

Grid-Computing: Technologien und Trends

Carsten Borngräber

17. Juni 2004

Technische Universität Braunschweig
Institut für Betriebssysteme und Rechnerverbund

Studienarbeit

Grid-Computing: Technologien und Trends

von
Carsten Borngräber

Aufgabenstellung und Betreuung:
Prof. Dr. S. Fischer und Dipl. – Inform. Christian Werner, Dipl. – Winform. Ulrich Abelmann

Braunschweig, den 17. Juni 2004

Erklärung

Ich versichere, die vorliegende Arbeit selbstständig und nur unter Benutzung der angegebenen Hilfsmittel angefertigt zu haben.

Kurzfassung

Das Ziel dieser Studienarbeit ist es, den Begriff „Grid Computing“ dem Leser näher zu bringen.

Hierbei wird in der vorliegenden Arbeit das Thema in 2 Abschnitte eingeteilt.

In dem ersten Abschnitt wird ein historischer Überblick über die Verwandtschaft von Grid und Cluster Computing gegeben und eine mögliche Definition des Begriff „Grid Computing“ aufgezeigt.

Zudem werden in diesem Abschnitt verschiedene Anwendungstypen mit deren Eigenschaften aufgezeigt, um dem Leser den Aufbau und die Funktionsweise eines Grid Systems darzustellen.

In dem zweiten Abschnitt dieser Arbeit wird ein kurzer Überblick über die wichtigsten Grid Technologien gegeben, sowie ein Ausblick auf die verschiedene Grid Konzepte bzw. Projekte aufgezeigt.

Aufgabenstellung

Inhaltsverzeichnis

Abbildungsverzeichnis	X
Tabellenverzeichnis	X
Glossar	XI
1. Einleitung	1
2. Das Grid Computing Konzept	4
2.1 Definition von Grid Computing	4
2.2 Einsatzbereiche von Grid Systemen	6
2.2.1 Rechen-Grids	7
2.2.2 Data-Grids	9
2.2.3 Kollaborative Computerarbeit	10
2.3 Der stufenweise Aufbau eines Grids	12
3. Architektur eines Grid Systems	15
3.1 Aufbau eines Grid	15
3.2 Sicherheit im Grid System	19
3.3 Die Ressourcentypen und das Ressourcenmanagement in einem Grid Computing	22
4. Grid Toolkits	26
4.1 Überblick über verschiedene Implementierungen	26
4.2 Das Globus Toolkit	28
4.3 Sun Grid Engine	30
5. Anwendungsbeispiele für Grid Computing	32
5.1 Seti@Home	32
5.2 Distributed.net - Projekt	36
5.3 LHC – Computing – Grid – Projekt	38
6. Ausblick	40
7. Zusammenfassung	42
Literaturverzeichnis	43

Abbildungsverzeichnis

Abbildung 1 Rechner und Grid als eine virtuelle Rechenplattform.....	5
Abbildung 2 Evolution der Grids	13
Abbildung 3 Grid Architektur	16
Abbildung 4 Anwendungen, Jobs und Subjobs	24
Abbildung 5 Architektur des Globus Toolkits	29
Abbildung 6 Datenübertragung vom Radioteleskop zum Client	34
Abbildung 7 Empfang und Bearbeitung der Datenpakete in dem Server – Komplex	35
Abbildung 8 Geplantes LHC-Netzwerk am CERN	39

Tabellenverzeichnis

Tabelle 1 Gesamtstatistik bei SETI@HOME, Stand: 22.05.04	33
--	----

Glossar

ADS	Active Directory Service ist ein Verzeichnisdienst der nur von Microsoft unterstützt wird.
Beowulf	Beowulf war der Name des ersten bekannten High Performance Clusters.
Distributed File System	Das Distributed File System (DFS) ermöglicht den Zugriff auf entfernte Ressourcen über ein Netzwerk.
Global Grid Forum	Ist ein offener Prozess für die Entwicklung von Standards und Empfehlungen für den Bereich Grid Technologien.
IPv6	Ipv6 ist die nächste Generation des IP-Protokolls, das zur Bildung von IP – Nummern 128 Bit zur Verfügung stellt.
Middleware	Middleware ist eine Menge von wenig spezialisierten Diensten, die zwischen der Systemplattform (Betriebssystem) und den Anwendungen angesiedelt ist und deren Verteilung unterstützen
MPI	Message Passing Interface: Ist ein Programmierschema, das bei parallelisierten Programmen zur Synchronisierung von Programmteilen durch Nachrichtenaustausch zum Einsatz kommt.
Network File System	Das Network File System ist das bekannteste DFS unter UNIX
Peer to Peer (P2P)	In P2P Netzwerken tauschen gleichberechtigte Arbeitsstationen (peers) Daten aus und arbeiten an verteilten Anwendungen.
PVM	Parallel Virtual Machine: Ist eine Programmierumgebung, die mehrere heterogene Rechner wie eine virtuelle Maschine erscheinen lässt und verteilte Programmierung unterstützt
VO	Virtual Organisation: Organisationsform im Grid, die durch den Verbund von den verfügbaren Ressourcen entsteht. Ein Grid kann dabei mehrere VO's enthalten.
Web Service	Web Service ist eine Schnittstelle für den netzbasierten Zugriff auf eine Anwendungsfunktionalität.

1. Einleitung

Einer der wichtigsten Erfolgsfaktoren in den Bereichen der Wissenschaft, Forschung und Entwicklung wird der schnelle und standortunabhängige Zugriff auf global verteilte Daten und IT – Ressourcen sein. Die Basis bildet hierfür eine komplexe Infrastruktur, die aus vernetzten Informationen mit unterschiedlichen Administrations- und Zugangsmechanismen besteht. Diese Infrastruktur wird allgemein als „Grid Computing“ bezeichnet.

Die Hauptaufgabe dieser Struktur ist es, komplexe Problemstellungen zu vereinfachen, in dem Ressourcen zur Verfügung gestellt bzw. Daten oder Netzdienste angeboten werden.

Diese Sachverhalte verfügen jedoch schon über mehrere Einzellösungen. Dabei bietet das Grid Computing aufgrund der Infrastruktur einerseits den Zusammenschluss unterschiedlicher Technologien und andererseits eine Verbesserung der Leistungsfähigkeit eines Systems.

Historischer Überblick über Cluster- und Grid Computing

Die Idee, ein Verbund von Computern für eine bestimmte Arbeit zu entwickeln, entstand schon in den 60' er Jahren. Der erste Rechner war auf dem Gebiet der parallelen Arbeit der Control Data Corporation (CDC) 6600 und wurde 1964 entwickelt.

Auch die amerikanische Luftwaffe hatte für ein Frühwarnsystem namens „SAGE“ mehrere separate Computersysteme zusammengeschlossen, um feindliche Flugzeuge über den nordamerikanischen Kontinent zu entdecken.

Der eigentliche kommerzielle Durchbruch auf dem Gebiet der Supercomputer und Cluster wurde in den 70' er Jahren verzeichnet. Der ehemalige Entwickler von CDC, Seymore Cray, entwickelte 1976 mit Cray 1, den ersten Vektorrechner mit einer Gesamtleistung von 160 MFlops (Million FLoatingpoint OPERations per Second)¹, einer Leistung von ca. 130 MFlops bei Matrizenmultiplikation und mit eine Speichergröße von 8 MB. In Deutschland kam 1979 dieser Rechner im Max – Planck – Institut für Berechnungen auf dem Gebiet der Plasmaforschung zum Einsatz.

Bis etwa Mitte der achtziger Jahre wurden die „Supercomputer“ ausschließlich nur im Bereich der Forschung, für komplexe Rechenaufgaben, verwendet.

Etwa 1985 wurde der Begriff „Cluster“ durch die Digital Equipment Corporation (DEC) geprägt. Das Cluster bestand aus vernetzten VAX-Minicomputern und wurde erstmals als

¹ Maß für die Leistungsfähigkeit eines Computers bzgl. Fließkommaoperationen.

Produkt auf dem Markt angeboten. Dabei führte eine Verifizierung dieses Clusters zu einer Verbesserung des lokalen Hochgeschwindigkeitsnetzes und zu einer zentralen Steuereinheit, die in der Lage war, mehrere Ressourcen zu verwalten.

1991 kam man erstmal auf die Idee, Rechenoperationen so zu steuern, dass beispielsweise rechenintensive Aufgaben auf mehrere Rechner gleichzeitig aufgeteilt wurden. Hierfür wurde innerhalb eines Softwareprojektes, das unter der Aufsicht von Vaidy Sundaram stand, die „Parallel Virtual Machine (PVM)“ entwickelt. Dabei wurden unterschiedliche Workstation Cluster kabellos miteinander zu einem virtuellen Parallelrechner verbunden.

Darüber hinaus wurde 1992 von einer Arbeitsgruppe das so genannte Message – Passing – Interface (MPI) vorgestellt. Das MPI ist eine interessante Alternative zum PVM und ist eine portable Parallelprogrammierung.

Das PVM und das MPI wurden als Standard im Bereich des Clusters bezeichnet und waren somit ausschlaggebend für verschiedene Clusterprojekte. Beispielsweise entstand 1993 das erste „Network of Workstation (NOW), ein Cluster, das ein Projekt der UC Berkeley war.

1994 wurde der erste PC – Cluster am Goddard Raumflugzentrum der NASA installiert.

Dieser war für typische Rechenoperationen der Erd- und Weltraumwissenschaft zuständig. Da zu dieser Zeit herkömmliche Supercomputer für diese Anforderung über eine Million Dollar kosteten, hat sich der Forscher (Dr. J. R. Sterling) für einen PC- Rechencluster entschieden, dem so genannten „Beowulf“.

Weiterhin wurde Mitte der neunziger Jahre aufgrund verschiedener Projekte der Grundstein für das Grid geschaffen. Das wichtigste Projekt, das Open Source Projekt besser bekannt als Globus wurde von der Firma Genias aus Regensburg durchgeführt. Das Globus dient daher als Basis für alle weiteren Open Source Grid Projekte.

Jedoch wurden nicht nur verschiedene Projekte ins Leben gerufen, sondern auch immer mehr Firmen interessierten sich für die neuartige Technologie. So kaufte die Firma Sun Microsystems im Juli 2000 die Firma Gridware (vorher Firma Genias) auf und entwickelte mit ihrer Hilfe eine neue Software, die Sun Grid Engine (SGE) für das Solaris Betriebssystem, welche auch später für das Linux – Betriebssystem erhältlich war.

Die neue Software hat sich aufgrund der Veröffentlichung des gesamten Codes recht schnell ab dem Jahr 2001 etabliert.

In den Folgejahren ist daher die Nachfrage an Grid Computings sehr stark angestiegen. So sind in kürzester Zeit mehrere Projekte, wie beispielsweise das EU DataGrid Projekt im Januar 2001 oder das US TeraGrid Projekt im August 2001, gestartet.

Darüber hinaus wurde, aufgrund des starken Interesses von Unternehmen an den Grid System und den Technologien, im Jahr 2003 das Globus Toolkit 3.0 mit der Grid Service Technologie fertig gestellt und somit die erste Implementierung abgelöst. Der Grundstein scheint somit für die Einsatzmöglichkeit von Grid System in den unterschiedlichen Unternehmen gelegt zu sein.

2. Das Grid Computing Konzept

In diesem Kapitel wird das Grundkonzept und Grundbegriffe von Grid Computing beschrieben. Dabei wird zunächst der Begriff „Grid Computing“ in seiner Definition dargestellt und sein Ursprung näher betrachtet. Des Weiteren wird eine Kategorisierung der Anwendungstypen vorgenommen und eine mögliche Entwicklung von Grid Computing aufgezeigt

2.1 Definition von Grid Computing

Im Laufe der Entwicklung von Grid Computing sind etliche Definitionen entstanden. Die nachfolgende Variante von Ian Foster und Carl Kesselman beschreibt die wichtigste Hauptfunktion eines Grid sehr gut, da diese das gemeinsame und koordinierte Nutzen von Ressourcen über Unternehmensgrenzen hinweg behandelt.

„The real and specific problem that underlies the Grid concept is coordinated resource sharing and problem solving in dynamic, multi-institutional virtual organizations.“²

Der Begriff „Grid Computing“ ist aus dem englischen Wort „electrical power-grid“ abgeleitet. Er entspricht der Struktur einer flächendeckenden Energieversorgung.

Unter einem Grid-Computing wird eine umfassende Architektur verstanden, welche die Koordination von IT-Ressourcen auf alle Ebenen und über deren geographische Einheiten hinweg übernimmt und diese zu einem Netzwerk verbindet.

Das Ziel ist es, eine möglichst sichere, kostengünstige, konsistente und transparente Nutzung dieser Ressourcen zu ermöglichen.

Zu den Rechner-Ressourcen zählen zum Beispiel: Soft- und Hardware, Supercomputer, Vektor-, Parallelrechner und Cluster.

Hieraus ergeben sich hohe Anforderungen an die Sicherheitsaspekte bei der Vergabe von Sicherheitszertifikaten und an die Qualität aller angebotenen Dienste.

Um diese Anforderungen zu erreichen, benötigt man eine Verbesserung der Standards im Bereich der Software und Technologien, wie Middleware, Web Services, IPv6 mit erweiterten Qualitäts- und Abrechnungs-Mechanismen, als auch eine Verbesserung im Hardwarebereich.

² Vgl. Ian Foster et al. , „The Anatomy of the Grid“
<http://www.globus.org/research/papers.html#anatomy>

Anders als die bisherigen Supercomputer, die an einzelnen Standorten betrieben werden, stellen Grids eine hohe Rechenleistung dar, die durch die Integration beträchtlicher Pools hervorgerufen werden.

