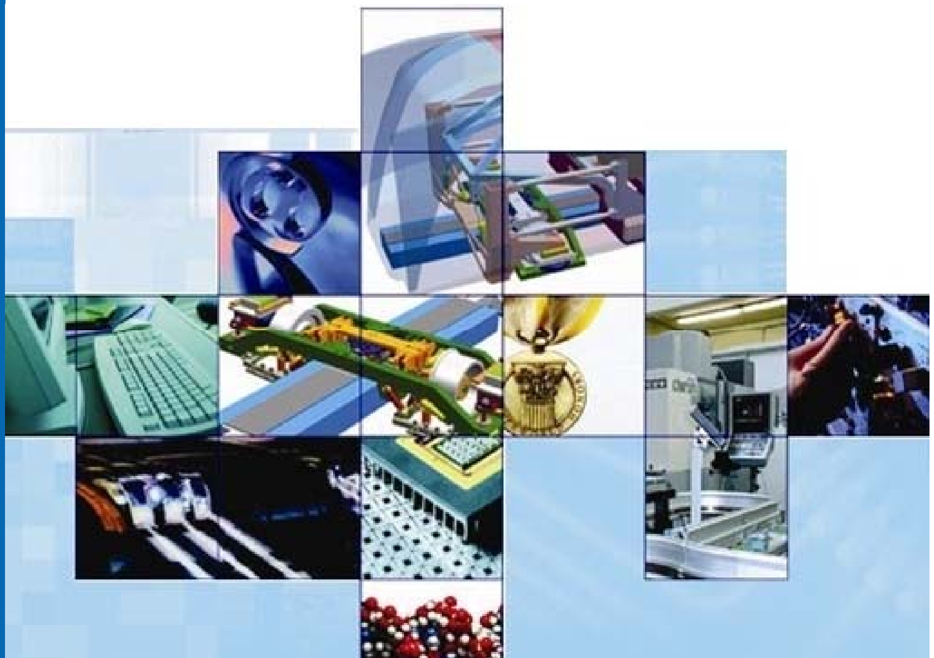
Sonderforschungsbereich 614
Selbstoptimierende Systeme des Maschinenbaus

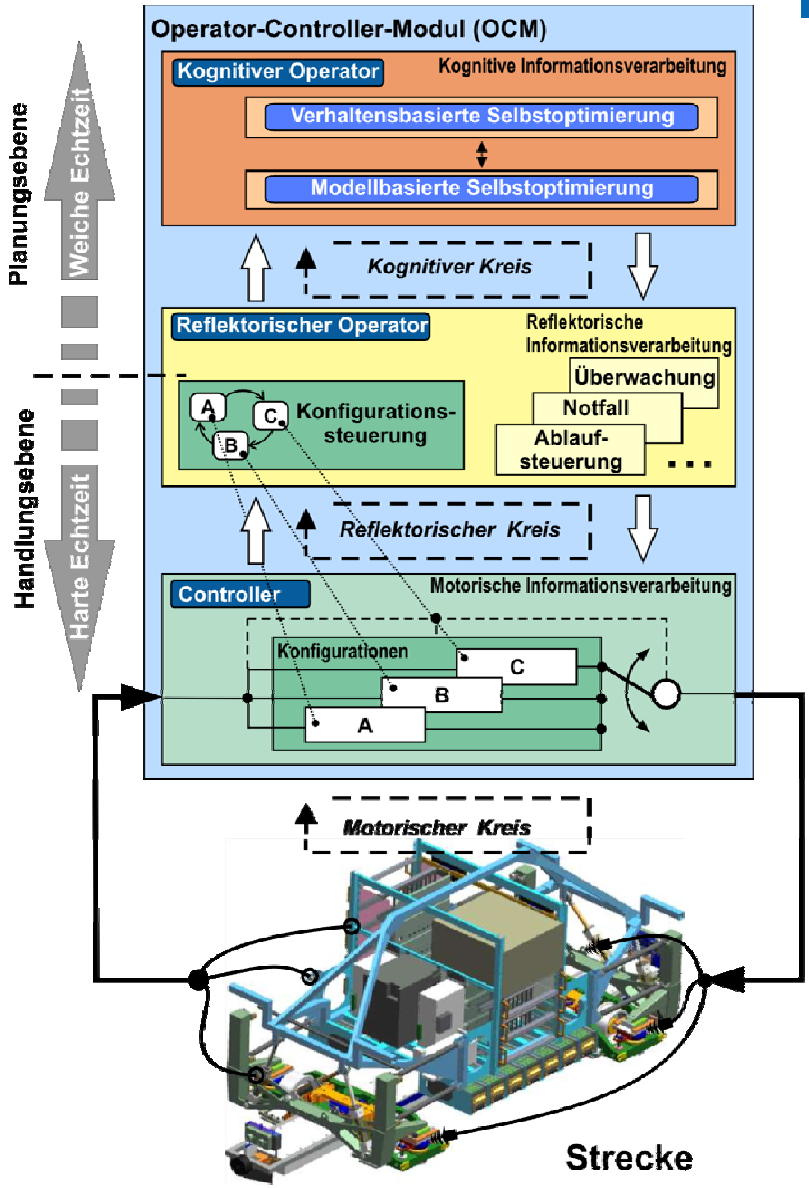
Formale Bestimmung von Systemparametern zum transparenten Scheduling virtueller Maschinen



Operator Controller Module

HEINZ NIXDORF INSTITUTE
University Paderborn
Design of Parallel Systems
Prof. Dr. rer. nat. Franz J. Rammig

