



BERGISCHE UNIVERSITÄT WUPPERTAL  
FACHBEREICH MATHEMATIK  
UND NATURWISSENSCHAFTEN  
FACHGRUPPE PHYSIK

**AMANDA und DØ als Testumgebung  
für das LHC Computing Grid**

**Dissertation  
von  
Torsten Harenberg**

August 2005  
WUB-DIS 2005-09

Diese Dissertation kann wie folgt zitiert werden:

urn:nbn:de:hbz:468-20050739

[<http://nbn-resolving.de/urn/resolver.pl?urn=urn%3Anbn%3Ade%3Ahbz%3A468-20050739>]

# Inhaltsverzeichnis

<b>Einleitung</b>	<b>1</b>
<b>1 Physikalischer Rahmen</b>	<b>3</b>
1.1 Das Standardmodell der Teilchenphysik . . . . .	3
1.2 Elektroschwache Wechselwirkung . . . . .	4
1.3 Starke Wechselwirkung . . . . .	6
1.4 Experimentelles Umfeld . . . . .	7
<b>2 Die kosmische Strahlung und der AMANDA-Detektor</b>	<b>12</b>
2.1 Hintergrund . . . . .	12
2.2 Das Messprinzip . . . . .	15
2.3 Der Detektor . . . . .	17
2.4 Die AMANDA-Simulationssoftware . . . . .	18
<b>3 Das Tevatron und das DØ-Experiment</b>	<b>22</b>
3.1 Der Tevatron-Beschleuniger . . . . .	22
3.2 Der DØ-Detektor . . . . .	24
3.2.1 Das DØ-Koordinatensystem . . . . .	25
3.2.2 Der innere Detektor . . . . .	27
3.2.3 Das Luminositätssystem . . . . .	27
3.2.4 Der Silizium- und Faser-Detektor . . . . .	28
3.2.5 Das Kalorimeter . . . . .	29
3.2.6 Das Myonsystem . . . . .	31
3.2.7 Die Simulation . . . . .	33

---

<b>4</b>	<b>Einführung in das Grid-Computing</b>	<b>35</b>
4.1	Definition . . . . .	35
4.2	Abgrenzung zu existierenden Technologien . . . . .	36
4.3	Existierende Grid-Implementierungen . . . . .	37
4.3.1	Das Globus-Toolkit . . . . .	39
4.3.2	Das European Data Grid Projekt . . . . .	40
<b>5</b>	<b>Erstellung einer Grid-basierten Softwareinfrastruktur für AMANDA</b>	<b>50</b>
5.1	Einleitung . . . . .	50
5.2	Aufbau des Grid-Projektes für AMANDA . . . . .	51
5.3	Beschreibung des gewählten Ansatzes . . . . .	53
5.3.1	Softwarepakete für die AMANDA-Software . . . . .	54
5.3.2	Der Sandbox-Mechanismus . . . . .	55
5.3.3	Erweiterung für beliebige Software . . . . .	57
5.3.4	Die graphische Benutzeroberfläche . . . . .	58
5.3.5	Zusammenfassung . . . . .	61
5.4	Flüsse von atmosphärischen Leptonen in AMANDA . . . . .	63
5.4.1	Hintergrund . . . . .	63
5.4.2	Simulation . . . . .	63
5.4.3	Messung der Spektren von Myonen, Myon- und Elektron- neutrinos . . . . .	66
5.4.4	Zusammenfassung . . . . .	71
<b>6</b>	<b>Integration des DØ-Experiments in das European Data Grid</b>	<b>72</b>
6.1	Einleitung . . . . .	72
6.2	Aufbau einer EDG-Infrastruktur . . . . .	73
6.3	Anpassung der DØ-Software . . . . .	74
6.3.1	DØ-spezifische Infrastruktur . . . . .	74
6.3.2	Anpassung der DØ-Software . . . . .	75
6.3.3	Überwachung und Buchführung . . . . .	77
6.3.4	Ablaufsteuerung einer DØ-Rechenanforderung . . . . .	80
6.4	Ergebnisdiskussion . . . . .	81
6.4.1	Stabilitäts- und Funktionalitätsprüfungen . . . . .	82

---

6.4.2	Entwicklung eines ausfallsicheren Ressource Brokers . . . .	84
6.4.3	Entwicklung einer webbasierten graphischen Benutzeroberfläche . . . . .	87
6.4.4	Zusammenfassung . . . . .	88
	<b>Zusammenfassung und Ausblick</b>	<b>93</b>
	<b>Literaturverzeichnis</b>	<b>95</b>
	<b>Abbildungsverzeichnis</b>	<b>101</b>
	<b>Tabellenverzeichnis</b>	<b>104</b>
	<b>Glossar</b>	<b>105</b>



# Einleitung

Die Suche nach den Grundbausteinen der Materie und nach den Kräften, die zwischen ihnen wirken, ist das Forschungsgebiet der Hochenergiephysik. Mit immer größer werdenden Teilchenbeschleunigern erreicht man immer tiefer gehende Erkenntnisse und dringt dabei in Energiebereiche vor, wie sie kurz nach dem Urknall herrschten. Im Moment ist das Tevatron am Fermilab der energiereichste Teilchenbeschleuniger der Welt mit einer Schwerpunktsenergie von  $\sqrt{s} = 1,96 \text{ TeV}$ . Um die bei diesen hohen Energien entstehende Anzahl von Teilchen zu messen, sind auch immer höher auflösende Detektoren nötig. Dem wurde dadurch Rechnung getragen, dass mit der Steigerung der Schwerpunktsenergie am Tevatron auch die Detektoren DØ und CDF<sup>1)</sup> teilweise erneuert wurden.

Als weitere Folge dieser Entwicklung steigen die Datenmengen, die die Detektoren liefern und die verarbeitet werden müssen, stetig an. Und damit einhergehend wächst der Rechenzeitbedarf, der für die Simulation und die Rekonstruktion von Ereignissen benötigt wird.

Auch in der Astroteilchenphysik sind ähnliche Tendenzen erkennbar. Gerade in der Neutrino-Astronomie instrumentiert man immer größere Volumen, um die nur schwach wechselwirkenden Neutrinos nachzuweisen. So wird am geographischen Südpol zur Zeit als Nachfolger des AMANDA<sup>2)</sup>-Experiments das IceCube-Experiment aufgebaut, bei dem ein Kubikkilometer Eis mit Photomultipliern instrumentiert wird. Auch hier wird die Datenrate stark ansteigen.

Man hat erkannt, dass dieser steigende Bedarf an Rechenzeit und Speicherkapazität nicht mehr alleine durch das jeweilige Labor, bei dem das Experiment betrieben wird, gedeckt werden kann. Vielmehr sind diese Anforderungen nur noch im weltweiten Verbund lösbar. Daher wurden und werden Techniken entwickelt, die unter dem Begriff „Grid-Computing“ zusammengefasst werden. Sie werden für die zukünftigen Experimente grundlegend sein.

Die vorliegende Arbeit bearbeitet zwei Themenkreise, deren Reihenfolge einer gewissen Willkür unterliegen. Zum einen soll sie den physikalischen Hintergrund beleuchten und damit das Grid-Computing motivieren. Zum anderen

---

<sup>1)</sup>CDF = Collider Detector at Fermilab

<sup>2)</sup>AMANDA = Antartic Muon and Neutrino Detector Array

soll sie eine Einführung in das komplexe Gebiet des Grid-Computings geben. Es bleibt daher dem Leser überlassen, je nach persönlicher Neigung die Reihenfolge der Kapitel beim Lesen zu vertauschen.

Die hier gewählte Gliederung beginnt mit einer Einleitung in den physikalischen Rahmen und damit mit einer Motivation für die Notwendigkeit zur Anwendung neuer Technologien im Bereich des Computings. Danach werden die Experimente, in deren Rahmen diese Arbeit erstellt wurde, vorgestellt. Es folgt eine Einleitung in das Grid-Computing. Da es dabei nicht eine einzige, allgemein gültige und einheitliche Technik gibt, werden verschiedene Initiativen, die sich mit der Entwicklung von Techniken des Grid-Computings befassen, vorgestellt. Insbesondere das Globus-Toolkit sowie das darauf aufbauende „European Data Grid“ werden genauer vorgestellt.

Die folgenden zwei Kapitel befassen sich dann mit der Anwendung von Techniken des Grid-Computings, zum einen in der Astroteilchenphysik im Rahmen des AMANDA-Experiments, zum anderen im Rahmen des DØ-Experiments. Beide sind laufende Experimente mit aktiver Datennahme, weshalb ihre Simulations- und Rekonstruktionssoftware nicht für den Einsatz in einem weltweiten Rechnernetz konzipiert wurden. Es wird gezeigt, dass es trotzdem ohne Eingriff in die experimentenspezifische Software möglich ist, Grid-Technologien für diese Experimente zu nutzen und damit zum einen einen signifikanten Beitrag zum erhöhten Rechenzeitbedarf der Experimente zu leisten. Zum anderen (und dies ist je nach Sichtweise der wichtigere Aspekt) wird evaluiert, inwieweit die Rechenverbände im Hinblick auf Stabilität, Skalierbarkeit und Reproduzierbarkeit schon heute den Anforderungen von laufenden Experimenten gewachsen sind. Damit sind Schlussfolgerungen für die weitere Entwicklung in Richtung LHC möglich.

Ein Glossar zum Ende der Arbeit soll als Nachschlagewerk für die vielen Fachbegriffe dienen, denen der Leser immer wieder begegnet.

# Kapitel 1

## Physikalischer Rahmen

### 1.1 Das Standardmodell der Teilchenphysik

Die Teilchenphysik beschäftigt sich mit den Grundbausteinen der Materie und den Kräften, die zwischen ihnen wirken. In der Natur beobachtet man vier fundamentale Wechselwirkungen: die elektromagnetische, die schwache und die starke Kraft sowie die Gravitation. Die Gravitation ist um viele Größenordnungen schwächer als die anderen Kräfte und wird vom Standardmodell nicht beschrieben.

Die Kräfte werden durch Teilchen mit ganzzahligem Spin übertragen: den Bosonen. Die Wechselwirkungen werden durch Eichtheorien mit zugrundeliegender Symmetriegruppe beschrieben. Die Bosonen nennt man daher auch Eichbosonen. Die relative Stärke der Kräfte, ihre zugrundeliegende Theorie sowie deren Eichbosonen fasst Tabelle 1.1 zusammen.

Kraft	relative Stärke	Theorie	Eichboson
stark	1	Quantenchromodynamik (QCD)	8 Gluonen
elektromagnetisch	$10^{-3}$	elektroschwache Theorie	Photon
schwach	$10^{-14}$		$W^{\pm}, Z$
Gravitation	$10^{-43}$	allgemeine Relativitätstheorie	Graviton (?)

Tabelle 1.1: Die fundamentalen Kräfte der Natur. Das Graviton wurde bisher noch nicht beobachtet.

Die Materie setzt sich aus Teilchen mit halbzahligen Spin, den sogenannten Fermionen zusammen. Diese unterteilen sich weiter in Quarks und Leptonen und erscheinen in drei Familien. Alle elementaren Fermionen nehmen an der schwachen Kraft teil, die elektrisch geladenen unter ihnen auch an der elektromagnetischen. Quarks tragen zusätzlich die sogenannte Farbladung. Sie sind die einzigen Fermionen, die an der starken Kraft teilnehmen. Tabelle 1.2 fasst die Fermionen, ihre Wechselwirkungen und ihre Quantenzahlen zusammen.

	Generation			Wechselwirkungen	Y	I	I <sub>3</sub>	Q
	1.	2.	3.					
Leptonen	$\begin{pmatrix} \nu_e \\ e \end{pmatrix}_L$	$\begin{pmatrix} \nu_\mu \\ \mu \end{pmatrix}_L$	$\begin{pmatrix} \nu_\tau \\ \tau \end{pmatrix}_L$	schwach	-1	1/2	1/2	0
	$e_R$	$\mu_R$	$\tau_R$	elektromagnetisch, schwach	-2	0	-1/2	-1
				elektromagnetisch, schwach			0	-1
Quarks	$\begin{pmatrix} u_i \\ d'_i \end{pmatrix}_L$	$\begin{pmatrix} c_i \\ s'_i \end{pmatrix}_L$	$\begin{pmatrix} t_i \\ b'_i \end{pmatrix}_L$	stark, schwach, elektromagn.	1/3	1/2	1/2	2/3
	$u_{i,R}$	$c_{i,R}$	$t_{i,R}$		4/3	0	0	2/3
	$d_{i,R}$	$s_{i,R}$	$b_{i,R}$		-2/3	0	0	-1/3

Tabelle 1.2: Elementare Fermionen, beteiligte Wechselwirkungen und Quantenzahlen (Hyperladung  $Y$ , schwacher Isospin  $I$  und elektrische Ladung  $Q = I_3 + Y/2$ ). Die Quantenzahlen der Antiteilchen haben umgedrehte Vorzeichen. Der Index  $i$  läuft über die drei Farbzustände der QCD und die gestrichelten Quarks deuten an, dass diese Zustände gemischte Zustände der starken Eigenzustände  $d$ ,  $s$  und  $b$  sind.

## 1.2 Elektroschwache Wechselwirkung

Die elektromagnetische und die schwache Wechselwirkung werden in der elektroschwachen Theorie zusammengefasst [1–3]. Ihr liegt die Symmetriegruppe  $SU(2) \otimes U(1)$  zu Grunde. Sie wird nach ihren Begründern Glashow, Weinberg und Salam auch als GWS-Theorie bezeichnet. In Abbildung 1.1 sind die fundamentalen Feynman-Graphen dieser Theorie abgebildet.

In dieser Theorie existieren das masselose Isospin-Triplett  $W_\mu^1, W_\mu^2, W_\mu^3$  und das masselose Isospin-Singlett  $B_\mu$  als Eichbosonen. Die experimentelle Beobachtung zeigt aber, dass die  $W^\pm$ -Bosonen und das Z-Boson ( $M_{W^\pm} \approx 80 \text{ GeV}$ ,  $M_Z \approx 91,2 \text{ GeV}$  [4]) massebehaftet sind. Um diesen Widerspruch aufzulösen, führt man in der

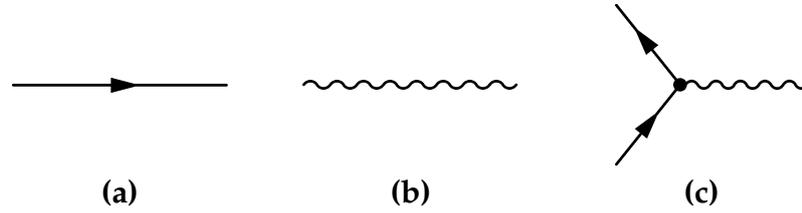


Abbildung 1.1: Fundamentale Feynman-Graphen der Quantenelektrodynamik (QED): (a) Fermionpropagator, (b) Photonpropagator und (c) Wechselwirkungsvertext

GWS-Theorie den Mechanismus der spontanen Symmetriebrechung ein. Diese erfordert die Existenz eines skalaren Higgsdubletts

$$\phi(x) = \frac{1}{\sqrt{2}} \begin{pmatrix} \phi^+ \\ \phi^0 \end{pmatrix} \quad (1.1)$$

Die physikalisch beobachtbaren Eichbosonen sind dann Linearkombinationen der ursprünglichen Eichbosonen:

$$\gamma = B_\mu \cos \Theta_W + W_\mu^3 \sin \Theta_W \quad (1.2)$$

$$Z = -B_\mu \sin \Theta_W + W_\mu^3 \cos \Theta_W \quad (1.3)$$

$$W^\pm = \sqrt{1/2} (W_\mu^1 \mp iW_\mu^2) \quad (1.4)$$

Dabei wird  $\Theta_W$  als schwacher Mischungswinkel oder auch WEINBERG-Winkel bezeichnet. Die Massen der drei realen Eichbosonen ergeben sich zu:

$$M_Z^2 = \frac{e^2 \cdot v^2}{4 \cdot \cos^2 \Theta_W \cdot \sin^2 \Theta_W} \quad (1.5)$$

$$M_W^2 = \frac{e^2 \cdot v^2}{4 \cdot \sin^2 \Theta_W} \quad (1.6)$$

$$M_\gamma = 0 \quad (1.7)$$

Dabei bezeichnet  $e$  die Elementarladung und  $v$  den Vakuumerwartungswert des Higgsfeldes.

