

Vergleichende Technologien - Grid-Computing

Sven Hartlieb

Fakultät für Informatik
Hochschule Mannheim
Paul-Wittsack-Straße 10
68163 Mannheim
sven.hartlieb@gmail.com

Zusammenfassung Immer größerer Bedarf nach steigender Rechenleistung zwingt zu entsprechender technischer Abhilfe. Ohne neue Technologien können riesige Datenmengen und Rechenleistung, wie sie heutzutage benötigt werden, nicht bewältigt werden. Eine dieser Technologien ist das Grid-Computing. Überblick, Definition und Abgrenzung zu anderen Technologien sind die Eckpunkte dieser Ausarbeitung.

1 Motivation und Hintergrund

1.1 Warum Grid-Computing?

Der Bedarf nach Grid-Computing beruht grob gesagt auf zwei Anforderungen. Zum einen, der immer größere Bedarf an Rechenleistung und der Anspruch, immer größere Datenmengen speichern und auswerten zu können, hierzu einige Beispiele: Der Large Hadron Collider (LHC) im CERN besitzt mehrere Detektoren, die allesamt Datenmengen durch die Versuche sammeln. Allein der CMS Detektor wird pro Sekunde etwa 100 Terabyte an Daten produzieren. Mit Technologien vor dem Grid-Computing ist es nicht möglich, diese Menge an Daten zu analysieren und auszuwerten. [1]

Ein anderes Beispiel: Die Suche nach einem Malaria Wirkstoff konnte mittels des Grids des EGEE (Enabling Grids for E-SciencE) in zwei Tagen abgeschlossen werden. Ohne die Nutzung eines Grids hätte dieser Prozess an einem einzelnen Rechner Monate gedauert. [2]

Zum anderen gibt es in einer immer globalisierteren Welt immer mehr standortübergreifende Projekte und Kooperationen, die eine entsprechende Infrastruktur benötigen, um effizient verwirklicht zu werden.

Diese Gründe veranlassen dazu, das Grid-Computing als Forschungsgegenstand weiter in Betracht zu ziehen, auf den in der vorliegenden Arbeit ein kurzer Einblick gewährt wird.

1.2 Der Begriff „Grid-Computing“

Der Begriff „Grid“ wurde gewählt, um den Vergleich zum Stromnetz zu haben („Power Grid“ ist die amerikanische Bezeichnung für das Stromnetz). Grund ist, dass es den Benutzer nicht interessiert, woher er den Strom bezieht. Er nutzt ihn einfach. Genauso war die Vision für das Grid-Computing: Transparenz bezüglich der Ressourcennutzung für den Benutzer.

1.3 Geschichte des Grid-Computing

Zunächst soll ein kurzer Abriss über die Geschichte des Grid-Computing gegeben werden, und welche Faktoren an dessen Entwicklung beteiligt waren. Etwa in den 1990ern gab es erste Bestrebungen, verschiedene US-Supercomputer-Center miteinander zu vernetzen, dies fiel unter das Schlagwort „Metacomputing“. [3] Im Jahr 1994 haben Ian Foster, Steven Tueck und Carl Kesselman das Globus Projekt ins Leben gerufen. Ziel war die Erschaffung von Software, die weltweit vernetzte wissenschaftliche Zusammenarbeit ermöglichte. 1997 veröffentlichten sie die erste Version des Globus Toolkits, der erste Vorläufer für Grid-Software. [4]

Etwa zur selben Zeit wurden die Projekte FAFNER (Factoring via Network-Enabled Recursion) und I-WAY (Information Wide Area Year) gegründet, die technologische Meilensteine legten, die für das Grid-Computing relevant sein würden. Sie beeinflussten damit unter anderem maßgeblich das SETi@home sowie das Globus Projekt. Der Begriff „Grid“-Computing wurde dann auch 1997 geboren, auf einem Workshop namens „Building a Computational Grid“, der an dem Argonne National Laboratory gehalten wurde, dem Labor von Ian Foster und Steven Tueck. 1998 haben Ian Foster und Carl Kesselman „The Grid: Blueprint for a New Computing Infrastructure“ veröffentlicht, das auch die „Grid Bible“ genannt wird. [5]

2001 startete das European Data Grid (EDG) einer europäischen Initiative, dass die Möglichkeiten des Grid Computing erforschte. Dieses wurde 2004 von Nachfolgeprojekt EGEE (Enabling Grids for E-Science) abgelöst. Dieses entwickelte unter anderem die Grid-Middleware gLite. 2008 begann die dritte Phase dieses Projektes und Ziel Schaffung einer nachhaltigen Europäischen Grid-Infrastruktur. [6]

2 Bekannte Grid Projekte

2.1 Das LHC Computing Grid

Das LHC Computing Grid (LCG) wurde 2003 ins Leben gerufen. Ziel dieses Projektes ist, mit den ungeheuren Datenmengen des LCG umgehen zu können.