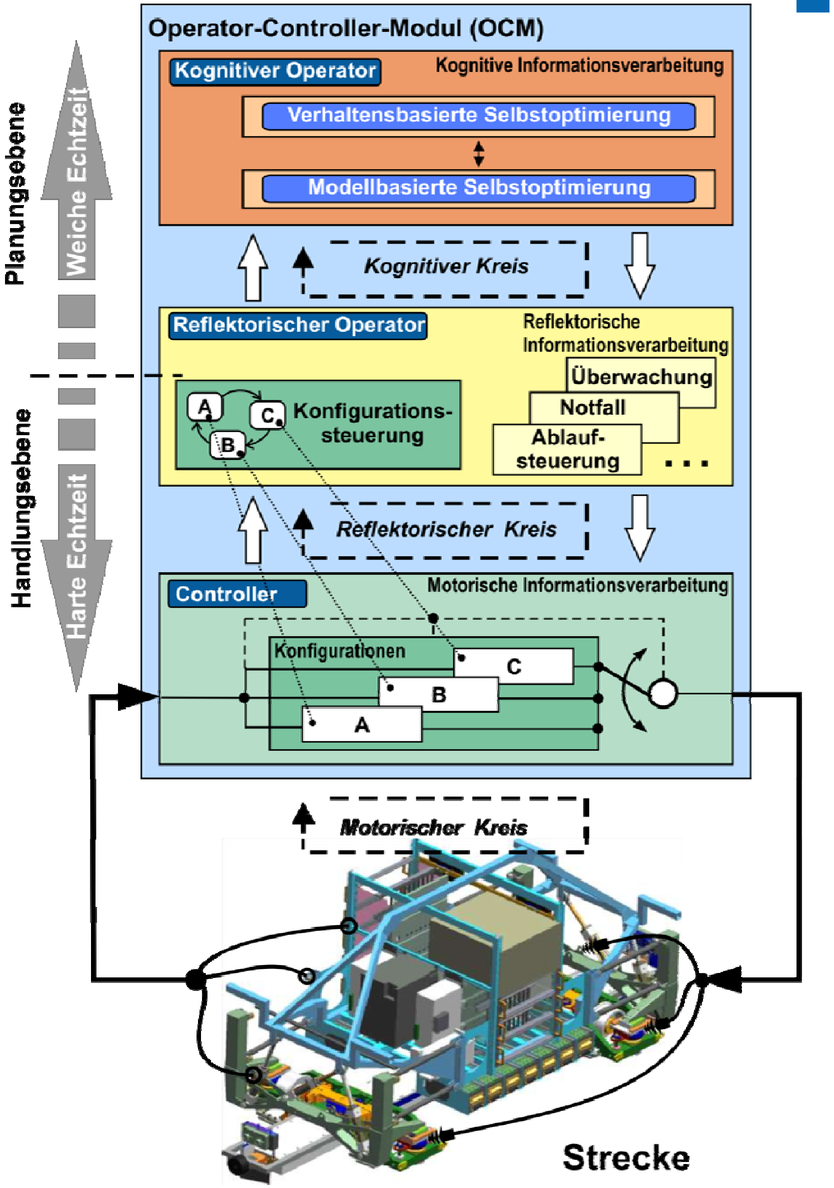
- Wichtige Eigenschaften
  - Weiche und harte Echtzeit
  - Dynamik
    - Ressourcenanforderungen
    - Benötigte Dienste
    - Unterschiedliche Betriebsmodi
  - Komfortkomponenten
    - Zuständig für Optimierung
    - Kognitiver Operator kann ausgeschaltet werden
  - Fail-Safe-Verhalten



HEINZ NIXDORF INSTITUTE
University Paderborn
Design of Parallel Systems
Prof. Dr. rer. nat. Franz J. Rammig

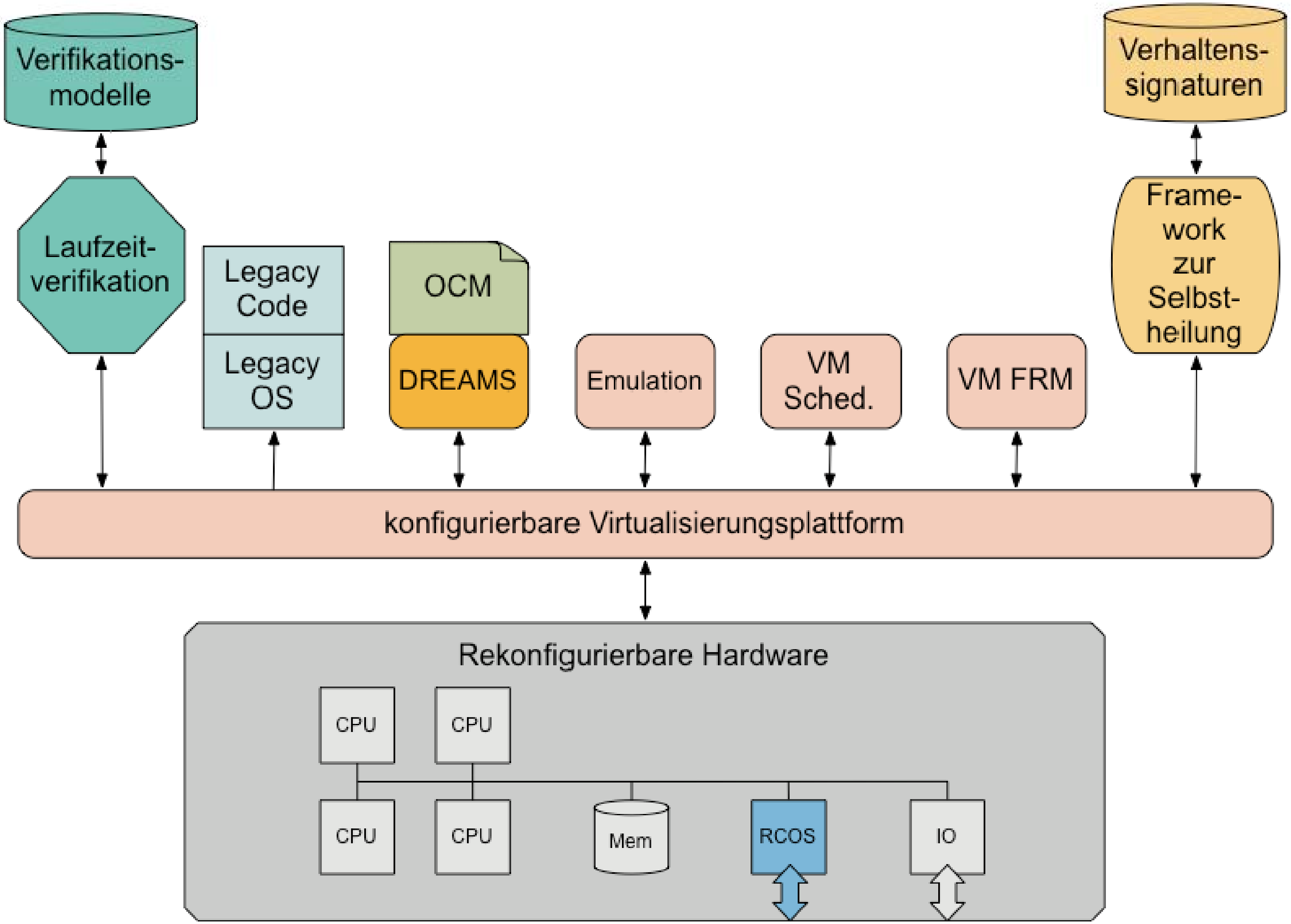
Motivation

- Problematik
  - Softwareintegration
  - Eingebettete Systeme sind oft sehr komplex
  - Sicherheit durch Redundanz
  - Kommunikation zwischen Systemen schwierig
- Virtualisierung bietet
  - Räumliche Isolation
  - Temporale Isolation
  - Sichere Integration verschiedener Software-komponenten
  - Vereinfachung von Kommunikationswegen
  - Sehr gute Skalierbarkeit



