
CLOUD-COMPUTING

SYSTEMTECHNIK



24. FEBRUAR 2016
BURKHARD HAMPL & SIMON WORTH

Inhalt

Grundlagen	3
Was ist eine Cloud?	3
Gemeinsame Benutzung der Ressourcen	3
Einfache Verwaltung	3
On demand	3
Ebenen.....	4
Hardware.....	4
Infrastruktur	4
Plattform.....	4
Anwendung	4
Client.....	5
Abgrenzung zu anderen Technologien.....	5
Grid-Computing	5
On premise.....	5
Offenheit der Plattformen	5
Public Clouds	5
Private Cloud.....	5
Hybrid Cloud	6
Zusammenfassung in Tabelle.....	7
Ansätze des Cloud-Computings.....	7
Infrastructure as a Service.....	7
Platform as a Service (PaaS)	8
Software as a Service (SaaS).....	8
Vergleich der Anwendungen und Marktführer.....	9
Weitere Ansätze	9
Virtualisierung	10
Arten der Virtualisierung	11
Server-Virtualisierung	11
Desktop-Virtualisierung	11
Applikations-Virtualisierung	11
Storage-Virtualisierung	11
Netzwerk-Virtualisierung.....	11
Hypervisor	11
Virtualisierungstechniken	12
Xen.....	12

Hyper-V	12
VMWare	12
KVM (Kernel-based Virtual Machine).....	13
Virtualisierung vs. Container	13
Selbstverwaltung und Management von Cloud-Computing Umgebungen	14
System Center	14
VMWare vCloud Director.....	14
VMWare vCloud Datacenter Services	15
Verteilte Betriebssysteme	16
Grundlagen.....	16
Multiprozessor Architektur	16
Multicomputer Architektur.....	16
Kernel.....	16
Multiprozessor Betriebssysteme.....	16
Multicomputer Betriebssysteme.....	17
Multiprozessor Betriebssysteme	17
Multiprozessor Hardware und Software Modelle.....	17
Memory Management	19
Process control	19
Multicomputer Betriebssysteme	21
Cluster Betriebssysteme.....	21
Verteilte und Parallele Funktionen	22
Remote Procedure Call (RPC)	22
Message-passing interface (MPI)	22
Cloud-Computing und Verteilte Betriebssysteme	24
Conclusio	24
Abbildungsverzeichnis.....	25
Literaturverzeichnis	26

Grundlagen

Was ist eine Cloud?

Das Wort Cloud kommt aus dem Englischen und bedeutet Wolke, der Grund für den Namen ist, dass Dienste, die ins Internet ausgelagert werden, auf Skizzen meistens als Wolke gekennzeichnet werden. Diese Wolke soll also das Internet symbolisieren. [2, 3]

Heutzutage setzen viele Hersteller auf Cloud-Computing und bieten Cloud-Plattformen an. Sogar IT-Unternehmen die primär für Software bekannt sind, bieten zunehmend Cloud-Dienste und -Plattformen an.

Konkret geht es bei Cloud-Computing um die Auslagerung von Anwendungen, Daten und Rechengvorgängen ins Web. Dies könnte zum Beispiel die Auslagerung von Bürosoftware wie Tabellenkalkulation oder Textverarbeitung oder CRM-System in die Cloud sein.

Diese Auslagerung bietet einige Vorteile. Die Synchronisation zwischen mehreren Rechnern wird unnötig und gemeinsame Arbeit an Dokumenten durch die zentrale Ablage vereinfacht. So sind Unternehmen auch in der Lage Dokumente freizugeben, wie zum Beispiel Vorlagen für Rechnungen oder Ähnliches. Natürlich sind auch die Datenspeicher in der "Wolke" begrenzt, sie bieten jedoch für fast jeden Anwendungsfall ausreichend Speicherplatz an.

Bei Rechengvorgängen ist der Vorteil, dass man auf einen großen Pool an Virtuellen Maschinen zurückgreifen kann. Dank Cloud-Computing kann man nun diese Server für die wenigen Stunden, in denen sie benötigt werden, mieten und somit Budget einsparen.

In einem Satz bedeutet Cloud-Computing also: [2, 3]

"Cloud-Computing ist die Auslagerung von Anwendungen, Daten und Rechengvorgängen in das Internet" [2]

Gemeinsame Benutzung der Ressourcen

Eine Anwendung verwendet gemeinsame Ressourcen, die sich verschiedenen Benutzer teilen. Das bedeutet, dass die Daten eines Benutzers in einer von mehreren Nutzern verwendeten Tabelle liegen. Für das Absichern der Datensätze ist die Cloud-Computing-Plattform und die Anwendung selbst verantwortlich. Ressourcen werden aber auch physisch verwendet. Eine Anwendung in der Cloud kann sich einen physikalisch vorhandenen Rechner mit einer anderen Anwendung teilen. Das funktioniert durch Virtualisierung (wird in einem Späteren Abschnitt noch erklärt). [2, 3]

Einfache Verwaltung

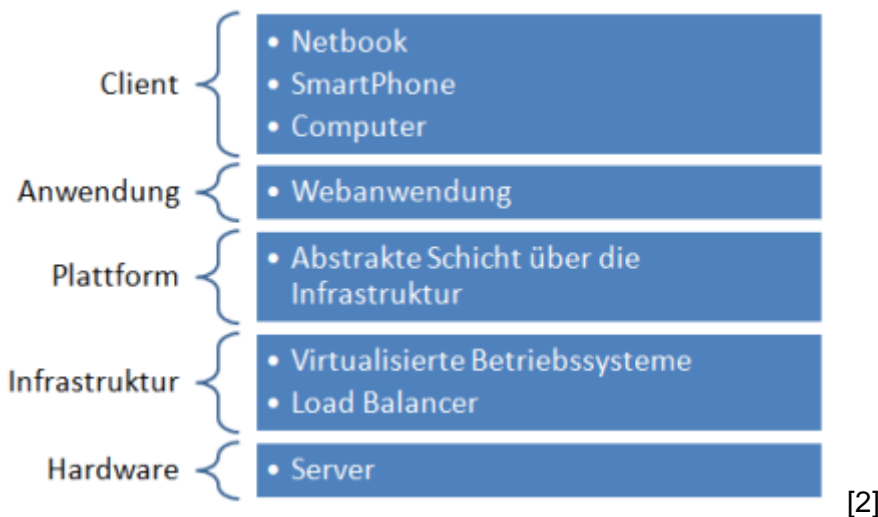
Verglichen mit traditionellen Systemumgebungen, sind Cloud-Computing-Plattformen wesentlich einfacher zu verwalten. Der Grund dafür ist der hohe Abstraktionsgrad der Plattformen, denn um typische Administrationsaufgaben wie Load Balancing oder Serverwartung kümmert sich bereits der Anbieter. Die meisten Plattformen bieten auch eine wesentlich vereinfachte Administrationsoberfläche an. [2]

On demand

Cloud-Computing-Plattformen sind on demand verfügbar. Das bedeutet, dass man keine komplexen Verträge abschließen muss, um die zu benutzen. Im Normalfall kann man einen

Cloud-Computing-Plattform bereits nach wenigen Minuten benutzen. Bei normalen On-premise-Hostings dauert dies, anhängig vom Unternehmen, einige Tage bis Wochen. Der Cloud-Computing-Anbieter Amazon rechnet beispielsweise über seine vorhandenen Konten für den Onlineshop ab. Hat man also schon einmal bei Amazon eingekauft, dann kann man den die Cloud-Computing-Plattform bereits verwenden. Benötigt man zusätzlichen Speicher, besteht die Möglichkeit, dass man diesen innerhalb von wenigen Minuten zur Verfügung hat. [2]

Ebenen



Hardware

Die Serverhardware, auf der normalerweise ein für die Virtualisierung optimiertes Betriebssystem (z.B.: Linux oder Windows Server) installiert ist. Auf dieser Ebene findest du noch keine Verteilung für späteres Cloud-Computing statt. [2, 3]

Infrastruktur

Hier kommen auch Load Balancer zum Einsatz, die Lasten auf verschiedene Server verteilen. Betriebssysteme werden hier nur mehr virtuell zur Verfügung gestellt. Das heißt, dass man auf dieser Ebene nicht mehr weiß, auf welchem Betriebssystem der Server läuft. Diese Ebene wird auch als Infrastructure as a Service (IaaS) bezeichnet. [2, 3]

Plattform

Hier hat man keinen Zugriff mehr auf das virtuelle Betriebssystem. Diese Schicht dient primär der Anwendungsentwicklung. Hier können auch APIs zur Verfügung gestellt werden. Die Plattformschicht verteilt die Lasten je nach Belastungsspitzen auf den verschiedenen virtuellen Betriebssystemen. Diese Schicht wird als Plattform as a Service (PaaS) bezeichnet. [2, 3]

