

# Endbericht

PG549

Cloudi

Benutzerfreundliche Erzeugung und effiziente Verwaltung von  
virtuellen Maschinen in Cloud-Umgebungen

Datum: 5. April 2011  
Wintersemester 2010 / 2011

Die Projektgruppe:

Abderrahim Elbaz	Sami Arslan
Erdal Sercan	Sebastian Müller
Feng Wang	Steffen Dickehage
Gerhard Sikora	Sven Terfort
Maximilian Ramke	Tobias „Lausi“ Lauszat
Mohammed Taybi	Wladimir Wenner

Copyright © Die Projektgruppe, Dortmund, Januar 2011.  
Alle Rechte sind den Autoren vorbehalten.

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
<b>2</b>	<b>Organisation</b>	<b>4</b>
2.1	Verlauf der PG . . . . .	4
2.1.1	Kick-Off . . . . .	4
2.1.2	Erstes Semester . . . . .	7
2.1.3	Zweites Semester . . . . .	8
2.2	Projektinfrastruktur . . . . .	8
2.2.1	Electronic Workspace . . . . .	8
2.2.2	Redmine . . . . .	8
2.2.3	Eclipse und Netbeans . . . . .	9
2.2.4	L <sup>A</sup> T <sub>E</sub> X . . . . .	9
2.2.5	Cloud Infrastruktur Manager . . . . .	9
<b>3</b>	<b>Anforderungsanalyse</b>	<b>10</b>
3.1	Funktionale Anforderungen . . . . .	10
3.2	Nichtfunktionale Anforderungen . . . . .	11
<b>4</b>	<b>Architektur</b>	<b>12</b>
4.1	Voraussetzungen für den Einsatz von Cloudi . . . . .	16
4.1.1	Hardware-Voraussetzungen . . . . .	16
4.1.2	Software-Voraussetzungen . . . . .	17
<b>5</b>	<b>Grundgerüst</b>	<b>18</b>
5.1	Wahl des Hypervisors . . . . .	18
5.2	Manager für die VM-Infrastruktur . . . . .	19
5.3	OpenNebula . . . . .	22
5.3.1	Tools . . . . .	22
5.3.2	Core . . . . .	23
5.3.3	Drivers . . . . .	24
5.4	Kommunikationsschnittstelle zwischen VM-Middleware und Cloudi . . . . .	24
5.5	Erstinstallation . . . . .	25
5.6	Scheduling mit Haizea . . . . .	27
5.6.1	Praxis . . . . .	29
5.7	Cloudi aus der Server-Ansicht . . . . .	30

---

<b>6</b>	<b>Monitoring</b>	<b>32</b>
6.1	Architektur Ganglia . . . . .	34
6.1.1	gmond . . . . .	34
6.1.2	gmetad . . . . .	35
6.1.3	Kommunikation . . . . .	36
6.2	RRDTool . . . . .	37
6.3	PHP Webfrontend . . . . .	38
6.4	Das Monitoring mit Einsatz von Ganglia . . . . .	41
6.5	Zusammenfassung und Ausblick . . . . .	43
<b>7</b>	<b>Ressourcenabschätzung</b>	<b>45</b>
7.1	Simulation . . . . .	45
7.1.1	CloudSim . . . . .	45
7.1.2	Haizea . . . . .	46
7.1.3	CloudSim – Architektur . . . . .	46
7.2	Implementierung in Cloudi . . . . .	49
7.3	Projekt zur Cloud Simulation an der TU Dortmund . . . . .	49
7.4	Zusammenfassung und Ausblick . . . . .	49
7.5	Vorhersage . . . . .	50
<b>8</b>	<b>Fazit und Ausblick</b>	<b>52</b>
<b>A</b>	<b>Benutzerhandbuch</b>	<b>54</b>
<b>B</b>	<b>Installationshandbuch</b>	<b>76</b>
B.1	Ganglia . . . . .	76
B.2	Apache inkl. OpenSSO und mod_jk . . . . .	78
B.2.1	OpenSSO . . . . .	78
B.2.2	mod_jk . . . . .	78
B.3	Opennebula . . . . .	79
B.4	cloudi . . . . .	80
<b>C</b>	<b>Entwicklerhandbuch</b>	<b>81</b>

# 1 Einleitung

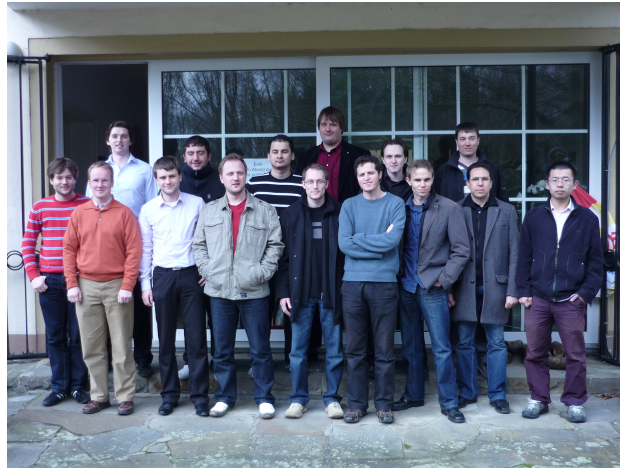


Abbildung 1: Die Projektgruppe

Virtualisierungstechniken und Cloud Computing haben sich zu wichtigen Größen beim Management von IT-Infrastrukturen entwickelt. So bieten virtuelle Maschinen den Vorteil, dass nur wenige Detailkenntnisse über die hinter der Virtualisierung liegenden Basisinfrastruktur wie z.B. der Hardware oder dem Betriebssystem, die dort laufen, benötigt werden. Gleichzeitig sorgt die Virtualisierung für viele Freiheiten in der Konfiguration von Diensten. So ist es möglich innerhalb der virtuellen Maschinen das favorisierte Betriebssystem, beliebige Softwarepakete oder sogar selbst entworfene Betriebssysteme zu verwenden – auch ohne Administrationsrechte auf oder irgendwelche Detailkenntnisse über das physikalische, zu Grunde liegende System zu besitzen.

Für die Anbieter hingegen ist es durch Virtualisierung und Cloud Computing möglich IT-Infrastrukturen zur Verfügung zu stellen, die trotz möglicherweise sehr heterogener physikalischer Grundlagen nach außen eine homogene – und dadurch für die Zielgruppe einfach handhabbare – Rechnerinfrastruktur darstellen. Darüberhinaus bietet Cloud Computing eine gute Möglichkeit die Auslastung der physikalischen Infrastruktur hinsichtlich Stromkosten, Klima, Netzwerk, Personal und weiteren Kostenfaktoren zu optimieren. Eine Vielzahl kommerzieller Angebote wie z.B. von Amazon oder Google zeugen vom Erfolg dieser Technologien.

Das IT und Medien Centrum der TU Dortmund (kurz: ITMC) betreibt diverse Hosting-Angebote für den Hochschulstandort Dortmund. Diese reichen von dem Angebot des wissenschaftlichen Rechnens auf ca. 5000 CPU-Kernen über den Betrieb virtueller Server für zentrale Dienste der TU Dortmund

(wie z.B. Mail-Server, Web-Server, eLearning-Plattformen) bis zu spezifischen Angeboten für einzelne Lehrstühle. Zurzeit werden schon über 100 virtuelle Maschinen in einer Virtualisierungsumgebung auf VMWare-Basis betrieben. Um dieses Angebot zukünftig noch weiter auszubauen, sind neben organisatorischen vor allem auch wissenschaftliche sowie softwaretechnische Aufgabenstellungen zu bewältigen. Einerseits besitzt die breite Zielgruppe an der TU Dortmund in der Regel weder über das notwendige Fachwissen noch über die notwendigen Fähigkeiten, um die Freiheiten für die Konfiguration der virtuellen Maschinen voll auszuschöpfen. Die Nutzung von virtuellen Maschinen erfordert die komplette Installation eines Betriebssystems und der Anwendungssoftware sowie jeweils deren Konfiguration. Darüberhinaus können vorgefertigte Festplatten-Abbilder (z.B. von bereits existierenden virtuellen Maschinen oder angebotenen Vorlagen) genutzt werden. Diese müssen zurzeit jedoch noch aufwendig in Handarbeit in die aktuelle VMWare-Lösung eingepasst werden (z.B. Netzwerkschnittstellen konfigurieren, Größe der Festplatten korrigieren, Software-Pakete nachinstallieren). Andererseits wird das ITMC hingegen technisch bisher nur unzureichend in der (dynamischen) Zuordnung von virtuellen Maschinen zur physikalischen Hardware unterstützt. Nicht berücksichtigt sind bisher die Verteilung des Datenspeichers, der Ausgleich von Datentransfers im Netzwerk und die Migration von virtuellen Maschinen auf fremde Infrastrukturen an anderen Hochschulen im Ruhrgebiet oder Infrastrukturen kommerzieller Angebote. Desweiteren existieren gegenwärtig keine integrierten Werkzeuge, um den zukünftigen Bedarf an Hardware abzuschätzen und somit effektiv weitere Beschaffungen zu planen. Um dem ITMC und seinem Personal das Leben in diesen Punkten zu vereinfachen und somit durch eine möglichst automatisierte Virtualisierungslösung Zeit und Geld für neue Herausforderungen zur Verfügung zu haben, ist die Aufgabe für die Projektgruppe einige der vorgenannten Herausforderungen prototypisch zu lösen.

Die Aufgabenstellung für die Projektgruppe liegt daher vor allem in drei Punkten: Zum einen muss eine Managementoberfläche entwickelt werden, mit der es möglich ist, die physikalischen Ressourcen und die entsprechende Zuweisung von virtuellen Ressourcen an diese zu verwalten. Dies soll dabei möglichst dynamisch und automatisiert erfolgen. Die Projektgruppe muss (mindestens eine) Möglichkeit(en) finden, den Investitionsbedarf sicherer und gleichzeitig einfacher abschätzen zu können.

Da die zu entwickelnde Lösung niederschwellig für die Zielgruppe erreichbar sein soll, muss sie eine Weboberfläche enthalten. Dies setzt z.B. auch eine einfache Verwaltung der Zugangsdaten zum System (Anlegen, Löschen, Rollen, Zugriffsrechte) voraus, wobei eine Anbindung an das bereits existierende Identity Management System der TU Dortmund durchaus wünschenswert

ist. Gleichzeitig müssen aber auch die virtuellen Maschinen stabil verwaltet werden (Starten, Stoppen, Anlegen, Konfigurieren). Gerade in diesem Punkt liegt der auf der Oberfläche sichtbare Mehrwert. Hinzu kommen Funktionalitäten wie die automatisierte Konfiguration von virtuellen Maschinen anhand von z.B. auswählbaren Software- und Dienstkonfigurationen, zeitlichen Vorgaben für die Standardoperationen sowie die Möglichkeit virtuelle Maschinen in Gruppen zusammenzufassen. So soll gewährleistet werden, dass die Virtualisierungsoberfläche so weit wie möglich automatisiert und autonom betrieben werden kann – ohne mehr Zeit zu investieren als notwendig.

Neben der Weboberfläche soll eine möglichst automatische Migration von virtuellen Maschinen auf andere Server – bei Bedarf sogar andere Infrastrukturen wie z.B. Amazon oder Rechenzentren innerhalb der Universitätsallianz Metropole Ruhr (UAMR) – ermöglicht werden. Dabei werden die verfügbaren Rechen-, Netzwerk- und Speicherkapazitäten berücksichtigt. Dadurch sollen drohende Überlasten, aber auch zu geringe Auslastungen einzelner physikalischer Server und damit unnötiger Stromverbrauch vermieden werden. Wichtige Voraussetzungen zum Gelingen des Projektes sind eine klare Schnittstellendefinition zwischen Web-Frontend und der Ressourcenverwaltung im Backend sowie die Zuverlässigkeit der eingesetzten Softwareprodukte.

Zur Planung zukünftiger Investitionen sollen anhand bereits vorhandener Daten mittels unterschiedlicher Methoden Vorhersagen getroffen werden, wie sich der Bedarf an Hardware in Zukunft entwickeln könnte. Dazu bedarf es neben der Erfassung der notwendigen Daten in der Ressourcenverwaltung vor allem entsprechender Algorithmen und Vorhersagemodelle, um den Ressourcenbedarf mit Hilfe der Weboberfläche abzuschätzen. Sofern beim Personal entsprechende Kenntnisse über die Modelle vorhanden sind, könnten dann später unterschiedliche Vorhersagemodelle zur Auswahl angeboten werden. Inwieweit die existierenden Daten genutzt werden können, um z.B. beim Anlegen neuer virtueller Maschinen und/oder der Konfiguration bereits existierender Maschinen zu beraten, bleibt zu prüfen. Dazu bleiben Ähnlichkeitsmaße zwischen Anforderungen und bereits laufenden virtuellen Maschinen zu entwickeln.

Diesen Herausforderungen haben sich die Teilnehmer der Projektgruppe 549 „Benutzerfreundliche Erzeugung und effiziente Verwaltung von virtuellen Maschinen in Cloud-Umgebungen“ erfolgreich gestellt. Der vorliegende Bericht dokumentiert die erzielten Ergebnisse und Erkenntnisse. Die Teilnehmer sind gemeinschaftlich Autoren dieses Berichtes und freuen sich über jede Art des Feedbacks. Die Teilnehmer bedanken sich für die durchaus hilfreichen Hinweise von Dr. Thomas Röblitz und Dipl.-Phys. Florian Feldhaus, die das Projekt von Seiten des Veranstalters begleitet haben und jederzeit ein offenes Ohr für uns hatten.

## 2 Organisation

### 2.1 Verlauf der PG

Die Projektgruppe lief vom Sommersemester 2010 bis zum Wintersemester 2010/2011 mit jeweils acht Semesterwochenstunden an der TU Dortmund. Um die erarbeiteten Ergebnisse zu präsentieren und das weitere Vorgehen abzustimmen, fanden während der Vorlesungszeit wöchentliche Treffen statt. Die Protokolle der wöchentlichen Treffen wurden allen Teilnehmern und Betreuern im SVN (Vgl. Kapitel 2.2.2 auf Seite 8) zur Verfügung gestellt. Die Moderation sowie die Vorbereitung der Treffen fand 2-wöchentlich-wechselnd durch die Teilnehmer statt.

#### 2.1.1 Kick-Off

Zu Beginn der Projektgruppe trafen sich die Teilnehmer und Betreuer zu einem 2-tägigen Kick-Off im Universitätskolleg Bommerholz in Witten. Ziel des Meetings war es zum einen, das Kennenlernen der Teilnehmer untereinander zu fördern. Zum anderen sollten im Rahmen dieser beiden Tage auch schon die ersten inhaltlichen Grundlagen gelegt werden. Dazu haben alle Teilnehmer jeweils ein Thema vorbereitet, das sie im Rahmen eines kurzen Vortrages den anderen Teilnehmern vorgestellt haben. Die Aufteilung der Themen erfolgte in einem ersten, vorbereitendem Treffen am 25. Januar 2010. Bei dem Kick-Off gab es folgende Ergebnisse.

**Projektmanagement** Im Vortrag über das Projektmanagement wurde den anderen Teilnehmern näher gebracht, was sie in einem Projekt erwartet, wie sie sich den Ablauf vorzustellen haben und welche Rollenverteilung unter den Projektmitgliedern erfolgen kann (z.B. Projektleiter). Definiert wurde das Projektmanagement nach DIN 69901-5:2009-01 als „Gesamtheit von Führungsaufgaben, -organisation, -techniken und -mitteln für die Initiierung, Definition, Planung, Steuerung und den Abschluss von Projekten“ [1].

**Projektinfrastruktur** Die Projektinfrastruktur wurde definiert als alle materiellen und immateriellen Einrichtungen und Hilfsmittel, die zur Durchführung eines Projektes notwendig sind. Dies umfasst insbesondere die benutzte Hard- und Software. Im Mittelpunkt des Vortrags standen vor allem verschiedene Softwarelösungen, die das Management in einem Projekt vereinfachen sollen. Dabei wurde auf eine funktionierendes Versionsmanagementsystem sehr viel Wert gelegt. Da neben der Software auch das Personal

zur Projektinfrastruktur gezählt werden kann, wurden aus der Praxis heraus auch allgemeine Tricks und Tipps für den Projektalltag präsentiert. Demnach sollte insbesondere die Kommunikation untereinander reibungslos funktionieren, um ein produktives Arbeiten miteinander zu gewährleisten.

**Software-LifeCycle und Qualitätsmanagement** Der Software-LifeCycle ist eine Bezugsbasis für die Phasen, die ein Softwareprodukt bei seiner Herstellung und dem späteren Einsatz durchläuft. Somit ist es die Grundlage für verschiedene Phasenmodelle, nach denen eine Softwareentwicklung erfolgen kann. Es wurden im Vortrag, die einzelnen Phasen mit den dazugehörigen Aktivitäten beschrieben. Anschliessend wurde das Software-Qualitätsmanagement definiert als alle Tätigkeiten der Gesamtführungsaufgabe, die die Qualitätspolitik, Ziele und Verantwortungen festlegen, sowie diese durch Mittel der Qualitätsplanung, Qualitätslenkung, Qualitätssicherung und Qualitätsverbesserung im Rahmen des Qualitätsmanagementsystems verwirklichen. Erwähnt wurden hier als Funktionen des Qualitätsmanagements die Ermittlung der Bedürfnisse des Kunden oder der Kundin, Einhaltung der Spezifikationen und die Ermittlung von Qualitätsrisiken.

**Webtechnologien – Frontend** Mit Webfrontend wird der Teil einer Internet-Anwendung bezeichnet, der dezentral jeweils auf dem PC im Internet-Browser verarbeitet wird. Das Webfrontend kommuniziert mit dem serverseitigen Teil der Anwendung per http oder https.

Ausgehend von JavaScript wurden aktuelle Technologien wie AJAX, XML, CMS und Frameworks betrachtet. An konkreten Lösungen wurden das Google Web Toolkit (GWT)<sup>1</sup> und das dojotoolkit<sup>2</sup> vorgestellt.

**Webtechnologien – Backend** In Abgrenzung zum Frontend ist das Backend der Teil der Internet-Anwendung, der dem Nutzer verborgen bleibt und umfasst die Programme zur zentralen Verwaltung von Daten, die auf den Servern laufen. Das Backend verarbeitet die Daten, die im Frontend eingegeben werden und sendet Ergebnisse zurück an das Frontend. Datenbanken sind wichtig für die Speicherung der Daten. Als populäre Datenbanken wurden innerhalb der Projektgruppe MySQL<sup>3</sup>, Microsoft SQL Server (MSSQL)<sup>4</sup> und Oracle<sup>5</sup> diskutiert. Darüberhinaus wurden als weitere Webtechnologien

<sup>1</sup><http://code.google.com/intl/de/webtoolkit/>

<sup>2</sup><http://dojotoolkit.org>

<sup>3</sup><http://www.mysql.com>

<sup>4</sup><http://www.microsoft.com/sqlserver>

<sup>5</sup><http://www.oracle.com/us/products/database/enterprise-edition/>

en „PHP: Hypertext Preprocessor“ (PHP)<sup>6</sup>, Java Enterprise Edition (Java EE)<sup>7</sup>, Java Naming and Directory Interface (JNDI)<sup>10</sup>, Java Transaction API (JTA)<sup>11</sup>, J2EE Connector Architecture (JCA)<sup>12</sup> und Java Persistence API (JPA)<sup>13</sup> vorgestellt und diskutiert.

**Virtualisierung** Neben einer Definition von Virtualisierung und den Vorteilen, die Virtualisierung mit sich bringt, wurden die grundlegenden Ideen von virtuellen Maschinen erläutert. Virtuelle Maschinen sind demnach eine Art Softwarecontainer, welche ein eigenes Betriebssystem besitzen und darauf basierend Anwendungen und Dienste ausführen. Die virtuelle Maschine verhält sich dabei wie ein physischer Rechner. Sie besitzt eigene (jeweils virtualisierte) Prozessoren, Hauptspeicher, Festplatte(n) und Netzwerkkarte(n). Die Anzahl der virtuellen Maschinen pro physischem Rechner ist lediglich durch die physischen Ressourcen beschränkt. Der Hauptgrund für Virtualisierungen ist die größere Flexibilität gegenüber echter Hardware. So ist es z.B. möglich unterschiedliche Betriebssysteme gleichzeitig auf der selben Hardware auszuführen. Durch Virtualisierung lässt sich die Anzahl der physischen Server verringern. Das führt zu Ersparnissen bei Strom, Platz, Anschaffung und Personal. Die vorhandene physische Hardware wird durch mehrere virtuelle Maschinen besser ausgelastet und gerade teure Serverressourcen werden so deutlich effizienter genutzt. Durch die flexible Verteilung von Prozessoren, Arbeitsspeicher und Netzwerk werden den virtuellen Maschinen genau die Ressourcen zugewiesen, die sie wirklich brauchen – ohne Umbau physischer Komponenten. Ein skalierbarer Ausbau der Kapazitäten und die Lastverteilung sind je nach Virtualisierungslösung im laufenden Betrieb möglich. Weiterhin wurde im Vortrag die Rolle des Hypervisors dargestellt. Der Hypervisor spielt in der Virtualisierung eine zentrale Rolle. Er ist eine abstrakte Verwaltungsschicht, über die die Gastbetriebssysteme auf die physischen Ressourcen wie Speicher, Ein-/Ausgabegeräte und Netzwerkschnittstellen zugreifen. Der Hypervisor ist quasi ein auf ein Minimum reduziertes Meta-Betriebssystem, das die Hardwareressourcen unter den Gastsystemen verteilt, so wie ein Betriebssystem sie unter den laufenden Prozessen verteilt.

---

<sup>6</sup><http://www.php.net>

<sup>7</sup><http://www.oracle.com/technetwork/java/javaee/>, Enterprise JavaBeans (EJB)<sup>8</sup>, JavaServer Pages (JSP)<sup>9</sup><http://java.sun.com/products/jsp/>

<sup>10</sup><http://www.oracle.com/technet/java/index-jsp-137536.html>

<sup>11</sup><http://www.oracle.com/technetwork/java/javaee/tech/jta-138684.html>

<sup>12</sup><http://java.sun.com/j2ee/connector/>

<sup>13</sup><http://www.oracle.com/technetwork/java/javaee/tech/persistence-jsp-140049.html>

**Cloud Computing** Durch Cloud Computing wird die Bereitstellung und Nutzung von IT-Infrastrukturen sowie von Plattformen und Anwendungen aller Art ermöglicht. Die Dienste werden von einem Provider im Internet oder Intranet erbracht. Aber auch die Nutzer von Cloud-Dienste können ihre eigenen Dienste im Internet anbieten. Cloud-Ressourcen sind in der Regel virtualisiert, so dass Cloud-Nutzer stets eine wunschgemäße, beliebige Sicht auf ihre jeweilige Infrastruktur haben. Bei der Cloud Architektur wird zwischen der organisatorischen und der technischen Sicht unterschieden. Die organisatorische Sicht zeichnet sich durch eine Trennung der organisatorischen Einheiten von Benutzern und Anbietern aus. Auf dieser Sicht aufbauend werden Public, Private und Hybrid Clouds unterschieden. Die technische Sicht der Cloud Architektur hingegen orientiert sich überwiegend an funktionalen Eigenschaften und beherbergt die unterschiedlichen Cloud Services wie z.B. Infrastructure as a Service oder Platform as a Service. Die Cloud Architektur der technischen Sicht beruht auf dem Grundmuster der Schichtung. Dabei sind die Schichten nach Ihrem Abstraktionsgrad angeordnet. Höhere, abstrakte Schichten können die Dienste der tieferen Schichten zur eigenen Dienstrealisierung benutzen. So können z.B. PaaS-Dienste auf Basis von IaaS-Diensten realisiert werden.

**Simulation** Neben dem Management der virtuellen Maschinen sowie der allgemeinen Ressourcenverwaltung stand in der Projektgruppe die Simulation von Cloud Umgebungen im Mittelpunkt. Dies soll helfen den zukünftigen Ressourcenbedarf voraussagen zu können. Dies war sowohl für die Beratung bei der Erzeugung von virtuellen Maschinen, als auch bei der Planung zur Anpassung der Hardware an hier prognostizierte Entwicklungen der Ressourcenauslastungen sehr wichtig. Die Cloud kann dabei als gesamtes System simuliert werden oder die Beschränkung auf Teilbereiche sollte möglich sein, wie Ressourcenauslastungen oder Benutzerzugriffe. Zentrale Punkte bei der Simulation waren allerdings immer die virtuellen Maschinen und deren Scheduling. Im Vortrag wurden Modelle und Programme für die Simulation im Bereich des Cloud Computings wie CloudSim, Haizea oder SPECI kurz vorgestellt.

### 2.1.2 Erstes Semester

Im ersten Semester ist es der Projektgruppe – neben dem inhaltlichen und gruppendynamischen Einarbeiten – gelungen einen ersten Prototypen realisiert zu haben. Die Weboberfläche bot bereits erste Funktionalitäten wie ein angedeutetes Benutzermanagement. Die virtuellen Maschinen, die angezeigt

wurden, existierten auch tatsächlich in der Cloud Manager-Installation. Insgesamt funktionierten bereits weite Teile der Standardfunktionen, die die Weboberfläche mitbringen musste.

Zur weiteren Abstimmung der Minimalziele wurde in Zusammenarbeit mit den Betreuern ein Pflichtenheft erstellt, welches für beide Seiten (Projektgruppe und Betreuer) als eine Art Vertrag dienen sollte.

### 2.1.3 Zweites Semester

Im zweiten Semester wurde das Monitoring der virtuellen Maschinen umgesetzt, sowie die Simulation und Vorhersage des Ressourcenbedarfs. Weiterhin wurde die Weboberfläche um Funktionalitäten erweitert, so dass die Anforderungen, die aus dem Pflichtenheft hervorgehen, erfüllt wurden. Hier sei nochmal auf das Benutzerhandbuch (Vgl. Kapitel A ab Seite 54) hingewiesen, das die Ergebnisse anschaulich wiedergibt. Dieses wurde ebenfalls während des zweiten Semesters geschrieben und im Laufe des Semesters fortlaufend aktualisiert.

