

Grid-Computing mit dem Globus Toolkit

Konzepte von Betriebssystem-Komponenten

Severin Strobl
severin@blue-neutrino.de

25. Januar 2005

Zusammenfassung

Dieses Dokument gibt einen Einblick in die Möglichkeiten des Grid-Computing im Allgemeinen und des Globus Toolkit im Besonderen. Dabei wird auf die grundlegenden Eigenschaften und Bestandteile des Globus Toolkit in den Versionen 2 und 3 eingegangen. Ebenso werden Hinweise zum Einsatz des Toolkits gegeben, sowie in einem eigenen Kapitel die Sicherheitsfunktionen des Globus Toolkits erläutert.

1 Grid-Computing

1.1 Was ist Grid-Computing?

„A computational grid is a hardware and software infrastructure that provides dependable, consistent, pervasive, and inexpensive access to high-end computational capabilities.“¹

Grid-Computing ist die nächste Stufe nach Cluster-Computing. Während in einem Cluster im Allgemeinen nur Rechner einer Architektur unter einem Betriebssystem auf relativ begrenztem Raum laufen, gilt diese Einschränkung bei einem Grid nicht. Ein Grid kann beliebig viele Rechner (Nodes genannt) an beliebigen Punkten der Erde verbinden, solange sie über eine Verbindung in Form eines Netzwerks, wie zum Beispiel des Internets, verfügen. Ein Grid bietet des Weiteren noch Abstraktionen bezüglich beliebiger Ressourcen wie Rechenleistung oder Speicherplatz oder auch spezieller Hard- oder Software. Für den Nutzer eines Grids sind alle verfügbaren und für ihn freigegebenen Ressourcen des Grids vollkommen unabhängig von örtlichen oder organisatorischen Eigenschaften nutzbar.

Ein Grid-System sollte hohe Anforderungen bezüglich der Ausfallsicherheit und Sicherheit erfüllen. Die Ausfallsicherheit wird in einem Grid meistens nicht durch teure Spezialhardware, wie redundante Auslegung aller wichtiger Teile wie CPU oder Festplatten, erreicht, sondern durch die Verteilung relativ einfacher Rechner auf verschiedene Gebiete. Die Sicherheit in einem Grid wird durch eine zentrale Verwaltung sichergestellt ebenso wie durch eine verschlüsselte Kommunikation der Rechenknoten und Clients untereinander.

1.2 Einsatzmöglichkeiten

Die zwei am meisten verbreiteten Einsatzgebiete von Grid Computing sind Data bzw. Computation Grids, welche im Folgenden kurz erläutert werden.

In vielen Bereichen der Technik und Forschung ist man inzwischen mit Datenmengen konfrontiert, die auf herkömmliche Art und Weise nicht mehr bewältigt werden können. Bei physikalischen Experimenten ergeben sich

¹Ian Forster und Carl Kesselman 1998 in „The Grid: Blueprint for a New Computing Infrastructure“

zum Beispiel durch die Auswertung von Messergebnissen Daten in der Größenordnung von mehreren Petabyte. Durch das Zusammenschalten von vielen normalen Rechnern zu einem Grid kann man die Ergebnisse in einem sog. Data-Grid speichern, wobei auf jedem der beteiligten Rechner jeweils nur ein kleiner Teil der insgesamt anfallenden Daten liegt. Ebenso kann man in einem Data-Grid beliebige Daten redundant speichern, was Vorteile wie einen schnelleren Zugriff (Striping) oder auch Sicherheit vor Datenverlust beim Ausfall einzelner Rechner bietet (Backup).

Ein weiteres mögliches Szenario für Grid-Computing ist die Verteilung von rechenintensiven Vorgängen auf mehrere Rechner. Dabei kann entweder ein Programm, das in mehreren Durchgängen laufen muss, auf mehreren Rechnern parallel statt auf einem Rechner nacheinander ausgeführt werden oder, bei entsprechender Programmierung, ein einziges Programm die Rechenleistung beliebig vieler Nodes für eine einzige Berechnung verwenden.

Das Interessante an einem Grid ist dabei, dass es auch organisationsübergreifend betrieben werden kann. So gibt es z.B. in Europa das DataGrid² oder in den USA das TeraGrid³, an dem jeweils zahlreiche Forschungsanstalten beteiligt sind.