Anwendung

Hier wird eine Anwendung für den Endbenutzer bereitgestellt. Anwendungen, die auf dieser Schicht laufen, benutzen meistens die APIs der Plattformschicht. Diese Schicht wird als Software as a Service (SaaS) bezeichnet. [2, 3]

Client

Hierbei handelt es sich um Geräte und Anwendungen, die Cloud-Computing-Dienste beanspruchen. Dies können verschiedene Geräte wie Smartphones oder Notebooks sein. Der Unterschied zur Anwendungs-Ebene ist, dass diese Anwendungen Cloud-Computing-Dienste verwenden. Diese Ebene wird auch als Software plus Service bezeichnet. [2, 3]

Abgrenzung zu anderen Technologien

Grid-Computing

Cloud-Computing ist Grid-Computing im Aufbau sehr ähnlich, jedoch unterscheiden sie sich doch in gewissen Punkten. Historisch gesehen ist Cloud-Computing aus dem Grid-Computing entstanden. Grid Computing wird aber wesentlich häufiger in wissenschaftlichen Bereichen eingesetzt. Der größte Unterschied ist, dass Cloud-Computing-Dienste von Anbietern für eine große Anzahl an Kunden bereitgestellt werden, während Grid-Computing meistens für sehr spezielle Anwendungsfälle im Unternehmen selbst aufgebaut wird. [2]

On premise

Unter "on premise" versteht man den klassischen Ansatz des Hostings. Hier werden alle Daten auf lokalen Servern im Unternehmen gehostet. On premise ist also das exakte Gegenteil von Cloud-Computing. Daher wird On-premise-Hosting oft mit Cloud Computing verglichen.

Auch bei den Kosten unterscheiden sich die beiden Technologien. Während bei On-premise Lösungen für die Hardware und Software-Lizenzen gezahlt wird, werden bei Cloud-Computing die Kosten anhand des tatsächlichen Verbrauchs berechnet. [2]

Offenheit der Plattformen

Neben den Ebenen des Cloud-Computing wird auch bei der Offenheit der Plattformen unterschieden. [2, 3]

Public Clouds

Die Public Cloud ist der am häufigsten verwendete Bereich, welcher meistens gemeint ist, wenn von Cloud-Computing gesprochen wird.

Ein Plattformanbieter stellt seine Dienste mehreren Kunden zur Verfügung, das bedeutet, dass die Ressourcen mit mehreren Kunden geteilt werden. Damit ist gemeint, dass Datenspeicher oder Maschinenstunden am selben physikalischen Server geschehen können. Die größten Public-Cloud-Anbieter sind Google, Amazon und Microsoft. [2, 3]

Private Cloud

Wie der Name schon sagt, kann auf Private Clouds nicht so einfach zugegriffen werden, wie das bei Public Clouds der Fall ist.

Private Clouds befinden sich innerhalb eines Unternehmens. Das hat den Vorteil, dass Datensicherheit und Privatsphäre einfacher zu kontrollieren sind. Der Unterschied zu On-premise-Lösungen ist, dass private Clouds Schnittstellen nach außen haben. Das bedeutet, dass die Clouds auch von Dritten verwendet werden können, was im Normalfall auf gewisse

Personen oder Gruppen eingeschränkt wird. Zum Beispiel Kunden oder Lieferanten, die gewisse Tools für die gemeinsame Arbeit mit dem Unternehmen benötigen.

Ein weiterer Vorteil ist, dass die Private Cloud ideal an die Unternehmensstrategie angepasst werden kann. On-premise-Hosting und Private Clouds sind im Prinzip das gleiche, jedoch werden beide Begriffe in der Literatur und im Internet häufig gebraucht.

Der Nachteil von privaten Clouds ist, dass hier sich wieder das Unternehmen um die Bereitstellung von Hard- und Software kümmern muss. Außerdem fallen wesentlich mehr Personalkosten, da die Systeme gewartet werden müssen. [2, 3]

Hybrid Cloud

Die Hybrid Cloud ist eine Mischform zwischen Public und Private Cloud. Hier wird versucht, die Vorteile beider Techniken zu vereinen und die Nachteile zu minimieren.

Mithilfe von Hybrid Clouds kann man unternehmensinterne Lösungen und Angebote mittels einer Middleware für außenstehende Verfügbar machen. Will ein Unternehmen zum Beispiel auch einen Service zusätzlich für ihre Kunden anbieten, kann das zum einen über eine Private Cloud erfolgen, wobei die Serviceendpunkte dem Kunden zur Verfügung gestellt werden. Genau hier würde eine Hybrid Cloud ansetzen und zwar an der Bereitstellung der Services. Hier wird der Service dem Kunden nicht direkt zur Verfügung gestellt, sondern über einen Cloud-Service, was den Vorteil hat, dass Kunden nicht auf unternehmensinterne Daten zugreifen können.

Der Cloud-Service bietet nun eine Schnittstelle zwischen Unternehmen und Kunden.

Ein Beispiel für eine Hybride-Cloud-Anwendung ist Windows Azure AppFabric. [2, 3]

Zusammenfassung in Tabelle

Art	Art der Ressourcenteilung	Hoster	Kosten
Private Cloud	Keine Ressourcenteilung	Unternehmensintern	Einmalig (Software und Hardware) und monatlich (Personal, Bandbreite)
Hybrid Cloud	Ressourcenteilung möglich	Unternehmensintern und durch Cloud-Plattform-Anbieter	Einmalig wie bei der Private Cloud und monatlich für die Cloud-Computing-Umgebung
Public Cloud	Ressourcenteilung	Cloud-Plattform-Anbieter	Monatlich

[2]

Ansätze des Cloud-Computings



Abbildung 1: Ansätze des Cloud-Computings [11]

Infrastructure as a Service

Infrastructure as a Service stellt die unterste Schicht des Cloud-Computings dar. Hierbei handelt es sich um die Infrastruktur, die vom Cloud-Computing-Anbieter zur Verfügung gestellt wird. Den Benutzer des Dienstes steht es nun frei Software auf diese Infrastruktur zu spielen. Im Normalfall stellt der Anbieter hier ein virtualisiertes Betriebssystem zur Verfügung. Häufig kommen als Betriebssysteme Linux und Windows Server zum Einsatz. Der Anbieter kümmert sich auch um die Lastenverteilung und darum, dass die gekauften Leistungen ständig verfügbar sind. Auch das Backup der Systeme wird in den meisten Fällen vom IaaS-Anbieter übernommen. [2, 3]

Welche Aufgaben eine Infrastruktur hier erledigt, kann sehr vielfältig sein. Da der Benutzer Zugriff auf das Betriebssystem hat, kann er jegliche Software installieren. Mögliche Anwendungsfälle sind Domänencontroller, SQL Server, Internetserver, Mailserver und vieles mehr. Will man Webanwendungen auf der Infrastruktur erstellen, muss man sich um die Installation der notwendigen Laufzeitumgebung wie etwas PHP, Java oder .NET kümmern. Der Administrationsaufwand ist gegenüber SaaS und PaaS deutlich höher, dafür bietet IaaS die höchste Flexibilität. Anwendungen auf dieser Ebene können sowohl an Kunden weiterverkauft, als auch für unternehmensinterne Zwecke genutzt werden. Für die Wartung der virtualisierten Umgebung ist der Kunde jedoch selbst verantwortlich. [2, 3]

Vorreiter für IaaS-Dienstleistungen ist Amazon mit Amazon Elastic Compute Cloud. [2, 3]

Platform as a Service (PaaS)

Im Gegensatz zu IaaS-Lösungen bieten PaaS-Lösungen keinen Zugriff auf die darunter liegende Betriebssysteme, aber es bietet eine Sandbox mit High-Level APIs zur Anwendungsentwicklung.

Einer der Vorteile ist, dass sich hier nicht mehr um Softwarelizenzen, Installationen und Wartungen gekümmert werden muss, dadurch lassen sich einfach eigene Anwendungen entwickeln, ohne Rücksicht auf Hardware- und Softwareabhängigkeiten nehmen zu müssen. [2, 3]

In den meisten Fällen, bieten PaaS-Plattformen verschiedene APIs für Management der Plattformen, Diagnostik und Datenspeicherung. Platform-as-a-Service-Plattformen haben oft die Aufgabe die Erstellung und Bereitstellung von Webanwendungen zu ermöglichen. Es werden jedoch auch häufig andere Anwendungstypen wie Worker Roles oder Tasks bereitgestellt. Tasks dienen der Abarbeitung von Aufgaben, die in einem separaten Thread oder sogar auf einer eigenen VM ausgeführt werden. Das Ziel von Tasks ist es Webanwendungen von komplexen Aufgaben oder aufwändigeren Berechnungen zu befreien. Dies wiederum erhöht die Leistungsfähigkeit der Webanwendung. Eine Worker Role erfüllt einen ähnlichen Bereich, läuft aber im Normalfall ständig, während ein Task nur gestartet wird, wenn er benötigt wird. [2, 3]

Mit Hilfe der APIs lassen sich aber auch zum Beispiel Datenspeicher-Operationen ausführen. Dafür bieten viele Plattformen Queues, Tabellen und Blobs an. Meistens kommen hier keine relationalen Datenbanken zum Einsatz. [2, 3]

Die Zielgruppe von PaaS-Plattformen sind primär Softwareentwickler.