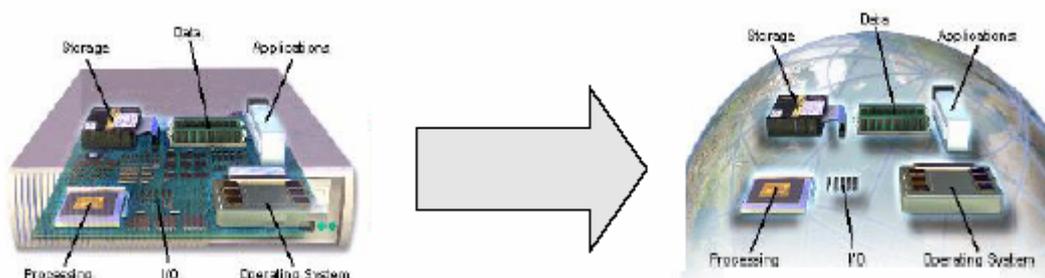


Abbildung 1 Rechner und Grid als eine virtuelle Rechenplattform ;[10]

Das bedeutet das jeder Client, der in dem Netzwerk angeschlossen ist, in einer Art Peer to Peer (2) Computing von den ungenutzten Ressourcen aller Clients profitiert und so eine Rechenleistung erzeugt, die weit aus höher ist, als die der eigenen Maschine.

Man spricht hierbei von einem Zusammenschluss der einzelnen Rechner zu einer virtuellen Organisation (VO), wo unter klaren Sicherheitsbestimmungen Regeln über die Mitgliedschaft und Benutzung von individuellen Ressourcen und Institutionen innerhalb der Gruppe definiert sind. Es entsteht somit ein virtueller Supercomputer, der für den Benutzer transparent erscheint und wie ein lokales System benutzt werden kann.

Weiterhin besteht in einem Grid System kein Single Point of Failure, d.h. bei einem Ausfall einer Grid Ressource in der virtuellen Organisation übernimmt eine andere deren Arbeit. Die Arbeit selbst wird meist durch eine Middleware auf die verschiedenen Ressourcen verteilt. So entsteht eine transparente Verteilung für den Grid-User (Verteilungstransparenz).

Die hier dargestellten Eigenschaften charakterisieren ein verteiltes System. Daher entspricht ein Grid System ebenfalls einem verteilten System.

Um die Dominanz eines Grid Systems weiter zu untermauern, wird hier ein Vergleich im Bereich verteilter Systeme zwischen zwei anderen benutzten Systemen (Cluster und Web) und dem Grid System vorgenommen. Hierbei zeigt sich eindeutig, dass weder das Web noch das Cluster Computing in der Lage ist, das beschriebene „Grid-Konzept“ zu lösen.

- **Cluster Computing** (Einheitliche Systemdarstellung bzw. Computerressource)
 - Homogenen verbundenen Knoten bilden eine Plattform
 - Eine administrative Domäne

- **Web - Rechnernetz** (ohne einheitliche Systemdarstellung bzw. Computerressource)
 - Rechner- und Kommunikations-Ressourcen bilden eine heterogene Menge
 - Mehrere administrative Domänen
- **Grid Computing** (Einheitliche Systemdarstellung bzw. Computerressource)
 - Die verteilten heterogenen Ressourcen bilden eine Gruppe
 - Mehrere administrative Domänen

Eine eindeutige Definition von Grid Systemen ist jedoch auf Grund des ständigen Wandels, der Architektur, sowie wegen der raschen Weiterentwicklung im Bereich der Technologie nicht möglich.

2.2 Einsatzbereiche von Grid Systemen

Im Bereich der Forschung und Entwicklung ist es wichtig sich über den Aufwand, die Schwierigkeiten, die Eigenschaften und die Möglichkeiten des zu entwickelnden Systems im Klaren zu sein. Dies gilt ebenso bei der Entwicklung für einen bestimmten Einsatzbereich von Grids.

Auch hier ist es wichtig die Möglichkeiten zu kennen und das der Aufwand gegenüber dem Nutzen in einem guten Verhältnis steht.

Obwohl es Grids derzeit nur als Pilotprojekte gibt, so kann man schon 3 Einsatzbereiche unterscheiden, für die Grids in Zukunft eingesetzt werden könnten:

- Verteiltes Supercomputing (Rechen Grid → engl. Computational grid)
- Data-Grids, d.h.
- Kollaborative Computerarbeit

Diese Unterscheidung ist sicherlich fließend und in der Praxis gibt es Systeme, die diese Formen und Eigenschaften bereits in sich vereinen. Daher sind die Grenzen nicht klar abgesteckt und man wählt jene Art, welche die Gesamtlösung am genauesten definiert.

Die Wahl des Einsatzbereiches ist ausschlaggebend bei der Entwicklung einer Software, die innerhalb eines Grid Systems implementiert werden soll, da sie bestimmte Entscheidungen, die für die Weiterentwicklung notwendig sind, beeinflusst.

2.2.1 Rechen-Grids

Mit diesem Begriff ist die Zusammenfassung der Rechenleistung mehrerer Rechner gemeint, wie man es bereits von Clustern her kennt.

In den nachfolgenden Abschnitt werden einige Eigenschaften des Rechen-Grids näher beschrieben.

Ausnutzung nicht verwendeter Ressourcen

Das Rechen-Grid greift auf ungenutzte Computing Ressourcen innerhalb einer Organisation oder über deren Grenzen hinweg zu, um wie ein virtueller Supercomputer zu funktionieren.

So kann es beispielsweise sein, dass die CPU eines Rechners durch eine Aufgabe maximal ausgelastet ist. Sollten nun weitere Aufgaben an diesen PC geschickt werden, die jedoch nicht bearbeitet werden können, so kann man mittels Grid Computing diese Aufgaben an andere freie Ressourcen, die bestimmte Rechner innerhalb des Grids zur Verfügung stellen, übertragen.

Hierbei sind einige Vorbedingungen an den entfernten Rechner zu beachten:³

- Der Rechner muss diese Aufgabe mit einem geringen Aufwand erledigen.
- Die für diese Aufgaben benötigte Hardware, Software und Ressource(n) muss der angesprochene Rechner im Grid zur Verfügung stellen

Trotz der Zusammenfassung von Rechenleistungen sind nicht alle Aufgaben für verteiltes Supercomputing gleichermaßen geeignet. Daher ergeben sich folgende Anforderungen an einer geeigneten Aufgabe:⁴

- die Aufgabe muss gut unterteilbar (parallelisierbar) sein
- keine rechenintensive Aufgabe, die den einzelnen Rechner überfordert.
- Es sollte eine möglichst in sich geschlossene Aufgabe sein (kein komplex zu konfigurierendes Programm).

³ Vgl. Ralf Grohmann und Dr. Sebastian Wedeniwski; Grid Computing, Die IBM ZetaGRID Lösung,

⁴ Vgl. Ralf Grohmann und Dr. Sebastian Wedeniwski; Grid Computing, Die IBM ZetaGRID Lösung,

Parallele Rechenleistung

Eine weitere Eigenschaft des Rechen-Computing ist die parallele Nutzung von CPU - Kapazitäten. Diese Eigenschaft ermöglicht einem Grid System eine sehr hohe Rechenleistung zu erzeugen, indem die verschiedenen Grid Knoten mit den jeweiligen CPU - Kapazitäten gleichzeitig für die Lösung einer Aufgabe verwendet werden.

Die parallele Rechenleistung hängt stark von der Komplexität der Anwendung ab. Sie muss die Möglichkeit bieten, dass der Job einer Applikation in mehrere Subjobs aufteilbar ist, von denen jeder einzelne auf verschiedene Knoten im Grid System abgearbeitet werden kann.

Die Skalierbarkeit einer Applikation nimmt zu, wenn die Unteranwendungen untereinander völlig unabhängig sind und wenn diese, bei Rückgabe des berechneten Ergebnisses, nicht mit einander kommunizieren müssen. Eine perfekte Skalierbarkeit liegt dann vor, wenn eine Applikation 10mal schneller ist, wenn die Ausführung auf 10 Knoten stattfindet.

Diese Angabe ist jedoch nur ein theoretischer Wert, da im Bereich der Skalierbarkeit noch anderen Faktoren zu berücksichtigen sind.

Diese Faktoren liegen beispielsweise dann vor, wenn

- der verwendete Algorithmus nur eine begrenzte Aufteilung von Subjobs zulässt, was zu Zeitverlusten bei bestimmten Aufgaben führen kann.
- einzelne Subjobs von einander abhängig sind. Dies kann zu Warteschlangen führen, wenn einerseits Lese- und Schreibzugriffe auf einzelne Ressourcen (z.B.: Datenbanken) ausgeführt werden müssen oder andererseits Nachrichten untereinander ausgetauscht werden.
- bei Problemen, die sich innerhalb der Grid-Infrastruktur ergeben. Beispielsweise kommt es zu Verzögerungen bei Nachrichtenübertragungen zwischen den Jobs, durch hohe Zugriffszeiten, beschränkter Bandbreite oder durch Engpässe im Speicherbereich.

Trotz dieser Faktoren zeichnen die positiven Eigenschaften das Grid Computing aus, so dass dieses nicht nur in den wissenschaftlichen Bereichen, sondern auch in den Anwendungsgebieten, wie Filmanimation und Biomedizin seine Zustimmung findet.

2.2.2 Data-Grids

Ein Data Grid erlaubt den entfernten und sicheren Zugang zu aktuellen Daten. Hierbei wird dem Benutzer und den Applikationen die Möglichkeit bereitgestellt, auf entfernte Datenbanken effizient zuzugreifen und diese zu verwalten. Diese Datenbanken können permanente oder auch temporäre Speicherplätze sein.

Ein temporärer Speicherplatz (RAM-Speicher) wird für einen schnellen Datenzugriff verwendet. Im Gegensatz hierzu dient ein permanenter Speicherplatz dazu, die Speicherkapazität und die Performance zu steigern.

Weiterhin bietet Data Grids die Möglichkeit, durch Replikation von Daten eine Verbesserung der Performance des Systems zu erreichen.

Der Hauptzweck der Replikation besteht darin, bessere Datenübertragungsgeschwindigkeiten dadurch zu erreichen, dass die Daten von lokal „gecachten“ Datenbanken abgerufen werden, anstatt einzeln über das gesamte Netzwerk übertragen zu werden.

Deshalb benutzen Grid Systeme oftmals so genannte Mountable Network File Systeme, d.h. solche Dateisysteme, die das Ein- und Aushängen von Verzeichnissen via Netzwerk ermöglichen. Das Network File System (NFS) oder das Distributed File System (DFS) sind einige Beispiele aus der Vielzahl von solchen Dateisystemen.

Des Weiteren dient nicht nur das Data Grid zur Verbesserung der Performance, sondern ermöglicht es auch, ein überflüssiges und unnötiges Verschieben von Daten zu vermeiden, um damit eine Kostenreduzierung im IT-Bereich durchzuführen.

In vielen Forschungseinrichtungen von Wirtschaft und Wissenschaft wird sehr stark an Lösungen in diesem Bereich gearbeitet. Zudem werden heute auf dem Markt Data Grids Lösungen im Bereich Geschäftapplikationen von Softwarehäusern und großen IT-Unternehmen angeboten.(siehe Kapitel 5 „Anwendungsbeispiele für Grid Computing)

Einen weiteren wichtigen Ansatzpunkt innerhalb des Data Grids stellt die Sicherheitspolitik des Grids dar. Die Data Grids verlassen sich auch wie die Rechen Grids auf Sicherheits- und Benutzungsrichtlinien und können innerhalb einer administrierten Domäne oder über mehrere Domänen hinweg aufgebaut werden. Bei Nutzung übergreifender Domänen kommen zusätzliche weitere Aspekte und Benutzungsrichtlinien zum Tragen. (ADS –Trust)

2.2.3 Kollaborative Computerarbeit

Kollaboration sieht ihr Hauptziel in der Interaktion von Mensch zu Mensch. Solche Applikation kann einen Verbund von heterogene und geographisch weit voneinander entfernte Ressourcen zu einer so genannten virtuellen Organisation (VO) herstellen, welche für die gemeinsame Nutzung von Datenarchiven oder zur Ausführung von Simulationen verwendet werden kann. Eine virtuelle Organisation ermöglicht also einen von Ressourcen-Anbietern und –Konsumenten klar definierten, in hohen Massen kontrollierten und in sich geschlossenen Austausch von Ressourcen.⁵

Zugriff auf andere Ressourcen

Der Ressourcenaustausch findet nicht nur zwischen den Dateien statt, sondern dieser bezieht sich ebenso auf den direkten Zugriff auf entfernte Großrechner (Hardware und Software), die für Lösungen von vernetzten Rechnern benötigt werden.

Weiterhin besteht für Mitglieder einer virtuellen Organisation die Möglichkeit, die benötigten Daten in Form von Dateien oder Datenbanken bereitzustellen. Diese Art der Bereitstellung wird aufgrund der Größe der Datenbanken wieder als so genanntes „data grid“ bezeichnet. Es findet in einem „data grid“ im Zusammenhang mit einer VO der Ressourcenaustausch nicht nur mit Daten statt, sondern es werden auch die Ressourcen geteilt.

Einige Beispiele dazu sind:

- Software/Lizenz: Die Rechner nutzen hier teure Softwarepakete, die auf andere Rechner installiert sind und zur Verfügung gestellt werden. Hierbei findet ebenso eine effizientere Nutzung der Lizenzen statt (unterschiedliche Lizenzmodelle der Hersteller sind zu berücksichtigen)
- Hardware: Hier besteht die Möglichkeit beispielsweise aufgrund der Bandbreite beim Internetzugang eine Verteilung des Zuganges auf mehrere Rechner zu ermöglichen. (Firewallkonzept ist zu berücksichtigen)

⁵ Vgl. Ian Foster et al., „The Anatomy of the Grid“
<http://www.globus.org/research/papers.html#anatomy>

Ressourcen Balancing

Das Grid System ist ebenfalls dafür ausgelegt, eine große Anzahl von Ressourcen, die von den einzelnen Rechnern zur Verfügung gestellt werden, zu einer großen virtuellen Ressourcensammlung zu vereinigen. In der virtuellen Umgebung ist das Grid in der Lage einen Ressourcen Balancing für die Applikation bereitzustellen.

