

# Sicherheit, Virtualisierung und Einbruchstoleranz: Herausforderungen für zukünftige Systemsoftware

Hans P. Reiser

Fachgruppe Betriebssysteme  
Herbsttreffen 2009, Bommerholz

13. November 2009

# Überblick

## 1 Motivation

- Aktuelle Trends und Herausforderungen
- Sicherheit und Einbruchstoleranz

## 2 Virtualisierung als Chance

- Virtualisierung und Sicherheit

## 3 Sicherheit und Einbruchstoleranz mit Virtualisierung

- VM-FIT: Einbruchstoleranz mit virtuellen Maschinen
- Der farbige Desktop: Isolierte Sicherheitsdomänen

## 4 Zusammenfassung und Ausblick

# Motivation: Aktuelle Trends und Herausforderungen

## Aktuelle Trends:

- CPU: Singlecore → Multicore → Manycore
- Heterogen, energiesparend, selbstorganisierend
- Ubiquitäre Systeme, alles mit „dem Internet“ verbunden

## Gesucht: Geeignete System-Software

- Software muss „mithalten“ mit Hardware-Entwicklungen

# Motivation: Aktuelle Trends und Herausforderungen

## Aktuelle Trends:

- CPU: Singlecore → Multicore → Manycore
- Heterogen, energiesparend, selbstorganisierend
- Ubiquitäre Systeme, alles mit „dem Internet“ verbunden

## Gesucht: Geeignete System-Software

- Software muss „mithalten“ mit Hardware-Entwicklungen

## Größtes Problem heutzutage: Sicherheit

- Jede Woche: Duzende neue Schwachstellen in aktuellen Systemen (siehe NVD, Bugtraq & Co)
- Unzählige Angriffe und Einbrüche (Beispiel: Okt. 2005: Botnet mit 1,5 Millionen Rechnern entdeckt)

# Motivation: Sicherheit und Einbruchstoleranz

„Vorbeugung ist die beste Verteidigung“, aber ...

- *Schwachstellen* lassen sich nicht vollständig vermeiden
  - Ständig wachsende Komplexität
  - Entwicklungskosten

# Motivation: Sicherheit und Einbruchstoleranz

„Vorbeugung ist die beste Verteidigung“, aber ...

- *Schwachstellen* lassen sich nicht vollständig vermeiden
  - Ständig wachsende Komplexität
  - Entwicklungskosten
- *Angriffe* lassen sich noch weniger vermeiden
  - Ubiquitäre Vernetzung, „always on“
  - Wachsende Interaktionen

# Motivation: Sicherheit und Einbruchstoleranz

„Vorbeugung ist die beste Verteidigung“, aber ...

- *Schwachstellen* lassen sich nicht vollständig vermeiden
  - Ständig wachsende Komplexität
  - Entwicklungskosten
  
- *Angriffe* lassen sich noch weniger vermeiden
  - Ubiquitäre Vernetzung, „always on“
  - Wachsende Interaktionen

⇒ Konstruktion von Systemen, die Fehler und Einbrüche *tolerieren*

# Überblick

- 1 Motivation
  - Aktuelle Trends und Herausforderungen
  - Sicherheit und Einbruchstoleranz
- 2 **Virtualisierung als Chance**
  - **Virtualisierung und Sicherheit**
- 3 Sicherheit und Einbruchstoleranz mit Virtualisierung
  - VM-FIT: Einbruchstoleranz mit virtuellen Maschinen
  - Der farbige Desktop: Isolierte Sicherheitsdomänen
- 4 Zusammenfassung und Ausblick

# Virtualisierung ist heute (wieder) Mainstream...

## Desktop-Virtualisierung

- Mehrere Betriebssysteme, isolierte Testumgebungen, einfache Migration, etc.

