

Jahresbericht 2010 der Gruppe „Betriebssysteme und Middleware“

<http://www.dcl.hpi.uni-potsdam.de/Jahresbericht2010.pdf>

1. Personelle Zusammensetzung

Leiter

Prof. Dr. rer. nat. habil. Andreas Polze
Tel.: (0331) 5509 231
email: andreas.polze@hpi.uni-potsdam.de

Sekretariat

Sabine Wagner
Tel: (0331) 5509 220
Fax: (0331) 5509 229
email: sabine.wagner@hpi.uni-potsdam.de

Wissenschaftliche Mitarbeiter

Dr. rer. nat. Martin von Löwis
Tel.: (0331) 5509 239
email: loewis@hpi.uni-potsdam.de

Dr. rer. nat. Peter Tröger
Tel.: (0331) 5509 234
email: peter.troeger@hpi.uni-potsdam.de

Dipl.-Inf. Bernhard Rabe
Tel.: (0331) 5509 236
email: bernhard.rabe@hpi.uni-potsdam.de

Dipl.-Inf. Alexander Schmidt
Tel.: (0331) 5509 238
email: alexander.schmidt@hpi.uni-potsdam.de

Michael Schöbel, M.Sc.
Tel.: (0331) 5509 109
email: michael.schoebel@hpi.uni-potsdam.de

Frank Feinbube, M.Sc.
Tel.: (0331) 5509 235
email: frank.feinbube@hpi.uni-potsdam.de

Dipl.-Ing. (FH) Uwe Hentschel, M.Sc.
Tel.: (0331) 5509 109
email: uwe.hentschel@hpi.uni-potsdam.de

Jan-Arne Sobania, M.Sc.
Tel.: (0331) 5509 237
email: jan-arne.sobania@hpi.uni-potsdam.de

Robert Wierschke, M.Sc.
Tel: (0331) 5509 214
email: robert.wierschke@hpi.uni-potsdam.de

Dipl.-Inf. Theodor Heinze
Tel: (0331) 5509 214
email: theodor.heinze@hpi.uni-potsdam.de

Dipl.-Inf. Alexander Schacht
Tel: (0331) 5509 233
email: alexander.schacht@hpi.uni-potsdam.de

Dipl.-Inf. Christian Neuhaus
Tel: (0331) 5509 215
email: christian.neuhaus@hpi.uni-potsdam.de

Unsere Mission

Moderne Betriebssysteme für Desktop und Server entwickeln sich in mehrere Richtungen. Einfache Benutzbarkeit, graphische Oberflächen, dynamische Erkennung von Geräten, Unterstützung verteilten Rechnens - dies sind Charakteristika moderner Desktop-Betriebssysteme wie Windows 7 oder Mac OS X. Im Server-Bereich finden Virtualisierung, die Nutzung von Cloud-Ressourcen und die Berücksichtigung von moderner NUMA-Hardware mit vielen Kernen immer weitere Verbreitung.

Noch rasanter entwickelt sich Middleware - eine Softwareschicht oberhalb der Betriebssystemebene. Häufig finden sich dort Dienste und Funktionen, die applikationsübergreifend benötigt werden und (noch) nicht Eingang in die Betriebssysteme gefunden haben. Im Laufe der Zeit werden Middleware-Dienste dann zu Bestandteilen der Betriebssysteme.

Handelsübliche Computer-Systeme (COTS - Commercial off-the-shelf) sind für hohe Leistung im lokalen Betrieb optimiert. Dies führt häufig zu unbefriedigendem Verhalten beim Einsatz in weit-verteilten Szenarien, bei kritischen Anforderungen an das Zeitverhalten oder im Fall von Hardware-Fehlern. Vorhersagbares und verlässliches Ende-zu-Ende Verhalten ist daher eine zentrale Forschungsaufgabe, die sich für Betriebssysteme und Middleware-Plattformen stellt.

Die Forschungsaktivitäten der Gruppe „Betriebssysteme und Middleware“ konzentrieren sich auf Paradigmen, Entwurfsmuster, und Implementierungsansätze für vorhersagbares Rechnen in Middleware-Umgebungen. Von zentralem Interesse ist dabei die Fragestellung, wie weit der Einsatz von Middleware-Technologie in der Domäne der eingebetteten und der verteilten Systeme die Vorhersagbarkeit unterstützen kann.

2. Lehrveranstaltungen

2.1. Vorlesungen

- Betriebssysteme I (WS 10/11)
- Betriebssysteme II (SS 10)
- Programmiertechnik I (WS 10/11)
- Programmiertechnik II (SS 10)
- Komponentenprogrammierung und Middleware (SS 10)
- Datenorientiertes XML (SS 10)
- Middleware und Verteilte Systeme (WS 10/11)
- Konzepte paralleler Programmierung (WS 10/11)
- Verlässliche Systeme (SS 10)

2.2. Übungen

- Betriebssysteme I (WS 10/11)
- Betriebssysteme II (SS 10)
- Programmiertechnik I (WS 10/11)
- Programmiertechnik II (SS 10)
- Datenorientiertes XML (SS 10)
- Middleware und Verteilte Systeme (WS 10/11)
- Konzepte paralleler Programmierung (WS 10/11)

2.3. Seminare/Praktika

- Analyse von Interpreterquelltext (SS 10)
- Forschungsseminar der Gruppe „Betriebssysteme und Middleware“
- Bachelorvorbereitungsseminar (WS 10/11)
- Studienbegleitendes Seminar/Seminarorganisation (WS 10/11)
- Verteilte Echtzeitsysteme (SS 10)
- Industrieseminar Cloud Computing (WS 10/11)
- Large-Scale Data Analysis on Cloud Platforms (SS 10, Kooperation Dr. Tröger / FG Informationssysteme)
- Large-Scale Processing for Semantic Web Technologies (WS 10/11, Kooperation Dr. Tröger / Dr. Sack)

2.4. Teleteaching

- Aufzeichnung der Veranstaltungen „Programmiertechnik II“ und „Datenorientiertes XML“

2.5. Unterricht

- Dr. Martin v. Löwis. Spiele mit vollständiger Information. Kurs der Schülerakademie des Hasso-Plattner-Instituts

3. Betreuung von Studierenden und Dissertationen

3.1. Betreuung von Bachelorprojekten

- Entwicklung eines Tools zur modellbasierten Spezifikation auf Basis von Eclipse und GEF/EMF, Bachelorprojekt mit der Firma Biotronik, Berlin (SS 10)
- Entwicklung einer Telemedizinplattform bei der GETEMED AG (WS 10/11)