2 Globus Toolkit

2.1 Entwicklung des Globus Toolkit

Der Globus Toolkit⁴ (GT) ist ein Open Source Programmpaket, welches eine Grundlage für die Entwicklung von Grid-Anwendungen darstellt. Der Toolkit selbst stellt dabei nur die für ein Grid unbedingt notwendige Basis wie Sicherheits-, Ressourcen- oder Jobmanagement zur Verfügung. Ebenso enthalten sind APIs für C/C++ und Java.

Der Toolkit ist innerhalb der letzten 7 bis 8 Jahre vor allem am Argonne National Laboratory⁵ entstanden. Im Moment ist die Version 3.2 aktuell, aber bereits in Kürze (geplant ist April 2005) soll Version 4 erscheinen. Die Entwicklung schreitet enorm schnell vorwärts, was auch der Unterstützung durch große Firmen (IBM, Intel, HP, Sun, Cisco, Microsoft etc) und Organisationen (NASA, DARPA, EU etc) zu verdanken ist. Der Globus Toolkit stellt eine Art Prototyp auf dem Gebiet des Grid-Computing dar und implementiert daher zahlreiche Standards des Global Grid Forums⁶, bevor andere Grid-Systeme dies tun. Durch die Standards des Global Grid Forums soll sichergestellt werden, dass auch kommerziell entwickelte Grid-Systeme eine gemeinsame Schnittstelle haben und sich so auch gegenseitig ergänzen können.

2.2 Globus Toolkit 2

Auch wenn im Moment die Version 3 des Globus Toolkit aktuell ist, soll hier dennoch auch die Vorgängerversion Erwähnung finden, da alle Funktionen dieses weiterhin verfügbar sind und die meisten Gridprojekte auf dem Globus Toolkit Version 2 (GT2) basieren. Diese geerbten Komponenten werden als *pre-web services* bezeichnet und sind ausschließlich unter UNIX Plattformen inklusive Linux lauffähig. Die *pre-web services* sind komplett in C implementiert, was sich auch anhand der Geschwindigkeit bemerkbar macht.

Die wichtigsten Komponenten sind:

- GRAM (Grid Resource Allocation and Management)
- MDS (Monitoring and Discovery Service)
- GridFTP

²<http://eu-datagrid.web.cern.ch/eu-datagrid/>

³<http://www.teragrid.org/>

⁴<http://www-unix.globus.org/toolkit/>

⁵<http://www-fp.mcs.anl.gov/division/welcome/default.asp>

⁶<http://www.ggf.org>

- GSI (Grid Security Infrastructure)