Als Beispiel sei genannt, dass ca. 15 Mio. GB pro Jahr an Datenmengen erzeugt werden.

Dieses Netzwerk verteilt die bei den Experimenten am CERN anfallenden Daten auf die ganze Welt. Beteiligt sind dabei 170 Organisationen aus 34 Ländern, vorwiegend aus dem universitären Bereich. Die angeschlossenen Rechenzentren sind dabei in einem hierarchischen System angeordnet und verarbeiten, analysieren und speichern somit die Daten aus dem LHC. Diese Hierarchie besteht dabei aus verschiedenen Schichten (Tiers). Das CERN selbst fungiert dabei als Tier-0 Zentrum, in dem die Daten erfasst und gespeichert werden. Von dort werden die Daten an 11 Tier-1 Zentren verteilt, die aufgrund ihrer Kapazität einen Großteil der Datenmengen aufnehmen können. Die Tier-2 Zentren übernehmen dann jeweils spezifische Analysen, in die Daten an sie weitergereicht werden. Über die Tier-3 Zentren können dann Forschende mittels ihrer Computer auf diese Daten zugreifen. [7]

2.2 Die D-Grid Initiative

Die D-Grid Initiative [8] wurde vom Bundesministerium für Bildung und Forschung (BMBF) im Jahr 2004 gegründet. Ziel war die Förderung von entsprechenden Grid-Projekten, die ersten Projekte starteten 2005. Einige dieser Projekte sind Aerogrid oder das D-Grid Integrationsprojekt (DGI). Das DGI war eine der ersten Grid-Projekte dieser Initiative und stellt unter anderem Infrastrukturkomponenten bereit, um eigene Grids aufbauen zu können. Um den Aufbau und Betrieb einer Grid-Plattform zu unterstützen wurden mehrere überlappende Stufen geplant. In der ersten Stufe (2005 – 2008) wurde begonnen, eine verteilte, integrierte Ressourcenplattform zu entwickeln. Hierbei wurden zunächst Grid-Dienste für Wissenschaftler bereitgestellt, die aufgrund des Bedarfs an rechen- und daten-intensiven Anwendungen besonderes Interesse hatten. In der zweiten Stufe (2007-2010) wird auf der Entwicklung der ersten Stufe aufgebaut und IT Services für Wissenschaftler als auch für interessierte aus der Industrie angeboten. Bis zum heutigen Tag wurde mehr als 100 Millionen Euro für diese Initiative ausgegeben und für ca. 35 Projekte verwendet. [9]

3 Grid-Computing

3.1 Definitionen

Nachdem der Nutzen und die Geschichte kurz umrissen wurden, soll nun erklärt werden, was Grid- Computing an sich denn ist, bzw. wie es definiert ist. Eine erste Definition haben Foster und Kesselman gegeben:

“A computational grid is a hardware and software infrastructure that provides dependable, consistent, pervasive, and inexpensive access to high-end computational capabilities.” [5]

Diese Definition erweiterten sie in der zweiten Auflage ihres Buches. Der Grund war, dass Grids eben nicht nur als Möglichkeit zur Benutzung von komplexen Berechnungsproblemen herangezogen werden, sondern Zugang für alle Arten von Ressourcen bildeten. Es wurde dann gesagt, dass Grid-Computing sich um die koordinierte Verteilung und Problemlösungsmöglichkeit in dynamischen, multi-institutionellen virtuellen Organisationen dreht.

Eine gute Umschreibung von virtuellen Organisationen wird auf gridcafe.org wiedergegeben:

Virtual organizations (VOs) are groups of people who share a data-intensive goal. To achieve their mutual goal, people within a VO choose to share their resources, creating a computer grid. This grid can give VO members direct access to each other's computers, programs, files, data, sensors and networks. This sharing must be controlled, secure, flexible, and usually time-limited. [10]

Eine weitere Definition von Grid-Computing lautet so:

„Per Definition ist das Grid Computing eine Technik zur Integration und zur gemeinsamen, institutionsübergreifenden, ortsunabhängigen Nutzung verteilter Ressourcen auf Basis bestehender Kommunikationsinfrastrukturen wie z.B. dem Internet“. [11]

3.2 Abgrenzung zu anderen Computing Systemen

Da zuvor Grids definiert wurden, bietet es sich an, mit anderen Computing Systemen zu vergleichen.