Damit erhält man – wie experimentell beobachtet – drei massive Bosonen der schwachen Wechselwirkung mit kurzer Reichweite und ein masseloses Boson der elektromagnetischen Wechselwirkung mit unendlicher Reichweite. Das verbleibende neutrale, skalare Higgs Teilchen  $H^0$  ist bisher experimentell noch nicht nachgewiesen worden.

### 1.3 Starke Wechselwirkung

Die starke Wechselwirkung wird mit der Theorie der Quantenchromodynamik (QCD) beschrieben, einer nichtabelschen Quantenfeldtheorie, die nach dem Vorbild der Quantenelektrodynamik (QED) entwickelt wurde. Ihr liegt die Symmetriegruppe  $SU(3)$  zu Grunde. Die Eichbosonen der QCD sind acht masselose Gluonen, die an die Farbladung der Quarks koppeln. Die Farbe ist eine Quantenzahl, die für Quarks die Eigenschaften *rot*, *grün* und *blau* annehmen kann (Antiquarks tragen Antifarbe). Gluonen tragen Farbe und Antifarbe. Somit kann Gluonenselbstkopplung auftreten, was die QCD von der QED unterscheidet. Abbildung 1.2 zeigt die fundamentalen Feynman-Graphen der QCD.

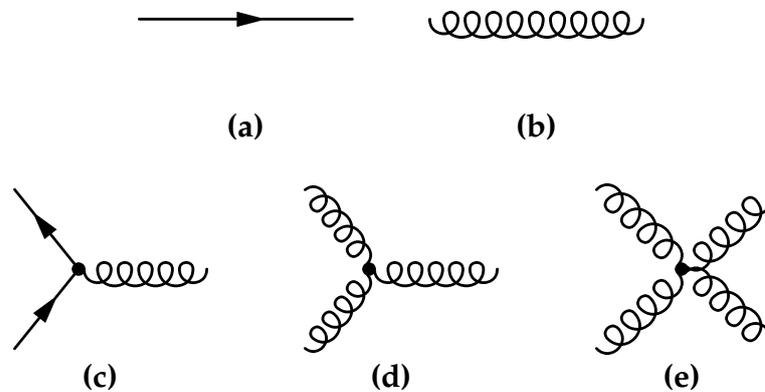


Abbildung 1.2: Fundamentale Feynman-Graphen der QCD: (a) Quarkpropagator, (b) Gluonpropagator und (c) Quark-Gluon-Vertex, (d) Drei-, (e) Vier-Gluon-Vertex.

Experimentell wird die Existenz der Farbladung am deutlichsten durch Messungen des R-Verhältnisses in der  $e^+e^-$ -Streuung

$$R = \frac{\sigma(e^+e^- \rightarrow \bar{q}q)}{\sigma(e^+e^- \rightarrow \mu^+\mu^-)} \quad (1.8)$$

bestätigt, die gezeigt hat, dass das Verhältnis sehr gut mit

$$R = \frac{\sigma(e^+e^- \rightarrow \bar{q}q)}{\sigma(e^+e^- \rightarrow \mu^+\mu^-)} = 3 \sum_q e_q^2 \quad (1.9)$$

übereinstimmt.

Weitere experimentelle Bestätigungen der QCD betreffen unter anderem die Entdeckung der Gluon-Bremsstrahlung durch den Nachweis von 3-Jet-Ereignissen [5], die Messung der Casimir-Operatoren  $C_A$  und  $C_F$  (zum Beispiel [6, 7]), sowie die Verifizierung der vorhergesagten Energieabhängigkeit der starken Kopplung [8].

## 1.4 Experimentelles Umfeld

Bei heutigen und zukünftigen Beschleunigern wie dem Tevatron (vergleiche Kapitel 3 und dem Large Hadron Collider LHC am CERN streuen Protonen beziehungsweise Antiprotonen aneinander. Bei diesen Streuprozessen dominiert die starke Wechselwirkung, so dass sie mit Hilfe der Quantenchromodynamik analysiert werden müssen.

Bei der Proton-Proton-Streuung können verschiedene Prozesse auftreten. Welche das im Einzelnen sind, hängt von der Energie, dem übertragenen Impuls und den Wirkungsquerschnitten der Einzelprozesse ab.

Bei der elastischen Streuung bleibt das stoßende Proton erhalten. Es wird lediglich leicht abgelenkt, wobei nur wenig Impuls übertragen wird. Bei der inelastischen Streuung hingegen bricht die Proton-Struktur auf und es entstehen durch Hadronisierung neue Teilchen. Ist der Prozess mit einem hohen Impulsübertrag verbunden, wechselwirken praktisch die einzelnen Partonen der stoßenden Protonen miteinander und werden in großem Winkel zur Strahlrichtung abgelenkt. Durch Hadronisierung erzeugen sie Bündel von Teilchen mit einer gemeinsamen Richtung, die *Jets* genannt werden. Die Analyse von Jets mit hohem Transversalimpuls lässt Rückschlüsse auf die Ausdehnung der Quarks zu. Die QCD sagt diese als punktförmig voraus, so dass Abweichungen hiervon auf eine Substruktur hindeuten würden.

Die meisten Prozesse sind allerdings solche mit geringem Impulsübertrag. Die Erzeugung von schweren Quarks oder Jets mit hohem Transversalimpuls ist selten und auch die Erzeugung von bisher nicht experimentell nachgewiesenen Teilchen wie dem Higgs-Boson findet nach den theoretischen Voraussagen mit geringer Wahrscheinlichkeit statt.

In Abbildung 1.3 sind die voraussichtlichen Wirkungsquerschnitte bzw. Ereignisraten für einige der bei Proton-Proton-Streuung stattfindenden Prozesse aufgetragen.

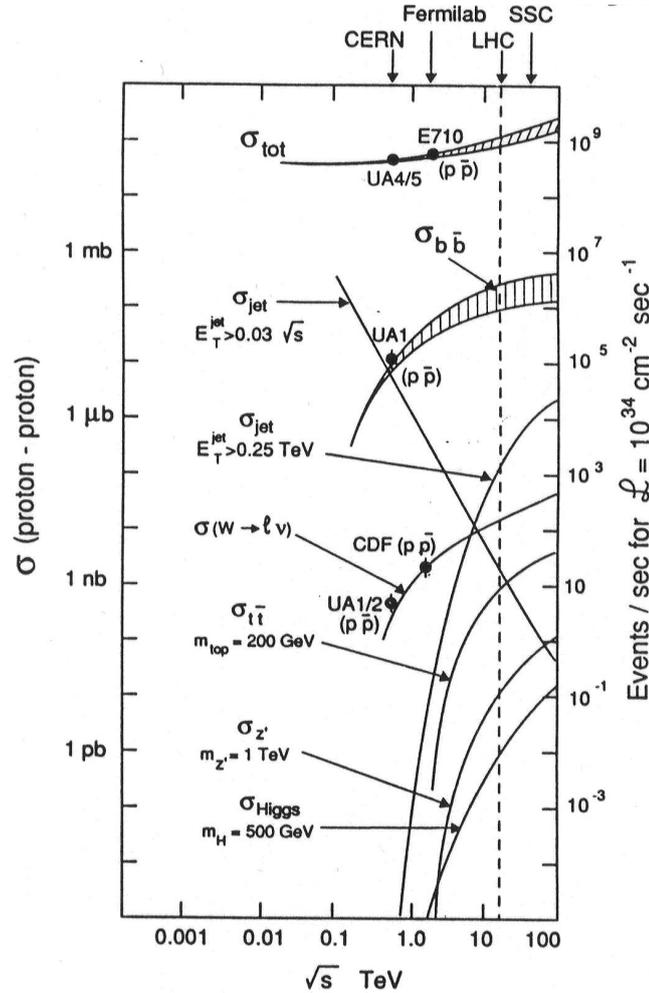


Abbildung 1.3: Wirkungsquerschnitte für die Proton-Proton-Streuung [9]

Man erkennt, dass der totale Wirkungsquerschnitt zum Beispiel bei voller LHC-Energie etwa 100 mb beträgt. Das heißt, es finden etwa  $10^9$  Proton-Proton-Stöße pro Sekunde statt. Der theoretisch vorhergesagte Wirkungsquerschnitt für physikalisch interessante Prozesse wie beispielsweise für die Produktion eines Higgs-Bosons beträgt aber bei einer angenommenen Masse des Higgs von 500 GeV lediglich 1 pb. Das heißt, dass nur eines von  $10^{11}$  Ereignissen die Produktion eines Higgs sein könnte. Daher müssen die Daten der Experimente entsprechend gefiltert werden. Auch wenn dies schon zum Zeitpunkt der Datennahme im ge-

wissen Umfang passiert, so sind die entstehenden Datenmengen immer noch enorm und der Rechenzeitaufwand entsprechend groß.

Abbildung 1.4 zeigt ein Ereignis des DØ-Experiments (vergleiche Kapitel 3) und Abbildung 1.5 zum Vergleich einen simulierten Luftschauder.

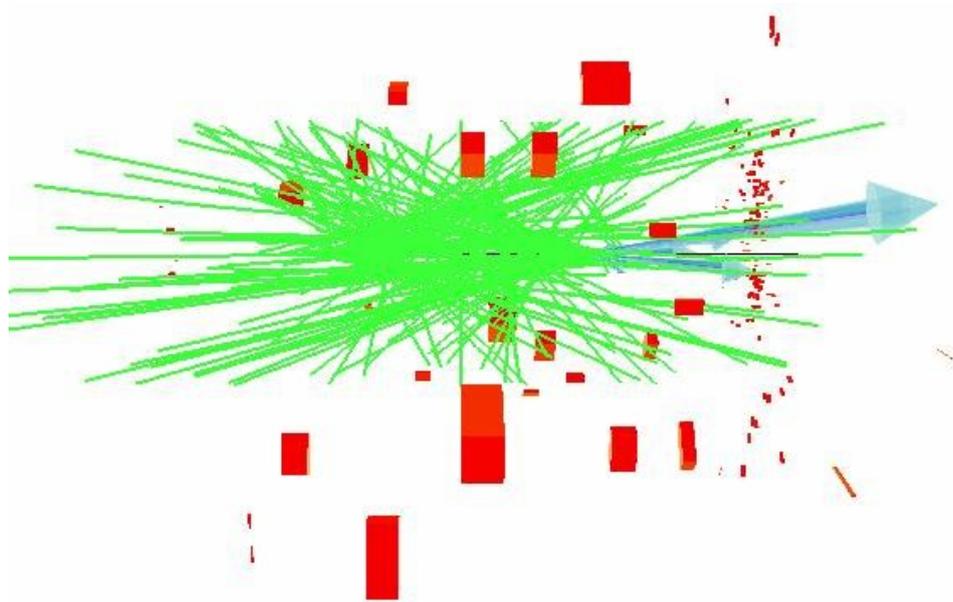


Abbildung 1.4: Beispiel eines DØ-Ereignisses (Run 172482 Event 5512259). Aus [10].

Man erkennt im Fall des DØ-Ereignisses die rekonstruierten Spuren in grün und die Treffer im Kalorimeter (also die Energiedeposition) in rot. Im Luftschauderbild ist ein Proton mit 100 GeV Energie simuliert. Die roten Spuren geben die Elektronen, Positronen oder Gammas wieder, grüne Spuren symbolisieren Myonen und blaue Spuren Hadronen. In beiden Bildern erkennt man die hohe Anzahl von Teilchenspuren, die durch Computeralgorithmen rekonstruiert werden müssen. Diese Rekonstruktion sowie die Simulation von Teilchendurchgängen durch Materie sind die zeitaufwändigsten Rechenaufgaben, die von den Kollaborationen dieser Experimente bewältigt werden müssen. Es ist einleuchtend, dass diese Aufgaben sowohl in der Hochenergiephysik als auch in den Experimenten der Astroteilchenphysik mit identischen Methoden gelöst werden können. Daher wurden für diese Arbeit das DØ-Experiment (siehe Kapitel 3) und das AMANDA-Experiment (welches in Kapitel 2.3 beschrieben ist) ausgewählt.

Bei den zukünftigen LHC-Experimenten werden dann erschwerend auch sehr hohe Datenraten erwartet. Das ATLAS-Experiment liefert beispielsweise Daten mit einer Rate von 200 Hz bei einer Größe von 1,6 Megabyte pro Event [12]. So

müssen pro Tag ungefähr 26 Terabyte an Daten gespeichert und gefiltert werden. Abbildung 1.6 zeigt den steigenden Bedarf an Rechenzeit<sup>1)</sup> und Speicherplatz am zentralen Analysezentrum („Tier-0“ genannt) vom ATLAS am CERN.