PaaS-Plattformen werden unter anderem von Microsoft mit Windows Azure, von Google mit der App Engine und von Salesforce gemeinsam mit VMware als VMwareforce angeboten. [2, 3]

Software as a Service (SaaS)

Hierbei wird eine Software für den Endanwender im World Wide Web angeboten. Dabei kann es sich um verschiedene Typen von Software handeln. Als Beispiel können hier Office-Lösungen und E-Mail Seiten genannt werden.

Der Vorteil davon ist, dass im Gegensatz zu lokal installierten Anwendungen, SaaS-Anwendungen lediglich am Webserver aktualisiert werden müssen. Außerdem hat der Anwender immer sofort die neuste Version zur Verfügung, was wiederum den Vorteil mit sich bringt, dass Sicherheitslücken schneller geschlossen werden können. Und Dokumente auf einem System zu vergessen gehört somit der Vergangenheit an, was leider wiederum auch die Abhängigkeit vom Internet verstärkt.

Die Anwendungen bieten jedoch meist keine Anpassungsmöglichkeiten und werden vom Hersteller als fertiges Produkt verkauft. [2, 3]

Die Zielgruppe von SaaS-Lösungen sind also somit Endanwender. [2, 3]

Vergleich der Anwendungen und Marktführer

	IaaS	PaaS	SaaS
Abstraktionsgrad	Sehr niedrig	mittel	Sehr hoch
Verwaltungsaufwand	Hoch	Mittel	Niedrig
Anpassbarkeit	Sehr hoch	Hoch	Sehr gering
Zielgruppe	Systemhäuser IT-Dienstleister IT-Abteilung Softwareentwickler	Softwareentwickler	Endanwender
Bezahlung	Pay per Use	Pay per Use	Pay per Use

[2]

Als Marktführer im Cloud-Computing-Bereich gelten Google (Google App Engine, Google Compute Engine, Container Engine, ...), Amazon (Amazon Web Services), Microsoft (Microsoft Azure) und IBM (IBM Cloud Services).

Alle Unternehmen bieten sowohl IaaS- als auch PaaS-Lösungen an.

Weitere Ansätze

Software plus Service

Im Fall von Software plus Service werden bestehende oder neue Anwendungen lediglich um Services aus der Cloud erweitert. (z.B.: Speicher- und Synchronisierungsdienste)

Der Benutzer der Anwendung hat hier also die Möglichkeit, seine Daten auf dem Gerät oder in der Cloud zu speichern. Idealerweise gilt die Cloud hier als Backup.

Als konkretes Beispiel kann Microsoft Word genannt werden, da es auch die Möglichkeit bietet, die Datei direkt in OneDrive zu speichern.

Der Hintergrundgedanke von Software plus Service ist, dass einige Anwendungen auch verschiedene Eigenschaften wie Offlinefähigkeit, gutes User Interface oder viele Systembestandteile nutzen müssen. [2]

Dieser Ansatz wird hauptsächlich von Microsoft gefördert. Er wird aber auch gerne auf mobilen Geräten verwendet. [2]

Everything as a Service (EaaS)

Mit Everything as a Service ist gemeint, dass sich Software mehr und mehr zu einem Service entwickelt. Anstatt Lizenzgebühren zu verkaufen, werden Abonnements verkauft. Das bietet wesentlich mehr Möglichkeiten auf der Angebots- als auch auf der Nachfrageseite. Man zahlt also nur für das, was man auch tatsächlich benutzt. [2]

Human as a Service (HuaaS)

Hier ist der Hintergedanke, dass Computer nicht alle Aufgaben erledigen können. Zum Beispiel bei der Bilderkennung ist der Mensch dem Computer weit überlegen. Weiters werden durch Menschen sehr viele Informationen generiert, wie zum Beispiel auf Facebook oder YouTube. Ein Anwendungsfall für HuaaS ist Amazon Mechanical Turk. Hier wird ein Webservice zur Verfügung gestellt, der die Integration von menschlichen Wissen in die eigene Anwendung erlaubt. [2]

Virtualisierung

Eine wichtige Technik für Cloud-Computing ist die Virtualisierung. Hier wird ein Betriebssystem auf einen anderen laufen gelassen. Durch Virtualisierung, täuscht man dem virtualisierten Betriebssystem vor, dass es auf einer realen Hardware läuft. [2, 3]

Mithilfe von Virtualisierung können Server auf denen komplexere Anwendungen laufen, kopiert und auf beliebig viele Rechner verteilt werden, ohne diese neu konfigurieren zu müssen. Sind Updates notwendig oder wird eine virtuelle Maschine nicht mehr benötigt, dann können sie einfach überspielt oder sogar gelöscht werden. Dadurch laufen auf einem Server mehrere Gastbetriebssysteme die aber keine Auswirkung auf das native Betriebssystem haben und sich so nicht gegenseitig mit Prozessen behindern können. [2, 3]

Benötigt eine VM mehr Speicherplatz oder mehr Rechenleistung, kann dies vom nativen Betriebssystem zugewiesen werden. Manche Virtualisierungslösungen erlauben es sogar, dass eine virtuelle Maschine im Betrieb auf einen anderen Server verschoben werden kann. Dadurch kann stets die ideale Auslastung und Performance erzielt werden. Auch der Ausfall eines Servers stellt keine große Herausforderung mehr da, da das virtuelle Betriebssystem auf ein anderes System verschoben wird. Der Endanwender bekommt den Ausfall im normal fall nicht mit. [2, 3]

Virtualisierung bringt jedoch unter Umständen auch Performanceprobleme mit sich, da jede zusätzliche "Schicht" mehr Aufwand bedeutet. [2, 3]

Arten der Virtualisierung

Server-Virtualisierung

Server-Virtualisierung ist die am meisten eingesetzte Technologie und geschieht über einen Hypervisor, welcher den zentralen Teil einer Lösung in der Cloud darstellt. Er ermöglicht es, dass mehrere verschiedene Betriebssysteme zur gleichen Zeit auf einem Prozessor lauffähig sind. Es laufen also mehrere virtuelle Server auf nur einen tatsächlichen Server. [7]

Desktop-Virtualisierung

Neben Servern können auch Desktopsysteme virtualisiert werden. Das Desktopsystem wird im Rechenzentrum auf einem Server installiert und der Anwender greift über ein remote auf den virtuellen Desktop zu. Ziel ist es, eine ähnliche Performance wie bei der direkten Nutzung eines Desktopsystems zu erreichen. [8]

Applikations-Virtualisierung

Nicht immer ist der Zugriff per Remotedesktop sinnvoll oder erwünscht.

Bei dieser Art der Virtualisierung wird eine bestimmte Applikation so zur Verfügung gestellt, dass auf dem Desktop des Benutzers keine Applikation installiert werden muss. [8]

Storage-Virtualisierung

Durch die Virtualisierung der Storage wird es möglich, virtuelle Daten und Maschinen von einem Speicherort unabhängig zu sichern und somit die Cloud vollständig zu nutzen. Der Endanwender hat demnach immer den gleichen Pfad zu seinen virtuellen Daten und wo die Daten tatsächlich gespeichert werden ist für den Benutzer uninteressant [7]

Netzwerk-Virtualisierung

Hier werden Netzwerk-Ressourcen zu logischen Einheiten zusammengefasst oder aufgeteilt. Dabei bekommt das lokale Netz eine virtuelle Schicht. Dadurch wird die Komplexität des Netzes verschleiert und Administratoren können das Netzwerk einfacher überwachen. Und wenn also Netzwerkkarten durch virtuelle Lösungen ersetzt werden können weitere Kosten vermieden werden. [7]

Hypervisor

Der Hypervisor ermöglicht es, dass mehrere verschiedene Betriebssysteme zur gleichen Zeit auf einem Prozessor lauffähig sind. Der Hypervisor wird auch als Virtual Machine Monitor (VMM) bezeichnet. Er steuert die Zuteilung der Ressourcen zu den einzelnen virtuellen Maschinen. Er basiert direkt auf einem bestehenden Betriebssystem (Typ 2 Hypervisor) oder aber auf der Hardware (Typ 1 Hypervisor). [7, 9]