## 2.2 Projektinfrastruktur

### 2.2.1 Electronic Workspace

Zu Beginn der Projektgruppe wurde vor allem zur Kommunikation der Projektgruppe untereinander und zum Austausch von Wissen die E-Learning-Plattform der TU Dortmund namens Electronic Workspace (EWS)<sup>14</sup> genutzt. Im Wiki wurden so z.B. die Kontaktdaten und Fähigkeiten der PG-Teilnehmer gesammelt. Im Dateibereich wurden die Präsentationen des Kick-Offs abgelegt. Die Mailingliste wurde zur Kommunikation der PG zwischen den Treffen genutzt. Gerade mit Blick auf die anstehende Entwicklungsarbeit erwies sich Electronic Workspace als unzureichend für die weitere Projektgruppenarbeit. Lediglich die Mailingliste wurde über die gesamte Projektlaufzeit verwendet.

### 2.2.2 Redmine

Ergänzend zum EWS hat das ITMC der Projektgruppe die Mitbenutzung seiner Redmine-Installation ermöglicht. Dazu wurde ein eigenes Projekt für

---

<sup>14</sup><http://www.ews.tu-dortmund.de>

die PG angelegt. Die Informationen aus dem EWS wurden hierher ins Wiki oder SVN übernommen. Redmine<sup>15</sup> brachte vor allem einen SVN-Dienst sowie ein Ticket-System mit sich, die von der Projektgruppe – neben dem Wiki – am meisten genutzt worden sind.

### 2.2.3 Eclipse und Netbeans

Auf Clientseite wurde mit Eclipse<sup>16</sup> oder Netbeans<sup>17</sup> oder teilweise vim<sup>18</sup> gemeinsam ein Java-Projekt entwickelt (Vgl. Kapitel 4 auf Seite 12). Anfangs wurde der Schwerpunkt auf Eclipse gelegt, später hingegen nutzten Teilnehmer ihre jeweils favorisierte Entwicklungsumgebung.

### 2.2.4 L<sup>A</sup>T<sub>E</sub>X

Während im Alltag aus Gewohnheit zur Protokollierung von z.B. Teamsitzungen meist einfach auf übliche Office-Produkte wie Microsoft Office oder OpenOffice.org zurückgegriffen wurde, kam zur Dokumentation von Cloudi, insbesondere beim Benutzerhandbuch, Entwicklerhandbuch, Zwischenbericht und dem hier vorliegenden Abschlußbericht L<sup>A</sup>T<sub>E</sub>X zum Einsatz. In Verbindung mit dem SVN ergibt sich insbesondere der Vorteil, gemeinsam an umfangreichen Dokumenten wie z.B. den Berichten arbeiten zu können.

### 2.2.5 Cloud Infrastruktur Manager

Zu Testzwecken hat die Projektgruppe neben der reinen Entwicklungsinfrastruktur noch zusätzlich einen Server erhalten. Nach der Entscheidung auf OpenNebula zu setzen stellte sich heraus, dass die Installation von OpenNebula mitsamt einer von der Maschine und von Ubuntu unterstützten Virtualisierung auf dieser Maschine mangels CPU-Unterstützung für KVM nicht funktionierte. Aus diesem Grund und auf Grund des Testens der Migration und des Scheduling wurden noch zwei weitere Maschinen zur Verfügung gestellt, auf denen die virtuellen Maschinen dann ausgeführt worden sind.

---

<sup>15</sup><http://www.redmine.org>

<sup>16</sup><http://www.eclipse.org>

<sup>17</sup><http://netbeans.org>

<sup>18</sup><http://www.vim.org>

## 3 Anforderungsanalyse

Zu Beginn der Projektgruppe definieren die Teilnehmer gemeinsam mit den Betreuern - auf Basis des Projektgruppenantrags - in einem Pflichtenheft verschiedene funktionale und nichtfunktionale Anforderungen, die während der Projektzeit als Ziele und Motivation dienen.

### 3.1 Funktionale Anforderungen

Die funktionalen Anforderungen teilen sich in drei Bereiche auf, welche benutzerspezifische Funktionen, administrative Funktionen und Funktionen bzgl. der Verwaltung von virtuellen Maschinen umfassen. Essentiell für den Zugriff auf die Funktionen für Nutzer ist es, dass diese das System erst verwenden können, wenn sie unter Angabe eines Benutzernames, ihrer Mailadresse, eines Passworts und allen für Abrechnungen notwendigen Daten (z.B. Fakultät) registriert und am System angemeldet sind. Alternativ zur kompletten Neuregistrierung für Mitglieder einer anderen akademischen Einrichtung als der TU Dortmund, soll es möglich sein, sich als Universitätsmitglied auch über das myITMC-Portal des Rechenzentrums der TU Dortmund unter Angabe der Uni-Account-Kennung bei Cloudi zu authentifizieren. Die Anmeldung erfolgt jeweils unter Angabe des Benutzernamens/Accountnames, sowie des zugehörigen Passworts. Jederzeit kann sich ein Nutzer auch wieder vom System abmelden. Alle nutzerspezifischen Daten (persönliche Daten, Konfigurationen, verwendete virtuelle Maschinen), die im System gespeichert sind, können immer angezeigt werden. Zusätzlich soll der angemeldete Benutzer die Möglichkeit haben, seine Daten zu ändern und in der Datenbank zu speichern.

Der Administrator ist ebenfalls ein Nutzer, der aber für den geregelten Ablauf des Systems verantwortlich ist und deswegen zusätzlich zu den normalen Nutzerfunktionen die Eigenschaften des Systems konfigurieren bzw. verändern kann. Ein Administrator kann außerdem Statistiken und Protokolle zur Benutzung des Systems einsehen, sowie Funktionen anderer Benutzer ausführen. Damit ist sicherzustellen, dass kritische Operationen überwacht, sowie fehlerhafte oder schädigende Aktionen seitens der Nutzer vom Administrator korrigiert werden können. Desweiteren steht, neben dem Anzeigen aktueller Statistiken zum Ressourcenverbrauch, die Möglichkeit zur Verfügung, Simulationen und Abschätzungen für den zukünftigen Ressourcenbedarf durchzuführen. Eine Anzeige der aktuellen Hostserver, die virtuelle Maschinen ausführen, soll ebenso verfügbar sein, wie die Anzeige und Änderung der aktuellen Partitionierung von Hardware zu virtuellen

Maschinen (Migration von virtuellen Maschinen auf andere Server im Cluster).

Die Kernfunktionalitäten des Portals bestehen in der Verwaltung von virtuellen Maschinen. So stehen dem Nutzer zu jeder Zeit eine Liste seiner angelegten virtuellen Maschinen zur Verfügung. Das Anlegen von virtuellen Maschinen muss über einen Assistenten erfolgen, aus dem eine Template/Image-Kombination hervorgeht (für den Administrator ermöglicht dieses Werkzeug eine detailliertere Parametrisierung als für den normalen Nutzer). Schließlich wird die virtuelle Maschine aus dieser Vorgabe erzeugt. Nutzer sollen u.a. in der Lage sein, hier die Anzahl der virtuellen Prozessoren, den Arbeitsspeicher und die Festplattengröße anzugeben. Virtuelle Maschinen sollen auch wieder gelöscht werden können. Es soll desweiteren möglich sein, angelegte virtuelle Maschinen sofort oder zu bestimmten Terminen zu starten und zu stoppen, sowie Anzahl der virtuellen Prozessoren und Arbeitsspeicher umzukonfigurieren. Eine Migrationsmöglichkeit ist ebenfalls vorgesehen. Zu jeder virtuellen Maschine können Statistiken bzgl. CPU-Last, Arbeitsspeicherauslastung, als auch Netzwerklast angezeigt werden.

### **3.2 Nichtfunktionale Anforderungen**

Unter den nichtfunktionalen Anforderungen gilt es in erster Linie die Robustheit und Zuverlässigkeit von ClouDi zu gewährleisten. Darunter sollte eine hohe Verfügbarkeit, Benutzerfreundlichkeit und hohe Effizienz erreicht werden. Weiterhin soll die Bedienung des Webportals mit gängigen Browsern wie Internet Explorer und Firefox möglich sein. Die BITV [2] (Barrierefreiheit) und LGG NRW (geschlechtergerechte Sprache) sollten Anwendung finden und das Produkt sollte unter die Lizenz GPL gestellt werden. Da das Projekt mit Ende des Wintersemesters 2010/2011 endet sollte die Projektgruppe bis dahin einen lauffähigen Prototypen bereitstellen, welcher die funktionalen Anforderungen erfüllt. Auch sollte ein Endbericht zur Verfügung gestellt werden.

## 4 Architektur

Die Architektur von Cloudi ist in verschiedene Schichten (Layers) unterteilt, um einen besseren Überblick über die Strukturierung der Komponenten, sowie den Informationsaustausch zwischen den Schichten zu bekommen (siehe Abbildung 2). Die oberste Schicht repräsentiert die Clients, die das Cloudi-Portal nutzen. Darunter befindet sich eine Schicht (Connection and Authentication Layer), die sich um die Authentifizierung der Clients beim ITMC kümmert und um die Verbindung mit dem Applikationsserver des Portals über einen vorgeschalteten Web-Server. Diese Schichten kommunizieren sowohl untereinander, als auch mit der folgenden Anwendungsschicht (Application Layer) über das Hypertext Transfer Protocoll (HTTP). Die Anwendungsschicht ist die zentrale Steuerungseinheit von Cloudi und kommuniziert über diverse Schnittstellen und Werkzeuge mit der Middleware-Schicht (Middleware-Layer). Diese Schnittstellen sorgen für das Ausführen von entfernten Methoden, objektrelationales Mapping oder konsolenbasiertes Ausführen von Skripten. Die Anwendungsschicht delegiert und stößt Prozesse an, welche die Middleware-Schicht ausführt und deren Resultate sie zurückmeldet. Die unterste Schicht in dieser Betrachtung ist dann die Virtualisierungsebene (Virtualization Layer) auf der die virtuellen Maschinen betrieben werden.

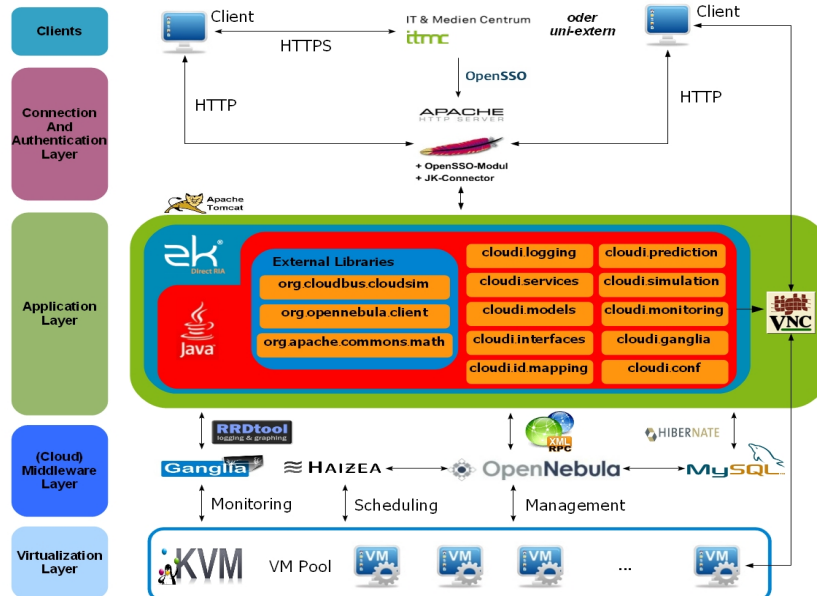


Abbildung 2: Die Architektur von Cloudi

Die Anwendungsschicht bildet eine Java-Webapplikation, welche das ZK-Framework<sup>19</sup> nutzt. Dieses Framework unterstützt hauptsächlich die unkomplizierte Erzeugung der Weboberfläche (Frontend). Es befindet sich seit 2000 in der Entwicklung und ist komplett in Java geschrieben. Seitdem es auf dem Markt verfügbar ist, setzen es 500 Firmen und Organisationen wie z.B. T-Systems, Oracle, Toyota oder die Vereinte Nationen in über 1,3 Millionen Projekten ein. Die Basis bildet ein intuitives ereignisgesteuertes und komponentenbasiertes Modell. Außerdem erfordert die Präsentationsschicht keinen clientseitigen JavaScript-Code, was zusätzlich zur Sicherheit beiträgt. Für die Entwicklungsumgebung Eclipse ist ein Plugin verfügbar, das einen gut zu bedienenden visuellen Editor beinhaltet. Die Entscheidung ZK einzusetzen ist auf die Erfahrungen einiger Teilnehmer mit diesem Framework aus der Vorlesung Webtechnologien I zurückzuführen, welche die schnelle Prototypenentwicklung mit diesem Framework herausstellt. Als Alternative zu ZK ist der Einsatz des Google Web Toolkit (GWT) denkbar, das ebenfalls die Entwicklung von Rich Internet Applications (RIA) unterstützt, aber einen clientzentrierten Ansatz vorgibt. Bei ZK hingegen verbleiben Zustands- und Ablauflogik auf dem Server, was besonders bei sicherheitskritischen Operationen von Vorteil ist.

Das vorherrschende Design-Pattern in der Umsetzung des Backends der Webanwendung ist das MVC-Prinzip (Model View Control Pattern). Die Control-Komponenten bieten auch durchaus die Möglichkeit sie als Web Services zu veröffentlichen. Dementsprechend tragen sie in der Implementierung die Bezeichnung Service. Diese Komponenten sind Java-Klassen, die teilweise auf etablierte externe Bibliotheken zurückgreifen. Die Klassen, welche die Repräsentation der Entitäten wie virtueller Maschinen, Hostserver und Middleware-Benutzer darstellen, sind in `cloudi.models` zusammengefasst. Im Paket `cloudi.interfaces` befinden sich die abstrakten Schnittstellen-Klassen, die implementiert werden, um das Portal mit einer Cloud-Middleware zu verbinden. Die konkreten Implementierungen der Interfaces `IVMService`, `IHostService` und `IUserService` aus `cloudi.interfaces` für die Kommunikation mit Open Nebula sind bereits im Paket `cloudi.services` zu finden. Die OpenNebula-Client-Bibliothek ist hier schon seit dem ersten Halbjahr im Einsatz und kapselt die Kommunikation mit der Middleware in Remote Procedure Calls. Mit Hilfe der Klassen aus `cloudi.monitoring` und `cloudi.ganglia` gelingt die Datenextraktion, Verarbeitung, Auswertung und Bereitstellung

---

<sup>19</sup><http://www.zkoss.org>

der Monitoring-Daten aus dem Ganglia Monitoring System<sup>20</sup>. Daten wie die IP-Adressen der OpenNebula- und Ganglia-Server, XML-RPC-Port und Administratorkennwörter zu Konfigurationszwecken bzgl. der Kommunikationsprozesse zwischen Web-Applikation und Middleware sind als Attribute im Paket `cloudi.conf` hinterlegt. Die Implementierung bietet eine Installationsklasse(`cloudi.conf.Installation`) an, um diese zur Konfiguration notwendigen Parameter aus einer ini-Datei zu übernehmen. Bis zum vorläufigen Entwicklungsstopp - zum Ende der Projektgruppe - werden die Parameterwerte jedoch im Entwicklungsprozess manuell in den Klassen `cloudi.conf.OneConfig` und `cloudi.conf.GangliaConf` modifiziert und durch erneute Kompilation des Projekts übernommen. Alle Klassen, die für die Durchführung von Simulationen und damit zur Abschätzung des zukünftigen Ressourcenbedarfs benötigt werden, sind in die Pakete `cloudi.simulation` und `cloudi.prediction` aufgeteilt. Als Basis für die Simulationen wird die Bibliothek CloudSim<sup>21</sup> eingesetzt, die speziell für den Einsatz diskreter Event-Simulationen in Cloud-Umgebungen entwickelt wird. Für die Auswertungen zur Ressourcenvorhersage wird auf die mathematische Funktionsbibliothek Apache Commons Mathematics zurückgegriffen, um lineare Regression zu berechnen. Um die Operationen, die der Nutzer im Portal ausführt, zu protokollieren, ist in `cloudi.logging` ein Mechanismus implementiert, der zu jeder dieser Operationen Datum, Uhrzeit, Operationsname, betroffene Entitäten/Parameter (z.B. ID einer virtuellen Maschine) sowie ausführende Person in der Datenbank speichert. Im Portal findet sich für die Administratoren hierzu eine Oberfläche, die das Durchsuchen des Logs nach Zeiträumen ermöglicht und dann die Sortierung der Resultate nach Benutzern und Aktionen erlaubt. Um verschiedene Middlewares mit Cloudi zu nutzen, sind alle Entitäten mit portalinternen IDs ausgestattet. Die Klassen im Paket `cloudi.id.mapping` sorgen dafür, dass die intern vergebenen Cloud-IDs auf die der Middleware abgebildet werden. Es könnten beispielsweise gleiche IDs für virtuelle Maschinen bei Verwendung von Open Nebula und Tashi vergeben werden. Um eben diese zwei virtuelle Maschinen auch im Portal zu unterscheiden, existiert ein internes Datenmodell, das durch Anpassungen der Java-Klassen `VmID`, `HostID`, `UserID` aus dem Paket `cloudi.id.mapping` nachträglich um Attribute erweiterbar ist. Momentan werden in diesen Klassen ausschließlich eine interne ID und eine ID für die jeweilige Entität der Middleware (bei Cloudi momentan nur Open Nebula) gespeichert.

---

<sup>20</sup><http://ganglia.sourceforge.net>

<sup>21</sup><http://www.cloudbus.org/cloudsim>

Die Webapplikation beinhaltet den javabasierten tightVNCViewer. VNC steht für Virtual Network Computing. Mit dieser Technik wird die grafische Ausgabe von entfernten Rechnern zu anderen Clients übertragen, während die Client-Eingaben an den entfernten Rechner gesendet wird. Das Cloudi-Portal ermöglicht die Übergabe der notwendigen Parameter für den Verbindungsaufbau vom Internet-Browser zum VNC-Server, der für die Verbindung zu der jeweiligen virtuelle Maschine verantwortlich ist. Dadurch kann der Nutzer sofort auf die Konsole der virtuellen Maschine zugreifen und bereits den Bootvorgang mitverfolgen.

Neben der normalen Benutzeranmeldung und Registrierung am Web-Portal ist auch eine Anbindung an das Identity-Management des ITMC per OpenSSO<sup>22</sup> realisiert. Beim Zugriff auf die Adresse des Prototypen erfolgt durch den Apache HTTP Server<sup>23</sup> eine Weiterleitung auf das myITMC-Portal. Nach der Anmeldung über myITMC und der Weiterleitung auf den Prototypen ist es möglich, die Daten zur Nutzererkennung aus dem mitgesendeten Header auszulesen und in eine an das Portal angebundene Datenbank zu übernehmen.

Die Webapplikation ist über verschiedene Schnittstellen und Tools mit der Software aus der Middleware-Schicht verbunden. Für die Kommunikation mit der Cloud-Middleware OpenNebula wird die XML-RPC Schnittstelle verwendet, die in der OpenNebula-Java-API gekapselt ist. Denkbar ist hier auch statt XML-RPC die – im Laufe der Standardisierungsbestrebungen des OGF(Open Grid Forum) entwickelte – OCCI Schnittstelle einzusetzen. Die direkte Verwaltung der virtuellen Maschinen wird von OpenNebula übernommen.

Zur Unterstützung des Monitorings wird das Ganglia Monitoring System eingesetzt. Dieses System ist ein weit verbreitetes Open-Source-Werkzeug, das bei Cloudi nicht nur die Überwachung der Server, sondern auch der virtuellen Maschinen übernimmt (für ausführlichere Informationen über Ganglia und Gründe warum diese Software eingesetzt wird, sei auf Kapitel 6 verwiesen). Hierfür muss allerdings in allen virtuellen Maschinen der Ganglia Monitor laufen. Unter Linux-Betriebssystemen sind meistens dafür Pakete in den Distributionen verfügbar. Unter Windows wird Ganglia separat installiert. Ganglia speichert die Daten in speziell dafür vorgesehenen Round-Robin-Datenbanken, kurz RRD. Als Schnittstelle zwischen der Web-

---

<sup>22</sup><http://opensso.dev.java.net>

<sup>23</sup><http://httpd.apache.org>

applikation und diesen Datenbanken dient RRDtool<sup>24</sup>. Mit RRDtool lassen sich Daten per „fetch“-Befehl leicht extrahieren oder direkt anschauliche Graphen erzeugen („createGraph“-Funktion).

Um das Scheduling weiter zu optimieren, mehr Einfluss auf die zu beachtenden Parameter zu haben sowie termingerechtes Scheduling anzubieten, wird der OpenNebula interne Scheduler durch Haizea ersetzt (Mehr Informationen zu Haizea in Kapitel 5.6). Durch Hinzufügen von Informationen wie Start- und Endzeitpunkt oder Preemption-Flag einer virtuellen Maschine wird so besser auf das Scheduling Einfluss genommen.

Zur dauerhaften Speicherung von aufkommenden Daten während des Betriebs von Cloudi ist ein MySQL-Server im Einsatz. Per Java Persistence API (JPA) werden die Objekte der Webapplikation hier persistiert. Im Laufe der Projektlaufzeit ist OpenNebula 2.0 veröffentlicht worden. OpenNebula 2.0 kann seine Daten direkt in MySQL speichern<sup>25</sup>. Dadurch werden die IDs der virtuellen Maschinen, Hosts und OpenNebula-Benutzer komfortabel in einer Tabelle im MySQL zusammengefasst.

## 4.1 Voraussetzungen für den Einsatz von Cloudi

Durch die im vorherigen Abschnitt beschriebene Architektur lassen sich die Hardware- und Software-Voraussetzungen für den Einsatz von Cloudi nun konkretisieren. Ein Handbuch zur Installation von Cloudi findet sich im Anhang.

### 4.1.1 Hardware-Voraussetzungen

Es wird ein zentraler Anwendungsserver benötigt, der leistungsfähig genug sein sollte, um auf ihm einen Web-Server, einen Datenbank-Server und eine Cloud-Middleware im Dauerbetrieb auszuführen. Hinzu kommen noch beliebig viele weitere Server, auf denen die virtuellen Maschinen ausgeführt werden. Bei diesen ist darauf zu achten, dass der Prozessorchipatz Virtualisierungstechnologien wie Intel-VT oder AMD-V unterstützt. Dies ist eine Voraussetzung der eingesetzten Virtualisierungssoftware KVM.

<sup>24</sup><http://www.mrtg.org/rrdtool>

<sup>25</sup><http://opennebula.org/documentation:rel2.0:mysql>

### 4.1.2 Software-Voraussetzungen

Cloudi wird als Web-Archiv(WAR) verteilt und wird einfach auf einem Java-kompatiblen Web-Container, wie zum Beispiel Apache Tomcat<sup>26</sup> deployt. Der zentrale Anwendungsserver sollte ein Linux-Betriebssystem verwenden, da die aktuellen Installationen von OpenNebula und des Ganglia Monitoring System die beste Paket- und Entwicklungsunterstützung für die Linux-Distributionen von OpenSuse, Ubuntu oder Debian haben. Generell bieten die in der Projektgruppe betrachteten Cloud-Middlewares Open Nebula, Eucalyptus und Tashi momentan noch keine Windows-Installationen an. Außerdem wird noch eine SQL-kompatible Datenbank benötigt. Das Auslesen der Monitoring-Daten funktioniert am Besten, wenn sich Cloudi und der Ganglia-Server auf der gleichen Maschine befinden. Bei virtuellen Maschinen, die Cloudi überwacht, ist die Installation des Ganglia-Clients in deren Image obligatorisch. Die Hosting-Server laufen ebenfalls mit Linux-Betriebssystem. Auf ihnen muss die Open Nebula Cluster-Node Software installiert sein, inklusive libvirt und KVM zum Ausführen der virtuellen Maschinen. Da OpenNebula die virtuellen Maschinen mit dem Umweg über libvirt realisiert, kann als Virtualisierungslösung statt dem KVM Linux hypervisor<sup>27</sup>, der Xen hypervisor auf Linux und Solaris Hosts<sup>28</sup>, der QEMU emulator<sup>29</sup>, die VMware ESX, GSX, Workstation und Player Hypervisoren in Frage. Die vollständige Liste findet sich auf der libvirt-Homepage<sup>30</sup>. Getestet wurde Cloudi allerdings ausschließlich mit KVM.

Die Anbindung an das ID-Management des ITMC erfolgt über den OpenSSO-Service des ITMCs. Da ein OpenSSO-Modul lediglich für den Apache HTTP Server und nicht für den Apache Tomcat existiert, muss der Apache HTTP Server noch so konfiguriert werden, dass er sämtliche Aufrufe auf den Prototypen an den Apache Tomcat weiterleitet und die Antwort des Tomcat als eigene Antwort dem Internetbrowser zurückzuliefert. Nur dadurch ist es möglich das OpenSSO-Modul für den Apache HTTP Server mit der Java-Applikation Cloudi im Apache Tomcat zu kombinieren. In der prototypischen Implementierung wurde der Connector mod\_jk<sup>31</sup> eingesetzt, der im Apache HTTP Server als Modul geladen wird.

---

<sup>26</sup><http://tomcat.apache.org>

<sup>27</sup><http://linux-kvm.org>

<sup>28</sup><http://www.xen.org>

<sup>29</sup><http://www.qemu.org>

<sup>30</sup><http://libvirt.org>

<sup>31</sup><http://tomcat.apache.org/connectors-doc>

## 5 Grundgerüst

Das technische Grundgerüst für ClouDi entwickelt sich als Folge der Evaluation einer Reihe geeigneter Tools. Die wichtigsten Entscheidungen bezüglich der eingesetzten Lösungen werden in diesem Kapitel dargelegt. Bei der Wahl zwischen zwei fast gleichwertigen Produkten werden die Vor- und Nachteile abgewägt.

### 5.1 Wahl des Hypervisors

Eine grundlegende Entscheidung betrifft die zu unterstützende Virtualisierungsinfrastruktur, genau genommen dem Hypervisor für virtuellen Maschinen. Zur Auswahl steht die Kernel-based Virtual Machine (KVM), Xen und ESX<sup>32</sup> von VMWare.