## Server-Virtualisierung

- Ressourcen-Konsolidierung, weniger Energieverbrauch, Flexibilität, bessere Verwaltung

## Hardware-Unterstützung für Virtualisierung

- CPU
- I/O-MMU

# Virtualisierung: Eine Chance für sicherere Systeme

## Virtualization als „Enabling Technology“

Neuartige Sicherheitstechniken:

- Monitoring (Erkennen von Angriffen und Fehlern)
- Feingranulare Isolation von Komponenten (z.B. Treiber)
- **Einbruchstoleranz** (insbesondere von Diensten)
- **Der farbige Desktop**: Private Sicherheitsdomänen

# Überblick

- 1 Motivation
  - Aktuelle Trends und Herausforderungen
  - Sicherheit und Einbruchstoleranz
- 2 Virtualisierung als Chance
  - Virtualisierung und Sicherheit
- 3 Sicherheit und Einbruchstoleranz mit Virtualisierung
  - VM-FIT: Einbruchstoleranz mit virtuellen Maschinen
  - Der farbige Desktop: Isolierte Sicherheitsdomänen
- 4 Zusammenfassung und Ausblick

## **VM-FIT: Virtual Machine-based Fault and Intrusion Tolerance**

- Virtualisierungsbasierte Replikationsarchitektur
- Replikationskontrolle (im wesentlichen) außerhalb von Middleware/Anwendung

## **Hybrides Fehlermodell**

- Beliebige Fehler in Anwendungsdomänen
- Nur Crash-Fehler im restlichen System (VMM, Replikationsschicht)
- Unterstützung für zustandsbehaftete heterogene Replikate

## **VM-FIT: Virtual Machine-based Fault and Intrusion Tolerance**

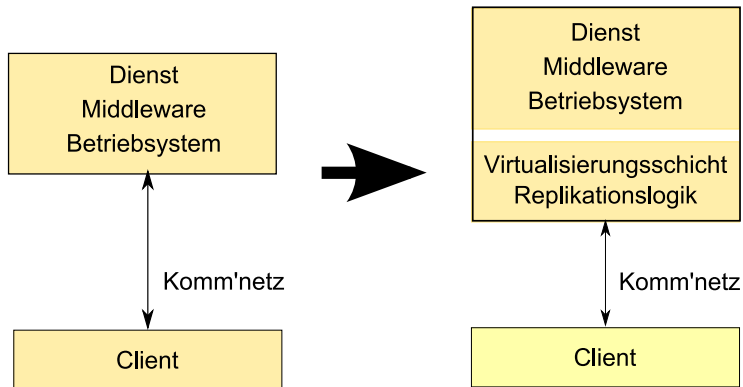
- Virtualisierungsbasierte Replikationsarchitektur
- Replikationskontrolle (im wesentlichen) außerhalb von Middleware/Anwendung

## **Hybrides Fehlermodell**

- Beliebige Fehler in Anwendungsdomänen
- Nur Crash-Fehler im restlichen System (VMM, Replikationsschicht)
- Unterstützung für zustandsbehaftete heterogene Replikate

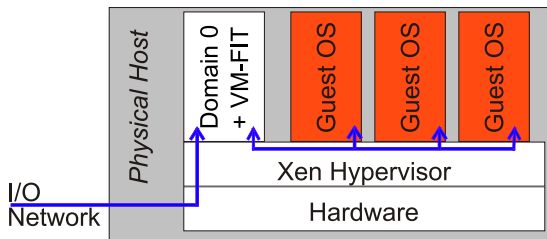
## **Softwarebasiertes „Wormhole“**

# VM-FIT: Architektur-Überblick



# VM-FIT: Variante 1: RESH

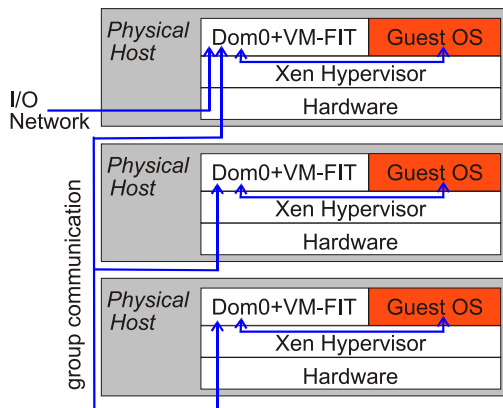
RESH: Redundant Execution on a Single Host



- Tolerierung von Einbrüchen in Gast-Domänen
- Opportunistische N-Versions-Programmierung auf nur einem Rechner
- Keine Komplet-Ausfälle des Rechners

# VM-FIT: Variante 2: REMH

REMH: Redundant Execution on Multiple Hosts



- Crash-Stop-Fehlermodell für Replikationslogik und Hypervisor
- Byzantinisches Fehlermodell für Anwendungsreplikat