3.2. Betreuung von Masterarbeiten

- Fabian Garagnon, „Integration eines Quellcode-Verwaltungssystems in einen Funktionsblockeditor (P80i)“
- Erik Schleiff, „Memory Hot Plug and Virtual Machines“

3.3. Betreuung von Dissertationen (intern, extern)

- Bernhard Rabe – Programmoptimierung für eingebettete Systeme
- Michael Schöbel – Verarbeitung von Ereignisströmen im Betriebssystemkern, im Forschungskolleg „Service-Oriented Systems Engineering“
- Alexander Schmidt – Ressourcenverwaltung in Betriebssystemen, im Forschungskolleg „Service-Oriented Systems Engineering“
- Frank Feinbube – Programmiermodelle für Multicore Computer, im Forschungskolleg „Service-Oriented Systems Engineering“
- Uwe Hentschel – Services for Real-Time Computing, im Forschungskolleg „Service-Oriented Systems Engineering“
- Jan-Arne Sobania – Statische Analyse von Programmen für eingebettete Systeme, im Forschungskolleg „Service-Oriented Systems Engineering“
- Alexander Schacht – „Echtzeitsysteme in der Telemedizin“, im Verbundprojekt „SaPiMa/PhysioGate“ / FONTANE
- Robert Wierschke – „Kommunikationsmiddleware für telemedizinische Systeme“, im Verbundprojekt „SaPiMa/PhysioGate“ / FONTANE
- Theodor Heinze – „Regelbasierte Entscheidungsunterstützung in der Telemedizin“, im Verbundprojekt „SaPiMa/PhysioGate“ / FONTANE
- Christian Neuhaus – „Secure Cloud-based Data-Store“, im DFG-Graduiertenkolleg SOAMED

4. Bearbeitete Forschungsthemen

4.1. Micro.NET

Ansprechpartner: Dipl.-Inf. Bernhard Rabe

In der Softwareentwicklung für eingebettete Systeme haben lange Zeit hardwarenahe Programmiersprachen dominiert. Virtuelle Ausführungsumgebungen wie Java VM und die Common Language Infrastructure (CLI) haben die Softwareentwicklung durch automatische Speicherverwaltung, Typsicherheit, Ausnahmebehandlung und Plattformunabhängigkeit vereinfacht. Im Rahmen des Micro.NET Projektes werden Techniken untersucht und entwickelt um .NET Technologie an die Anforderungen eingebetteter Systeme anzupassen. Betrachtet werden insbesondere Systeme deren spezielle Eigenschaften und Anforderungen von bestehende CLI Implementierungen (.NET, .NET CF, .NET MF, Mono) nicht berücksichtigt werden. Mit dem entstandenen Bytecode-Interpreter für die Lego Mindstorm Plattform konnte die Ausführbarkeit von .NET auf ressourcenbeschränkten eingebetteten Systemen gezeigt werden. Um Speicheranforderungen von .NET Programmen zu minimieren wurde das Konzept der self-contained Assemblies entwickelt. Damit ist es möglich die lose Kopplung zwischen .NET Programmen und Bibliotheken zugunsten eines geringeren Speicherplatzbedarfs aufzuheben, indem alle benötigten Bibliotheksfunktionen in das Programm integriert werden. Basierend auf dem ECMA 335 Standards werden Methoden zur Spezifikation und Entwicklung einer minimalen Laufzeitumgebung für Self-contained Assemblies untersucht.

4.2. Windows Research Kernel / Windows Monitoring Kernel

Ansprechpartner: Michael Schöbel, M.Sc., Dipl.-Inf. Alexander Schmidt

Unter dem Namen „Windows Operating System Internals Curriculum Resource Kit“ (CRK) entstand 2005 eine Lehrmaterialsammlung zum Thema Betriebssystemarchitektur. Das CRK ist eng mit dem „Windows Research Kernel“ (WRK) verknüpft, einer Plattform, die Modifikationen am Windows-Kern ermöglicht.

Forschungsfragen, wie neue Scheduling-Algorithmen und Bandbreiten-Reservierungen im Ein-/Ausgabesystem lassen sich so untersuchen. Dazu entstanden verschiedene Testanwendungen und Kernmodifikationen. Außerdem wurde das Open-Source Werkzeug Doxygen zur Aufbereitung des Quellcodes verwendet.

Auf Basis des Windows Research Kernels wurde ein Framework entwickelt, das eine Black-Box-Analyse von Anwendungen erlaubt. Dazu wurde der Windows-Kern um ein effizientes Event-Log erweitert und geeignet instrumentiert. Außerdem wurden neue Schnittstellen eingeführt, mit deren Hilfe auch Ereignisse in verschiedenen Bibliotheken aufgezeichnet werden können.

4.3. KStruct – Live Kernel Data Inspection

Ansprechpartner: Dipl.-Inf. Alexander Schmidt

Beteiligt: Jan-Arne Sobania, M.Sc., Lysann Kessler, Frank Schlegel

Im Mittelpunkt des Projekts steht der Versuch, den Prozess für das Verstehen eines komplexen Softwaresystems zu vereinfachen. Ein Entwickler verfügt typischerweise über den

Quellcode sowie die Dokumentation des Softwaresystems, in das er sich einarbeiten möchte. Dies reicht jedoch nicht aus, um das System gänzlich zu verstehen. Vielmehr wäre es wünschenswert, den Zustand des Systems, d.h. Exemplare verwendeter Datenstrukturen, zur Laufzeit zu beobachten, um daraus weitere Schlüsse zu ziehen. Eine solche Beobachtung kann beispielsweise durch die Verwendung von Debuggern oder Instrumentierung erfolgen. Als Beispiel für ein solches komplexes Softwaresystem betrachten wir den Windows Research Kernel.

Die Idee hinter KStruct ist es, Datenstrukturen, die beobachtet werden sollen, semantisch so zu beschreiben, dass ein Treiber generiert werden kann, der ein Exemplar der Datenstruktur lokalisieren und sicher referenzieren kann. Die Beschreibung der Datenstrukturen erfolgt mit einer C-ähnlichen Syntax.

4.4. Eingebettete Systeme im Distributed Control Lab

Ansprechpartner: Dipl.-Inf. Bernhard Rabe

Im Distributed Control Lab wird die Verbindung von Middleware-basierten Komponenten und eingebetteten Systemen untersucht. Ziel ist es vorhersagbares Systemverhalten in instabilen Umgebungen zu erreichen. Zur Untersuchung von verschiedenen Ansätzen sind Fallstudien in Form von Experimenten umgesetzt, die sich über das Internet steuern lassen. Im Vordergrund stehen dabei die nichtfunktionalen Eigenschaften, wie z.B.: Echtzeitverhalten, Fehlertoleranz und Sicherheit.