KVM ist eine Virtualisierungsinfrastruktur, die direkt in den Linuxkernel integriert wurde, und läuft lediglich auf x86-Prozessoren mit den Virtualisierungstechniken von Intel oder AMD (c.o. Intel VT bzw. AMD V) sowie auf der System-z-Architektur. KVM wurde im Oktober 2006 veröffentlicht und ist seit der Kernelversion 2.6.20 im Linux-Kernel enthalten. Es wurde von dem israelischen Unternehmen Qumranet entwickelt und steht dort unter Beaufsichtigung von Avi Kivity. Qumranet wurde im September 2008 von Red Hat aufgekauft<sup>33</sup>. Dadurch hat es gute Chancen in Zukunft sich als Standard zu etablieren. Bestandteile von KVM sind die Kernel-Module `kvm.ko`, sowie die hardware-spezifischen Module `kvm-intel.ko` oder `kvm-amd.ko`. KVM unterstützt vollständige Virtualisierung und läuft auf einem Betriebssystem und ist damit ein Typ-2-Hypervisor [3] (nutzt die Treiber des Betriebssystems). Vollständige Virtualisierungslösungen bieten einer virtuellen Maschine eine vollständige Umgebung inklusive eigenem BIOS: Jedes Gastbetriebssystem hat einen eigenen virtuellen Rechner mit virtuellen Ressourcen wie CPU, Hauptspeicher, Laufwerken, Netzwerkkarten usw. zur Verfügung. Nach dem Laden des Moduls arbeitet der Linux-Kernel selbst als Hypervisor für virtuelle Maschinen. Als Gastsysteme unterstützt KVM Linux (32 und 64 Bit), Windows (32 und 64 Bit), Haiku OS, AROS, ReactOS, FreeDOS, Solaris und diverse BSD-Derivate. KVM läuft auch auf SMP<sup>34</sup>-Hostsystemen. SMP-Gastsysteme sind ebenfalls möglich. Die Unterstützung für Paravirtualisierung ist mittlerweile in KVM vorhanden.

---

<sup>32</sup><http://www.vmware.com/de/products/vi/esx>

<sup>33</sup><http://www.redhat.com/promo/qumranet>

<sup>34</sup>Symmetrisches Multiprozessorsystem

Xen ist heutzutage vor allem im Serverbereich durch die bekannten Citrix Xenserver weit verbreitet. Es läuft auf fast jeder aktuellen Hardware, auch ohne Virtualisierungsfeatures. Als Gastbetriebssysteme werden unterstützt: Linux, Windows, Solaris, BSD, FreeBSD, und andere. Es ist inzwischen sehr ausgereift und sehr stabil im Einsatz vieler Unternehmen. Vor allem bietet es hochgradige Gastisolation, sowie Nutzung von Hardware-Sicherheitsfeatures (z.B. TPM).

ESX von VMware setzt vor allem auf vollständige Virtualisierung, (setzt Intel VT/AMD-V voraus). Es bietet genauso viele Vorteile wie KVM und Xen, hat aber den Nachteil, dass es nicht OpenSource ist und somit kostenpflichtig ist. Der kleine Bruder ESXi aber dagegen ist OpenSource, hat geringe Hardwareanforderungen und arbeitet mit gängiger 32 und 64 Bit-Hardware.

Eine echte Entscheidung für einen Hypervisor kann es hier nicht geben, da Ziel ist möglichst alle Plattformen zu unterstützen. Dieser Faktor muss bei der Wahl des Manager der Virtualisierung-Infrastruktur berücksichtigt werden. Auf den Hosts, die für die PG zur Verfügung gestellt werden, läuft eine Linux-Distribution (SLES 11). Dort lässt sich KVM einfach durch den Paketmanager nachinstallieren. Xen wird dort offiziell nicht mehr unterstützt, es muss umständlich manuell nachinstalliert werden, was sehr problematisch ist. Somit bleibt es bei KVM, aber ein Wechsel auf Xen oder ESX sollte aber ohne viel Aufwand möglich sein.

## 5.2 Manager für die VM-Infrastruktur

Wie bereits im letzten Abschnitt beschrieben, fehlt auf der untersten Ebene für Cloudi ein Manager für die VM-Infrastruktur. Die Open Source-Variante von Eucalyptus<sup>35</sup> und OpenNebula<sup>36</sup> kamen als einzige in Frage, da Open Source ein entscheidendes Kriterium bei der Auswahl der einzusetzenden Softwarelösungen ist. Eucalyptus und OpenNebula unterscheiden sich lediglich in wenigen Punkten: OpenNebula implementiert eine administrative Virtualisierungskontrolle und enthält dabei Cloud-Semantiken (EC2 Query API und der OGF OCCI-API). Eucalyptus basiert hingegen nur auf Cloud-Semantiken, hier können mehrere Benutzer administrative Aufgaben durchführen. Bei Eucalyptus hat jeder Nutzer Kontrolle über seine eigene Cloud. Bei OpenNebula ist der Administrator dafür zuständig.

---

<sup>35</sup><http://open.eucalyptus.com>

<sup>36</sup><http://opennebula.org>

Diese Details sind für den Einsatz in Cloudi allerdings nicht entscheidend. Folgende Kriterien sind mit Blick auf Cloudi eher relevant:

- Bei der Virtualisierungstechnik hat OpenNebula den Vorteil als Hypervisor nicht nur Xen und KVM zu unterstützen, sondern auch die Hypervisoren von VMware. Dies ist bei Eucalyptus in der Open Source-Variante nicht möglich. Da im ITMC bisher VMware eingesetzt wird, ist für eine möglichst breite Unterstützung somit OpenNebula vorteilhafter.
- Beim Image Management unterscheiden sich OpenNebula und Eucalyptus kaum, beide enthalten die Transferfunktion von VM-Images. D.h. in der Praxis, dass VM-Images aus einem Image-Repository zu dem entsprechenden Host transferiert werden können. Mit OpenNebula lassen sich beim Virtual Network Management von einander isolierte virtuelle Netze definieren, über die virtuelle Maschinen untereinander verbunden werden können. OpenNebula kann dabei insbesondere die Vergabe der IP- und MAC-Adressen übernehmen. Bei Eucalyptus fehlt leider jegliche Möglichkeit für das Verwalten von virtuellen Netzwerken. Ein wichtiger Kernpunkt für die Entwicklung von Cloudi und damit das entscheidende Kriterium ist das Scheduling. Eucalyptus bot zum Zeitpunkt der Entscheidung keinen eigenen Scheduler an. Inzwischen gibt es einen Scheduler für Eucalyptus der zwei Scheduling-Modi (GREEDY und ROUNDROBIN)<sup>37</sup> bietet, aber keine Schnittstelle um eigene Scheduler einsetzen zu können. OpenNebula hingegen bietet standardmäßig zwar auch nur einen Scheduler (match-making-Scheduler; siehe 5.3.1 auf Seite 22), kann allerdings über seine XML-RPC-Schnittstelle auch andere Scheduler wie z.B. Haizea unproblematisch verwenden. Haizea übernimmt dann in diesem Fall das Scheduling und sendet die Befehle (für VM starten, stoppen etc.) an Open Nebula. Ein weiterer Punkt ist das Service Management. OpenNebula bietet eine Unterstützung von sogenannten Multi Tier Services bestehend aus Gruppen von miteinander verbundenen virtuellen Maschinen (VMs können nochmal zu einem Cluster gruppiert werden) und deren Auto Konfiguration während des Bootvorgangs. Das ist nur möglich wenn die VM nicht sofort gestartet wird, sondern nach dem Erstellen in den sog. *deployed*-Status versetzt wird. In diesem Zustand können die Werte noch verändert werden, aber die Zuweisung und der Transfer der VM-Image zu einem Host ist bereits abgeschlossen. Mit Eucalyptus ist es hingegen nicht möglich

---

<sup>37</sup><http://open.eucalyptus.com/wiki/>

virtuelle Maschinen während des Bootvorgangs zu konfigurieren. Das erschien zwar nur als kleines Details, wovon allerdings unsicher war, ob es nicht im Laufe der Projektlaufzeit noch hilfreich sein könnte. OpenNebula und Eucalyptus unterstützen beide Hybrid Clouds durch das EC2 Interface.

- *Live-Migration* wird innerhalb von OpenNebula unkompliziert unterstützt, sofern es sich um die gleiche Ziel- und Quellinfrastruktur handelt. Somit müssen virtuelle Maschinen bei Wartungsarbeiten an einem Rechner o.ä. nicht heruntergefahren werden, um einen physikalischen Rechner abschalten zu können. Eucalyptus unterstützt Live-Migration bis dato noch nicht.

OpenNebula bietet folglich mehr Features (Vgl. dazu auch Abb. 3), auch die Dokumentation für Nutzer und Entwickler ist besser gepflegt als bei Eucalyptus [4] [5]. Darum wird für den Einsatz in Cloudi primär auf OpenNebula gesetzt. Gleichzeitig ist in Cloudi allerdings die Kommunikation zu OpenNebula derart gekapselt (durch entsprechende Interfaces im Quellcode), dass OpenNebula vergleichsweise einfach durch andere Middlewares oder Eucalyptus ersetzt werden könnte.

Management der Infrastruktur

Features	Eucalyptus	Open Nebula
Virtualization Management	Xen, KVM	Xen, KVM, VMware
Virtual Network Management	Nein	Ja
<i>Scheduling</i>	<i>Nein</i>	<i>Ja</i>
Hybrid Cloud Computing	Nein	Ja
Cloud Interface	EC2	EC2 Query
Live-Migration	Nein	Ja

Abbildung 3: Vergleich zwischen OpenNebula und Eucalyptus

## 5.3 OpenNebula

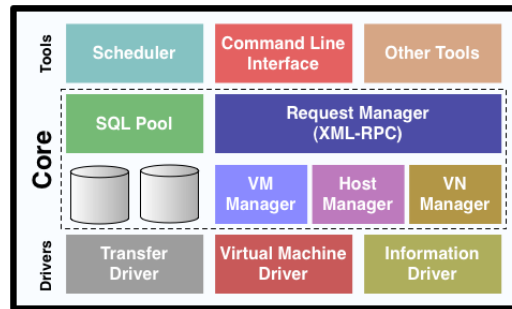


Abbildung 4: OpenNebula Architektur

OpenNebula (ONE) ist ein „Virtual Infrastructure Manager“, der in die drei Schichten „Tools“, „Core“ sowie „Drivers“ aufgeteilt ist (Vgl. Abb. 4).

### 5.3.1 Tools

Diese Schicht enthält den Scheduler sowie Schnittstellen zur Kommunikation mit OpenNebula. Dazu gehört das Command Line Interface (CLI) von OpenNebula, aber auch andere Tools von Fremdanbietern, wie z.B. die libvirt-API<sup>38</sup> oder externe Scheduler. Der OpenNebula-Scheduler ist eine austauschbare Einheit innerhalb der OpenNebula Architektur. Der Basis-Scheduler ist ein sogenannter *match-making-Scheduler*, implementiert ist eine sogenannte *Rank-Scheduling-Policy*. Das Ziel hierbei ist es den passenden Host für eine VM anhand von vorgegebenen Kriterien zu finden. Der Algorithmus funktioniert folgendermaßen:

1. Jeder Host, der nicht genügend RAM und CPU für die virtuelle Maschine vorweist, wird aussortiert.
2. Evaluierung der *Rank Expression* im *VM-Template* zur virtuellen Maschine. Ein *VM-Template* ist eine Definitionsdatei einer virtuellen Maschine. Dort wird angegeben wie viele CPU's, Arbeits-, Festplatten-speicher etc. die VM haben soll.
3. Die virtuelle Maschine wird einer Ressource mit höchstem Rang zugewiesen.

<sup>38</sup><http://libvirt.org>

Die *Rank Expression* bietet drei Verteilungsstrategien an, durch die entschieden wird, wie die virtuelle Maschinen auf die Hosts verteilt werden. Die *Rank Expression* wird über das jeweilige *Template* explizit für jede virtuelle Maschine festgelegt.

- Rank= RUNNING-VMS  
Falls diese *Rank Expression* im *Template* angegeben wird, wird folgendermaßen vorgegangen: Die Anzahl der Clusterknoten, die genutzt werden, wird minimiert. Hierbei wird so verfahren, dass möglichst viele virtuelle Maschinen auf einen Host ausgeführt werden. Das reduziert die Fragmentierung im Cluster.
- Rank= -RUNNING-VMS  
Falls diese *Rank Expression* gesetzt wird, werden die Ressourcen in einem Clusterknoten, die für eine virtuelle Maschine verfügbar sind, maximiert. Dabei werden die virtuellen Maschinen in den Clusterknoten verteilt, auf denen bisher am wenigsten virtuelle Maschinen gestartet wurden. Das hat den Vorteil, dass eine VM mit hohen Hardware-Anforderungen auch auf einem gut ausgelasteten Cluster Ressourcen findet, damit sie ausgeführt werden kann.
- Rank= FREECPU  
Bei dieser Strategie wird auf Grundlage der CPU-Auslastung verteilt. Eine virtuelle Maschine wird also dort ausgeführt, wo der Clusterknoten die geringste CPU-Auslastung vorweist.

### 5.3.2 Core

Zu den Kernkomponenten von OpenNebula gehört der Request Manager. Dieser stellt eine XML-RPC-Schnittstelle<sup>39</sup> bereit. Der VM-Manager ist für die Verwaltung und das Monitoring der virtuellen Maschinen zuständig. Der Transfermanager kümmert sich um den Transfer der Images zu den Cluster Nodes sowie in das Image Repository, aber auch den Transfer von Checkpoint-Dateien zwischen mehreren Cluster Nodes, wenn z.B. eine virtuelle Maschine suspendiert ist. Der Host Manager verwaltet und überwacht die physikalischen Hosts bzw. Clusterknoten. Er ist für die gesamte Infrastruktur zur Überwachung der Hosts verantwortlich. Über ihn lassen sich auch neue Hosts z.B. per CLI-Befehl in die bestehende Infrastruktur einbinden. Der Virtual Network Manager (VNM) ist für die Verwaltung der IP- und MAC-Adressen zuständig. Mit dem VNM lassen sich neue virtuelle Netzwerke erstellen und er verteilt die festgelegten IP- und MAC-Adressen an die

---

<sup>39</sup><http://www.xmlrpc.com>

virtuellen Maschinen. In der SQL-Datenbank werden alle relevanten Daten über virtuelle Maschinen, Netzwerke, Hosts und User gespeichert.

### 5.3.3 Drivers

OpenNebula enthält noch eine Vielzahl von Modulen, die Drivers genannt werden. Dabei handelt es sich um Treiber zu verschiedenen Hypervisoren wie Xen, KVM oder VMware, zu Cloud Services wie EC2, Dateiübertragungsmechanismen und anderen Informationsdiensten.

## 5.4 Kommunikationsschnittstelle zwischen VM-Middleware und Cloudi

Neben der Frage der einzusetzenden Software-Lösungen gilt es außerdem über die Kommunikationsschnittstelle zu entscheiden, über die Cloudi mit OpenNebula kommuniziert. Zur Kommunikation mit der Middleware gibt es mehrere Möglichkeiten. Zum einen bietet OpenNebula eine XML-RPC-API. Bei diesen RPCs handelt es sich um eine Technik, bei der Methoden auf entfernten Rechner über ein XML-Format aufgerufen werden können. Auch die Antworten dieser Aufrufe erfolgen wieder im XML-Format. Hierfür gibt es neben der Möglichkeit über die Apache-XML-RPC-Bibliothek für Java, auch eine direkt von OpenNebula bereitgestellte API zu nutzen, die die RPCs kapselt und auch die Auswertungen der Antwort sehr erleichtert. Neben dieser Möglichkeit Daten mit OpenNebula auszutauschen, könnte auch das ebenfalls von OpenNebula unterstützte OCCI-Schnittstelle<sup>40</sup> des OGF<sup>41</sup> (als REST-Service<sup>42</sup> auf HTTP-Basis<sup>43</sup>) oder der Java Message Service (JMS)<sup>44</sup> genutzt werden.

Die Wahl fällt auf die XML-RPC-API, die mithilfe guter Dokumentation schnell implementiert werden kann. OCCI verlangt mehr Einarbeitungszeit und lässt sich nicht mehr zeitlich realisieren, doch ein entsprechendes Interface für weitere Schnittstellen wird in Cloudi zur Verfügung gestellt.

---

<sup>40</sup><http://occi-wg.org>

<sup>41</sup><http://www.gridforum.org>

<sup>42</sup>[http://ics.uci.edu/~fielding/pubs/dissertation/rest\\_arch\\_style.htm](http://ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm)

<sup>43</sup><http://tools.ietf.org/html/rfc2616>

<sup>44</sup><http://java.sun.com/products/jms>

## 5.5 Erstinstantiation

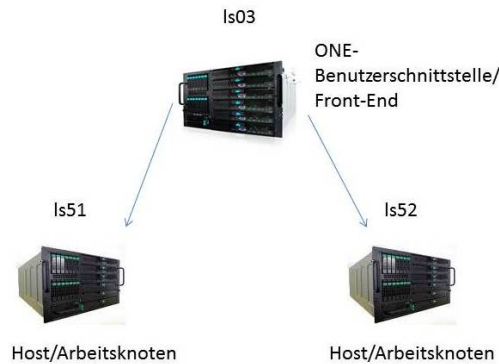


Abbildung 5: Servergrundgerüst

Das Testsystem für das Projekt besteht aus einem Server (ls03) des ITMC, der als Frontend von OpenNebula dient und 2 weiteren Servern (ls51 und ls52), deren CPUs die Virtualisierungstechnik bzw. deren Chipsatz VirtIO unterstützen und darum als Hosts(Clusterknoten) von OpenNebula dienen (vgl. Abb. 5). Die Clusterknoten verfügen jeweils über 8 CPU-Kerne und 24GB RAM, für Testzwecke der PG ist dieses System mehr als ausreichend. Speicherbegrenzungen gibt es nicht, da auf genügend Festplattenspeicher zugegriffen werden kann. Als Betriebssysteme werden Ubuntu-Server 10.04 64 (auf den Frontend ls03), sowie SLES 11 64Bit (auf den Arbeitsknoten, ls51-52) eingesetzt. Ein Versuch komplett auf Ubuntu zu wechseln, schlägt fehl. Der Grund ist, dass die Netzwerkverbindung bei den Hostknoten nicht stabil ist. Der Server ist einfach nicht zu erreichen, ohne das manuell nochmal der Netzwerkservice neugestartet wird. Mit Suse Linux läuft dies problemlos. Bevor die Installation von OpenNebula (Version 1.4) durchgeführt werden kann, sind noch einige Hürden zu nehmen. Es ist notwendig noch einiges an der Konfigurationsdatei von OpenNebula zu ändern, damit OpenNebula überhaupt gestartet wird. Zum Beispiel fehlen noch in der 1.4-Version von ONE Standardeinträge für die MySQL-Datenbank-Anbindung. Bevor OpenNebula installiert werden kann, sind viele zusätzliche Pakete für Linux nötig, sonst schlägt die Kompilierung fehl. In der Dokumentation von ONE sind für die Installation die notwendigen Pakete aufgelistet. Doch das Problem liegt vor allem darin, dass sich bei jeder Linux-Distribution die Pakete unterscheiden (vor allem zwischen Debian- und Redhat-System). Auch viele Paketabhängigkeiten müssen erst nachgeschlagen und manuell nachinstalliert werden.

Um Live-Migration zu ermöglichen, wird den Arbeitsknoten per NFS<sup>45</sup> Zugriff auf das Installationsverzeichnis von OpenNebula auf dem Frontend-Rechner ls03, mitsamt der Festplattenimages, ermöglicht. Hierbei tritt ein Problem bei Suse Linux auf, wenn die von ls03 exportierten Dateisysteme per fstab<sup>46</sup> auf den Clusterknoten eingebunden werden. Beim Booten wurde die fstab bereits zu einem Zeitpunkt ausgelesen und interpretiert, zu dem das Netzwerk anscheinend noch nicht funktioniert. Der Wechsel auf den Dienst Autofs<sup>47</sup> bringt hier eine einfache, pragmatische Lösung des Problems.

Damit OpenNebula die Werte der Clusterknoten (Kernanzahl der CPU, RAM-Größe etc.) richtig auslesen kann, muss darüberhinaus auf sämtlichen Hosts, als Standard-System-Sprache Englisch eingestellt werden.

Die Kommunikation mit den Arbeitsknoten erfolgt, indem sich der oneadmin-Nutzer des ls03-Rechners per SSH<sup>48</sup> auf dem jeweiligen Host einloggt und dort mittels der per NFS und AutoFS importierten Skripte Befehle der libvirt-API ausführt, um eine virtuelle Maschine zu erstellen oder zu manipulieren. Das heißt, der oneadmin-Nutzer muss alle Rechte haben, um auf den jeweiligen Knoten die entsprechenden Befehle wie „virsh“ ausführen zu können. Konkret müssen an den Hosts die libvirt- und KVM-Konfigurationsdateien, Gruppenrechte und SSH-Keys angepasst werden. Trotz dieser Einstellungen können die virtuellen Maschinen immer noch nicht mit OpenNebula gestoppt, pausiert und wieder gestartet werden, weil der oneadmin-Nutzer immer noch nicht alle notwendigen Rechte besitzt. Das Problem ist in der Umask<sup>49</sup> zu finden, die standardmäßig auf einen Wert eingestellt ist, der die Schreib-/Lese-Rechte bei neu erstellten Dateien weiter eingeschränkt als in einer Testumgebung notwendig ist. Durch Änderung des Default-Wertes besteht dieses Problem nicht mehr.

Einen NTP-Server<sup>50</sup> zu nutzen erweist sich im Laufe der Zeit als vorteilhaft. Dadurch können entsprechende Fehlermeldungen in den virtuellen Maschinen vermieden werden, die dadurch verursacht werden, dass das

---

<sup>45</sup><http://tools.ietf.org/html/rfc3530>

<sup>46</sup><http://linux.die.net/man/5/fstab>

<sup>47</sup><http://www.autofs.org>

<sup>48</sup><http://linux.die.net/man/1/ssh>

<sup>49</sup><http://linux.die.net/man/1/umask>

<http://linux.die.net/man/2/umask>

<http://linux.die.net/man/3/umask>

<sup>50</sup><http://www.ntp.org>

Datum im Superblock in der Zukunft liegt.

Während der Projektlaufzeit erscheint OpenNebula 2.0<sup>51</sup> mit vielen für das Projekt hilfreichen Features wie z.B. ein Image Repository, so dass die Projektgruppe dazu entschieden OpenNebula auf die neue Version zu aktualisieren. Insgesamt wird das Testsystem während der gesamten Projektlaufzeit dreimal aktualisiert. Zum Ende der Projektlaufzeit wird OpenNebula 2.01 genutzt. OpenNebula 2.01 erweist sich im Vergleich zu den vorherigen Versionen als recht stabil und einfach konfigurierbar. Inzwischen wird bei den virtuellen Maschinen zwischen 32- und 64-Bit Architektur unterschieden, dies muss jetzt beachtet werden. Falls in der Konfigurationsdatei oder im Template das nicht explizit als 64-Bit angegeben wird, erweisen sich die originalen VM-Images als unbrauchbar (falls die Images nicht explizit auf den Hosts kopiert werden und die original Images unangetastet bleiben).

Im Laufe des Jahres kann die Weiterentwicklung von OpenNebula gut mitverfolgt werden. Verschiedene zusätzliche Sicherheitsansätze werden realisiert, z.B. ist es nun Pflicht, Hostzertifikate zu installieren, um die Kommunikation zwischen den Knoten sicherer zu machen.

## 5.6 Scheduling mit Haizea

Wie in Kapitel 5.3 beschrieben, arbeitet OpenNebula mit einem eigenen Scheduler. Für einfache Anforderungen ist der Scheduler ausreichend, aber für die Minimalziele der Projektgruppe nicht geeignet. Wie schon oben beschrieben, gibt es die Möglichkeit den OpenNebula-Scheduler durch das Open-Source-Tool Haizea<sup>52</sup> zu ersetzen. Bei Haizea handelt es sich um einen sogenannten „VM-basierten Lease-Manager“. Dieses Programm kann sowohl stand-alone zur Simulation genutzt werden, als auch als Scheduling-Backend für virtuelle Infrastruktur-Manager wie z.B. OpenNebula (vgl. Abb. 6). Interessant ist die Möglichkeit mit Haizea Simulationen über das Scheduling von virtuellen Maschinen durchzuführen. Im Zentrum stehen hier Leases, eine Ressourcenabstraktion für virtuelle Maschinen, welche drei Dimensionen umfassen (Hardware, Software und Verfügbarkeit). Ein Lease lautet umgangssprachlich zum Beispiel: „Ich brauche 4 Nodes, jeder mit 2 CPUs, 6GB Arbeitsspeicher, auf denen Ubuntu mit Apache läuft, von 12:00h bis 14:00h“. Für das Scheduling bietet Haizea verschiedene Modi an. So kann das Scheduling auf einem laufenden OpenNebula durchgeführt werden (nach

<sup>51</sup><http://opennebula.org/software:rnotes:rn-rel2.0>

<sup>52</sup><http://haizea.cs.uchicago.edu>

durchgeführten Schedulingprozess werden sogenannte ENACT-Befehle an OpenNebula geschickt um z.B. eine virtuelle Maschine zum Zeitpunkt X zu starten) oder aber unabhängig von einer bestimmten Middleware in Echt- oder auch Simulationszeit. Durch die verschiedenen Lease-Typen lassen sich vielfältige Szenarien darstellen. Anfragen können als sofort durchführbar (Immediate Lease), pausierbar (Preemptible Best-Effort Lease), nicht unterbrechbar (Non-Preemptible Best-Effort Lease) bzw. zu bestimmten Uhrzeiten aktiv werdend (Advance Reservation Lease) modelliert werden. In zukünftigen Versionen soll es auch auf speziellen Verhandlungen basierende Leases geben, sowie von Deadlines beschränkte Leases [6]. Die Handhabung ist sehr einfach gehalten. Die Konfiguration der vorhandenen Ressourcen, Start-, Endzeiten, Schedulingmechanismen und weiteren Strategien (Policies) wird einfach in einer Datei übergeben. Genauso wird mit den Leases verfahren. Sie speichert man in einer XML-Datei.