Abgrenzung zu parallelem Rechnen: Das Hauptaugenmerk beim parallelen Rechnen liegt auf der Aufteilung von Rechenzeit auf mehrere Prozessoren. Dies kann mittels einzelner Computer mit mehreren Prozessoren geschehen, als auch durch mehrere Rechner. Die effiziente Lösung rechenintensiver Aufgaben steht hier also im Vordergrund. Ein weiterer Punkt ist, dass es hierbei meist eine zentrale Instanz gibt, die die Berechnungen kontrolliert und die Ergebnisse zusammenführt. Dies ist bei Grid-Computing nicht der Fall. [10]

Abgrenzung zu Cluster-Computing: Die Ressourcen in einem Grid gehören in der Regel mehreren, unabhängigen Organisationen an. In einem Cluster gehören die Ressourcen für gewöhnlich nur einer organisatorischen Einheit an. Weitere entscheidende Merkmale, die bei Cluster Computing fehlen, sind die verteilte Kommunikation sowie die offenen Schnittstellen. [11]

Abgrenzung zu Cloud-Computing: Der Unterschied zum Cloud-Computing ist in kurzen Worten gefasst, dass Grid-Computing weniger, aber aufwändige Anfragen an die Ressourcen stellt. Während beim Cloud-Computing häufige, aber dafür wenige Ressourcenanfragen beansprucht werden. [12] Ein weiterer

Punkt ist, dass Cloud-Computing der zentralen Kontrolle des Betreibers unterliegt, Grid-Computing jedoch nicht. Zudem muss für das Grid-Computing eine Grid-Middleware installiert lokal werden, während beim Cloud-Computing alle Ressourcen durch Netzwerkdienste zur Verfügung gestellt werden. [13]

3.3 Arten von Grids

In der Anfangszeit des Grid-Computing gab es Bestrebungen, Grids hinsichtlich ihres Aufgabengebietes zu kategorisieren. Diese Unterteilung ist jedoch nicht mehr zeitgemäß, da Grids rein praktisch für alle Aufgabengebiete angewendet werden. Diese Einteilung ist somit historisch zu betrachten und soll im folgenden eher als Auflistung betrachtet werden, welche Aufgabengebiete mit Grids bearbeitet werden können.

1. Das **Information-Grid** stellt lediglich Daten bereit. Damit kann das Internet als allseits bekanntes Grid bezeichnet werden, dass durch seine Dienste world wide web, ftp, etc Informationen liefert.
2. Das **Ressource-Grid** liefert Ressourcen, die von anderen genutzt werden können. Dazu zählt zum Beispiel Rechenleistung und Speicherkapazität. Zu Beginn des Grid-Computings wurde noch weiter unterschieden, zum einen in Computational-Grids. Hierbei werden die Ressourcen mehrerer Rechner zusammengefasst, um die Gesamtrechenleistung zu erhöhen. Je nach verwendeter Plattform kann man hierbei noch in Desktop-, Server- oder High Performance-Grids unterscheiden. Zum anderen in Data-Grids, hierbei steht die gemeinsame Nutzung und Verarbeitung großer Datenmengen im Vordergrund. Die Unterscheidung in Computational- und Data-Grids wird heute jedoch nicht mehr vorgenommen, da Grids heutzutage nicht mehr für nur eine Art von Ressourcen genutzt werden.
3. Im **Service-Grid** hat man nicht mehr direkten Zugriff auf Informationen und Ressourcen, sondern diese werden durch bereitgestellte Dienste (Services) zur Verfügung gestellt. Dies wird zum Beispiel mittels OGSA oder SOAP bereitgestellt.

3.4 Merkmale von Grids

Das Auftreten von vielen Begriffen wie Data-Grids, Science-Grids, Campus-Grids, etc. veranlassten Ian Foster, einer der Pioniere des Grid-Computing, sich Gedanken zu machen, was ein Grid letztlich ausmacht. Er hat in einem Artikel [14] mehrere Punkte herauskristallisiert, um ein Grid eindeutig zu kennzeichnen:

1. Ein Grid koordiniert dezentrale Ressourcen. Das heisst, dass die Ressourcen von mehreren Benutzern in Anspruch genommen werden können. Zudem existiert keine zentrale Einheit zur Koordination, sondern die Abstimmung untereinander wird durch die Prozesse untereinander geregelt.