Der Schutz der Experimente (Hardware) vor potentiell fehlerhaften Kontrollalgorithmen aus dem Internet wird abhängig von den Anforderungen der Experimente mittels Soft- bzw. Hardwaremechanismen realisiert.

Als wichtiger Pfeiler für praktische Erfahrungen floss die verteilte Laborinfrastruktur in die Lehrveranstaltung „Betriebssysteme für Embedded Computing“ ein. Dabei wurden in verschiedenen Praktikumsveranstaltungen die Steuerung eines Intelligenten Hauses mit Beckhoff Soft-SPS (speicherprogrammierbare Steuerungen) realisiert, die Eignung der .NET Plattform für eingebettete Geräte (.NET Compact Framework) wurde an Hand des „Hau-den-Lukas“-Experiments evaluiert. Auch die Steuerung einer komplexen Fischertechnik Fertigungsstraße mit Hilfe von industriellen Steuerungssystemen wurde im Rahmen eines Bachelorprojekts in das Distributed Control Lab integriert und verschiedene Sicherheitsstrategien implementiert. Zusätzlich kam die Laborinfrastruktur als Lernumgebung für eine Veranstaltung zum Thema „Programmierung eingebetteter Systeme“ an der Blekinge Tekniska Högskola in Ronneby, Schweden zum Einsatz. Studenten der Veranstaltung führten physikalisch entfernte Experimente am Hasso-Plattner-Institut als Praktikumsaufgaben durch, um das in der Vorlesung erworbene Wissen zu vertiefen.

Im EU-Projekt Vet-Trend wird untersucht wie, die Laborinfrastruktur in weitere Veranstaltungen auf europäischer Ebene integriert werden kann und auch wie existierende Infrastrukturen der Projektpartner am HPI nutzbar gemacht werden können. Zusätzlich umfasst das Projekt Standardisierungsbestrebungen auf dem Gebiet der verteilten und virtuellen Labore

4.5. Real-Time.Net

Ansprechpartner: Dr. Martin v. Löwis

Beteiligt: Sven Widmer

Virtuelle Laufzeitumgebung wie Java und Microsoft .NET haben die Entwicklung von Software in den letzten Jahren verbessert. Automatische Speicherverwaltung, erhöhte Typsicherheit und umfangreiche Klassenbibliotheken reduzierten die Entwicklungszeit von Softwaresystemen. Diese Vorteile sind auch für die Entwicklung von eingebetteten Systemen wünschenswert. Um eine deterministische Ausführung von Programmcode und die Implementierung von Hardware-nahen Treibern zu ermöglichen, müssen die vorhandenen Ansätze erweitert werden. Im Projekt wird die Integration solcher Konstrukte in die .NET Plattform untersucht. Dazu zählen eine deterministische Speicherverwaltung, uneingeschränkter Hardwarezugriff, Unterbrechungsbehandlung, zeitbasierende Programmausführung sowie deterministische Ausführung von Programmabschnitten selbst.

Ausgehend vom Lego.Net Compiler-Ansatz wurde im Projekt ein Laufzeitsystem für das Echtzeitbetriebssystem Windows CE.NET implementiert und an Experimenten im Distributed Control Lab evaluiert.

4.6. IPv6 Testbed

Ansprechpartner: Dr. Martin von Löwis

Nach verschiedenen Schätzungen von APNIC, ARIN, LACNIC und RIPE aus dem Jahr 2007 werden die regionalen Registrierungen ihren Vorrat an IPv4-Adressen im Laufe des Jahres 2010 erschöpfen. Gleichzeitig gibt es aber keine ausreichenden Bestrebungen, Internetnutzern den Zugang zum IPv6-Wirknetz bereitzustellen, und kein ausreichendes Interesse dieser Nutzer, von bestehenden IPv6-Angeboten Gebrauch zu machen.

Ziel des IPv6-Testbeds am HPI ist, Studenten, Lehrkräften und Technikern eine realistische Umgebung bereitzustellen, um die Auswirkungen von IPv6 auf Netzbetrieb und Softwareentwicklung zu studieren. Dieses Netz ist an das globale IPv6-Netz angeschlossen, so dass praxisnahe Arbeiten mit IPv6 und den dazugehörigen Technologien und Protokollen möglich ist.

4.7. PXR – Ein HTML-basierter Dokumentations-Compiler

Ansprechpartner: Dipl.-Inf. Alexander Schmidt

Beteiligte: Lysann Kessler, Frank Schlegel

Für das Verstehen von Quellcode in beliebig komplexen Softwaresystemen ist es hilfreich, den Quellcode strukturiert aufzubereiten und durch zusätzliche Informationen anzureichern. Wird der Quellcode im Rahmen eines Teams analysiert, bspw. im Rahmen einer Vorlesung, stellt sich des Weiteren das Problem der Verteilung der Quellen.

Dem PXR-Projekt liegt die Idee zu Grunde, den Quellcode des Windows Research Kernels in HTML umzuwandeln auf einem Webserver HPI-intern verfügbar zu machen. Ein ähnlicher Ansatz ist bereits in Tools wie Doxygen oder LXR zu finden. Der Vorteil der Verwendung von HTML liegt darin, dass ein Nutzer sich natürlich durch die Quellen navigieren kann, in dem er Querverweisen folgt. Bei den genannten Tools ist jedoch lediglich ein (C-) Parser im Einsatz, was mit unter zu ungültigen Verweisen oder Ungenauigkeiten führt. PXR erweitert diesen Ansatz daher, indem zusätzlich Informationen aus dem Compiler extrahiert und in den HTML-Code einbettet. So lässt sich beispielsweise genau herausfinden, an welcher Stelle im

Code auf eine bestimmte Variable zugegriffen wird oder an welcher Stelle einer Funktion Schreib- bzw. Lesezugriffe auf eine Variable erfolgen.

4.8. Serverbetriebssystem-Labor

Ansprechpartner: Dr. Martin von Löwis, Dipl.-Inf. Bernhard Rabe

An Serverbetriebssysteme werden höhere Anforderung an Fehlertoleranz, Verfügbarkeit, Sicherheit und Skalierbarkeit gestellt im Vergleich zu Desktop-Betriebssysteme. In diesem Zusammenhang untersuchen wir Techniken für effiziente Betriebsmittelverwaltung, Sicherheitskonzepte und Hochverfügbarkeitsansätze. Virtualisierung ist ein Ansatz in Serverbetriebssystemen um Betriebsmittel besser auszunutzen, Isolation von Betriebssysteminstanzen zu realisieren und Betriebsmittelbenutzung zu regeln. Wir betreiben Virtualisierungslösungen von Microsoft (Hyper-V), Hewlett-Packard (HP-VM), VMWare (ESX), Sun (Solaris Zones) und Linux 2.6 (KVM) auf x64, IA64 und Sparc-Architektur.