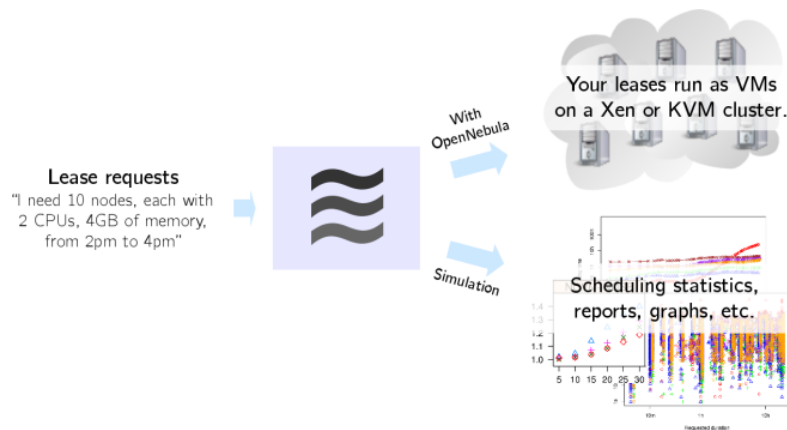


Abbildung 6: Stand-Alone Simulationen oder Scheduling mit Haizea

In der Praxis stellt sich der Simulationsmodus als unbrauchbar heraus, die einzigen Simulationsergebnisse die für die PG relevant sind, ist die CPU-Auslastung der Hostknoten nach einem Simulationslauf. Das reicht nicht, deshalb werden dafür andere Tools (siehe Kapitel 7.1 Simulation, S. 45) genutzt. Aber Haizea bietet wie bereits beschrieben auch einen OpenNebula-Modus an, der für OpenNebula das Scheduling übernimmt. Aber leider stellt sich HAIZEA als nicht so erfolgversprechend heraus wie zuerst gedacht und beworben wurde.

### 5.6.1 Praxis

Haizea verspricht als Scheduler für die Anforderungen der Projektgruppe bestens geeignet zu sein, da zumindest ein Greedy- bzw. Backfilling-Algorithmus für das Scheduling implementiert waren (es gibt die Möglichkeit zwischen den beiden wechseln, standardmäßig ist Greedy aktiv), virtuelle Maschinen in der Zukunft gestartet werden können und Haizea kompatibel mit OpenNebula ist. Die Praxis sieht leider anders aus. Um Haizea mit OpenNebula zu nutzen, muss Haizea gestartet und der Scheduler von OpenNebula manuell beendet werden. Auf ls03 erfolgt dies per Startskript beim Booten des Rechners.

Der offizielle Entwicklungsstand von Haizea ist bereits seinem einem Jahr auf der Version 1.0 stehen geblieben. Haizea 1.0 wird mit OpenNebula 2.0 Beta getestet. Dabei werden allerdings die CPU-Werte der Hosts nicht korrekt angezeigt. Die Arbeitsknoten haben 8 CPU-Kerne, bei OpenNebula wird dies mit der Nummer 800 gekennzeichnet. Haizea zeigt aber nur den Wert 100. Als ein Schedulingtests damit durchgeführt werden, arbeitet das Scheduling nicht korrekt. Es wird jeweils nur eine virtuelle Maschine je Host hinzugefügt und gestartet. Da Haizea jeder VM die volle CPU (als Wert 100) gibt, startet nur eine VM pro Host, jede weitere VM die erstellt wurde, blieb im Pending-Modus. Die Logs von Haizea zeigt die Meldung, dass angeblich keine weitere Ressourcen frei wären. Hier liegt die Vermutung nahe, dass auch mit den falschen CPU-Werten gerechnet wurde. Debugging ergibt, dass die Werte richtig ausgelesen werden. Folglich erscheint die Berechnung in Haizea falsch zu sein.

Es gibt noch die Möglichkeit eine inoffizielle Haizea-Version aus dem SVN-Verzeichnis der Entwicklerseite (markiert als 1.1-Version) auszuprobieren. Hier muss nur das Python-Installationsskript angepasst werden. Die Konfigurationsdateien können aus 1.0 übernommen werden. Das sind alles Schritte, die undokumentiert sind und durch Ausprobieren heraus gefunden werden. Die Version 1.1 kann nun ohne Fehlermeldung installiert werden und die Host-CPU-Werte werden ebenfalls richtig ausgelesen. Auch das Scheduling funktioniert wieder einwandfrei.

Ein weiteres Problem ist aber in einem Dauertest aufgefallen. Es werden virtuelle Maschinen willkürlich gestartet. Werden mehr als drei virtuelle Maschinen in kurzer Zeit erstellt, quittiert Haizea komplett den Dienst. Da ein Neustart von Haizea nicht möglich auch nicht mehr möglich ist, ist jedes Mal eine Neuinstallation notwendig.

Haizea 1.1 ist einfach noch nicht stabil genug. Die offizielle Version 1.0 von Haizea sollte mit OpenNebula 1.4 kompatibel sein. Bisherige Tests wurden lediglich mit OpenNebula 2.0 Beta sowie OpenNebula 2.01 durchgeführt. Doch hier zeigt sich dasselbe Problem. Eine naive Idee den entsprechenden Quelltext von 1.1 für das korrekte Auslesen der CPU-Werte einfach in Haizea 1.0 zu übernehmen, schlägt fehl, weil ein anderes Datenmodell zugrunde liegt.

Der nächste Schritt besteht darin verschiedene Konfigurationseinstellungen in Haizea 1.1 durchzuprobieren, um evtl. eine funktionsfähige Einstellung zu finden. Zum Beispiel wird zwischen dem Greedy und dem Backfilling-Algorithmus hin und her geschaltet. Außerdem wird eine Option aktiviert, die die Übermittlungszeit der VM-Images auf die Hostknoten berücksichtigt. Durch diese Einstellungen und weitere Tests bringt man Haizea dazu, dass ein korrektes Scheduling durchgeführt wird. Leider wurde die Kommunikationsschnittstelle mit OpenNebula nur notdürftig in der neuen Version 1.1 angepasst, somit bleiben fast alle virtuellen Maschinen einfach im Pending-Modus stehen, weil die Kommunikation zwischen Open Nebula und Haizea nicht fehlerfrei funktioniert. Hier besteht noch Handlungsbedarf im Quellcode von Haizea.

Eine Anfrage an die Entwickler von Haizea blieb bis dato erfolglos. Trotzdem entscheidet sich die Projektgruppe dazu, weiterhin auf Haizea zu setzen und Cloudi für den Einsatz mit Haizea vorzubereiten. Vorübergehend wird OpenNebula das Scheduling übernehmen. Aber auch eine einfache Lösung ist in Cloudi eingebaut, so gibt es einen Timer-Service der eine Anfrage zum Starten einer VM entgegen nimmt, bis zum Startzeitpunkt wartet und erst dann den Befehl an ONE schickt. Dieser Service kann aber in der Praxis nicht getestet werden. Eine weitere Lösung wäre natürlich Haizea zu fixen oder einen eigenen Scheduler für ONE zu schreiben.

## 5.7 Cloudi aus der Server-Ansicht

Rückblickend auf die vergangenen Abschnitte ergibt sich folgender Aufbau, siehe Abb. 7. Die Benutzerschnittstelle von OpenNebula befindet sich genauso wie Haizea auf ls03. OpenNebula kommuniziert mit Hilfe von SSH mit den Clusterknoten. Dort werden die entsprechenden `virsh`-Befehle ausgeführt, um die virtuellen Maschinen zu erstellen und zu steuern. OpenNebula wird komplett in einem Ordner platziert, auf dem jeder Host per NFS Zugriff hat. Dieses Verfahren wird auch bei der Installationsanleitung so vorgeschrieben, damit das Live-Migrieren der VMs möglich ist. Um das Kopieren der Images der virtuellen Maschinen zu beschleunigen, werden sie mit in das

NFS eingefügt. Cloudi wird auf einem Apache Tomcat deployed. Natürlich muss Cloudi nicht explizit auf dem gleichen Server liegen wie das Virtualisierungsmanagement. Der ls03 dient auch als NFS-Server. Hier wäre es aus Perfomanzgründen vorteilhafter einen dedizierten Server zu verwenden, der evtl. zur Sicherheit nochmal das komplette NFS als Backup spiegelt.

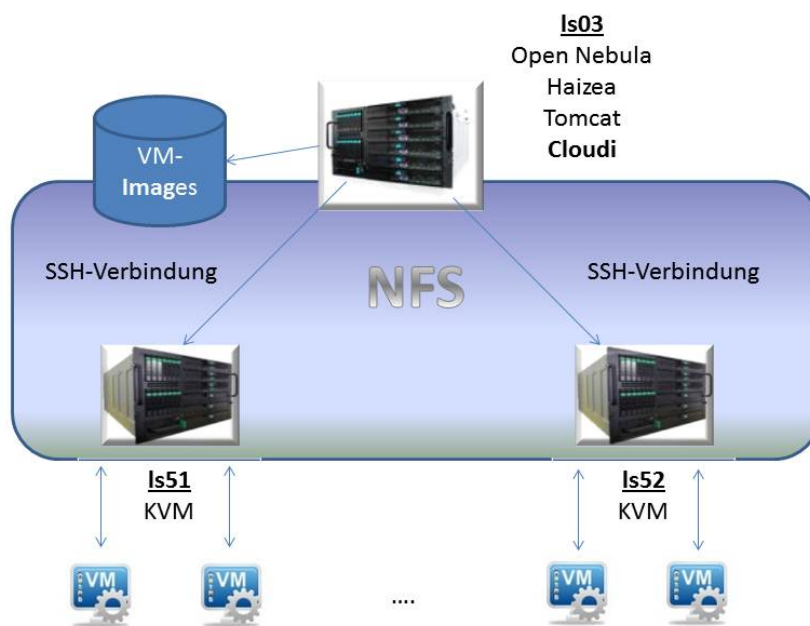


Abbildung 7: Cloudi aus der Server-Ansicht

## 6 Monitoring

Nach einer Recherche, welche Tools für das Projekt in Frage kommen könnten, stehen für die Projektgruppe drei Tools zur Auswahl: Ganglia Monitoring System, libvirt oder eine eigene Implementierung in Java.

Da OpenNebula libvirt benutzt, um die virtuellen Maschinen zu verwalten, bietet es sich an, das Monitoring mit libvirt umzusetzen. libvirt enthält eine Ansicht, die eine Zusammenfassung der laufenden Domains, ihre Live-Performance und eine Statistik über die Ressourcenauslastung liefert. Der Versuch die libvirt Tools wie virsh und virt-manager mit OpenNebula zu verbinden, um die Monitoringdaten zu bekommen, ist vermutlich auf Grund der verwendeten OpenNebula-Version (2.0 Beta) jedoch nicht erfolgreich verlaufen. Deswegen kommt dieser Ansatz für das Monitoring nicht in Frage. Die Entwickler wurden informiert, da dieser Weg aber zeitaufwändiger ist, probiert es die Projektgruppe mit anderen Methoden. Libvirt ist ein guter Kandidat für das Monitoring und bietet eine gute API. Der Versuch libvirt mit OpenNebula zu verbinden ist fehlgeschlagen und erschwert somit das Monitoring.

Der Weg ein eigenes Programm in Java zu implementieren, ist nur eine provisorische Lösung. Die Idee dahinter ist, dass je nach Betriebssystem ein entsprechendes Java-Programm geschrieben werden muss. Das Java-Programm ist zuständig dafür, dass die Daten, die für das Monitoring gebraucht werden, gesammelt werden. Mit anderen Worten läuft für jede virtuelle Maschine ein zusätzlicher Prozess, der dieses Programm in einem bestimmten Zeitintervall aufruft. Danach muss auch noch ein Prozess erzeugt werden, der die Daten aus jedem Knoten sammelt. Dieser muss auf einem Knoten laufen und die Daten von allen Rechnern, die mit dem Netzwerk verbunden sind über die dafür vorgesehenen Sockets sammeln. Eine andere Möglichkeit ist, die Daten über NFS zu sammeln. Dieser Ansatz ist die komplexere von beiden.

Um das Java-Programm zu implementieren, benutzt die Projektgruppe die Bibliothek „System Information Gatherer And Reporter“ (Sigar)<sup>53</sup>. Sigar ist eine kostenlose Software Bibliothek, die unter der Apache Lizenz 2.0 lizenziert ist. Es bietet eine cross-plattform, cross-language Programmierschnittstelle an, welche die niedrigste Ebene der Informationen für Computer Hardware und Betriebssysteme ist. Von über 25 verschie-

---

<sup>53</sup><http://www.hyperic.com/products/sigar>

denen Betriebssystemen werden Informationen mit Hilfe von Sigar gesammelt.

Die Java Schnittstelle für die Bibliothek Sigar setzt voraus, dass auf jeder virtuellen Maschine die Java Virtual Machine (JVM)<sup>54</sup> installiert sein muss, damit das Java Programm in der jeweiligen, virtuellen Maschine lauffähig ist. Je nach Betriebssystem sammelt das Java Programm die aktuellen Informationen von der CPU, dem Arbeitsspeicher und dem Netzwerk.

Ergänzend muss noch ein Prozess erstellt werden, der mit jedem Rechner im Netzwerk kommuniziert. Der Prozess muss wissen, welche virtuellen Maschinen zum Netzwerk gehören. Somit kann er in bestimmten Zeitintervallen die Informationen von der CPU, dem Arbeitsspeicher und vom Netzwerk aus jeder virtuellen Maschine sammeln. Die gesammelten Daten werden in eine Datenbank geschrieben, damit diese in der Zukunft zur Analyse bereitstehen.

Wie bereits erwähnt stellt dieser Ansatz nur eine provisorische Lösung dar. Die Idee hat sich gut bewährt. Ein Vorteil liegt vor allem darin, dass die Lösung sehr flexibel ist. Es ist sogar ohne großen Aufwand möglich, Informationen die über das reine Monitoring hinaus gehen, zu gewinnen. Der Nachteil dieses Ansatzes liegt jedoch bei ihrer Robustheit. Durch den vergleichsweise begrenzten Zeitrahmen ist es leider nicht möglich ein Programm zu schreiben, zu testen und zu evaluieren, so dass ein robustes Monitoring System, wie es das Ganglia Monitoring System ist, entsteht. Aus diesem Grund setzt die Projektgruppe Ganglia ein, obwohl das eigens geschriebene Java- Programm schon teilweise implementiert ist.

Ganglia ist ein verteiltes, skalierbares Monitoringsystem für Cluster und Grids, welches auf über 500 Clustern weltweit eingesetzt wird und stellt wichtige Performancemetriken und Systemstatistiken bereit. Für den Betrieb werden allgemeingebäuchliche Technologien, wie RRDtool, XDR<sup>55</sup>, XML<sup>56</sup> und PHP-unterstützte Webinterfaces eingesetzt. Ganglia ist hierarchisch strukturiert, um Messdaten von mehreren Clustern zu sammeln und gemeinsam darzustellen. Entwickelt wurde es von einer Arbeitsgruppe der Berkeley University um Matthew L. Massie in Cooperation mit dem San Diego Supercomputer Center und anderen. Zurzeit ist Ganglia auf diversen Betriebssystemen und CPU-Architekturen lauffähig. Erweiterungen für eigene Messdaten können durch eine bereitgestellte Bibliothek hinzugefügt

---

<sup>54</sup><http://java.sun.com/docs/books/jvms/>

<sup>55</sup><http://tools.ietf.org/html/rfc4506>

<sup>56</sup><http://www.w3.org/XML>

werden. Derzeit ist das Ganglia Monitoring Toolkit in der Version 3.1.7 verfügbar.

Da die ersten beiden Ideen nicht von Erfolg gekrönt waren und Ganglia als Open Source Projekt durchaus für Clouidi in Frage kommt, wurde Ganglia installiert und konfiguriert.

## 6.1 Architektur Ganglia

Das Ganglia Monitoring System besteht aus zwei Hauptkomponenten. Diese sind der Ganglia-Monitoring-Daemon (gmond) und der „Messdaten-Sammel-Daemon“ (gmetad), der die Messwerte sammelt. Hinzu kommt ein Webfrontend zur Visualisierung der Ergebnisse.

### 6.1.1 gmond

Der gmond überwacht einen Knoten und tauscht Messdaten mit anderen gmond-Daemons aus. Es werden auf Anfrage Monitoringdaten an einen gmetad-Daemon veröffentlicht. Dieser Ablauf ist als Sammlung von Threads organisiert und jeder Thread hat eine spezielle Aufgabe. Die collect- and publish-Threads sammeln und veröffentlichen Informationen über den überwachten Knoten und senden sogenannte Heartbeat-Nachrichten an andere Knoten des Clusters. Der listening-Thread hört den Multicastkanal auf eingehende Daten von anderen Clusterknoten ab und aktualisiert den Speicher des gmond-Daemons (hierarchische Hashtabelle, in der die Clustermonitoringdaten gespeichert werden). Der export-Thread beantwortet Anfragen von einem gmetad-Daemon. Die Daten werden nicht persistent gespeichert, sondern in einem 2 MB grossen Stackabschnitt im virtuellen Speicher gehalten. Um hohe Nebenläufigkeit zu erreichen benutzt die Hashtabelle ein feingranulares read-/write-locking. Dies ist entscheidend, da sowohl der listening- als auch der export-Thread auf die Hashtabelle zugreifen und notwendigerweise fast gleichzeitig Daten schreiben oder lesen können bzw. müssen und Daten von vielen Quellen gleichzeitig gespeichert werden. Daten zwischen gmond-Daemons werden im XDR-Format ausgetauscht. Die Threadkomponenten eines gmond-Daemon wird in Abb. 8 dargestellt. In der Abbildung ist auf der linken Seite unten die Beschriftung Metric Data zu sehen. Hier werden die Metriken, die an das gmetad des Clusters übergeben werden, festgelegt. Der Weg führt von hier zu den Threads deren Aufgabe weiter oben erläutert wurde. Die gesammelten Daten die über den Multicast-Kanal zum Listening Thread gelangen werden anschließend gespeichert und ins XML-Format konvertiert. Erst jetzt werden die Daten an gmetad des Clusters überge-

ben. gmond veröffentlicht zwei Arten von Monitoringdaten, nämlich Built-In Metriken die abhängig vom Betriebssystem und der CPU-Architektur gesammelt werden, und andere Monitoringarten, welche benutzerdefiniert sind und auch anwendungsspezifische Metriken enthalten.

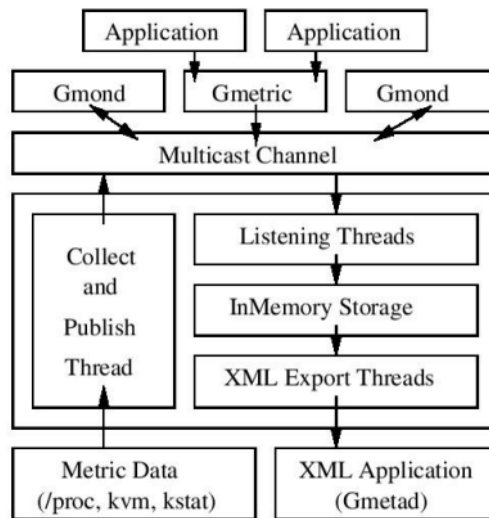


Abbildung 8: Ganglia Implementation [7]

### 6.1.2 gmetad

Der gmetad-Daemon läuft üblicherweise auf einem Clusterfrontend. Entweder ist dies der Webserver selbst oder ein Rechner, der mit dem Webserver kommuniziert. gmetad fragt die gmonds des Clusters ab und unterstützt die Aggregation der Daten über TCP-Verbindungen. Die Verknüpfung von Monitoringdaten wird in Ganglia über einen Baum von Punkt-zu-Punkt Verbindungen erreicht, um den Zustand mehrerer Cluster zu aggregieren. Dies geschieht durch Abfragen der Clusterknoten mit installiertem gmetad. Die gmetad, die auf den Baumknoten laufen, fragen ihre Blattdatenquellen ab, parsen deren XML-Daten, speichern numerische Metriken in Round-Robin Datenbanken und exportieren gesammelte XML-Daten über TCP-Sockets zu Verbrauchern. Verbraucher sind wiederum die höher liegenden gmetad oder ein Webfrontend, wie in Abb. 9 zu erkennen ist. Datenquellen können gmond-Dämonen, die spezifische Cluster repräsentieren, oder andere gmetad-Dämonen, die Sets von Clustern repräsentieren, sein. Durch die Multicastkommunikation der gmond-Daemon untereinander erhält jeder gmond-Daemon den gesamten Clusterzustand. Auch wenn ein Knoten wegfällt, kann

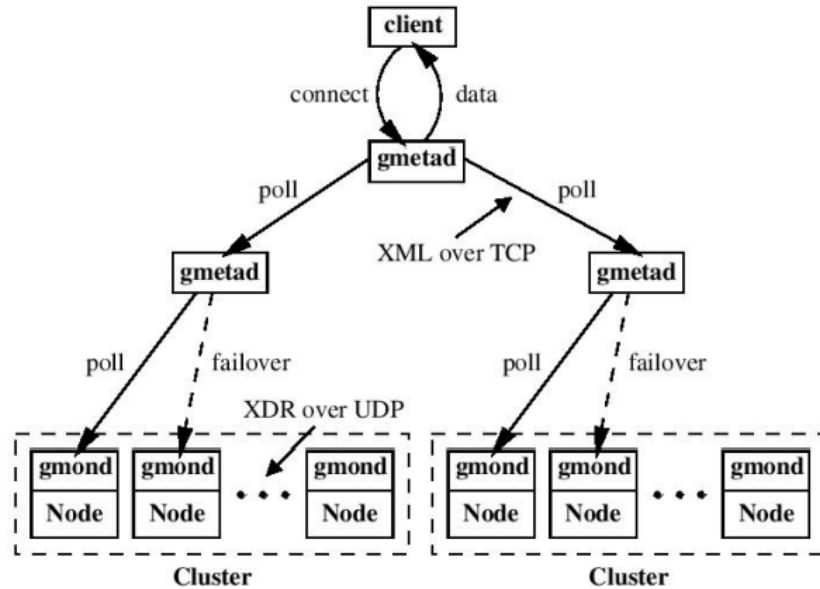


Abbildung 9: Die Architektur von Ganglia [7]

die Funktionalität der restlichen Knoten des Clusters überwacht werden (Failover). Die Architektur von Ganglia ist in Abb. 9 dargestellt.

### 6.1.3 Kommunikation

Die Metriken werden aus Portabilitäts- und Effektivitätsgründen im XDR-Format veröffentlicht. Dies erfolgt über den Multicastkanal. Es existieren sogenannte „metric lookup table“, die sich im collect- und publish-Thread in den gmond-Daemons wiederfinden. Diese enthalten statische Repräsentationsdefinitionen. Somit ist gewährleistet, dass ein eindeutiger Schlüssel `xdr_u_int` und der dazugehörige Wert pro Veröffentlichung gesendet wird. Jede Metrik, die gesendet wird hat einen bestimmten Typ. Tabelle 1 stellt ein paar Beispielmetriken aus der metric lookup table eines gmond-Daemons dar. Weiterhin gibt es eine clientseitige Bibliothek, um benutzerdefinierte Eigenschaften zu realisieren und das Kommandozeilenprogramm `gmetric`, um selbst definierte Datensätze zu veröffentlichen. In Abb. 9 ist zu sehen, dass die einzelnen Cluster in der untersten Ebene angesiedelt sind. Auf den Knoten der Cluster befinden sich die gmond-Daemons. Diese werden durch übergeordnete Aggregationspunkte (gmetad) verknüpft. gmetad-Daemons fragen in regelmäßigen Abständen Daten von den mit ihnen verbundenen gmond-Daemons ab, um sie weiter nach oben zu veröffentlichen.

Schlüssel (xdr_u_int)	Metrik	Wertformat
0	user-defined	explicit
1	cpu_num	xdr_u_short
2	cpu_speed	xdr_u_int
3	mem_total	xdr_u_int
4	swap_total	xdr_u_int
...	...	...
15	load_one	xdr_float
16	load_five	xdr_float
17	load_fifteen	xdr_float
...	...	...

Tabelle 1: Beispielmetriken aus der gmond metric lookup table[7]

Die Daten innerhalb von Clustern werden anschließend über einen Multicastkanal als UDP-Pakete im XDR-Format gesendet. Außerhalb der Cluster erfolgt die Kommunikation zwischen den gmetad-Daemons hingegen über TCP-Verbindungen in XML-Datenrepräsentationen.

## 6.2 RRDTool

RRDtool ist ein Programm, mit der zeitbezogene Messdaten gespeichert, zusammengefasst und visualisiert werden. Das Programm wurde ursprünglich von Tobias Oetiker entwickelt und unter der GNU General Public License (GPL) lizenziert. Durch die Veröffentlichung als freie Software, haben inzwischen viele weitere Autoren neue Funktionalitäten beigesteuert und Fehler behoben. RRDtool ist als Quelltext und als ausführbares Programm für viele Betriebssysteme verfügbar.

Die Abkürzung RRD steht für „Round-Robin-Database“ und bezieht sich auf die Art und Weise, in der die Daten von RRDtool gespeichert werden. Beim Anlegen einer Datenbank, einer sogenannten „RRD-Datei“, wird genug Speicher für eine angegebene Zeitspanne angelegt. Diese beträgt z.B. für die Vorhersagedaten mindestens zwei Jahre, da die Projektgruppe eine aussagekräftige Vorhersage erreichen möchte. Nach dieser Zeitspanne wird die Datenbank nicht erweitert (die Datei wird also nicht vergrößert), sondern die ältesten Daten werden überschrieben. Dieses „Reihum-Verfahren“ wird in der Informatik als Round-Robin bezeichnet.

Die Benutzer-Schnittstelle von RRDtool ist ein Kommandozeilen-Pro-

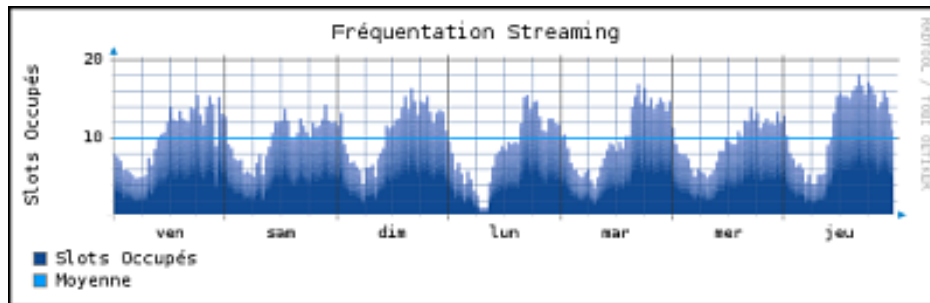


Abbildung 10: Demografik aus RRDtool

grammen. Damit es Möglich ist RRDtool von anderen Programmen zum Speichern von Daten und Erstellen von Graphen zu verwenden, stehen Programmierschnittstellen (APIs) für mehrere Programmiersprachen bereit, unter anderem für C und Perl. Üblicherweise wird RRDtool nicht direkt über die Kommandozeile verwendet, sondern dient anderen Programmen als Datenspeicher bzw. -quelle. Ein Beispiel ist das Ganglia Monitoring System, bei dem als Datenbanksystem RRD-Datenbanken eingesetzt werden.

