

OpenNebula

The Open Source Solution for Data Center Virtualization

Jürgen Hock

Hochschule Mannheim
Institut für Softwaretechnik
und Datenkommunikation

21. Juni 2012

Abstract: OpenNebula ist eine quelloffene Werkzeugsammlung für Cloud Computing. Es ist zum Management von virtuellen Infrastrukturen gedacht. Als IaaS-Cloud-Lösung kann OpenNebula eingesetzt werden, um private, öffentliche oder hybride Clouds zu erstellen. In diesem Beitrag sind die wesentlichen Charakteristiken von OpenNebula beschrieben.

1 Einleitung

Trotz der verschiedenen Interpretationen des Begriffs *Cloud Computing* sind einige gemeinsame Anhaltspunkte auch in den folgenden Definitionen enthalten:

Karlsruhe Institute of Technology (KIT)

„By using virtualized computing and storage resources and modern Web technologies, cloud computing provides scalable, network-centric, abstracted IT infrastructures, platforms, and applications as on-demand services. These are billed on a usage basis.“ [BKNT11]

National Institute of Standards and Technology (NIST)

„Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. This cloud model is composed of five essential characteristics, three service models, and four deployment models.“ [PG11]

Es geht also um bestimmte Ressourcen, die bei Bedarf mit minimalem Aufwand über das Netzwerk unbegrenzt bereitgestellt werden können. Diese Idee kann demnach auf Ressourcen von Datenspeicher bis hin zu Softwarediensten übertragen werden.

Denkbar ist auch eine Bereitstellung von kompletten IT-Infrastrukturen, sog. *Infrastructure as a Service* (IaaS), die ihrerseits aus einem virtualisierten Rechnerpark bestehen. Dies ist auch der nächste Entwicklungsschritt in der Konsolidierung von Rechenzentren, siehe Abbildung 1.

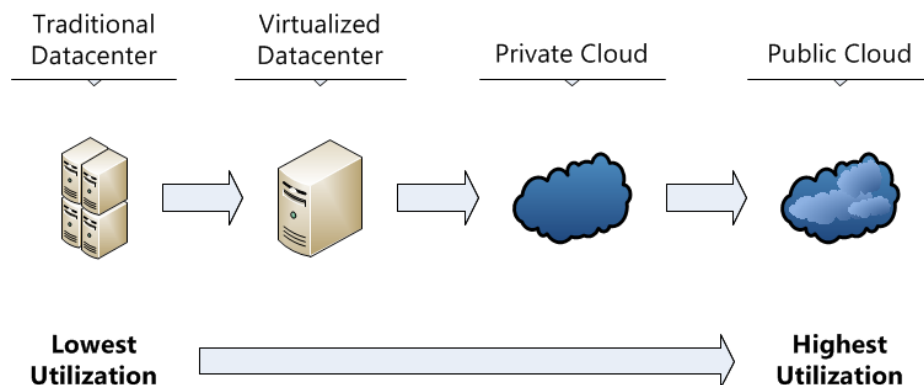


Abbildung 1. Hardwarekonsolidierung in Rechenzentren [Tul10]

OpenNebula bietet eine mögliche Lösung zur Realisierung von IaaS-Diensten. Dazu gehört unter anderem eine automatisierte Verwaltung von virtuellen Maschinen, die auf bereits erprobten Techniken und gängigen Softwarebibliotheken basiert.

In folgenden Kapiteln werden nur die ausgewählten Aspekte des OpenNebula-Systems erwähnt, die ausführlichen Informationen sind unter [Ope12] und [C1212] erhältlich.

2 OpenNebula

OpenNebula ist ein Open-Source-Projekt, welches im Jahre 2005 von *Distributed Systems Architecture Research Group*¹ initiiert wurde. Daraus hat sich eine Community² entwickelt, die den Quellcode des Systems sowie zugehörige Installationsanleitungen öffentlich bereitstellt. Entwickler des OpenNebula gründeten außerdem das Unternehmen *C12G Labs*³ und bieten auch kommerzielles Support⁴ für gewerbliche Kunden an.