Für Fehlertoleranz- und Hochverfügbarkeitsuntersuchungen betreiben wir mehrere OpenVMS-Cluster auf verschiedenen Architekturen (IA64, AXP, VAX). Serverbetriebssystemkonzepte sind Bestandteil in Forschung und Lehre und werden in den Vorlesungen Betriebssysteme I+II und Serverbetriebssysteme vermittelt, sowie für tiefgehenden Untersuchungen im Rahmen von Masterarbeiten behandelt.

4.9. Ressourcenmanagement in Betriebssystemen

Ansprechpartner: Frank Feinbube, M.Sc.

Beteiligte: Fabian Bornhofen, Thomas Bünger

Das Ressourcenmanagement von Betriebssystemen ist traditionell auf die Prozessabstraktion als Basisgröße für die Granularität ausgerichtet. Die simple Lösung einen Prozess (oder eine virtuelle Maschine) einem Prozessorkern zuzuweisen muss mit dem Aufkommen von Mehrkernarchitekturen und der Unterstützung von Hyperthreading neu beurteilt werden. Wir erforschen die Begrenzung des Threadmanagements auf diversen Betriebssystemen (Windows Server 2003, Windows Server 2008 R2, HP-UX, Solaris, Ubuntu). Neben der Anzahl von Threads, die ein Betriebssystem unterstützt, werden auch die Fragen nach dem Einfluss auf den verwendeten Hauptspeicher, die Zeit zur Threaderzeugung und der Einfluss auf den Schedulingoverhead untersucht.

Bei der Diskussion der Architektur komplexer Softwaresysteme in Hinblick auf die Verwendung von Threads oder Prozessen als grundlegende Mechanismen der Strukturierung müssen auch weitere Qualitätsmerkmale wie Zugriffsschutz, Isolation und Fehlereingrenzung berücksichtigt werden. Weiterhin untersuchen wir inwiefern strukturierte Ausnahmebehandlung (SEH) unter Windows als ein Mechanismus verwendet werden kann, um in einem Thread-basierten System einen Zugriffsschutz und eine Isolation zu erreichen, die mit denen von traditionellen Prozess-basierten System vergleichbar ist.

4.10. Fontane – Gesundheitsregionen der Zukunft / Nordbrandenburg

Ansprechpartner: Dr. Martin von Löwis, Dipl.-Inf. Theodor Heinze, Uwe Hentschel, M.Sc., Dipl.-Inf. Alexander Schacht, Robert Wierschke, M.Sc.

Das Fontane-Projekt konzentriert sich auf die Fernüberwachung und die Nachsorge für Schlaganfall-Patienten und Patienten mit Herzkrankheiten im ländlichen Raum von Nordbrandenburg. Das Projekt wird von einem Konsortium von 20 Partnern durchgeführt, darunter die Charité, die größte Universitätsklinik in Deutschland, Hersteller von medizinischen

Geräten, Ärzten und Herz-Spezialisten. Für Fontane entwickelt und implementiert die Fachgruppe „Betriebssysteme und Middleware“ des HPI eine selbstadaptive, priorisierende Middleware (SaPiMa), die auf zuvor entwickelten Konzepten von objektbasierten Echtzeitsystemen aufbaut.

Das Fontane-Projekt baut auf den Erfahrungen des Vorgänger-Projektes „Partnership for the Heart“ auf, das das Problem der Überwachung von 300 Patienten im Großraum Berlin gelöst hat. Im Gegensatz zu dem Vorgänger-Projekt und mit mehr als 1000 teilnehmenden Patienten, wird Fontane mit Fragen der Skalierbarkeit konfrontiert werden, die neue Ansätze für die Priorisierung der Übertragung und Darstellung von Daten erfordern. Ein weiteres Novum der von uns vorgeschlagene Architektur wird ihre allgemeine Gültigkeit sein: Dank ihrer Selbstadaptivität und ihrer regelbasierten inneren Abläufe wird sie nicht nur zur Überwachung von Herz-Patienten, sondern auch zur Versorgung von Patienten mit anderen Krankheiten geeignet sein.

4.10.1. SaPiMa Testbed für Fontane

Zur Evaluierung der entwickelten Konzepte wurde eine Testumgebung für das „Fontane-System“ eingerichtet. Das Testbed umfasst ein prototypisches Telemedizinzentrum, PhysioGate-Ersatz und PhysioGate-Light sowie verschiedene medizinische Messgeräte (Elektrokardiogramm, Blutdruckmessgerät, Waage). Das Testbed ermöglicht die vollständige Simulation der Fontane-Szenarien: Übermittlung der Vitalparameter von den Messgeräten, Verwaltung der patientenseitigen Kommunikation über das PhysioGate, Analyse und Verwaltung der Patienten- sowie Messdaten in einer elektronischen Patientenakte.

4.10.2. Live EKG Streaming für Fontane

Die Priorisierung der Übertragung von Daten im Fontane Projekt wird in diesem Teilprojekt untersucht. Die Übertragung von einfachen Vitalparametern (Bspw. Gewicht) mit Hilfe von Mobilfunk ist selbst im ländlichen Raum ohne größere Probleme zu realisieren. Die Übertragung von datenintensiven medizinischen Messungen (Bspw. Elektrokardiogramm (EKG)) ist unter Umständen nicht ohne weiteres möglich, insbesondere wenn mehrere Teilnehmer (Patienten) solche Messungen durchführen. Als erste Machbarkeitsstudie wurde ein Experiment aufgesetzt welches, das EKG von mehreren Läufern beim Berlin- und Bonnmarathon aufzeichnet und live in die Charité nach Berlin überträgt. Die in der Charité empfangenden EKGs wurden von Experten überwacht. Das Experiment war eine Kooperation mit den Partnern aus dem Fontane Projekt.