Je nach Zeitspanne der in einem Graphen dargestellt wird, werden Daten in verschiedenen Auflösungen benötigt. Häufig ist eine kurze Zeitspanne von Interesse. Es geht meistens um einen Tag, eine Woche, ein Monat oder um ein Jahr. Von RRDtool werden die Daten entsprechend zusammengefasst. Für jede gewünschte Zeitspanne wird ein „Round-Robin-Archiv“ (RRA) angelegt, das eine feste Anzahl an Datenpunkten aufnehmen kann. Dieser Platz wird dann im Round-Robin-Verfahren gefüllt. Jedes RRA gibt dabei eine Auflösung vor, die (zusammen mit der Anzahl der Datenpunkte) die Zeitspanne festlegt, für die Daten gespeichert werden. Die „Konsolidierungsfunktion“ (consolidation function, CF) legt fest, wie mehrere Datenpunkte zusammengefasst werden. Die am meisten verwendeten Funktionen sind der Mittelwert, das Minimum und das Maximum. Eine Demografik, die mit Hilfe von RRDTool erstellt wurde, ist in Abb. 10 zu sehen.

### 6.3 PHP Webfrontend

Gesammelte Monitoringdaten werden geparsed und in Round-Robin-Datenbanken gespeichert, um dann mit RRDtool visualisiert zu werden. Per RRDtool werden Graphen generiert und über ein PHP-Webfrontend dargestellt<sup>57</sup>. Zur individuellen Anpassung wird hier Templatepower

<sup>57</sup><http://ganglia.wikimedia.org>

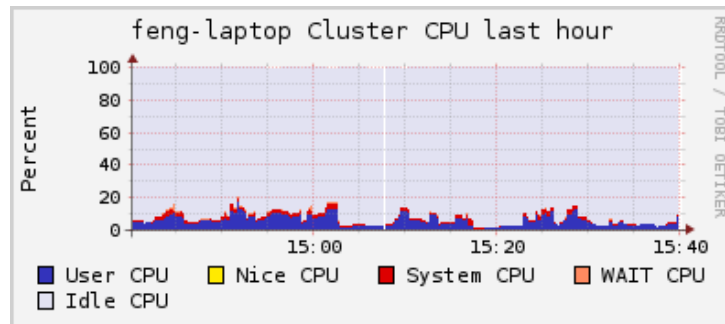


Abbildung 11: Graph für CPU-Auslastung der letzten Stunde

([templatepower.codocad.com](http://templatepower.codocad.com)) genutzt. Das Ganglia-Webfrontend gibt eine Übersicht über die gesammelten Informationen auf einer Webseite. Die Ganglia-Daten werden in aussagekräftiger Form dargestellt. Es kann z.B. die CPU-Benutzung der letzten Stunde, Tag, Woche, Monat oder Jahr angezeigt werden. In Abb. 11 wird z.B. die CPU-Benutzung der letzten Stunde dargestellt. In der Abbildung ist zu erkennen, dass unter Ganglia die CPU in fünf Bereiche aufgeteilt ist. Dies ist in Ganglia so festgelegt. In Abb. 12 wird die Unterteilung der Netzwerkstatistik in In&Out-Bytes dargestellt. Bei der Speicher- und Festplattennutzung erfolgt eine Unterteilung in Memory Used, Memory Buffered, Total In-Core Memory, Memory Shared, Memory Cached und Memory Swapped, wie in Abb. 13 zu erkennen ist.

Damit das Ganglia-Webfrontend funktioniert, muss der gmetad-Daemon ein entsprechendes Socket vorweisen. Das Ganglia-Webfrontend öffnet den Port 8151 auf dem der Ganglia-XML-Baum erreichbar ist. Dieser wird anschließend mit Hilfe eines SAX-XML-Parsers verarbeitet und im Web-Frontend graphisch aufbereitet dargestellt. Auf Änderungen der im gmond-Daemon enthaltenen Daten wird somit dynamisch reagiert. Dazu muss allerdings zu jedem Zeitpunkt eines Zugriffs auf die Seite, der XML-Baum geparsed und aufbereitet werden. Deswegen empfehlen die Entwickler von Ganglia einen entsprechend leistungsstarken Rechner einzusetzen, der die Anfragen an das Webfrontend verarbeiten kann, welches eine eher Waage Empfehlung ist, da keine konkreten Angaben gemacht werden.

Das Webfrontend ist in PHP geschrieben und nutzt die durch den gmetad-Daemon mit Hilfe des RRD-Tools generierten Graphen, um aktuelle Metriken aber auch Verlaufsinformationen ausgewählter Metriken zu veröffentlichen. Zur Trennung des PHP-Codes vom HTML-Layout wurde das Tool Template-Power genutzt.

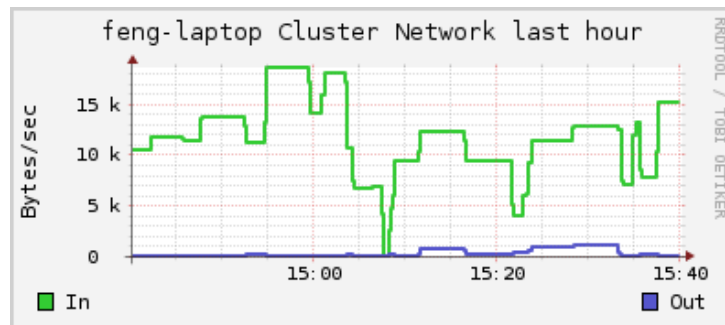


Abbildung 12: Graph für die Netzwerkstatistiken

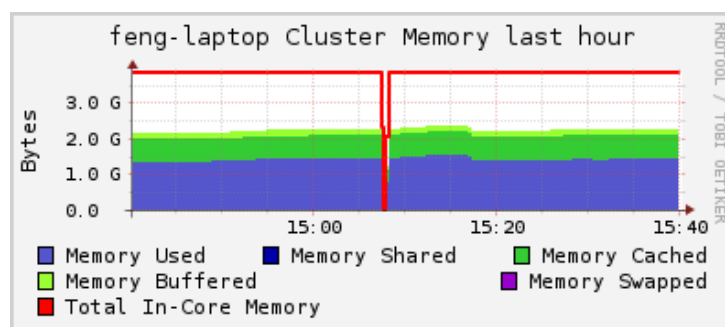


Abbildung 13: Graph für die Speicherbenutzung

Arten von Metriken	RRD in Ganglia
CPU	cpu_user.rrd cpu_nice.rrd cpu_system.rrd cpu_idle.rrd cpu_wio.rrd
Netzwerk	bytes_in.rrd bytes_out.rrd
Arbeitsspeicher	mem_buffers.rrd mem_cached.rrd mem_free.rrd mem_shared.rrd mem_total.rrd

Tabelle 2: RRD in Ganglia

## 6.4 Das Monitoring mit Einsatz von Ganglia

Im Abschnitt 6.2 wurde deutlich, dass man auf die RRD (Round Robin Database) mit Hilfe von RRDTool zugreifen kann. Den Namen der Verzeichnisse auf dem Master-Knoten identifiziert Ganglia durch den DNS-Namen des Rechners und nicht durch die IP-Adresse. OpenNebula nutzt hierzu jedoch die IP-Adresse. Darum mussten die IP-Adressen automatisch auf die Hostnamen gemappt werden. Anschließend konnten im Weiteren die Monitoring Daten, die als RRD in bestimmten Verzeichnissen gespeichert waren, herausgelesen werden. Entsprechend wurden im `package cloudi.ganglia` (siehe Entwicklungshandbuch) Java-Klassen geschrieben. Es wurde eine Java Schnittstelle zur Shell benutzt. Dort wurden die ausgegebenen Werte gefiltert. Es wird z.B. bei der Ausgabe CPU-Auslastung: 50% nur die Zahl 50 übernommen. Die gefilterten Daten werden letztlich in der zentralen MySQL-Datenbank gespeichert. Eine Tabelle für die Daten des Arbeitsspeicher, eine für die Daten von der CPU und eine weitere für die Daten vom Netzwerk.

Zum Auslesen wurden drei Parameter verwendet: die ID der virtuellen Maschine (`integer`) sowie das jeweilige Start- bzw. Enddatum (`java.util.Date`).

Als Ausgabe werden dann die Monitoring Daten für den angegebenen Zeitraum und für die virtuelle Maschine mit der entsprechenden ID zurückgegeben.

Für die Daten der CPU erstellt Ganglia mehrere RRD: `cpu_user.rrd`, `cpu_nice.rrd`, `cpu_system.rrd`, `cpu_idle.rrd` und `cpu_wio.rrd`. Diese Dateien befinden sich für jede virtuelle Maschine in einem bestimmten Verzeichnis. Für die Daten des Netzwerks erstellt Ganglia die folgenden RRDs: `bytes_in.rrd` und `bytes_out.rrd`. Und für den Arbeitsspeicher werden von Ganglia die folgenden RRD-Datenbanken erstellt: `mem_buffers.rrd`, `mem_cached.rrd`, `mem_free.rrd`, `mem_shared.rrd` und `mem_total.rrd`. Tabelle 2 bietet hierzu nochmal eine Übersicht.

Die MySQL-Tabellen werden mit der gleichen Struktur wie in den RRD erzeugt. Der einzige Unterschied ist, dass nun auch die IDs der virtuellen Maschinen in der Tabelle gespeichert werden. Wenn die Monitoringdaten gleich null sind, bedeutet dies, dass die entsprechende Maschine ausgeschaltet ist.

Zur Darstellung wurden zwei Varianten umgesetzt. Eine benutzt das RRDTool, um die entsprechende Grafik zu erstellen. In der anderen Variante werden die gebrauchten Daten aus der RRD-Datenbank oder aus der MySQL-Datenbank herausgelesen. Im Anschluss daran werden die Daten im Webfrontend direkt visualisiert. Dank des ZK-Frameworks, können die Daten direkt zur Grafik-Engine des ZK-Frameworks weitergeleitet werden, und dort visualisiert. Die Abb. 14, Abb. 15 und Abb. 16 stellen ein paar Beispiele aus dem ZK-Framework dar.

Die Lösung mit dem Ganglia Monitoring System wurde erfolgreich implementiert und getestet.

Der große Vorteil liegt hier darin, dass alle Daten, die über die RRD-Datenbank hinausgehen, auch mit gespeichert werden können. Denn das Prinzip der Round Robin Datenbanken ist, dass die älteren Daten irgendwann gelöscht werden. Da das Ziel der Projektgruppe die Speicherung aller Daten ist, ist somit gewährleistet, dass ältere, nicht mehr relevante Daten durch das Round-Robin-Verfahren verloren gehen und Platz für aktuellere Daten geschaffen wird. Ein zusätzlicher Vorteil ist, dass das Ganglia Monitoring System ein stabiles, gut entwickeltes, durchdachtes und mächtiges Monitoring System ist. Der Datenaustausch ist effizient, da Ganglia Monitoring System hierfür eigens ein Kommunikationsprotokoll entwickelt hat.

Der Nachteil dieses Ansatzes liegt darin, dass es sehr umfangreich und zeitaufwändig ist. Die Daten sind erneut zu lesen und zu speichern. Dies

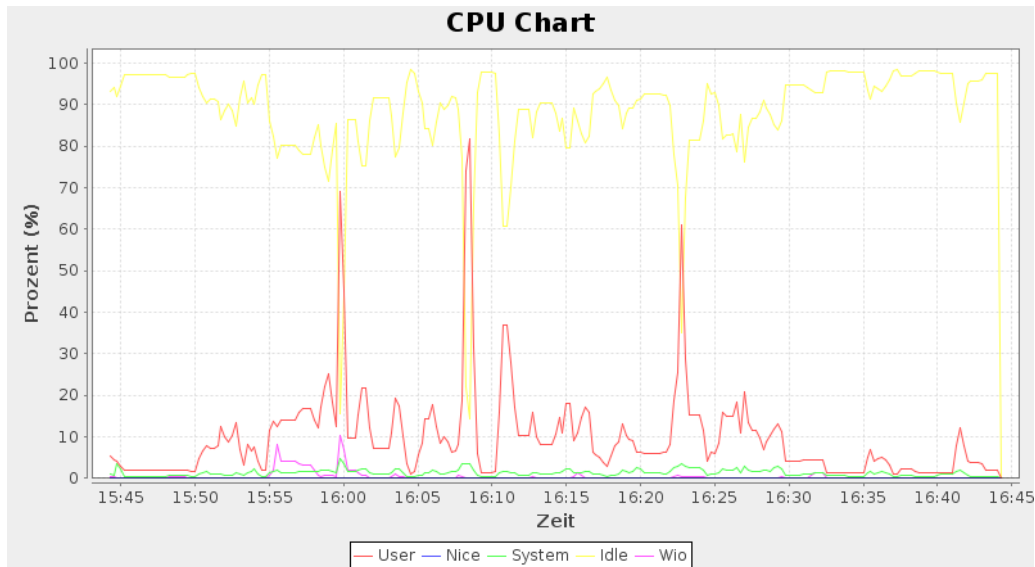


Abbildung 14: Graph für die CPU Auslastung aus ZK

benötigt auch mehr Speicherplatz. Das Herauslesen der Daten aus der MySQL-Datenbank dauert viel länger als aus der RRD. Der Grund ist, dass für die RRD eine Zeitspanne vorgegeben wird und sie somit nicht mehr wächst. Diese Zeitspanne stellt für die RRD die Grenze dar, ab der alte Daten hinzugefügt und ältere gelöscht werden. Die MySQL-Datenbank jedoch wächst mit hinzukommen von neuen Daten immer weiter. Mit der Zeit nehmen die Tabellen dementsprechend auch an Größe zu und es wird nötig die MySQL-Datenbank entsprechend zu pflegen.

## 6.5 Zusammenfassung und Ausblick

Den Schwerpunkt der Projektgruppe bildete das Monitoring. Dieser Teil ist der schwierigste gewesen und die Entwicklung war sehr zeitintensiv. Im ersten Semester wurden die Grundlagen für die Verwaltung der virtuellen Maschinen mit Hilfe von OpenNebula gelegt. Danach wurde festgestellt, dass OpenNebula die Schnittstelle für das Monitoring der jeweiligen virtuellen Maschine nicht anbietet. Deshalb mussten weitere Werkzeuge wie Ganglia benutzt werden, um das Ziel zu erreichen. Es bleibt zu hoffen, dass in Zukunft mit neueren Versionen von OpenNebula das Monitoring aus libvirt heraus direkt unterstützt wird. Dann können alle Daten durch XML-RPC direkt aufgerufen werden, was eine schnellere und in Hinsicht auf die Entwicklung effektivere Lösung ist.

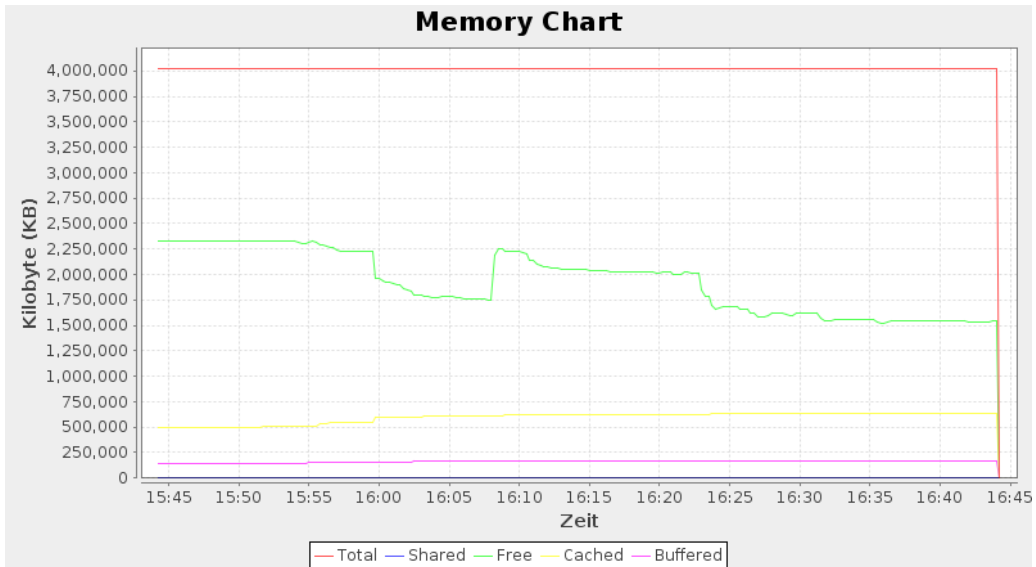


Abbildung 15: Graph für die Arbeitsspeicher Benutzung aus ZK

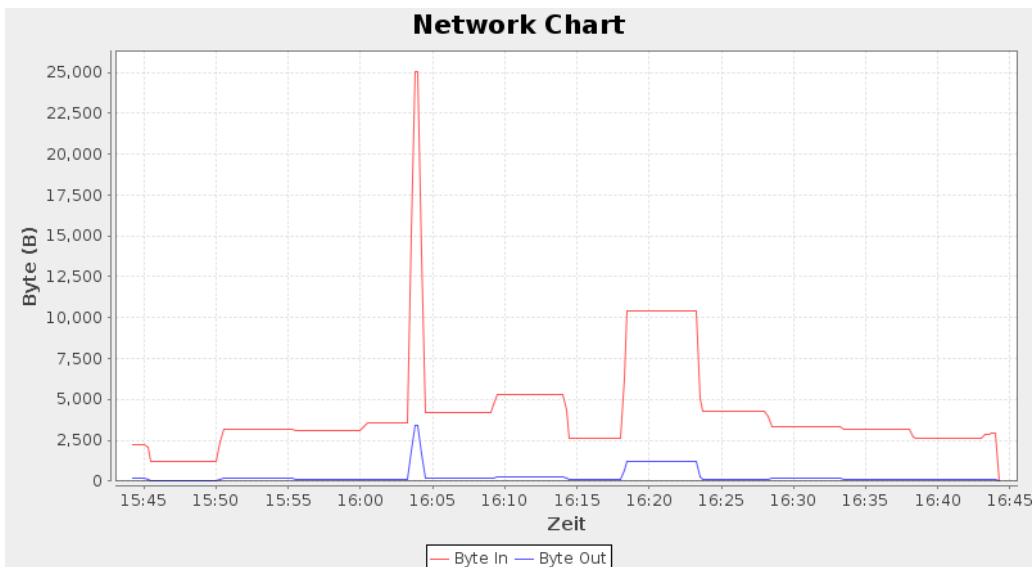


Abbildung 16: Graph für die Netzwerkstatistiken aus ZK

## 7 Ressourcenabschätzung

### 7.1 Simulation

Neben dem Management der virtuellen Maschinen sowie der allgemeinen Ressourcenverwaltung, steht in dieser Projektgruppe auch noch ein dritter Aufgabenbereich im Fokus, der sich mit der Vorhersage des zukünftigen Ressourcenbedarfs auseinandersetzt. Dies ist sowohl für die Beratung bei der Erzeugung von virtuellen Maschinen als auch bei der Planung zur Anpassung der Hardware an hier prognostizierte Entwicklungen der Ressourcenauslastungen sehr wichtig. Der Betreiber muss in die Simulation eingeben, wie viele Server mit wieviel Hardwareressourcen (siehe Oberfläche Simulation im Handbuch) reserviert werden sollen. Anschließend wird berechnet, ob diese Ressourcen die Anforderungen erfüllen. Die Cloud kann dabei als gesamtes System simuliert werden oder der Betreiber beschränkt sich zunächst auf Teilbereiche wie Ressourcenauslastung oder Benutzerzugriffe. Zentrale Punkte sind allerdings immer die virtuellen Maschinen und deren Scheduling.

Durch die Simulation wird herausgefunden mit welchen Ressourcen die Nachfrage der Benutzer abgedeckt werden kann. Die Kosten, die für die jeweiligen Benutzer entstehen, werden ebenfalls mit der Simulation ermittelt. Die Kosten werden dabei für den jeweiligen Verbrauch aufsummiert. Somit wird ein Überblick über die Gewinnspanne geliefert. Unnötige Kosten werden dadurch vermieden. Man kann auch für mehrere Benutzer die Kosten gleichzeitig erstellen. In den folgenden Abschnitten wird die Simulation in Bezug auf Cloudi dargestellt.

#### 7.1.1 CloudSim

CloudSim ist eine Open Source Software für die Simulation von Cloud Umgebungen. CloudSim modelliert und erzeugt eine oder verschiedene virtuelle Maschinen in einem Knoten von einem Datenzentrum. Die Vorteile von CloudSim liegen dabei insbesondere

- im Modellieren und Simulieren von skalierbaren Cloud Computing Datenzentren auf einem einzigen Knoten (Rechner)
- für das Modellieren von Datenzentren, Service, Broker, Scheduling und Allocations Policies
- in der Erzeugung und dem Management von mehreren unabhängigen und co-hosted virtualized Services auf einem Datenzentrum



Die CloudSim-Architektur besteht aus mehreren Schichten. Die unterste Schicht ist die physikalische Schicht(Hosts). Den Servern stehen CPUs, Arbeitsspeicher, Netzwerk und Festplatten zur Verfügung. Diese Information ist die Basis der Simulation. Man kann sie auch als Infrastruktur interpretieren. Mehrere Cluster heißen in CloudSim Datacenter. Diese bilden die zweite Schicht. Die GridSim Schicht bildet eine Basis für die dritte Schicht. Folgende Funktionalitäten sind in dieser Schicht enthalten:

- Ausfälle von Grid-Ressourcen zur Laufzeit
- verfügt über die Infrastruktur zur Reservierung eines Grid-Systems
- enthält eine Netzwerkverkehr Funktionalität auf einer probabilistischen Distribution. Dies ist nützlich für die Simulation über ein öffentliches Netzwerk, wenn das Netz überlastet ist

In der dritten Schicht ist das Cloud Service. Es wird festgelegt, wie virtuelle Maschinen konfiguriert werden, wie z.B. CPU, Arbeitsspeicher und Festplatte für die jeweilige virtuelle Maschine. Des weiteren wird auch beschrieben, wie diese virtuelle Maschinen verwaltet werden (Policies).

Ein Beispiel ist, wenn zwei virtuelle Maschinen VM1 und VM2 jeweils eine CPU haben. Es gibt einen Task, der für eine CPU eine Sekunde braucht. Dieser Task läuft sowohl auf der virtuellen Maschine VM1 als auch auf der VM2. Die beiden virtuellen Maschinen VM1 und VM2 laufen jeweils unter einem Host, der nur eine CPU hat. Wenn space-shared eingesetzt wird, läuft die virtuelle Maschine VM1 mit ihrem Task eine Sekunde lang, während der Task auf der virtuellen Maschine VM2 zwei Sekunden braucht, weil die virtuelle Maschine VM2 zunächst auf VM1 warten muss. Wenn time-shared policy eingesetzt wird, brauchen sowohl VM1 als auch VM2 2 Sekunden, weil jede virtuelle Maschine anteilig sozusagen nur eine halbe CPU bekommen. Der Vorteil dabei jedoch ist, dass die VM unabhängig voneinander, nämlich ohne Wartezeit auf die Freigabe der CPU von der anderen VM arbeiten können. Die Applikationen, die vom Benutzer beantragt werden, werden auch mit diesen Policies verwaltet. Die höchste Schicht ist das Cloudlet, die konkrete Applikationen(Tasks) bestimmt. Ein Browser verbraucht beispielsweise 0.3 CPU, 0.2 Arbeitsspeicher und 0.1 Netzwerk, NetBeans verbraucht 1 CPU, 20 Arbeitsspeicher und 0.3 Netzwerk. Diese Applikationen laufen unter virtuellen Maschinen. Ob diese Applikationen nur unter einer einzigen virtuellen Maschine oder mehreren virtuellen Maschinen laufen, kann nun in der Simulation angegeben werden. Dann gibt es die User oder Broker. Der Broker ruft die Cloudlets auf. Cloudlets rufen wiederum die virtuellen Maschinen auf. Die virtuellen Maschinen verbrauchen die Ressourcen der

Hosts (Datacenter). Der Ablauf von CloudSim wird in Abb. 17 dargestellt. Das Ziel der Simulation ist, herauszufinden, ob die Datacenter die Nachfrage der Broker erfüllen können. Wenn ja, wie viel Zeit gebraucht wird und wieviel und welche Ressourcen (CPU, Arbeitsspeicher, Netzwerk und Festplatten) benutzt werden. Des weiteren werden die Kosten der CPU, Arbeitsspeicher, Netzwerk und Festplattenspeicher bestimmt und für jeden Broker eine eigene Kostentabelle erstellt.