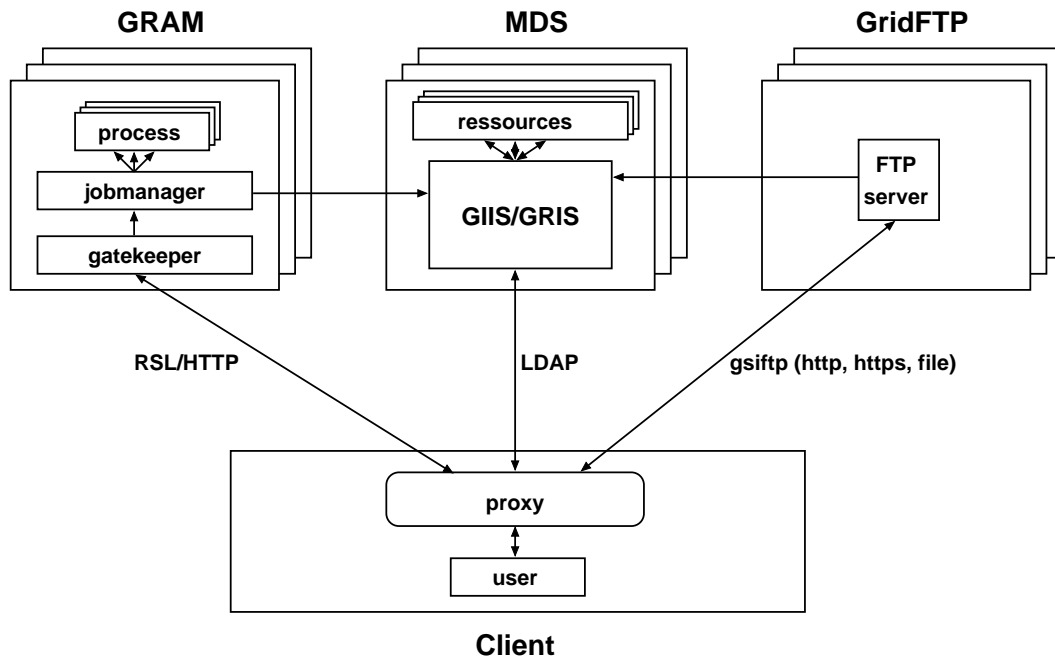


Abbildung 1: Bestandteile des Globus Toolkit 2

Das *GRAM*-Paket beinhaltet den *Gatekeeper*, der die Aufträge der Clients annimmt und einem *Job Manager* übergibt, der die Aufträge weiter verwaltet. Im Prinzip handelt es sich bei dem Jobmanager also um einen Scheduler. Der Standard-Jobmanager erzeugt einfach auf dem Node, welcher den Job annimmt, einen neuen Prozess. Durch den Einsatz von anderen Jobmanagern ist es möglich, Aufträge dynamisch im Grid zu verteilen, abhängig von Parametern wie der Auslastung der einzelnen Nodes oder den Anforderungen des Jobs.

Auch das Clientprogramm **globusrun**, welches zum Absetzen von Jobs an verschiedene Nodes des Grids dient, ist Bestandteil des *GRAM*. Die Jobs selbst werden in der *Resource Specification Language (RSL)* kodiert. In der *RSL*-Datei ist genau definiert, welches Programm mit welchen Parametern gestartet wird. Ebenfalls möglich sind die Angabe von Ein- bzw. Ausgabedatei sowie Einstellungen wie die maximal zu verwendende Menge an RAM. Die Ausgabedateien selbst werden durch den *Global Access to Secondary Storage (GASS)* zurückgeliefert.

Der *Monitoring and Discovery Service (MDS)* fasst den *Grid Resource Information Service (GRIS)* und den *Grid Index Information Service (GIIS)* zusammen. Der *GRIS* läuft auf jedem Node und sammelt Informationen zu allen Ressourcen des Nodes, die er dann dem *GIIS* zugänglich macht, welcher entweder auf jedem Node oder an einer zentralen Stelle im Grid läuft. Auf den *GIIS* kann dann mit den im Globus Toolkit schon enthaltenen Clientprogrammen zugegriffen werden. Dies geschieht unter Verwendung von *LDAP*⁷. Auf Basis der durch den *MDS* gewonnenen Daten können Clients dynamisch abhängig etwa von der momentanen CPU-Last entscheiden, auf welchem Rechner der nächste Auftrag laufen soll. Auch der Jobmanager kann bei Bedarf auf den *MDS* zugreifen und diese Informationen in die Verteilung der Aufträge einfließen lassen.

GridFTP bietet einen sicheren (verschlüsselten) und zuverlässigen FTP-Dienst innerhalb des Grids an. Es sind sowohl ein Server- als auch Clientprogramme enthalten. *GridFTP* unterstützt neben gewöhnlichen Transfers auch sog. *third-party file transfers*, bei denen Dateien zwischen zwei Rechnern kopiert werden, die Kontrolle jedoch durch einen dritten Rechner stattfindet, ohne dass die Daten über diesen laufen.

⁷<http://www.openldap.org/>

Die *Grid Security Infrastructure (GSI)* sorgt für eine sichere Kommunikation innerhalb des Grids durch die Verschlüsselung von `http` bzw. `rls` durch *SSL*⁸ (*Secure Socket Layer*). Ebenfalls enthalten sind Möglichkeiten zur Authentifizierung und Autorisierung der Benutzer und beteiligten Rechner im Grid durch Public-Key-Verfahren und Zertifikate. Alle oben aufgeführten Komponenten enthalten die Funktionen des GSI. In Kapitel 4 wird näher auf die Sicherheitsfunktionen des Globus Toolkit eingegangen.