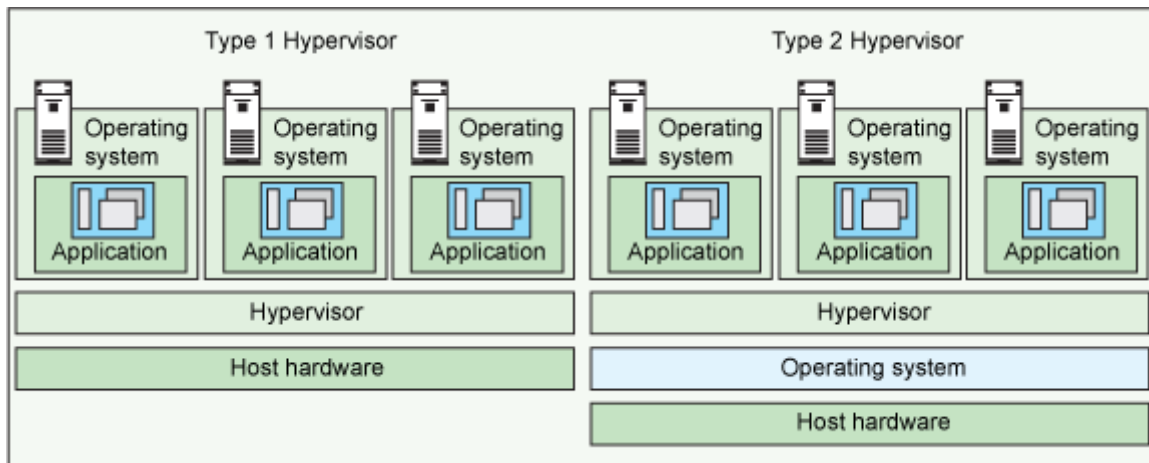


Abbildung 2: Virtualisierung [9]

Es wird von Para-Virtualisierung gesprochen, wenn der Hypervisor direkten Zugriff auf die Hardware hat und somit Befehle des virtuellen Betriebssystems direkt weitergeben kann (Hier weiß der Kernel sozusagen, dass er nur virtuell ist). Wenn dies nicht der Fall handelt es sich um Voll-Virtualisierung. Hier kann der Hypervisor die Befehle nur an das darunterliegende Betriebssystem weitergeben und dieses kümmert sich an um die Umsetzung auf Hardwareebene.

Virtualisierungstechniken

Xen

Xen erfreut sich vor allem in der Open-Source-Welt an großer Beliebtheit. Doch auch viele Unternehmen, wie zum Beispiel Amazon, greifen auf XEN zurück.

Xen ist auf den meisten Prozessorarchitekturen lauffähig und unterstützt die meisten Betriebssystem bis auf Mac-OS.

Der Hypervisor befindet sich direkt an der Hardware und stellt alle Anfragen der einzelnen Betriebssysteme an die Hardware durch. [2]

Hyper-V

Bei dieser Virtualisierungsplattform gibt es ein Hauptbetriebssystem, das die Virtualisierung vornimmt. Dieses Hauptbetriebssystem wird bei Microsoft als "Parent System" und alle darauf aufbauende Systeme als "Child System" bezeichnet. Hyper-V benötigt Windows Server als "Parent System". Als "Child System" können aber durchaus andere Betriebssysteme eingesetzt werden, wie zum Beispiel Linux. [2]

VMWare

VMWare bietet Sowohl Produkte für Typ 1 als auch Typ 2 Hypervisor an. VMWare Workstation ist das Produkt für Desktop-Virtualisierung und VMWare Server hat seinen Einsatz auf der Serverseite. Es ist unter anderem für Linux, Windows und Solaris geeignet. [2]

VMWare bietet auch zusätzlich eine Lösung für Para-Virtualisierung an. Dieses Produkt nennt sich VMware ESXi. [13]

KVM (Kernel-based Virtual Machine)

KVM ist ein relativ junges Virtualisierungstool das mittlerweile von Red Hat übernommen wurde und welches ebenfalls Open-Source ist. KVM benötigt als Grundlage ein Linuxsystem auf welches eine Vielzahl von Gastsystemen spielen lässt, wie etwa Windows, Linux oder CentOS. [2]

Virtualisierung vs. Container

Wie vorher schon erwähnt, hat Virtualisierung durch die zusätzlich benötigte Schicht eine geringere Performance. Zusätzlich entsteht eine Redundanz, da jede virtuelle Maschine einen eigenen Kernel benötigt. [10]

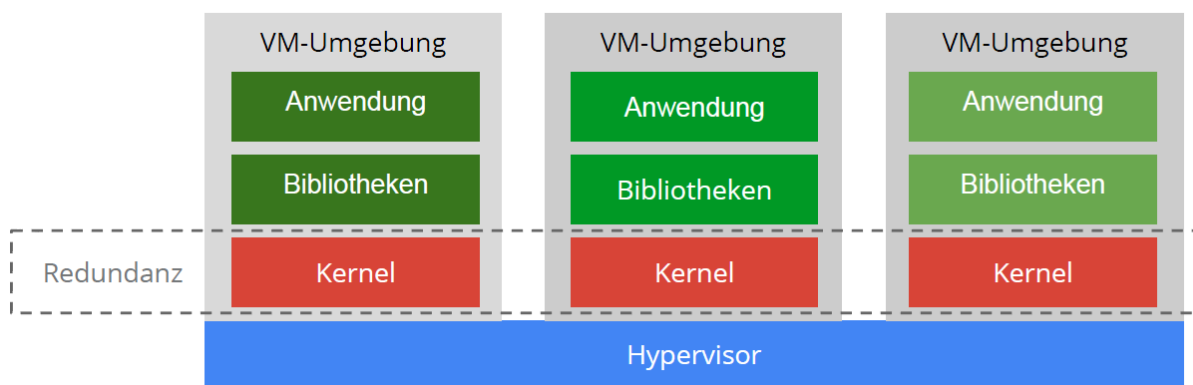


Abbildung 3: Virtualisierung vs. Container [10]

Genau für dieses Problem gibt es Container. Denn diese Technologie abstrahiert die Umgebung besser und ist ebenfalls Hochskalierbar. [10]

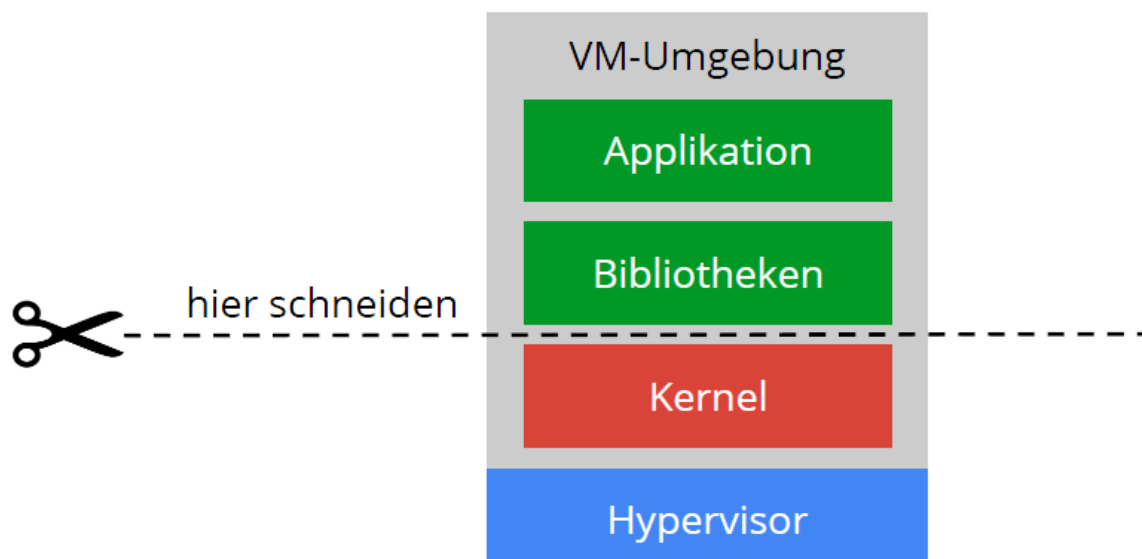


Abbildung 4: VM-Umgebung [10]

Man schneidet also die Virtuelle Umgebung zwischen dem Kernel und den Bibliotheken ab. Somit benutzen alle Container den gleichen Kernel. Dies wiederum hat den Nachteil, dass alle Anwendungen auch auf dem gleichen Kernel laufen müssen. [10]