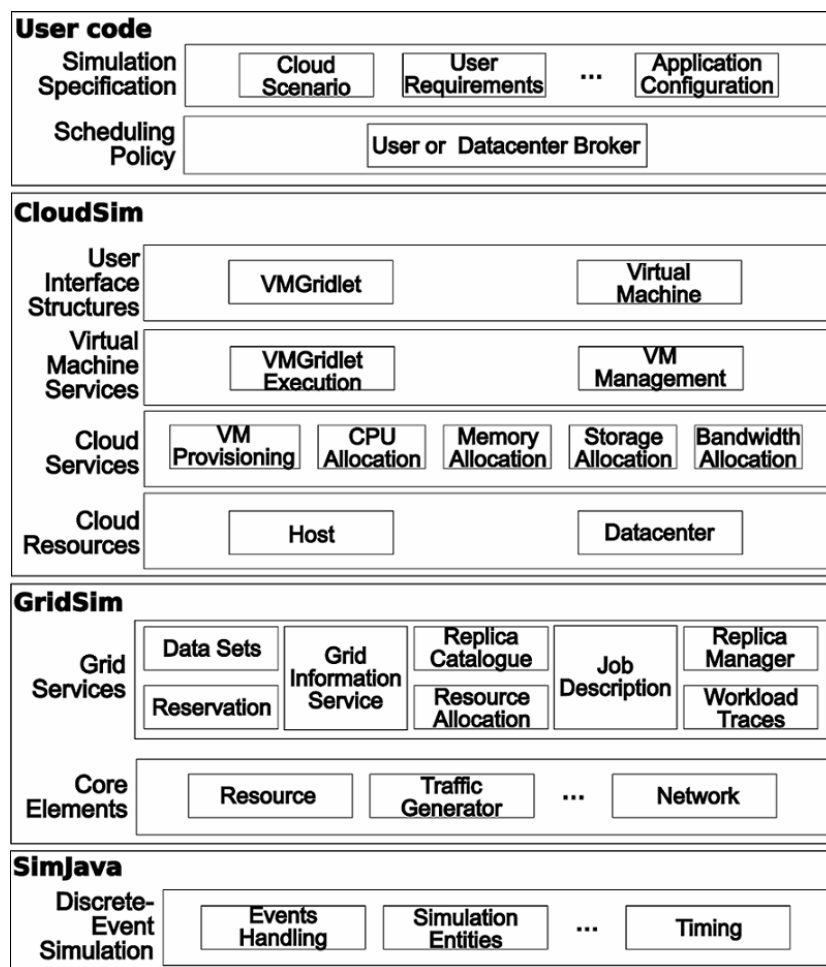


Abbildung 18: CloudSim-Architektur

## 7.2 Implementierung in Cloudi

Die Integration in Cloudi erfolgt wie folgt: Die Eingaben für die Simulation im Cloudi werden durch ein Remapping in CloudSim simuliert. Zum Einstieg und zur Einfachheit werden die virtuellen Maschinen und Hosts als gleich modelliert. Es gibt einmal die Java Klassen Applikationsansicht, Kostenansicht, Serveransicht und VM-Ansicht im Backend. Diese realisieren die Anfragen in Cloudi.

Die zentrale Klasse ist CloudSimulation, die sich die ganzen Informationen aus Cloudi holt und diese so transformiert, dass diese in CloudSim simuliert werden können. Die Klasse SimulationRun ist für die Kommunikation mit CloudSim zuständig. Für die Ergebnisse ist die Klasse SimulationErgebnis verantwortlich. In der Ausgabe werden die Kosten für die eingegebenen Werte für die Simulation ausgegeben und wieviel Zeit für die Simulation gebraucht wird. Da für das Webfrontend das ZK-Framework benutzt wird, müssen die Datentypen ebenfalls umkonvertiert werden. Auch hierfür ist die Klasse SimulationErgebnis zuständig. Für weitere Frontends, muss lediglich die Klasse Simulationsergebnis entsprechend geändert werden. Die Klasse SimulationErgebnis bringt dafür die nötige Übersicht. CloudSimulation kann unverändert weiter verwendet werden.

## 7.3 Projekt zur Cloud Simulation an der TU Dortmund

Auch an der TU Dortmund wird im Bereich der Cloud-Simulationen geforscht. Unter anderem ist die Projektgruppe so auch auf eine Studien-/Diplomarbeit gestoßen, die auf eine seit Mai 2007 am Lehrstuhl für Kommunikationsnetze an der Fakultät Elektrotechnik geschaffene Simulationsumgebung zur Plattformmodellierung und zum Testen von Kommunikationsprotokollen unter realistischen Netzwerkbedingungen, die Simulationen von Cloud-Umgebungen anpassen soll. Die Umgebung vereinigt Tools, wie OMNeT++, MATLAB, RPS und MOOSE. Ziel soll es sein, eine Möglichkeit zu schaffen, über ein Remote-Web-Interface Cloud-Systeme und Konfigurationen zu testen. Als Simulationsparameter sind Netzwerkarchitektur, Benutzeranzahl, Protokolle und Umgebung vorgesehen. Da die Aufgabe der Projektgruppe (siehe Pflichtenheft) eine andere ist, wird diese Arbeit nicht verwendet.

## 7.4 Zusammenfassung und Ausblick

Für die Simulation des zukünftigen Ressourcenbedarfs für die Investitionsplanung wird die Bibliothek von CloudSim benutzt. Allerdings wird lediglich eine homogene Struktur vorausgesetzt, d.h. es wird davon ausgegangen, dass

alle Rechner auf denen die virtuellen Maschinen simuliert werden, die identische Konstellation an Hardware aufweisen. Als zukünftige Weiterentwicklung kann z.B. eine heterogene Struktur eingesetzt werden. Das Ergebnis der Simulation wird momentan nur in einer Tabelle dargestellt, wie in Abb. 53 zu erkennen ist. Diese kann auch graphisch erweitert werden. Wegen der beschränkten Zeit kann dies nicht weiter entwickelt werden. Dieses Ergebnis kann ebenfalls in die Datenbank gespeichert werden, um somit zu vergleichen, ob die simulierten Daten mit den Realitätsdaten übereinstimmen.

## 7.5 Vorhersage

Ein weiteres Ziel, welches in Cloudi verfolgt wird, ist die Vorhersage des zukünftigen Bedarfs an physischer Hardware für die Unterstützung des Betreibers bei der Investitionsplanung. Die Vorhersage stützt sich auf der Methode der kleinsten Quadrate, die in der Apache-Bibliothek (`org.apache.commons-math.stat.regression.SimpleRegression`) unter dem Namen Lineare Regression wiederzufinden ist. Bei der Methode der kleinsten Quadrate wird zu einer

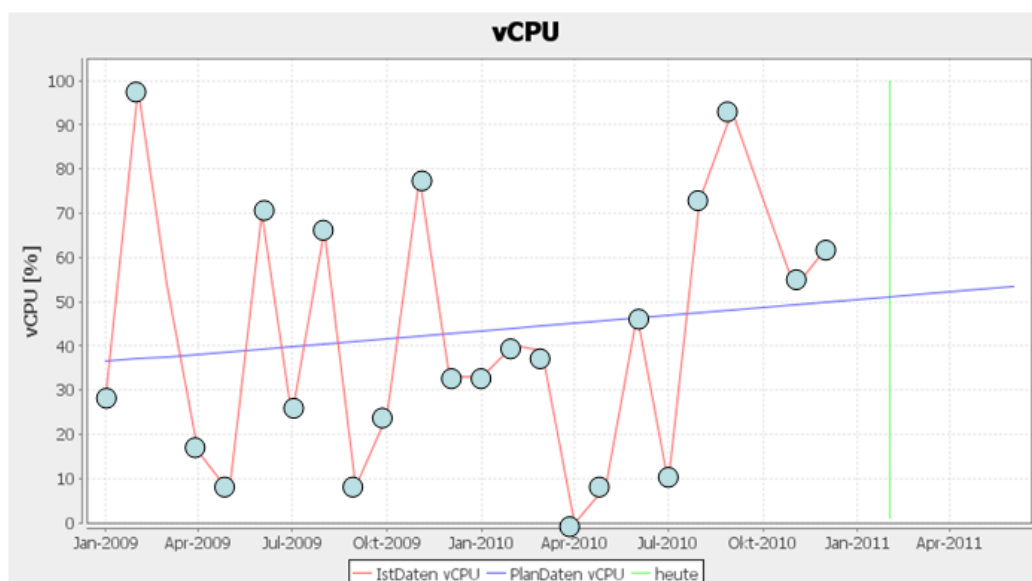


Abbildung 19: Methode der kleinsten Quadrate

Datenpunktswolke eine Kurve gesucht, die möglichst nahe an den Datenpunkten verläuft mit dem Ziel, die Kurvenparameter so zu bestimmen, dass die Summe der quadratischen Abweichungen der Kurve von den beobachteten Punkten minimiert wird. Zurzeit werden für die Vorhersage noch Pseudodaten verwendet. In einem weiteren Schritt kann Ganglia auf den VMs einge-

richtet werden, um somit speziell für die aus Ganglia bezogenen Daten, die Vorhersage durchzuführen.

## 8 Fazit und Ausblick

Nach einem Jahr Projektgruppe kann selbige durchaus einige Ergebnisse vorweisen. Cloudi ist eine Weboberfläche, über die virtuelle Maschinen erstellt und zeitgesteuert gestartet werden können. Sie bietet außerdem weitere Möglichkeiten zur Steuerung der virtuellen Maschinen. Sogar Gruppen von virtuellen Maschinen können gleichzeitig gestoppt oder gestartet werden. Aber neben der einfachen Oberfläche sind noch zahlreiche, große und kleine Erkenntnisse und Erfahrungen in den Bereichen Scheduling, Monitoring, Simulation und Vorhersage hinzugekommen.

Für die Anwender ergeben sich genauso wie für die Betreiber einige Vorteile durch den Einsatz von Cloudi. Die komfortable und leicht verständliche Benutzungsführung, um eine virtuelle Maschine zu erstellen, ermöglicht es, sogar mit geringem Hintergrundwissen schnell und einfach eine passende virtuelle Maschine anzulegen – und zwar rund um die Uhr. Mit einem Klick ist es dann möglich sich aus dem Portal heraus über VNC mit der Konsole oder der grafischen Oberfläche der virtuellen Maschine zu verbinden. Die Überwachung der Maschinen erfolgt vollautomatisch und so bleibt zu jeder Zeit der Ressourcenverbrauch im Blick. Auf dessen Grundlage ist auch die Abrechnung über die in der Oberfläche hinterlegten Kostenstelle möglich. Da in absehbarer Zeit ein SAP-System im ITMC realisiert werden soll, kann so die Abrechnungen direkt über eine dortige Anbindung erfolgen. Zudem fällt das Ausfüllen und Einreichen, sowie die manuelle Bearbeitung von Papieranträgen weg. Generell werden die Aufgaben rund um das Erzeugen und Überwachen von virtuellen Maschinen automatisiert und komfortabler gestaltet. Als Beispiel sei hier das Überwachen und Auswerten der Hosting-Daten genannt. Eher umständliche administrative Aufgaben, wie das Starten und Stoppen von mehreren virtuellen Maschinen (auch zu bestimmten Terminen) ist in Cloudi mit geringem Aufwand schnell erledigt. Die in diesem Projekt eingesetzte Software ist ausnahmslos als Open Source unter Apache-Lizenz und der GNU General Public License veröffentlicht. Es entstehen also keine Lizenzkosten durch den Einsatz der verwendeten Software. Cloudi ist sehr gut geeignet, um eine hohe Anzahl unkritischer virtueller Maschinen zu betreiben. Kritische virtuelle Maschinen sollten, wie bisher beim ITMC, über die Lösung von VMWare betrieben werden, da diese ausgereifter und verlässlicher ist als das bei Cloudi zum Einsatz kommende Open Nebula.

OpenNebula hat viele gute Ansätze und in der neuen Version 2.0 funktioniert schon sehr viel. Gleichzeitig zeigt sich aber auch, dass gerade im Bereich Cloud Computing zur Verbesserung von OpenNebula noch sehr viele Funktionalitäten denkbar sind. Ein Beispiel ist ein verbessertes, polyparametrisiertes Scheduling in der Middleware, das sich natürlich auch auf die Stabilität von

cloudi auswirken würde. Die Migration zu anderen Infrastrukturen innerhalb der UAMR ist mit Blick auf den zusammenwachsenden Hochschulstandort Ruhrgebiet ebenfalls vorteilhaft. Das Accounting von OpenNebula hat ebenfalls noch deutlich – trotz entsprechendem Toolset – Verbesserungspotential: Die Zuweisung von IT-Infrastruktur an mehrere Benutzer sowie eine einfache Möglichkeit unterschiedliche Kostenmodelle und Abrechnungsmechanismen für unterschiedliche Zielgruppen abzubilden fehlt für einen langfristigen, gewinnbringenden Einsatz an der TU Dortmund.

Haizea verspricht viel und wurde immer als Standardlösung und etabliert beworben, ist allerdings für die Praxis noch nicht wirklich als stabil zu bezeichnen. Es bleibt zu hoffen, dass Haizea mittelfristig auch für die Anwendung mit OpenNebula so stabil wird, wie es für Simulationen bereits ist.

Dass libvirt bereits Monitoringdaten sammelt, ist sicherlich eine gute Idee. Schade ist jedoch, dass über OpenNebula auf genau diese Daten nicht zugegriffen werden kann. Auch hier ist mit Blick auf cloudi noch Verbesserungsbedarf.

CloudSim stellt eine gute Möglichkeit dar, ganze Rechenzentren und Cloud-Infrastrukturen zu simulieren. Seine volle Stärke kann CloudSim allerdings erst ausspielen, wenn es – eventuell auch mit Hilfe der Apache Commons-Maths-Library – auf Grundlage der bisherigen Monitoringdaten aus der libvirt oder dem Ganglia Monitoring System zuverlässige Voraussagen sowohl für die gesamte Infrastruktur als auch für einzelne „Kunden“ mit ihren einzelnen Netzwerken, virtuellen Maschinen und Speichern liefern wird.

Trotz der Arbeit von 12 Studierenden über die Dauer eines ganzen Jahres hinweg erscheint noch genügend Arbeit sowohl für das ITMC wie auch für etwaige Diplomanden übrig geblieben zu sein. Bis Cloudi wirklich im ITMC produktiv eingesetzt werden kann, wird es allerdings wohl noch eine Zeit dauern. Die Investitionen werden sich aber unzweifelhaft lohnen.

## A Benutzerhandbuch

Cloudi ist ein Webportal, welches eine benutzerfreundliche Oberfläche zur Verwaltung von virtuellen Maschinen zur Verfügung stellt. Mit Cloudi lassen sich einfach virtuelle Maschinen erstellen und verwalten. Sowohl für einen Nutzer von Cloudi, welcher seine eigenen virtuelle Maschinen starten, stoppen oder migrieren kann, als auch für Administratoren welche viele virtuelle Maschinen auf unterschiedlichen Hosts verwalten können und Vorhersagen ist Cloudi geeignet.

Cloudi ist sowohl für TU Dortmund Nutzer, als auch extern erreichbar. Die TU Dortmund Nutzer können Cloudi über die Webseite des ITMC erreichen, wohingegen externe Nutzer zu der Anmeldeseite unter [www.cloudi.in](http://www.cloudi.in) gehen müssen. Da Cloudi eine webbasierte Lösung ist, lässt sich der Service in gängigen Browsern öffnen. Ein typisches Layout der Bedienoberfläche ist in vier verschiedene Abschnitte unterteilt.

The screenshot shows the Cloudi web interface. At the top, there is a header with the Cloudi logo and the text 'Willkommen bei Cloudi'. Below the header is a navigation menu (red border) with items: 'VM-Anzeige', 'VM-Verwaltung', 'Neue VM erstellen', and 'Benutzerdaten'. The main content area (green border) is titled 'VM erstellen (einfach)' and contains a form with the following fields:

Information	Ihre Angabe
Name der VM*:	<input type="text"/>
Betriebssystem*:	<input type="radio"/> Windows <input checked="" type="radio"/> Ubuntu Desktop
Vorlagen verwenden:	<input type="text" value="Eigene Einstellungen"/>
Anzahl der CPUs*:	<input type="range" value="1"/> 1 Stück
Größe des Arbeitsspeichers (GB)*:	<input type="range" value="1"/> 1 GB
Größe der 2. Festplatte (GB):	<input type="range" value="0"/> Keine 2. Festplatte
VNC-Passwort*:	<input type="text" value="lycqZ"/>

\*) Pflichtfelder

At the bottom, there is a navigation bar (yellow border) with 'VM erstellen' and 'VM-Details bearbeiten', and a 'Zurück zur Startseite' link. The footer (orange border) contains 'Max abmelden', '© by PG-Cloudi 2011 | Impressum', and 'Release Date: 02.02.11'.

Abbildung 20: Standardlayout einer Cloudi Webseite

Der rote Bereich ist die Menüleiste. Der Cloudi-User hat hier die Möglichkeit verschiedene Funktionen des Programms aufzurufen. Die Menüleiste ist statisch was bedeutet, dass die Einträge sich nicht verändern im Laufe der Benutzung.

Der grüne Bereich ist der Informationsbereich. Hier werden die passenden Informationen für den User bereitgestellt.

Der **gelbe** Bereich ist eine Taskleiste, die dem Benutzer verschiedene Programmfeatures bereit stellt, die abhängig von der aktuellen Informationsbereichsseite sind. Sie passt sich somit dynamisch dem Informationsbereich an.

Der **orange** Bereich ist eine Informationszeile, die dem Nutzer Informationen über das aktuelle Release-Datum liefert und zusätzlich weitere Links zu den PG-Urhebern bereitstellt. Wichtigstes Element ist aber der Abmelde-Button, der es den Benutzern erlaubt sich von dem Web-Service abzumelden und die Web-Session zu beenden.

Dieses Layout, lässt sich auf allen Seiten des Web-Service finden und dient somit einer besseren Orientierung und erleichtert das Nutzen des Web-Services. Ein Ausnahme stellt hier nur das starten der VNC-Konsole dar. Dort wird eine neues Fenster geöffnet, welches nicht im Cloudi Design zur Verfügung steht. Ein einfaches Arbeiten und eine intuitive und leicht erlernbare Bedienung sind somit mit Cloudi gegeben.

## Das Arbeiten als Benutzer

### Erste Verwendung über das ITMC

Wenn Sie Angehöriger der TU Dortmund sind, sollten Sie sich über die Webseite des ITMC anmelden. Ansonsten steht Ihnen noch die Webseite [www.cloudi.in](http://www.cloudi.in) zur Verfügung. Unter der Rubrik myITMC gibt es die Möglichkeit durch einen Link **Virtuelle Maschine erstellen** auf die Cloudi-Oberfläche zu gelangen.



Abbildung 21: myITMC Seite mit dem Hyperlink auf das Cloudi-Portal

Benutzen Sie Cloudi das erste Mal, so kommen Sie auf eine Anmeldeseite, wo die vollständigen Benutzerdaten eingeben werden müssen. Einige der Felder sind schon vorausgefüllt und können nicht geändert werden.

Ein besonderes Feld bei der Anmeldung ist der Passwort-Parameter. Dieser Parameter muss nicht ausgefüllt werden, kann aber. Sie haben 2 verschiedene Möglichkeiten.

1. Wenn ein Passwort eingegeben wird, so muss das Passwort bei jeder Einwahl über das ITMC beim Cloudistart eingegeben werden. Jedoch können Sie sich bei Cloudi nicht nur über die ITMC Seite anmelden, sondern zusätzlich auch noch über die Webseite [www.cloudi.in](http://www.cloudi.in). Falls Sie die TU Dortmund verlassen, haben Sie so weiterhin Zugriff auf Ihr virtuelle Maschinen.
2. Wenn kein Passwort eingegeben wird, haben Sie nur über die ITMC Webseite Zugriff auf Cloudi. Beim späteren Starten von Cloudi muss kein Passwort eingegeben werden. Die Anmeldeinformationen bezieht Cloudi aus der MyITMC Anmeldung.

Bitte beachten Sie, dass das Ändern des Benutzernamens nicht möglich ist, da die Authentifizierung beim Anmelden am ITMC erfolgt. Das Ändern des Passwortes hingegen ist jederzeit noch möglich. Nach der erfolgreichen Registrierung und dem Starten von Cloudi kommen Sie zur Startseite.



Abbildung 22: Cloudi Startseite

Die Startseite beinhaltet alle virtuellen Maschinen die der angemeldete Benutzer betreibt. Die nachfolgenden Attribute sind wichtige Kennzahlen, die jede virtuelle Maschine hat. Neben dem vom Benutzer festgelegten Namen, kann der aktuelle Status der virtuellen Maschine, die aktuelle RAM Größe und die CPU Anzahl angezeigt werden. Zusätzlich wird auch noch angezeigt, auf welchem Host die virtuelle Maschine läuft. Nach der Registrierung und der ersten Anmeldung sind hier noch keine virtuellen Maschinen zu sehen. Diese müssen erst noch angelegt werden. Im weiteren Verlauf dieses Kapitels werden Sie verstehen, wie dies funktionieren wird.

Im oberen Bereich des Fensters ist die Menüleiste zu sehen.



Abbildung 23: Menüleiste

Durch einen Klick auf VM-Anzeige, werden alle Ihre virtuellen Maschinen angezeigt. Wenn Sie nun eine virtuelle Maschine verändern möchten, gibt es hier mehrere Möglichkeiten. Wählen Sie von der Startseite eine virtuelle Maschine durch einen Klick mit der rechten Maustaste aus und das folgende Kontextmenü erscheint.



Abbildung 24: Kontextmenü der Startseite

Hier stehen viele unterschiedliche Funktionen zur Verfügung, die Sie für die ausgewählte virtuelle Maschine aufrufen können. Diese werden im späteren Verlauf dieses Handbuches noch vorgestellt.

**Hinweis:**

Geben Sie kein Passwort bei der Registrierung ein und verlassen Sie die TU Dortmund, haben Sie keinen Zugriff auf Ihre virtuellen Maschinen. Diese werden beim Verlassen der TU Dortmund nach 3 Monaten unwiderruflich gelöscht.

**Erste Verwendung bei externen Nutzern (nicht TU Dortmund)**

Wenn Sie nicht Angehöriger der TU Dortmund sind haben Sie trotzdem Zugriff auf Cloudi, indem Sie über Ihren Browser auf die Webseite [www.cloudi.in](http://www.cloudi.in) gehen.



Abbildung 25: Anmeldeseite von Cloudi

Durch einen Mausklick auf den Button **Registrieren** gelangen Sie zu dem Registrierungsformular, in dem die für Ihre Anmeldung notwendigen Daten erfasst werden.

Tragen Sie bitte in die dafür vorgesehenen Formularfelder die entsprechenden Daten ein. Die mit einem \* gekennzeichneten Felder sind Pflichtfelder. Geben Sie einen von Ihnen wählbaren Benutzernamen ein. Das Kennwort muss zur Bestätigung ein zweites Mal eingegeben werden.



Abbildung 26: Registrierungsformular von Cloudi

Wenn Sie das Formular vollständig ausgefüllt haben, bestätigen Sie bitte Ihre Eingaben mit Klick auf den Button Benutzer anlegen. Sie gelangen wieder zur Anmeldeseite, wo Sie sich nun sofort als registrierter Benutzer anmelden können.

Sollte die Registrierung nicht funktionieren, kontrollieren Sie bitte, ob Sie alle Pflichtfelder korrekt ausgefüllt haben und versuchen Sie es erneut.

Als Nutzer haben Sie die Möglichkeit alle Ihre Informationen die Sie bei der Registrierung eingegeben haben, exklusive das Passwort anzeigen zu lassen. Das Ändern Ihre Benutzerinformation ist inklusive das Passwortes möglich.

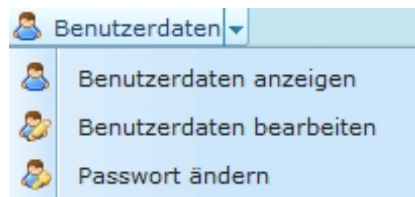


Abbildung 27: Benutzerdaten

## Funktionen von Cloudi

Als angemeldeter Benutzer können Sie verschiedene Funktionen für virtuelle Maschinen aufrufen über die Menüleiste aufrufen.

### Menüpunkt 1 - VM-Anzeige

Im ersten Menüpunkt können Sie alle ihre virtuellen Maschinen sich anzeigen lassen. Dieser Menüleisteneintrag bringt Sie zur Startseite zurück.

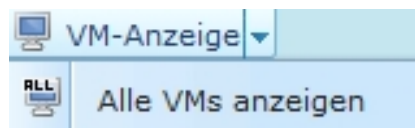


Abbildung 28: Erster Menüpunkt

### Menüpunkt 2 - VM-Verwaltung

Im zweiten Menüpunkt gibt es weitere Funktionen für virtuelle Maschinen. Wählen Sie auf der Startseite gezielt eine oder mehrere virtuelle Maschinen durch eine Klick auf diese aus (gekennzeichnet durch den Haken an der linken Seite) und führen Sie die gewünschte Funktion im Menüpunkt VM-Verwalten auf diesen aus.

Sie können virtuelle Maschinen starten, pausieren, fortsetzen, stoppen, migrieren und löschen. Dann verändern die virtuellen Maschinen entsprechend

Ihrer Anwahl Ihren Status. Über das Menü können Sie gleichzeitig beliebig viele virtuelle Maschinen ansteuern. Anders als im Kontextmenü, wo jeweils nur Kontrolle über eine virtuelle Maschine besteht. Das Kontextmenü öffnen Sie durch einen Klick mit der rechten Maustaste auf eine beliebige virtuelle Maschine.

Neben dem Ändern des Status können Sie auch eine virtuelle Maschine auswählen und sich unter VM-Details anzeigen alle Details dieser virtuellen Maschine anzeigen lassen, wie viele CPUs und Arbeitsspeicher diese virtuelle Maschine besitzt oder welchen Namen Sie hat. Zusätzlich werden Graphen zur Prozessor-, Speicher- und Netzwerkauslastung dieser virtuellen Maschine angezeigt. Unter Virtuelle Maschine-Bearbeiten können Sie die Daten einer virtuellen Maschine auch verändern. Mit Klick auf den Button VNC-Konsole gelangen Sie zum Desktop dieser virtuellen Maschine. Diese Funktion können Sie nur für genau eine virtuelle Maschine aufrufen.

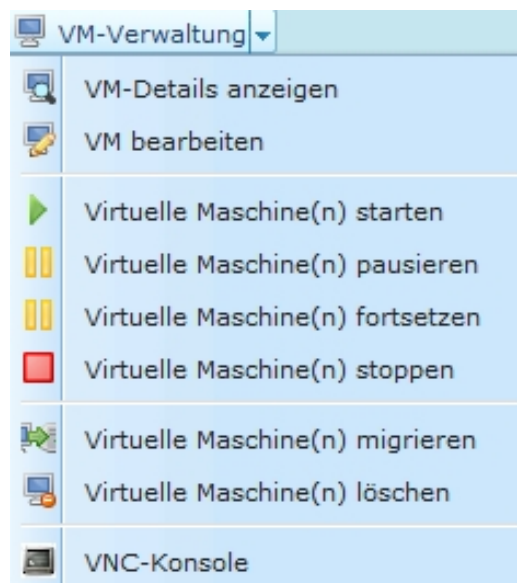


Abbildung 29: Zweiter Menüpunkt

Zum migrieren einer virtuellen Maschine auf einen andern Host wählen Sie eine oder mehrere virtuelle Maschinen aus und klicken Sie auf Virtuelle Maschine(n) migrieren. Es öffnet sich ein kleines Pop-up Fenster, wo Sie auf der linken Seite die zu migrierende virtuelle Maschine und auf der rechten Seite den neuen Host auswählen können.