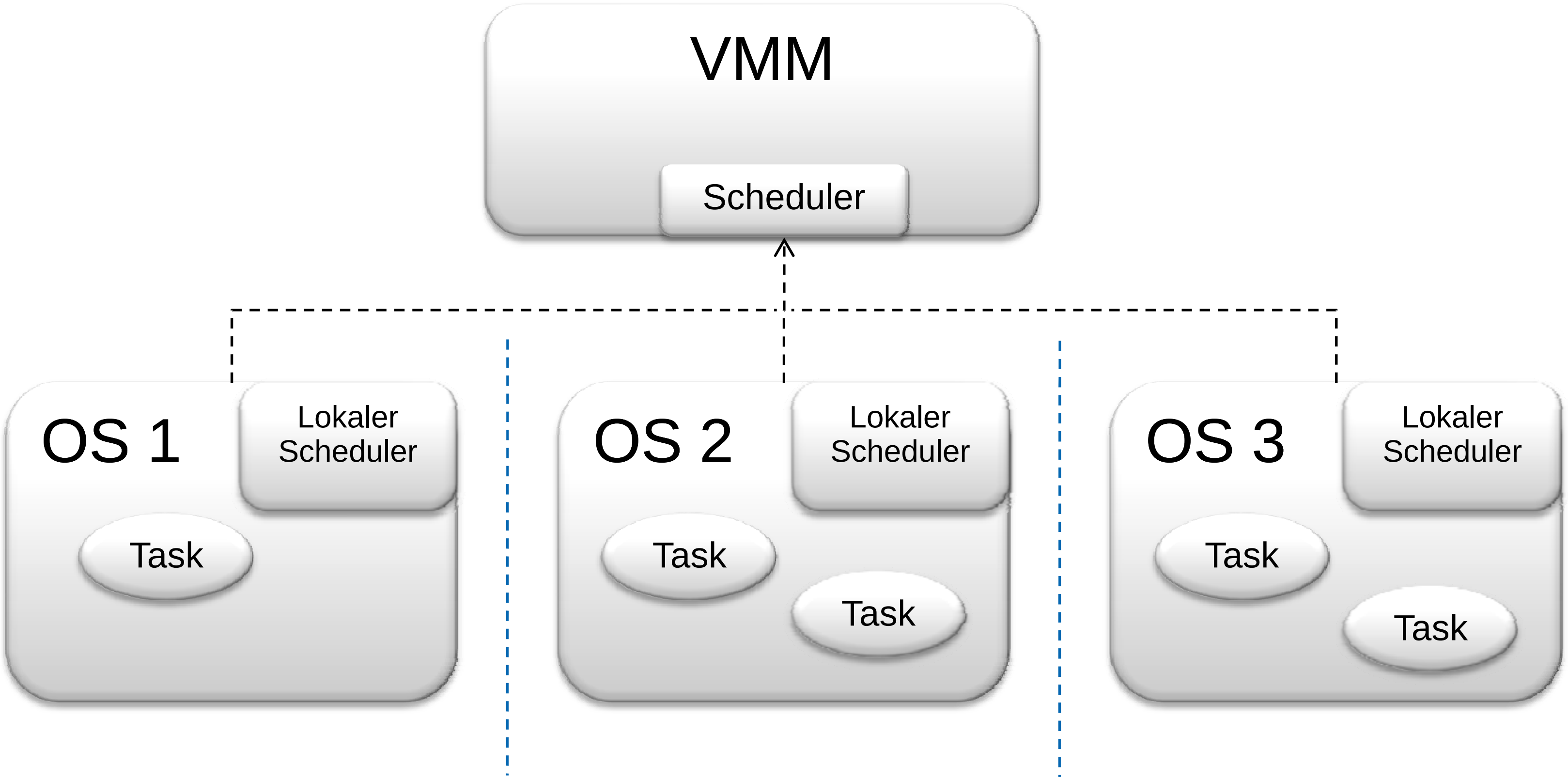
Das Framework

HEINZ NIXDORF INSTITUTE
University Paderborn
Design of Parallel Systems
Prof. Dr. rer. nat. Franz J. Rammig