OpenNebula basiert auf Bibliotheken und Tools des Linux Betriebssystems. Des- sen Quellcode unterliegt der *Apache License Version 2* Lizenzbedingungen. Es kann entweder als ein TAR-Paket heruntergeladen und manuell kompiliert oder, sofern vorhanden, aus dem Softwarerepositorium des Betriebssystems bereits vorkompiliert installiert werden.

¹ <http://dsa-research.org> – Distributed Systems Architecture Research Group

² <http://opennebula.org> – OpenNebula Community

³ <http://c12g.com> – C12G Labs

⁴ <http://opennebula.pro> – OpenNebulaPro Support Portal

OpenNebula selbst bezeichnet sich als ein „Open-Source-Industriestandard zur Virtualisierung von Rechenzentren“. Man verspricht eine umfangreiche Lösung zur totalen Verwaltung von virtualisierten Rechenzentren. OpenNebula hat das Ziel, IaaS-Dienste in eine vorhandene IT-Infrastruktur zu integrieren. Bereits getätigte Investitionen sollten erhalten bleiben und das Vendor-Lock-In vermieden werden.[C1212]

Es handelt sich um einer reinen IaaS-Lösung. Zu den Kernfunktionen von OpenNebula gehören eine einfache und automatisierte Verwaltung von:

- Virtuellen Maschinen,
- dafür benötigten Datenspeicher und
- virtuellen Netzwerkschnittstellen.

Neben diesen Kernfunktionen stellt OpenNebula ebenso bereit:

- Verwaltung von Benutzerkonten,
- Schnittstellen zur Fernverwaltung des Systems,
- Möglichkeiten zur weiterführenden Systemausbau
- sowie Skalierbarkeit zum beliebigen Zeitpunkt.

3 Grundlegende Systemaufbau und Anforderungen

OpenNebula baut sich auf drei Komponenten auf:

- Front-End,
- Hosts und
- Datastore.

Front-End, das zentrale Element des Systems, übernimmt die Verwaltung von Hosts und Organisation von Datastores. Auf den Host-Rechnern wird die Hypervisor-Software ausgeführt und virtuelle Maschinen gehostet. Ein Datastore ist eine Netzwerkdateifreigabe oder ein Netzwerkspeicherdienst zur Bereitstellung von Abbildern für die virtuellen Maschinen. Mehrere Datastores können auch auf mehreren Rechnern verteilt werden. Ein typischer Systemaufbau ist in der Abbildung 3 dargestellt.

Beim Front-End werden keine besondere Anforderungen an Hardware gestellt. Falls weitere Systemkomponenten im eigenen Subnetz befinden sollten, ist eine zweite Netzwerkkarte notwendig. Auf dem Front-End können auch Netzwerkdienste wie DHCP installiert werden. Jedoch müssen IP-Adressen für die virtuellen Maschinen fest vergeben werden, da OpenNebula keine dynamische IP-Vergabe unterstützt.

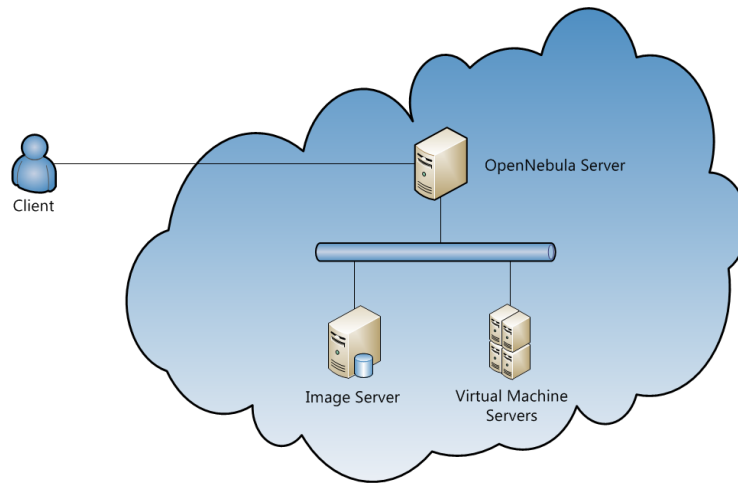


Abbildung 2. Grundlegende Systemaufbau von OpenNebula

Hardwareausstattung der Nodes richtet sich an den jeweiligen Hypervisor. In der Regel wird verhältnismäßig mehr RAM-Kapazität als CPU-Leistung und Prozessoren mit Intel-VT bzw. AMD-V Erweiterungen benötigt.