4.10.3. Priorisierung von Patientendaten für Fontane

Die Konzeption von SaPiMa ermöglicht eine effiziente tägliche Übertragung von nichtinvasiv gemessenen Vitalparametern der telemedizinisch betreuten Patienten zum Telemedizinzentrum, wo die Daten dann vom medizinischen Personal begutachtet werden. Dabei ist es sehr wichtig die Reihenfolge der Begutachtung der Patientendaten von dem voraussichtlichen Gesundheitszustand abhängig zu machen, so dass auffällige Patientendaten zuerst begutachtet werden. Die automatische Einstufung der Patientendaten und ihre Sortierung nach Auffälligkeit, die Priorisierung, wird durch SaPiMa automatisch durchgeführt. Dabei werden historische Daten, die bereits von Medizinern klassifiziert worden sind, von SaPiMa mit Machine Learning und Data Mining Techniken analysiert und das extrahierte Wissen dann verwendet, um noch nicht klassifizierte Daten genau einzustufen. Mediziner haben dann bei Bedarf noch die Möglichkeit die Klassifikation durch manuell festgelegte Regeln zu ergänzen. Die Priorisierung wurde in mehreren Prototypen im SaPiMa Testbed eingesetzt. Ende 2010 wurde ausserdem die Publikation „Hybrid Artificial Intelligence System for Remote Patient Monitoring in Telemedicine“ angefangen.

4.11. InstantLab – Betriebssystemexperimente in der Cloud

Ansprechpartner: Dipl.-Inf. Alexander Schmidt

Experimente sind ein wesentlicher Bestandteil der Lehre um Prinzipien besser zu verstehen und zu analysieren. Gerade im Kontext der Betriebssystemlehre ist der Versuchsaufbau für ein Experiment mitunter sehr zeitintensiv: Soll ein bestimmtes Phänomen aus dem Betriebssystemkern näher untersucht werden, muss u.U. eine spezielle Betriebssystemversion zusammen mit Lastgeneratoren und Messwerkzeugen installiert und entsprechend konfiguriert werden. Dieser Zusatzaufwand überlagert dann nicht selten den Fokus auf das eigentliche Problem, das untersucht werden soll. Zudem stellt ein Betriebssystemexperiment stark einschränkende Anforderungen an die Ausführungsumgebung.

Im Rahmen des InstantLab-Projektes soll untersucht werden, inwieweit sich Cloud Computing-Umgebungen, z.Bsp. Windows Azure oder Amazon EC2, für die Ausführung solcher Experimente eignen. Dabei werden vorkonfigurierte Labore, d.h. virtuelle Maschinen, die einen kompletten Versuchsaufbau enthalten, mit Hilfe der Cloud-Infrastruktur zur Ausführung gebracht, die sich dann vom Nutzer grafisch fernsteuern lassen. Dazu wird eine prototypische Implementierung entwickelt, mit deren Hilfe sich weitere Fragestellungen, z.Bsp. Performanzeigenschaften unterschiedlicher Cloud-Anbieter, effektive Kostenverwaltung und -überwachung für die Ausführung eines Labors, etc. untersuchen lassen.

4.12. Intel-Single Chip Cloud Computer (SCC)

Ansprechpartner: Jan-Arne Sobania, M.Sc.

Der Intel Single-Chip Cloud Computer (SCC) ist ein experimenteller Prozessor, der von Intel Labs Bangalore in Indien, Intel Labs Braunschweig in Deutschland sowie Intel Labs in den Vereinigten Staaten von Amerika entwickelt wurde. Er enthält mit 48 Kernen die bisher größte Anzahl von x86-Recheneinheiten, die jemals auf einem einzelnen Silizium-Chip erreicht wurde. Anstelle eines aus herkömmlichen PCs bekannten Bussystems verwendet der Chip ein 2D-Mesh-Netzwerk zur Kommunikation der Kerne untereinander und mit den Speichercontrollern, und enthält Hardware-Unterstützung für Nachrichtenaustausch.

Als „Concept Vehicle“ ist der SCC weder als Produkt noch als Prototyp für ein Produkt geplant, sondern als Forschungsplattform für künftige Multi- und Many-Core-Landschaften. Grundlegende Aspekte wie Nachrichtenaustausch anstelle eines gemeinsamen Speichers sind aus dem Bereich des High-Performance-Computing bekannt, und mit dem SCC besteht die Möglichkeit zu untersuchen, inwieweit sich eine solche Architektur auch für herkömmliche Anwendungen und Betriebssysteme eignet.

Zu diesem Zweck entwickelt das Fachgebiet ein Single-System-Image-Betriebssystem (SSI) auf Basis eines verteilten Virtual Machine Monitors (VMMs).

4.13. Fehlerinjektion durch Firmware

Ansprechpartner: Dr. Peter Tröger

In Zusammenarbeit mit Fujitsu Technology Solutions (FTS) beschäftigt sich dieses Forschungsprojekt mit der Frage, wie software-basierte Fehlerinjektion in der Firmware von Computer-Systemen realisiert werden kann. Die Grundlage dafür bildet das *Unified Extensible Firmware Interface (UEFI)*, eine Weiterentwicklung des klassischen BIOS von namhaften Hardware-Herstellern, welches das Nachladen von Code in der Firmware

ermöglicht. Durch diesen neuartigen Ansatz können zusätzliche Fehlerarten emuliert werden. Zudem werden bisher nicht realisierbare Verlässlichkeitsstudien (z.B. zu Hypervisor-Technologien) ermöglicht. FTS kooperiert mit der Arbeitsgruppe bei technischen Fragestellungen zu UEFI-Technologie, und evaluiert den entstandenen Prototypen für den Einsatz bei der Qualitätssicherung.

4.14. Analyse von Fehler-Logs auf Mainframe-Systemen

Ansprechpartner: Dr. Peter Tröger

Das Fachgebiet kooperiert mit verschiedenen Abteilungen der IBM Labs Böblingen zu Forschungsfragen im Bereich der Verlässlichkeit und Skalierbarkeit von parallelen und verteilten Systemen. Das Themenspektrum umfasst einerseits RAS-Analysen für aktuelle z/Series – Systeme, andererseits aber auch die Parallelisierung von Datenverarbeitungsaufgaben in der IBM Public Cloud.

4.15. HPI Future SOC Lab

Ansprechpartner: Prof. Andreas Polze, Dipl.-Inf. Bernhard Rabe

Beteiligte: Stefan C. Richter

In Kooperation mit Partnern aus der Industrie etabliert das Hasso-Plattner-Institut (HPI) das “HPI Future SOC Lab”, das eine komplette Infrastruktur von hochkomplexen on-demand Systemen auf neuester, am Markt noch nicht verfügbarer, massiv paralleler (multi-/many-core) Hardware mit enormen Hauptspeicherkapazitäten und dafür konzipierte Software bereitstellt.