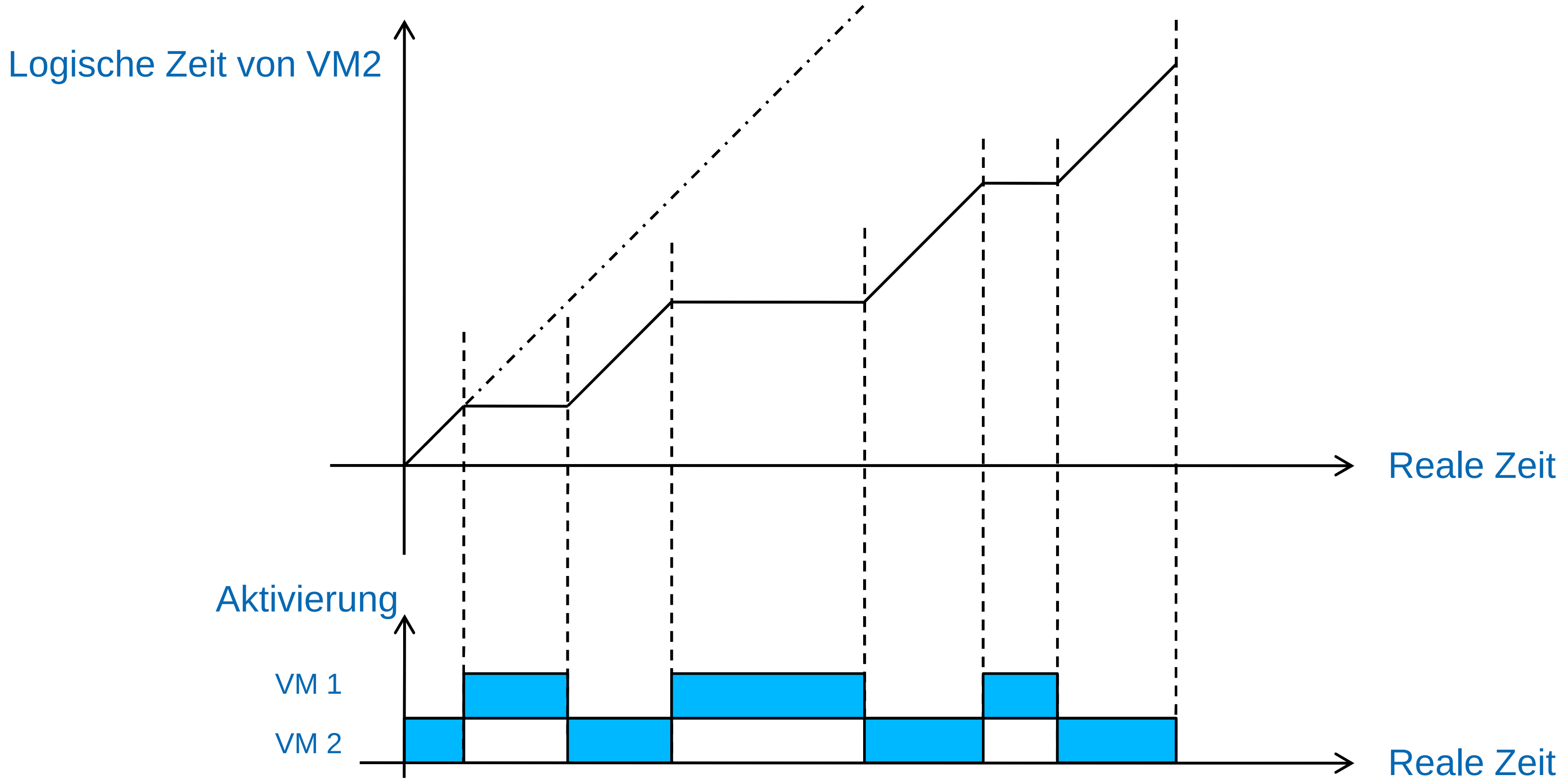
Scheduling

HEINZ NIXDORF INSTITUTE
University Paderborn
Design of Parallel Systems
Prof. Dr. rer. nat. Franz J. Rammig



HEINZ NIXDORF INSTITUTE
University Paderborn
Design of Parallel Systems
Prof. Dr. rer. nat. Franz J. Rammig

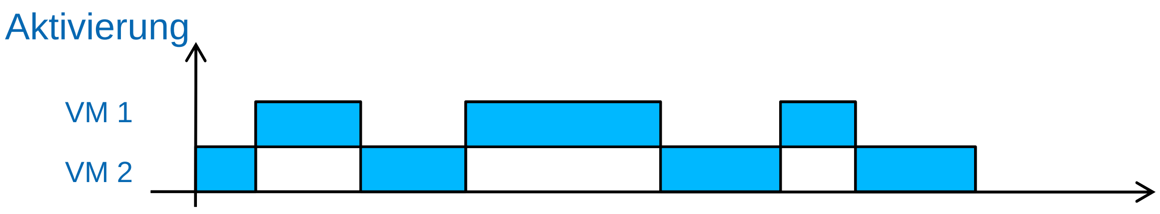
Scheduling



Scheduling

HEINZ NIXDORF INSTITUTE
University Paderborn
Design of Parallel Systems
Prof. Dr. rer. nat. Franz J. Rammig

- Scheduling auf verschiedenen Ebenen
  - In jeder virtuellen Maschine
  - Im Virtual Machine Monitor
  - Voraussetzung:
    - Vollvirtualisierung
  - Fragestellung:
    Wie müssen die virtuellen Maschinen geplant werden,um harte Echtzeitanforderungen einhalten zu können.



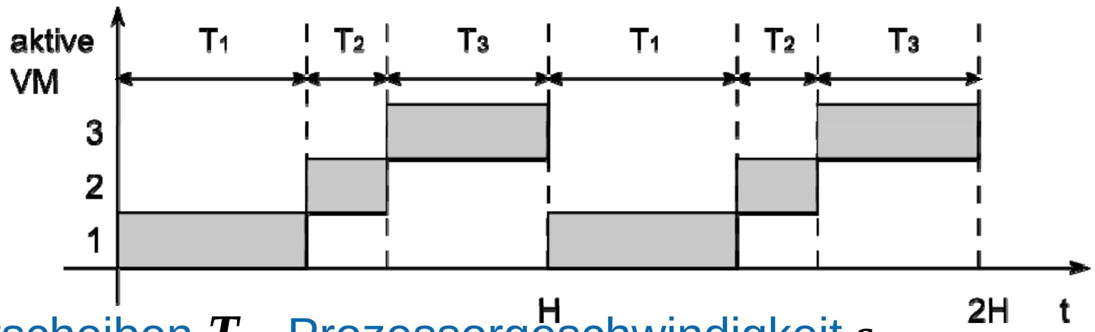
HEINZ NIXDORF INSTITUTE
University Paderborn
Design of Parallel Systems
Prof. Dr. rer. nat. Franz J. Rammig

Die Idee

- Der Einsatz einer schnelleren CPU ermöglicht die Ausführung mehrerer virtueller Maschinen.

- Aber wie müssen die virtuellen Maschinen geplant werden?

FTS Scheduling



Parameter: Zeitscheiben T_i , Prozessorgeschwindigkeit s

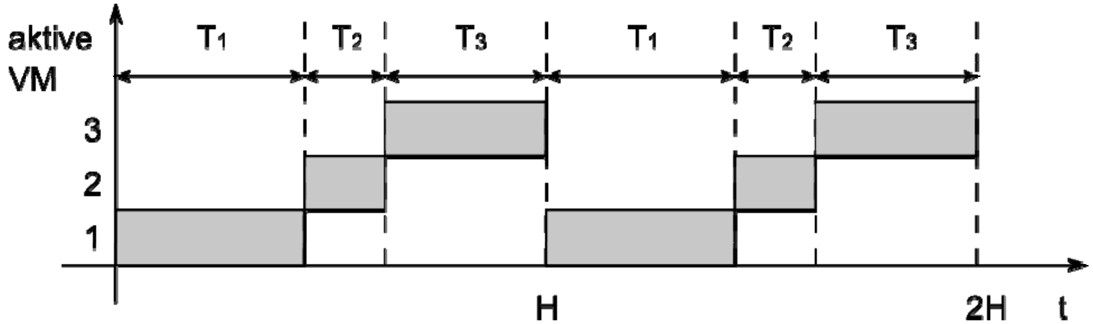
Pro:

- á priori Wissen der Aktivierungszeitpunkte
- Leicht zu implementieren

Contra:

- Planung eventuell zu restriktiv

Die Idee



- Prozessor des Hostsystems soll so klein wie nötig gehalten werden.