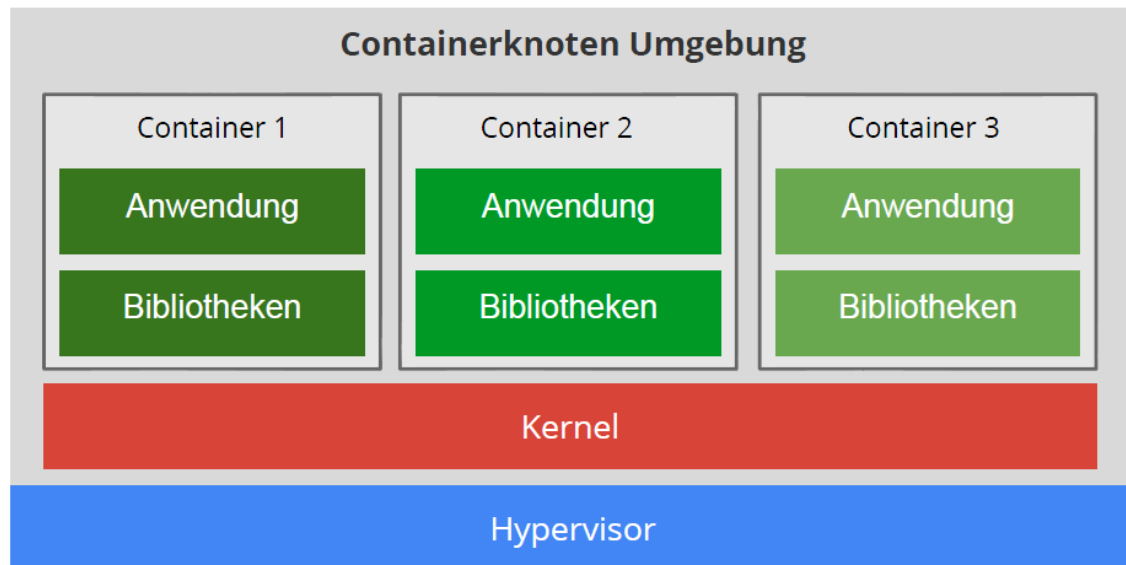


Abbildung 5: Containerknoten Umgebung [10]

Bei Google laufen alle Dienste in Containern. Weiters startet Google pro Woche 2 Milliarden Container auf ihren Servern. [10]

Selbstverwaltung und Management von Cloud-Computing Umgebungen

Als Benutzer einer Cloud-Computing-Plattform muss man sich nicht um die Verwaltung oder das Management (Zum Beispiel die Bereitstellung und Verteilung einzelner Instanzen für einen Kunden) der einzelnen Systeme kümmern. Doch Anbieter wie Amazon, Google oder Microsoft müssen sich damit herumschlagen. Dazu gibt es einige Produkte am Markt die versuchen dieses Problem zu lösen. Im Idealfall verwalten sich die Rechenzentren sogar selbst. Dafür gibt es Systeme, die das Management für private Clouds vereinfachen. [2]

System Center

System Center sind eine Verwaltungssoftware für mehrere Instanzen, die von Microsoft angeboten wird. Es gibt verschiedene Versionen davon, für Cloud-Computing ist jedoch der System Center Operation Manager am wichtigsten. Diese Software überwacht Server und Dienste und kann bei Bedarf die Systeme auch skalieren. [2]

VMWare vCloud Director

Mit VMware vCloud Director kann man virtuelle Instanzen zusammenfassen und eine private Cloud-Computing-Lösung erstellen. Es wird auch die logische Aufteilung der Ressourcen unter den vorhandenen Instanzen. Die Private Cloud kann auch über ein Webinterface

überwacht werden. Im Hintergrund verwendet VMWare vCloud Director eine auf REST basierende API. [2]

VMWare vCloud Datacenter Services

Diese Version soll die benötigte Kompatibilität, Sicherheit und Kontrolle von Unternehmen gewährleisten. Man kann sowohl interne und externe IT-Ressourcen verwalten. Das bedeutet, dass VMWare vCloud Datacenter Services den Ansatz einer Hybrid-Cloud unterstützt. [2]

Verteilte Betriebssysteme

Grundlagen

Ein verteiltes System besteht aus einer Auswahl von Computern die mittels Software und Hardware miteinander kommunizieren. Generell wird zwischen zwei verschiedenen Arten von verteilten Systemen unterschieden: [6] [5]

Multiprozessor Architektur

Bei der Multiprozessor Architektur sind zwei oder mehr Microprozessor/CPU's voll miteinander, mittels eines Busses oder Switches, verbunden und greifen auf den gleichen Arbeitsspeicher zu. Dabei hat jeder/jede Microprozessor/CPU gleichen Zugriff auf den ganzen physikalischen Arbeitsspeicher. Diese Art der Architektur findet man in manchen Supercomputern. [6] [12]

Multicomputer Architektur

Dabei sind mehrere unabhängige Computer physikalisch miteinander verbunden und dynamisch mit Software gekoppelt, um ein Computer-Netzwerk aufzubauen. Beispiele für so eine Architektur sind LANs und Computer-Cluster in einem Bürogebäude.

[6]

Kernel

Software für verteilte Systeme benötigt einen Kernel der die gleiche Rolle wie bei Einzelprozessor- oder zentrierten Systemen spielt. Der Kernel, welches das verteilte Betriebssystem definiert, verwaltet die verteilten Ressourcen, übernimmt das Prozess-Scheduling und koordiniert die Kommunikation zwischen den verschiedenen Teilen eines verteilten Systems.

[6]

Es gibt zwei Arten eines solchen Betriebssystems:

Multiprozessor Betriebssysteme

Multiprozessor Betriebssysteme bestehen meistens aus einem "normalen" Betriebssystem. Doch besitzen diese einzigartige Features, wie Prozess-Synchronisierung, Ressourcen-Management und Scheduling.

[6] [12]

Multicomputer Betriebssysteme

Die drei wichtigsten Arten von Multicomputer Betriebssystemen sind Cluster Betriebssysteme, Netzwerk Betriebssysteme und Parallele Betriebssysteme.

[6]

Multiprozessor Betriebssysteme

Multiprozessor Hardware und Software Modelle

Es gibt zwei große Kategorien an Multiprozessoren: Shared-memory und No Remote Memory Access (NoRMA) Multiprozessoren:

Shared-memory Multiprozessoren

Bei Shared-memory Multiprozessoren wird der ganze Haupt-Memory von allen Prozessoren geteilt.

[6] [14]