Das bedeutet, dass bei der Jobverteilung im Grid eine gleichmäßige Auslastung aller beteiligten Komponenten erreicht wird. Diese Eigenschaft kann von großer Bedeutung sein, wenn einzelne Rechner mit unterschiedlicher Konfiguration installiert sind.

Hierfür bestehen zwei Lösungsansätze, um diese ausgewogene Belastung zu erreichen:

- Eine aufwendige Anfrage kann von einem schon belasteten Rechner selbst an einen unbelasteten Rechner im Grid weitergeleitet werden
- Ist ein Rechner für längere Zeit belastet, so werden die Jobs mit der geringere Wertigkeit temporär gestoppt oder komplett gelöscht, so dass die Jobs mit der höheren Priorität ausgeführt werden können. So verfügt man für diese Aufgaben mit hoher Wertigkeit immer über genügend viele Ressourcen.

Diese Eigenschaft erweist sich ebenso für zeitlich gebundene Aufgaben als nützlich. Denn bei zeitlichen Engpässen kann auf diese Weise der Job in Subjobs aufgeteilt werden und durch die unbelasteten Rechner bearbeitet werden, so dass noch die terminliche Vereinbarung über die Fertigstellung eingehalten wird.

Zuverlässigkeit

Übliche High-End-Computersysteme verwenden meist sehr teure Hardwarekomponenten, um das Ausfallrisiko zu minimieren und damit die Zuverlässigkeit der Systeme zu erhöhen.

Beispielsweise nutzt man mehrere Prozessoren, die bei laufendem Betrieb leicht und ohne das Ausschalten des Rechners, ausgetauscht werden können.

Desweiteren werden oft für diese Systeme ein eigenes Stromaggregat und mehrere Kühlsysteme verwendet.

Bei einem Grid System ergibt sich diese Vorkehrung nicht, da ein gewisser Grad an Redundanz schon vorhanden ist.

Falls ein Rechner aufgrund eines Absturzes ausfällt, wird eine so genannte Gridmanagement Software (autonomic computing) genutzt, die automatisch den anstehenden Job auf andere Ressourcen verteilt.

Weiterhin wird die Zuverlässigkeit in einem Grid dadurch erhöht, dass ein Job auf mehrere unabhängige Rechner gesendet wird, um eine sichere Ausführung des Jobs zu gewährleisten. Hierbei ergibt sich zusätzlich die Möglichkeit, dass die Ergebnisse miteinander verglichen werden können, um eine mögliche Fehlerquelle zu erkennen.

Die Zuverlässigkeit in einem Grid System wird daher nicht durch spezielle Hardware erreicht, sondern wird durch den Einsatz von der „autonomic computing“ Software erzielt.

2.3 Der stufenweise Aufbau eines Grids

Ein Grid System wird unter anderem auch in seiner Ausdehnung bzw. Größe unterschieden. Die Ausdehnung eines implementierten Grids stellt natürlich verschiedene Anforderungen dar, wie Trust & Confidence und Sicherheitsaspekten.

Man unterscheidet daher 5 verschiedene Ausbaustufen.

Endsystem

Diese Stufe ist die kleinste in einem Grid System. Hierzu zählen einzelne Rechner, Speichersysteme und Sensoren.

Ein Endsystem zeichnet sich dadurch aus, dass es homogen, hochintegriert und in der Software- und Hardwarearchitektur sehr gut erforscht ist. Zudem sind hier verschiedene Dienste, wie Authentifizierung und Prozesskommunikation enthalten.

Weiterhin wird in einem Endsystem der Zugriff auf die zur Verfügung stehenden Ressourcen des Systems über das Betriebssystem geregelt. Dadurch wird eine einfache Steuerung, Verwaltung und eine hohe Performance erzeugt.

Des Weiteren wird die Programmierung von effizienter und hochperformanter Software, durch das hohe Maß an Integration zwischen Prozessorarchitektur, Speichersystem und Compilertechnik in Endsystemen, wesentlich vereinfacht.

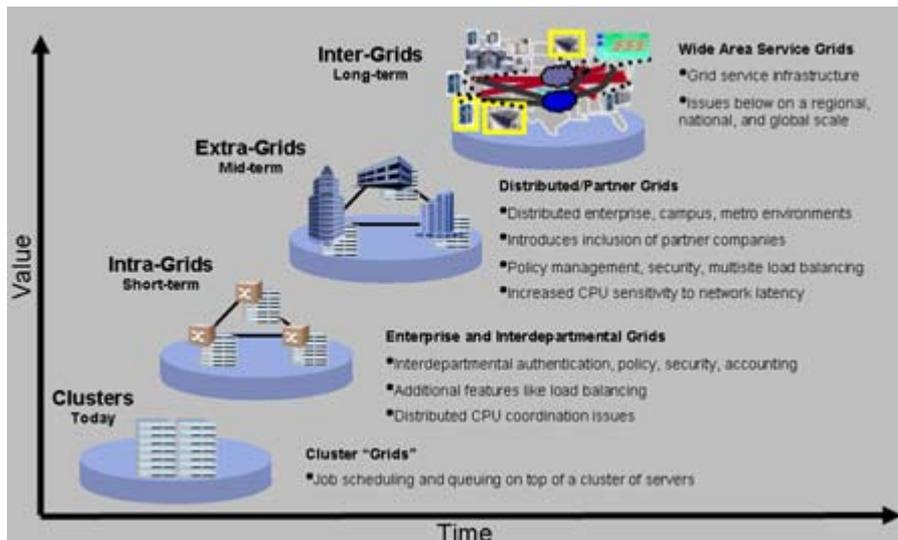


Abbildung 2 Evolution der Grids [11]

Cluster

Die zweite Stufe stellt in einem Grid Computing das Cluster dar.

Ein Cluster ist ein Verbund von Rechnern, die durch ein Hochgeschwindigkeitsnetzwerk (HiPPI) miteinander verknüpft sind.

Dabei bilden die beteiligten Rechner ein einheitliches Rechnersystem, das heißt, sie verfügen über die gleiche Hardwarekonfiguration und das gleiche Betriebssystem. Man spricht hier auch von einer homogenen Einheit.

Weiterhin befinden sich die Rechner von der geographischen Ausdehnung her, eng bei einander. Sie werden beispielsweise in Abteilungen von Unternehmen für Tests von Simulationen bzw. zum Testen von Grid Software eingesetzt. Die Verwendung dieser Cluster weisen meist ein Größe von zwei bis zu 6000 Knoten im Forschungsbereich auf, die der Steuerung bzw. Verwaltung von wenigen Personen unterliegt.

Des Weiteren ergeben sich für das Cluster-„Grid“ aufgrund der geringen Ausdehnung im Bereich Sicherheit und Policies wenige Probleme. Diese werden auch somit nicht weiter betrachtet.

Die zu bewältigenden Probleme in einem Cluster-„Grid“ sind die Verbesserung der Zugriffe auf den Datenspeicher und die Verbesserung der Performance des Gesamtsystems.

Intra-Grids

Ein Intra-Grid ist eine Vernetzung heterogenen Rechnersystemen, die räumlich weit entfernte Ressourcen nutzen.

Die Anwendung eines Intra Grid findet meist in mehreren Abteilungen einer Organisation statt, beispielsweise im IT Sektor oder im Forschungsbereich.

Aufgrund der Ausdehnung zu einem Organisation-Grid können im Gegensatz zu einem Cluster Grid mehrere Arten von Ressourcen genutzt werden, wobei die Möglichkeit über einen Überblick von sämtlichen Ressourcen und deren Zustände für den Benutzer nicht mehr gegeben ist.

Ein weiteres Ziel ist es, aufgrund der verteilten Nutzung von Ressourcen, die Rechenleistung zu erhöhen und den Zugriff auf den Datenspeicher allgemein zugänglich zu machen.

Hierfür entstehen in einem Intra-Grid ebenfalls im Bezug auf die Sicherheits- und Policiesaspekte keine hohen Anforderungen, obwohl dieser Bereich von enormer Bedeutung ist. (siehe Kapitel 3.2)

Jedoch ergeben sich durch das Netzwerk von heterogenen Rechnern verschiedene Anforderungen an die verschiedenen Schichten eines Systems.

Im Intra-Grid wird in der Softwareschicht die Implementierung von Mechanismen benötigt, die aufgrund der zunehmenden Komplexität dafür zuständig sind, dass die zu verwendenden Ressourcen kompatibel zu der eigenen Software ist.

Des Weiteren muss die Middleware einfache Scheduling-Mechanismen zur Verfügung stellen, damit festgestellt wird, wann welche Aufgabe auf welche freien Ressourcen ausgeführt werden kann. Auch für die darunter liegende Netzwerkschicht werden hohe Anforderungen in Bezug auf Bandbreite und Übertragungsgeschwindigkeit eines Netzwerkes gestellt.

Extra-Grids

Das Extra Grid ist eine weitere Vernetzung von mehreren Cluster-Grids oder Intra-Grids, das über große geografische Distanzen funktionieren soll. Hierbei werden Rechenleistungen und Daten über die Organisationsgrenzen hinweg zur Verfügung gestellt.

Im Gegensatz zu einem Intra-Grid steigen aufgrund der Ausbreitung in einem Extra-Grid natürlich auch die Anforderung an das Grid-Management, sowie die Sicherheitsanforderungen, wie Authentifizierung und Autorisierung.

Darüber hinaus müssen eine Topologieerkennung und ein geeignetes Load-Balancing für eine ausreichende Performance des Grid vorhanden sein.

In der Praxis wird das Extra Grid in Forschungsabteilungen oder –einrichtungen eingesetzt.

Inter-Grids

Ein Inter-Grid besteht aus mehreren heterogenen Rechnern, die in einem Rechnerverbund miteinander verknüpft sind und geographisch weit auseinander liegen.

Im Gegensatz zu einem Extra-Grid werden in dieser Form des Grids unterschiedliche Unternehmen und Nationen involviert und es liegt kein zentraler Administrationsbereich für die Verwaltung und Steuerung der Ressourcen vor.

Natürlich sind bei einem Grid, mit dieser Ausdehnung, die Anforderungen an die Sicherheitsaspekte, wie Authentifizierung und Autorisierung sowie die Anforderungen an effiziente Scheduling und Policies von großer Bedeutung. Dabei wird die Ausdehnung des Grid hierarchisch geordnet, um die Organisation zu vereinfachen und die Skalierbarkeit zu verbessern.

Des Weiteren ergibt sich auch aufgrund der steigenden Komplexität der gesamten Infrastruktur eine immer höhere Anforderung an die Middleware.

Interessant bei dem Inter-Grid ist, dass hier nicht nur wie bei einem Intra-Grid oder Cluster Ressourcen verwaltet und benutzt werden, sondern hier ergibt sich in wirtschaftlicher Hinsicht die Möglichkeit, aufgrund der Einbindung verschiedener Unternehmen, die Ressourcen zu vermieten bzw. zu „leasen“.

Es wird aber eingeschätzt, dass diese Form des Grids erst in ca. 10 Jahren zu erreichen ist.

3. Architektur eines Grid Systems

In diesem Kapitel wird zunächst eine mögliche Grid Architektur dargestellt, die von dem Global Grid Forum (GGF) entwickelt wird. Des Weiteren wird die Sicherheit von Grid Systemen beschrieben, sowie deren Ressourcen betrachtet.

3.1 Aufbau eines Grid

Eines der schwierigsten Probleme beim Entwurf von Grid Systemen liegt in der allgemeinverbindlichen Definition von Protokollen und Diensten, da diese eine Gewährleistung von Interoperabilität von Systemen und Anwenderprogrammen darstellen.

Um die vernetzte Zusammenarbeit unterschiedlicher Systeme zu erreichen, verwendet das Grid in ihrer Architektur ein Schichtenmodell⁶.

Die Besonderheit dieses Schichtenmodells ist im Gegensatz zu dem Schichtenmodell des Internets, dass höhere Schichten die Funktionsweisen und Funktionalitäten niederer Schichten

⁶ Ian Foster: The Anatomy of the Grid: Enabling Scalable Virtual Organizations.
Software Services Grid Workshop, Danvers, 2001

erkennen und diese direkt nutzen können. Diese Eigenschaft entsteht durch die „Durchsichtigkeit“ der Schnittstellen zwischen den einzelnen Schichten, was zu einer wesentlichen Verbesserung der Performance des Gesamtsystems führt.