- Einfaches Geschwindigkeitsmodell mittels Speedup Faktor

Aufgabe:

- Gegeben sei eine Menge virtueller Maschinen. Finde Zeitscheiben T_i und den kleinsten Speedup s, damit die Echtzeitschranken aller virtueller Maschinen weiterhin eingehalten werden können.

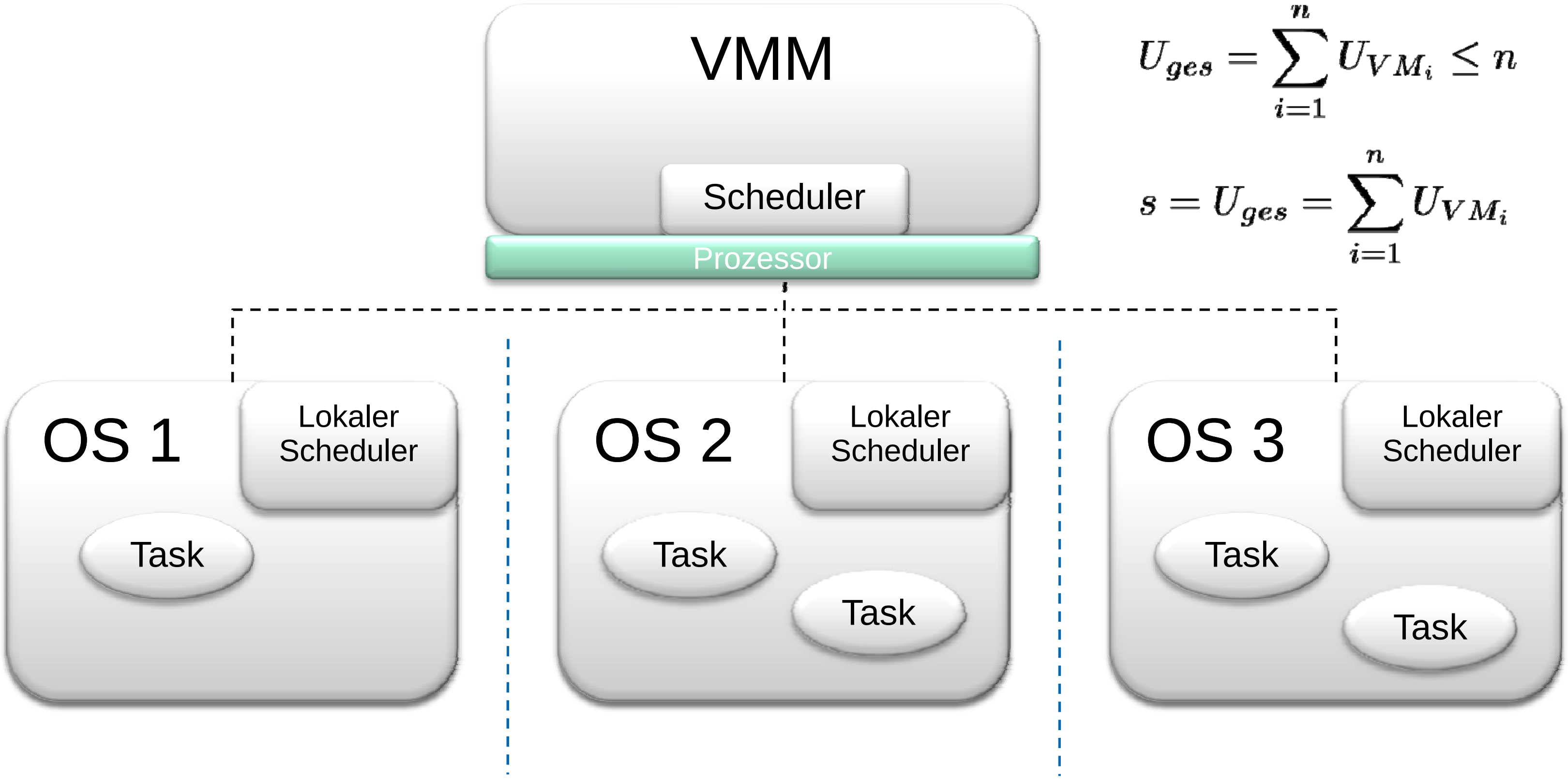
Die Idee

$$\tau = \{(C,T) | C,T \in \mathbb{R} \land C \leq T\}$$

$$t \in \tau = (C_t, T_t)$$

$$U_{ges} = \sum_{i=1}^{n} U_{VM_i} \leq n$$

$$s = U_{ges} = \sum_{i=1}^{n} U_{VM_i}$$



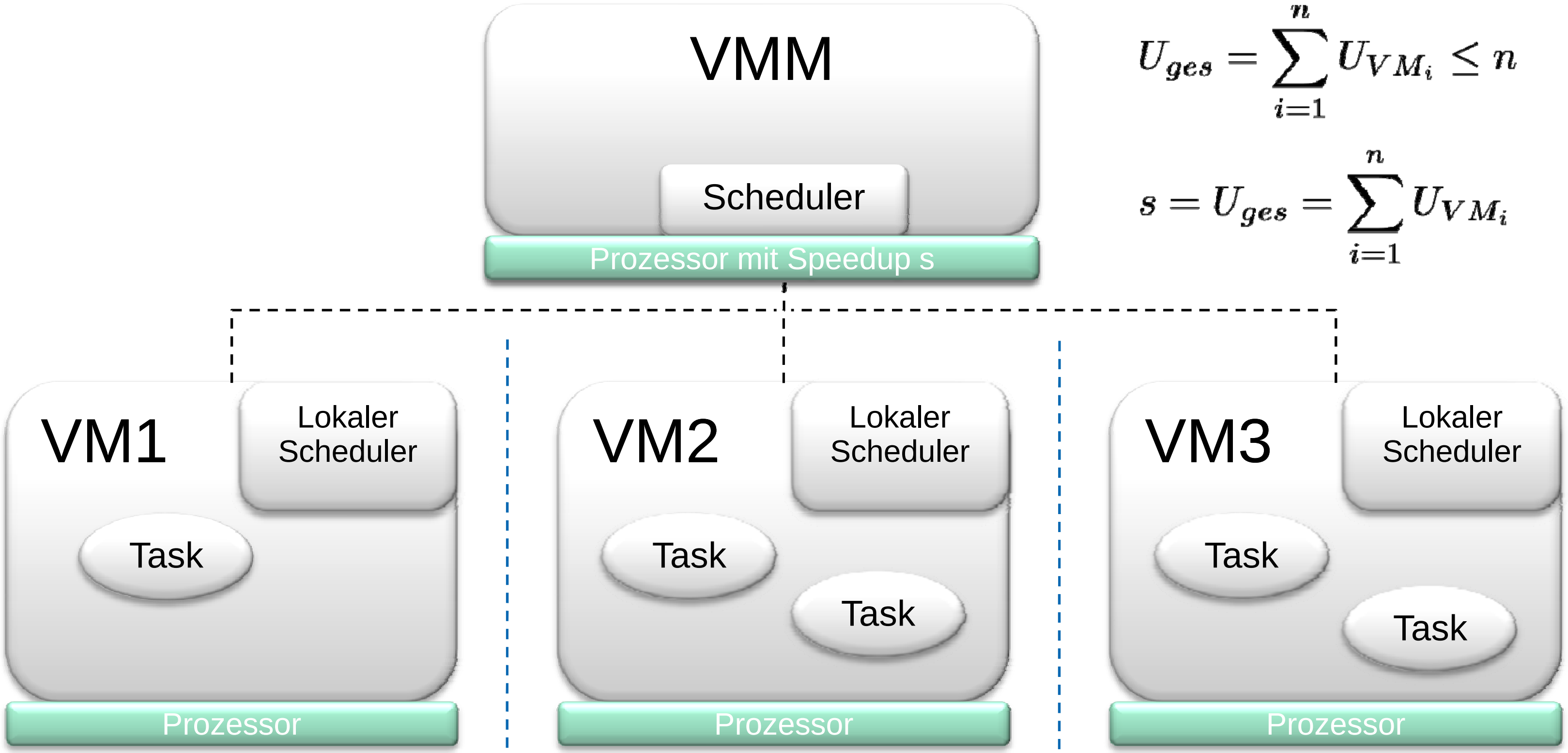
$$VM_i \subseteq \tau, i \in \{1, ..., n\}$$

$$U_{VM_i} = \sum_{j \in VM_i} \frac{C_j}{T_j} \leq 1$$

Die Idee

$$\tau = \{(C,T) | C,T \in \mathbb{R} \wedge C \leq T\}$$

$$t \in \tau = (C_t, T_t)$$



$$U_{ges} = \sum_{i=1}^{n} U_{VM_i} \leq n$$

