

Neues aus Wissenschaft und Lehre

**Jahrbuch der Heinrich-Heine-Universität
Düsseldorf 2008/2009**

Heinrich Heine
HEINRICH HEINE
UNIVERSITÄT
DÜSSELDORF



d|u|p

düsseldorf university press

**Jahrbuch der
Heinrich-Heine-Universität
Düsseldorf
2008/2009**

**Jahrbuch der
Heinrich-Heine-Universität
Düsseldorf
2008/2009**

**Herausgegeben vom Rektor
der Heinrich-Heine-Universität Düsseldorf
Univ.-Prof. Dr. Dr. H. Michael Piper**

**Konzeption und Redaktion:
Univ.-Prof. em. Dr. Hans Süßmuth**

d|u|p

© düsseldorf university press, Düsseldorf 2010
Einbandgestaltung: Monika Uttendorfer
Titelbild: Leben auf dem Campus
Redaktionsassistentz: Georg Stüttgen
Beratung: Friedrich-K. Unterweg
Satz: Friedhelm Sowa, L^AT_EX
Herstellung: WAZ-Druck GmbH & Co. KG, Duisburg
Gesetzt aus der Adobe Times
ISBN 978-3-940671-33-2

Inhalt

Vorwort des Rektors	13
Gedenken	15
Hochschulrat	17
ULRICH HADDING und ERNST THEODOR RIETSCHEL 18 Monate Hochschulrat der Heinrich-Heine-Universität: Sein Selbstverständnis bei konkreten, strategischen Entscheidungsvorgängen	19
Rektorat	25
H. MICHAEL PIPER Ein Jahr des Aufbruchs	27
Medizinische Fakultät	
<i>Dekanat</i>	33
<i>Neu berufene Professorinnen und Professoren</i>	35
JOACHIM WINDOLF (Dekan) Bericht der Medizinischen Fakultät	41
MALTE KELM, MIRIAM CORTESE-KROTT, ULRIKE HENDGEN-COTTA und PATRICK HORN Stickstoffmonoxid und Nitrit als Mediatoren im kardiovaskulären System: Synthesewege, Speicherformen und Wirkmechanismen	49
JULIA SZENDRÖDI und MICHAEL RODEN Die Bedeutung der mitochondrialen Funktion für die Entstehung von Insulinresistenz und Typ-2-Diabetes	63
BETTINA POLLOK, MARKUS BUTZ, MARTIN SÜDMEYER, LARS WOJTECKI und ALFONS SCHNITZLER Funktion und Dysfunktion motorischer Netzwerke	81
WOLFGANG JANNI, PHILIP HEPP und DIETER NIEDERACHER Der Nachweis von isolierten Tumorzellen in Knochenmark und Blut von Patientinnen mit primärem Mammakarzinom – Standardisierte Methodik und klinische Relevanz	95
ROBERT RABENALT, VOLKER MÜLLER-MATTHEIS und PETER ALBERS Fortschritte in der operativen Behandlung des Prostatakarzinoms	111

MARCUS JÄGER, CHRISTOPH ZILKENS und RÜDIGER KRAUSPE Neue Materialien, neue Techniken: Hüftendoprothetik am Anfang des 21. Jahrhunderts	121
CHRISTIAN NAUJOKS, JÖRG HANDSCHEL und NORBERT KÜBLER Aktueller Stand des osteogenen Tissue-Engineerings.....	137
ULLA STUMPF und JOACHIM WINDOLF Alterstraumatologie: Herausforderung und Bestandteil der Zukunft in der Unfallchirurgie	153
ALFONS LABISCH Die säkularen Umbrüche der Lebens- und Wissenschaftswelten und die Medizin – Ärztliches Handeln im 21. Jahrhundert	161
Mathematisch-Naturwissenschaftliche Fakultät	
<i>Dekanat</i>	175
<i>Neu berufene Professorinnen und Professoren</i>	177
ULRICH RÜTHER (Dekan) Die Mathematisch-Naturwissenschaftliche Fakultät im Jahr 2008/2009	181
FRITZ GRUNEWALD Primzahlen und Kryptographie	185
WILLIAM MARTIN Hydrothermalquellen und der Ursprung des Lebens	203
PETER WESTHOFF C4-Reis – Ein Turbolader für den Photosynthesemotor der Reispflanze	217
MICHAEL BOTT, STEPHANIE BRINGER-MEYER, MELANIE BROCKER, LOTHAR EGGELING, ROLAND FREUDL, JULIA FRUNZKE und TINO POLEN Systemische Mikrobiologie – Etablierung bakterieller Produktionsplattformen für die Weiße Biotechnologie	227
SUSANNE AILEEN FUNKE und DIETER WILLBOLD Frühdiagnose und Therapie der Alzheimerschen Demenz	243
ECKHARD LAMMERT Die Langerhanssche Insel und der Diabetes mellitus	251
THOMAS KLEIN Was kann man von der Fliegenborste lernen?	261
REINHARD PIETROWSKY und MELANIE SCHICHL Mittagsschlaf oder Entspannung fördern das Gedächtnis	275
PETER PROKSCH, SOFIA ORTLEPP und HORST WEBER Naturstoffe aus Schwämmen als Ideengeber für neue <i>Antifouling</i> -Wirkstoffe	281