2. Grids verwenden offene, standardisierte Protokolle und Schnittstellen. Dieser Grund ist deswegen unverzichtbar, da die Koordination von getrennt verwalteten Rechnern nur dann geschehen kann, wenn die hierzu nötigen Schnittstellen bekannt und hinreichend verbreitet sind.
3. Grids bieten nicht-triviale Dienste an. Durch das Angebot mehrerer Dienste bietet das Grid Möglichkeiten, die mehrere einzelne Rechner nicht bieten könnten.

Dunkel et al. [10] zählen noch weitere Punkte als Merkmale eines Grid-Systems auf: Sie betonen, dass die verwendeten Ressourcen nicht in einer gemeinsamen verwalteten Domäne liegen. Rein praktisch gesehen bedeutet dies, dass an einem Grid mehrere Organisationen beteiligt sind. Die gute Service-Qualität ist ein weiterer Punkt. Damit sind Aspekte wie Verfügbarkeit der Ressourcen und kurze Antwortzeiten gemeint. Ohne diese Zuverlässigkeit wird ein Benutzer kaum mit einem Grid arbeiten wollen.

4 Realisierung von Grid-Systemen

4.1 Middleware für Grid-Systeme

Nachdem nun die theoretischen Grundlagen geklärt wurden, sollen nun einige technische Erklärungen folgen, die sich um die notwendige Middleware drehen. Middleware wird für Grid-Computing benötigt, damit verteilte Anwendungen eine Möglichkeit haben, überhaupt das Grid benutzen zu können. Middleware ist also eine Vermittlungs- und Verwaltungssoftware. Damit verschiedene Systeme über das Grid überhaupt miteinander kommunizieren können, werden gewissen Standards benötigt, zum Beispiel Protokolle. Diesem Problem entgegenzuwirken, hat sich das Open Grid Forum (OGF) bemüht. Das OGF ist dabei ein Gremium, das Standards und Spezifikationen für das Grid-Computing liefert. [15] Es besteht aus Mitarbeitern aus Industrie und Forschung aus über 400 Organisationen aus über 50 Ländern. Mitglieder sind unter anderem IBM, Microsoft und Oracle. Das OGF hat zum Beispiel eine standardisierte Architektur vorgeschlagen. Diese nennt sich Open Grid Services Architecture (OGSA). Die drei bekanntesten Middleware Systeme sind hierbei das Globus Toolkit, gLite sowie Unicore. Auf letzteres wird in Kapitel 4.3 eingegangen.

4.2 Architektur für Middleware-Systeme

Die OGSA spezifiziert hierbei 4 aufeinander aufbauende Schichten.

1. Die unterste Schicht (Ressourcen Schicht) besteht aus denjenigen Ressourcen, die nach außen hin angeboten werden. Zugriff erfolgt hierbei über bestimmte Zugriffsprotokolle.

2. Die Open Grid Services Infrastructure (OGSI) stellt die Verbindung zwischen den Ressourcen der untersten Schicht her und den den zentralen benötigten Grid-Diensten der Schicht darüber. Ein Grid-Dienst kann hierbei als eine Art Web Service betrachtet werden, der intern Zustände verwalten kann. Dies geschieht wie bei den Web Services durch den Austausch von Nachrichten mittels XML sowie die Beschreibung des Dienstes. Dies geschieht hier jedoch mit einer erweiterten Form, die GWDSL heisst, im Gegensatz zu WDSL der typischen Web Services.
3. Die OGSA Platform Services stellen einerseits den Einstiegspunkt für Anwendungen dar, und bieten andererseits zentrale, für das Grid benötigte Dienste an. Hierzu gehören zum Beispiel Optimierung der Ressourcennutzung oder Verwaltung der Grid-Ressourcen.
4. Die oberste Schichte besteht dann aus den Anwendungen selbst, die das Grid benutzen möchten.

4.3 UNICORE

4.4 Hintergrund

Als Beispiel für eine Grid-Middleware soll UNICORE kurz aufgezeigt werden, das seit 1997 am Forschungszentrum Jülich entwickelt wird. Der Grund für die Wahl von UNICORE hierfür liegt unter anderem in seiner leichten Benutzbarkeit aufgrund seiner grafischen Benutzeroberfläche. Während das Globus Toolkit keine GUI und das gLite durch mangelnde Portabilität Nachteile aufweisen, kann Unicore hier punkten. Neben des Vorteils der GUI ist es betriebssystemunabhängig, da es in Java implementiert ist. Dies wird durch eine leichte Installation und Handhabung abgerundet.