$$s = U_{ges} = \sum_{i=1}^{n} U_{VM_i}$$

$$\tau' = \{(\frac{C_t}{s}, T_t) | t \in \tau\}$$

$$VM_i' = \{(\frac{C_t}{s}, T_t) | t \in VM_i\}$$

$$U_{VM_i'} = \sum_{j \in VM_i} \frac{C_j}{s \cdot T_j} = \frac{1}{s} \sum_{j \in VM_i} \frac{C_j}{T_j} = \frac{1}{s} \cdot U_{VM_i}$$

$$U_{ges}' = \frac{1}{s} \sum_{i=1}^{n} U_{VM_i} = \frac{1}{s} \cdot U_{ges} = 1$$

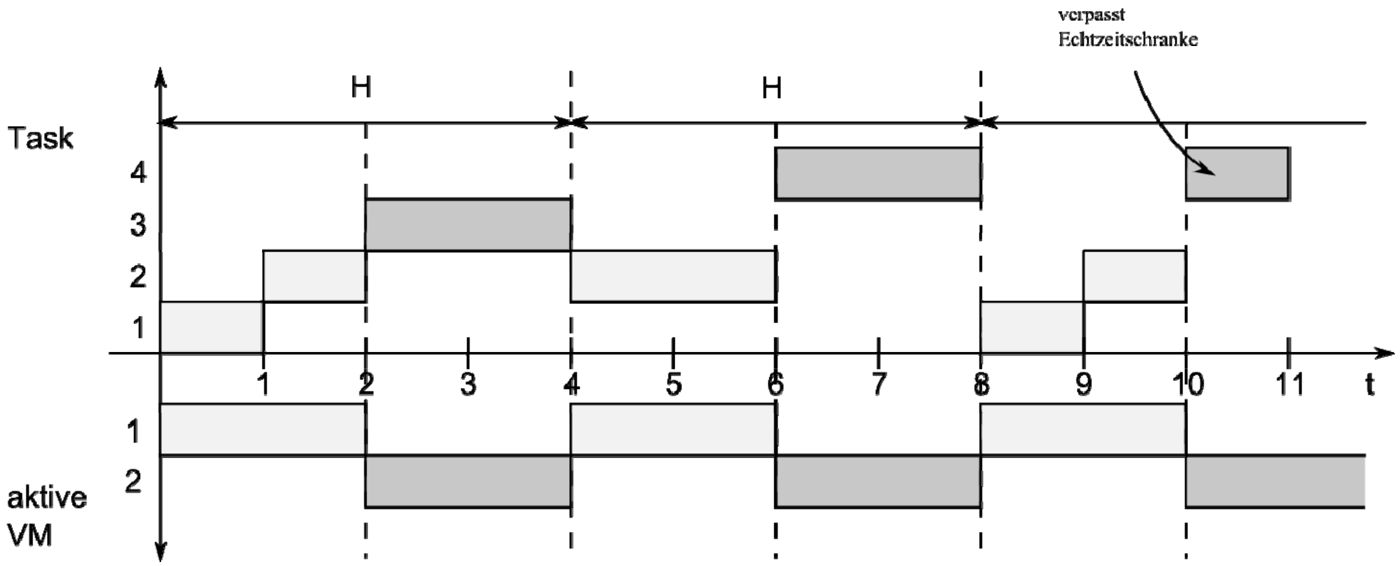
Beispiel

Die richtige Wahl der Zeitscheiben T_i ist von großer Bedeutung

Betrachten wir zwei virtuelle Maschinen:

$$VM_1 = \{(1,8),(3,8)\}$$
$$VM_2 = \{(2,10),(3,10)\}$$

Beide virtuelle Maschinen haben eine Auslastung von 50%: $T_{1,2} = 2$



Zeitscheibenwahl

HEINZ NIXDORF INSTITUTE
University Paderborn
Design of Parallel Systems
Prof. Dr. rer. nat. Franz J. Rammig