STEPHAN RAUB, JENS ECKEL, REINHOLD EGGER und STEPHAN OLBRICH Fortschritte in der Forschung durch Hochleistungsrechnen – Kooperation von IT-Service, Informatik und Physik	291
Philosophische Fakultät	
<i>Dekanat</i>	305
<i>Neu berufene Professorinnen und Professoren</i>	307
HANS T. SIEPE (Dekan) Die Philosophische Fakultät im Spiegel der Publikationen ihrer Mitglieder	309
BRUNO BLECKMANN Römische Politik im Ersten Punischen Krieg	315
RICARDA BAUSCHKE-HARTUNG Minnesang zwischen Gesellschaftskunst und Selbstreflexion im Alter(n)sdiskurs – Walthers von der Vogelweide „Sumerlaten“-Lied	333
HENRIETTE HERWIG Altersliebe, Krankheit und Tod in Thomas Manns Novellen <i>Die Betrogene</i> und <i>Der Tod in Venedig</i>	345
ROGER LÜDEKE Die Gesellschaft der Literatur. Ästhetische Interaktion und soziale Praxis in Bram Stokers <i>Dracula</i>	361
SIMONE DIETZ Selbstdarstellungskultur in der massenmedialen Gesellschaft	383
MICHIKO MAE Integration durch „multikulturelle Koexistenz“, durch „Leitkultur“ oder durch eine „transkulturelle Partizipationsgesellschaft“?	393
Wirtschaftswissenschaftliche Fakultät	
<i>Dekanat</i>	411
<i>Neu berufene Professorinnen und Professoren</i>	413
GUIDO FÖRSTER (Dekan) und DIRK SCHMIDTMANN Auswirkungen des Bilanzrechtsmodernisierungsgesetzes auf die steuerliche Gewinnermittlung	415
HEINZ-DIETER SMEETS Finanzkrise – Schrecken ohne Ende?	433
PETER LORSCHIED Praxisorientierte Besonderheiten der Statistik im Düsseldorfer Bachelorstudiengang „Betriebswirtschaftslehre“	457

Juristische Fakultät

Dekanat 467

DIRK LOOSCHELDERS (Dekan)

Neuregelung der Obliegenheiten des Versicherungsnehmers
durch das Versicherungsvertragsgesetz 2008 469

HORST SCHLEHOFER

Die hypothetische Einwilligung – Rechtfertigungs-
oder Strafrechtsausschließungsgrund für einen ärztlichen Eingriff? 485

ANDREW HAMMEL

Strategizing the Abolition of Capital Punishment
in Three European Nations 497

Partnerschaften der Heinrich-Heine-Universität Düsseldorf

JIRÍ PEŠEK

Die Partnerschaft zwischen der Karls-Universität Prag
und der Heinrich-Heine-Universität Düsseldorf 513

**Gesellschaft von Freunden und Förderern der
Heinrich-Heine-Universität Düsseldorf e.V.**

OTHMAR KALTHOFF

Jahresbericht 2008 525

GERT KAISER und OTHMAR KALTHOFF

Die wichtigsten Stiftungen der Freundesgesellschaft 527

Forscherguppen an der Heinrich-Heine-Universität Düsseldorf

KLAUS PFEFFER

Die Forschergruppe 729
„Anti-infektiöse Effektorprogramme: Signale und Mediatoren“ 535

PETER WERNET und GESINE KÖGLER

Die DFG-Forschergruppe 717 „Unrestricted Somatic Stem Cells from Hu-
man Umbilical Cord Blood (USSC)“/„Unrestringierte somatische Stamm-
zellen aus menschlichem Nabelschnurblut“ 545

Beteiligungen an Forschungsgruppen

DIETER BIRNBACHER

Kausalität von Unterlassungen – Dilemmata und offene Fragen 565

Sofja Kovalevskaja-Preisträger

KARL SEBASTIAN LANG

Das lymphozytäre Choriomeningitisvirus – Untersucht mittels eines
Mausmodells für virusinduzierte Immunpathologie in der Leber 583

Graduiertenausbildung an der Heinrich-Heine-Universität Düsseldorf

- SONJA MEYER ZU BERSTENHORST, KARL-ERICH JAEGER und
JÖRG PIETRUSZKA
CLIB-Graduate Cluster Industrial Biotechnology:
Ein neuer Weg zur praxisnahen Doktorandenausbildung 597
- JOHANNES H. HEGEMANN und CHRISTIAN DUMPITAK
Strukturierte Promotionsförderung in der Infektionsforschung durch die
Manchot Graduiertenschule „Molecules of Infection“ 607

Nachwuchsforschergruppen an der Heinrich-Heine-Universität Düsseldorf

- ULRICH HEIMESHOFF und HEINZ-DIETER SMEETS
Empirische Wettbewerbsanalyse 623
- WOLFGANG HOYER
Selektion und Charakterisierung von Bindeproteinen
für amyloidogene Peptide und Proteine 631

Interdisziplinäre Forscherverbände an der Heinrich-Heine-Universität Düsseldorf

- ULRICH VON ALEMANN und ANNIKA LAUX
Parteimitglieder in Deutschland.
Die Deutsche Parteimitgliederstudie 2009 641
- JULIA BEE, REINHOLD GÖRLING und SVEN SEIBEL
Wiederkehr der Folter? Aus den Arbeiten einer interdisziplinären Studie
über eine extreme Form der Gewalt, ihre mediale Darstellung und ihre
Ächtung 649
- KLAUS-DIETER DRÜEN und GUIDO FÖRSTER
Düsseldorfer Zentrum für
Unternehmensbesteuerung und -nachfolge 663
- KLAUS-DIETER DRÜEN
Der Weg zur gemeinnützigen (rechtsfähigen) Stiftung –
Stiftungszivilrechtliche Gestaltungsmöglichkeiten
und steuerrechtliche Vorgaben 665
- GUIDO FÖRSTER
Steuerliche Rahmenbedingungen für Stiftungsmaßnahmen 677

Kooperation der Heinrich-Heine-Universität Düsseldorf und des Forschungszentrums Jülich