2.3 Globus Toolkit 3

2.3.1 Grid Services

Mit der Version 3 wurde im Globus Toolkit das Konzept der *Web* bzw. *Grid Services* eingeführt. Der sog. *WS-Core* (*Web Services Core*) ist in Java implementiert. Daher besteht für die Web Services nicht mehr die Einschränkung auf UNIX-Varianten als Betriebssystem. Allerdings macht sich bei nicht ständig laufenden Anwendungen die Ladezeit der Java Virtual Machine deutlich bemerkbar. Mit Version 4 des Toolkit soll unter anderem aus diesem Grunde auch ein in C geschriebener *Web Services Core* eingeführt werden.

Unter einem Web Service hat man sich eine Schnittstelle vorzustellen, über die Clients mit einem Server kommunizieren. Dabei kommen aus dem Gebiet der RPCs bekannte Konzepte wie Client- und Server-Stubs zum Einsatz. Über Web Services kann ein Client bestimmte Aufgaben durch einen Server erledigen lassen oder Informationen von diesem beziehen. Da die Standards für Web Services allerdings nicht alle Anforderungen für ein Grid erfüllen, werden die Web Services im Globus Toolkit zu Grid Services erweitert, basieren jedoch noch auf den gleichen Grundlagen wie *SOAP*⁹ und *XML* zur Kodierung und *http* bzw. mit SSL-Verschlüsselung *https* zur Übertragung der Daten.

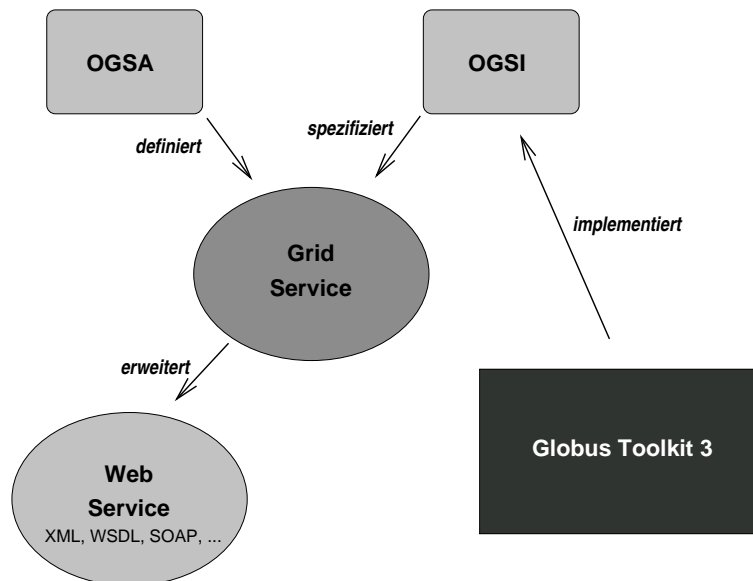


Abbildung 2: Grid Services im Globus Toolkit 3

Der Zusammenhang zwischen Grid Services, Web Services, den diversen Standards und dem Globus Toolkit wird in der Abbildung 2 verdeutlicht.

Durch die *Open Grid Services Architecture (OGSA)* werden die Anforderungen an Grid Services abstrakt beschrieben, die grundlegende Funktionen wie Sicherheit und Ressourcenverwaltung betreffen. Die genauen for-

⁸<http://www.openssl.org/>

⁹<http://www.w3.org/TR/soap/>

malen und technischen Details der Grid Services werden durch die *Open Grid Services Infrastructure (OGSI)* spezifiziert. Der Globus Toolkit selbst implementiert diese OGSI-Funktionen zu Grid Services, die wiederum eine Erweiterung der Web Services darstellen.

2.3.2 Komponenten des Globus Toolkit 3

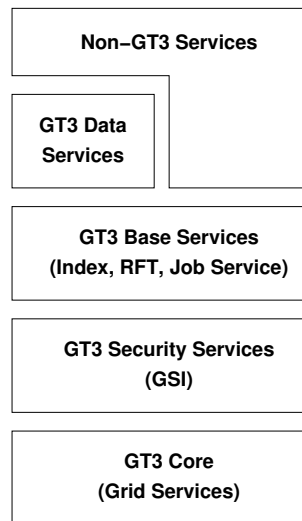


Abbildung 3: Komponenten des Globus Toolkit 3

Die Grid Services bilden den grundlegenden Teil des Globus Toolkit 3, den *Core*. Auf diesem setzen die *Security Services* auf, die sowohl für die sichere Übertragung der Daten als auch für Authentifizierung und Autorisierung verantwortlich sind. Dies geschieht durch den Einsatz von SSL und Public-Keys in Form von *X.509*¹⁰-Zertifikaten.