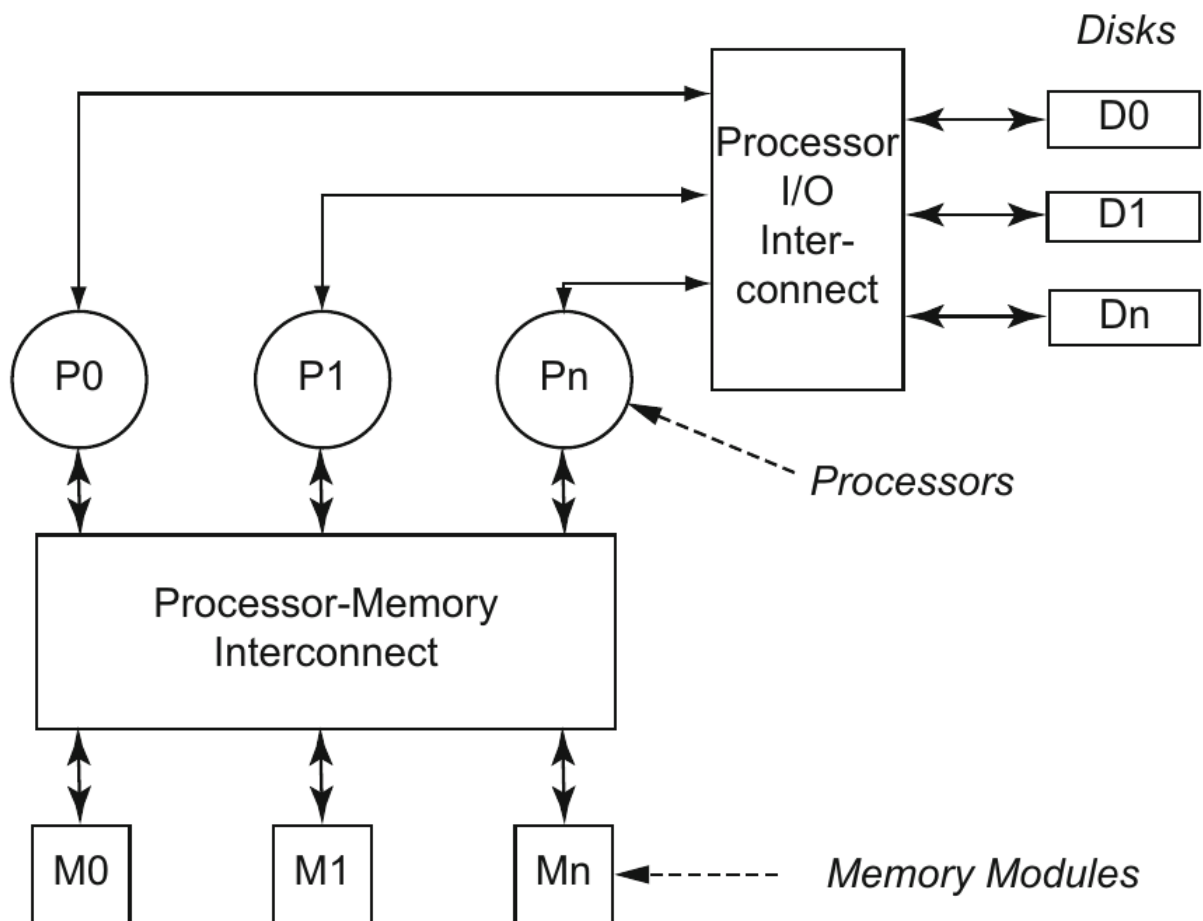


Abbildung 6: Shared-memory Multiprozessoren [6](S.2)

Dabei gibt es mehrere Klassen:

Uniform Memory Access (UMA)

Bei Uniform Memory Access (UMA) Multiprozessoren ist die Zugriffszeit auf den Shared-memory für alle Prozessoren gleich. Die einfachste Architektur ist die Single-Bus (Siehe Grafik "Drei Bus basierende Multiprozessoren" (a)), dabei greifen zwei oder mehr CPUs auf ein oder mehr Memory-Modul über einen Bus zu. In der Grafik "Drei Bus basierende Multiprozessoren" (b), haben jede CPU noch einen zusätzlichen Cache und in Grafik "Drei Bus basierende Multiprozessoren" (c), ist noch zusätzlich eine lokaler und privater Memory, welcher über einen eignen (private) Bus angesprochen wird.

[6] [14]

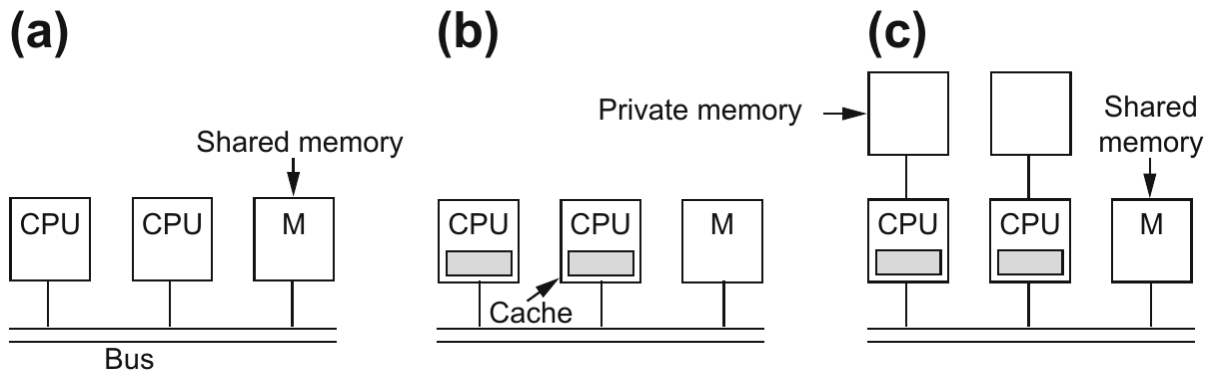


Abbildung 7: Uniform Memory Access [6](S.3)

Drei Bus basierende Multiprozessoren: (a) ohne caching; (b) mit caching; (c) mit caching und privaten Memory [6](S.3)

Non-Uniform Memory Access (NUMA)

Bei Non-Uniform Memory Access Multiprozessoren wird der physikalische Memory in module partitioniert, jeder dieser Teile ist lokal und einem Prozessor zugeordnet. Dadurch wird die Zugriffszeit von lokalem Memory im Gegensatz zu nicht lokalem verkürzt, dadurch wird diese Art von Speicher sehr gerne als Cache eingesetzt, wenn auch nur in sehr kleinen Maßen.

[6] [14]

UMA Architekturen sind die am gebräuchlichsten bei Parallelen Macheinen, doch löst die NUMA Architektur Probleme bei der Skalierbarkeit.

[6] [14]

No Remote Memory Access (NoRMA) Multiprozessoren

In dieser Klasse hat jeder Prozessor sein eigenes lokales Memory, welches nicht mit anderen Prozessoren im System geteilt wird. Durch das simple Design, ist NoRMA derzeit die Wahl bei Supercomputern.

[6]

Memory Management

Shared virtual Memory

Memory-Management Services und Implementationen sind generell Betriebssystem und Maschinen-Architektur abhängig. Doch wurde das Virtual Memory Management Maschinen und Betriebssystem unabhängig gestaltet. Alle benötigten Daten, um das Management von Virtual Memory zu erledigen, wird als Maschinen unabhängige Datenstruktur abgespeichert. Ähnlich wurde ein skalierbare, Kernel unabhängig und generisches Memory-Management Interface geschaffen, welches sich für verschiedene Architekturen eignet.

[6]

NUMA und NORMA Memory Management

Für NUMA Multiprozessoren gibt es eine Anzahl von Memory-Management-Algorithmen:

Migration Algorithmus

Wenn ein Prozessor auf die Daten zugreift, werden diese auf den lokalen Memory migriert. Der Nachteil dabei ist, dass nur ein Task auf einem Prozessor effizient auf die Daten zugreifen kann.

[6] [15]

Read-Replication Algorithmus

Replikation verringert die durchschnittlichen Lese-Kosten, denn durch die Replikation können Lesezugriffe simultan lokal auf mehreren Prozessoren ausgeführt werden. Doch werden wenige Lese-Zugriffe kostspielig, weil die Replikation die Konsistenz wahren muss.

[6] [15]

Full-Replication Algorithmus

Voll-Replikation erlaubt das Datenblöcke auch beim Schreiben repliziert werden, sodass die Daten Konsistent bleiben.

[6] [15]

Page placement

Dabei werden Virtual Memory Pages dem zugehörigen lokalen physikalischen Memory Zugewiesen. Eine einfache Strategie ist, dass man den lokalen Memory als eine Art Cache für den Verteilten Memory verwendet. Dynamisches Page placement, unterstützt sowohl das Migrieren als auch Replizieren.

[6]

Process control

Wenn ein Prozess einen System-Call macht, der auf einige kritische Kernel-Tabelle zugreifen muss, muss der Kernel temporär Interrupts ausschaltet, bevor auf die Tabelle zugegriffen wird. Danach können dann die benötigten Schritte abgearbeitet werden, ohne unterbrochen zu werden. Das Problem Multiprozessoren ist nun, dass das ausschalten der Interrupts nur auf einen Kernel auswirkt. Andere CPUs können immer noch die kritischen Tabellen angreifen.

Um diese Problem zu lösen, werden verschiedene Synchronisations-Konstrukte eingesetzt, um den Zugriff für alle anderen Prozessoren zu unterbinden.

[6]

Locks

Spin- und Blockin-Locks

Bei einem Spin-Lock, prüft der wartende Prozess dauernd (busy-waits) ob die Sperre noch da ist. Bei einem Blockin-Lock wird so lange blockiert, bis die Sperre gelöst wird.

[6]

Read-Write Locks

Es gibt einen Read- oder mehrere Write-Locks. Read-Locks können von mehreren Leser Threads gehalten werden, solange keiner schreiben will, Write-Locks sind exklusiv und können nur von einem gehalten werden. Diese Locks werden mittels Spin- oder Blockin-Locks umgesetzt.

[6]

Konfigurierbare Locks

Damit können Anwendungen unter anderem dynamisch Warte-Algorithmen (Spin, Block oder beide) verändern und dadurch kann eine höhere Performance erreicht werden.

[6]

Berrier Locks

Berrier Locks werden in der Parallelen Programmierung eingesetzt. Welche sicher stellen das kein Prozess über einen bestimmten Punkt im Programm läuft. Denn erst wenn alle anderen Prozesse, die mit diesem Prozess kooperieren, auch an diesem Punkt angelangt sind, dürfen sie weiterlaufen. Diese Locks werden auch mittels Spin- oder Blockin-Locks umgesetzt.