- ULRICH SCHURR, UWE RASCHER und ACHIM WALTER
Quantitative Pflanzenwissenschaften – Dynamik von Pflanzen
in einer dynamischen Umwelt am Beispiel der Schlüsselprozesse
Photosynthese und Wachstum 691

Ausgründungen aus der Heinrich-Heine-Universität Düsseldorf

DETLEV RIESNER und HANS SÜSSMUTH

Die Gründung des Wissenschaftsverlags *düsseldorf university press
GmbH* 709

Zentrale Einrichtungen der Heinrich-Heine-Universität Düsseldorf

Zentrale Universitätsverwaltung

JAN GERKEN

Der Umstieg auf das kaufmännische Rechnungswesen:
Die Heinrich-Heine-Universität Düsseldorf nutzt als
Vorreiter die Chancen der Hochschulautonomie 729

Universitäts- und Landesbibliothek

IRMGARD SIEBERT

Sammelleidenschaft und Kulturförderung.
Die Schätze der Universitäts- und Landesbibliothek Düsseldorf 737

GABRIELE DREIS

Das Kulturgut Buch für die Zukunft bewahren:
Bestandserhaltung in der Universitäts- und Landesbibliothek Düsseldorf ... 751

Zentrum für Informations- und Medientechnologie

MANFRED HEYDTHAUSEN und ROBERT MONSER

Die Entwicklung eines Portals für
die Heinrich-Heine-Universität Düsseldorf 769

STEPHAN RAUB, INGO BREUER, CHRISTOPH GIERLING und STEPHAN
OLBRICH

Werkzeuge für Monitoring und Management von Rechenclustern –
Anforderungen und Entwicklung des Tools <myJAM/> 783

Sammlungen in der Universitäts- und Landesbibliothek Düsseldorf

KATHRIN LUCHT-ROUSSEL

Die Düsseldorfer Malerschule in der
Universitäts- und Landesbibliothek Düsseldorf 795

Ausstellungen

ANDREA VON HÜLSEN-ESCH

Jüdische Künstler aus Osteuropa und die
westliche Moderne zu Beginn des 20. Jahrhunderts 813

JENS METZDORF und STEFAN ROHRBACHER

„Geschichte in Gesichtern“ 827

Geschichte der Heinrich-Heine-Universität Düsseldorf

SVENJA WESTER und MAX PLASSMANN

Die Aufnahme des klinischen Unterrichts an der
Akademie für praktische Medizin im Jahr 1919 853**Forum Kunst**

HANS KÖRNER

Frömmigkeit und Moderne.
Zu einem Schwerpunkt in Forschung und Lehre
am Seminar für Kunstgeschichte 865**Chronik der Heinrich-Heine-Universität Düsseldorf**

ROLF WILLHARDT

Chronik 2008/2009 897

Campus-Orientierungsplan 919**Daten und Abbildungen aus dem
Zahlenspiegel der Heinrich-Heine-Universität Düsseldorf** 925**Autorinnen und Autoren** 937

**STEPHAN RAUB, INGO BREUER, CHRISTOPH GIERLING
und STEPHAN OLBRICH**

**Werkzeuge für Monitoring und
Management von Rechenclustern –
Anforderungen und Entwicklung des Tools <myJAM/>**

Rechencluster, wie sie beispielsweise im Zentrum für Informations- und Medientechnologie an der Heinrich-Heine-Universität Düsseldorf betrieben werden, dienen vorwiegend als Werkzeug für die rechenintensive Forschung – an der Heinrich-Heine-Universität im Wesentlichen in den Naturwissenschaften: Physik, Chemie und Biologie. Zur Unterstützung des Betriebs solcher komplexen IT-Infrastrukturen werden wiederum Softwarewerkzeuge benötigt. Die Anforderungen dafür sind vielfältig: Es soll eine homogene Sicht auf die zunehmend heterogenen Rechnerkonfigurationen geboten werden, die Benutzung soll über einen herkömmlichen Webbrowser erfolgen und die Intensität und Effizienz der Nutzung des Rechenclusters soll kategorisiert nach Nutzergruppen, Projekten und Anwendungen ausgewertet werden – nicht zuletzt auch als Basis für eine institutionenübergreifende Abrechnung und als Hilfsmittel für die Planung zukünftiger Hochleistungsrechner (Großgeräte).

Aufgrund der Tatsache, dass ein solches Werkzeug nicht verfügbar war, gehörte zu den Teilprojekten einer Kooperation mit der IT-Firma Bull GmbH die Entwicklung eines Accounting- und Monitoring-Tools <myJAM/>¹. In dem zweijährigen Projekt wurde ein wissenschaftlicher Mitarbeiter von der Bull GmbH finanziert. Die Arbeiten wurden am Lehrstuhl für IT-Management (Univ.-Prof. Dr. Stephan Olbrich) – in enger Zusammenarbeit mit dem Zentrum für Informations- und Medientechnologie – durchgeführt. Nach dem erfolgreichen Abschluss am 30. September 2009 wurde die Projektförderung inzwischen verlängert und beim Bundesministerium für Bildung und Forschung eine Forschungsprämie mit dem Ziel eingeworben, für <myJAM/> im Laufe des Jahres 2010 eine Nachhaltigkeitsstrategie zu entwickeln.

Die Entwicklung des Accounting- und Monitoring-Tools <myJAM/> ist das wohl erfolgreichste Teilprojekt der Kooperation zwischen der Heinrich-Heine-Universität und der Bull GmbH. Bei <myJAM/> handelt es sich um ein netzverteiltes und webbasiertes System für Accounting und Monitoring von Batchjobs auf Rechenclustern. Bei der Vorstellung von <myJAM/> auf mehreren internationalen Konferenzen hat das Projekt jeweils starke Aufmerksamkeit erregt. Viele Universitäten und andere Forschungseinrichtungen, vom Forschungszentrum Jülich über das Max-Planck-Institut Göttingen bis hin zu Forschergruppen mit eigenen kleineren Clustern, setzen <myJAM/> bereits ein oder haben sich an das Entwicklerteam gewandt, um die Einsatzmöglichkeiten von <myJAM/> in ihrem IT-Umfeld zu erörtern.

¹ Vgl. Raub *et al.* (2008).

Einleitung

Wenn in der IT von einem Cluster gesprochen wird, ist damit eine Menge von eng verbundenen, autarken Computersystemen gemeint (Compute-Nodes), die auch physikalisch nah beieinanderliegen. Diese zusammengeschlossenen Computer ermöglichen es, rechenintensive Probleme aus verschiedenen (interdisziplinären) Bereichen effizient zu lösen. Ein solcher Cluster wird als *High Performance Computing Cluster* (HPC-Cluster) oder einfach als Compute-Cluster bezeichnet.

Um die vorhandene Rechenkapazität eines Compute-Clusters nach Möglichkeit voll auszuschöpfen, ist in den häufigsten Fällen eine Verwaltungssoftware vorhanden, die die Zuweisung vorhandener Ressourcen übernimmt. Diese Art von Software wird im Folgenden als Batchsystem bezeichnet (Beispiele sind PBSpro², TORQUE³, SunGridEngine⁴).

Die Nutzer des Clusters müssen ihre Aufträge („Jobs“) inklusive einer Liste der notwendigen Ressourcen, die diese benötigen, an das Batchsystem übermitteln. Die Software entscheidet autonom wann, wo und wie der Job ausgeführt wird; das heißt, es wird ein freies Zeitintervall auf einem der Compute-Nodes determiniert, in dem der Job ausgeführt werden kann. Das Batchsystem verwaltet zusätzlich Details wie die notwendige Systemarchitektur, die Priorität, die Laufzeit und das Ausführungsverhalten des Jobs.

Für Betreiber eines Compute-Clusters ist es im höchsten Maße interessant, Informationen über die laufenden Jobs ihrer Nutzer zu gewinnen, unter anderem, um das Angebot der bestehenden Ressourcen an die meist wachsenden Ansprüche der Nutzer anzupassen zu können. Sehr oft fällt die Beantwortung von Fragen wie „Welche Software wird häufig verwendet?“ oder „Wäre es interessant, die nächste Generation einer Software zu erwerben?“ sehr schwer.

Die notwendigen Informationen, um solche Fragen zu beantworten, gehen über das hinaus, was normale Hardware und Systemüberwachungsdienste (zum Beispiel Ganglia, Nagios, Netperf, top) leisten können (Aufzeichnen der Systemlast, Speicherverbrauch und so weiter). Dieser Umstand wird für heterogene Compute-Cluster zusätzlich erschwert. Ihre physikalische Verteilung macht die Nutzung von bestehenden Softwarelösungen nur teilweise bis gar nicht möglich.

Altair bietet ein kommerzielles Tool zur Analyse von Log-Files an, jedoch fehlt diesem Tool die Möglichkeit, verschiedene Architekturen eines heterogenen Clusters zu unterscheiden, so dass eine architekturbezogene Analyse von Daten nicht möglich ist.

PBS bietet von Haus aus keine automatische Erkennung von laufenden Anwendungen und keine Darstellung der Wartezeiten der Jobs in ihren Warteschlangen an. Da insbesondere das Analyse-Tool nicht Open-Source-Status besitzt, stellt sich nicht zuletzt auch die Frage nach der Erweiterbarkeit für spezielle Problemstellungen des homogenen Managements eines heterogenen Clusters.

Als Lösung dieser Problemstellung wurde das Job-Accounting- und Monitoring-System für homogene und heterogene Cluster <myJAM/> entworfen und entwickelt.

Um Informationen über wartende oder laufende Jobs zu ermitteln, interagiert es mit PBSPro, Torque oder anderen Batchsystemen. Bevor Jobs vom Queueing-System starten,

² Vgl. <http://www.altair.com> (29.10.2009).

³ Vgl. <http://www.clusterresources.com/products/torque> (29.10.2009).

⁴ Vgl. <http://gridengine.sunsource.net> (29.10.2009).

werden diese durch <myJAM/> auf entsprechende Berechtigungen überprüft und gegebenenfalls abgelehnt. Vielfältige Visualisierungsmöglichkeiten in Verbindung mit zahlreichen Anzeigekriterien erlauben einen schnellen Überblick, was, wie viel und wo gerade auf dem Cluster etwas läuft oder in der Vergangenheit gelaufen ist.

<myJAM/> macht sich ausschließlich Open-Source-Softwaretechnologien wie PHP⁵, Perl, Apache⁶, MySQL⁷ und OpenFlashChart⁸ zunutze. Bei OpenFlashChart handelt es sich um eine in PHP und ActionScript entwickelte Komponente, mit der sich Diagramme optisch ansprechend in Webseiten einbinden lassen. Wenn auch alle bisherigen *Proof-of-Concept*-Installationen unter Linux betrieben werden, sind die genannten Softwarekomponenten auch für Windows und Apples OS X verfügbar.

Architektur und Design von <myJAM/>

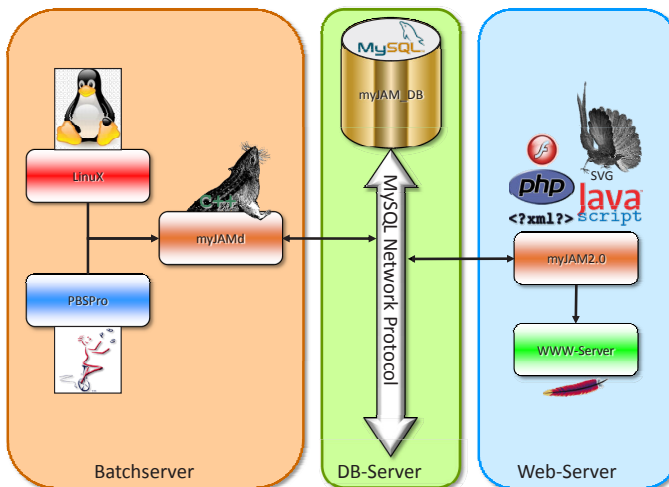


Abb. 1: Architektur von <myJAM/>

Das <myJAM/>-System lässt sich in drei Kernkomponenten gliedern (Abb. 1): Der Batchserver, auf dem der Daemon (myJAMd) läuft, der Webserver, auf dem das Webfrontend (myJAM_www) aufgesetzt ist, und der Datenbankserver mit der <myJAM/>-Datenbank (myJAM_db). Die Komponenten können netzverteilt, das heißt auf separaten Systemen unterschiedlicher Architektur, aufgesetzt werden.

Die Datenbank

Die <myJAM/>-Datenbank dient der Akkumulation der vom Daemon eingesammelten und aufgearbeiteten Daten, die vom Webfrontend abgerufen werden. Die Datenbank

⁵ Vgl. <http://www.php.net> (29.10.2009).

⁶ Vgl. <http://httpd.apache.org> (29.10.2009).

⁷ Vgl. <http://www.mysql.com> (29.10.2009).

⁸ Vgl. J. Glazerbrook, <http://teethgrinder.co.uk/open-flash-chart> (29.10.2009).

dient darüber hinaus als Vermittlungsstelle zwischen Daemon und Webfrontend und der persistenten Speicherung der im Betrieb gesammelten Cluster-Daten (*Cluster History*).

Die Datenbank von <myJAM/> wird mit MySQL5.x realisiert. Das Datenbankdesign berücksichtigt weitgehend die dritte Normalform, um Dateninkonsistenzen vorzubeugen. Aus Performancegründen wurden lediglich einige wenige Tabellen auf die zweite Normalform denormalisiert. User, Projekte, Warteschlangen, Hosts, Anwendungen und Kostenmodelle bilden die zentralen Tabellen. Alle Einträge besitzen Surrogatschlüssel als Primärschlüssel. Attribute, die auf Einträge einer anderen Tabelle referenzieren, benutzen diese Surrogatschlüssel als Fremdschlüssel.

Für das Webfrontend werden die Einträge in den Tabellen und die Referenzen auf andere Tabellen in PHP5-Objekte im Rahmen der <myJAM/>-Klassenbibliothek umgesetzt (*object-relational mapping*).

Der Daemon

Eine Kernkomponente von <myJAM/> ist ein auf dem Batchsystem-Server laufender *Daemon-Prozess*. Der Daemon interagiert mit dem Batchsystem und dem Betriebssystem auf den einzelnen Compute-Nodes. Durch das Batchsystem sammelt dieser Informationen über alle aktiven Jobs, wie die angeforderten und bisher genutzten Ressourcen sowie User- und Projektdaten. Über das Betriebssystem der Compute-Nodes werden zusätzliche Informationen über das tatsächlich ausgeführte Programm (Binary oder Skript) in Erfahrung gebracht.

Diese Informationen werden anhand eines numerischen Modells auf SUs (*Service Units*, entspricht einer Core-Stunde) abgebildet. Die Berechnung der SUs geschieht ebenso wie die Datensammlung in Echtzeit. Die so ermittelten SUs können vom Serviceanbieter mit den Nutzern verrechnet werden.

Unterstützt wird der Daemon durch *Prolog-* und *Epilog-Skripte*, die entsprechend zu Beginn und am Ende eines Jobs ausgeführt werden. Diese überprüfen unter anderem die Legitimation der User sowie bestehende SU-Guthaben und entscheiden daran zum Beispiel, ob ein Jobs mit den angeforderten Ressourcen gestartet werden darf.

Das neueste Feature des Daemons ist die aktive Steuerung von Cluster-Ressourcen. Neben dem Datamining-Thread des Daemons wird parallel auf Eingaben vom Webinterface überprüft und diese – sofern sie die Sicherheitsprüfungen des Daemons bestehen (zum Beispiel ob der Benutzer, der das Kommando geschickt hat, das auch wirklich durfte) – als abstrahiertes Kommando umgesetzt. Exploitversuche werden neben der Sicherheitsprüfung zudem durch das „Laden“ aller legalen Kommandos und die Zuordnung einer Ausführungs-ID abgewehrt; das heißt, alle Kommandos, die der Daemon ausführen kann, werden in regelmäßigen Abständen aus einer Datei eingelesen. Unbekannte Kommandos werden einfach ignoriert und sofern erwünscht gemeldet.

Das Webfrontend

Konzepte

Das Webfrontend von <myJAM/> (Abb. 2) ist eine hoch interaktive Webanwendung (*rich internet application*), die für User und Administratoren das zentrale Interface darstellt, um zu konfigurieren, den Status von Warteschlangen oder des Clusters im Ganzen zu erfahren,

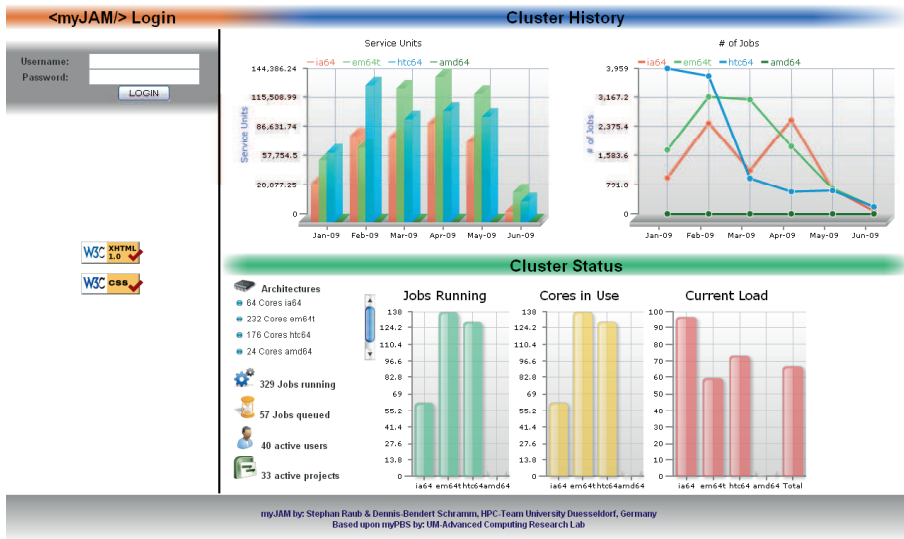


Abb. 2: Login-Seite der <myJAM/>-Webanwendung

sich über laufende Jobs zu informieren, SU-Guthaben abzufragen oder neues Guthaben anzufordern sowie aktuelle und historische Analysen und Statistiken zu visualisieren.

Die Anwendung sollte dem Benutzer eine hohe Übersichtlichkeit bieten, nicht künstlich in ihrer Funktionalität beschnitten sein und an die Performance einer nativen Desktop-Anwendung heranreichen. Neben diesen Gesichtspunkten wählten wir durch die Entwicklung einer Webanwendung einen Betriebssystem-agnostischen Ansatz, wodurch zusätzlich die Installation und Wartung vereinfacht werden sollte.

Konzeptionell stehen das Funktionsspektrum einer Anwendung und die Einfachheit, diese zu bedienen, in einem Spannungsfeld zueinander. Wird eine Ansicht überfrachtet, verliert der Benutzer die Übersicht und wird frustriert, oder er kann die gewünschte Einstellung oder Information nicht erreichen und muss andere Werkzeuge zur Hilfe nehmen.

Die vom Daemon gesammelten Daten sollen in Form von diversen Diagrammen aufgearbeitet und präsentiert werden. Diese wiederum sind mit interaktiven Steuerelementen zur Filterung und Manipulation des Datenstroms verknüpft, um dem Benutzer die optimale Ansicht auf die Daten zu ermöglichen. Eine informationsüberladene Darstellung, in der die wesentlichen Informationen untergehen könnten, kann auf diese Weise vermieden werden, und dadurch behält der Anwender die Kontrolle. Hinzu kommt noch ein intelligentes Caching, um die beste Balance zwischen der Echtzeitdatenerfassung des <myJAM/>-Daemons und der Darstellung der Daten ohne Wartezeiten zu erreichen. Sämtliche Anfragen werden dadurch innerhalb einer Sekunde beantwortet.

Zur Reduzierung von Abhängigkeiten und um die Komplexität möglichst gering zu halten, wurde beim Softwaredesign des <myJAM/>-Webfrontends ein strenges *Schichtenmodell* zugrunde gelegt (Abb. 3).

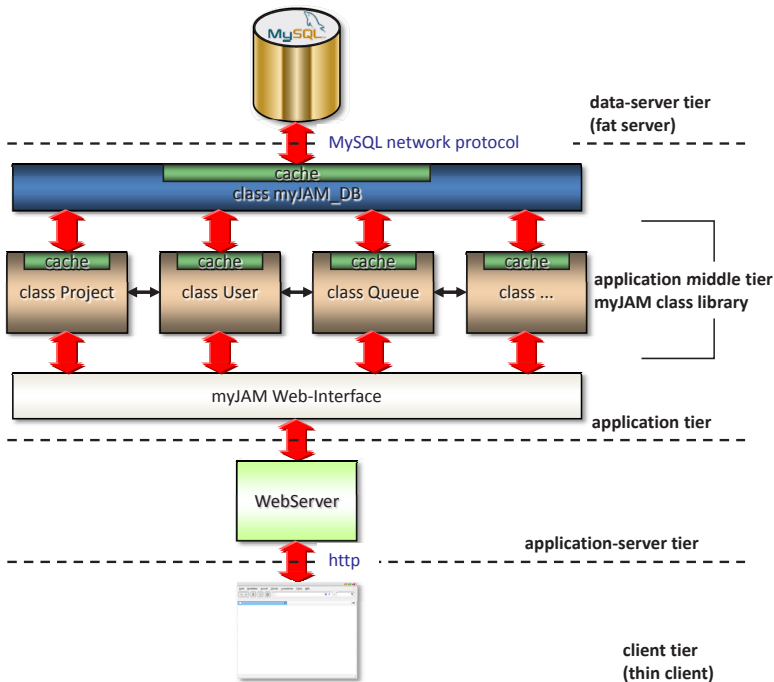


Abb. 3: Softwareschichten von <myJAM/>

Benutzerschnittstelle

Ein intuitiv bedienbares und konsistentes Design entscheidet maßgeblich über die Benutzbarkeit einer Anwendung. Das gilt umso mehr bei Webapplikationen, von denen seitens der Anwender mittlerweile erwartet wird, dass sie selbsterklärend und robust gegen *Trial-and-Error*-Bedienung sind.

Die einzelnen Funktionen von <myJAM/> sind in einer permanent sichtbaren Weise am linken Bildschirmrand in einem vertikalen Menü angeordnet, so dass es sich ständig im peripheren Sichtfeld des Benutzers befindet. Dabei weist das Menü eine sehr flache Struktur mit maximal einer weiteren Stufe auf. Dieser Aspekt wird konsequent in der gesamten Benutzerführung umgesetzt, um die Auffindbarkeit von Funktionalität zu unterstützen.

Die grafische Darstellung von Daten aus der <myJAM/>-Datenbank erfolgt mit Hilfe der Open-Source-Bibliothek *OpenFlashChart*. Diese Bibliothek stellt für verschiedene Diagrammtypen (wie Linien-, Balken- oder Tortendiagramme) PHP-Objekte zur Verfügung und stellt die Diagramme dann als Flash-Content dar.

Um in einen anderen Funktionsbereich zu wechseln, wurde <myJAM/> stark intern verlinkt. Dieses Konzept ermöglicht es, Bereiche und Funktionen auf mehreren Wegen innerhalb von <myJAM/> zu erreichen. User, Projekte, Anwendungen, Jobs und so weiter können in jedem Kontext angeklickt werden und führen direkt zu der entsprechenden Detailansicht auf Basis der Methoden dieses Objekts. Auf diese Weise wird der Anwender

anhand der Datendarstellung und dem seitlichen Menü durch die Anwendungsgebiete von <myJAM/> geführt.

Features von <myJAM/>

Cluster-Status

Mit <myJAM/> ist die Anzeige detaillierter Informationen über den aktuellen Status des Clusters möglich. Dazu gehören eine Übersicht über die Auslastung der einzelnen Architekturen, die Anzahl der benutzten Cores pro Architektur und die Anzahl der Jobs pro Architektur. Ein Histogramm über die Häufigkeit von Jobs nach der Anzahl der genutzten Cores pro Job gibt Auskunft darüber, wie stark parallele Anwendungen vertreten sind.

Auch hier greift wieder das auch in anderen <myJAM/>-Bereichen eingesetzte Konzept der gegenseitigen Verlinkung. So lassen sich die Diagrammelemente anklicken und geben Details preis, anstatt erst von anderer Stelle abgerufen werden zu können, wie zum Beispiel der Applikationsadministration.

Cluster History

Aus der zeitlichen Abbildung des Cluster-Status lassen sich vielfältige Berichte und Analysen erstellen. Aus diesen lassen sich beispielsweise Nutzungsprofile und die Lastverteilung zur weiteren Planung von Kapazitäten und Kostenverteilung des Clusters ermitteln.

Projekte

Hier können neue Projekte angelegt und bereits existierende administriert werden. Mit Hilfe von Projektstatistiken in Diagrammform kann man hier einen Überblick über die Anzahl der benutzten Service-Units und der Jobs des ausgewählten Projekts sowie der benutzten Prozessorarchitekturen erhalten. Die Projektkonfiguration ermöglicht dem Administrator, Änderungen an wichtigen Projektparametern vorzunehmen, wie zum Beispiel Mitglieder des Projekts hinzuzufügen, zu entfernen oder auch die Ressourcen festzulegen, zu denen das Projekt Zugang erhalten kann.

Applikationen und Software-Accounting

<myJAM/> besitzt als ein herausragendes Feature die Fähigkeit, die gerade auf dem Cluster laufenden Anwendungen zu erkennen (*Software-Accounting*). Als Anwendung verstehen wir in diesem Zusammenhang eine ausführbare Datei (Binary), ein Skript oder eine zusammengehörende Sammlung von Binaries und/oder Skripten, die zusammen ein in sich geschlossenes Programmpaket ergeben.

Die Information, was genau gerade auf welchem Knoten läuft, gehört zu einem umfassenden Monitoring dazu. Die historische Analyse dieser Informationen erlaubt es, Anschaffungen an den Nutzerkreis anzupassen oder gezielt Schulungen für die „Hauptnutzergemeinde“ anzubieten.

Doch genau hier tun sich die meisten etablierten Tools sehr schwer: Batchsysteme geben hierbei meist nur einen Surrogatschlüssel an (zum Beispiel eine fortlaufende Jobnummer). Über diesen Surrogatschlüssel sind Zugriffe auf Detailinformationen dieses Jobs möglich,




Admin Applications				
 Application Statistics	 Add New Application	 Assign Applications		
Name	Field	License	OS	Type
BLAST	Bioinformatics	GPL	LinuX	EXE
CCP4	Bioinformatics	no-cost Academic License	LinuX	EXE
CharMM	Computational Chemistry	no-cost Academic License	LinuX	EXE
Dirac	Quantum Chemistry	own development	LinuX	MPI
DL_POLY	Molecular Dynamics	free for academic usage	LinuX	EXE
EMBOSS 6.0	Bioinformatics	GPL	LinuX	EXE
Gaussian03	Computational Chemistry	commercial	LinuX	EXE
Intel MPI Benchmark 3.1 (em64t)	Bioinformatics	commercial	LinuX	MPI
ISPI (sequential)	Condensed Matter Physics	own development	LinuX	EXE
MolCAS 7.2	Quantum Chemistry	commercial	LinuX	EXE
Quapi	Solid State Physic	own development	LinuX	EXE
Segregation	Soft Matter Physics	own development	LinuX	EXE
TurboMole 5.10	Quantum Chemistry	commercial	LinuX	EXE
TurboMole 5.10 (MPI)	Quantum Chemistry	commercial	LinuX	EXE
TurboMole 5.9.1	Quantum Chemistry	commercial	LinuX	EXE
TurboMole 5.9.1 (MPI)	Quantum Chemistry	commercial	LinuX	EXE

Abb. 4: Applikationsverwaltung

worunter sich auch ein vom User frei wählbarer Jobname befinden kann. Um jedoch zuverlässig anzeigen zu können, welche Anwendung zu diesem Job gehört, wäre man auf die Unterstützung der User angewiesen, die für jeden Job angeben müssen, um was für eine Anwendung es sich handelt. Das ist für die User lästig und insgesamt fehleranfällig. Eine automatische, erweiterbare Erkennung der Anwendung gab es unseres Wissens bisher nicht. <myJAM/> soll diesen Mangel beheben.

Das Betriebssystem kennt den Dateinamen des gerade ausgeführten Binarys oder Skripts. Doch das allein reicht nicht, da viele Nutzer dazu neigen, jedes ihrer selbst entwickelten Binaries „a.out“ zu nennen, obwohl es sich *de facto* um völlig unterschiedliche Anwendungen handelt. Umgekehrt bestehen Softwaresuiten teilweise aus über 100 Teilprogrammen – also völlig verschiedene Binarys oder Skripte, die jedoch unter dem gleichen Namen (den Namen der Softwaresuite) erfasst werden sollten.

<myJAM/> nutzt daher digitale Signaturen der Binarys und Skripte. Mit der Signatur kann dann aus einer Datenbank abgefragt werden, um welche Anwendung es sich tatsächlich handelt.

Dieses Konzept lebt und stirbt damit, wie einfach und komfortabel diese Datenbank aktuell gehalten werden kann. Deshalb haben wir ein durchgängiges Konzept umgesetzt, bei dem man an jeder Stelle des Frontends, an der man mit einem Binary (oder Skript) in Kontakt kommt, durch einen einfachen Klick den entsprechenden Hash einer Anwendung zuordnen kann.

Die Klartextnamen von Applikationen sowie die dazugehörigen Zusatzinformationen über Lizenz, Betriebssystem und so weiter können einfach angelegt und verwaltet werden. Im „Assign Applications“-Fenster werden alle jemals gelaufenen Binarys mit ihrem vollen Pfad nach absteigender Häufigkeit sortiert aufgelistet. Jedem Eintrag kann jetzt eine der bereits angelegten Applikationen zugewiesen werden, oder es kann der Eintrag direkt als Ausgangspunkt für das Anlegen einer neuen Applikation verwendet werden.

Warteschlangen

Host: gauss0 (PBSPro_10.0.0.62981)									
In	Queue ID	Queue Name	Queue Weight	Running Jobs	Queued Jobs	Details	Out	Load	
=>	4	short_cmn	1	0	0	Details =>			
=>	3	verylong_cmn	1	7	0	Details =>			
=>	5	medium_cmn	1	47	1	Details =>			
=>	6	long_cmn	1	4	0	Details =>			
=>	7	BenchMarking	1	1	0	Details =>			
=>	8	BioJob	1	0	0	Details =>			
=>	9	DVRP	1	0	0	Details =>			Queue BioJob
=>	10	parallel	1	0	0	Details =>			Queue is running . Jobs in this queue will be started when the time is right.
=>	11	spartan	1	0	0	Details =>			
=>	12	dirac	1	1	2	Details =>			Admin: Click to stop this queue.
=>	13	mrcc	1	8	0	Details =>			
=>	14	TheoPhys	1	24	0	Details =>			
=>	17	default	1	0	0	Details =>			
Total				92	3	Details			
Host: gauss.rz.uni-duesseldorf.de (TORQUE 2.3.5)									
In	Queue ID	Queue Name	Queue Weight	Running Jobs	Queued Jobs	Details	Out	Load	
=>	19	infinity	1	0	0	Details =>			
=>	18	FlexX	1	0	0	Details =>			
Total				0	0	Details			

Abb. 5: Queue-Übersicht

<myJAM/> bietet eine tabellarische Übersicht über die Warteschlangen des Queueing-Systems pro Host. Zu jeder Warteschlange wird die Anzahl der laufenden und wartenden Jobs angezeigt. Besitzt der angemeldete User Administratorrechte, so zeigt ein Klick auf eine der beiden Zahlen eine detaillierte Liste der betreffenden Jobs. Außerdem sehen User mit Administratorrechten zu jeder Warteschlange den Link „Details“, der eine nach Usern aufgeschlüsselte Übersicht der Meldungen vom Scheduler darüber liefert, warum ein bestimmter Job nicht gestartet werden kann, sowie über die Anzahl der betroffenen Jobs. Ein Klick auf diese Anzahl führt direkt zur bereits bekannten Detailansicht dieser Jobs.

Durch einfaches Klicken auf die Pfeile kann jede Queue ganz bequem vom Webinterface aus gesperrt werden. Dabei ist es möglich, beide „Enden“ einer Queue separat zu sperren: Entweder werden keine Jobs mehr angenommen oder keine Jobs mehr aus der Queue genommen. In einem Wartungsintervall ist es zum Beispiel wünschenswert, nur die Entnahme von Jobs aus der Queue zu sperren.

Literatur

RAUB, S., D.-B. SCHRAMM und S. OLBRICH (2008). „myJAM – Accounting und Monitoring auf Rechenclustern“, in: P. MÜLLER, B. NEUMAIR und G. D. RODOSEK (Hrsg.). *Lecture Notes in Informatics – Proceedings*. Bd. P-149, 147–158. (Series of the GI)

ISBN 978-3-940671-33-2



9 783940 671332