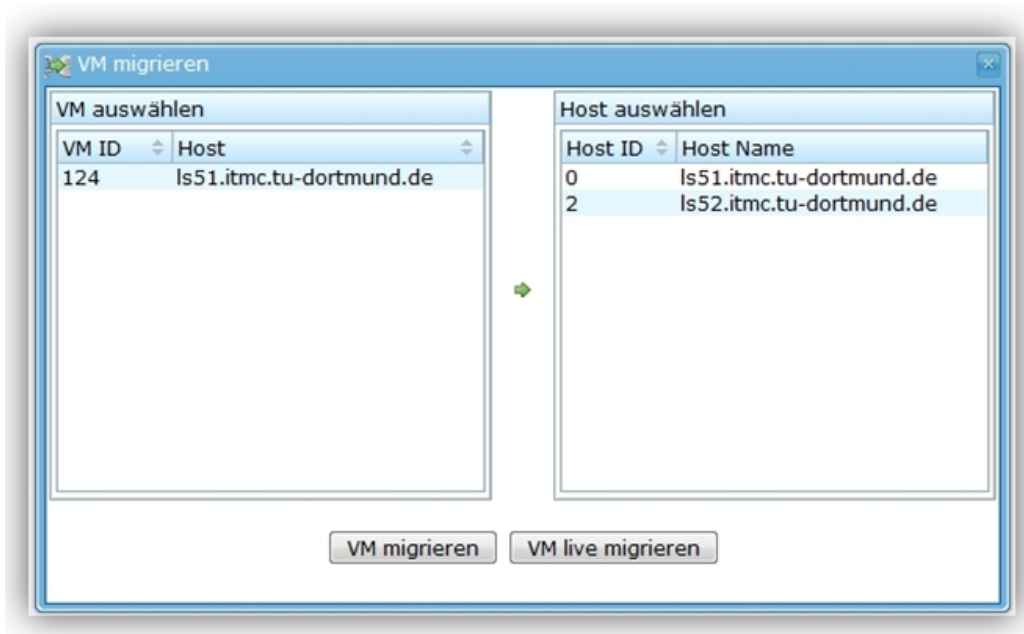


Abbildung 30: Virtuelle Maschine migrieren

Das reine VM migrieren bedeutet, dass die virtuelle Maschine heruntergefahren wird, dann migriert und anschließend wieder hochgefahren wird. Live Migrieren bedeutet, dass die VM sofort auf den anderen Server migriert wird ohne dass sie heruntergefahren und neu gestartet werden muss.

### Menüpunkt 3 - Neue VM erstellen

Im dritten Menüpunkt können Sie neue virtuelle Maschinen erstellen. Hier gibt es verschiedene Detaillierungsgrade beim Erstellen; Einfach und Erweitert.

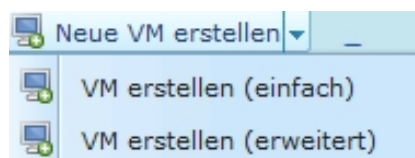


Abbildung 31: Dritter Menüpunkt

Durch einen Klick auf Einfach gelangt der User zu einer Übersichtsseite mit den wichtigsten Parameter die zur Erstellung einer VM wichtig sind. Vorgefertigte Templates erleichtern die Hardwareauswahl.

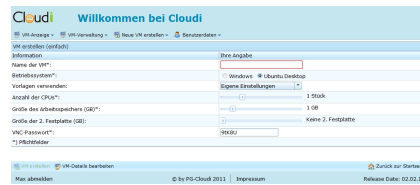


Abbildung 32: VM erstellen einfach

Klickt der User auf Virtuelle Maschine-Erstellen Erweitert gelangt er ins Untermenü. Es gibt fünf Schritte, die Sie durchlaufen müssen. Der so genannte Wizard hilft Ihnen beim Erstellen einer virtuellen Maschine. Bitte füllen Sie alle Angaben korrekt aus. Erst dadurch kann gewährleistet sein, dass Ihre virtuelle Maschine Ihren Wünschen entspricht richtig erstellt wird. Im Gegensatz zu der einfachen Erstellung gibt es hier mehr Parameter die frei wählbar sind.

In der Taskleiste können Sie durch Klicks auf Weiter und Zurück entsprechend die Daten ändern. Durch einen Klick auf dem entsprechenden Schritt des Erstellen Wizards gelangen Sie gleich mehrere Schritte zurück oder nach vorne.

Im ersten Schritt müssen Sie mindestens den Namen der VM angeben Alle weiteren Felder sind keine Pflichtfelder.



Abbildung 33: Step 1/5 Namensvergabe

Nachdem Sie die Felder ausgefüllt haben klicken Sie unten in der Menüleiste auf den Button Weiter. Sie können auch jeder Zeit das Erstellen abbrechen. Dazu klicken Sie auf den Button Anlegen Beenden rechts unten in der Menüleiste.

Wenn Sie auf weiter geklickt haben gelangen Sie zum folgenden Fenster. Auf diesen Seiten steht eine Hilfe zur Verfügung, indem Sie auf die Fragzeichen neben den Parametern klicken

**Cloudi Willkommen bei Cloudi**

VM-Anzeige VM-Verwaltung Neue VM erstellen Benutzerdaten

Wizard Step 2 / 5 - Hardwarekonfiguration

VM Art	CPU [in Cores]	RAM Größe [in GB]
<input checked="" type="radio"/> S	1	2
<input type="radio"/> M	2	4
<input type="radio"/> L	2	8
<input type="radio"/> Eigene Spezifikation		

Festplatte 1 - Bootfestplatte  
 Größe in GB:   
 SATA oder PATA-Laufwerk   
 Die Art des Mediums   
 Save  yes  no  
 Read Only  yes  no  
 Clonen  yes  no

Festplatte 2  
 Festplatte 3  
 Festplatte 4

Bootauswahl   
 Imageauswahl

← zurück  Schritt 1 Schritt 2 Schritt 3 Schritt 4 Schritt 5 Zurück zur Startseite  
 Max abmelden © by PG-Cloudi 2011 Impressum Release Date: 02.02.11

Abbildung 34: Step 2/5 Hardwarekonfiguration

**Cloudi Willkommen bei Cloudi**

VM-Anzeige VM-Verwaltung Neue VM erstellen Benutzerdaten

Wizard Step 3 / 5 - Netzwerkkonfiguration

Netzwerk 1  
 Virtuelles Netzwerk:   
 VLan

Netzwerk 2  
 VNC Passwort  
 Passwort:

← zurück  Schritt 1 Schritt 2 Schritt 3 Schritt 4 Schritt 5 Zurück zur Startseite  
 Max abmelden © by PG-Cloudi 2011 Impressum Release Date: 02.02.11

Abbildung 35: Step 3/5 Netzwerkkonfiguration



Abbildung 36: Step 4/5 Zeitraum festlegen

Im vierten Schritt können Sie bestimmen, ob Ihre virtuelle Maschine sofort oder zu einem bestimmten Zeitpunkt gestartet werden soll.



Abbildung 37: Step 5/5 Bestätigung

Sie können Ihre neue virtuelle Maschine über den Button **VM anlegen** erstellen/starten.

## Menüpunkt 4 - Benutzerdaten

Diese Funktion steht nicht jedem Nutzer im vollen Umfang zur Verfügung. Nutzer der TU Dortmund können nicht Ihren Benutzernamen ändern.

Sie können sich Ihre Benutzerdetails anzeigen lassen und gewisse Parameter ändern. Bitte wählen Sie den dazu passenden Menüpunkt aus.

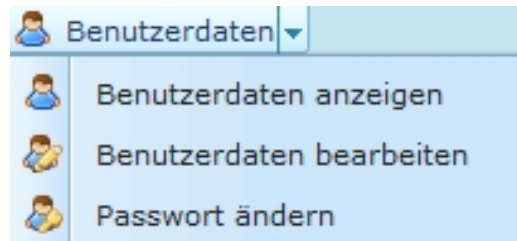


Abbildung 38: Vierter Menüpunkt

## Das Arbeiten als Administrator

### Funktionen im Administratorbereich

Wenn Sie sich als Administrator anmelden haben Sie neben den Benutzermöglichkeiten, weitere Funktionen die Ihnen zur Verfügung stehen.

Daher sollten Sie zuerst den vorherigen Abschnitt zum Thema Arbeiten als Benutzer lesen, bevor Sie mit diesem Kapitel anfangen. Wenn Sie sich als Administrator angemeldet haben, erhalten Sie eine ähnlich aussehende Startseite wie der Benutzer.



Abbildung 39: Startseite von Cloudi als Administrator

Die erste Änderung die auffällt ist, dass ein Administrator alle virtuellen Maschinen aller Benutzer sieht. Somit kann der Administrator alle virtuellen Maschinen von allen Benutzern *steuern*. Das Ändern des Status, das Löschen und Migrieren oder das Ändern von virtuellen Maschinen Details ist möglich.

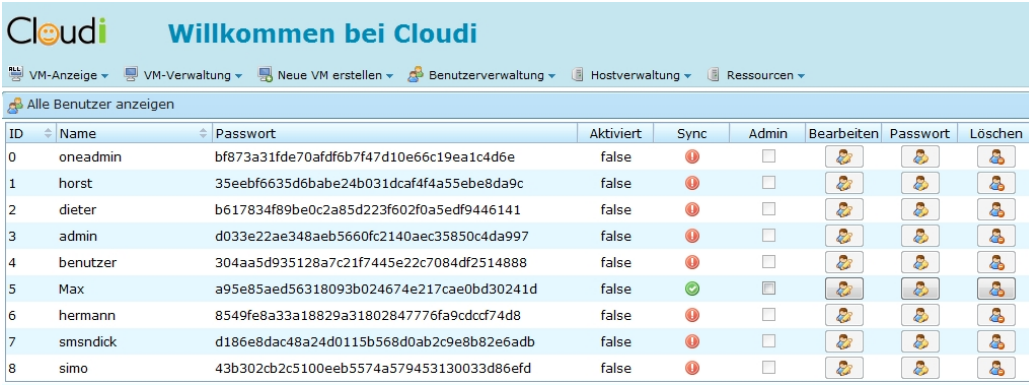
Der Administrator hat im Vergleich zum Benutzer eine leicht veränderte Menüleiste.



Abbildung 40: Administratoren Menüleiste

Als Administrator sehen Sie alle virtuelle Maschinen aller Benutzer gleichzeitig. Dies kann die Startseite unübersichtlich werden. Bei einem Klick auf VM-Anzeige haben die die Möglichkeit alle virtuellen Maschinen nach ihrem Status zu filtern.

Neu hinzugekommen in der Menüleiste sind die Hostverwaltung, Ressourcen und die Benutzerverwaltung. Die Benutzerverwaltung erlaubt es dem Administrator, ähnlich wie dem Benutzer seine Details sich anzuschauen und zu ändern. Zusätzlich besteht aber noch die Möglichkeit, auf alle registrierten Cloudi Nutzer zuzugreifen.



ID	Name	Passwort	Aktiviert	Sync	Admin	Bearbeiten	Passwort	Löschen
0	oneadmin	bf873a31fde70afdf6b7f47d10e66c19ea1c4d6e	false	!	<input type="checkbox"/>			
1	horst	35eebf6635d6babe24b031dcaf44a55ebe8da9c	false	!	<input type="checkbox"/>			
2	dieter	b617834f89be0c2a85d223f602f0a5edf9446141	false	!	<input type="checkbox"/>			
3	admin	d033e22ae348aeb5660fc2140aec35850c4da997	false	!	<input type="checkbox"/>			
4	benutzer	304aa5d935128a7c21f7445e22c7084df2514888	false	!	<input type="checkbox"/>			
5	Max	a95e85aed56318093b024674e217cae0bd30241d	false	✓	<input checked="" type="checkbox"/>			
6	hermann	8549fe8a33a18829a31802847776fa9cdccf74d8	false	!	<input type="checkbox"/>			
7	smsndick	d186e8dac48a24d0115b568d0ab2c9e8b82e6adb	false	!	<input type="checkbox"/>			
8	simo	43b302cb2c5100eeb5574a579453130033d86efd	false	!	<input type="checkbox"/>			

admin abmelden | © by PG-Cloudi 2011 | Impressum | Release Date: 02.02.11

Abbildung 41: Benutzerverwaltung

Über dem Menüpunkt **Alle Benutzer anzeigen** besteht die Möglichkeit sich alle Benutzer anzeigen zu lassen, um so eine Liste aller Benutzer zu bekommen. In der Liste kann ein ausgewählter Nutzer gelöscht, bearbeitet oder ein neues Passwort vergeben werden. Durch das Setzen des Hakens auf Admin kann ein User zum Administrator erklärt werden. Seien Sie aber Vorsichtig bei der Rechtevergabe. Ebenso wird dem Administrator ein Log zur Verfügung gestellt. Hierdurch kann eine Kontrolle der Benutzer erfolgen und nachvollzogen werden, welcher Benutzer welche Aktivitäten durchgeführt hat. Dazu steht ein Datumsfilter zur Verfügung, um einen Zeitraum zur Verfolgung der Benutzeraktivitäten festzulegen.



Abbildung 42: Menü für die Benutzerverwaltung

Während ein Benutzer nur die Möglichkeit hat virtuelle Maschinen zu verwalten, muss der Administrator natürlich auch in der Lage sein, die Hosts zu verwalten. Unter dem Menüpunkt **Hostverwaltung** findet sich diese Möglichkeit.

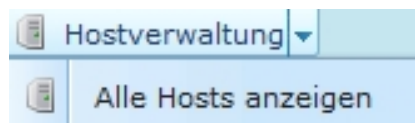


Abbildung 43: Menü für die Hostverwaltung

### Vorhersage für Ressourcen

Die Vorhersage ermittelt den zukünftigen Bedarf an physischer Hardware für die Unterstützung des Betreibers. Die Vorhersage ist in der Administratoransicht in der Menüleiste an letzter Stelle zu finden. Um die Vorhersage durchzuführen, wählen Sie unter dem Menüpunkt **Ressourcen** die Vorhersage aus.

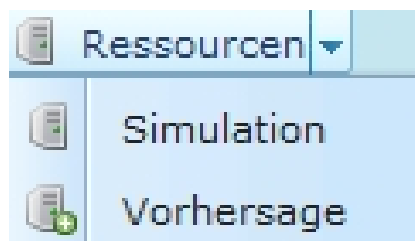


Abbildung 44: Menü für die Vorhersage

Sie gelangen anschliessend in das Untermenü, wo Sie das Start- und das Enddatum festlegen können, um die Ist-Daten für die Vorhersage zu Selekt-

tieren. Anschliessend wählen Sie das Intervall, um die Plan-Daten, welche die Vorhersage darstellen, zu bekommen. Ein Intervall von 6 bedeutet, dass eine Vorhersage für die nächsten 6 Monate erstellt wird.

Abbildung 45: Eingabe-Menü

Klicken Sie im nächsten Schritt auf Vorhersage um eine Tabellen mit den Ist- und Plan-Daten sowie eine Graphische Auswertung dieser Daten zu erhalten.

Datum	Anzahl VMs	vCPU [%]	vMEM [GB]	Storage [GB]
01/2009	7981	28	47	138
02/2009	7397	98	95	37
03/2009	17196	54	40	108
04/2009	12159	16	81	170
05/2009	1526	7	54	182
06/2009	19161	70	14	34
07/2009	2147	24	34	118
08/2009	4530	66	35	179
09/2009	9287	8	24	4
10/2009	6176	24	36	190
11/2009	10831	78	95	192
12/2009	10999	33	55	77
01/2010	11747	33	54	38
02/2010	1603	40	54	165
03/2010	6307	39	66	37

Abbildung 46: Tabelle-IstDaten

In der Tabelle können die Ist-Daten im gewählten Zeitraum abgelesen werden, welche noch zufällig generiert werden (Pseudo-Daten). Dazu gehören Anzahl VMs, vCPU, vMem und Storage. Dabei bedeuten vCPU virtuelle CPU, vMem virtueller Arbeitsspeicher und Storage virtuelle Festplatte.

Willkommen bei Cloudi

VM-Anzeige VM-Verwaltung Neue VM erstellen Benutzerverwaltung Hostverwaltung Ressourcen

Vorhersage: Wählen Sie ein Start und ein Enddatum

PlanDaten [ 1 - 15 / 30 ]

Datum	Anzahl VMs	vCPU [%]	vMEM [GB]	Storage [GB]
01/2009	8904	36	57	112
02/2009	8913	37	56	112
03/2009	8920	38	55	113
04/2009	8929	38	54	114
05/2009	8937	39	53	115
06/2009	8946	39	53	115
07/2009	8954	40	52	116
08/2009	8963	41	51	117
09/2009	8971	41	50	118
10/2009	8980	42	49	119
11/2009	8988	42	48	119
12/2009	8996	43	48	120
01/2010	9005	43	47	121
02/2010	9013	44	46	122
03/2010	9021	45	45	123

admin abmelden © by PG-Cloudi 2011 Impressum Zurück zur Startseite Release Date: 02.02.11

Abbildung 47: Tabelle-PlanDaten

Ebenso erhält der Benutzer die Plan-Daten entsprechend dem oberen Intervall.

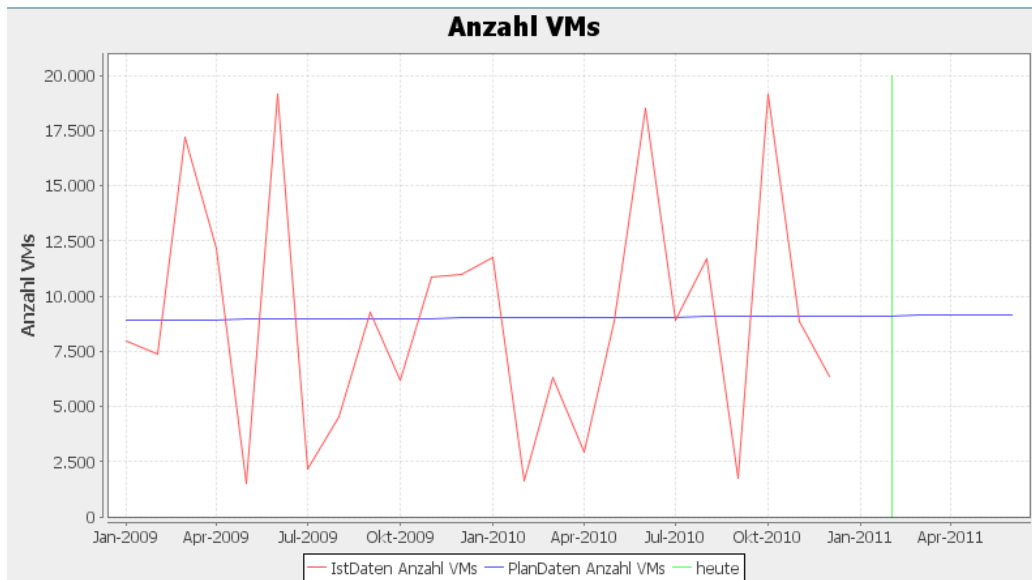


Abbildung 48: Graphische Auswertung der Entwicklung von Anzahl VMs

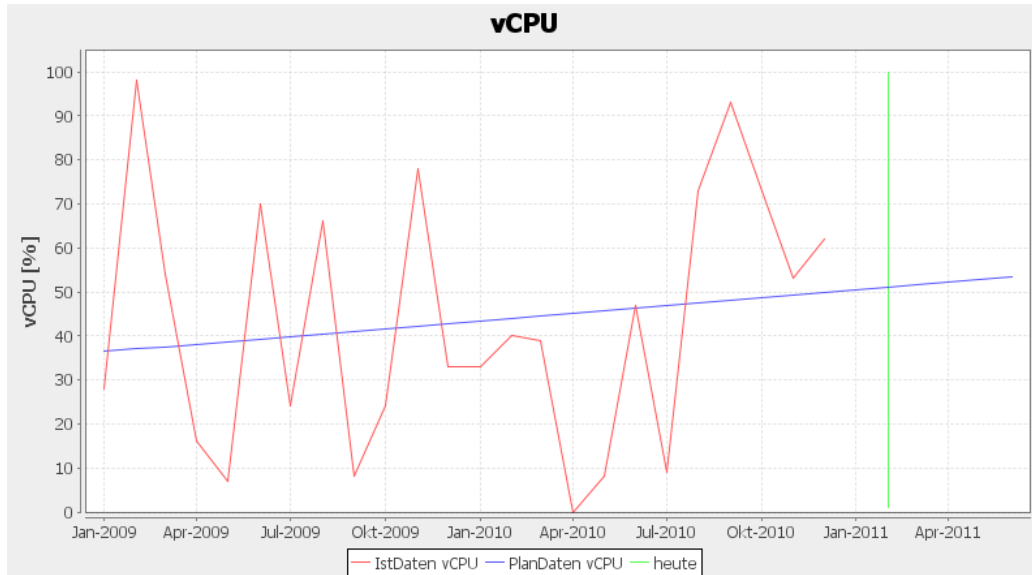


Abbildung 49: Graphische Auswertung der Entwicklung von vCPU



Abbildung 50: Graphische Auswertung der Entwicklung von vMEM

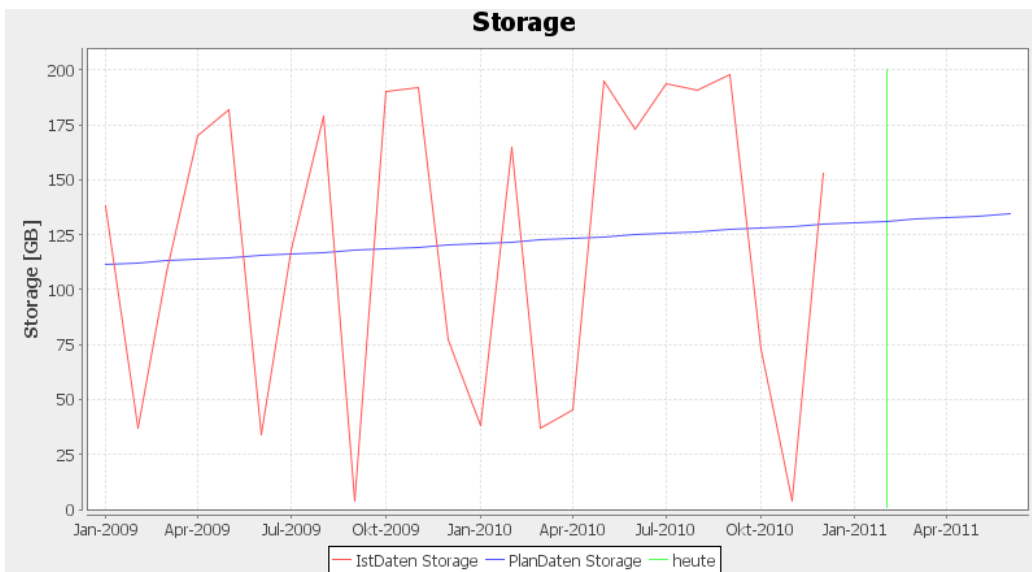


Abbildung 51: Graphische Auswertung der Entwicklung von Storage

In der graphischen Auswertung wird ersichtlich, wie sich die Ist-Daten (rote Kurve) im Vergleich zu den Plan-Daten (blaue Gerade) verhalten. Die grüne Gerade spiegelt das heutige Datum wieder. So ist es möglich den zukünftigen Bedarf an physischer Hardware vorherzusehen und entsprechende Maßnahmen einzuleiten.

## Simulation

Die Simulation beschäftigt sich mit dem Bedarf an Servern und virtuellen Maschinen für die Unterstützung des Betreibers bei der Investitionsplanung. Durch die Simulation können sie herausfinden mit wieviel Ressourcen die Nachfrage der Benutzer abgedeckt werden kann. Die Kosten, die für die jeweiligen Benutzer entstehen würden, können ebenfalls mit der Simulation ermittelt werden. Somit hätten Benutzer eine Vorstellung von den Kosten, die auf Sie zukommen würden.

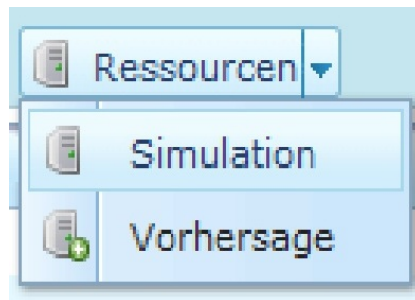


Abbildung 52: Menü für die Simulation

Um die Simulation durchzuführen, wählen Sie unter dem Menüpunkt **Ressourcen** die Simulation aus.

Sie gelangen anschliessend in das Untermenü, wo Sie Angaben zu den Servern, Kosten, virtuellen Maschinen und den Applikationen die simuliert werden sollen machen müssen.

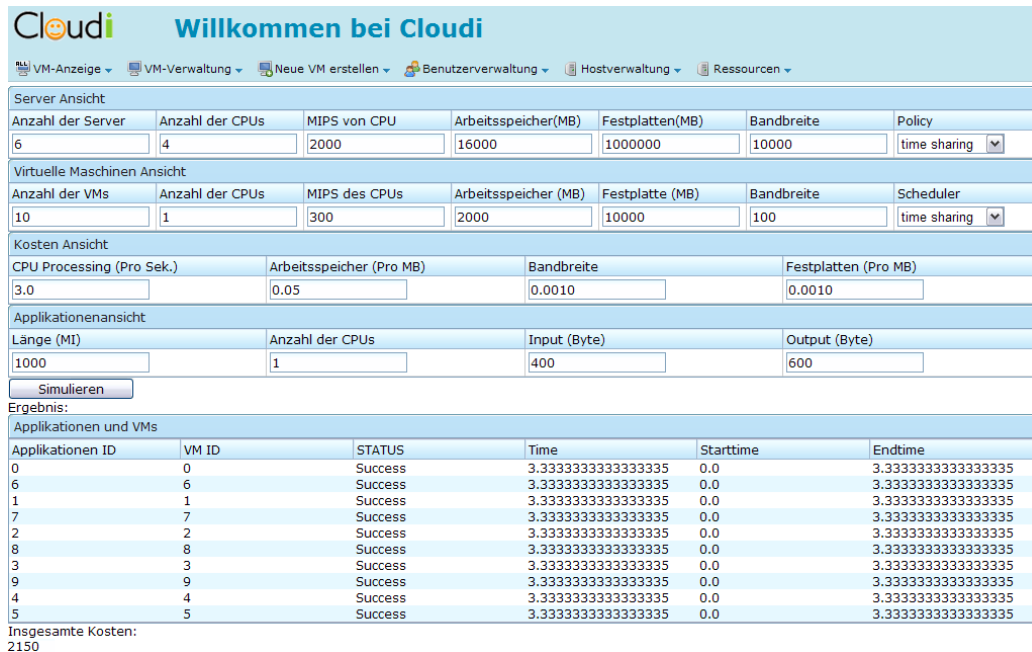


Abbildung 53: Simulation-Untermenü

Angaben zu den Applikationen (MI=Million instruction) , die auf den virtuellen Maschinen laufen sollen, sind dahingehend wichtig, das man sehen möchte, ob die virtuellen Maschinen dem gerecht werden können. Damit werden die Ansprüche an die virtuellen Maschinen definiert, die wiederum auf den Servern laufen. Die virtuellen Maschinen müssen sich die Ressourcen der Server teilen und deshalb gibt es die Möglichkeit zwischen den Schedulingverfahren Time Sharing und Space Sharing zu wählen.