Das Sanduhr-Modell definiert so ein Schichtenmodell. Der Hals der Sanduhr, der in dem Modell durch die Connectivity- und die Ressourcen-Schicht dargestellt ist, wird mit Absicht recht schmal gehalten, da die Schnittstellen recht schlank und die Anzahl der Protokolle möglichst gering gehalten werden. Um die Interoperabilität zu gewährleisten, müssen die beiden Schichten zur Konstruktion eines Grid folgendes leisten:

- Koallokation und Koordination von Ressourcen, ohne damit einer zentralen Kontrollinstanz unterworfen zu sein
- Verwendung von offenen Standardprotokollen und Schnittstellen

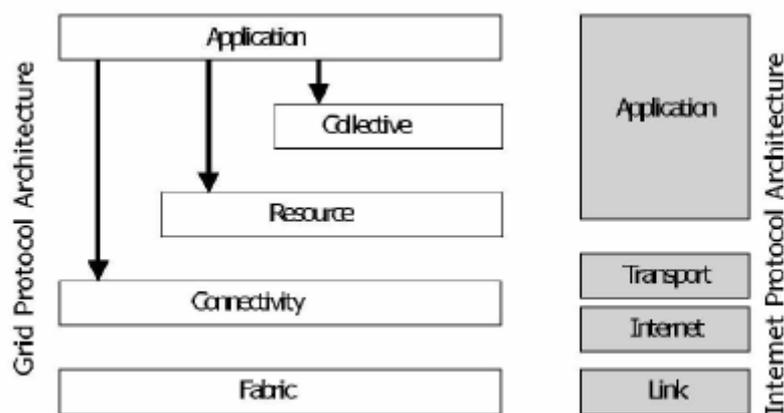


Abbildung 3 Grid Architektur ;[7]

In den nachfolgenden Abschnitt werden die einzelnen Schichten einer Grid Architektur näher betrachtet.

1. Fabric – Schicht

Die Fabric Schicht bildet die unterste Ebene. In dieser Ebene werden für einen gridweiten Zugriff die benötigten Ressourcen zur Verfügung gestellt.

Die Ressourcen können von unterschiedlicher physischer oder logischer Natur sein (Netzwerk-, Rechnerressourcen, Code, Kataloge, Sensoren) und stellen daher spezifische Funktionen zur Verfügung. Mögliche Beispiele könnten sein:

- Überwachung und Steuerung von Rechen-Ressourcen
- Steuerung des Netzwerkes

- Laden bzw. Speichern von Dateien in Speicher-Ressourcen
- Ausführen von Programmen

Außerdem sind die Funktionen, die in der Fabrikebene realisiert werden, in der Lage, die oberen Schichten zu beeinflussen. Beispielsweise kann eine höhere Schicht die Aggregation einer Ressource ausführen, wenn diese Ressource, die in dieser Schicht zur Verfügung steht, die Advanced Reservation unterstützt.

Für diesen Zugriff wird eine standardisierte Schnittstelle vorausgesetzt, die zur Abfrage der Fähigkeiten dient, sowie den Zustand einer Ressource beschreibt.

2. Connectivity – Schicht

Die Connectivity – Schicht definiert die Kommunikations- und Authentifizierungsprotokolle, die für Grid spezifische Anwendungen benötigt werden. Die Kommunikationsprotokolle dienen zum Transport, Namensauflösung und Routing von Daten zwischen den einzelnen Fabric-Layer-Ressourcen. Für diese Art der Kommunikation verwendet man die Protokolle der TCP/IP – Familie.

Die Authentifizierungsprotokolle setzen auf die Kommunikationsdienste auf und ermöglichen einen kryptographischen Sicherheitsmechanismus für die Identifikation- und Verifikationsbestimmungen des Benutzers bzw. der Ressourcen.

Sie sollten folgende Eigenschaften besitzen:

- **Single-Sign-On**

Der Benutzer eines Grid Systems muss sich für den Zugriff auf die vorhandenen Ressourcen nur einmal, zentral anmelden.

- **Delegation**

Ein Nutzer soll die Möglichkeit haben, seine administrativen Rechte an andere Nutzer zu delegieren. (Rechtestruktur beachten)

- **Integration in verschiedene lokale Sicherheitslösungen**

Hier besteht die Möglichkeit, dass verschiedenen Sicherheitslösungen untereinander interagieren können.

3. Ressource – Schicht

Die Ressourcen Schicht baut auf die Connectivity – Schicht auf und definiert die Protokolle für die Interaktion, Initiierung, Zuteilung, Steuerung und Überwachung einzelner Ressourcen der Fabric – Schicht.

Hierbei werden zwei Arten von Protokollen unterschieden:

- Informationsprotokolle
Das Protokoll liefert die Information über die Struktur und den aktuellen Status einer Ressource (z.B. Auslastung bzw. Kosten einer Ressource)
- Managementprotokolle
Das Protokoll wird benötigt, um den Zugriff auf die Steuerung dieser Ressourcen zu gewährleisten.

Die Ressourcen – Schicht darf nur die Funktionen bereitstellen, die von den, in der Fabric – Schicht vorhandenen Ressourcen realisiert werden können.

4. Collective Layer

Die Collective Schicht koordiniert die Interaktion zwischen den einzelnen Ressourcen.

Im Gegensatz zu der Ressourcen – Schicht enthält diese Schicht Dienste, Protokolle, das Application Programming Interface (API) und das Software Development Kit (SDK), die nicht in Verbindung mit anderen Ressourcen stehen.

Weiterhin besteht die Möglichkeit für die kollektiven Komponenten, die auf der Connective – Schicht und Ressourcen – Schicht aufbauen, dass sie sich an den Anforderung bzw. den Eigenschaften der virtuellen Organisation anpassen und somit auch übergreifende Aufgaben ausführen.⁷

Mögliche Aufgaben sind:

- Verzeichnisdienste für die VO –Teilnehmern anzubieten, um in den Ressourcen nach Namen oder Attributen zu suchen
- Co-allocation, Scheduling und Brokerring Service dienen dazu, für einen VO – Teilnehmer, die von ihm ausgewählten Ressourcen für einen bestimmten Verwendungszweck zu reservieren.

⁷ Ian Foster: The Anatomy of the Grid: Enabling Scalable Virtual Organizations. Software Services Grid Workshop, Danvers, 2001

- Monitoring- und Diagnose-Dienste dienen zur Überwachung von Ressourcen, um einen Angriff bzw. eine Auslastung oder eine Überlastung zu erkennen.
- Software – Discovery – Dienste werden genutzt, zur Auffindung bestimmter Plattformen, für die Ausführung und Implementierung von Softwarekomponenten.
- Gemeinschaftauthorisationsserver ermöglichen den Zugriff auf die Gemeinschaftsressourcen für eine bestimmte Gruppe von Mitgliedern

Die Funktionen der Collective – Schicht als SDK's oder als Dienste werden mit den dazu gehörigen Protokollen implementiert.

5. Applications Layer

Die Applications – Schicht präsentiert die Nutzungsapplikation, die ein Benutzer oder VO – Teilnehmer innerhalb eines Grids bzw. der virtuellen Organisation verwendet. Diese Nutzungsapplikationen sind Systeme wie Bibliotheken und Frameworks.

Des Weiteren werden hier ebenfalls Protokolle verwendet, die die Nutzung durch einen direkten Zugriff auf bestimmte Dienste ermöglichen (z.B. Verwaltung bzw. Verteilung von Ressourcen).

3.2 Sicherheit im Grid System

Die Aufwendungen für zusätzliche Sicherheitsmaßnahmen in einer dezentralen Systemarchitektur können sehr unterschiedlich sein.

In einem Cluster-„Grid“ sind diese relativ gering, da es sich hier um eine kleine, überschaubare und in sich geschlossene Gruppe von Anwendern und Rechnern handelt. Ein bekanntes Beispiel hierfür ist die Vereinigung aller Ressourcen in einer Abteilung, die ohnehin durch verschiedene Sicherheitsaspekte, wie Firewalls oder Authentifizierung für das entsprechende System, von anderen Organisationseinheiten des Unternehmens getrennt ist.

Der Sicherheitsaspekt in einem Extra oder Inter Grid gestaltet sich gegenüber einem Cluster Grid etwas aufwendiger. Da es sich hier meist um einen Zusammenschluss mehrere Abteilungen eines Unternehmens handelt, um etwa Spitzenbelastungen besonders rechenintensiver Applikationen abzufangen.

Der höhere Aufwand für die Sicherheitsmaßnahmen zeichnet sich dadurch aus, dass man mitunter ein Regelwerk für die wechselseitige Nutzung von Ressourcen benötigt, da der Zugriff möglicherweise über die Grenzen von den organisatorischen Einheiten erfolgt.

Außerdem muss die Verwaltung und die Kontrolle der User vereinheitlicht werden, so dass die Überwachung von den Usern einfacher zu realisieren ist.

Für diese Art des Grids sollte jedoch als Minimalmaßnahme, zur Absicherung von Sicherheitsanforderungen, die Zusammenfassung aller Systeme in einem Virtual Private Network (VPN) erfolgen.(Firewall, Proxy)

Unter einem VPN versteht man, dass Datenpakete eines beliebigen Protokolls verschlüsselt und verpackt über das Internet gesendet werden und es zu einer Überwachung bei der Übertragung kommt. Die unterschiedlichen Adressbereiche werden wie ein Gesamtsystem betrachtet.

Ein Inter-Grid stellt an die Sicherheitsaspekte eine hohe Anforderung, da allein die Benutzerverwaltung eine sehr aufwändige Infrastruktur benötigt.

Generell ergeben sich in einem Grid die folgenden sicherheitsrelevanten Ansatzpunkte:

- **Vertraulichkeit**

Hier soll der Inhalt einer Nachricht sichergestellt werden, so dass der Zugriff bzw. die Nutzung der Nachricht für andere User verweigert wird. Dieser Aspekt ist für einen Gridbenutzer von besonderer Bedeutung, wenn er den Zugriff auf die Informationen bzgl. seiner Grid – Jobs, für Konkurrenten verhindern möchte. Daher besteht eine Maßnahme gegen diese Bedrohung in der Sicherung von Daten durch Verschlüsselung.

- **Zugangskontrolle**

Die Zugangskontrolle entscheidet über den Zugriff auf eine bestimmte Ressource eines Grid – Users. (Berechtigungskonzept)

Die Voraussetzung für eine sichere Zugangskontrolle ist in jedem Fall, die zuvor beschriebene sichere Authentifizierung, die den Zugang bzw. die Registrierung des Users überprüft.

Diese Sicherheitsmaßnahme bietet für den Gridbenutzer einen weiteren wichtigen Sicherheitsaspekt, denn andere Gridbenutzer erhalten dadurch keinen unerlaubten Zugang auf seinen Gridjob, um somit beispielsweise den Verlauf einer Berechnung zu beeinflussen. Damit wird sichergestellt, dass nur die jeweils berechtigten User die entsprechenden Ressourcen nutzen dürfen. Die hierfür benötigten Regeln für die Erlaubnis bzw. Verweigerung des Zugangs legt jeder Gridbetreiber selbst in einer so genannten Policy fest. (Admin.-Konzept berücksichtigen)

- **Authentifizierung**

Die Authentifizierung stellt die wichtigste Maßnahme für eine sichere Gridbenutzung da. Hierbei werden die Grid- User eindeutig über die User ID bzw. über das dazu gehörige Passwort identifiziert.

Von der Seite des Gridbetreibers wird sichergestellt, dass nur registrierte User die bereitgestellten Ressourcen benutzen dürfen. Ebenso besteht auf Seiten des Gridnutzer das Interesse, dass ein Angreifer nicht unter einer falschen Identität den Zugang zu einem Gridjob oder anderen Daten erlangt.

Weiterhin bestehen für diese Maßnahmen zur Sicherheit in Grids zwei Varianten: ein passwort- bzw. eine zertifikatsbasierte Lösung, die unterschiedliche Sicherheitsstufen darstellen. Diese Lösung, die zur Identifikation eines Users dient, wird als Credential bezeichnet.

Ein Passwort ist ein Geheimnis, das den Nachteil besitzt, bei der Übertragung zur Identitätsprüfung abgefangen zu werden bzw. weitergegeben werden kann. Vor diesem Hintergrund wird deshalb meist bei einer Identitätsprüfung das zertifikatsbasierte Verfahren verwendet.

Ein Zertifikat dient als Beglaubigung dafür, dass einem User ein Public Key für ein asymmetrisches Verfahren gehört.

Da in einem kryptographischen Verfahren zu einem Public Key immer ein Privat Key generiert ist, wird in einem Identitätsbeweis überprüft, ob der User diesen Private Key kennt. Ist dies der Fall, so wird angenommen, dass es sich um genau den User handelt, deren Identität in dem Zertifikat beglaubigt wurde.

Jedoch gilt das kryptographische Verfahren nur unter der Annahme als sicher, wenn es sicher eingestuft wurde. (Certification Authority CA)

Desweiteren sollte die Authentifizierung folgende Eigenschaften besitzen:

- *Single sign on*: Der Gridbenutzer braucht sich nur einmal im Grid anzumelden, um ohne weitere Authentifizierung auf alle zugriffsberechtigten Ressourcen zuzugreifen.
- *Delegation*: Ist die (oft zeitlich begrenzt) Übertragung von Rechten einer Identität zu einer anderen. (Berechtigungskonzept)

Generell gibt es jedoch keine Garantie dafür, dass ein User keinen illegalen Zugriff auf eine Grid Umgebung bekommt, um mit den sich dort befindenden Ressourcen zu manipulieren.

Daher muss die Sicherheit in einer Grid Umgebung in gleicher Weise zunehmen, wie die Komplexität des Grids zunimmt.

3.3 Die Ressourcentypen und das Ressourcenmanagement in einem Grid Computing

Ein Grid System ist ein Verbund aus Systemkomponenten, die oft als „Ressourcen“, „Knoten“ oder „Rechner“ bezeichnet werden. Die Hauptaufgabe der Systemkomponenten ist es, dass sie dem Benutzer ihre Ressourcen, unter Berücksichtigung der Zugriffsrechte, zur Verfügung stellen. Diese Bereitstellung an Ressourcen muss jedoch durch eine effiziente Verwaltung koordiniert werden.

Um die Vielfalt und das Management der Ressourcen einmal aufzuzeigen, werden im nachfolgenden Abschnitt diese näher beschrieben.

Ressourcentypen

Die Vielfalt der Ressourcentypen in einem Grid System ist sehr groß. Daher werden in diesen Abschnitt nur die wichtigsten Ressourcen, wie Rechenzeit, Speicher und Job – Applikationen dargestellt.