Die *Base Services* beinhalten eine Reihe von entscheidenden Komponenten: Zum einen den *Index Service*, der es ermöglicht, für eine bestimmte Problemstellung den passenden Web Service zu finden, unabhängig davon, an welcher Stelle im Grid dieser angeboten wird. Auch kann ein und derselbe Web Service auf verschiedenen Nodes mit unterschiedlichen Eigenschaften angeboten werden. Eine Rechnerarchitektur kann zum Beispiel für ein ganz bestimmtes Problem besser geeignet sein als eine andere. Diese Unterscheidungen sind durch den *Index Service* möglich. Der *Reliable File Transfer (RFT) Service* ermöglicht die zuverlässige Übertragung von großen Datenmengen zu einem Grid Service. Ebenfalls enthalten ist der *Managed Job Service*, der wie der Name schon andeutet für die Verwaltung der Aufträge verantwortlich ist. Die Möglichkeiten beschränken sich nicht nur auf das einfache Absetzen von Jobs und anschließende Sammeln der Ergebnisse, sondern es können auch Jobs angehalten und später fortgesetzt oder abgebrochen oder teilweise sogar von einem Node zu einem anderen verschoben werden.

Die *Grid Data Services* ermöglichen es durch das sog. *Replica Management* nur bestimmte Teile von sehr großen Datensätzen herauszunehmen und auch nur diese zwischen den Nodes zu übertragen. Dabei wird genau Buch darüber geführt, welche Teile gerade von welchem Node bearbeitet werden, so dass es nicht zu Inkonsistenzen kommen kann.

Auf, bzw. neben all diesen Komponenten des Globus Toolkit 3 können natürlich weitere Dienste laufen, die nicht zum Toolkit gehören, aber die Architektur und Infrastruktur mitbenützen.

¹⁰<http://en.wikipedia.org/wiki/X.509>

3 Benutzung des Globus Toolkit

3.1 Erstellung von Web Services für den GT3

Da eine ausführliche Anleitung zur Erstellung eines Web Services den Rahmen dieser Ausführung sprengen würde, wird im folgenden nur kurz auf die notwendigen Schritte eingegangen.

Das *Interface* eines Web Services wird in einer speziellen XML-Sprache mit dem Namen *Grid Web Services Description Language* (*GWSDL*) verfasst. Darin ist festgelegt, mit welchen Parametern die einzelnen Funktionen des Dienstes aufgerufen und welche Werte jeweils zurückgegeben werden.

Der Dienst selbst ist dann für gewöhnlich in Java implementiert, allerdings ist es auch möglich (geeignete Wrapper vorausgesetzt) andere Sprachen wie z.B. C zu verwenden.

Um den neu entwickelten Service durch den Web Server (den *Grid Service Container*) zugänglich zu machen, muss noch ein *deployment descriptor* im *WSDD*-Format (*Web Service Deployment Descriptor*) angelegt werden. Diese Datei beschreibt, unter welchem Namen und unter welchen Bedingungen der Service für Clients zugänglich ist.

Im nächsten Schritt werden die oben erzeugten Dateien zu einer Datei zusammengebunden. Dies geschieht unter Verwendung des Tools *ant*¹¹, welches eine Art *make* für Java darstellt. In diesem Schritt werden aus der *GWSDL*-Datei Stub-Klassen erzeugt und die Stubs sowie der eigentliche Service kompiliert. Dieses Paket enthält nun alle für die Serverseite notwendigen Dateien und wird dann mit Hilfe von *ant* auf dem Server installiert.

3.2 Arbeiten mit dem GT2

Es gibt im Prinzip zwei Möglichkeiten die Funktionen der Version 2 des Globus Toolkit zu verwenden, den Batchbetrieb und die Nutzung der C API.