- Wie werden die Zeitscheiben bestimmt, so dass die Echtzeiteigenschaften der einzelnen VMs erhalten werden können?
- Beobachtung:
  - Im allgemeinen gilt nicht, dass jede VM zu jedem beliebigen Zeitpunkt ihre angeforderte CPU Zeit erhalten hat.
  - Erst am Ende eines Umlaufs H des FTS Schedulers haben alle VMs ihre angeforderte CPU Zeit erhalten.
  - Zum Zeitpunkt jeder Echtzeitschranke muss der zugehörigen VM ihre benötigte CPU Zeit zugewiesen worden sein!

$$H = ggT(\{T_t | t \in \tau'\})$$
$$\Rightarrow T_t = c \cdot H, c \in \mathbb{N}^+$$

  - Bestimmung der einzelnen Zeitscheibenlängen T_{VMi} der VM_i

$$T_{VM'_i} = \frac{U_{VM'_i}}{U'_{ges}} \cdot H \qquad \frac{T_{VM'_i}}{H} = \frac{U_{VM'_i}}{U'_{ges}} = \frac{U_{VM_i}}{U_{ges}}$$

Beispiel

HEINZ NIXDORF INSTITUTE
University Paderborn
Design of Parallel Systems
Prof. Dr. rer. nat. Franz J. Rammig

Source System

$\tau = \{(1,4),(2,4),(3,8),(1,4)\}$ $\quad U_1 = \frac{3}{4},\ U_2 = \frac{5}{8}$

$VM_1 = \{\tau_1, \tau_2\},\ VM_2 = \{\tau_3, \tau_4\}$ $\quad U_{ges} = \frac{11}{8}$

$$s = U_{ges} = \frac{11}{8}$$

Target System

$$\tau' = \{(\frac{8}{11},4),(\frac{16}{11},4),(\frac{24}{11},8),(\frac{8}{11},4)\}$$

$$U_1' = \frac{6}{11},\ U_2' = \frac{5}{11}$$

Beispiel

HEINZ NIXDORF INSTITUTE
University Paderborn
Design of Parallel Systems
Prof. Dr. rer. nat. Franz J. Rammig

Target System

$$\tau' = \{(\frac{8}{11},4),(\frac{16}{11},4),(\frac{24}{11},8),(\frac{8}{11},4)\}$$

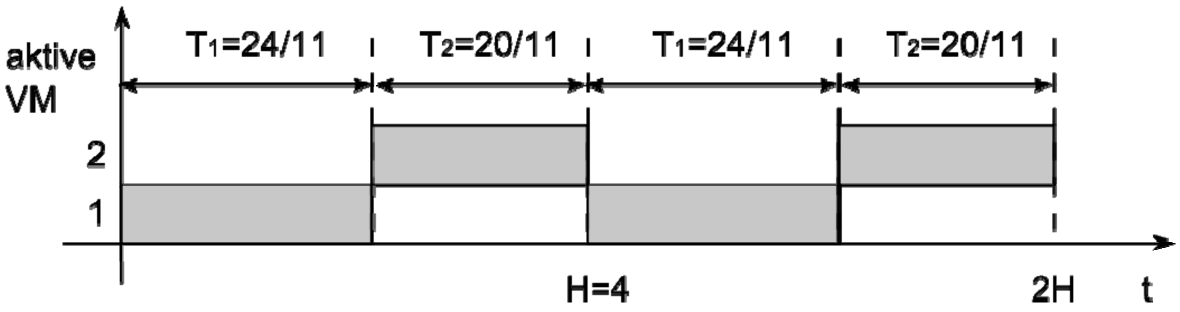
$$U'_1 = \tfrac{6}{11},\ U'_2 = \tfrac{5}{11}$$

$$H = ggT(\{T_t | t \in \tau'\}) = 4$$

$$T_{VM_i} = \frac{U_{VM'_i}}{U'_{ges}} \cdot H$$

$$T_1 = \frac{6}{11} \cdot 4 = \frac{24}{11}$$

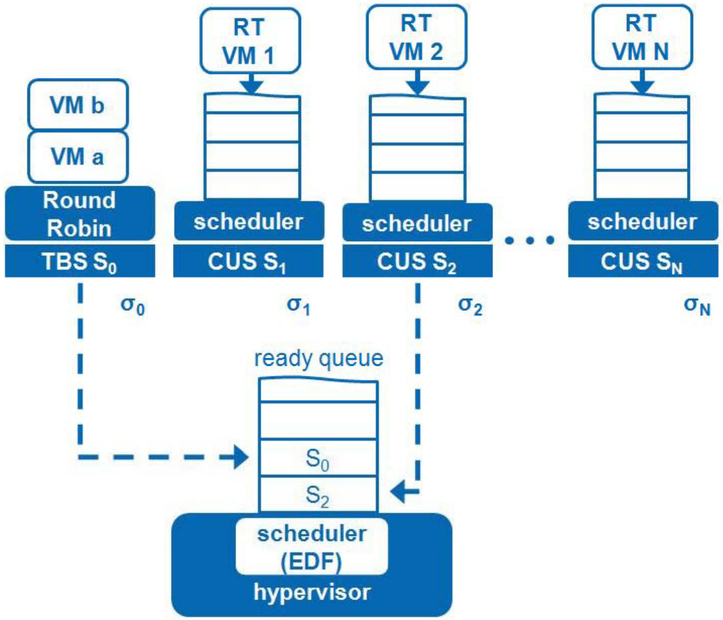
$$T_2 = \frac{5}{11} \cdot 4 = \frac{20}{11}$$



Verwandte Arbeiten

- Deng, Liu and Sun
  - Virtual Processor

  - Scheduling in an Open System Environment



- Feng und Mok: Bounded Delay Resource Partition Model

Zusammenfassung

HEINZ NIXDORF INSTITUTE
University Paderborn
Design of Parallel Systems
Prof. Dr. rer. nat. Franz J. Rammig

- Verwandte Arbeiten beziehen sich auf
  - Paravirtualisierung
  - Schedulability Analyse
- Integration mehrerer harter Echtzeit VMs auf ein System
  - Virtual Processor approach
  - Vollvirtualisierung
- Einsatz von Fixed Time Slice Scheduling zum Scheduling der VMs
  - Wahl des Parameters H