Rechenzeit

Ein Grid System ist in der Lage die Rechenzeit als Ressource anzubieten. Diese Form der Bereitstellung dient dazu, ungenutzte CPU-Leistung dem User zur Verfügung zu stellen. Sie ist damit auch die am häufigsten verwendete Ressource, die ein Grid System anbietet.

Hierbei kann natürlich eine heterogene Landschaft an Rechnern vorliegen, die sich in ihrer Architektur, Geschwindigkeit, Speicher und anderen Komponenten unterscheiden.

Prinzipiell gibt es drei Möglichkeiten, die Rechenkapazität eines Grids zu nutzen.

Der erste Weg ist es, eine fremde Rechenleistung zu nutzen, in dem eine Applikation auf einem anderen Rechner verwendet wird. Die zweite Möglichkeit besteht darin, Applikationen zu benutzen, die parallel oder nacheinander auf verschiedene Prozessoren ausgeführt werden können. Die letzte Möglichkeit besteht in der Verwendung einer Applikation, die öfter auf verschiedenen Knoten aufgerufen werden muss.

Speicher

Der Datenspeicher gehört ebenfalls zu den wichtigsten Ressourcen, die ein Grid System anbietet. Bei Grid Systemen, bei denen die Verwaltung von Daten bzw. ganzen Datenbanken im Vordergrund steht, werden meist als „data grids“ bezeichnet.

Hierbei verfügt jeder Rechner über verschiedene Arten und Größen von Datenspeichern. Zum einem verwendet ein Rechner temporäre Speicher, beispielsweise RAM, die eine niedrige Latenzzeit besitzen, diese werden meist nur vom eigenen Rechner in Anspruch genommen.

Zum anderen verwendet ein Rechner auch sekundäre Datenspeicher, wie Festplatten oder Wechseldatenträger, die eine hohe Kapazität an Speichervolumen besitzen.

Die Nutzung des Datenspeichers führt dazu, um in Grid Systemen eine höhere Performance bzw. eine höhere Sicherheit der Daten zu erzielen. Hierzu nutzt meist ein Grid so genannte mountbare Dateisysteme, wie das Network File System (NFS), das Distributed File System (DFS) oder das Andrew File System (AFS), die verschiedene Einstellungsmöglichkeiten und Sicherheitsfunktionen dem Grid anbieten.

Weiterhin ergibt sich eine Erhöhung der Speicherkapazität für das Gesamt-Grid, im dem eine Verteilung des Speichers auf mehreren Rechner durch ein einheitliches File System ermöglicht wird. Dadurch ist die Datenbank nicht an der Software mit seinem jeweiligen File System gebunden.

Dies wiederum ermöglicht den Benutzern einen einfacheren Datenzugriff, ohne den Standort zu kennen, da er in einem einheitlichen File System auf einer einheitlichen Speicherplattform zugreift.

Ausgereifere File Systeme verfügen über eine Technologie, die in der Lage ist, Datensätze zu kopieren, um die Sicherheit der Daten weiter zu erhöhen. Dabei ergibt sich als Synergieeffekt eine Erhöhung der Performance. Ein intelligentes Grid Schedulingverfahren ist in der Lage die Systeme auf die CPU-Belastung und Netzwerkauslastung optimal zu überwachen. Die Prozesse, die diese Daten zur Lösung benötigen, können durch die Unterstützung des Schedulingverfahrens, in der Nähe des Speichermediums ausgeführt werden. Als Nebeneffekt würde sich auch hier eine niedrigere Latenzzeit ergeben, was wiederum zu einer Erhöhung der Performance in einem Grid System führt.

Außerdem machen weitere Funktionen, wie Data Striping oder Mirrors, das Grid sehr mächtig und bieten insgesamt hohe Performancegewinne an. Dabei werden auch verbesserte Sicherheitsaspekte berücksichtigt.

Job und Applikationen

In dieser Arbeit wurde bereits erwähnt, dass ein Grid auf Ressourcen innerhalb des Systems zugreifen kann. Dieser Zugriff erfolgt jedoch nicht direkt vom User aus, sondern wird über so genannte Applikationen oder Jobs ausgeführt. Dabei besteht eine Applikation aus mehreren Jobs. Ein Job wiederum kann in unterschiedliche Subjobs segmentiert werden. (siehe Abb.4)

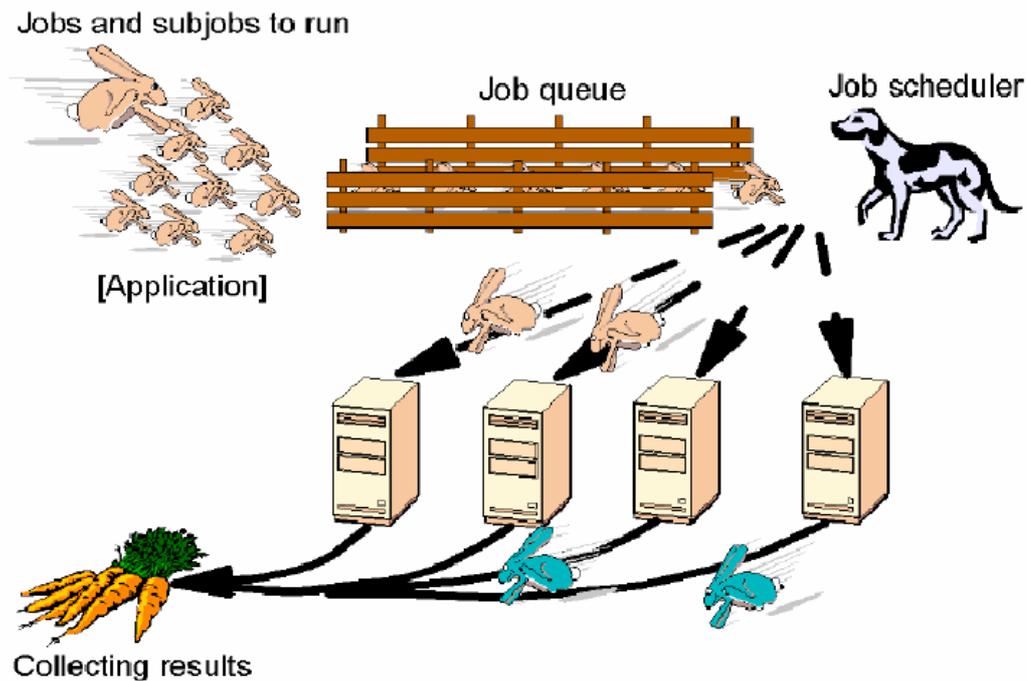


Abbildung 4 Anwendungen, Jobs und Subjobs ;[1]

Prinzipiell werden Jobs verwendet, um Berechnungen, Bearbeitungen und Verschiebungen von Daten durchzuführen. Die Ausführung der Jobs sollte daher in dem jeweiligen Ereignisbereich realisiert werden, um eine bestmögliche Performance und eine niedrige Latenzzeit zu erreichen.

Des Weiteren kann aber die Ausführung einiger Applikationen zu Warteschlangen von Jobs führen. Diese Form der Abhängigkeit zwischen den Jobs wird dadurch verursacht, dass Jobs auf die Resultate von anderen Jobs warten oder diese miteinander kommunizieren müssen.

In einem Grid System kommt der Datenzugriff über die Jobs bzw. Subjob nur dann zum Tragen, wenn die Jobs hierarchisch geordnet sind und die einzelnen Resultate zu einem Endresultat der Applikation gesammelt werden.

Ressourcenmanagement

Die Ressourcen in einem Grid System können nur durch eine optimale Verwaltung effizient genutzt werden. Dabei sind die Begriffe „Scheduling und Reservation“ von zentraler Bedeutung.

Scheduling

Ein Ressourcenmanagement ist dafür verantwortlich, dass eine Aufgabe an den Rechner gesendet wird, der für die Bearbeitung und Ausführung der Aufgabe zuständig ist. In einem Grid wird dies durch ein so genanntes Job Scheduling realisiert. Das Schedulingverfahren bedeutet nicht, dass es für die Reservation von Ressourcen zuständig ist, sondern es ist für die Verwaltung der Jobs und die damit verbundene Verteilung an die ausführenden Rechner verantwortlich.

Da die Verteilung der Jobs die Performance eines Grid System mitbestimmt, muss der Job Scheduler auch die Ressourcenauslastung kennen. Eine nicht effiziente Verteilung der Jobs kann zu Warteschlangen führen, wenn die Jobs vom Scheduler an ausgelastete Rechner gesendet werden. Die Grid Systeme, bei denen die Rechner ihren Status an den Scheduler schicken, werden als so genanntes „Scavenging Grids“ bezeichnet.

Weiterhin besteht die Aufgabe für ein Job Scheduler darin, die Ausführung eines zugeteilten Jobs zu überwachen. Im Falle einer nicht erfolgreichen Bearbeitung muss der Scheduler diesen Job für den jeweiligen Rechner erneut aufrufen bzw. an andere Rechner schicken. Jedoch sollte dabei ein Scheduler die Verteilung zu einer Endlosschleife vermeiden.

Reservation

Das Ressourcenmanagement in einem Grid System bietet die Möglichkeit Ressourcen zu reservieren. Dieses Feature reserviert, wie bei einer Raumbuchung von Konferenzräumen, eine geeignete Ressource für eine bestimmte Dauer, die für die Bearbeitung des Jobs vorgesehen wird. Dies führt nicht nur zu einer Performancesteigerung, sondern führt ebenso zu einer Steigerung der Qualität der gesamten Anwendung.

Eine Reservierung ist aber nicht für alle Ressourcen durchführbar. Insbesondere gilt dies für Ressourcen, die von dedizierten Systemen bereitgestellt werden. Ein dediziertes System ist ein Rechnersystem, das nur sekundär bzw. freiwillig für die Bearbeitung von Jobs in einem Grid zur Verfügung steht und daher keine ständige Verfügbarkeit und Dienstqualität bereitstellt.

4. Grid Toolkits

In diesem Kapitel werden verschiedene Software Pakete, so genannte Grid Toolkits zur Implementierung eines Grid Systems aufgezeigt. Dabei wird im ersten Abschnitt ein kleiner Überblick über unterschiedliche Implementierungen dargestellt und im zweiten Abschnitt werden die beiden wichtigsten Pakete, das Globus Toolkit und die Sun Grid Engine beschrieben.

4.1 Überblick über verschiedene Implementierungen

In dem Abschnitt werden kurz bekannte Softwarepakete für Grid Projekte vorgestellt, die ebenfalls einen großen Teil zur Etablierung des Grid Computings beigetragen haben.

Legion

Legion⁸ ist seit 1993 ein Projekt des Department of Computer Science der University of Virginia und wird als ein objektbasiertes Metasystem bezeichnet. Die Idee dieses Projektes ist es, mittels eines Hochgeschwindigkeitsnetzes, Millionen von Hosts und Objekten zu einem System zusammen zu fassen. Den Benutzern dieses System werden durch einen transparenten Zugriff verteilte Ressourcen zur Verfügung gestellt, die zusammen einen virtuellen Supercomputer erzeugen.

Ermöglicht wird die Nutzung der verteilten Ressourcen durch spezielle Komponenten, die das Legion - Projekt bereitstellt, wie beispielsweise transparentes Scheduling, Fehlertoleranz, Site Autonomie, Datenmanagement und eine Vielzahl von Sicherheitsmechanismen.

Zwischen den Anwendungen und dem Betriebssystem stellt die Legion eine Middleware dar. Dabei werden die verteilten Ressourcen als Objekte präsentiert. Somit bildet das Legion ein objektorientiertes System.

Darüber hinaus stellt das System verschiedene Dienste und Service, wie „Legion file system“, die „Ressource management service“ zur Verfügung, die für die Verwaltung und das Management der Objekte zuständig sind. Die Kommunikation zwischen den einzelnen Objekten wird durch einen asynchronen Methodenaufruf realisiert. Der Zugriff auf die

⁸ Vgl. <http://legion.virginia.edu/index.html>

Schnittstelle des Objektes wird mittels der CORBA Interface Definition Language (IDL) beschrieben.

Grundsätzlich besitzt jedoch das ganze Projekt eher einen wissenschaftlichen Charakter, da dieses Projekt mehr als eine Sammlung von Schnittstellen und Spezifikationen angesehen wird und weniger als eine Implementierung einer Gesamtlösung. Dabei entstanden kommerziellen Produkte (wie AVAKI), die als Basis die Legion besitzen.⁹

PUNCH

Der Punch¹⁰ ist eine Implementierung, die das world wide web in ein Grid Portal verwandelt. Dabei bildet der Punch eine so genannte Internet Computing Plattform. Die Benutzung des Grid Portals erfolgt hier über den Webbrowser, wie beispielsweise den Internet Explorer oder Netscape, der mit Hilfe von verschiedenen Protokollen den Zugriff auf Daten oder das Starten von Applikationen ermöglicht.

Weiterhin besteht die Möglichkeit in einem Punch, dass verschiedene Ressourcen, wie Daten, Applikationen oder Programme an unterschiedliche Plätze abgelegt werden können und verschiedene Komponenten diese verwalten.

Condor

Das Condor Projekt¹¹ entstand 1985 an der university of Wisconsin-Madison. Dieses Projekt basiert auf den Auswertungen des Remote Unix (RU) –Projektes und ist ein so genanntes „workload management system“ für rechenintensive Aufgaben.

Der Condor kann zur Steuerung und Verwaltung der Jobs innerhalb eines Clusters von Rechnern genutzt werden.