Die Ausstattung der Datastores ist lösungsabhängig. Je nach Verteilungsstrategie von Festplattenabbildern müssen Hosts außerdem über ausreichend lokalen Datenspeicher verfügen.

Installiert wird OpenNebula auf einem der unterstützten Linux Betriebssystem: Debian/Ubuntu, CentOS/RHEL oder openSUSE. Für Hosts und Datastores gelten hier lösungsspezifische Anforderungen.

4 Benutzerschnittstellen

Benutzer greifen in der Regel auf die Schnittstellen des Front-Ends zu um z.B. virtuelle Maschinen hochzufahren. Der Zugriff auf diese erfolgt direkt ohne den Umweg über Front-End. Hierfür stellt OpenNebula eine VNC-Schnittstelle. Weitere Zugriffsmechanismen wie SSH, RDP oder HTTP sind ebenfalls möglich, müssen aber von der jeweiligen virtuellen Maschine bereitgestellt werden. Im Falle, dass Hosts sich im eigenen Subnetz befinden, wird möglicherweise ein VPN-Tunnel benötigt.

Opennebula stellt eine Reihe von Bedienungsschnittstellen für den Benutzer zur Verfügung:

- Webfrontends „Sunstone“ und „Self-Service“,

- CLI-Tools
- XML-RPC API.

Mittels Sunstone können die meisten Verwaltungsaufgaben erledigt werden, z.B. Verwaltung von Hosts, virtuellen Maschinen, Netzwerkverbindungen, Abbildern sowie Systembenutzern. Self-Service ist eine stark vereinfachte Version von Sunstone.

Sollte in Sunstone eine Funktion nicht zur Verfügung stehen oder Verwaltungsaufgaben automatisiert werden, so können hierfür die CLI-Tools in Betracht gezogen werden. Diese befinden sich nach der Installation auf dem Front-End und beginnen mit dem Präfix „one“.

Dem Webinterface und CLI-Tools liegt die XML-RPC API zugrunde, d.h. das System kann jederzeit fernverwaltet und Aufgaben entsprechend automatisiert werden.

Noch vor der Installation von OpenNebula muss das Systembenutzerkonto/-gruppe angelegt werden, z.B. „oneadmin:onecloud“. Dasselbe Konto ist auch auf jedem Host anzulegen, bestenfalls mit derselben ID.

5 Gastbetriebssysteme und Kontextualisierung

Hypervisors unterstützen in der Regel sowohl Linux als auch Windows Betriebssysteme. Im Falle von KVM ist der Support von paravirtualisierten SCSI-Festplattentreibern für Windows eher mangelhaft. Zwar ist es im SWT⁵ gelungen unterschiedliche Windows Versionen (XP, 7 und 2008) mit unterschiedlichen Treibern zu installieren, jedoch erfolgten die Lese-/Schreibzugriffe auf die virtuelle Festplatte recht langsam.

Zur Kontextualisierung von generischen Festplattenabbildern generiert OpenNebula ein virtuelles ISO-Abbild mit kontextabhängigen Informationen, z.B. zugewiesener IP-Adresse. Fernen können auch benutzerspezifische Einstellungen auf diesem Wege übermittelt werden. Auf dem generischen Festplattenabbild muss ein Startskript vorinstalliert werden, welcher beim Hochfahren des virtuellen Rechners nach dem ISO-Laufwerk sucht und all diese Einstellungen umsetzt. Gewöhnlich werden unter Linux Betriebssystemen solche Startskripte in der Datei „rc.local“ und unter Windows mittels Gruppenrichtlinien festgelegt.

6 Systemkern

Der Systemkern von OpenNebula besteht aus zwei Diensten: OpenNebula Daemon und Scheduler. OpenNebula Daemon ist für die zentrale Verwaltung zuständig. Die

⁵ Institut für Softwaretechnik und Datenkommunikation

Aufgabe des Schedulers ist virtuelle Maschinen auf physische Knoten zu verteilen. Über die XML-RPC Schnittstelle kann der Systemkern angesprochen und erweitert werden. Der Zugang zur physischen Knoten erfolgt über jeweilige Treiber. Der Zusammenhang zwischen einzelnen Komponenten ist in der Abbildung 6 dargestellt.