# VM-FIT: Proaktives Recovery

Einbruchstolerante Systeme tolerieren eine begrenzte Anzahl von Fehlern

- Zuverlässiges Erkennen von Einbrüchen kaum möglich
- Zu viele Fehler: korrekte Funktion nicht mehr gewährleistet
- **Proaktives Recovery:** Periodisches Auffrischen aller Replikate

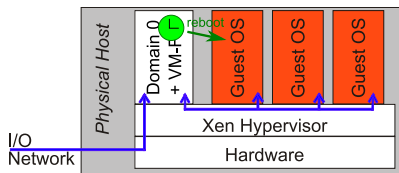
# VM-FIT: Proaktives Recovery

Einbruchstolerante Systeme tolerieren eine begrenzte Anzahl von Fehlern

- Zuverlässiges Erkennen von Einbrüchen kaum möglich
- Zu viele Fehler: korrekte Funktion nicht mehr gewährleistet
- **Proaktives Recovery:** Periodisches Auffrischen aller Replikate

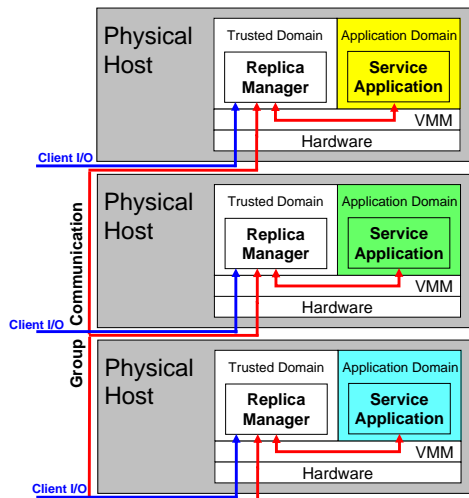
Proaktives Recovery in VM-FIT:

- Vertrauenswürdiger Trigger in Replikationslogik
- Geschützt vor mutwilligen Manipulationen
- Koordination von Recovery und Dienstausführung



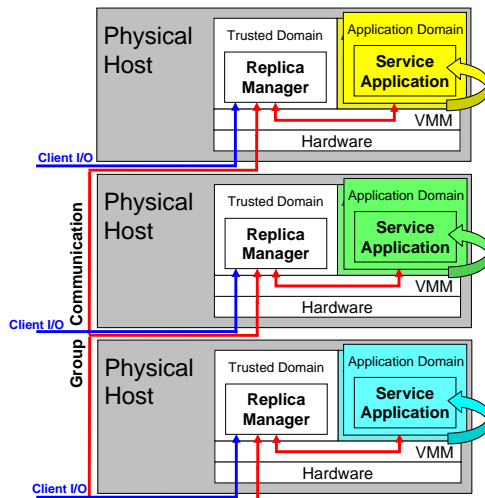
# VM-FIT: Verfügbarkeit bei proaktivem Recovery

- Diversität der Replikat-Software
  - Betriebssystem
  - Middleware
  - Dienst-Implementierung



# VM-FIT: Verfügbarkeit bei proaktivem Recovery

- Diversität der Replikat-Software
  - Betriebssystem
  - Middleware
  - Dienst-Implementierung
- Proaktives Recovery der Replikate
  - Austausch der Anwendungsdomänen
  - Simultanes Recovery aller Replikate
  - Zustandstransfer: Konvertierung zw. abstraktem und lokalem Format



# Diversität der Replikate

Diversität der Replikate notwendig für Einbruchstoleranz

- Gemeinsame Schwachstellen: alle Replikate können gleichzeitig angegriffen werden

FOREVER (RESIST NoE, 2008):

- Proaktives und reaktives Recovery, um Replikate zu verjüngen
- Fehlertoleranz durch Diversitäts-Maßnahmen

⇒ Was bringt uns dabei Virtualisierung?

# Diversität in Raum und Zeit

Diversität in Raum und Zeit:

- Raum: unterschiedliche Replikate mit unterschiedlichen Software-Versionen
- Zeit: Recovery von Replikaten wechselt zu anderer Software-Version

# Diversität in Raum und Zeit

Diversität in Raum und Zeit:

- Raum: unterschiedliche Replikate mit unterschiedlichen Software-Versionen
- Zeit: Recovery von Replikaten wechselt zu anderer Software-Version

Problem: Echte N-Versions-Programmierung zu teuer und unrealistisch

# Diversität in Raum und Zeit

Diversität in Raum und Zeit:

- Raum: unterschiedliche Replikate mit unterschiedlichen Software-Versionen
- Zeit: Recovery von Replikaten wechselt zu anderer Software-Version

Problem: Echte N-Versions-Programmierung zu teuer und unrealistisch

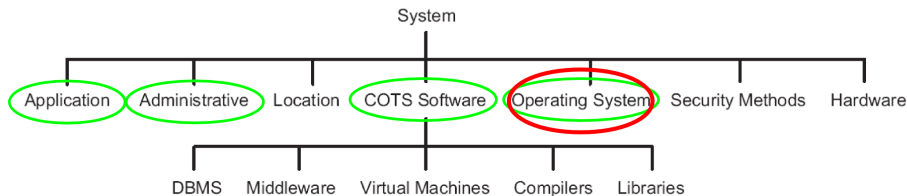
Diversitäts-Techniken in FOREVER:

- Diversität von COTS-Software (OS, JVM, Middleware)
- Erzeugte Diversität (Instruction-Set Randomization, Address-Space Randomization, Protokolle, Ports, Authentifizierungs-Methoden)

# Axen von Diversität

Zwei wesentliche Konzepte:

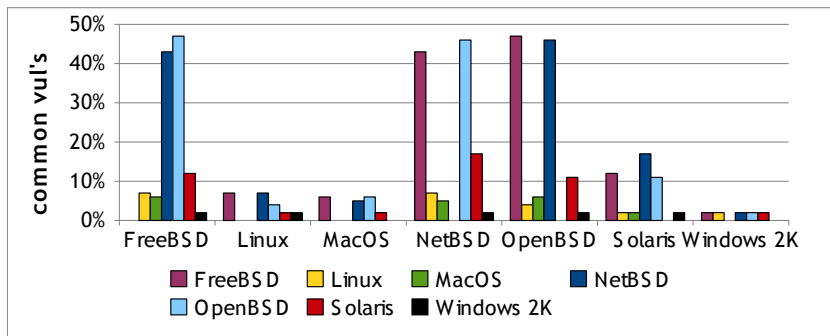
- Achsen von Diversität: Eine Systemkomponente, die „diversifiziert“ werden kann
- Grad an Diversität: Anzahl an Wahlmöglichkeiten für eine Diversitäts-Achse



# Diversitäts-Beispiel: Betriebssysteme

Haben COTS-Betriebssysteme gemeinsame Schwachstellen?

- Daten aus NVD extrahiert



# Diversität: Vorteile durch Virtualisierung

Diversität ist Voraussetzung für Einbruchstoleranz

FOREVER: COTS-Diversität, automatische Erzeugung von Diversität

Virtualisierung erlaubt Verwaltung der Diversität in Raum und Zeit

- Unterschiedliche Komponenten und Konfigurationen für jede virtuelle Maschine
- Individuell generierte Diversität für jede einzelne virtuelle Maschine

# VM-FIT: Messungen: Recovery mit Zustandstransfer

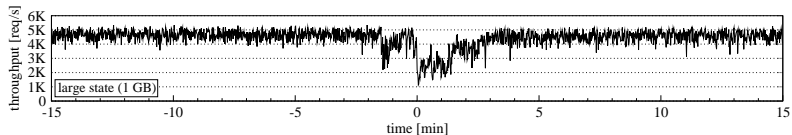
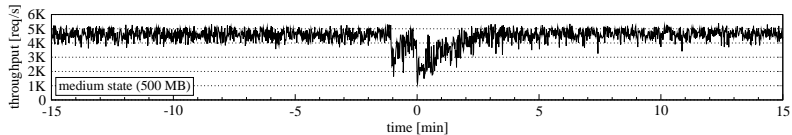
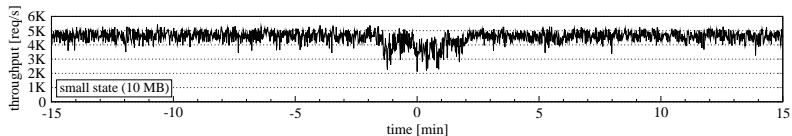
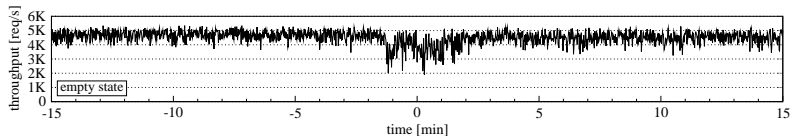
## Testbed

- 3 VM-FIT-Rechner, REMH, 1 Client-Rechner
  - 2.4 GHz Intel Core 2, 2 GB RAM, 1 Gb/s Ethernet
- Xen Version 3.1
- Heterogene Replikate (OS, Middleware)
  - OS: Linux, Net BSD and Open Solaris
- Proaktives Recovery alle 30 Minuten

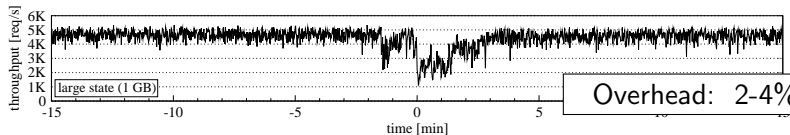
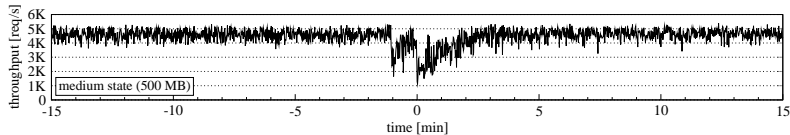
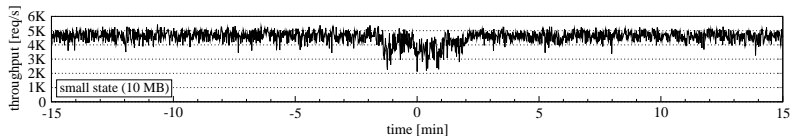
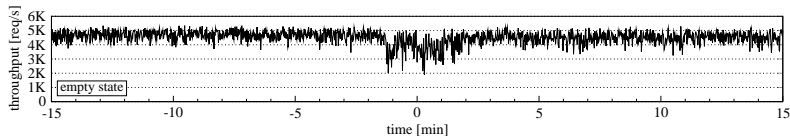
## 2 Messungen:

- „Micro-Benchmark“: Einfaches CGI-basiertes Telefonbuch
- RUBiS-Benchmarks

# VM-FIT: Messungen: Micro-Benchmark

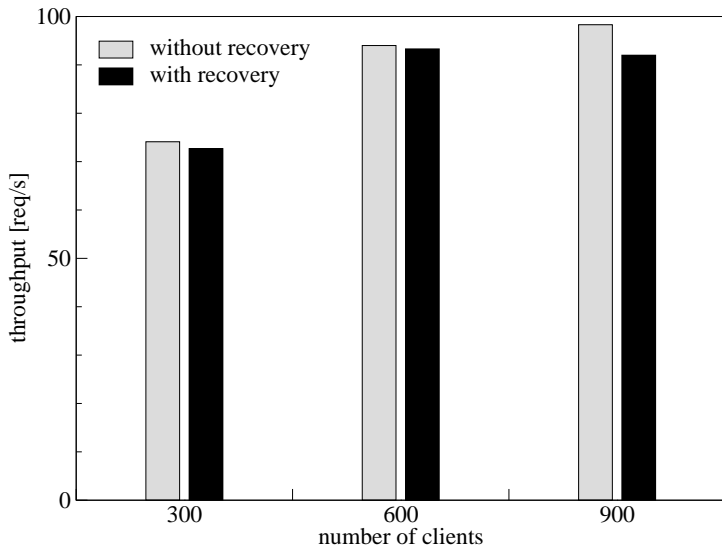


# VM-FIT: Messungen: Micro-Benchmark

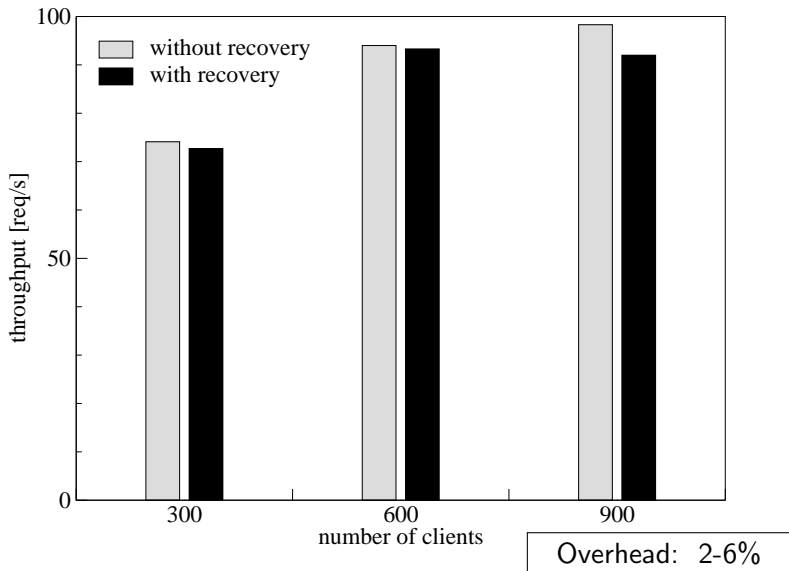


Overhead: 2-4%

# VM-FIT: Messungen: RUBiS-Benchmark



# VM-FIT: Messungen: RUBiS-Benchmark



## Proaktives Recovery: Vorteile durch Virtualisierung

- Rechtzeitiges Recovery ohne zusätzliche Spezialhardware
- Minimale Nichtverfügbarkeit durch paralleles Recovery
- Gemeinsames Recovery von allen Replikate
- Effizienter Virtualisierungs-gestützter Zustandstransfer

# Überblick

## 1 Motivation

- Aktuelle Trends und Herausforderungen
- Sicherheit und Einbruchstoleranz

## 2 Virtualisierung als Chance

- Virtualisierung und Sicherheit

## 3 Sicherheit und Einbruchstoleranz mit Virtualisierung

- VM-FIT: Einbruchstoleranz mit virtuellen Maschinen
- Der farbige Desktop: Isolierte Sicherheitsdomänen

## 4 Zusammenfassung und Ausblick

# Der farbige Desktop: Isolierte Sicherheitsdomänen

Desktop-Sicherheit mit Virtualisierungstechnik

- Laufende Master-Arbeit (João Ramos)

# Der farbige Desktop: Isolierte Sicherheitsdomänen

## Desktop-Sicherheit mit Virtualisierungstechnik

- Laufende Master-Arbeit (João Ramos)
- Arbeiten mit drei virtuellen Maschinen („rot“, „gelb“, „grün“)
  - Bestens geschützter Bereich (Online-Banking, etc.)  
Minimalsystem
  - Normaler Arbeitsbereich  
Kontrollierter Netzzugriff, nur ausgewählte vertrauenswürdige Anwendungen
  - Experimenteller Bereich  
Ausführung von nicht vertrauenswürdigen Code möglich

# Der farbige Desktop: Isolierte Sicherheitsdomänen

## Desktop-Sicherheit mit Virtualisierungstechnik

- Laufende Master-Arbeit (João Ramos)
- Arbeiten mit drei virtuellen Maschinen („rot“, „gelb“, „grün“)
  - Bestens geschützter Bereich (Online-Banking, etc.)  
Minimalsystem
  - Normaler Arbeitsbereich  
Kontrollierter Netzzugriff, nur ausgewählte vertrauenswürdige Anwendungen
  - Experimenteller Bereich  
Ausführung von nicht vertrauenswürdigen Code möglich
- Herausforderung: Usability
  - Einfaches konzeptionelles Modell, das der Nutzer versteht
  - Einfaches bewusstes Wechseln zwischen Farben
  - Problematik des Daten-Austausch

# Zusammenfassung und Ausblick

Virtualisierung: Eine Basis für sichere und einbruchstolerante Systeme

- Einbruchstolerante Dienste
- Sicherer Desktop

## Voraussetzung dafür:

- Vertrauenswürdiger Hypervisor
- **Herausforderung für Forschung im Bereich Systemsoftware**

Besorgniserregende Tendenz:

- Immer komplexere, fehleranfälliger Hypervisor
- Weg von „bare metal“ hin zu „hosted“ VMM

# Studenten gesucht

Ab Januar 2010

Studenten gesucht!

Themen im Kontext Virtualisierung und Systemsicherheit

- Master-Studenten: Stipendium für bis zu 9 Monaten (ca. 750 EUR/Monat)
- Doktoranden: Stipendium für 2 Jahre (ca. 1000 EUR/Monat)

Fragen? Kommentare?