Das Ziel dieses Projektes ist die Implementation, die Entwicklung und Anwendung von Komponenten, welche das High Throughput Computing von verteilten Computing Ressourcen ermöglicht. Hier steht nicht nur die technologische IT – Plattform im Vordergrund sondern auch die soziologische Umgebung wird dabei wahrgenommen.

Im Gegensatz zu einem Grid System hat der Condor den Nachteil, dass er ohne zusätzliche Mechanismen nicht organisationsübergreifend verwendet werden kann. Darüber hinaus können Ressourcen, die mit keinem Condor ausgestattet sind, nicht bereitgestellt werden.

⁹ Die vorliegenden Ausführungen sind Vgl. / <http://www.avaki.com> / entnommen worden

¹⁰ Die vorliegenden Ausführungen sind Vgl. / <http://www.punch.purdue.edu> / entnommen worden.

¹¹ Vgl. <http://www.cs.wisc.edu/condor/>

Die Entwicklung neuer Mechanismen, wie der „flocking“ – Mechanismus, sind in der Lage verschiedene Condor – Cluster zu vereinen. Zudem kann über ein Condor – G ein Condor im Globus – Grid implementiert werden.

Nachfolgend werden in diesen Abschnitt die beiden wichtigsten Pakete beschrieben.

4.2 Das Globus Toolkit

Das Globus Toolkit¹² ist ein Middleware – System, das den Aufbau von Grid System unterstützt. Hierfür werden Dienste und Bibliotheken für das Ressourcen- und Filemanagement, sowie für die Sicherheit angeboten, um einen organisationsübergreifenden Zugriff auf verschiedene Datenbanken oder Rechenleistungen zu ermöglichen.

Weiterhin beinhaltet dieses Tool Software für Sicherheit, Informationen zur Infrastruktur, Ressourcenmanagement, Datenmanagement/Kommunikation, Fehlererkennung und Portabilität. Darüber hinaus dient als Basis für das Globus Toolkit ein heterogenes Low-Level-Protokoll. Weitere Spezifikationen oder Standards wie CORBA oder Message Passing Interface, die zur Kommunikation innerhalb eines Grids dienen, nutzen dieses Toolkit.

Insbesondere bieten bestimmte Dienste und Protokolle für einen Benutzer des Globus Toolkits die Möglichkeit, aufgrund der Zugriffe auf organisationsübergreifende Ressourcen, den genauen Überblick über seinen Rechner nicht zu verlieren.

Das Globus Toolkit besteht aus drei unterschiedlichen Komponenten (siehe Abbildung 5). Diese Komponenten stehen dem Benutzer unabhängig voneinander zur Verfügung und setzen gemeinsam auf die Security – Schicht auf.

Diese Komponenten, sowie die Security – Schicht werden im Folgenden etwas näher betrachtet.

¹² Die vorliegenden Ausführungen sind Vgl. / <http://www-unix.globus.org/toolkit/> / entnommen worden.

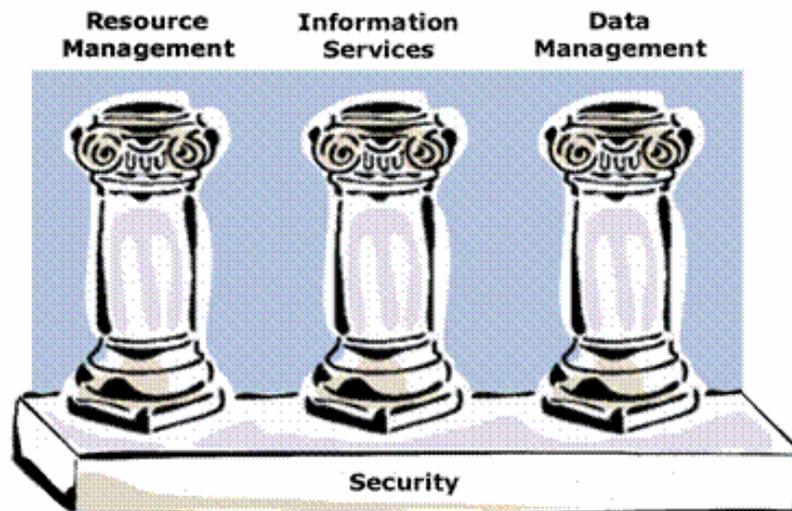


Abbildung 5 Architektur des Globus Toolkits ; [12]

Grid Security Infrastructure

Die Grid Security Infrastructure(GSI) ist für eine sichere Kommunikation zwischen den verteilten Rechnern zuständig. Darüber hinaus stellt das GSI Dienste für die Authentifizierung von Benutzern und die Geheimhaltung von Daten bereit.

Weiterhin sind aufgrund eines dezentralen Sicherheitssystems in einem Grid, für den Nutzer verschiedene Mechanismen, wie die „single sign – on“ Funktionalität vorgesehen.¹³

Die Grid Security Infrastruktur dient als Grundlage für die folgenden drei Komponenten des Globus Toolkits.

Ressourcen Management

Das Grid Ressource Allocation Management ist für die Ausführung, das Überwachen und das Beenden eines Jobs verantwortlich. Zudem bietet es die Möglichkeit bestimmte Ressourcen zu reservieren. Während der Bearbeitung des Jobs sendet das GRAM einen Statusbericht über den Zustand des Jobs an den Client und aktualisiert den Information Service (MDS = Distributed access to structure and state information).

¹³ Vgl. [Globus Project 2002i]

Information Service

Der Informationsdienst in einem Globus Toolkit wird als Monitoring and Discovery Service (MDS) bezeichnet. Der Service hat einerseits die Aufgabe in bestimmte Verzeichnisstrukturen Informationen über die Art und den Status von Ressourcen abzuspeichern und andererseits bietet er die Möglichkeit auf weitere Informationen zuzugreifen. Diese Informationen können statische (Bsp.: Art der verwendeten Software) oder dynamische (Bsp.: CPU- Auslastung, ungenutzter Speicherplatz) Informationen von Ressourcen sein.

Des Weiteren wird in diesem Service das Resource Management Protocol implementiert, das die Abfrage der Informationen ermöglicht.

Daten Management

Das Data Management ist für eine zuverlässige und effiziente Datenübertragung zwischen einzelnen Rechnern innerhalb eines Grids entwickelt worden.

Aufgrund dieser Eigenschaften und der Architektur dient das Globus Toolkit als DV – Basis für die Entwicklungs- und Forschungsaufgaben in Unternehmen. Daher findet das Tool ein starkes Ansehen, was durch unterschiedliche Projekte zu erkennen ist, wie zum Beispiel

- *Earth System Grid II*¹⁴ ist ein Projekt des US Department of Energy zu Erforschung eines globalen Modells des Erdsystems
- *DataGrid* ist ein Projekt der Europäischen Union, das Rechenkapazität und die Forschung an verteilten Datenbanken anbietet.

4.3 Sun Grid Engine

Die Sun Grid Engine (SGE)¹⁵ ist eine verteilte Ressourcenmanagement – Software und liegt in zwei Versionen vor. Die Standard Version kann sich der Nutzer kostenlos aus dem CVS (Concurrent Versions System) vom Internet downloaden, sowie den dazugehörigen

¹⁴ Die vorliegenden Ausführungen sind Vgl. / www.earthsystemgrid.org / entnommen worden.

¹⁵ Die vorliegenden Ausführungen sind Vgl. / <http://www.sun.com/software/grid/> / entnommen worden.

Sourcecode. Hingegen stellt Sun Microsystems die Sun Grid Engine Enterprise Edition (SGEEE) kostenpflichtig bereit.

In der Gesamtarchitektur eines Grids nimmt die SGE nur einen geringen Teil ein, da diese auf der Applikationsschicht implementiert ist.

Das Einsatzgebiet der Grid Engine wird von Sun mehr in einer Umgebung aus Hard- und Software von Sun Microsystems gesehen. Dabei erstreckt sich das Einsatzgebiet von Servern und Speichersysteme über Runtime – Bibliotheken, Ressourcenmanagement und Entwicklungstools bis hin zu einer System Management Software.¹⁶

Die Aufgabe der Sun Grid Engine ist wegen ihrer Architektur und ihren Eigenschaften für die Distributed Resource Management (DRM) zuständig. Sie verwaltet und regelt verschiedene Software- und Hardwareressourcen in heterogenen und verteilten Systemen, um eine effiziente Ausnutzung dieser Ressourcen zu gewährleisten. Sie stellt beispielsweise die verschiedenen Rechenleistungen, die in einem Grid angeboten werden, als eine zusammenhängende Ressource dar. Der Benutzer erhält somit auf dem Grid nur einen einfachen Zugriffspunkt, für die Nutzung der Ressource.

Des Weiteren besitzt die SGE, wie Condor, eine DRM – Funktion um in einem traditionellen Stapelverarbeitungssystem, Warteschlangen, Statistiken über Jobverarbeitungen, Sperren und Wiederaufnehmen von Jobs und benutzerspezifische Ressourcen auszuwerten.

Darüber hinaus enthält die SGE eine Verbesserung, wie beispielsweise die Batch – Aware – Shell (Qtsch). Diese ermöglicht es zum Einen, dass interaktive Anwendungen in Verbindung mit der Sun Grid Engine Software verwendet werden können und zum Zweiten, werden ressourcenintensive Anwendungen ohne die Steuerung des Benutzers an einen geeigneten Server automatisch weitergeleitet.¹⁷

Allerdings sind die Verwendungsmöglichkeiten der Grid Engine sehr gering. Dies erklärt sich dadurch, dass die Sun Microsystems für den IT – Bereich nur eine sehr spezielle Hard- und Softwareumgebung zur Verfügung stellt.

¹⁶ Vgl. / [SUN Microsystems, Inc. 2002a]; S. 19-20 /

¹⁷ Vgl. / [SUN Microsystems, Inc. 2002a]; S. 21-22 /

5. Anwendungsbeispiele für Grid Computing

In diesem Kapitel werden einige Anwendungsbeispiele aus dem Bereich des Grid Computing vorgestellt. Das erste Beispiele, das Seti@Home – Projekt, ist ein sehr erfolgreiches Projekt aus dem Internet-Computing-Bereich und repräsentiert ein Rechen-Grid.

Das zweite Beispiel ist das dintrubuted.net – Projekt und beschreibt ebenfalls ein Rechen-Grid.

Am Ende dieses Kapitels wird das LHC-Computing-Grid-Projekt vorgestellt. Diese Projekt befindet sich momentan im Aufbau und repräsentiert ein Data-Grid.

5.1 Seti@Home

Das erste Anwendungsbeispiel, das hier näher betrachtet werden soll, ist ein Projekt des Space Sciences Laboratory an der University of Carlifornia in Berkeley.

Das Projekt trägt den Namen „SETI@HOME“¹⁸ und steht für Search for Extraterrestrial Intelligence. Das Ziel dieses Projektes ist es, nach intelligentem Leben außerhalb unseres Sonnensystems zu suchen. Dabei findet nur eine ausgiebige Analyse in dem sichtbaren Lichtteil und dem Radiowellenbereich des elektromagnetischen Spektrums statt. Jedes Signal, das einen kleinen Frequenzbereich besitzt und durch das größte Radioteleskop (Arecibo Radio Observatory auf Puerto Rico in der Karibik) der Welt empfangen wird, wäre wahrscheinlich ein Nachweis für außergewöhnliche Technologie und damit eine Hoffnung auf die Existenz von anderen Lebensformen.

In Wirklichkeit entsteht aufgrund der hohen Anzahl von empfangenen Radiosignalen eine solche Datenflut, die normalerweise von einem Großrechner analysiert bzw. bearbeitet werden muss.

Aus dieser Tatsache heraus entstand die Idee vom „SETI@HOME“ Projektes, das heutzutage mehr als 4,6 Millionen Benutzer umfasst und daher zu den größten verteilten Projekten der Welt zählt. Die Benutzer müssen sich hierfür ein SETI-Client auf ihrem Rechner installieren. Dies ist ein Programm in Form eines Bildschirmschoners, welches Daten von einem Server downloadet und diese auswertet (Auswertung der eingehenden Signale vom Radioteleskop). Anschließend werden die Ergebnisse zu dem Server zurückgesendet und neue Daten zum Bearbeiten heruntergeladen.

¹⁸Die vorliegenden Ausführungen sind Vgl. / <http://www.setiathome.de> / entnommen worden.

Damit eine uneingeschränkte Nutzung der freiwilligen Ressourcen zur Verfügung steht, wurde die SETI@HOME – Software für die Teilnehmer inzwischen für ca. 50 Rechnerarten und Betriebssysteme programmiert. Die Software wurde unter anderem für die Betriebssysteme von Windows, als Bildschirmschoner, und von Macintosh als ein performance schwacher Hintergrundprozess bereitgestellt.

	Total	Letzten 24 Stunden
Nutzer	5000947	1421
Empfangene Resultate	1389275089	1456710
Gesamte CPU-Zeit	1931910.006 Jahre	1193.903 Jahre
Floating Piont Operation	5.002358e+21	5.681169e+18
Durchschnittliche Rechendauer pro Work Unit	12 Std. 10 Min 53.5 Sek	7 Std. 10 Min 46.5 Sek

Tabelle 1 Gesamtstatistik bei SETI@HOME, Stand: 22.05.04 [14]

Dabei ist die Software auf allen Betriebssystemen so programmiert, dass sie nur dann gestartet wird, wenn der Rechner des Benutzers ungenutzt ist. Der Vorteil hierbei ist, dass die Anwendungen des Nutzers in keiner Weise beeinträchtigt werden und für die Software stehen somit alle Ressourcen frei zur Verfügung.

Aufgrund dieser Form der Verteilung der Daten an unabhängige Rechner entsteht ein so genannter „virtueller Supercomputer“, der um ein vielfaches effizienter und leistungsstärker ist als ein Großrechner. Einen Eindruck über die Mächtigkeit des „virtuellen Supercomputers“ und den Umfang des SEIT@HOME – Projektes stellt die Tabelle 1 dar.

Datenübermittlung

Der Ablauf der Datenübertragung wird in dem SETI@HOME – Projekt von dem Radioteleskop zum Client in der Abb. näher dargestellt.

Die von dem Radioteleskop empfangenen Signale werden zunächst digital auf einem Band gespeichert. Dabei weisen die erhaltenen Signale meist ein permanentes Rauschen auf, die dadurch keine Komprimierung zulassen. Infolgedessen werden die Daten in 256 Bänder mit je 10kHz zu so genannte „Work Units“ gesplittet, von denen jeder Datensatz 107 Sekunden lang ist und sie sich zudem um ca. 20 Sekunden überlappen. Für ein einzelnes „Work Unit“ ergibt sich somit eine Länge von ca. 350 kB.

Diese Datenpakete werden zu Datenbankservern weitergeleitet, die zur Verwaltung bzw. Speicherung von Informationen der Datenbänder, Teilnehmer und Ergebnisse dienen.

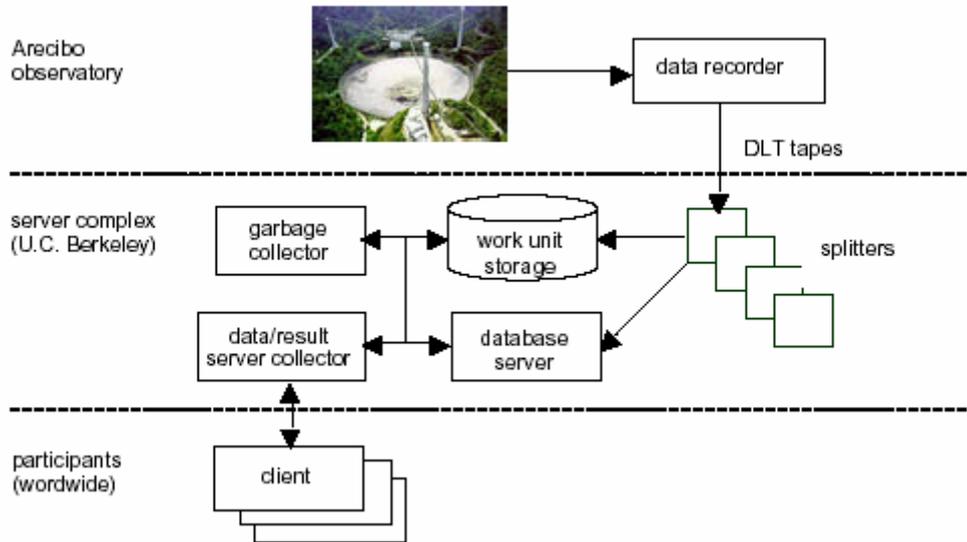


Abbildung 6 Datenübertragung vom Radioteleskop zum Client ; [14]

In diesem Server-Komplex hat der „garbage collector“ die Aufgabe, verbrauchte work units zu löschen. Daher stellt diese Komponente einen gewissen Ausgleich zwischen den Servern und Splittern dar. Hier werden nichtnutzbare Signale schnell gelöscht, um damit die Verarbeitung von neuen work units zu ermöglichen.

Weiterhin ist der so genannte „data/result-server“ für die Verteilung und den Empfang der Datenpakete zuständig. Hierbei sendet der Server die Datenpakete an den jeweiligen Client des Teilnehmers. Nach einiger Zeit erhält der Server das bearbeitete work unit von dem Client zurück und legt es in einer Datenbank ab. Ein spezielles Programm, was in der Lage ist, diese Dateien zu lesen, erstellt daraufhin die Ergebnisse und speichert diese wiederum in einer Datenbank ab.

Daraufhin werden für jedes Ergebnis in einem separaten Server, dem User database, die Informationen über den entsprechenden Teilnehmer, die Anzahl der work units und die jeweiligen Rechenzeiten aktualisiert bzw. gespeichert und an eine Online-Datenbank gesendet. In diesen Servern werden Statistiken über ca. 5 Mio. User geführt.

Des Weiteren scannt ein Programm („redundancy elimination“) die Daten auf redundante Signale bezüglich deren Parameter und deren Inhalte ab. Dabei werden bestimmte Signale herausgefiltert, um ein effizientes Ergebnis für diese work units zu erzielen. Die hier autorisierten Ergebnisse werden in einer Datenbank, der science database, gespeichert.

Dieser soeben beschriebene Ablauf des Empfangs der Datenpakete vom Client über den Server-Komplex bis hin zur Bekanntmachung der Informationen wird in der Abb. 7 grafisch dargestellt.

Der Austausch der Datenpakete zwischen dem Client und dem „data / result“ Server erfolgt über das http – Protokoll. Daher gehört auch das SETI@HOME – Projekt zu dem Bereich „Internet Computing“

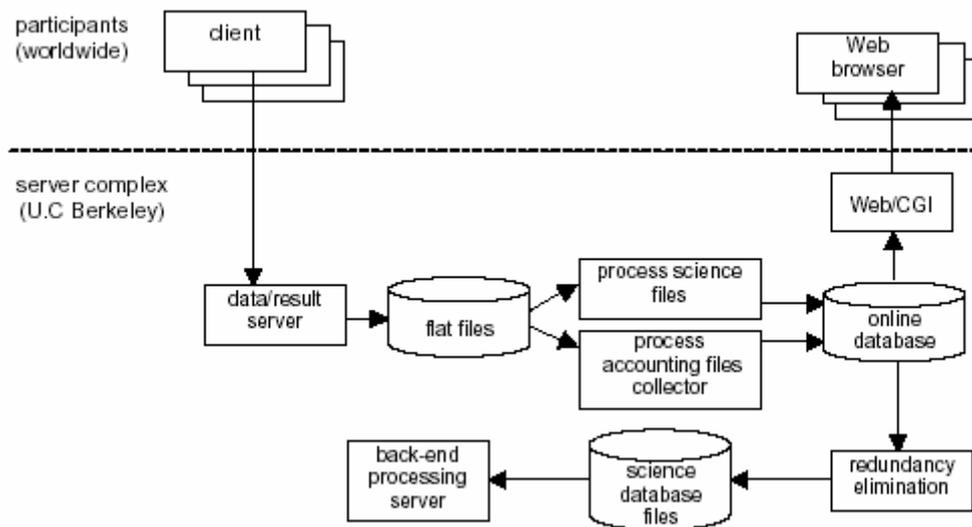


Abbildung 7 Empfang und Bearbeitung der Datenpakete in dem Server – Komplex ; [14]

Sicherheit und Zuverlässigkeit

Das SETI@HOME Projekt verwendet bei dem Empfang und der Verteilung der Daten mehrere Verfahren, um die Sicherheit und die Zuverlässigkeit der Daten zu gewährleisten. Ein Aspekt bei der Verteilung der Datenpakete ist, dass diese zeitlich versetzt an mehrere Teilnehmer weitergeleitet werden. Dadurch wird sichergestellt, dass der „data / result“ – Server die Verarbeitung noch nicht beginnt, wenn ein falsches oder unvollständiges Datenpaket von einem Teilnehmer ankommt.

Ein Teilnehmer ist nicht verpflichtet das empfangene Datenpaket vollständig zu bearbeiten und an den Server zurückzusenden. Der Server wird erst ein Datenpaket als korrekt verabschieden, wenn er mehrere empfangene Resultate erhält.

Darüber hinaus ergibt sich durch diesen Sicherheitsmechanismus ein hoher Grad an Zuverlässigkeit für das Projekt, da es unempfindlicher gegenüber einigen Ausfällen und unabhängiger gegenüber den Resultaten einzelner Rechner wird.

Ein weiterer wichtiger Sicherheitsmechanismus für den Teilnehmer besteht im Verbindungsaufbau zum Server. Der Verbindungsaufbau zwischen dem Client und dem Server wird durch diesen Mechanismus nur durch den Client initiiert und nicht durch den Server. Sollte bei einer Verbindungsanfrage ein Client keine positiven Response vom Server erhalten, so versucht er stündlich eine neue Verbindung herzustellen.

Dies zeigt wiederum, dass einzig und allein nur der Teilnehmer entscheidet, wann er seine Ressourcen für dieses Projekt zur Verfügung stellt.

Erfolge des SETI@HOME – Projekts

Das Ziel dieses Projektes wurde bislang noch nicht erreicht, da alle auffälligen Signale ihren Ursprung auf der Erde hatten. Selbst wenn ein Signal empfangen wird, welches nicht irdische Natur ist, ist allein aufgrund der schwierigen Auswertung der Daten kein Nachweis für die Existenz von außerirdischen Leben gegeben.

Des Weiteren müssten in diesem Fall weltweite Forschungseinrichtungen so gekoppelt werden, dass sie ebenfalls das Signal erhalten und dieses als bedeutsam einschätzen.

Doch bietet das SETI@HOME – Projekt Erfolge in zweifacher Hinsicht: Zum einen hilft es dabei, der Öffentlichkeit den Sinn dieses Projektes zu vermitteln und zu zeigen, auf welcher Art nach außerirdischer Intelligenz gesucht werden kann, z.B. mit Radioteleskopen. Zum anderen ermöglicht es die Auswertung und die Analyse der immensen Datenmengen.

Um die Datenflut, die bei der Bearbeitung der über 100 Millionen Frequenz-Kanäle entsteht, analysieren zu können, werden Rechenkapazitäten und Speichermedien benötigt, die sich ein überwiegend privatwirtschaftlich finanziertes Projekt, wie SETI@home, nicht leisten kann. Dieser finanzielle Nachteil wurde durch die zunehmende Beteiligung der Bevölkerung nahezu ausgeglichen.

5.2 Distributed.net - Projekt

Das Distributed.net – Projekt¹⁹ wurde 1997 gegründet und ist eine Vereinigung zur Forschung in dem Bereich des verteilten Rechnens.

Ähnlich wie bei Seti@Home hat das distributed.net – Projekt das Ziel, Ressourcen von Privatanwendern zu nutzen, welche nur für bestimmte Zeit mit dem Internet verbunden sind.

¹⁹ Die vorliegenden Ausführungen sind Vgl. / <http://www.distributed.net> / entnommen worden

Im Gegensatz zu dem Seti@Home – Projekt, das nach intelligentem Leben mittels Radiosignale sucht, können in diesem Projekt verschiedene Teilprojekte involviert werden, die unterschiedliche Problemstellungen, so genannten generische Aufgaben, bearbeiten. Dies hat den Vorteil, dass gegenüber dem Seti@Home, mehrere Aufgabentypen gleichzeitig bearbeitet werden können.

Des Weiteren ermöglicht dieses Projekt durch die Kompatibilität von unterschiedlicher Hardware und unterschiedlichen Betriebssystemen, dass jeder User an diesem gemeinsamen Projekten teilnehmen kann. Diese Projekte sind oft Wettbewerbe, die von verschiedenen Organisationen ausgeschrieben werden und deren Lösungen mit einem hohen finanziellen Gewinn dotiert sind.

In dem Distributed.net – Projekt ist zurzeit die aktuellste Anwendung das „RC5-72“ Projekt.

Projekt „RC5-72“

Die Firma „RSA – Labs“, die im Bereich der Ver- und Entschlüsselungstechnologie Hard- und Software anbietet, hat das Projekt „RC5 -72“ ins Leben gerufen. Das Ziel dieses Projektes ist es, eine Nachricht mittels des „Brute – Force – Algorithmus“ zu entschlüsseln, die mit dem „RC5 – Verfahren“ mit einer Schlüssellänge von 72 Bit verschlüsselt wurde.

Dabei stellt einerseits der Brute – Force – Algorithmus eine Methode dar, die innerhalb des Suchraumes (2^{72} – Schlüssel) systematisch jeden möglichen Schlüssel ausprobiert. Andererseits ist das RC5 – Verfahren ein Verschlüsselungsmechanismus, der auf dem Blockchiffrierungsalgorithmus beruht, das heißt hier werden Teile vom Klartext mit fester Länge verschlüsselt.

Weiterhin wurde aufgrund der Größe des Suchraumes ein in C++ programmierter Masterserver implementiert, der in der Lage ist, den Suchraum in Blöcke aufzuteilen und diese dann an anfragende Server zu schicken. Die Server wiederum splitten den erhaltenen Suchraum erneut auf und leiten dann die Blöcke an den endgültigen Client weiter.

Wie bei dem Seti@Home – Projekt erhält jeder Teilnehmer ein vom Distributed.net programmiertes Client Programm, das für diverse Plattformen bereitgestellt wird. Dieses Programm läuft als Hintergrundprozess auf dem Rechner ab, so dass der User keine Einschränkungen seiner eigenen Ressourcen durch dieses Programm erhält.

Grundsätzlich ergibt sich für dieses Projekt die Schwierigkeit festzustellen, ob die Nachricht korrekt entschlüsselt wurde, da immer ein Schlüssel zu einer dekodierten Nachricht führt. Aus diesem Grund wurde der Beginn der geheimen Nachricht („The unknown message is:“) für

die Öffentlichkeit zugänglich gemacht. Hätte man keinen Anhaltspunkt auf den Inhalt der Klartextmitteilung, müsste man jedes Ergebnis auf Worte oder Silben in der entsprechenden Sprache prüfen und so entscheiden, ob tatsächlich die gesuchte Nachricht vorliegt.

Der Client bietet daher die Möglichkeit, nach der Bearbeitung des Schlüssels, den dekodierten Text auf Übereinstimmung mit den veröffentlichten Teil der geheimen Nachricht zu überprüfen. Bei erfolgreicher Bearbeitung folgt dann die Ergebnisübergabe vom Client über die entsprechenden Server bis hin zum Masterserver.

Ein Authentifizierungsprotokoll verhindert zudem die Ergebnisübergabe von nicht autorisierten Rechnern.

5.3 LHC – Computing – Grid – Projekt

Das LHC – Computing – Grid – Projekt²⁰ ist ein so genanntes Data Grid. Dabei steht das LHC für Large Hadron Collider und wird derzeit am CERN, der europäischen Organisation für nukleare Forschung in Genf durchgeführt. Das LHC ist ein Forschungsprogramm, indem ein Teilchenbeschleuniger konstruiert wird. Seine Fertigstellung ist für das Jahr 2005 geplant. Das Ziel dieses Forschungsprogramm ist es, den Ursprung der Masse zu untersuchen, um damit die Antworten auf die grundlegendsten Fragen der Wissenschaft zu geben.

Das CERN Institut erwartet im Rahmen seiner Untersuchungen ein sehr hohes Datenaufkommen, das mit Hilfe eines Grids bearbeitet werden soll. Speziell soll hier ein Datennetzwerk, ein so genanntes DataGrid, entstehen, das weltweit freie oder ungenutzte Prozessor- und Speicherkapazitäten von tausenden Computern beinhaltet.

Hierbei ist die Idee, nicht wie bisher Daten zentral zu speichern und zu verarbeiten, sondern auf mehrere Standorte zu verteilen und transparent zu nutzen.

Als Grundlage für diese Anforderungen soll ein hierarchisches System mit verschiedenen Ebenen (Tiers) benutzt werden (siehe Abb. 8).

In der obersten Ebene (Tier – 0) des Systems befindet sich das CERN. Die Aufgabe in dieser Stufe ist die Speicherung und Rekonstruktion von Experimentdaten.

Des Weiteren werden in der zweiten Ebene (regionalen Tier – 1 – Zentren), die Analyse und die Erzeugung simulierter Ereignisse vorgenommen. An dieser Ebene schließt sich dann wiederum eine weitere Hierarchie – Ebene, die nationalen Tier – 2 – Zentren an.

²⁰ Die vorliegenden Ausführungen sind Vgl. / <http://www.lhc.de> / entnommen worden.

Abschließend befinden sich mehrere Institutsrechner in der vierten Ebene (Tier – 3), sowie mehrere Arbeitsplatzrechner in der fünften Ebene (Tier – 4).

Wie bei vielen anderen Projekten in diesem Bereich, entstehen auch bei der Durchführung des LHC-Projektes finanzielle Probleme.

Die 20 CERN – Mitgliedstaaten hatten sich 1996 auf einen Preis von 1,79 Milliarden Euro für den Bau des Projektes geeinigt. Zusätzlich bewilligten sie 150 Millionen Euro für die Nachweisgeräte, Hightech-Detektoren.

Doch aufgrund verschiedener Probleme im Bereich der Verwaltung und Entwicklung, wird das Projekt 570 Millionen Euro teurer als geplant. Dies entspricht fast dem gesamten Jahresbudget der CERN von 650 Millionen Euro, zu dem Deutschland ca. 144 Millionen Euro beiträgt.

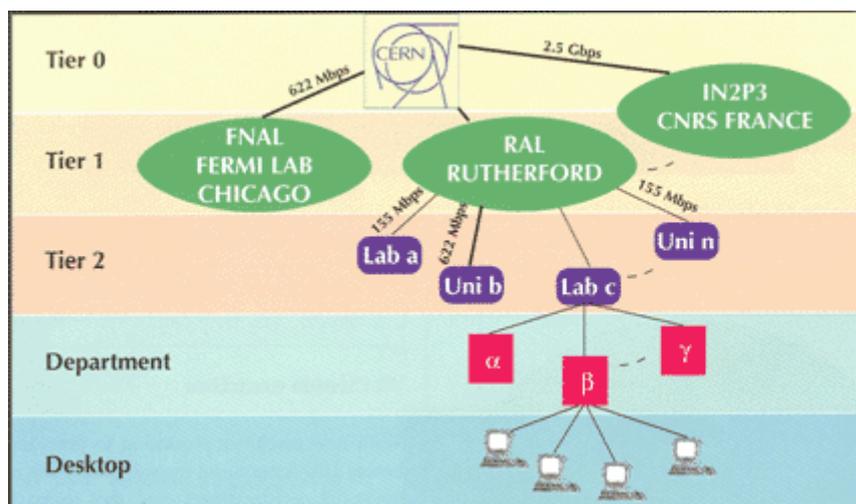


Abbildung 8 Geplantes LHC-Netzwerk am CERN; [21]

6. Ausblick

Das Grid Computing wird als nächster Schritt des Internets angesehen. Das zukünftige Grid System soll ein großes verteiltes und heterogenes System sein, das die Ressourcen in einer Organisation und organisationsübergreifend bündelt und verwaltet.

Das Grid soll Rechenleistung vermitteln und eine landensweite Nutzung ermöglichen, wie beispielsweise das Internet verschiedene Informationen bereit hält und vielfältige Kommunikation erlaubt.

Falls sich die Grid Technologie weiter entwickelt und somit durchsetzt, so ist ein Weg für einen großräumigen Einsatz des Grid in die operativen Bereiche von Unternehmen und Institutionen geschaffen.

Dabei dienen zum Beispiel die „kleinen“ Grid Systeme, wie das aktuellste Projekt, das US – VISIT Programm, als Grundstein und Wegweiser für das zukünftige Grid Computing.

Das US –Visit Programm ist ein Überwachungs- und Datenbanksystem, das bei der Vergabe von Visa und bei der Überwachung an Häfen, Flughäfen und an den 300 US – amerikanischen Grenzübergangsstellen zu Kanada und Mexiko zum Einsatz kommen soll.

Zudem sollen die realen Grenzen mit virtuellen Kontrollen ergänzt werden. Hierbei sollen biometrische und andere Daten bei der Visaerteilung erfasst und verglichen werden. Darüber hinaus sollen an den Grenzübergängen zu Kanada und zu Mexiko die Reisenden einer „Real – Time – Identification“ unterzogen werden, um ihre eindeutige Identifikation zu gewährleisten. Das wäre ein komplexes System, das 20 unterschiedliche Datenbanken mit Informationen von mehr als 300 Millionen Besuchern jährlich verknüpfen soll.

Trotz der konkreten Vorgehensweise müssen jedoch noch viele Probleme bewältigt werden, um ein einfachere Nutzung der Ressourcen zu ermöglichen.

Zum einen entstehen sehr schnell Probleme bei der Realisierung der Implementierungen, da

- eine große Anzahl von unterschiedlichen Systemen bestehen und
- derzeit noch viele offene Probleme bei Standards und Programmierung vorhanden sind.

Zum anderen ist man der Meinung, dass die Zukunft des Grids ebenfalls von der Bandbreite der Internetverbindung und den damit verbundenen Kosten, sowie von der Akzeptanz der Nutzer abhängt.

Eine hohe Bandbreite ist daher wichtig, um eine niedrige Latenzzeit zwischen den einzelnen Rechnern zu erzielen. Zudem muss die Bandbreite erweitert werden, weil die Übertragungsdaten in ihrer Größe und Menge immer mehr zunehmen.

Der ausschlaggebende Faktor für die Richtung des Grid Computing ist immer noch der Mensch. Daher muss das Vertrauen der Rechenzentren und Dienstleistungsanbieter garantiert werden. Doch selbst dann stellt sich die Frage, welcher normale User wird verteilte Ressourcen nutzen, wenn die eigene DV – Infrastruktur über ausreichende Kapazitäten verfügt. Selbst rechenintensive Aufgaben, wie beispielsweise Videokodierung, werden heutzutage von Rechnern mit 2,2 GHz Taktfrequenz in kurzer Zeit bewältigt.

Weiterhin könnte man die Vielzahl der Internetnutzer für das Grid Computing gewinnen, indem ein System bereitgestellt wird, das Rechenleistung als komplette Dienstleistung kostengünstig anbietet.

7. Zusammenfassung

Das Grid Computing ist eine sehr interessante Technologie, die aber noch am Anfang ihrer Entwicklung steht.

Im Rahmen dieser Arbeit wurden daher die wesentlichen Eigenschaften und die Möglichkeiten, die das Grid Computing bietet, dargestellt. Hierzu wurden, um die Möglichkeiten des Grid Computing tiefgründiger zu betrachten, verschiedene Anwendungstypen beschrieben, die mit den unterschiedlichsten Systemkonstellationen agieren können.

Außerdem wurden in dieser Arbeit die Entwicklungsstufen des Grid aufgezeigt, um dem Leser einen Eindruck über die Komplexität des Systems zu vermitteln. Es wurde versucht einen Überblick über die technischen und administrativen Anforderungen für den Betrieb des Grids zu geben.

Ferner wurde die Architektur, mit den einzelnen Schichten und Ressourcen, die innerhalb eines Grids verwendet werden können, beschrieben.

Da das Grid noch in einem sehr frühen Entwicklungsstadium steht, ist es schwierig, eine Aussage über den Erfolg des Grids für die Zukunft zu treffen. Wesentliche Erfolgsfaktoren werden sein, wie man in verteilten Datenbanken ein konsistentes Backup bzw. Restore von Daten gewährleisten kann. Die Grundsteine für das Grid System sind gelegt, welche im zweiten Abschnitt dieser Arbeit verdeutlicht wurden.

Ein kurzer Überblick über die vorhandene Projekte, die das Grid Computing in seinem Ausmaß und seiner Wirkung näher beschreiben, sowie einen Ausblick, über die zukünftige Richtung des Grids, runden diese Arbeit ab.

Literaturverzeichnis

- [1] Berstis V., „Fundamentals of Grid Computing.“ , IBM Redbooks Paper, (2002)
<http://www.redbooks.ibm.com/redpapers/pdfs/redp3613.pdf> , letzter Zugriff: 11.05.04
- [2] Grid Computing Info Centre, <http://www.gridcomputing.com> , letzter Zugriff: 30.05.04
- [3] The Globus Project, <http://www.globus.org> , letzter Zugriff: 5.06.04
- [4] I. Foster, „What ist the Grid?“, <http://www.gridtoday.com/02/0722/100136.html> ,
letzter Zugriff: 12.06.04
- [5] I. Foster, C. Kesselmann, J. Kick, S. Tuecke, „The Physiology of the Grid: An Open Grid
Services Architechure for Distributed Systems Integration.“ June 22, 2002
<http://www.globus.org/research/papers/orgsa.pdf> , letzter Zugriff: 29.05.04
- [6] Ian Foster: „The Anatomy of the Grid: Enabling Scalable Virtual Organizations“.
Software Services Grid Workshop, Danvers, 2001.
<http://swradio.omg.org/workshop/proceedings/Foster.ppt> . , letzter Zugriff: 29.05.04
- [7] Ian Foster et al. ,“The Anatomy of the Grid“
<http://www.globus.org/research/papers.html#anatomy> , letzter Zugriff: 31.05.04
- [8] Herrmann, W. ,“Grid Computing – mehr Vision als Realität. Kommerzielle Nutzung liegt
in weiter Ferner“ , Computerwoche Bd. 29, Nr. 4, S. 12-13, (2002)
- [9] Computer Produktion, Konradin Verlag Robert Kohlhammer, Heft 3, (2002)
- [10] Jen-Yao Chung, „Beyond e-Marketplace & Next Generation e-Business: Grid,
Autonomic Computing & Web Services“.
4th International Conference on Electronic Commerce, Hong Kong, October 2002.
<http://www.eti.hku.hk/eti/web/download/GACWS2002.pdf> ,
letzter Zugriff: 14.04.04

- [11] Vgl. Abbildung / http://www.owenwalcher.com/grid_networking.htm /,
letzter Zugriff: 20.05.04
- [12] Vgl. Abbildung / http://www.globus.org/ogsa/releases/final/docs/infosvcs/indexsvc_overview_files/image002.gif / , *letzter Zugriff: 5.05.04*
- [13] Vogel, H.-R. „KES – Die Zeitschrift für Informations-Sicherheit, Rechenraster. Sicherheit beim Grid Computing“ Bd. 19, Nr.5, S. 6 – 8, 10 – 12, (2002)
- [14] Okasha S., „Security mechanisms for Grid Computing“, Seminar WS 2002 - 2003
- [15] Ian Foster, Carl Kesselman, Gene Tsudik, Steven Tuecke „A Security Architecture for Computational Grids“.
- [16] Was ist SETI@Home ?
<http://home.t-online.de/home/heider.home/c59seti.htm> , *letzter Zugriff: 3.06.04*
- [17] Homepage des Seti@home-Projekts
<http://setiathome.ssl.berkeley.edu/totals> , *letzter Zugriff: 3.06.04*
- [18] Hipschman :Radioteleskop;
Ron Hipschman: SETI@home.
http://www.alein.de/seti/radio_search_3_german.htm , *letzter Zugriff: 3.06.04*
- [19] Silvia Michel, „Wie funktioniert SETI@home?“
http://www.alein.de/seti/about_seti_at_home_2_german.htm , *letzter Zugriff: 4.06.04*
- [20] Report of the Steering Group of the LHC Computing Review, CERN/LHCC/2001-004,
http://lhc-computing-review-public.web.cern.ch/lhc-computing-review-public/Public/Report_final.pdf , *letzter Zugriff: 4.06.04*
- [21] Vgl. Abb. / http://sbnt.jinr.ru/From_the_Web_to_the_Grid.html /
,letzter Zugriff: 15.05.04

[22] Ralf Grohmann und Dr. Sebastian Wedeniwski; Grid Computing, Die IBM
ZetaGRID Lösung,
http://www.zetagrid.net/zeta/ZetaGRID4customers_de.pdf , letzter Zugriff: 14.05.04