4.5 Beispiel

Nun sollen exemplarisch zwei Beispiele gezeigt werden, wie UNICORE genutzt werden kann. Dazu wird die Rich Client Version der UNICORE Software verwendet, einer Software, die auf Basis der Eclipse Rich Client Plattform geschaffen wurde. [16]

Sind die Vorbereitungen getroffen (Installation, hinzufügen eines Keystores sowie einer Registry) kann begonnen werden. Das Forschungszentrum Jülich bietet zwei sogenannte „Targets“ an. Dies sind Ziele, an die die zu verarbeitenden Anweisungen zu schicken sind; diese werden „Jobs“ genannt. Mit einem Rechtsklick auf ein Target kann dann im Menü „create job“ ausgewählt werden. Für das „Hello world!“ Beispiel wird hier eine „Script V.2.0“-Applikation ausgewählt. Es erscheint dann ein Fenster, in dem ein Skript eingegeben werden kann. Für unser Beispiel ist das „echo „Hello world!““. Durch Anklicken des „submit“ Buttons wird der Job übertragen. Ist der Job übertragen und ausgeführt worden, erscheint die Möglichkeit das Ergebnis mittels des Buttons „fetch output files“

zu holen. In einem eigenen Abschnitt in der GUI erscheint dann die erwartete Ausgabe: Hello World.

Für das andere Beispiel wird bei „create job“ die Option POV-Ray v.3.51 ausgewählt. Dies ist ein Computergrafikprogramm, dessen Rechenaufwand zur Bilderzeugung so hoch ist, dass es Minuten bis Tage dauern kann, bis ein Bild dargestellt werden kann. Im POV-Ray Reiter „Source File“ kann dann ein Quellcode für eine zu erstellende Grafik ausgewählt werden.

Diesen Quellcode bekommt man zum Beispiel durch mitgelieferte Beispiele. Anschließend wird wieder mittels submit der Job übertragen und mittels fetch output files das Ergebnis geholt. In diesem Fall ist es nun eine erzeugte Grafik.

5 Schlusswort

Ziel dieser Ausarbeitung soll sein, die fundamentalen Grundlagen des Grid-Computing zu vermitteln. Der Leser sollte nun einen Überblick gewonnen haben, was unter Grid-Computing zu verstehen ist, sowohl auf theoretischer Ebene als auch die praktische Anwendung. Auch wenn Grid-Computing nicht mehr die Technologie der Stunde ist, bietet sie doch ein interessantes Feld, das im Endeffekt nur durch aktuellere Technologien wie das Cloud-Computing ersetzt werden kann.

Literatur

1. Dr. Berlich, Rüdiger und Dr. Kunze, Marcel. *Grid Computing: Von der Forschung zur Anwendung*. Institut für Wissenschaftliches Rechnen, Forschungszentrum Karlsruhe
2. EEEG - Malaria <http://public.eu-egge.org/files/EGEE-Malaria-Medizin-DA.pdf>
3. Gridcafe <http://www.gridcafe.org/grid-computing-ancestors.html>
4. The Globus Alliance <http://www.globus.org/>
5. Foster, Ian and Kesselman, Karl *The Grid: Blueprint for a New Computing Infrastructure*
6. Enabling Grids for E-Science <http://www.eu-egge.org/>
7. LHC Computing Grid <http://www.swiss-lhc.ch/index.php>
8. Die D-Grid Initiative <http://www.d-grid.de>
9. Das D-Grid <http://www.d-grid-ggmbh.de/index.php?id=51>
10. Dunkel, Jürgen et al. *Systemarchitekturen für verteilte Anwendungen*, Carl Hanser, 2008
11. Bengel, Günther et al. *Masterkurs Parallele und Verteilte Systeme*, Vieweg+Teubner, 2008
12. Lotze, Martin *Grid-Computing - Rechenpower aus der Steckdose?* Seminararbeit an der der FH Würzburg-Schweinfurt, 2009
13. Karger, Björn *The Cloud vs. The Grid* Proseminar Technische Informatik. 28. Januar 2009
14. Foster, Ian *What is the Grid? A Three Point Checklist*. Argonne National Laboratory and University of Chicago, July 20, 2002

15. Open Grid Forum http://www.ogf.org/About/abt_overview.php
16. Unicore Rich Client [http://sourceforge.net/projects/unicore/files/2%20UNICORE%206%20Clients%2C%20Applications%2C%20and%20APIs/2.1%20UNICORE%20Rich%20Client%20\(URC\)/6.2.2/](http://sourceforge.net/projects/unicore/files/2%20UNICORE%206%20Clients%2C%20Applications%2C%20and%20APIs/2.1%20UNICORE%20Rich%20Client%20(URC)/6.2.2/)