Thank you for your Attention

Open Problems

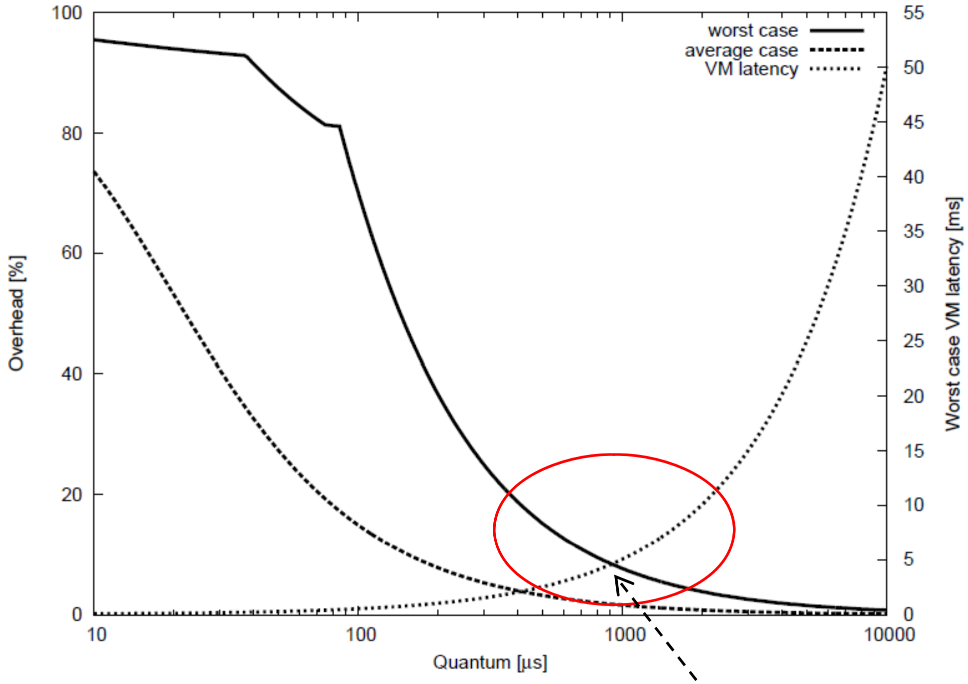
HEINZ NIXDORF INSTITUTE
University Paderborn
Design of Parallel Systems
Prof. Dr. rer. nat. Franz J. Rammig

$$H = ggT(\{T_t | t \in \tau'\})$$
$$\Rightarrow T_t = c \cdot H, c \in \mathbb{N}^+$$

- What if H becomes too small?
  - E.g H = 1?

- Small time slices introduce more context switching overhead.

- Long time slices will increase latency!



Good Solution Space

HEINZ NIXDORF INSTITUTE
University Paderborn
Design of Parallel Systems
Prof. Dr. rer. nat. Franz J. Rammig

Solving the H-Problem?!

Example:

$$VM_1 = \{(10,20),(8,23)\}$$
$$VM_2 = \{(20,40),(8,23)\}$$

H = 1

Possible: Increase speedup in order to allow higher values for H!

Examine Coefficient : Time Provided / Time Needed!

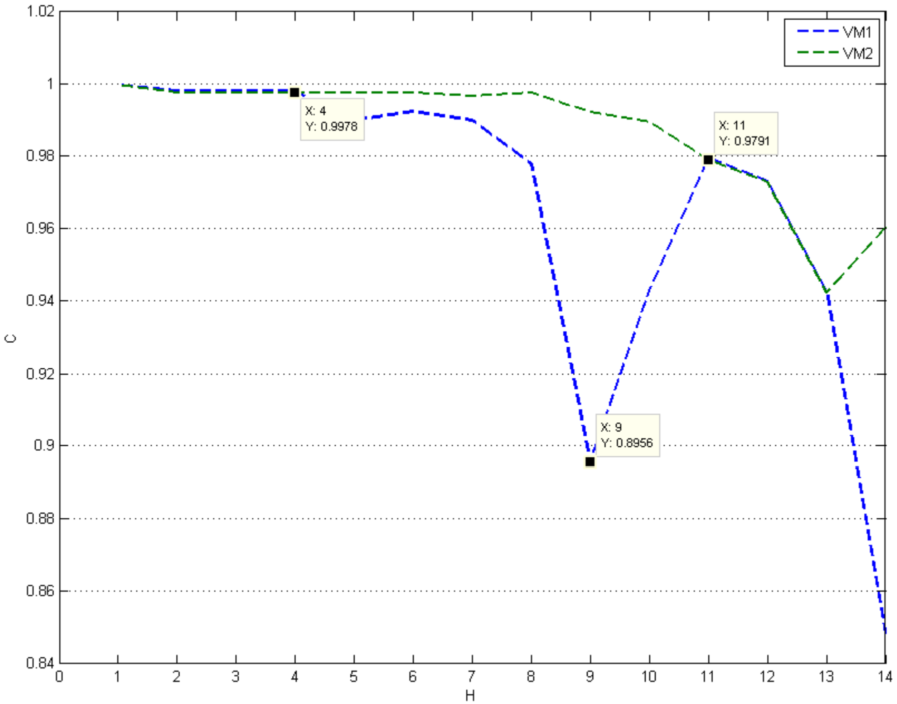
$$C(t)_i = min_{t \in T_{D_i}} \frac{Z(t)_i}{N(t)_i}$$

$$R(t)_i = t - \left\lfloor \frac{t}{H} \right\rfloor \cdot H - \delta_i$$

$$Z(t)_i = \left\lfloor \frac{t}{H} \right\rfloor \cdot Ts_i + 1(R(t)_i)_{[0\ldots\infty]} \cdot R(t)_i$$

$$N(t)_i = \sum_{\tau_j \in VM_i} \left\lfloor \frac{t}{T_j} \right\rfloor \cdot C_j$$

$$T_{D_i} = \{T_1, T_2, .., 2T_1, ..., lcm(\{T_1, T_2, ..., H\})\}$$



Open Problems

- What if Period != Deadline?
- How to solve the discretization problem?
- We can do better if we use Para-Virtualization?!? How?
- Dynamic Approaches?

Status Quo

HEINZ NIXDORF INSTITUTE
University Paderborn
Design of Parallel Systems
Prof. Dr. rer. nat. Franz J. Rammig

- Hybrid Real-Time Virtual Machine Monitor for PPC-E
  - Full-Virtualization and Para-Virtualization
  - High Performance Emulation

- ARM ISA Emulator

- Example Scenario: