

Hochschule für Technik und Wirtschaft HTW Chur

Telekommunikation/Elektrotechnik

Verteilte Systeme

Grid Computing

Verteilung von Ressourcen zur Lösung eines Problems

Reto Castellazzi

Dieser Bericht wurde im Rahmen des Bachelor-Studienganges
Telekommunikation/Elektrotechnik an der Hochschule für Technik und Wirtschaft
HTW Chur erstellt.

Studierender

Reto Castellazzi
Büheli 3
7204 Untervaz

Dozent

Prof. Martin Studer
HTW Chur
Ringstrasse/Pulvermühlestrasse 57
7004 Chur

Inhaltsverzeichnis

1	Was ist Grid Computing	4
1.1	Merkmale eines Grid-Systems	4
2	Arten von Grids.....	5
2.1	Rechen-Grid	5
2.2	Datenbank-Grid	5
2.3	Ressourcen-Grid.....	6
3	BOINC	6
3.1	Was ist BOINC.....	6
3.2	Rechenleistung für BOINC zur Verfügung stellen	6
3.2.1	Funktionen der Software.....	7
3.3	Ein eigenes BOINC-Projekt erstellen	7
3.4	Eine eigene BOINC-Applikation erstellen.....	8
4	Literaturverzeichnis	9

Abbildungsverzeichnis

Abbildung 1	Ein Grid stellt Ressourcen zur Verfügung [1].....	4
Abbildung 2	Aufgaben von BOINC auf dem PC [3].....	7
Abbildung 3	Funktionen der Software [3]	7
Abbildung 4	BOINC Wrapper [6]	8

1 Was ist Grid Computing

Im Bereich der IT wird der Begriff Grid im Sinne von Strom- oder Wassernetz verwendet. Man soll Rechenleistung mit Hilfe eines Grid wie Strom aus der Steckdose beziehen können. Wenn Rechenleistung benötigt wird, um ein bestimmtes Problem zu lösen, sollte man seinen lokalen Computer mit der zu lösenden Problemstellung an ein Grid anschliessen können und nach möglichst kurzer Zeit die Problemlösung aus dem Grid bekommen. Es spielt dabei keine Rolle, welches Element des Grid wirklich die Leistung zur Verfügung stellt. Für den Endanwender bleibt dies transparent. Heutige Grids sind von dieser Vision jedoch noch weit entfernt. [1]

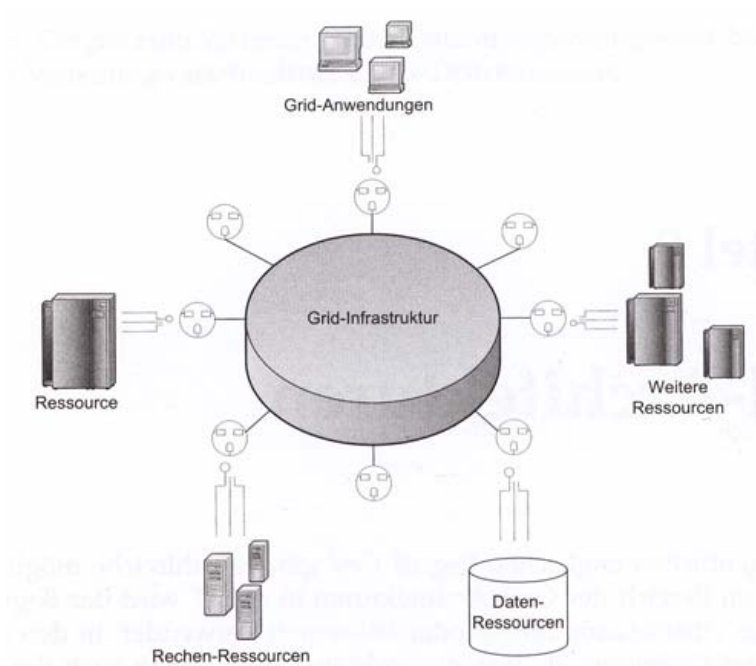


Abbildung 1 Ein Grid stellt Ressourcen zur Verfügung [1]

1.1 Merkmale eines Grid-Systems

Grid ist die Middleware, die Anwendungen den Zugang zu leistungsstarken Berechnungsmöglichkeiten bereitstellt.

Die wichtigsten Merkmale eines Grid-Systems:

Ressourcen werden in einer koordinierten Art und Weise geteilt. Die Ressourcen eines Grid können von mehreren Teilnehmern am Grid genutzt werden. Dabei wird die Nutzung der Ressourcen so gesteuert, dass eine möglichst gleichmässige Auslastung erreicht wird. Ressourcen können unter anderem Cluster, Massenspeicher, Datenbanken, Anwendungen oder Messgeräte sein.

Es gibt keinen zentralen Kontrollpunkt, der sämtliche Aktivitäten koordiniert. Es gibt keine zentrale Einheit für die Koordination, sondern die Koordination wird durch die nutzenden Prozesse untereinander geregelt.

Die verwendeten Ressourcen liegen in getrennt administrierten Domänen. Im Unterschied zu einem Cluster spricht man nur von einem Grid, wenn die Ressourcen nicht in einer zentralen Administrationsdomäne liegen.

Die Kommunikation mit den Ressourcen verwendet standardisierte, offene und allgemeine Protokolle und Schnittstellen. Die Schnittstellen und Protokolle gewährleisten grundsätzliche Funktionen für die Authentifizierung, die Autorisierung, die Ressourcen-Ermittlung und den Ressourcen-Zugriff. Da Prozesse von getrennt administrierten Rechnern, sich selbst bei der Nutzung von Ressourcen koordinieren sollen, ist dieses Merkmal unverzichtbar.

Die Dienste bieten eine gute Service-Qualität. Um die flexible Nutzung der Ressourcen zu ermöglichen, ist es nötig, dass die Ressourcen eine gute Verfügbarkeit und kurze Antwortzeiten verfügen. Auch Durchsatz, Erreichbarkeit oder Sicherheit sollten eine hohe Dienstgüte aufweisen.

[1] [2]

2 Arten von Grids

2.1 Rechen-Grid

Das Rechen-Grid bündelt in erster Linie die Leistung vieler einzelner Rechner und bietet diese zentral an. So können Probleme, deren Berechnung auf wenigen Rechnern lange dauern würde durch den Einsatz von Parallelisierung schneller berechnet werden. Die Aufgaben eines Rechen-Grid liegen im Bereich der rechenintensiven Anwendungen wie z.B. der Vorausberechnung des Wetters. Das Grid dient dazu, sehr hohe Rechenleistung zur Verfügung zu stellen.

2.2 Datenbank-Grid

Bei einem Datenbank-Grid geht es darum, riesige Datenbankkapazitäten, die die Kapazitäten eines einzelnen Datenbankservers sprengen würden, für Nutzer eine einheitliche Zugriffsstelle anzubieten. Der Nutzer stellt seine Anfrage an das Grid und dieses holt automatisiert die Anfrageergebnisse aus den unterschiedlichen Datenspeichern. Zusätzlich werden diese in einer einzigen Ergebnisliste für den Nutzer kombiniert. Dem Nutzer bleibt dank des Grid der Zugriff auf unterschiedliche, verteilte Datenbestände erspart. Die Verteilung verbessert ausserdem die Service-Qualität durch verkürzte Antwortzeiten.

2.3 Ressourcen-Grid

Ressourcen-Grid stellen dem Nutzer bestimmte Ressourcen zur Verfügung, die er temporär oder dauernd nutzen kann. Bei diesen Ressourcen handelt es sich meist um Speicherplatz. In diesem Zusammenhang muss man sich die Grid-Anwendung eher als eine Art Client-Server-Anwendung vorstellen, die das Grid zur Nutzung der Ressourcen verwendet. Die Anwendungen sind eher einfach gehalten. Es gibt bereits zahlreiche praktische Implementierungsbeispiele von Ressourcen-Grids wie z.B. Amazon S3 (stellt zentralen Speicherplatz zur Verfügung) oder das Fraunhofer Resource-Grid (stellt zentrale Grid-Aufgaben mehreren Fraunhofer-Gesellschaften zur Verfügung).

[1]

3 BOINC

3.1 Was ist BOINC

BOINC ist die Abkürzung von Berkeley Open Infrastructure for Network Computing, übersetzt bedeutet das Berkeleys offene Infrastruktur für verteiltes Rechnen.

BOINC ist eine quelloffene Software-Architektur und Plattform für Verteiltes Rechnen. Sie wurde entwickelt um wissenschaftliche Projekte zu unterstützen. Privatleute können ihre ungenutzte Prozessorleistung den Projekten zur Verfügung stellen. Diese Projekte dienen beispielsweise der Suche nach Medikamenten, der Klimaforschung, sowie dem Lösen physikalischer und mathematischer Probleme.

Für solche Berechnungen werden sogenannte Supercomputer benötigt, welche aber nicht jedem Projekt zur Verfügung stehen. Mittels BOINC werden tausende Private Computer zu einem Supercomputer verbunden, sodass das Projekt trotzdem über eine große Rechenleistung verfügen kann. [3]

3.2 Rechenleistung für BOINC zur Verfügung stellen

Es ist sehr einfach Rechenleistung für BOINC zur Verfügung zu stellen. Dazu wird die Client-Anwendung von der BOINC-Website heruntergeladen und auf dem PC installiert. Danach muss man seine E-Mail-Adresse und das zu unterstützende Projekt angeben und schon funktioniert es. Der Client kann für viele verschiedene Plattformen heruntergeladen werden.

[4]

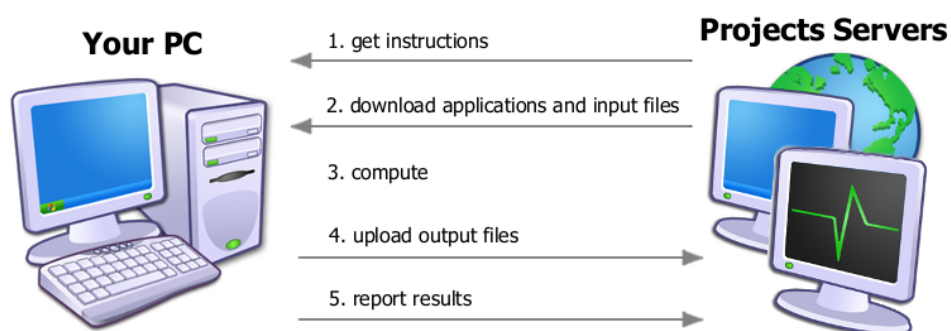


Abbildung 2 Aufgaben von BOINC auf dem PC [4]

3.2.1 Funktionen der Software

Die Scheduler und Daten-Server sind auf Computern installiert, die die Projekte verwalten. Mit diesen Elementen, wird die Rechenleistung auf die verschiedenen Client-Computen verteilt.

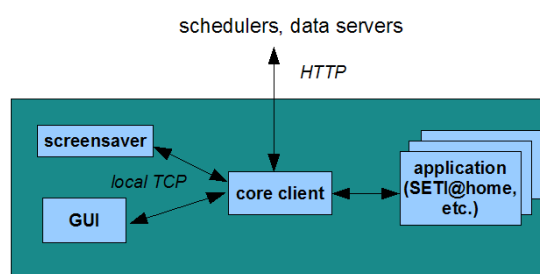


Abbildung 3 Funktionen der Software [4]

Der Core-Client kommuniziert mit externen Servern über das HTTP-Kommunikationsprotokoll um Aufgaben zu bekommen und Bericht über die getätigten Arbeit zu erhalten. Der Core-Client kontrolliert und startet die Anwendungen.

Die Applikationen sind die Programme, die die Rechenleistung benötigen. Es können mehrere gleichzeitig auf einem Computer mit mehr als einer CPU laufen.

Das GUI ist eine grafische Oberfläche zum steuern des Core-Client. Es kommuniziert mit dem Core-Client über eine TCP-Verbindung. Normalerweise ist dies eine lokale Verbindung, jedoch ist es möglich, einen Core-Client fernsteuern.

Man hat auch die Möglichkeit, den aktuellen Stand der Projekte jeweils am Screensaver anzeigen zu lassen. [4]

3.3 Ein eigenes BOINC-Projekt erstellen

Um ein eigenes BOINC-Projekt zu erstellen, benötigt man einen Server. Man kann diesen selber aufsetzen, oder einen vorbereiteten VM-Server herunterladen [5]. Nachdem man den Server eingerichtet hat, kann man ein Projekt anlegen.

3.4 Eine eigene BOINC-Applikation erstellen

Mit BOINC kann man verschieden Anwendungen in jeder Programmiersprache verteilen.

Wenn man eine BOINC-Anwendung entwickelt, will man diese auf möglichst vielen Plattformen ausführen können. Das zu erreichen kann eine Herausforderung BOINC bietet jedoch Tools und Beispiel Dateien an, um diese Aufgabe zu vereinfachen.

Subversion vereinfacht die Entwicklung von BOINC-Applikationen dazu kann das SVN-Verzeichnis eingebunden und der Source Code benutzt werden [6].

Wenn man mit Microsoft Visual Studio entwickelt, kann man einfach die Datei boinc_samples.sln aus dem SVN-Ordner boinc/win_build öffnen und builden.

Jede bestehende Anwendung (oder eine Folge von Anwendungen) kann unter BOINC mit Hilfe eines Wrapper-Programms laufen gelassen werden. Der Wrapper führt die Anwendungen als Teilprozesse aus und erledigt die gesamte Kommunikation mit dem Core-Client. [7]

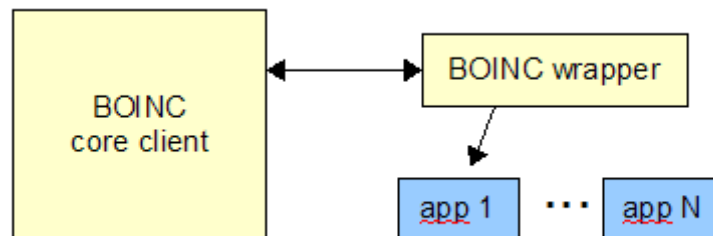


Abbildung 4 BOINC Wrapper [7]

4 Literaturverzeichnis

- [1] Jürgen Dunkel, Andreas Eberhart, Stefan Fischer, Carsten Kleiner, Arne Koschel: System Architekturen für verteilte Anwendungen. Hanser Verlag, München, 2008.
- [2] Wikipedia, Wikimedia Foundation.: Grid computing
[http://en.wikipedia.org/wiki/Grid_computing], Mai 2010.
- [3] University of California.: BOINC - Open-Source Software für Volunteer Computing und Grid Computing [<http://boinc.berkeley.edu/>], Mai 2010.
- [4] University of California.: How BOINC works
[http://boinc.berkeley.edu/wiki/How_BOINC_works], Juni 2010.
- [5] University of California.: The BOINC server virtual machine
[<http://boinc.berkeley.edu/trac/wiki/VmServer>], Juni 2010.
- [6] University of California.: Getting source code
[<http://boinc.berkeley.edu/trac/wiki/SourceCode>], Juni 2010.
- [7] University of California.: The BOINC Wrapper
[<http://boinc.berkeley.edu/trac/wiki/WrapperApp>], Juni 2010.