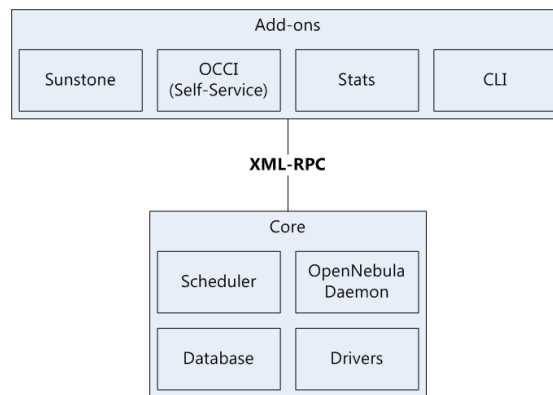


Abbildung 3. OpenNebula Systemkern

Zur Erfüllung von verschiedenen Verwaltungsaufgaben werden in OpenNebula spezifische Treiber eingesetzt. Beispielsweise stehen Treiber zur Überwachung diverser Hypervisors oder zur Verteilung von Festplattenabbildern über diverse Netzwerkprotokolle zur Verfügung. Außerdem basiert die Verwaltung von virtuellen Maschinen auf der *libvirt* Bibliothek, die wiederum diverse Hypervisors unterstützt.

Ein wichtiger Bestandteil des Systems ist der Scheduler-Dienst. Zur Verteilung von virtuellen Maschinen auf Knoten-Rechner werden diese benötigt. Die Benotung erfolgt je nach Wunsch der jeweiligen virtuellen Maschine. So können diverse Verteilungsstrategien durchgeführt werden, z.B. am wenigsten belastete Knoten zuerst.

Des Weiteren kann das System mit Skripten erweitert werden, die bei bestimmten Ereignissen ausgeführt werden. Beispielsweise können virtuelle Maschinen bei Ausfall eines Nodes auf einen anderen umgezogen werden. Solche ereignisgesteuerte Aufrufe von Skripten werden von OpenNebula als *Hooks* bezeichnet.

7 Fazit

OpenNebula bietet eine solide Basis zur Bereitstellung von IaaS-Diensten. Die Entwicklung befindet sich bereits im fortgeschrittenen Stadium und das System macht

einen guten und stabilen Eindruck. Vor allem wird es auch bei Großkunden wie z.B. CERN, ESA eingesetzt⁶.

Dass OpenNebula den Cloud Computing Kriterien entspricht, zeigt sich vor allem durch die flexible Erweiterbarkeit des Systems durch unproblematisches Hinzufügen von weiteren Hypervisor-Knoten und Möglichkeiten zum Reagieren auf dessen Ausfällen.

Das Vendor-Lock-In wird aus der Sicht eines OpenNebula-Endbenutzers zwar nicht vermieden, jedoch ist das OpenNebula intern weitgehend unabhängig von den jeweils eingesetzten Virtualisierungslösungen.

Durch die bereitgestellte API kann OpenNebula als Grundfundament für PaaS und SaaS eingesetzt werden.

Literatur

- [BKNT11] BAUN, Christian ; KUNZE, Marcel ; NIMIS, Jens ; TAI, Stefan: *Cloud Computing: Web-Based Dynamic IT Services*. Springer, 2011
- [C1212] C12G LABS: *OpenNebula Guides*. 2012
- [Hoc11] HOCK, Jürgen: *Cloud Computing für Anwendungen in der Telekommunikation*. T-Systems Darmstadt / Hochschule Mannheim, 2011
- [Ope12] *OpenNebula: The Open Source Solution for Data Center Virtualization*. 2012. – <http://openebula.org>
- [PG11] PETER, Mell ; GRANCE, Timothy: *The NIST Definition of Cloud Computing*. NIST, 2011
- [Tul10] TULLOCH, Mitch: *Understanding Microsoft Virtualization Solutions, From the Desktop to the Datacenter, 2nd Edition*. Redmont, Washington : Microsoft Press, 2010

⁶ <http://openebula.org/start>