Im folgenden werden die Eingabefelder einzeln beschrieben:

**Server Ansicht**

Hier geben Sie bitte ein wieviele Server mit der Anzahl der CPU die diese besitzen sollen ein. Weitere Eingabe sind MIPS (Million instructions per second) der CPU, wieviel Arbeitsspeicher, Festplattenkapazität und Netzwerkbandbreite vorhanden sein soll.

**Kosten Ansicht**

Hier geben Sie bitte ein, welche Kosteneinheiten pro CPU Processing pro sekunde, Arbeitsspeicher pro MB, Festplatte pro MB und Bandbreite dem Benutzer belastet werden sollen.

### Virtuelle Maschinen Ansicht

Hier geben Sie entsprechende Daten für die virtuellen Maschinen ein.

### Applikationen Ansicht

Hier geben Sie entsprechende Daten für die Applikationen ein, die auf den virtuellen Maschinen laufen sollen.

Nun klicken Sie bitte auf den Button **Simulieren**. Unten erscheint das Ergebnis der Simulation. Hier sehen Sie, ob die Applikationen mit jeweilige ID erfolgreich durchgelaufen sind, wie lange Sie gelaufen sind, Start und Ende und die Kosten die dadurch verursacht wurden

Ergebnis:

Applikationen und VMs					
Applikationen ID	VM ID	STATUS	Time	Starttime	Endtime
0	0	Success	3.333333333333335	0.0	3.333333333333335
6	6	Success	3.333333333333335	0.0	3.333333333333335
1	1	Success	3.333333333333335	0.0	3.333333333333335
7	7	Success	3.333333333333335	0.0	3.333333333333335
2	2	Success	3.333333333333335	0.0	3.333333333333335
8	8	Success	3.333333333333335	0.0	3.333333333333335
3	3	Success	3.333333333333335	0.0	3.333333333333335
9	9	Success	3.333333333333335	0.0	3.333333333333335
4	4	Success	3.333333333333335	0.0	3.333333333333335
5	5	Success	3.333333333333335	0.0	3.333333333333335

Insgesamte Kosten:  
2150

Abbildung 54: Simulation-Ergebnisse

## B Installationshandbuch

Für den Betrieb von Cloudi sind folgende Softwarepakete notwendig:

- MySQL
- Tomcat, Jetty, o.ä.
- Apache Webserver inkl. OpenSSO- und mod\_jk-Modul
- OpenNebula 2.0 (inkl. dahinterliegender Infrastruktur mit z.B. NFS, NTP, LDAP, kvm und Co.
- Ganglia
- Haizea

Sämtliche Pakete können dabei auf unterschiedlichen Rechnern und teilweise sogar Clustern betrieben und gewartet werden. Teilweise ist es lediglich notwendig Dateisysteme per z.B. NFS zu importieren. Die Verbindungen zwischen den einzelnen Produkten stellt die Tabelle 3 dar.

### B.1 Ganglia

Auf jeder virtuellen Maschine sowie auf einem Server muss ein Ganglia-Monitor installiert werden. Bei Ubuntu geht das relativ leicht durch

```
sudo apt-get -y install ganglia-monitor
```

und einem Neustart. Alternativ kann Ganglia auch auf <http://ganglia.sourceforge.net> heruntergeladen und gemäß Anleitung installiert werden. In der Datei `/etc/ganglia/gmond.conf` muss zusätzlich noch folgende Variablen folgendermaßen geändert werden:

```
Name= "PGCloudi"  
Owner="PGCloudi"
```

In der Projektgruppe werden die Packages im Ubuntu-Repository verwendet. Außerdem ist noch unter <http://ls03.itmc.tu-dortmund.de/ganglia> das PHP-Webfrontend zu Ganglia installiert.

Von auf	Tomcat	Apache Webserver	OpenNebula 2.0	Ganglia	Haizea
MySQL	Hibernate TCP-Port 3306	bei Bedarf: phpmyadmin TCP-Port 3306	TCP-Port 3306	-	-
Apache	mod_jk TCP-Port 8009	-	-	-	-
OpenNebula	Java OCA API 2.0 TCP-Port 2633	bei Bedarf: Management Console TCP-Port 2633	- -	UDP-Port 8649 TCP-Port 8649 TCP-Port 8652	TCP-Port 2633
Ganglia	Dateisystem	bei Bedarf: Ganglia-Webfrontend	-	-	-

Tabelle 3: Verbindungen zwischen den einzelnen Softwarepaketen im Umfeld von ClouDi

## B.2 Apache inkl. OpenSSO und mod\_jk

Der Apache kann einfach aus dem Repository der verwendeten Distribution oder von der Homepage <http://httpd.apache.org/> heruntergeladen und installiert werden. Die Schwierigkeiten liegen in den Modulen mod\_jk sowie OpenSSO.

### B.2.1 OpenSSO

OpenSSO wird als Policy-Agent-Modul von sun angeboten und wird einfach gemäß Anleitung der Entwicklungsabteilung des ITMCs installiert. Die notwendigen Dateien sowie Konfigurationsparameter stellt dabei ebenfalls das ITMC zur Verfügung.

Durch das Laden des Moduls im Apache Webserver erfolgt eine Weiterleitung auf das myITMC. Nach der dort erfolgten Authorisation wird man wieder auf die ursprüngliche Homepage weitergeleitet. Dort stehen jetzt diverse weitere Header-Parameter zur Verfügung, die u.a. in cloudi verwendet werden, aber auch aus Effizienzgründen in der OpenNebula Management Console auf dem PG-Server ls03 zur Authentifizierung verwendet wurden.

### B.2.2 mod\_jk

mod\_jk kann ebenfalls einfach aus dem Repository der verwendeten Distribution oder von der Homepage <http://tomcat.apache.org/connectors-doc/> heruntergeladen und installiert werden. Unter Ubuntu erfolgt mittels

```
sudo apt-get install libapache2-mod-jk
```

Unter Ubuntu ist ergänzend noch der Parameter JKMountCopy On für den virtuellen Standard-Host zu setzen. In der prototypischen Implementierung funktioniert Cloudi nicht mit dem Tomcat-Server von Ubuntu 10.10. Darum ist unter /srv/tomcat ein seperater Tomcat-Server installiert. Dies muss dem mod\_jk-Modul in der Datei /etc/libapache2-mod-jk/workers.properties entsprechend mitgeteilt werden:

```
#
# workers.tomcat_home should point to the location where you
# installed tomcat. This is where you have your conf, webapps and lib
# directories.
#
#workers.tomcat_home=/usr/share/tomcat5
workers.tomcat_home=/srv/tomcat
```

## B.3 Opennebula

Für die Installation von OpenNebula werden unter Ubuntu folgende Pakete benötigt: `libsqlite3-dev`, `libxmlrpc-c3-dev`, `scons`, `g++`, `ruby`, `libssl-dev`, `libxml2-dev`, `libmysqlclient-dev` und `libmysql++-dev` sowie bei Bedarf `sqlite` oder `MySQL`. Detaillierte Informationen über die Installation und Konfiguration von Opennebula, findet man auf der Seite: <http://www.opennebula.org/documentation:rel2.0:ignc> Dabei sind insbesondere folgende Punkte aufgefallen:

1. Auf den Hosts und Opennebula-Frontend müssen Englisch als Sprache eingestellt sein.
2. Auf den Hosts muss die Uhrzeit/Datum richtig eingestellt sein (überall gleich; idealerweise einfach per NTP-Client)
3. Auf dem FrontEnd sollte ein NFS-Server installiert werden.
4. Bei den Knoten/Hosts sollte entsprechend das Paket `autofs` nachinstalliert werden, um dann folgende Schritte folgen zu lassen:
  - (a) Beim Host `„/srv /etc/auto.one“` in der Datei `„/etc/auto.master“` eintragen.
  - (b) Eine Datei `„auto.one“` in der Datei `„/etc/“` erstellen.
  - (c) Dort reinschreiben: `cloud „PGCloudi“<IP des Servers>:/srv/cloud`
  - (d) `srv/cloud „IP des Hosts“/„Subnetzmaske“ (rw,async)` eintragen (im Front-End in der Datei `/etc/exports`)
  - (e) den Befehl `„sudo /etc/init.d/autofs reload“` eingeben, um `autofs` neu zu laden
5. Die Ports zwischen 5000 und 5999 für `vnc`<sup>58</sup> auf den OpenNebula Hosts müssen zur Nutzung der VNC-Applets von den Webclients aus erreichbar sein..
6. Zur Vermeidung von Problemen mit den Dateirechten sind in der Datei `„/etc/profile“` `umask` auf 000 gesetzt.

---

<sup>58</sup><http://www.tightvnc.com>

## B.4 cloudi

cloudi wurde in der Projektgruppe mittels folgendem Skript deployt:

```
#!/bin/sh
export DIR=/tmp/'pwgen';
sudo true;

# Tomcat neu installieren
mkdir $DIR;
curl http://ftp.halifax.rwth-aachen.de/apache/tomcat/tomcat-6/\
v6.0.30/bin/apache-tomcat-6.0.30.tar.gz\
-o $DIR/apache-tomcat-6.0.30.tar.gz
sudo /etc/init.d/tomcat stop;
sudo rm -Rf /srv/apache-tomcat-6.0.30;
sudo tar -xzf $DIR/apache-tomcat-6.0.30.tar.gz --directory=/srv/;
sudo chown -R tomcat.tomcat /srv/apache-tomcat-6.0.30;
sudo /etc/init.d/tomcat start;
rm -Rf $DIR;

# Cloudi deployen
mkdir $DIR;
cd $DIR;
svn checkout https://projekte.itmc.tu-dortmund.de/svn/clauidpg/cloudi/;
cd $DIR/cloudi;
ant -Dj2ee.server.home=/srv/tomcat/\
-Dlibs.CopyLibs.classpath=\
/usr/share/netbeans/java3/ant/extra/\
org-netbeans-modules-java-j2seproject-copylibstask.jar\
-Dlibs.hibernate-persistence.classpath=\
./web/WEB-INF/lib/hibernate-entitymanager.jar:\
./web/WEB-INF/lib/hibernate-commons-annotations.jar:\
./web/WEB-INF/lib/hibernate3.jar:\
./web/WEB-INF/lib/hibernate-annotations.jar;
sudo /bin/cp $DIR/cloudi/dist/cloudi.war /srv/tomcat/webapps;
rm -Rf $DIR;
```

## C **Entwicklerhandbuch**

### **Backend**

Das Backend von Cloudi besteht aus folgenden Teilen, welche unterschiedliche Aufgaben erfüllen und für einen reibungslosen Ablauf an die jeweilige Umgebung einer Installation angepasst werden müssen.

1. META-INF
2. cloudi.conf
3. cloudi.crud
4. cloudi.ganglia
5. cloudi.id.mapping
6. cloudi.interfaces
7. cloudi.logging
8. cloudi.model
9. cloudi.model.monitoring
10. cloudi.monitoring
11. cloudi.service
12. cloudi.simulation
13. cloudi.timer
14. cloudi.prediction

### **META-INF**

Dies ist ein Teil der grundlegenden Konfiguration von Cloudi und beinhaltet die Konfigurationsdatei `persistence.xml`, in welcher der Zugriff auf die Datenbank konfiguriert wird. Zur Lauffähigkeit muss der Abschnitt `<properties>` modifiziert werden. Insbesondere gilt es die Attribute `hibernate.connection.username`, `hibernate.connection.password` sowie `hibernate.connection.url` an die vorhandene Infrastruktur anzupassen.

### **cloudi.conf**

Das Package `cloudi.conf` ist ein weiterer Teil der grundlegenden Konfiguration von Cloudi und beinhaltet die zwei folgenden Java Klassen. Die Klasse `GangliaConf` enthält die Konfigurationsparameter für Ganglia und `rrdtool`. Anzupassen sind hier die Attribute `GRAPH_FOLDER`, welcher den Ordner in dem die Statistikgraphen gespeichert werden angibt, `CLUSTER`, welcher auf den Name des Clusters in Ganglia gesetzt werden muss, sowie `RRD_DIR`, welcher auf den Ordner der rrd Datenbanken verweist. Die Klasse `OneConfig` ist für die OpenNebula RPC Konfiguration zuständig. Die Variablen `ONE_RPC_SERVER`, URL zum OpenNebula RPC Server, `ONE_ADMIN`, Name des OpenNebula Admin Accounts, sowie `ONE_ADMIN_PWD`, Passwort des OpenNebula Admin Accounts, müssen an die jeweilige Installation angepasst werden.

Als Alternative bietet die Konfigurationsdatei `config.ini`, welche über die Klasse `Installation` analysiert wird, die Möglichkeit oben angegebene Attribute außerhalb des Quelltextes zu initialisieren. Dieses Vorgehen wird auch bei der Nutzung mehrerer Instanzen von Cloudi empfohlen.

### **cloudi.crud**

In diesem Package kann, mithilfe der Klasse `CRUDCloudiVM`, ein Objekt der Klasse `CloudiVM` in die Datenbank gespeichert werden, respektive bietet sie die Möglichkeit ein solches Objekt aus der Datenbank zu laden. Hiermit wird ein großer Teil der Persistenz von Cloudi realisiert.

### **cloudi.ganglia**

Unter dem Package `cloudi.ganglia` befinden sich diejenigen Klassen, welche für die Anbindung zwischen Cloudi und Ganglia genutzt werden. Die Namensgebung der jeweiligen Klassen deutet bereits ihre Aufgabe an. `VMCpuAuslesen` dient zum Auslesen und Übertragen der CPU Aktivitäten aus der rrd Datenbank. Analog hierzu existieren die Klassen `VMMemoryAuslesen` und `VMNetzwerkAuslesen` welche zum Auslesen von Speicher und Netzwerk Verbrauch bzw. Aktivität dienen. Zusätzlich liegt eine Hilfsklasse `IPToDNS` vor, welche die Möglichkeit bietet zu einer IP oder VM dynamisch den zugehörigen DNS Hostnamen auszugeben, welcher wiederum für die Anbindung der rrd Datenbank benötigt wird.

### **cloudi.id.mapping**

Dieses Package beinhaltet Klassen, welche per persistenzierter Relationen eine Abbildung von verschiedenen Cloudi Entitäten auf die entsprechenden Middleware Entitäten bieten. Beispielsweise ist über die Klassen `VmID` und `VMMapping` eine Abbildung von Cloudi VMs zu OpenNebula VMs und umgekehrt möglich. Dieses Package ist für die Anbindung von Middleware essentiell und muss bei Verwendung von weiterer Middleware neben OpenNebula erweitert werden. Als Designmuster weiterer Klassen für andere Middleware neben OpenNebula können die bereits implementierten Klassen für OpenNebula dienen.

### **cloudi.interfaces**

Die hier bereitgestellten Schnittstellen wie `HostService`, `VMService` und `UserService` werden über `cloudi.service` implementiert.

### **cloudi.logging**

Dieses Package dient zur Realisierung eines persistenten Ereignisprotokolls. über die Klasse `ActionController` können verschiedene Ereignisse wie `ACT_LOGIN`, `ACT_LOGOUT`, `ACT_VM_START`, `ACT_VM_STOP`, `ACT_VM_PAUSE`, `ACT_VM_RESUME`, `ACT_VM_MIGRATE`, `ACT_VM_DELETE`, `ACT_VM_CREATE`, `ACT_USR_REGISTER`, `ACT_USR_DATA_CHANGED` und `ACT_USR_PASS_CHANGED` in das Ereignisprotokoll gespeichert werden. Ebenso kann das Ereignisprotokoll, unter Verwendung verschiedener Filter, ausgelesen werden.

### **cloudi.model**

Hier sind die verschiedenen Cloudi Entitätstypen wie `CloudiHost`, `CloudiVM`, `CloudiUser`, etc. realisiert. Diese Entitätstypen dienen der jeweiligen Cloudi Entitäten als Grundlage. Diese wiederum stellen persistierbare eindeutige Objekte, welchen die jeweiligen Informationen zugeordnet wurden, dar.

### **cloudi.model.monitoring**

Analog zu `cloudi.model` werden hier die Entitätstypen bezüglich des Monitorings realisiert.

### **cloudi.service**

Unter `cloudi.service` werden die Schnittstellen aus `cloudi.interfaces` implementiert. Es werden die verschiedenen Dienste zu den Cloudi Entitäten

angeboten. Dieses Package ist ebenfalls für die Middleware essentiell und muss bei Verwendung von weiterer Middleware neben OpenNebula erweitert werden.

### **cloudi.simulation**

Das Package `cloudi.simulation` bietet über seine beinhalteten Klassen eine Schnittstelle zu CloudSim. Hierdurch wird eine Unterstützung zur Modellierung und Simulation von Cloud-Umgebungen geboten. Für andere Simulationsumgebungen ist dieses Package Ausgangspunkt für weitere Entwicklung.

### **cloudi.timer**

Bietet eine rudimentäre Möglichkeit Dienste über einen Timer zu starten. Hierbei wird die Ressourcenverwaltung der Middleware überlassen.

### **cloudi.prediction**

Dieses Package dient zur Berechnung zukünftigen Hardwarebedarfs. über die Klasse `LinearRegression` wird die Möglichkeit geboten aus gegebenen Verbrauchs-/Quelldaten der Klasse `DatenSource` Vorhersagen der Klasse `VorhersageDaten` über den zukünftigen Bedarf zu berechnen. Hierzu wird die Methode der linearen Regression genutzt. Für die Anwendung weiterer, angepasster Berechnungen des zukünftigen Bedarfs ist `cloudi.prediction` der passende Ort für weitere Implementierungen.

## Abbildungsverzeichnis

1	Die Projektgruppe . . . . .	1
2	Die Architektur von Cloudi . . . . .	12
3	Vergleich zwischen OpenNebula und Eucalyptus . . . . .	21
4	OpenNebula Architektur . . . . .	22
5	Servergrundgerüst . . . . .	25
6	Stand-Alone Simulationen oder Scheduling mit Haizea . . . . .	28
7	Cloudi aus der Server-Ansicht . . . . .	31
8	Ganglia Implementation [7] . . . . .	35
9	Die Architektur von Ganglia [7] . . . . .	36
10	Demografik aus RRDtool . . . . .	38
11	Graph für CPU-Auslastung der letzten Stunde . . . . .	39
12	Graph für die Netzwerkstatistiken . . . . .	40
13	Graph für die Speicherbenutzung . . . . .	40
14	Graph für die CPU Auslastung aus ZK . . . . .	43
15	Graph für die Arbeitsspeicher Benutzung aus ZK . . . . .	44
16	Graph für die Netzwerkstatistiken aus ZK . . . . .	44
17	CloudSim Klassendiagramm für Kernkomponenten . . . . .	46
18	CloudSim-Architektur . . . . .	48
19	Methode der kleinsten Quadrate . . . . .	50
20	Standardlayout einer Cloudi Webseite . . . . .	54
21	myITMC Seite mit dem Hyperlink auf das Cloudi-Portal . . . . .	56
22	Cloudi Startseite . . . . .	57
23	Menüleiste . . . . .	57
24	Kontextmenü der Startseite . . . . .	58
25	Anmeldeseite von Cloudi . . . . .	59
26	Registrierungsformular von Cloudi . . . . .	59
27	Benutzerdaten . . . . .	60
28	Erster Menüpunkt . . . . .	60
29	Zweiter Menüpunkt . . . . .	61
30	Virtuelle Maschine migrieren . . . . .	62
31	Dritter Menüpunkt . . . . .	62
32	VM erstellen einfach . . . . .	63
33	Step 1/5 Namensvergabe . . . . .	63
34	Step 2/5 Hardwarekonfiguration . . . . .	64
35	Step 3/5 Netzwerkkonfiguration . . . . .	64
36	Step 4/5 Zeitraum festlegen . . . . .	65
37	Step 5/5 Bestätigung . . . . .	65
38	Vierter Menüpunkt . . . . .	66
39	Startseite von Cloudi als Administrator . . . . .	66

---

40	Administratoren Menüleiste . . . . .	66
41	Benutzerverwaltung . . . . .	67
42	Menü für die Benutzerverwaltung . . . . .	68
43	Menü für die Hostverwaltung . . . . .	68
44	Menü für die Vorhersage . . . . .	68
45	Eingabe-Menü . . . . .	69
46	Tabelle-IstDaten . . . . .	69
47	Tabelle-PlanDaten . . . . .	70
48	Graphische Auswertung der Entwicklung von Anzahl VMs . . . . .	71
49	Graphische Auswertung der Entwicklung von vCPU . . . . .	71
50	Graphische Auswertung der Entwicklung von vMEM . . . . .	72
51	Graphische Auswertung der Entwicklung von Storage . . . . .	72
52	Menü für die Simulation . . . . .	73
53	Simulation-Untermenü . . . . .	74
54	Simulation-Ergebnisse . . . . .	75

# Index

- LaTeX, 9
- AJAX, 5
- Apache HTTP Server, 15
- Apache Tomcat, 17
- Backend, 5–6
- Benutzermanagement, 15
- Cloud Computing, 6–7
- cloudi
  - .conf, 14
  - .ganglia, 13
  - .id.mapping, 14
  - .interfaces, 13
  - .logging, 14
  - .models, 13
  - .monitoring, 13
  - .prediction, 14
  - .simulation, 14
- CloudSim, 14, 45–50
- CMS, 5
- Commons Mathematics, 14
- Deployment, 16
- dojotoolkit, 5
- Eclipse, 9, 13
- EJB, 5–6
- Electronic Workspace, 8
- Entwicklung, 9
- Eucalyptus, 19–21
- Frontend, 5
- Ganglia, 14, 15, 32–39
- gmetric, 36
- gmond, 34
- Google Web Toolkit, 5
- Haizea, 16, 27–30, 46
- http, 5
- IDE, 9
- ITMC, 12
- Java EE, 5
- JavaEE, 6
- JCA, 5–6
- JMS, 24
- JNDI, 5–6
- JPA, 5–6
- JSP, 5–6
- JTA, 5–6
- JVM, 33
- Kick-Off, 4
- Konfiguration, 14
- KVM, 17–19
- libvirt, 26, 32
- Logging, 14
- mod\_jk, 17
- Monitoring, 15–16, 32–43
- MS-SQL, 5–6
- myITMC, 15
- MySQL, 5–6, 16
- Netbeans, 9
- OCCI, 15, 24
- OpenNebula, 9, 13, 14, 16, 19–27, 32
- OpenSSO, 15
- Oracle, 5–6
- PHP, 5–6, 33, 38–39
- Projektgruppe, 2–10
- Projektinfrastruktur, 4–5
- Projektmanagement, 4
- QEMU, 17

Qualitätsmanagement, 5

Redmine, 8–9

RRDtool, 16, 33, 37–38

Scheduling, 16

Sigar, 32

Simulation, 7

Systemvoraussetzung, 16

Virtualisierung, 6

VMware, 17

VNC, 52

webserver, 15

XDR, 33, 34, 36, 37

Xen, 17–19

XML, 5, 33, 35, 37

XML-RPC, 15, 24

    Port, 14

ZK-Framework, 13, 42, 49

## Literatur

- [1] DIN 69901-5: Projektmanagement – Projektmanagementsysteme – Teil 5: Begriffe, Januar 2009.
- [2] Bundesministerium des Inneren. Verordnung zur Schaffung barrierefreier Informationstechnik nach dem Behindertengleichstellungsgesetz (Barrierefreie Informationstechnik-Verordnung – BITV). In Bundesministerium der Justiz, editor, *Bundesgesetzblatt Teil 1*, number 49, pages 2654–2662. Bundesanzeiger Verlag, Bonn, Juli 2002.  
[http://www.bit.bund.de/cIn\\_092/nn\\_374978/BIT/DE/Shared/Publikationen/BGG\\_BITV,templateId=raw,property=publicationFile.pdf/BGG\\_BITV.pdf](http://www.bit.bund.de/cIn_092/nn_374978/BIT/DE/Shared/Publikationen/BGG_BITV,templateId=raw,property=publicationFile.pdf/BGG_BITV.pdf).
- [3] Robert Warnke Thomas Ritzau. *gemu-kvm & libvirt*. Books on Demand GmbH, Norderstedt, Norderstedt, 4th edition, 2010.
- [4] Damien Cerbelaud, Shishir Garg, and Jeremy Huylebroeck. Opening the clouds: qualitative overview of the state-of-the-art open source vm-based cloud management platforms. In *Middleware '09: Proceedings of the 10th ACM/IFIP/USENIX International Conference on Middleware*, pages 1–8, New York, NY, USA, 2009. Springer-Verlag New York, Inc.
- [5] Borja Sotomayor, Rubén S. Montero, Ignacio M. Llorente, and Ian Foster. Virtual infrastructure management in private and hybrid clouds. *IEEE Internet Computing*, 13(5):14–22, 2009.
- [6] Borja Sotomayor. *The HAIZEA Manual*, September 2009.  
[http://haizea.cs.uchicago.edu/haizea\\_manual.pdf](http://haizea.cs.uchicago.edu/haizea_manual.pdf).
- [7] Matthew L. Massie, Brent N. Chun, and David E. Culler. The ganglia distributed monitoring system: design, implementation, and experience. *Parallel Computing*, 30(7):817 – 840, 2004.
- [8] Rodrigo N. Calheiros, Rajiv Ranjan, César A. F. De Rose, and Rajkumar Buyya. Cloudsim: A novel framework for modeling and simulation of cloud computing infrastructures and services. *CoRR*, abs/0903.2525, 2009. informal publication.