3.2.1 Batchbetrieb

Anwendungen für den Batchbetrieb auf Basis des Globus Toolkit 2 sind komplett anders aufgebaut als die oben beschriebenen Web Services. Während es bei den Web Services darum geht, dem Nutzer eine möglichst einfache und abstrakte Schnittstelle zur Verfügung zu stellen ist der Ansatz beim GT2 eher, dem Benutzer Kommandozeilentools zu geben mit denen er selbst beliebige Programme auf einem oder mehreren Nodes des Grids steuern kann. Aus diesem Grund unterscheiden sich Serveranwendungen für den GT2 nicht von normalen Diensten auf einer UNIX-Plattform. Praktisch jedes Programm, welches per Kommandozeile bedienbar ist, kann somit im Grid eingesetzt werden. Die Aufträge werden dabei in der *Resource Specification Language* kodiert und an den bereits erwähnten *Gatekeeper* gesandt. Je nach verwendetem Clientprogramm und Konfiguration wird der Output des Auftrags entweder direkt zurückgeliefert oder zur späteren Abholung zwischengespeichert. Der Toolkit selbst bringt selbst etliche Kommandozeilenprogramme mit, mit denen sich zum Beispiel Informationen über den *MDS* abrufen oder Jobs verwalten lassen.

3.2.2 C API

Die zweite Möglichkeit ist die Verwendung der C API. Hier kann der Nutzer eigene Programme in C/C++ entwickeln und dennoch auf die Infrastruktur des Grid zugreifen. Es gibt Schnittstellen zu allen Funktionalitäten des Globus Toolkit wie der Jobverwaltung durch den *GRAM* oder den Informationen des *MDS*. Auch *GridFTP* kann sowohl server- als auch clientseitig eingebunden werden. Nicht zu vergessen ist auch die *Grid Security Infrastructure*, die man zur Autorisierung und Authentifizierung in eigenen Programmen verwenden kann. Es eröffnet sich also ein weites Feld zur Entwicklung eigener Gridanwendungen auf Basis der Funktionalität des Globus Toolkit.

¹¹<http://ant.apache.org/>

4 Sicherheit des Globus Toolkit

Eine der wichtigsten Eigenschaften des Globus Toolkit ist die Sicherheit, zusammengefasst unter der *GSI* (*Grid Security Infrastructure*). Das Sicherheitskonzept des Globus Toolkit basiert hauptsächlich auf einem Public-Key-Verfahren, welche in der *Public Key Infrastructure* (*PKI*) verankert ist.

Das Public-Key-Verfahren generiert für einen Nutzer einen öffentlichen und einem privaten (geheimen) Schlüssel. Die beiden Schlüssel sind im Prinzip gleichberechtigt: eine mit dem öffentlichen Schlüssel verschlüsselte Nachricht kann nur durch den privaten Schlüssel wieder entschlüsselt werden und umgekehrt. Dieses Verfahren ist natürlich nur sicher, solange der privaten Schlüssel wirklich nur dem Besitzer bekannt ist. Als zusätzliche Sicherheitsmaßnahme ist der private Schlüssel durch ein Passwort geschützt, welches vor jeder Verwendung angegeben werden muss.

Beim Public-Key-Verfahren besteht die Möglichkeit öffentliche Schlüssel zu *signieren*. Dadurch garantiert der Unterzeichner des Schlüssels, dass er die Identität des Schlüsselbesitzers geprüft hat. Beim Globus Toolkit werden die öffentlichen Schlüssel in einem dem Standard X.509 entsprechenden Zertifikat gespeichert. In dem Zertifikat sind neben dem Schlüssel auch noch eindeutige Informationen über den Besitzer des Schlüssels sowie den Unterzeichner enthalten.

Will ein Nutzer Zugang zu einem bestehenden Grid erhalten, so muss er sich zunächst ein Zertifikat generieren und dieses dann an die *Certificate Authority* (*CA*) des Grids zur Signierung schicken. Die CA prüft dann, ob der Nutzer berechtigt ist, das Grid zu nutzen und prüft seine Identität. Bei Erfolg dieser Prüfungen signiert die CA das Zertifikat des Nutzers mit dem geheimen Schlüssel der CA und sendet das Zertifikat an den Nutzer zurück. Anhand der Signatur können die Nodes des Grid feststellen, dass die Identität des Users durch die CA geprüft wurde und der User zur Nutzung der Ressourcen berechtigt ist.

Die Authentifizierung läuft beim Globus Toolkit normalerweise über einen Proxy ab, den der Benutzer lokal auf seinem Rechner startet und der dann die Authentifizierung allen Nodes gegenüber durchführt. Auf diese Weise muss der Benutzer nicht für jeden Job den er absetzt seinen privaten Key neu freischalten. Er autorisiert den Proxy dadurch, dass er den öffentlichen Schlüssel des Proxys unterschreibt und dem Proxy sein Zertifikat zur Weitersendung gibt. Dafür gibt es das Kommando **grid-proxy-init**, welches einen temporären Proxy erzeugt, nachdem es die entsprechenden Schlüssel generiert bzw. signiert hat.

Die Autorisierung erfolgt beim GT über eine sogenannte *gridmap* Datei. In dieser Datei werden Nutzer, die Zugriff auf eine bestimmten Ressource auf einem Node haben sollen auf lokale User des Node gemappt. Durch die *PKI* ist sichergestellt, dass ein Nutzer, der eine Anfrage stellt, auch wirklich er selbst ist. Daraufhin findet eine Überprüfung statt, ob der User in der *gridmap* aufgeführt ist. Gegebenfalls erhält der Nutzer dann Zugriff auf die Ressourcen des Node.

Listing 1: Beispiel für eine *gridmap*

```
"/O=Grid/OU=Globus Test/OU=Globus Test CA/CN=Globus Toolkit Admin"      globus
"/O=Grid/OU=Globus Test/OU=Globus Test CA/CN=Severin Strobl"           severin
```

Der erste Eintrag ist dabei jeweils der Distinguished Name (DN) aus dem Zertifikat, also eine eindeutige Bezeichnung für den User, der zweite der lokale User auf den der Besitzer des Zertifikats gemappt wird. Dabei können auch mehrere Zertifikate auf einen User gemappt werden.

Die Sicherung der Kommunikation innerhalb des Grids erfolgt bei den Diensten der Version 2 durch SSL-Verschlüsselung der Protokolle (*http*, *ftp*, etc). Seit Version 3 wird nicht mehr die ganze Kommunikation zwischen Server und Client verschlüsselt, sondern nur noch die eigentlichen Werte im *SOAP*-Protokoll.

5 Bemerkungen

Der Globus Toolkit stellt ein enorm mächtiges und umfangreiches Werkzeug dar. Die Entwicklung schreitet extrem schnell voran, neue Möglichkeiten werden fast täglich implementiert. Dort liegt aber auch das in meinen

Augen größte Problem des Toolkits: Durch die schnelle Weiterentwicklung veraltet die einmal geschriebene Dokumentation sehr schnell. Es ist fast unmöglich, wirklich aktuelle Informationen über alle Bestandteile des Toolkits zu bekommen. Das ist auch ein Grund für die alles andere als triviale Installation. Selbst wenn die Grundinstallation glatt läuft, benötigt man noch zahlreiche Erweiterungen um ein wirklich funktionierendes Grid zu erhalten. Die Funktionalitäten sind sehr weit gefächert, teilweise existieren verschiedene Modelle nebeneinander (vgl. GT2- und GT3-Funktionen), ohne dass eine klare Trennung erkennbar ist. Wenn man den Globus Toolkit ernsthaft einsetzen will, muss man sich darauf gefasst machen, zunächst einmal viel Dokumentation und diverse Howtos zu wälzen, bis man die für einen bestimmten Einsatzzweck sinnvollen Funktionen findet. Trotz alledem ist das Konzept des Toolkits auf jeden Fall sehr interessant und eine durchaus spannende Entwicklung.

Literatur

- [1] Ian Forster. What is the Grid? A Three Point Checklist.
URL: <http://www-fp.mcs.anl.gov/~foster/articles/WhatIsTheGrid.pdf>
- [2] Homepage des Globus Projekts
URL: <http://www.globus.org/>
- [3] Homepage des Globus Toolkit
URL: <http://www-unix.globus.org/toolkit/>
- [4] IBM RedBooks. Introduction to Grid Computing with Globus.
URL: <http://www.redbooks.ibm.com/abstracts/sg246895.html?Open>
- [5] IBM RedBooks. Globus Toolkit 3.0 Quick Start.
URL: <http://www.redbooks.ibm.com/abstracts/redp3697.html?Open>
- [6] IBM RedBooks. Enabling Applications for Grid Computing with Globus.
URL: <http://www.redbooks.ibm.com/abstracts/sg246936.html?Open>
- [7] Borja Sotomayor. The Globus Toolkit 3 Programmer's Tutorial.
URL: <http://www.casa-sotomayor.net/gt3-tutorial/>