Das HPI Future SOC Lab stellt somit eine sonst im Hochschulbereich kaum finanzierbare experimentelle Basis zur Verfügung, um Forschungen rund um innovative Konzepte für zukünftige IT-Systeme auch außerhalb geschlossener Industrielabs zu ermöglichen. Interessierte Wissenschaftler aus universitären und außeruniversitären Forschungsinstitutionen können im HPI Future SOC Lab zukünftige hoch-komplexe IT-Systeme untersuchen, neue Ideen / Datenstrukturen / Algorithmen entwickeln und bis hin zur praktischen Erprobung verfolgen. Sie sind eingeladen (Call for Projects), sich um die Nutzung des Labs und seiner Hochleistungsrechner für ihre eigene Forschung zu bewerben. Die ausgewählten Forscher können die Ihnen zugewiesenen Lab-Ressourcen für eine bestimmte Zeit kostenlos nutzen.

4.16. *Hybrid-Parallel-Library*

Ansprechpartner: Frank Feinbube, M.Sc.; Jan-Arne Sobania, M.Sc.

Um die vielen rechenintensiven Probleme zu lösen, mit denen wir heutzutage konfrontiert sind, wird enorme Rechenleistung benötigt. Prozessoren werden jedoch nicht mehr – wie bisher – von allein schneller, sondern bekommen stattdessen immer mehr Kerne. Die Vielzahl an Kernen erschwert es nicht nur den Programmierern effiziente Anwendungen zu schreiben, sondern stellt auch Hardware-Designer vor neue Herausforderungen. Neuartige Hardware-Architekturen werden konzipiert und evaluiert, um Wege zu finden, die Bedürfnisse zukünftiger zufriedenstellend Computersysteme zu erfüllen. Prototypen wie Intels Single-Chip Cloud Computer (SCC) sind Versuche, die ständig steigende Anzahl an Kernen zu bewältigen, indem man sich von wesentlichen Merkmalen aktueller Prozessortypen – wie z.B. Hardware-Cache-Kohärenz – trennt. Ein weiterer Ansatz ist es, gewöhnlichen Allzweck-

CPUs, ausgefeilte Spezialprozessoren zur Seite zu stellen. Diese sogenannten Beschleuniger (Accelerators) sind einfacher herzustellen und liefern eine hohe Performanz für spezifische Anwendungszwecke. Sie sind die Grundlage für den neuen Trend der Hybrid-Computersysteme.

Programmiermodelle und -Werkzeuge für den Bereich der Hybrid-Systeme haben nicht nur die Probleme der Entwicklung für parallele Systeme zu bewältigen, sondern auch die der unterschiedlichen Ausführungsmerkmale von Prozessoren und Beschleunigern in einer gegebenen Systemkonfiguration.

Ziel des Hybrid-Parallel-Library-Projektes ist es Entwicklern, die Hochsprachen wie .NET oder Java verwenden, intelligente Werkzeuge und Softwarebibliotheken zur Verfügung zu stellen, mit denen es ihnen einfach möglich ist, Anwendungen zu erstellen, welche das große Performanzpotential moderner Beschleunigertechnologien ausschöpfen.

4.17. Sicherer Austausch von Patientendaten über Cloud-Infrastrukturen

Ansprechpartner: Dipl.-Inf. Christian Neuhaus

An der Behandlung eines Patienten ist üblicherweise eine Vielzahl von Institutionen beteiligt, wie z.B. Krankenkasse, Hausarzt, Krankenhaus und Reha-Klinik. All diese Institutionen müssen gemeinsam auf die Daten des Patienten zugreifen können und sich gegenseitig austauschen um die Behandlung durchführen zu können. Daher wäre es wünschenswert, diese Daten frei von Medienbrüchen und zusammenhängend vorzuhalten und den beteiligten Institutionen zur Verfügung zu stellen. Dazu eignen sich die Lösungen von Cloud-Speicher-Anbietern: Sie bieten eine verfügbare, leistungsfähige und skalierbare Infrastruktur zur Speicherung und zum Austausch von Daten.

Größte Herausforderung ist dabei allerdings der Schutz der Daten vor unberechtigtem Zugriff. Insbesondere kann Vertrauenswürdigkeit der Anbieter bei der Vielzahl der Angebote im Einzelfall nicht vorausgesetzt werden. Stattdessen muss die Sicherheit der Daten auf technischem Weg gewährleistet werden. So müssen die Daten während Transport und Speicherung geschützt und gleichzeitig Zugriff durch Berechtigte jederzeit gewährleistet sein.

Dazu sollen Dateninhalten Sicherheits-Metadaten beigeordnet werden, welche erlaubte Zugriffsoperationen auf diesen Daten beschreiben. Ziel ist die Entwicklung einer Softwarearchitektur, welche den Zugriff auf die Daten gemäß dieser Sicherheits-Metadaten schützt. Um den Besonderheiten des Gesundheitswesens gerecht zu werden, soll diese Architektur am praktischen Beispiel des Neugeborenen-Hörscreening-Programms in Berlin-Brandenburg erprobt werden.

5. Drittmittelprojekte

- Förderung für Teilprojekt SaPiMa / PhysioGate im Rahmen des Fontane-Projektes durch die Zukunftsagentur Brandenburg, Fördersumme: 1,2 Mio EUR, 5 Doktoranden-/Mitarbeiterstellen
- Förderung für das InstantLab-Projekt durch Microsoft Corp., Fördersumme USD 20k

6. Forschungs Kooperation (außerhalb des HPI)

- Microsoft Corp., Redmond, Windows Academic Team
- Getemed Medizintechnik AG, Teltow
- DFG-Graduiertenkolleg SOAMED, gemeinsam mit Charité Berlin, Humboldt-Universität zu Berlin, Technischer Universität Berlin

7. Publikationen

7.1. Begutachtete Konferenzartikel

- Polze, Andreas; Tröger, Peter; Hentschel, Uwe; Heinze, Theodor
A scalable, self-adaptive architecture for remote patient monitoring.
In: First IEEE International Workshop on Object/component/service-oriented Real-time Networked Ultra-dependable Systems. Carmona, Spain : 2010
- Tröger, P.; Salfner, F., and Tschirpke, S.
Software-Implemented Fault Injection at Firmware Level.
in: IEEE Proceedings of the Third International Conference on Dependability (DEPEND), 2010
- Frank Feinbube, Bernhard Rabe, Martin von Löwis and Andreas Polze.
NQueens on CUDA: Optimization Issues
In ISPDC'2010: Proceedings of the 9th International Symposium on Parallel and Distributed Computing, July 2010, Istanbul, Turkey.
- Alexander Schmidt and Andreas Polze.
KAdvice: Inferring Synchronization Patterns From an Existing Codebase
In RSSE'10: Proceedings of Second International Workshop on Recommendation Systems for Software Engineering, Cape Town, South Africa, May 2010.
- Alexander Schmidt, Andreas Polze, and Dave Probert.
Teaching Operating Systems - Windows Kernel Projects.
In SIGCSE'10: Proceedings of the 41st Technical Symposium on Computer Science Education, Milwaukee, WI, USA, March 2010.
- Frank Feinbube, Peter Tröger, and Andreas Polze. “Joint Forces: From Multithreaded Programming to GPU Computing,” *IEEE Software (Software)*, vol. 28, Oct. 2010, pp. 51-57.

- Andreas Polze and Peter Tröger, “Trends and Challenges in Operating Systems - from Parallel Computing to Cloud Computing,” *Special Issue: Concurrency and Computation: Practice & Experience*. Wiley. (Under review)

7.2. Bücher

- Michael Schöbel, Dissertationsschrift, „Verarbeitung von Ereignisströmen im Betriebssystemkern“, Cuvillier Verlag Göttingen, 2010, ISBN 3869556013
- Peter Tröger. „Cloud Computing“. Buchkapitel in *Service-orientierte Architekturen mit Web Services. Konzepte - Standards - Praxis*. Spektrum Akademischer Verlag, April 2010, ISBN/ISSN 978-3827425492

7.3. Technische Berichte / Standardisierungsdokumente

- "Proceedings of the 5th Ph.D. Retreat of the HPI Research School on Service-oriented Systems Engineering"
Herausg. Meinel, C. , Plattner, H. , Döllner, J. , Weske, M. , Polze, A. , Hirschfeld, R. , Naumann, F. , Giese, H.
Technische Berichte Nr. 31 des Hasso-Plattner-Instituts für Softwaresystemtechnik an der Universität Potsdam ISBN 978-3-86956-036-6
- Proceedings of the Fall 2010 Future SOC Lab Day
Herausg. Christoph Meinel, Andreas Polze, Alexander Zeier, Gerhard Oswald, Dieter Herzog, Volker Smid, Doc D'Errico, Zahid Hussain (Hrsg.)
Technische Berichte Nr. 42 des Hasso-Plattner-Instituts für Softwaresystemtechnik an der Universität Potsdam (2011), ISBN 978-3-86956-114-1

8. Vorträge

8.1. Vorträge auf Tagungen

- Martin v. Löwis. “reverse.name”, 3. Deutscher IPv6-Gipfel, Potsdam, 2010.
- Alexander Schmidt, Andreas Polze, „Teaching Operating Systems - Windows Kernel Projects“, SIGCSE 2010.
- Alexander Schmidt, Andreas Polze. „KAdvice: Inferring Synchronization Patterns From an Existing Codebase“, RSSE'10: Second International Workshop on Recommendation Systems for Software Engineering, Cape Town, South Africa, May 2010
- Peter Tröger. „Software-Implemented Fault Injection at Firmware Level“, The Third International Conference on Dependability (DEPEND), Juli 2010, Venedig, Italien
- Peter Tröger. „System Dependability in the Many-Core Era“, Intel European Research and Innovation Conference (ERIC) 2010.
- Frank Feinbube. “NQueens on CUDA: Optimization Issues”. International Symposium on Parallel and Distributed Computing (ISPDC) 2010. Istanbul, Turkey
- Robert Wierschke. “A scalable, self-adaptive architecture for remote patient monitoring”, First IEEE International Workshop on Object/component/service-

oriented Real-time Networked Ultra-dependable Systems (WORNUS), Mai 2010, Carmona, Spain

- Alexander Schacht. “Vorstellung des FONTANE Forschungsprojekts”. 1. Workshop Technologische Kompatibilität in der Medizintechnik durch Service-orientierte Architekturen, Juni 2010, Lübeck

8.2. Vortragseinladungen außerhalb des HPI

- Andreas Polze, “Cloud Computing – ein Überblick”, Frühjahrstagung Weimarer Kreis, 06.05.2010
- Andreas Polze, “Service Computing in the Cloud”, SOAMED Ringvorlesung – 17.05.2010
- Andreas Polze, “OSM @ FutureSOC Lab”, Steering Committee Meeting FutureSOC Lab, June 16, 2010
- Andreas Polze, “Comparative Evaluation of Cloud Computing Environments”, Windows Core Workshop, April 15/16 2010, Beijing
- Andreas Polze, “Skylab: Operating System Experiments in the Cloud”, Sino-German Forum on Virtual Universities, Beijing University of Technology, Sep. 16-18 2010
- Andreas Polze, “Paradigmenwechsel: Neuerungen in der Rechnerarchitektur und ihr Einfluss auf die Software”, IBM Labor Böblingen, 4.10.2010
- Andreas Polze, “OS Research @ HPI”, Microsoft Research, Mountain View, November 1. 2010
- Andreas Polze, “Introduction to HPI Research School”, SAP Labs Palo Alto, November 2, 2010
- Andreas Polze, “The FutureSOC Lab @ HPI”, HP Labs, Palo Alto, November 3, 2010
- Andreas Polze, Robert Hirschfeld, “HPI Research School on Service-Oriented Computing”, Workshop at Technion, Haifa, Nov. 22, 2010
- Theodor Heinze, Uwe Hentschel, Robert Wierschke, Martin Wolf, Dr. Irene Eiermann, Dr. Martin von Löwis, Prof. Dr. Andreas Polze, “SaPiMa – eine Selbst-adaptive, Priorisierende Middleware”, Workshop bei Getemed, 25. November 2010
- Robert Wierschke, Theodor Heinze, Uwe Hentschel, Jan-Arne Sobania, Martin Wolf, Dr. Martin von Löwis, Prof. Dr. Andreas Polze, „SaPiMa – Self-Adaptive Prioritizing Middleware“, Fontane-Partner Projekttreffen am Döllnsee, 23-24. Februar 2010.
- Andreas Polze, “Research School on Service-Oriented Systems Engineering”, SAP Audimax, St. Leon-Rot, December 6, 2010
- Alexander Schmidt, “Extending Operating System Research into the Cloud”, Cloud Futures 2010 Workshop, Redmond, WA, USA, April 2010
- Alexander Schmidt, “Windows Research Kernel. Faculty Experiences”, Windows Core Workshop, Beijing, China, April 2010
- Alexander Schmidt, “Tool Support for Monitoring Parallel Systems”, University of Cape Town, Kapstadt, Südafrika

- Alexander Schmidt, “ManyCore Systems. A Monitoring Challenge”, SAP Labs, Palo Alto, CA, November 2010
- Alexander Schmidt, “Monitoring ManyCore Systems”, HP Labs, Palo Alto, CA, November 2010
- Alexander Schmidt, “Monitoring ManyCore Systems: Lessons Learned”, Technion, Israel Institute of Technology, Haifa, Israel, November 2010
- Peter Tröger, “System Dependability in the Many-Core Era”, IBM Böblingen, Oktober 2010
- Frank Feinbube. “Programming Models for Hybrid Computing”, Technion – Hasso-Plattner Institute Research Meeting, November 2010
- Peter Tröger, “System Dependability in the Many-Core Era”, Intel ERIC Konferenz, Braunschweig, 2010
- Peter Tröger, “Firmware-Related Research Activities”, Fujitsu Technology Solutions, Paderborn, Dezember 2010
- Uwe Hentschel, “Predictable Communication for Mobile Systems”, Technion, Israel Institute of Technology, Haifa, Israel, November 2010
- Uwe Hentschel, “Predictable Communication for Mobile Systems”, SAP Audimax, St. Leon-Rot, December 6, 2010
- Jan-Arne Sobania, “Setup Challenges and Operating System Experiments on the Intel Single-Chip Cloud Computer”, 1st Many-Core Applications Research Community (MARC) Symposium, Braunschweig, Germany, November 2010

9. Herausgeberschaft

9.1. Websites

- Operating Systems and Middleware Homepage (www.dcl.hpi.uni-potsdam.de)
- Adaptive Services Grid (www.asg-platform.org)
- The Python Programming Language (www.python.org)
(Mitherausgeber Martin v. Löwis)
- Deutscher IPv6-Rat (www.ipv6council.de)
(Mitherausgeber Martin v. Löwis)
- The LOOM.NET Project Page (<http://www.rapier-loom.net>)
(Herausgeber Wolfgang Schult)
- Windows Research Kernel @ HPI
(<http://www.dcl.hpi.uni-potsdam.de/research/WRK/>)
Michael Schöbel, Alexander Schmidt
- Distributed Control Lab
(<http://tb4.asg-platform.org:8000>)
Bernhard Rabe
- OGF DRMAA Arbeitsgruppe
(www.drmaa.org)

10. Mitgliedschaften, Programmkomitees, Gutachtertätigkeiten

10.1. Mitgliedschaften

- Prof. Dr. Andreas Polze
 - Mitglied "Weimarer Kreis" IBM Deutschland.
 - Mitglied IEEE, GI, DECUS (HP User Society)
- Dr. Martin v. Löwis
 - Schatzmeister des SDL Forum e.V.
 - Mitglied im Board of Directors, Python Software Foundation
 - Mitglied im Deutschen IPv6-Rat
- Dr. Peter Tröger
 - Co-Chair der DRMAA Arbeitsgruppe beim Open Grid Forum (OGF)
 - Mitglied IEEE, GI
- Uwe Hentschel, M.Sc.
 - Mitglied VDE Verband der Elektrotechnik Elektronik Informationstechnik e.V.
- Dipl.-Inf. Alexander Schmidt
 - Mitglied IEEE Computer Society
 - Mitglied ACM, ACM SIGOPS

10.2. Mitarbeit in Programmkomitees

- Prof. Dr. Andreas Polze
 - ISORC – Intl. Symposium on Object-ORiented Realtim Systems
 - SAKS – Selbstorganisierende, Adaptive, Kontextsensitive verteilte Systeme
 - ICSOFT – Intl. Conference on Software and Data Technologies
 - ITSIM – International Symposium on Information Technology
- Dr. Martin v. Löwis
 - 6th Workshop on System Analysis and Modelling (SAM 2010)
- Dr. Peter Tröger
 - International Conference on Service-Oriented Computing (ICSOC 2010)
 - International Conference on Software Engineering and Knowledge Engineering (SEKE 2010)

10.3. Gutachtertätigkeiten

- Prof. Dr. Andreas Polze:

- The Computer Journal, Oxford University Press
- Journal on Systems and Software, Elsevier
- Journal on Systems Architecture, Elsevier
- Real-Time Systems Journal, Springer
- South African Computer Journal (SACJ)
- Concurrency and Computation: Practice and Experience (CCPE)
- Gutachter Microsoft Research Ph.D. programme
- Sprecher HPI Forschungskolleg „Service-Oriented Systems Engineering“
- Dr. Martin v. Löwis
 - Gutachter der Dissertation: Antonio Cuni. High performance implementation of Python for CLI/.NET with JIT compiler generation for dynamic languages. DISI-TH-2010-05, Università di Genova

10.4. Gastaufenthalte

- Frank Feinbube – Universal Parallel Computing Research Center (UPCRC) Summer School, University of Illinois, July 2010
- Alexander Schmidt – Windows Academic Program, Microsoft Corp. Redmond, WA, September 2010
- Fahad Khalid - Gastaufenthalt in der Arbeitsgruppe, Oktober/November 2010

11. Workshops

Eröffnung HPI Future SOC Labor

16. Juni 2010

In Kooperation mit Partnern aus der Industrie etabliert das Hasso-Plattner-Institut (HPI) das “HPI Future SOC Lab”, das eine komplette Infrastruktur von hochkomplexen on-demand Systemen auf neuester, am Markt noch nicht verfügbarer, massiv paralleler (multi-/many-core) Hardware mit enormen Hauptspeicherkapazitäten und dafür konzipierte Software bereitstellt.

Es wurden erste Projekte und Ideen rund um das Labor vorgestellt. Höhepunkt der Veranstaltung war die feierliche Eröffnung des Future SOC Labors.

5th Symposium on Future Trends in Service-Oriented Computing

17.-18. Juni 2010

Zum fünften Mal fand das Symposium on “Future Trends in Service-Oriented Computing” (FutureSOC) am Hasso-Plattner-Institut statt. Das Symposium stellte Arbeiten des HPI-Forschungskollegs vor und diskutierte Trends im Bereich der dienstbasierenden Systeme.

Das HPI-Forschungskolleg ist eine interdisziplinäre Graduiertenschule die von den HPI-Fachgebieten getragen wird. Das FutureSOC-Symposium deckt einen weiten Bereich von Themen ab, darunter Fragen der Dienstbeschreibung, des Auffindens und der Komposition

von Dienstan, der Plattform-Konfiguration und der Kapazitätsplanung und des Monitoring. Forschungsarbeiten beschäftigen sich mit Dienst-Middleware, dienstbasierenden Architekturen sowie Fragen der Verwaltung von Diensten. Darauf aufbauend werden Geschäftsmodelle für SOA und deren ökonomische Implikationen studiert.

Das FutureSOC-Symposium wurde unter Beteiligung exzellenter Sprecher aus Industrie und Forschung von unserer Arbeitsgruppe organisiert.