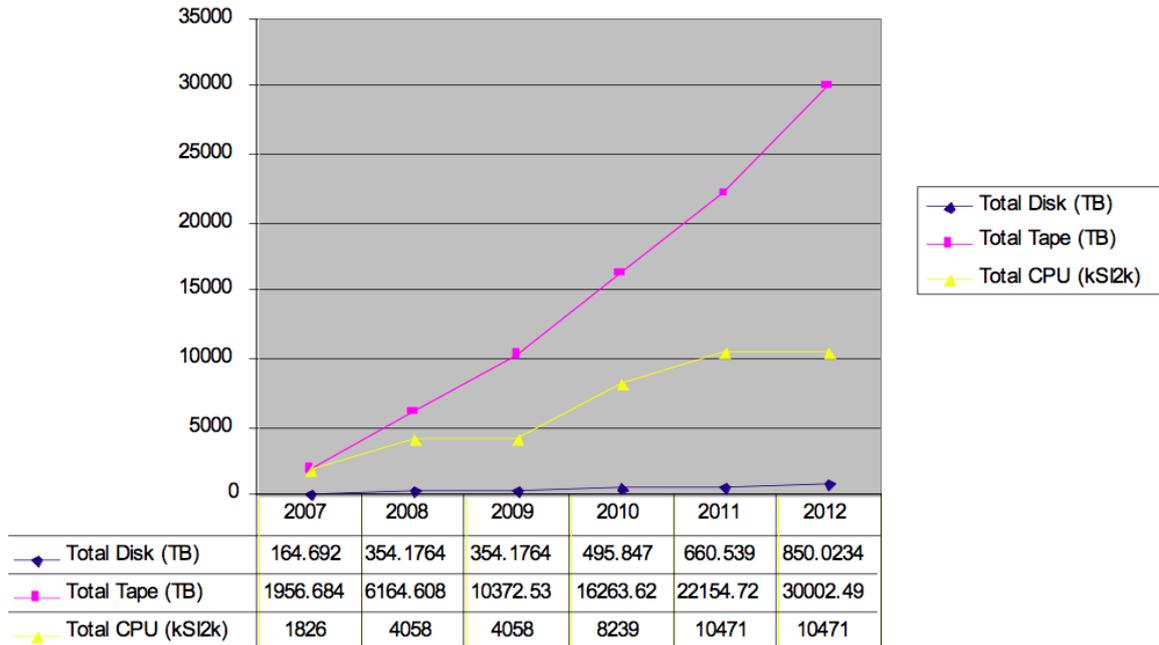


Abbildung 1.6: Zeitliche Entwicklung des Ressourcenbedarfs am zentralen ATLAS-Analysezentrum. Aus [12].

Dabei sind nur bestimmte Rechenaufgaben, die im Zusammenhang mit dem ATLAS-Experiment entstehen, von diesem zentralen Analysezentrum abgedeckt. So werden fast alle Simulationen dezentral in einem weltweit verteilten Rechenverbund im Rahmen des Grid-Computings durchgeführt.

<sup>1)</sup>Rechenzeit ist hier in kSI2k angegeben. SI2k ist die Kurzschreibweise für „SpecInt2000“, einem Benchmark der „Standard Performance Evaluation Corporation“ für Integer-Operationen. Ein zur Zeit aktueller PC hat eine Leistung von grob einem kSI2k.

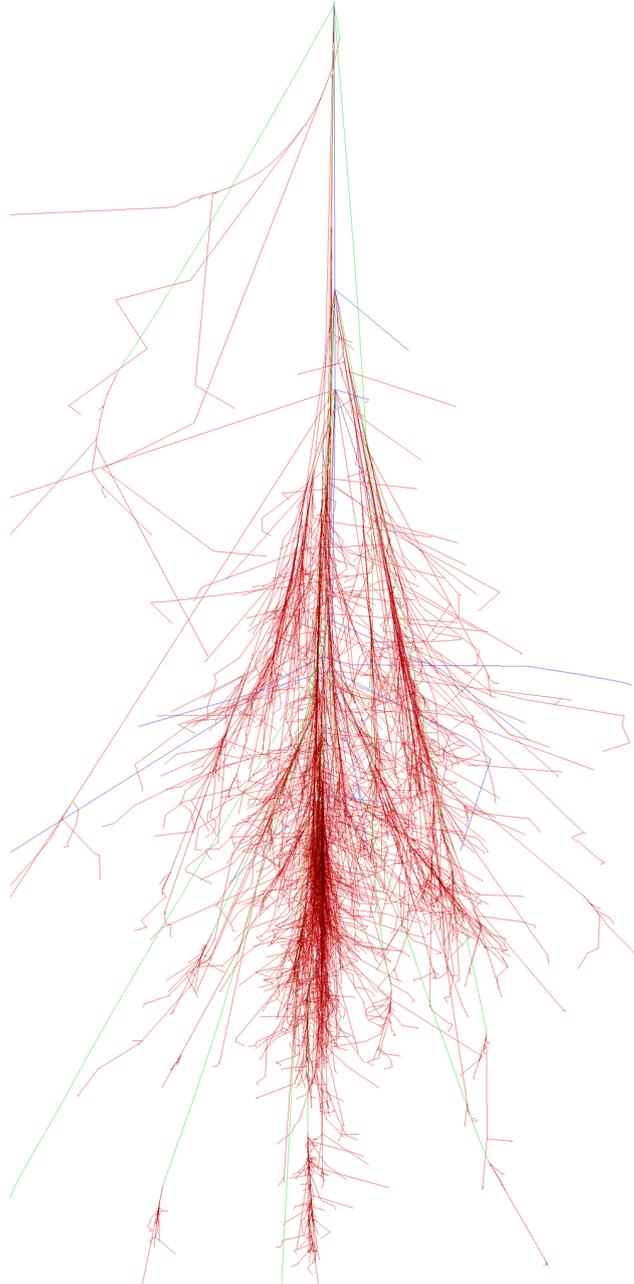


Abbildung 1.5: Beispiel eines mit CORSIKA simulierten Luftschauers. Simuliert wurde ein 100 GeV Proton, welches unter einem Zenitwinkel von Null Grad auf die Erdatmosphäre trifft. Aus [11].

# Kapitel 2

## Die kosmische Strahlung und der AMANDA-Detektor

### 2.1 Hintergrund

Obwohl die kosmische Strahlung von Victor Hess bereits 1912 bei Versuchen mit bemannten Ballonfahrten entdeckt wurde [13], ist das Spektrum eben jener kosmischen Strahlung bis heute nicht komplett verstanden. Abbildung 2.1 zeigt das Spektrum der kosmischen Strahlung mit dem Messbereich verschiedener Experimente.

Dabei besteht die kosmische Strahlung überwiegend aus Protonen ( $\approx 90\%$ ) und zu kleineren Anteilen aus  $\alpha$ -Teilchen ( $\approx 9\%$ ) und Ionen schwerer Nuklide. Der integrale Fluss über alle Teilchen und Energien kann man mit  $1000 \text{ m}^{-2}\text{s}^{-1}$  annehmen [15]. Das Spektrum wird recht gut durch ein gebrochenes Potenzgesetz beschrieben:

$$\frac{dN_{prim}}{dE} \propto E^{-\gamma}, \text{ mit } \gamma = \begin{cases} 2.7 & \text{für } E \leq 4 \text{ PeV} \\ 3.0 & \text{für } 4 \text{ PeV} \leq E \leq 3 \cdot 10^{19} \text{ eV} \\ 2.7 - 3.0 & \text{für } E \geq 3 \cdot 10^{19} \text{ eV} \end{cases} \quad (2.1)$$

Das Spektrum wird bei  $3 \cdot 10^6 \text{ GeV}$  steiler (dieser Knick wird als „Knie“ – oder englisch „knee“ – bezeichnet) und bei gegen  $3 \cdot 10^9 \text{ GeV}$  wieder flacher (was als „Knöchel“ – oder englisch „ankle“ – bezeichnet wird). Den entstehenden drei Bereichen ordnet man verschiedene Entstehungsprozesse zu. Unter  $100 \text{ MeV}$  blockiert der Sonnenwind Protonen, die von außerhalb unseres Sonnensystems kommen.

Bei  $10^{19} \text{ eV}$  müsste eigentlich aufgrund von Wechselwirkungen mit der kosmischen Hintergrundstrahlung der so genannte GZK-Cutoff auftreten, zu dem jedoch widersprüchliche Messdaten vorliegen.

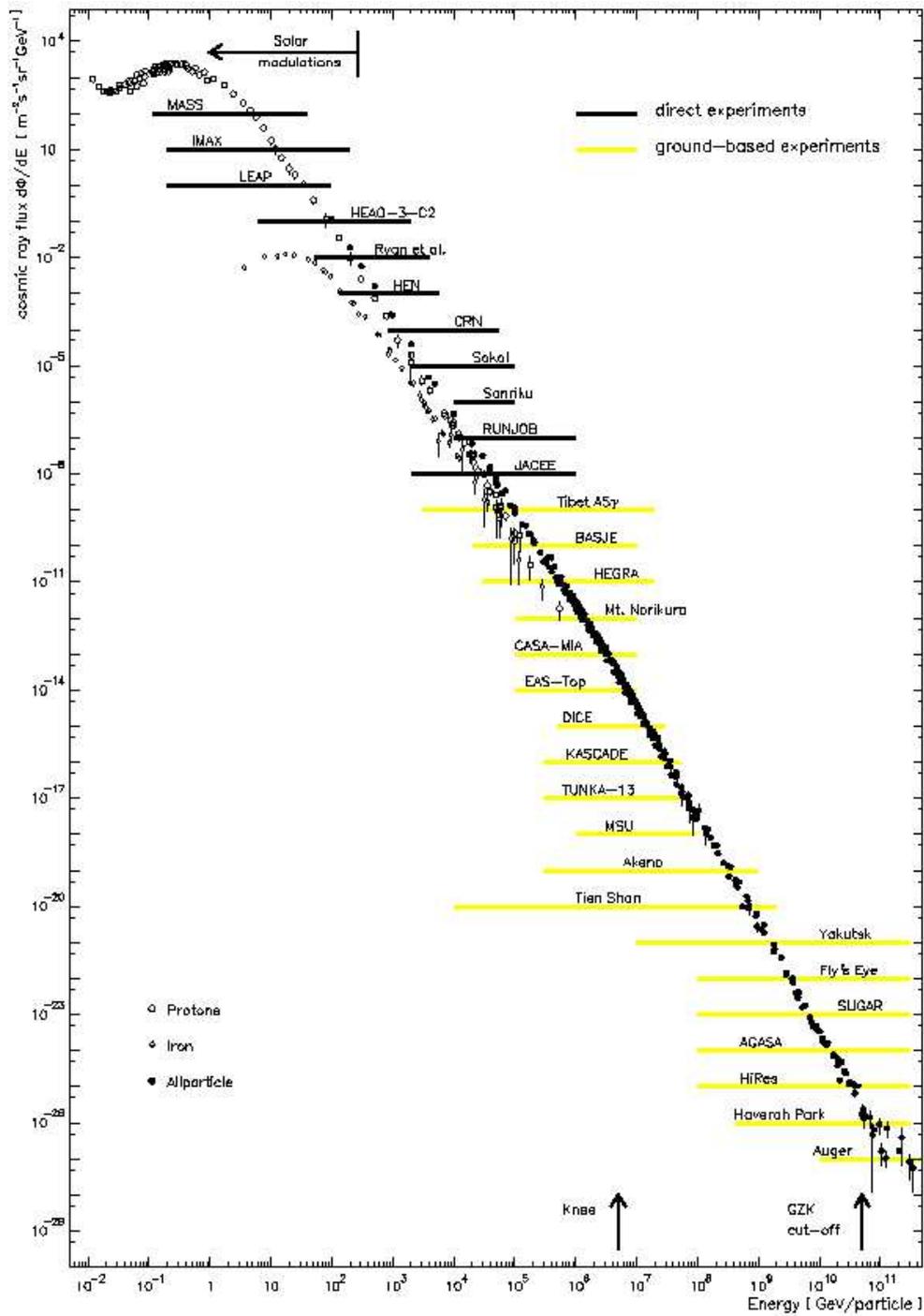


Abbildung 2.1: Das Spektrum der kosmischen Strahlung und der Messbereich verschiedener Experimente. Aus [14].

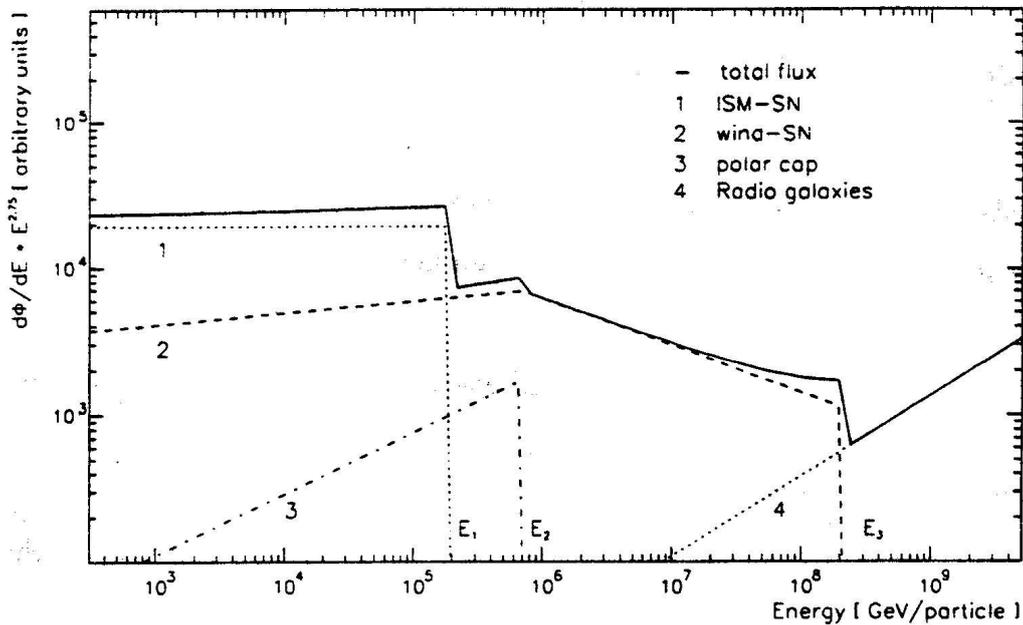


Abbildung 2.2: Die drei Hauptbeiträge zum Spektrum der kosmischen Strahlung [17].

Im Allgemeinen lässt sich das Spektrum der kosmischen Strahlung durch Beschleunigung von Teilchen in Schockwellen und durch den Durchgang dieser Teilchen durch das interstellare Medium interpretieren. Biermann und Stanev haben in [16] quantitativ den Ursprung der kosmischen Strahlung beschrieben. Eine wichtige Eigenschaft ihrer Theorie ist, eine quantitative Aussage über den Ursprung, die Energien, die Form des Spektrums und die chemische Zusammensetzung der Teilchen zu machen. In diesem Vorschlag geht die kosmische Strahlung von drei verschiedenen Quellen aus: Supernovae Explosionen in das interstellare Medium (mit ISM-SN abgekürzt), Supernovae Explosionen in den eigenen stellaren Wind (wind-SN) und extragalaktische Quellen oberhalb von  $10^{18}$  eV. Ihre Beiträge sind in Abbildung 2.2 dargestellt.

Allerdings lässt sich die Richtung der kosmischen Strahlung nicht rekonstruieren, da diese durch galaktische Magnetfelder auf ihrem Weg von der Quelle zur Erde abgelenkt werden. Daher möchte man Neutrinos benutzen, die einen Teil der ungeladenen kosmischen Strahlung darstellen.

Neutrinos sind schwer nachzuweisende Teilchen, sie zeichnen sich durch eine extrem geringe Wechselwirkung mit Materie aus: ihre mittlere freie Weglänge im All liegt bei etwa 1000 Lichtjahren. Dies macht es nötig, große Detektorvolumen zu benutzen und diese von allen anderen Einflüssen abzuschirmen. Da-

für hat man den Vorteil, dass Neutrinos als elektrisch neutrale Teilchen keiner elektromagnetischen Wechselwirkung unterliegen und auf ihrem Weg von der Quelle zur Erde nicht durch die interstellaren Magnetfelder abgelenkt werden. Auch entkommen sie aufgrund ihrer Eigenschaft als schwach wechselwirkende Teilchen ihren Quellen nicht nur an der Oberfläche, sondern sie werden nach ihrer Produktion im Inneren von Sternen oder in sehr dichter Materie quasi nicht mehr abgelenkt. Daher sind sie als Informationsträger sehr gut geeignet.

Die seltenen Wechselwirkungen von Neutrinos mit Materie macht es naturgemäß auch schwierig, sie in Detektoren nachzuweisen.

Obwohl der Fluss an Neutrinos sehr groß ist – es durchqueren etwa  $10^9$  Neutrinos pro Tag den AMANDA-Detektor – bedingt die sehr geringe Wechselwirkungswahrscheinlichkeit, dass sich nur circa zwanzig von ihnen im Eis in ein Myon umwandeln. Von diesen zwanzig neutrino-induzierten Myonen kann durch Akzeptanzeffekte und Filteralgorithmen bei der Ereignisrekonstruktion nur ein einziges – wie nachfolgend beschrieben – sicher nachgewiesen werden.

## 2.2 Das Messprinzip

Eine der Möglichkeiten ist der Nachweis des Leptons, welches bei einem Stoß eines Neutrinos  $\nu_l$  mit einem Atomkern  $N$  entsteht:

$$\nu_l + N \rightarrow l + X \quad (2.2)$$

Dabei wird typischerweise die Hälfte der Energie des Neutrinos auf das Lepton übertragen, der Rest teilt sich auf hadronische Nebenprodukte  $X$  auf.

Die Leptonen senden, wenn sie schneller als die Lichtgeschwindigkeit im durchquerten Medium sind, wie jedes geladene Teilchen Cherenkov-Licht aus.

$$v = \beta c > \frac{c}{n} \quad (2.3)$$

mit  $v$ : Teilchengeschwindigkeit,

$\beta$ : Teilchengeschwindigkeit in Einheiten der Lichtgeschwindigkeit,

$c$ : Lichtgeschwindigkeit,

$n$ : Brechungsindex des Mediums.

Für den Winkel  $\theta$ , unter dem das Cherenkov-Licht emittiert wird, gilt (siehe dazu auch Abbildung 2.3):

$$\theta = \arccos\left(\frac{1}{\beta n}\right). \quad (2.4)$$

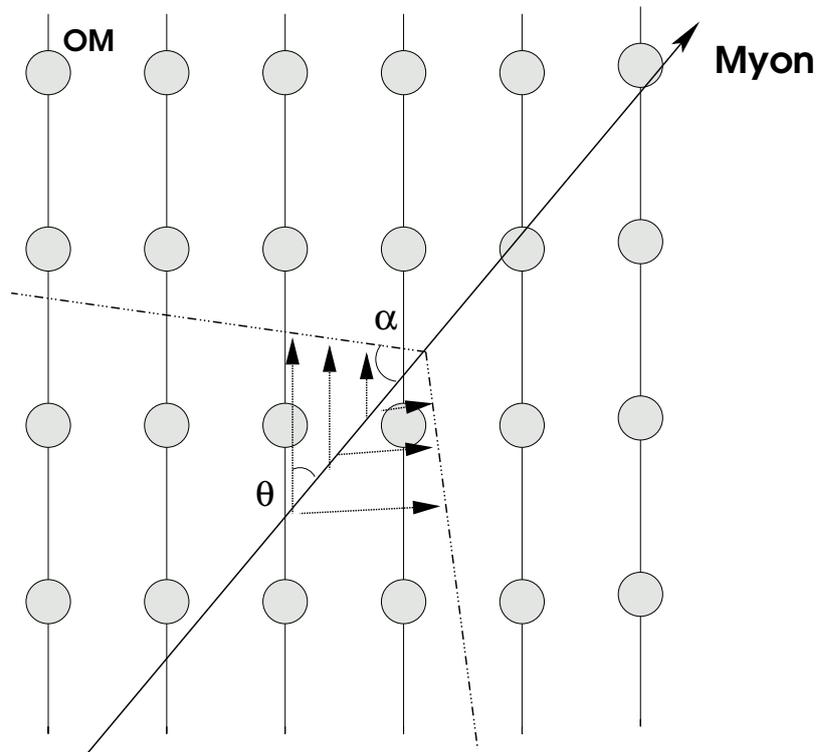


Abbildung 2.3: Schematische Darstellung der Messung des Cherenkov-Lichts. Die Kegelfront bewegt sich durch die Anordnung von optischen Modulen (OM) des AMANDA-Detektors.

Am Standort des AMANDA-Detektors hat das Eis einen Brechungsindex von ungefähr  $n \approx 1,32$ . Für Myonen ergibt sich damit eine Mindestenergie von 160 MeV, die nötig ist, um Cherenkov-Licht zu erzeugen. Ab etwa 1 GeV ist  $\beta \approx 1$ , so dass sich ein Grenzwinkel von  $\theta \approx 41^\circ$  ergibt.

Handelt es sich bei dem Lepton um ein hochenergetisches Myon, so hat dieses im Eis eine sehr große Reichweite von einigen Kilometern. Man nutzt nun aus, dass das Myon im Wesentlichen die gleiche Richtung wie das Neutrino aufweist. Die mittlere Winkelabweichung zwischen Myon und Neutrino beträgt ungefähr  $1,8^\circ$  bei 1 TeV und nimmt zu höheren Energien weiter ab.

## 2.3 Der Detektor

Der AMANDA-Detektor befindet sich am Südpol und hat die Aufgabe, hochenergetische Neutrinos – bei einer Energie von ungefähr 1 TeV – von astrophysikalischen Punktquellen zu messen.

Bei AMANDA misst man das Cherenkov-Licht, welches beim Durchgang hochenergetischer Teilchen durch das Eis entsteht, mit Hilfe von im antarktischen Eis eingelassenen lichtempfindlichen Detektoren, sogenannten „Photomultipliern“.

Diese lichtempfindlichen Detektoren werden zusammen mit der zur Auslese nötigen Elektronik an sogenannten „Strings“, also in Strängen, ins Eis gebracht. Eine Einheit aus Detektor (dabei handelt es sich um 8 Zoll Photomultiplier (PMT) der Firma Hamamatsu) und Elektronik, eingebettet in einer lichtdurchlässigen und druckresistenten Kugel, bezeichnet man als optisches Modul. Abbildung 2.4 zeigt schematisch den Aufbau eines solchen optischen Moduls.

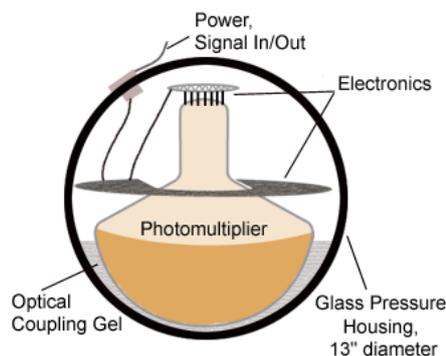


Abbildung 2.4: Schematische Darstellung eines optischen Moduls des AMANDA-Detektors. Aus [18].

Der AMANDA-Detektor besteht aus mehreren Strings, die seit 1993 im Eis eingelassen wurden. Die einzelnen Ausbaustufen sind die Folgenden:

- Die erste nennenswerte Ausbaustufe fand im antarktischen Sommer 1993/1994 statt. Dabei wurden vier Strings in einer Tiefe von 800-1000 m im Eis eingelassen. Diese Ausbaustufe wird AMANDA-A genannt.

Später hat man erkannt, dass in dieser Tiefe das Eis noch durch Lufteinschlüsse verunreinigt ist. An diesen Lufteinschlüssen streut das Cherenkov-Licht sehr stark, so dass eine Rekonstruktion des Teilchendurchgangs schwie-

rig ist. Daher wurde entschieden, die nächsten Ausbaustufen in einer Tiefe von 1500-2000 m einzulassen.

- Die nächste Ausbaustufe ist der AMANDA-B Detektor, dessen vier innere Strings im antarktischen Sommer 1995/1996 aufgebaut wurden. In Anlehnung an diese Anzahl nennt man diesen Teil auch AMANDA-B<sub>4</sub>.

Die optischen Module sind mit Koaxialkabeln verbunden, um ein Übersprechen der elektrischen Signale zu verhindern. Allerdings ist Koaxialkabel dispersiv, weshalb man kurze Impulse nicht mehr gut auflösen kann. Auch sind die verwendeten Kabel recht dick, was die Anzahl an Kabel in einem String limitiert. Diesen Problemen wurde in der nächsten Stufe Rechnung getragen.

- 1996/1997 wurden sechs weitere Strings aufgebaut und um den AMANDA-B<sub>4</sub> Detektor ins Eis eingelassen. Dabei wurden verdrehte Leitungen – sogenanntes „Twisted Pair“-Kabel – verwendet, welche dünner sind. Zusammen mit den vier Strings der AMANDA-B<sub>4</sub> Ausbaustufe bezeichnet man diese sechs Strings als AMANDA-B<sub>10</sub>.
- Von 1997 bis 1999 wurden dann noch die Strings 11-19 aufgebaut. Zusammen mit den zehn älteren bilden sie die sogenannte AMANDA-II Detektorgeometrie.

Die Abbildung 2.5 zeigt schematisch den kompletten Detektor.

## 2.4 Die AMANDA-Simulationssoftware

Die AMANDA-Simulationssoftware ist in mehrere Programmpakete aufgeteilt. Diese Programmpakete haben unterschiedliche physikalische Problemstellungen, die die einzelnen Prozesse von der Simulation eines Ereignisses in der Atmosphäre über die Lichtausbreitung im Eis, die Simulation der Detektorantwort und schließlich der Rekonstruktion berücksichtigen. Es ist offensichtlich, dass dies eine nicht-triviale Ablaufsteuerung beinhaltet.

Einen Überblick über alle Programmpakete der AMANDA-Kollaboration gibt [19]. Die folgende Übersicht stellt die wichtigsten Programmpakete vor, soweit sie für die im folgenden Kapitel 5 vorgestellte Grid-Anwendung von Belang ist:

- **CORSIKA**  
CORSIKA<sup>1)</sup> simuliert Luftschauer, die von hochenergetischen Teilchen aus

---

<sup>1)</sup>CORSIKA = COsmic Ray Simulations for KASCADE

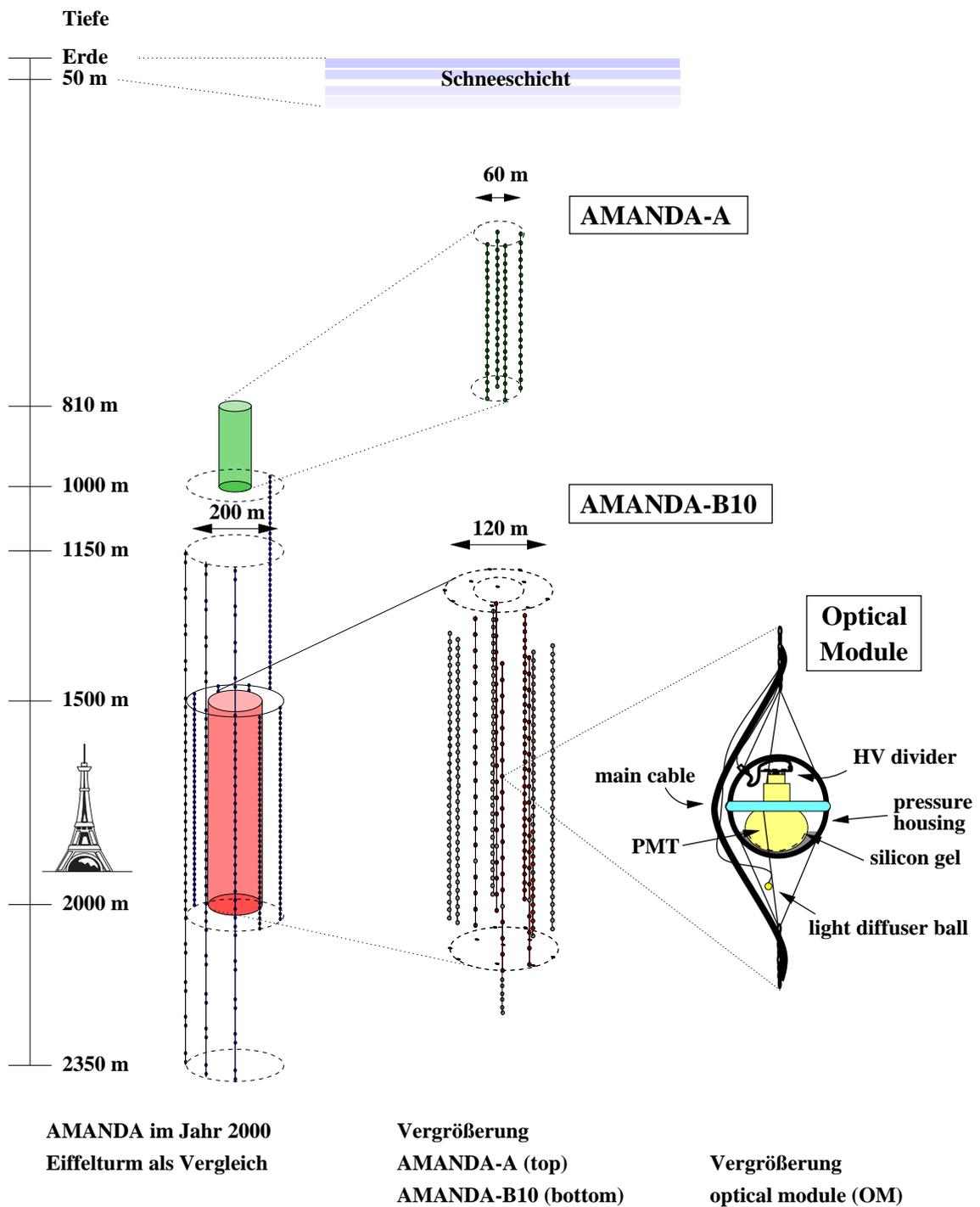


Abbildung 2.5: Schematische Darstellung des AMANDA-Detektors.

der kosmischen Strahlung ausgelöst werden. Das Programm wurde im Rahmen der KASCADE-Kollaboration<sup>2)</sup> [20] entwickelt, einem Array von Szintillationszählern am Forschungszentrum Karlsruhe, welches die Teilchen (Elektronen, Photonen und Myonen) aus Luftschauern nachweist.

CORSIKA wird mittlerweile von vielen Kollaborationen als Grundlage ihrer Simulationen benutzt. AMANDA benutzt eine leicht modifizierte Version des Programms, unter anderen werden Datenströme in das von AMANDA benutzte Datenformat „F2000“ umgewandelt. Eine Beschreibung des Formates findet sich in [21].

Das Programm CORSIKA ist in der Programmiersprache FORTRAN geschrieben.

- **MMC**

MMC<sup>3)</sup> simuliert den Durchgang von Myonen durch Materie. Dabei wird der Energieverlust des Myons durch Ionisation, Bremsstrahlung, Photon-Nukleon-Wechselwirkungen und Paarproduktionen errechnet. Es wurde von Dmitry Chirkin während eines Aufenthaltes an der Bergischen Universität Wuppertal entwickelt und sollte unter anderem auch die Tauglichkeit der JAVA-Programmiersprache im Rahmen von wissenschaftlichen Anwendungen demonstrieren.

- **AMASIM**

AMASIM<sup>4)</sup> simuliert die Ausbreitung von Photonen, die durch die Cherenkov-Strahlung entstehen, durch das Eis. Dabei sind insbesondere Streuprozesse zu beachten. Da diese Simulation sehr rechenzeitaufwändig ist, werden diese Berechnungen für ein Photon mit seinen spezifischen Parametern einmal durchgeführt und das Ergebnis in Tabellen abgespeichert. Das verringert den Rechenzeitaufwand, hat aber zur Folge, dass der Bedarf an verfügbarem Hauptspeicher deutlich steigt, was in einer verteilten Rechenumgebung problematisch sein kann.

AMASIM wurde in der Programmiersprache C++ geschrieben.

Der Rechenzeitbedarf für die Simulation von atmosphärischen Myonen und die daraus entstehenden Dateigrößen sind in Tabelle 2.1 wiedergegeben. Dort angegebene, aber nicht in der vorherigen Übersicht aufgeführte Simulationsprogramme sind in [19] beschrieben. Die Simulation für etwa eine Minute Datennahme des AMANDA-Detektors erfordert 1,6 Stunden Rechenzeit.

---

<sup>2)</sup>KASCADE = KARlsruhe Shower Core and Array Detector. Mittlerweile hat sich die Kollaboration in „KASCADE Grande“ umbenannt.

<sup>3)</sup>MMC = Muon Monte Carlo

<sup>4)</sup>AMASIM = Simulation tool for AMANDA

Simulationprogramm	CPU (sec)	Dateigröße (MB)
CORSIKA	2.100	3
Corama	80	3
MMC	450	45
AMASIM	3.000	6
Minbias	800	8
RecoL1	150	0.5
RecoL2	17	0.1
<b>Summe</b>	<b>ungefähr 5.800</b>	<b>66</b>

Tabelle 2.1: Übersicht über den Rechenzeitbedarf (normiert auf eine Pentium-III CPU mit 1 GHz Taktfrequenz) und die Dateigrößen der AMANDA-Simulationsprogramme für  $10^5$  Primärteilchen. Dies entspricht einer Datennahmezeit des Detektors von 53,94 Sekunden. Nach [22].

## Kapitel 3

# Das Tevatron und das DØ-Experiment

Das Tevatron am „Fermi National Accelerator Laboratory“ – auch kurz Fermilab genannt – ist der momentan weltweit leistungsfähigste Teilchenbeschleuniger. Dort werden Protonen und Antiprotonen bei einer Schwerpunktsenergie von 1,96 TeV zur Kollision gebracht. Das Fermilab befindet sich ungefähr 70 km entfernt von Chicago im amerikanischen Bundesstaat Illinois. Eine Luftaufnahme ist in Abbildung 3.1 zu sehen. Darauf eingezeichnet ist die ungefähre Lage der Stadt Chicago sowie die zwei Beschleunigerringe des Tevatrons und auch die Lage der Experimente DØ und CDF. Der obere Ring ist der Hauptbeschleunigerring mit den Kollisionspunkten am Ort der Experimente, während der untere Ring („Main Injector“) zur Vorbeschleunigung und Injektion der Teilchen in das Tevatron dient.

Man unterscheidet zwei Datennahmeperioden: während des sogenannten „Run I“, der von 1992 bis 1996 lief, haben die beiden Experimente DØ und CDF ungefähr  $125 \text{ pb}^{-1}$  von  $\bar{p}p$  Daten bei einer Schwerpunktsenergie von 1,8 TeV genommen. Die neuere Datennahmeperiode („Run II“) startete im März 2001 und hat zum Ziel, bis zum Jahr 2009 zwischen  $4 \text{ fb}^{-1}$  und  $9 \text{ fb}^{-1}$  an Daten zu liefern. Dazu wurden auch die beiden Experimente verbessert.

### 3.1 Der Tevatron-Beschleuniger

Der Tevatron-Beschleuniger ist der letzte in einer Kette von Beschleunigern, die hochenergetische Proton- und Antiproton-Strahlen erzeugen. Abbildung 3.2 zeigt schematisch den Aufbau der Beschleunigerkette.

Das Tevatron hat einen Umfang von 6,4 km. Der Beschleunigungsprozess beginnt mit Wasserstoff-Atomen, die zu  $\text{H}^+$  ionisiert werden. Diese werden im sogenannten „Cockcroft-Walton“-Beschleuniger auf eine Energie von 750 keV be-



Abbildung 3.1: Luftbildaufnahme des Tevatron. Aus [23].

schleunigt. Die nächste Stufe ist dann ein Linearbeschleuniger (in Abbildung 3.2 mit „LINAC“ bezeichnet), der die  $H^-$ -Ionen auf 400 MeV beschleunigt. Anschließend werden die zwei Elektronen des  $H^-$ -Ions entfernt, so dass Protonen entstehen, die dann in einem „Booster“-Synchrotron eine Energie von 8 GeV erhalten und in den Vorbeschleunigerring („Main Injector“) eingespeist werden. Dieser „Main Injector“ hat einen Umfang von 3,2 km.

Zur Erzeugung von Antiprotonen wird ein Teil der Protonen auf 120 GeV beschleunigt und auf eine Nickel-Scheibe in der „Target Hall“ gelenkt. Die hierbei erzeugten Antiprotonen werden dann in den „Main Injector“ eingespeist, der diese zusammen mit den Protonen auf 150 GeV beschleunigt, bevor beide Teilchenarten in das „Tevatron“ eingebracht werden. Dort werden sie dann schließlich auf jeweils 0,98 TeV beschleunigt. Dazu wird ein Magnetfeld von 4,2 T verwendet, welches von supraleitenden Spulen erzeugt wird.

Die Protonen und Antiprotonen werden in Paketen beschleunigt, die man „bun-

## Fermilab's ACCELERATOR CHAIN

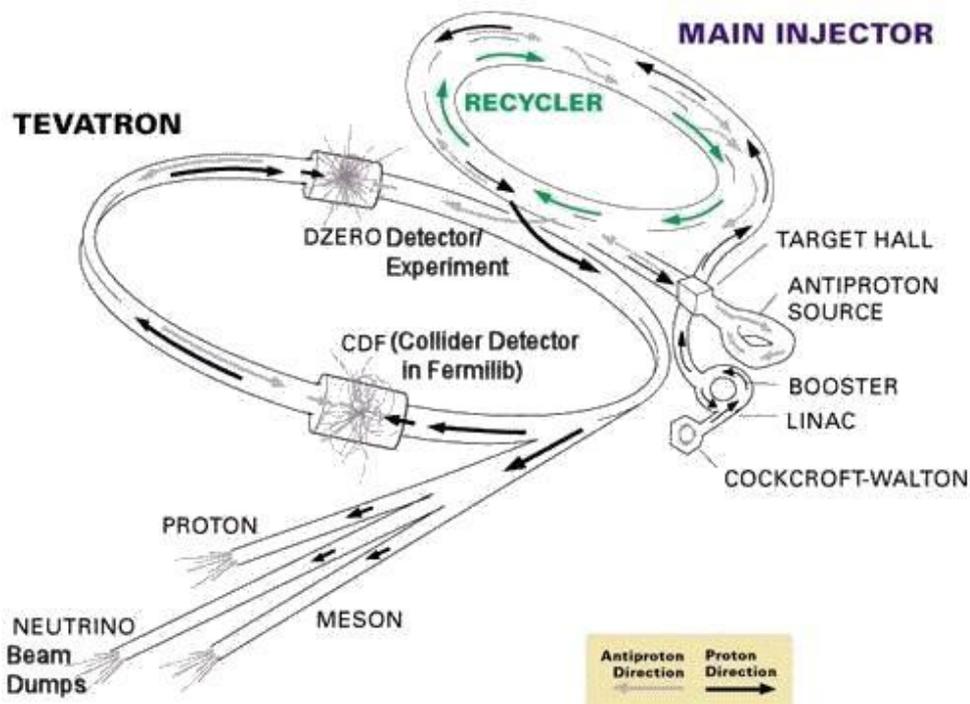


Abbildung 3.2: Schematische Darstellung der Beschleunigerstufen.  
Aus [24].

ches“ nennt. Im „Run II“ sind jeweils 36 Pakete von Protonen und 36 Pakete von Antiprotonen in einem zeitlichen Abstand zwischen zwei Paketen von 396 ns im Ring.

Folgende Tabelle 3.1 fasst die wichtigsten Parameter des Tevatrons zusammen.

### 3.2 Der DØ-Detektor

Der DØ-Detektor befindet sich am Kollisionspunkt  $\mathcal{D}_0$  des Tevatrons, woher er seinen Namen hat. Er ist ein typischer Mehrzweck-Detektor für einen symmetrischen Beschleuniger. Er wurde darauf ausgelegt Elektronen, Myonen und Jets möglichst genau zu messen und indirekt fehlende transversale Energie zu bestimmen. Beim „Run II“-Detektor handelt es sich um eine Weiterentwicklung

Parameter		Run Ib	Run II
Umfang	[km]	6,28	6,28
Protonen pro Paket		$2,3 \cdot 10^{11}$	$2,7 \cdot 10^{11}$
Antiprotonen pro Paket		$5,5 \cdot 10^{10}$	$7,5 \cdot 10^{10}$
Anzahl der Pakete		6	36
zeitlicher Abstand der Pakete	[ns]	3500	396
Luminosität	$[\text{cm}^{-2}\text{s}^{-1}]$	$1,6 \cdot 10^{31}$	$2,1 \cdot 10^{32}$
Schwerpunktsenergie	[TeV]	1,8	1,96

Tabelle 3.1: Eigenschaften des Tevatron-Beschleunigers. Zum Vergleich sind die Werte für den „Run Ib“ (1992-1996) mit angegeben.

des „Run I“-Detektors, dessen Stärke bei einer guten Elektron-Identifizierung lag. Abbildung 3.3 zeigt einen Querschnitt durch den DØ-Detektor.

Man erkennt vom Wechselwirkungspunkt ausgehend das Spurfindungssystem (als „Tracking System“ bezeichnet). Dieses dient dazu, dreidimensionale Flugbahnen von geladenen Teilchen zu vermessen. Es ist unterteilt in einen inneren hochauflösenden Silizium-Vertexdetektor und einem äußeren Spurfindungssystem. Um eine Impuls- und Ladungsmessung vornehmen zu können, befindet sich das gesamte Spurfindungssystem in einem solenoidalen Magnetfeld von 2 T.

Vom Wechselwirkungspunkt nach außen gehend schließt sich das Kalorimeter-system an. Es misst die Energien von elektromagnetischen und hadronischen Teilchen. Es besteht hauptsächlich aus drei Teilen: einem sogenannten „Preshower Detector“, der zur genauen Vermessung der Teilchenspuren nach dem Durchgang durch den Magneten und der dabei auftretenden Mehrfachstreuung dient, sowie dem elektromagnetischen und dem hadronischen Kalorimeter. Nach außen schließt sich das Myonsystem an.

Die einzelnen Komponenten werden nun im Folgenden genauer vorgestellt.

### 3.2.1 Das DØ-Koordinatensystem

In DØ wird ein rechtshändiges Koordinatensystem verwendet. Die z-Achse zeigt dabei in Richtung des Protonenstrahls, die x-Achse zeigt von der Mitte des Be-

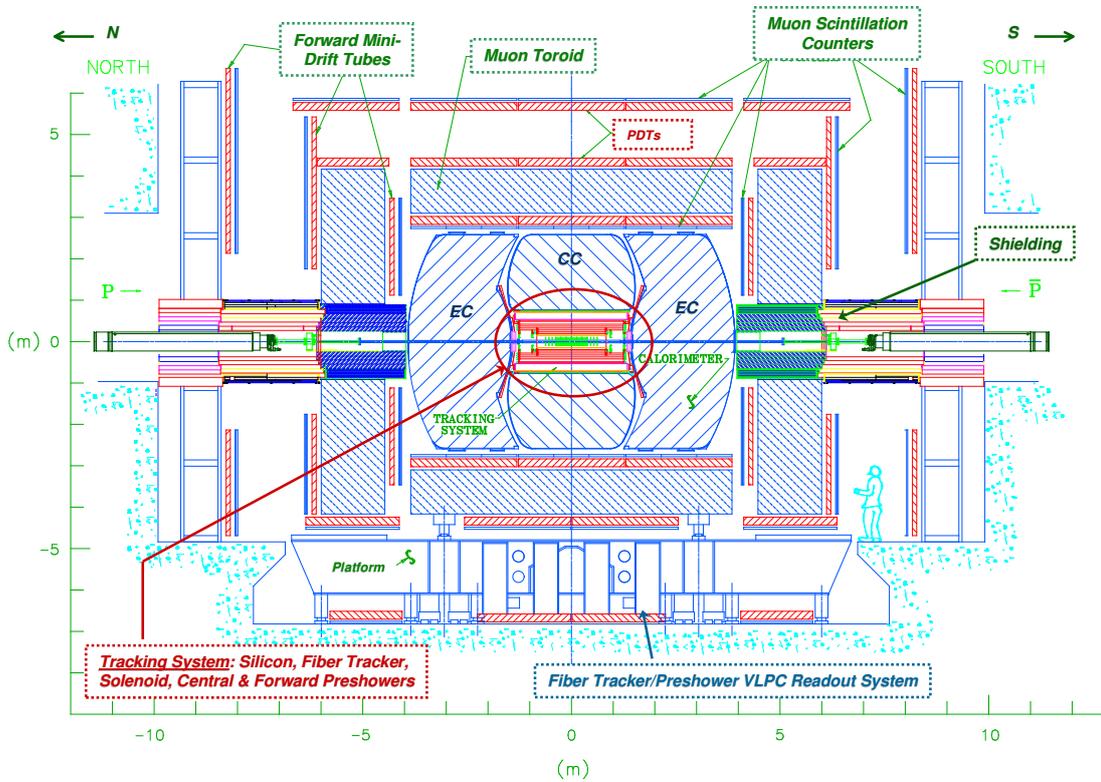


Abbildung 3.3: Querschnitt des DØ-Detektors. Aus [23].

schleunigerrings radial nach außen und die  $y$ -Achse aufwärts.

Statt kartesischen Koordinaten  $(x, y, z)$  werden oft Polarkoordinaten  $(r, \theta, \phi)$  verwendet, die wie folgt definiert werden:

$$r = \sqrt{x^2 + y^2} \quad (3.1)$$

$$\phi = \tan \frac{y}{x} \quad (3.2)$$

Zusammen mit dem Polarwinkel  $\theta$ , der bezüglich der positiven  $z$ -Achse definiert wird, führt man wie bei allen Hadron-Beschleunigern die Pseudorapidität  $\eta$  ein, die wie folgt definiert ist:

$$\eta = -\ln \tan \frac{\theta}{2}. \quad (3.3)$$

Diese Größe hat den Vorteil, dass die Multiplizität von hochenergetischen Teilchen in der QCD-Produktion ( $dN/d\eta$ ) nahezu konstant ist.

### 3.2.2 Der innere Detektor

Abbildung 3.4 zeigt einen Überblick über die Komponenten des inneren Detektorsystems, auf die im Folgenden näher eingegangen wird:

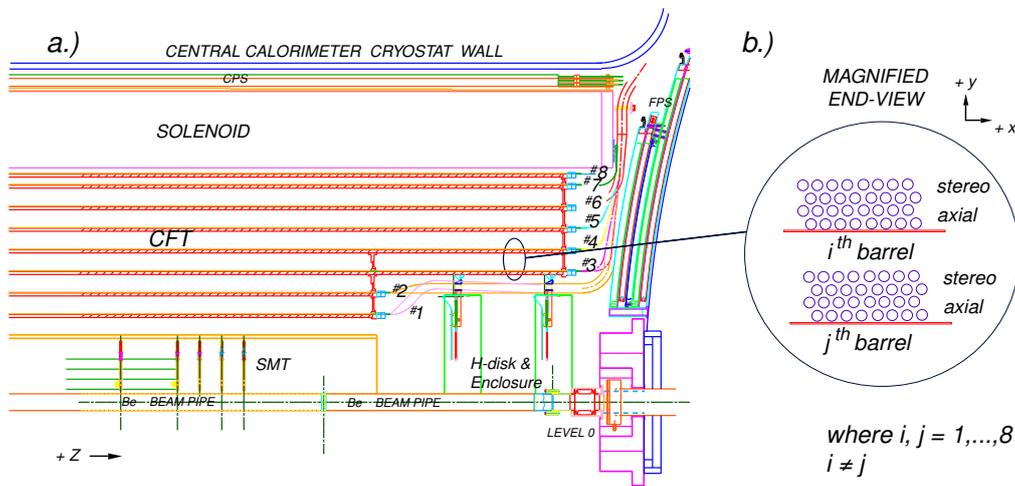


Abbildung 3.4: Innere Komponenten des DØ-Detektors. Aus [25].

### 3.2.3 Das Luminositätssystem

Dieses System dient zur Messung der momentanen Luminosität, die vom Beschleuniger an das Experiment geliefert wird. Dazu misst man die Rate der inklusiven  $p\bar{p}$ -Streuung, die sich allerdings durch einen geringen Transversalimpuls auszeichnet. Das heißt, dass man diesen Detektor möglichst nahe am Strahlrohr platzieren muss.

Bei DØ besteht dieses System aus zwei Paaren von Plastik-Szintillatoren, die in der Abbildung 3.4 als „LEVEL 0“ bezeichnet sind. Sie sind symmetrisch um  $\phi$  um das Strahlrohr angebracht und decken einen Pseudorapiditätsbereich von  $2,7 < |\eta| < 4,4$  ab. Ihre Zeitauflösung ist kleiner als 0,2 ns.

### 3.2.4 Der Silizium- und Faser-Detektor

Der Siliziumvertex-Detektor [26] („Silicon Microstrip Tracker“, abgekürzt SMT) ist der innerste Spurdetektor von DØ und in Abbildung 3.5 schematisch dargestellt. Man erkennt die Trennung in konzentrisch um die Strahlachse aufgebaute Zylinder (auch als „Barrel“ bezeichnet) und weiter vom Wechselwirkungspunkt entfernt liegende Scheiben, die als „Disks“ bezeichnet werden. Es handelt sich dabei um einen Silizium-Streifendetektor, der teilweise mit zweiseitigen Detektoren ausgestattet ist, um zweidimensionale Ortsinformationen zu erhalten. Dabei kommen je nach Lage verschiedene Stereowinkel zum Einsatz.

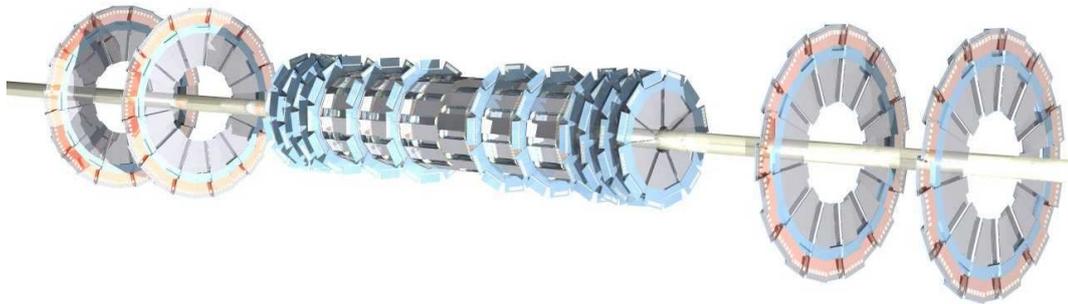


Abbildung 3.5: Schematischer Aufbau des Silizium-Detektors von DØ.  
Aus [23].

Die zylindrischen Detektoren bestehen aus vier Lagen und sind in sechs 12 cm lange Segmente unterteilt. Dazwischen sowie jeweils am Ende befinden sich sechs Scheiben mit einem Innen- und Außenradius von  $r_i=2,6$  cm bzw.  $r_a=14,5$  cm. Die Scheiben mit dieser Geometrie werden „F-Disk“ genannt. Daneben sind bei  $z = \pm 110$  cm und  $z = \pm 120$  cm noch vier größere Scheiben mit  $r_i=9,4$  cm bzw.  $r_a=24,0$  cm montiert, die „H-Disks“ genannt werden.

Umgeben ist der SMT von einem Faserdetektor [27] („Central Fiber Tracker“, abgekürzt CFT). Er besteht aus acht konzentrischen Zylindern mit Radien von  $19,5$  cm  $< r < 51,5$  cm und einer Länge von 1,7 m (bei den innersten zwei Lagen) bzw. 2,5 m, wie es auch in Abbildung 3.4 deutlich wird. Er trägt wesentlich zur Messung des Transversalimpulses  $p_T$  der durch ihn fliegenden Teilchen bei. Außerdem trägt er bis  $|\eta| < 1,6$  zum Level 1 Trigger bei.

Insgesamt besteht der CFT aus 76.800 szintillierenden Fasern mit einem Durchmesser von jeweils  $835$   $\mu\text{m}$ . Sie werden einseitig durch sogenannte „Visible Light Photon Counters“ (VLPC) ausgelesen.

Zusammengenommen messen beide Subdetektoren den Transversalimpuls für ein geladenes Teilchen mit einer Auflösung von  $\delta p_T/\text{GeV} = 0,002 \cdot (p_T/\text{GeV})^2$ .

### 3.2.5 Das Kalorimeter

Das DØ-Kalorimeter ist bei der Modifikation für den „Run II“ nicht verändert worden. Es benutzt flüssiges Argon als aktives Medium und angereichertes Uran als Absorbermaterial. Es ist selbstkompensierend und überdeckt den Bereich bis ungefähr  $|\eta| < 4$ . Es ist unterteilt in einen zentralen Bereich (CC) und zwei Endkappen-Bereiche (EC), wobei der zentrale Bereich den Pseudorapiditätsbereich bis ungefähr  $|\eta| < 1,2$  abdeckt. Jeder Bereich ist wiederum in drei Regionen unterteilt: in eine elektromagnetische (EM), die dem Wechselwirkungspunkt am nächsten ist, gefolgt von einer fein segmentierten hadronischen (FH) und einer gröber segmentierten hadronischen Region (CH). Abbildung 3.6 zeigt diese Unterteilung.

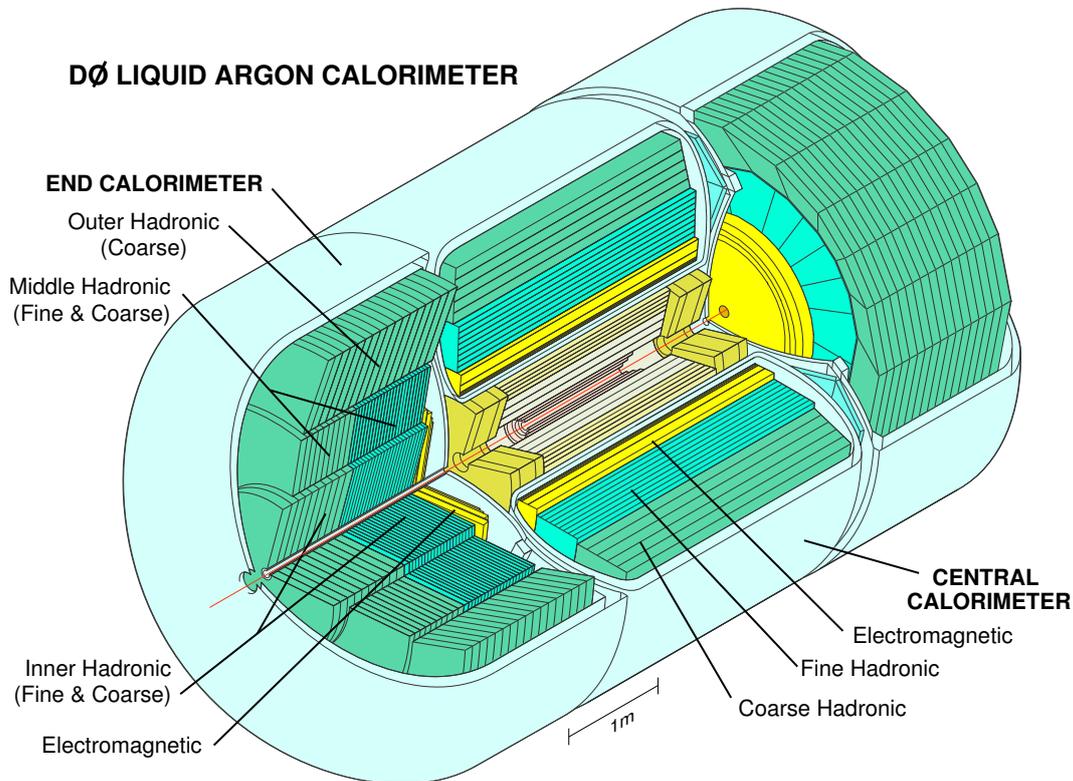


Abbildung 3.6: Überblick über das DØ-Kalorimeter. Aus [25].

Die Granularität beträgt fast im gesamten Kalorimeter  $\delta\eta \times \delta\phi = 0,1 \times 0,1$ , lediglich eine Lage des elektromagnetischen Kalorimeters ist feiner unterteilt ( $\delta\eta \times \delta\phi = 0,05 \times 0,05$ ), um das Maximum des elektromagnetischen Schauers genauer ausmessen zu können. Tabelle 3.2 gibt die Parameter des Kalorimeters wieder.

Lage	d [cm] ( $\eta = 0$ )	d [ $X_0$ ] ( $\eta = 0$ )	d [ $\lambda_a$ ] ( $\eta = 0$ )	$\delta\eta \times \delta\phi$
EM 1		2,0		$0,1 \times 0,1$
EM 1		2,0		$0,1 \times 0,1$
EM 1		6,8		$0,05 \times 0,05$
EM 1		9,8		$0,1 \times 0,1$
<b>EM</b>	$\approx 25$	20,6	0,76	
FH 1			1,3	$0,1 \times 0,1$
FH 1			1,0	$0,1 \times 0,1$
FH 1			0,9	$0,1 \times 0,1$
<b>FH</b>	$\approx 75$		3,2	
<b>CH</b>	$\approx 60$		3,2	$0,1 \times 0,1$

Tabelle 3.2: Überblick über die Parameter der Kalorimeters von DØ.

Zwischen Zentral- und Endkappenbereich des Kalorimeters befindet sich ein Bereich mit uninstrumentiertem Material. Um den Energieverlust in diesem Bereich zumindest teilweise zu kompensieren, existiert der sogenannte „InterCryostat Detector“ (ICD). Abbildung 3.7 zeigt ein Querschnitt des Kalorimetersystems mit der Lage des ICD. Es besitzt ebenfalls eine Granularität von  $\delta\eta \times \delta\phi = 0,1 \times 0,1$  passend zu den umliegenden Kalorimeterzellen.

Die Energieauflösung des Kalorimeters für den „Run I“<sup>1)</sup> wurde mit Hilfe von Pion- und Elektron-Testbeam-Messungen bestimmt [25]. Die relative Auflösung als Funktion der Energie kann wie folgt parametrisiert werden:

$$\frac{\sigma_E}{E} = \sqrt{C^2 + \frac{S^2}{E} + \frac{N^2}{E^2}} \quad (3.4)$$

Die Konstanten C, S und N wurden bestimmt zu:

Teilchenart	C	S	N
e	0,003	$0,175 \cdot \sqrt{\text{GeV}}$	0,14 GeV
$\pi$	0,032	$0,41 \cdot \sqrt{\text{GeV}}$	1,28 GeV

Tabelle 3.3: Testbeam-Ergebnisse zur Parametrisierung der Kalorimeterauflösung.

<sup>1)</sup>Neuere Messungen liegen nicht vor. Da das Kalorimeter zum „Run II“ aber nicht verändert wurde, werden die vorhandenen Ergebnisse hier angegeben.

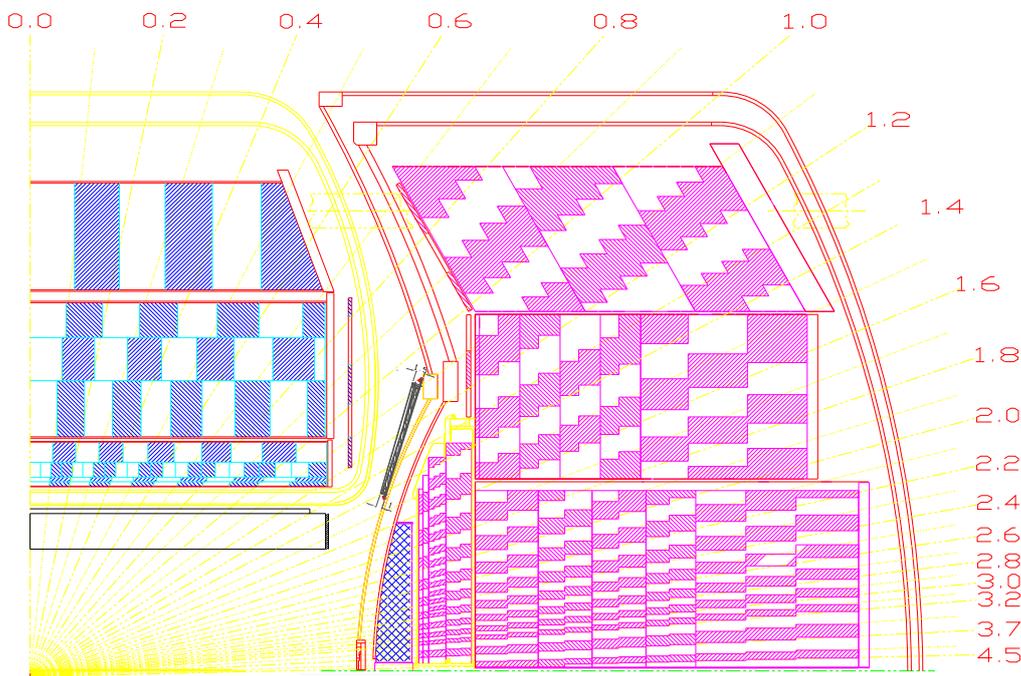


Abbildung 3.7: Seitenansicht des DØ-Kalorimeters. Aus [28].

### 3.2.6 Das Myonsystem

Das Myonsystem ist das vom Wechselwirkungspunkt entfernteste Teilsystem des DØ-Detektors. Es besteht aus Driftkammern und deckt den Pseudorapiditätsbereich bis  $|\eta| < 2$  ab. Ein 1,8 T Magnetfeld wird von einem toroidialen Eisenmagneten erzeugt. Abbildung 3.8 zeigt den Aufbau des Myonsystems.

Im Zentralbereich bis  $|\eta| < 1$  bezeichnet man das System als „Wide Angle Muon Spectrometer“ (WAMUS). Hier werden Proportionaldriftkammern eingesetzt – in der Zeichnung als PDT bezeichnet. Diese sind mit Gas gefüllt, welches beim Durchgang von Teilchen ionisiert wird. Durch eine an Drähten angelegte Hochspannung wird die erzeugte Ladung gesammelt.

Im Vorwärtsbereich ( $1 < |\eta| < 2$ ) wird das System „Forward Angle Muon Spectrometer“ genannt und ist mit kleineren Proportionaldriftkammern ausgestattet, die daran anlehnend „Mini Drift Tubes“ (MDT) genannt werden.

Jeder Teilbereich ist in drei Lagen unterteilt, die mit den Buchstaben A, B und C bezeichnet werden. Dabei findet sich die „A-Lage“ jeweils zwischen Kalorimeter und dem Magneten, während die „B-“ und die „C-Lage“ außerhalb des Magneten angebracht sind.

Zusätzlich zu den Driftkammern werden im Zentralbereich Szintillationszähler

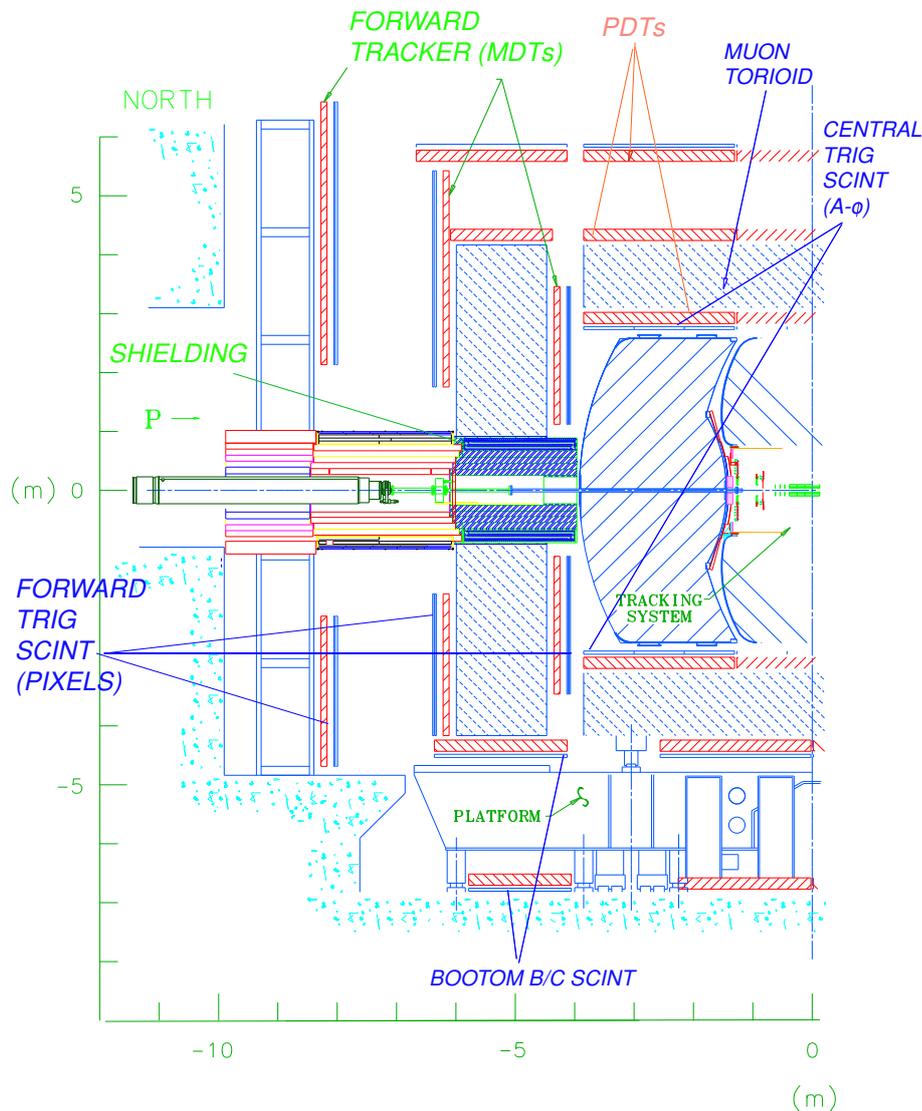


Abbildung 3.8: r-z Ansicht des DØ-Myonsystems. Aus [25].

innerhalb der „A-Region“ („A- $\phi$  Lage“) und außerhalb der „C-Region“ („cosmic cap“) angebracht. Diese dienen einer Messung von  $\phi$  beziehungsweise als Veto gegen kosmische Strahlung.

Im Vorwärtsbereich sind alle drei Myonkammern-Lagen mit einer Lage von szintillierenden Pixeln versehen, die eine Segmentierung von  $\delta\eta \times \delta\phi = 0,12 \times 0,08$  (äußere neun Reihen) beziehungsweise  $\delta\eta \times \delta\phi = 0,07 \times 0,08$  (für die inneren zwei Reihen) aufweisen.

Insgesamt muss ein Teilchen, um das äußere Myonsystem zu erreichen, 7-9

Wechselwirkungslängen im Kalorimeter und zusätzlich 1-2 Wechselwirkungslängen im Toroid passieren. Abbildung 3.9 zeigt die Lage der einzelnen Subdetektoren des Myonsystems.

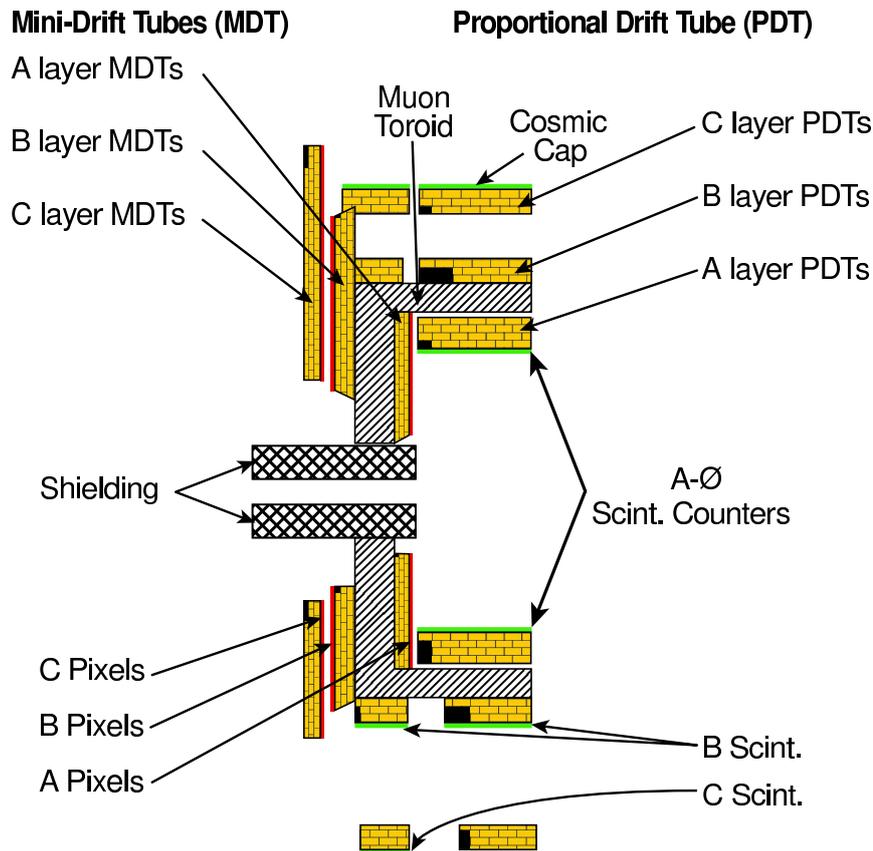


Abbildung 3.9: Seitenansicht des Myonsystems. Aus [29].

### 3.2.7 Die Simulation

Wie die vorherigen Abschnitte zeigen, ist der DØ-Detektor als Mehrzweck-Detektor komplex aufgebaut und verfügt über eine große Anzahl von Subdetektoren. Durch die Verwendung von Protonen und Antiprotonen im Tevatron ist die Multiplizität von Teilchenspuren im Detektor sehr hoch und es treten häufig Ereignisse mit vielen Jets (sogenannte „Multijet-Ereignisse“) auf.

Für die Simulation von Wechselwirkungen der Teilchen beim Durchgang durch Materie wird das Programm GEANT [30] eingesetzt. Dabei ist die Beschreibung der im Kalorimeter entstehenden Schauer mit ihren vielen Spuren besonders

aufwändig. Ein anderer zeitintensiver Prozess ist das Zusammenführen der rekonstruierten Spuren der einzelnen Subdetektoren [31].

Die Erzeugung von simulierten Ereignissen des DØ-Detektors ist daher insgesamt sehr rechenzeitintensiv. Eine Simulation von 250 Ereignissen benötigt ungefähr acht Stunden auf einem AMD Opteron Prozessor mit 1,8 GHz Taktfrequenz.

# Kapitel 4

## Einführung in das Grid-Computing

In diesem Kapitel sollen die Grundbegriffe des Grid-Computings definiert werden. Die für diese Arbeit wichtigsten Grid-Initiativen werden vorgestellt und Definitionen gegeben.

### 4.1 Definition

Eine heute oft zitierte und allgemein anerkannte Definition für Grid-Computing wird von Ian Foster gegeben als:

„Infrastructure that enables the integrated, collaborative use of high-end computers, networks, databases, and scientific instruments owned and managed by multiple organizations.“ [32]

Nach dieser Definition ist das Grid also ein Modell für verteiltes Rechnen, welches den Idealzustand einer Infrastruktur beschreibt, die jedem einzelnen Benutzer eine einheitliche Schnittstelle zu Ressourcen jeder Art ermöglicht.

Im engeren Sinn wird das Grid-Computing oft aber als eine Sammlung von Datenprotokollen bezeichnet, die über eine als „Middleware“ bezeichnete Software-Schicht einen einheitlichen Zugang zu Datenspeichern, Datenbanken und Rechenkapazitäten ermöglicht.

Das Grid Computing wird oft mit dem Elektrizitätsnetzwerk verglichen, welches die Industrie und Haushalte mit Strom versorgt. Denn auch hier ist der einzelne Verbraucher nicht daran interessiert, auf welchen Wege die elektrische Energie zu ihm gelangt. Vielmehr ist durch die Standardisierung der Schnittstellen ein einheitlicher Austausch zwischen Produzenten und Verbrauchern möglich.

Das real existierende Problem, welches den Grid-Ideen zu Grunde liegt, lässt sich dadurch charakterisieren, dass Personen in *instititusübergreifenden virtuellen Organisationen Ressourcen teilen und Probleme lösen* wollen. Dabei geht es nicht nur um das Mitbenutzen von Dateien oder Datensätzen, sondern um gemeinsamen und einheitlichen Zugriff auf alle Arten von Rechenressourcen, Software und Daten, die erforderlich sind, um die konkrete Problemstellung zu lösen. Dafür müssen Regeln aufgestellt werden, die die Verteilung dieser Ressourcen und den Zugriff darauf reglementieren. Eine Gruppe von Personen und/oder Institutionen, die solch ein gemeinsames Regelwerk anwenden, nennt man *virtuelle Organisation (VO)* [33]. Virtuelle Organisationen können sich sehr in ihren Eigenschaften unterscheiden, so kann die Größe, die Dauer, der Zweck, etc. sehr unterschiedlich sein.

Beispielhaft seien hier einige mögliche virtuelle Organisationen genannt:

- Mitglieder eines industriellen Konsortiums, die an einem Projekt arbeiten,
- Teilnehmer an einer Schulung,
- Mitglieder einer Kollaboration, die kurz- oder langfristig mit gemeinsamen Daten arbeiten.

Trotz aller Unterschiede erkennt man gemeinsame Belange und Anforderungen. Man benötigt extrem flexible Algorithmen, um das gemeinsame Nutzen der Ressourcen zu verwalten und Kontroll-Mechanismen, die eine genaue und feine Zuteilung von Rechten erlauben. Diese Mechanismen zur Rechteverteilung möchte man auf eine einfache Art verbreiten können, ohne dabei die Kontrolle zu zentralisieren. Und schließlich braucht man einheitliche Zugangsmechanismen zu den verschiedenen Ressourcen, die von Programmen und Dateien über Datenbanken und Sensoren bis hin zu Netzwerken reichen können. Dazu wird eine hohe Effizienz verlangt, um einen gewissen Grad von Funktionalität garantieren zu können (oft mit dem englischen Begriff „Quality of Service“ bezeichnet). Es soll der Zugriff auf die Ressourcen planbar sein („Scheduling“) und natürlich soll über den Verbrauch von Ressourcen auch Buch geführt werden („Accounting“).

## 4.2 Abgrenzung zu existierenden Technologien

Bisherige Technologien im Bereich des verteilten Rechnens haben diese Anforderungen nicht oder zumindest nicht vollständig erfüllt. Zum Beispiel haben die im Internet eingesetzten Technologien den Informationsaustausch zum Ziel, aber nicht die gemeinsame Benutzung von Rechenkapazitäten an verschiedenen Orten. Sogenannte „Business to Business“-Technologien wie Marktplätze,

Dienste im World Wide Web – oft als „Webservice“ bezeichnet – oder die unter dem Sammelbegriff „Electronic Data Interchange“ (EDI) bezeichneten Verfahren, dienen dem Informationsaustausch und benutzen dazu oft zentrale Server [34]. Auf der anderen Seite stehen Technologien wie „Common Object Request Broker Architecture“ (CORBA) und „Enterprise JAVA“, die die gemeinsame Benutzung von Ressourcen ermöglichen, allerdings nur in einer einzigen virtuellen Organisation mit zentralisierten Konfigurationen.