[6]

Andere Synchronisations-Konstrukte

Test-und-set Instruktionen

Bei Test-und-Set (test-and-set) Instruktionen wird zuerst getestet, z.B. ob ein gewisser Wert einem andren gleicht, und dann je nach Ergebnis dieser Test(s) wird dann in das Memory geschrieben. Das Besondere dabei ist, das die ganze Operation Atomic ist, also kann nicht von einem anderen Prozess unterbrochen werden.

[6]

Bedingungs-Variablen

Mit Bedingungs-Variablen (Condition variables) ist es möglich, einen Task (oder Thread) auszusetzen, während er auf eine Aktion eines anderen Task wartet. Bedingungs-Variablen sind einige geteilte Variablen.

[6]

Event Brokers

Events sind Hauptsächlich für die Task Ordnung da. Ein Prozess wartet auf ein Event, bis zum Auftreten des Events blockiert er. Nachdem ein Signal aufgetreten ist, wird einer oder alle wartende Prozesse aufgeweckt. Kompliziertere Konstrukte ermöglichen es, auf

bestimmte Events und Zustände zu reagieren, da es vorkommen kann, dass zwar ein Event auftritt, aber der gewünschte Zustand nicht erreicht ist.

[6]

Interprozess-Kommunikation

Kooperierende Prozesse oder Tasks kommunizieren oft miteinander, um bei der Synchronisation und Nebenläufigkeit zu helfen. Beim Austausch von Nachrichten muss auf die Kommunikation, aufs Buffering und auf die Synchronisation geachtet werden. Allgemein ist das Austauschen von Nachrichten viel abstrakter als das Zugreifen auf verteilte Memory.

Wegen verschiedensten Faktoren und Probleme bei IPC unterstützen die meisten verteilten Betriebssysteme Remote Procedure Calls (RPC) als eine Art der Interprozess-Kommunikation. RPC ist eine höhere Abstraktion als IPC und bietet eine weit effizientere und sichere Kommunikation. Und es macht keinen Unterschied, ob mit einem lokalen oder entfernten Prozess kommuniziert wird.

[6]

Multicomputer Betriebssysteme

Multicomputer Systeme sind entweder ein Netzwerk (lose gekoppelte Computer) oder ein Cluster (eng gekoppelte Computer).

Cluster sind eine Gruppe physikalisch verbundener Computer, welche so nah miteinander zusammenarbeiten, dass sie in vielen Hinsichten als ein einziger Computer betrachtet werden können. Ein Cluster besteht aus mehreren alleinstehenden Computern, welche mit lokalem high-speed Netzwerk miteinander verbunden sind.

[6]

Cluster Betriebssysteme

In einem verteilten Betriebssystem ist es zwingend ein einziges System-Image zu haben, damit es gegenüber dem User, als ein einziger Computer aufzutreten. Weiteres ist es nötig, dass es Fehlertoleranz und Error Recovery Features implementiert, damit, wenn ein Node ausfällt oder vom Cluster entfernt wird, der Rest der Prozesse nicht beeinträchtigt werden. Das Cluster-Betriebssystem sollte skalierbar sein und das System hochverfügbar machen.

Das System-Image sollte einen einzigsten Entrypoint, Dateihierarchie, Kontrollpunkt, Virtual-Networking, Memory-Space, Job-Management-System und User-Interface haben. Für die Hochverfügbarkeit ist zusätzlich noch ein einzigster I/O Space, Prozess-Space und Prozess-Migration zu implementieren.

[6] [16]

Solaris MC

Solaris MC ist ein Prototyp von einem Betriebssystem für Cluster, welches auf Solaris aufbaut. Die meisten Funktionen in Solaris MC sind ladbare Module, welche Solaris erweitern, weiteres wurden nur wenige Veränderungen am existierenden Kernel getätigt, deswegen können schon existierende Programme und Device-Treiber unverändert auf dem Cluster laufen.

Solaris MC ist hochverfügbar entworfen, deswegen läuft auf jedem Node ein separater Kernel, damit wird ein Node, der ausgefallen ist, automatisch erkannt und die System-

Services werden so konfiguriert, dass die übrigen Nodes verwenden, so dass nur die Programme, welche die Ressourcen des ausgefallenen Nodes verwendet haben, davon beeinträchtigt sind.

[6] [16]

MOSIX

MOSIX ist ein Software-Paket, welches den Linux-Kernel mit Cluster Funktionalitäten erweitert. Nachdem ein Prozess gestartet wird, wird er auf das System zugewiesen, welches, zu diesem Zeitpunkt, am besten verfügbar ist. MOSIX überwacht alle Prozesse und verschiebt automatisch welche, wenn der Cluster unausgeglichen ist.

Eine Anwendung braucht keinerlei Anpassungen, um auf MOSIX zu laufen, nicht einmal spezielle Bibliotheken werden benötigt. Man muss nicht einmal den Node angeben, auf welche die Anwendung laufen soll, MOSIX macht das alles automatisch und transparent.

MOSIX besteht aus drei Kern-Algorithmen: Load-Balancing, Memory-Ushering und Ressourcen-Management, welche dazu da sind, um die Performance aller laufenden Prozesse zu verbessern. Die Algorithmen verwenden Prozessmigration, um Prozesse zwischen Nodes zuzuweisen und neu zuzuweisen, damit der Cluster das volle Potential der verfügbaren Ressourcen ausnutzen kann. Der dynamische Load-Balancing Algorithmus gewährleistet, dass die Auslastung gleichmäßig über den Cluster verteilt ist. Der Memory-Ushering Algorithmus schützt vor "Hard-Disk Swapping", indem er Prozesse auf Nodes migriert, welche noch genug RAM zur Verfügung haben. Die MOSIX Ressourcen-Management Algorithmen sind dezentralisiert, das heißt, jeder Node ist beides, ein Master der lokalen Prozesse und ein Server für remote (migrierte) Prozesse. Das hat den Vorteil, dass wenn ein Node hinzugefügt wird, die anderen Nodes nur minimal beeinflusst werden.

MOSIX basiert auf dem Home-None-Model, dabei geht es darum, alle Prozesse scheinen auf dem Login-Node des Users zu laufen. Dabei werden alle Prozesse auf dem Home Node gestartet und Prozesse, welche wegmigriert worden sind, sprechen mit dem Home Node. MOSIX unterstützt Cluster-Konfigurationen von wenigen Computern bis viele mit einem minimalen Overhead.

[6] [5] [16]

Verteilte und Parallele Funktionen

Remote Procedure Call (RPC)

Remote procedure call wird in verteilten Betriebssystemen eingesetzt, um Prozeduren auf anderen Maschinen über das Netzwerk aufzurufen.

[6] [5]

Message-passing interface (MPI)

Das Message-passing interface ist ein standardisiertes Interface für den Nachrichtenaustausch von Programmen auf verteilten Computern.

MPI ist nicht dazu entworfen, um Compiler zu parallelisieren, noch hat es keinen Support für Fehlertoleranz und es unterstützt Multitasking nicht explizit.

[6] [5]

MPI Konzepte

Prozess Gruppen

Eine Prozess Gruppe ist eine sortierte Kollektion von Prozessen, in der jeder Prozess eindeutig, mit dessen Rang in der Ordnung, identifizierbar ist. In einer Gruppe mit n Prozessen die Ränge gehen von 0 bis $n-1$. Gruppen werden zum Trennen von verschiedenen Aufgaben.

[6]

Kommunikations-Kontexte

Kommunikations-Kontexte wurden ursprünglich dazu geplant, um es zu ermöglichen, mehrere Nachrichten Streams zwischen Prozessen zu erlauben, jeder Stream hat seinen eigenen Kontext. Ein wird weitverbreitet eingesetzt um, das unterbrechen von Nachrichten zu vermeiden. Denn es kann schon sein, das man eine dritte Anbieter Bibliothek verwendet und dies auch Nachrichten versendet und der Entwickler nicht weiß ob diese nicht seine eigene Nachrichten stören könnten.

[6]

Communicator Objekte

Es gibt zwei Arten der Communicator Objekte, die Intra-Communicator und die Inter-Communicator. Die Intra-Communicator, zur intra Gruppen Kommunikation (also gleiche Gruppe), sind dazu da um einen Kontext mit einer Gruppe zu verbinden. Inter-Communicator, zur inter Gruppen Kommunikation (also zwei Gruppen), zur Verbindung von zwei Gruppen, die eine Source und die andere Destination, und einem Kontext.

[6]

Anwendungstopologien

Prozesse in einer Anwendung können in bestimmten Topologien angeordnet werden, wie z.B. eine Drei- oder Vier-Dimensionales Gitter.

[6]

Benutzerdefinierte Datentypen

Es ist erlaubt benutzerdefinierte Datentypen (welche aus einem Set von Typen und Memory-Offsets besteht) zu verwenden, welche dann bei jeder Routine in MPI verwendet werden können.

[6]

Point-to-Point Kommunikation

MPI bietet eine Point-to-Point Kommunikation an. Die Nachrichten werden durch ihren Source-Prozess, Message Tag und Communication-Kontext identifiziert. Wobei die Source und der Tag "wild-carded" sein dürfen.

Eine Sendoperation kann durch einen der drei Kommunikationsmodelle stattfinden:

1. Standard-Mode: Im Standard-Mode, der Empfänger muss noch keine Nachricht an den Empfänger gesendet worden sein.
2. Ready-Mode: Im Ready Mode muss der schon einmal eine Nachricht an den Empfänger gesendet worden sein, sonst das Ergebnis unbestimmt. Im Standard-Mode der Sender kann weitermachen, ohne dass die Nachricht abgeschickt worden ist.

3. Synchronous-Mode: Im Synchronous-Mode das Programm kehrt nicht zurück, solange die Nachricht nicht beim Empfänger angekommen ist.

Für jede dieser Kommunikationsmodelle gibt es "blocking" und "non-blocking" Routinen. Das gilt für Empfänger und Sender.

[6]

Collective Kommunikation

Collective Kommunikationsroutinen bieten eine koordinierte Kommunikation zwischen Prozessen. Dabei ist zu beachten, dass MPI keine Synchronisationsbarrieren implementiert, d.h. wenn Synchronisation benötigt wird, muss man sich selber darum kümmern.

MPI Collective Kommunikation wird in zwei breite Klassen unterteilt:

[6]

Collective data movement routines

Es gibt drei Basis-Typen der Collective data movement routines: broadcast, scatter, und gather. Von jedem gibt es zwei Versionen. Im "one-all" Fall, werden Daten zwischen einem Prozess und allem anderen kommuniziert. Im "all-all" Fall werden Daten zwischen jedem Prozess und allem anderen kommuniziert.

[6]

Global computation routines

Es gibt zwei Basis-Typen Global computation routines in MPI: reduce und scan. Beide der Routinen benötigt eine Input-Funktion. MPI enthält vier reduce und scan Routines.

[6]

Cloud-Computing und Verteilte Betriebssysteme

Wenn man sich die heutigen Cloud-Anbieter ansieht, merkt man schnell, dass die Motivationen und Vorteile einer solchen Cloud sich sehr mit denen von verteilten Betriebssystemen decken. Manche Anbieter setzen spezielle Cloud Betriebssysteme, welche eine spezielle Art von verteilten Betriebssystemen sind. Ohne verteilte Betriebssysteme würde Cloud-Computing nicht funktionieren und könnte nicht so eine Flexibilität zur Verfügung stellen. [17] [18]

Conclusio

Cloud-Computing ist nicht ohne Grund weitverbreitet und wird von vielen Unternehmen eingesetzt. Die Flexibilität und Transparenz der dahinterliegenden Technologien, gewährleisten eine einfache und sichere Lösung. Doch wird sich Cloud-Computing erst ab einer gewissen Größe auszahlen.

Abbildungsverzeichnis

Abbildung 1: Ansätze des Cloud-Computings [11].....	7
Abbildung 2: Virtualisierung [9].....	12
Abbildung 3: Virtualisierung vs. Container [10]	13
Abbildung 4: VM-Umgebung [10].....	13
Abbildung 5: Containerknoten Umgebung [10]	14
Abbildung 6: Shared-memory Multiprozessoren [6](S.2).....	17
Abbildung 7: Uniform Memory Access [6](S.3)	18

Literaturverzeichnis

[1] Titel: Virtualisierung: Geschichte, Techniken und Anwendungsfälle; [Studienarbeit]

Person/Institution:Mühe, Henrik

Ausgabe: 1. Aufl.

Verlag: München: GRIN Verl.

Erscheinungsjahr: 2007

ISBN/ISSN: ISBN 978-3-638-95252-1

[2] Titel:Cloud Computing : Praxisratgeber und Einstiegsstrategien

Verfasserangabe: Mario Meir-Huber

Person/Institution:Meir-Huber, Mario

Verlag: Frankfurt am Main: Entwickler.press

Erscheinungsjahr: 2010

Beschreibung: Literaturangaben

ISBN/ISSN: ISBN 978-3-86802-055-7

[3] Titel:Ratgeber Virtualisierung & Cloud Computing : Grundlagen, Planung, Praxis

Verfasserangabe: [Chefred.: Michael Eckert]

Sonstige Beteiligte:Eckert, Michael [Red.]

Verlag: München: TecChannel

Erscheinungsjahr: 2011

ISBN/ISSN: ISBN 978-3-942922-01-2

[4] An Open MPI-based Cloud Computing Service Architecture, WEI-MIN JENG and HSIEH-CHE TSAI

[5] Title: Verteilte Systeme - Prinzipien und Paradigmen

Verfasserangabe: Andrew S. Tanenbaum, Maarten van Steen

Auflage: 2. Auflage

Verlag: Pearson

Erscheinungsjahr: 2008

ISBN: 978-3-8273-7293-2

[6] Titel: Distributed operating systems-CHAPTER 17

Verlag: Elsevier Inc.

Erscheinungsjahr: 2010

ISBN: 978-1-4377-7807-6

[7] Titel: Virtualisierung als Basis des Cloud Computings

Autor: Klaus Foitzick

Datum: 17.08.2012

Online: <https://www.global.de/blog/virtualisierung-als-basis-des-cloud-computing/>

Zuletzt abgerufen am 18.02.2016

[8] Titel: Virtualisierung als Basis des Cloud Computings

Autor: Happyware Server Europe GmbH

Datum: 2015

Online: <https://www.happyware.com/de/it-loesungen/virtualisierungsloesungen/arten-der-virtualisierung/>

Zuletzt abgerufen am 18.02.2016

[9] Titel: Hypervisors, virtualization, and the cloud: Learn about hypervisors, system virtualization, and how it works in a cloud environment

Autor: Bhanu P Tholeti, Systems Engineer and Architect, IBM

Datum: 23 September 2011

Online: <http://www.ibm.com/developerworks/cloud/library/cl-hypervisorcompare/>

Zuletzt abgerufen am 18.02.2016

[10] Titel: Zuverlässiger IT-Betrieb mit Containern

Autor: Jens KuehlersDACH + CEE, Cloud Platform Solutions Engineer, Google Berlin

Datum: 10. November 2015

Aus Folien.

[11] Titel: IaaS, PaaS and SaaS Terms Clearly Explained and Defined

Autor: Bart Czernicki

Datum: 27.2.2011

Online: <http://www.silverlighthack.com/post/2011/02/27/laaS-PaaS-and-SaaS-Terms-Explained-and-Defined.aspx>

Zuletzt abgerufen am 18.02.2016

[12] Titel: Introduction to the New Mainframe: z/OS Basics

Autor: Mike Ebbers, John Kettner, Wayne O'Brien, Bill Ogden, IBM Redbooks

Verlag: IBM Redbooks

Erscheinungsjahr: 2012

ISBN 978-0-7384-3534-3

[13] Titel: The Architecture of VMware ESXi (White paper)

Autor: VMWare Inc.

Erscheinungsjahr: 24.10.2008

Online: http://www.vmware.com/files/pdf/ESXi_architecture.pdf

Zuletzt abgerufen am 21.2.2016

[14] Titel: Shared Memory Multiprocessors

Autor: Marco Ferretti

Online: <http://www-5.unipv.it/mferretti/cdol/aca/Charts/07-multiprocessors-MF.pdf>

Zuletzt abgerufen am 21.2.2016

[15] Titel: Algorithms Implementing Distributed Shared Memory

Autoren: Stumm, M. and Zhou, S.

Online: <http://www.eecg.toronto.edu/~stumm/papers/Zhou-IEEEComputer90.pdf>

Zuletzt abgerufen am 21.2.2016

[16] Titel: Cluster Operating Systems

Autoren: Russell W Clarke und Benjamin T B Lee

Online: <http://www.buyya.com/csc433/ClusterOS.pdf>

Zuletzt abgerufen am 21.2.2016

[17] Titel: Distributed and Cloud Computing: From Parallel Processing to the Internet of Things

Autoren: Kai Hwang, Jack Dongarra, Geoffrey C. Fox

VerlagMorgan Kaufmann

Jahr: 2013

ISBN: 978-0-1280-0204-9

[18] Titel: Amazon Elastic Compute Cloud (EC2)

Autor: Amazon

Online: <https://aws.amazon.com/ec2/>

Zuletzt abgerufen am 21.2.2016