Desweiteren existieren Protokolle und Verfahren zur gemeinsamen Verwendung von Speicherplatz sowie zur entfernten Ausführung von Programmen. Anbieter solcher Dienste werden mit dem englischen Begriff „Storage Service Provider“ (SSP) beziehungsweise „Application Service Provider“ (ASP) bezeichnet. All diese Dienste sind aber nur sehr speziell einsetzbar; die Verwendung unterliegt starr festgelegten Protokollen und Regeln. Beispielsweise wird oft die Benutzung eines virtuellen privaten Netzwerks (VPN) vorgeschrieben.

## 4.3 Existierende Grid-Implementierungen

Es existiert eine Reihe von Projekten, die sich mit der Herstellung von Software beschäftigen, deren Funktionalität die Ideen des Grid-Computings teilweise umsetzen. Diese sind größtenteils im Bereich der öffentlichen Forschung entstanden, teilweise im kommerziellen Umfeld beheimatet. Die folgende Liste soll einige der wichtigsten Projekte ohne Anspruch auf Vollständigkeit aufzählen:

### 1. Globus Toolkit

Das Globus Toolkit [35] ist durch eine Kollaboration zwischen dem Argonne National Laboratory [36] in Argonne, Illinois, dem „Information Sciences Institute“ an der Universität von Kalifornien in San Diego [37], der Universität von Chicago [38], der Universität von Edinburgh [39] und dem schwedischen Zentrum für Parallelcomputer [40] entstanden. Es ist Quelltext offen unter einer Open-Source-Lizenz erhältlich und hat sich sehr schnell im Forschungsbereich zum Quasi-Standard für die grundlegende Funktionalität etabliert. Der folgende Abschnitt 4.3.1 widmet sich deswegen detailliert dieser Software.

### 2. European Data Grid (EDG)

Das „European Data Grid“ war ein von der Europäischen Union gefördertes Projekt mit der Zielsetzung, die weltweit verteilten Rechen- und Speicherkapazitäten, die vor allem im Bereich der Hochenergiephysik vorhanden sind, gemeinschaftlich nutzbar zu machen. Die Software, die im

Rahmen dieses Projektes erstellt wurde, wird nun im Rahmen des Projektes „Enabling Grids for E-science in Europe“ (EGEE) weiterentwickelt und im Rahmen des „LHC Computing Grid“-Projektes angewandt. Wegen der grundlegenden Bedeutung wird auch hier im folgenden Abschnitt 4.3.2 detaillierter auf dieses Projekt eingegangen.

### 3. UNICORE

Das Projekt „UNiform Interface to COmbine REsources“ [41] - besser bekannt unter seiner Abkürzung UNICORE - ist im Umfeld der deutschen Supercomputer-Zentren entstanden. Es ist in JAVA implementiert. Sowohl UNICORE als auch das Nachfolgeprojekt „UNICORE plus“ (2000-2002) sind mittlerweile abgeschlossen. Abgelöst wurden sie durch das von der Europäischen Union geförderte EUROGRID-Projekt, welches mittlerweile aber auch beendet wurde (01. November 2000 - 31. Januar 2004). Auch dieses konzentrierte sich auf der softwareseitigen Vernetzung von Supercomputer-Zentren innerhalb Europas.

### 4. Legion

„Legion“ [42] war ein Projekt der Universität von Virginia in Charlottesville und wollte eine objektorientierte Architektur für Grids erschaffen. Eine der Zielsetzungen war es, keine zentrale Kontrollinstanz zu verwenden, sondern jede Ressource als unabhängiges Element zu betrachten. Dabei sollte die Skalierbarkeit erhalten bleiben. Es war nie Quelltext offen erhältlich und wird seit 2003 nur noch kommerziell innerhalb eines Softwarepaketes namens „Avaki“ [43] vertrieben.

### 5. Condor

„Condor“ [44] - ein Projekt der Universität von Wisconsin in Madison - ist im engeren Sinne kein Grid Projekt. Es ist ein sogenanntes „Workload Management System“, also ein System zum möglichst effizienten Verteilen von Rechenaufträgen auf einzelne Rechner. Dank umfangreicher Schnittstellen, auch an die Protokolle des Globus Projektes (siehe auch Kap. 4.3.1), hat es als globaler Verteilungsmechanismus in einigen Grid-Projekten – wie zum Beispiel auch im European Data Grid – Einzug gehalten. Dazu wurde der Verwaltungsteil der Software als „Condor-G“ abgespalten und wird separat ausgeliefert.

Neben den hier vorgestellten Projekten gibt es noch eine unüberschaubare Anzahl weiterer Projekte mit mehr oder weniger gleichen Zielsetzungen. Das Global Grid Forum [45] kann man dabei als übergeordnete Initiative verstehen, die die Zielsetzung hat, die grundlegenden Funktionalitäten, Protokolle und Schnittstellen der verschiedenen Initiativen zu standardisieren, um damit eine gewisse Interoperabilität und eine grundlegende Planung für zukünftige Initiativen zu schaffen. Die für diese Arbeit wichtigsten Initiativen, auf deren Software

diese Arbeit beruht, sind das Globus Toolkit und das European Data Grid, die daher im Folgenden näher erläutert werden.

### 4.3.1 Das Globus-Toolkit

Das Globus-Toolkit ist das Ergebnis der Arbeit des Globus-Projektes. Es ist eine Software, die als „Middleware“ eine Zwischenschicht zwischen dem Benutzer, den benutzereigenen Programmen und dem Betriebssystem bildet. Es bietet verschiedene Dienste an, die zusammen die grundlegende Funktionalität ergeben, die nötig ist, um ein Grid aufzubauen, wie es eingangs in Kapitel 4.1 definiert wurde.

Es ist in folgende Komponenten unterteilt<sup>1)</sup>:

- **Globus Resource Allocation Manager (GRAM)**  
Diese Komponente ist zuständig für die Bereitstellung („Allocation“) von Ressourcen und für die Erstellung, Überwachung und Verwaltung von Prozessen. Die Beschreibung von Anforderungen eines Prozesses muss dabei in der Globus-eigenen Sprache „Resource Specification Language (RSL)“ erfolgen.
- **Grid Security Infrastructure (GSI)**  
Hier wird der Benutzer einmalig für die Benutzung im Grid identifiziert. Diese Komponente enthält die notwendige Software, die es dem lokalen Administrator ermöglicht, Zugriffsrechte zu definieren und Zuordnungen von lokalen zu Grid-weiten Benutzern einzurichten. Detailliert wird diese Komponente in [46] beschrieben.
- **Monitoring and Discovery Service (MDS)**  
Der Informationsdienst des Globus Toolkits basiert auf dem Lightweight Directory Access Protocol (LDAP). Eine detaillierte Beschreibung findet sich in [47]. Erweitert um einen Suchdienst, ermöglicht MDS eine Umgebung, um Statusinformationen wie den Zustand einzelner Rechner, Netzwerke oder den Ort abgespeicherter Dateien zu finden.
- **Global Access to Secondary Storage (GASS)**  
Ermöglicht den Zugang zu lokalen Massenspeichersystemen. Beinhaltet auch die nötigen Komponenten, um auf Daten Grid-weit lesend und schreibend zuzugreifen und zu verwalten (löschen, verschieben, etc.). Auch zu dieser Komponente gibt es eine detaillierte Beschreibung in [48].

---

<sup>1)</sup>Die Bezeichnungen werden hier der besseren Lesbarkeit wegen in ihrer englischen Originalfassung vorgestellt und nicht übersetzt.

- **Nexus & globus\_io**  
Kommunikations-Modul für benutzereigene Programme.
- **Heartbeat Monitor (HBM)**  
Überwachungssystem für Rechnerkomponenten; ermöglicht das schnelle Auffinden fehlerhafter Rechner [49].

Das Globus-Toolkit ist in der Programmiersprache C implementiert, Schnittstellen zu anderen Programmiersprachen wie JAVA und Python sind für die wichtigsten Schnittstellen vorhanden.

Mittlerweile ist das Globus-Toolkit in der Version 4 verfügbar. Für die vorliegende Arbeit wurde jedoch die stabile Version 2 verwendet. Auch das European Data Grid Projekt, welches im folgenden Abschnitt 4.3.2 beschrieben wird, basiert auf dieser Version des Globus-Toolkits.

Die folgende Tabelle soll einen Überblick über die Protokolle geben, die in einem auf dem Globus-Toolkit basierten Grid verwendet werden, sowie jeweils einen Hinweis darauf, mit welchem klassischen Protokoll diese Dienste von ihrer Funktionalität her verglichen werden können.

Aufgabe	Standard Protokoll	Grid Äquivalent	Kommandozeilenbefehl
<b>Zugriff auf Rechner</b>	telnet, rlogin, rsh, ssh, ...	GRID Security Infrastructure [46]	globusrun, globus-job-run, ...
<b>Datentransfer</b>	ftp, scp, ...	Global Access to Secondary Storage (GASS) [48], Grid-FTP [50]	globus-url-copy
<b>Ressourcensuche</b>	keine (externe Software wie OpenLDAP)	Metadata Directory Service (MDS) [47]	ldapsearch
<b>Überwachung</b>	keine (externe Software)	Heartbeat Monitor (HBM) [49]	keine

Tabelle 4.1: Überblick über die Grid-Protokolle des Globus-Toolkit v2

### 4.3.2 Das European Data Grid Projekt

Das „European Data Grid Projekt“ – im Folgenden auch kurz EDG genannt – war ein von der Europäischen Union gefördertes Grid Projekt, welches von Ja-

nuar 2001 bis März 2004 existierte und vom europäischen Zentrum für Teilchenphysik CERN<sup>2)</sup> geleitet wurde. Es ist mittlerweile in das EGEE-Projekt übergegangen.

Mit einer Gesamtförderung von fast zehn Millionen Euro seitens der Europäischen Union wurde im Rahmen dieses Projektes die Grundlage für die Gründung eines Grid für die Hochenergiephysik geschaffen, aber auch die Biologie und die Erdobservierung waren Zielgruppen dieses Projektes. Dabei wurden 500 Benutzer in zwölf virtuellen Organisationen registriert und eine Rechnerinfrastruktur geschaffen, die von mehr als zwanzig Instituten in ganz Europa bedient wurden.

Der erste Benutzer konnte seinen Rechenauftrag im Rahmen der Hochenergiephysik am 11. Dezember 2001 bearbeiten lassen.

Während die erste Version der entwickelten Software noch auf dem Globus-Toolkit in der Version 1 basierte, ist das Projekt bereits Mitte 2001 auf das Globus-Toolkit Version 2, wie im vorherigen Kapitel 4.3.1 beschrieben, übertragen worden.

Dabei geht die Funktionalität naturgemäß über die des Globus-Toolkits heraus. Da gerade im Rahmen der Hochenergiephysik sehr große Datenmengen anfallen, wurden zur Verwaltung und zum Zugriff auf Daten neue Softwarestrukturen geschaffen.

Im EDG gibt es verschiedene Dienste, die auf unterschiedlichen Rechnern installiert sind. Zum besseren Verständnis werden diese Dienste hier aufgelistet und beschrieben<sup>3)</sup>:

- **User Interface - kurz UI**

Die Schnittstelle zum Benutzer. Hier meldet sich der Benutzer an, verwaltet seine Rechenaufgaben und authentifiziert sich auch einmalig gegenüber dem EDG. Das UI beinhaltet zwei hauptsächliche Komponenten:

- Eine Beschreibungssprache, die die Anforderungen von Rechenaufgaben an die Ressourcen und die charakteristischen Eigenschaften der Rechenaufgabe beschreibt. Diese Sprache nennt sich „Job Description Language“ (JDL) und ist in [51] definiert.
- Eine Menge an Befehlen und Schnittstellen, die Objekte auf dem Grid manipulieren.

- **Storage Element - kurz SE**

Die Komponente zur Massenspeicherverwaltung. Es bildet verschiedene

---

<sup>2)</sup>CERN = Organisation Européenne pour la Recherche Nucléaire

<sup>3)</sup>Auch hier werden die englischen Bezeichnungen für die Dienste übernommen und nicht übersetzt.

Massenspeichersysteme in einem einheitlichen Protokoll, dem sogenannten „Storage Resource Manager“, ab. Dieses Protokoll erlaubt im Wesentlichen drei Arten von Operationen:

1. Eine Datei erstellen.
2. Eine Datei lesend öffnen.
3. Eine Datei kopieren.

Dabei unterscheidet man verschiedene Arten, um Dateien oder deren Kopien anzusprechen:

– **GUID** – Grid Unique Identifier

Eine Datei kann immer über die GUID angesprochen werden. Er wird bei der Registrierung einer Datei automatisiert zugewiesen, ist weltweit eindeutig und hat die Form `guid:<eindeutige Zeichenkette>`. Alle Kopien einer Datei tragen den selben GUID. Im Gegensatz zu Logical File Names (LFNs) hat aber eine Datei nie mehr als eine GUID.

Beispiel:

```
guid:579fa133-5751-4e23-bcd3-b0db36436902
```

– **LFN** – Logical File Name

Ein benutzerdefiniertes Synonym zum Zugriff auf Dateien. Der Benutzer kann für eine Datei einen beliebigen Namen vergeben, solange dieser im Bereich seiner VO noch nicht anderweitig verwendet wurde.

Beispiel:

```
myfile_run13
```

– **SFN** – Site File Name

Auch „Physical File Name“ genannt. Bezeichnet genau eine Kopie einer Datei. Das SFN enthält den Namen des Rechners, auf dem die Dateikopie gespeichert ist und den vollständigen Pfad der Datei. Damit ist ein SFN weltweit eindeutig.

Beispiel:

```
grid-se.physik.uni-wuppertal.de/gridstore/dzero/generated/  
2004-09-09/filea47f5a06-e521-418f-b938-1fe7e2c49bb5
```

– **SURL** – Storage Universal Resource Locator

Enthält ein SFN und das Protokoll zur Verwaltung für selbiges. Im EDG ist das Protokoll immer `srm://`. Es hat damit die selbe Funktion wie ein Kontrollprotokoll einer normalen Dateiübertragung im Internet, über das nur Verwaltungsdaten ausgetauscht werden.

Beispiel:

```
srm://grid-se.physik.uni-wuppertal.de/gridstore/dzero/  
generated/2004-09-09/filea47f5a06-e521-418f-b938-1fe7e2c49bb5
```

- **TURL** – Transfer Universal Resource Locator  
Besteht aus einem Zugriffsprotokoll und einem Speicherort für eine Dateikopie und ist damit weltweit eindeutig. Als Zugriffsprotokoll dient meist GridFTP, auch gsiftp genannt. Im Gegensatz zum SURL werden hier die physikalischen Daten übertragen.

Beispiel:

```
gsiftp://gppse01.gridpp.rc.ac.uk/flatfiles/01/data
/d0/d073888b9c936f8790a3cd58ea60625624
```

Im EDG-System existierten zwei Implementationen eines „Storage Elements“: ein klassischer Ansatz (der oft als „SE Classic“ bezeichnet wurde) und das sogenannte „WP5 Storage Element“ [52]. Letzteres ist eine komplette Neuentwicklung des European Data Grid Projektes und zeichnet sich vor allem durch eine schichtartige Softwarestruktur aus, wie sie in der folgenden Abbildung 4.1 dargestellt ist.

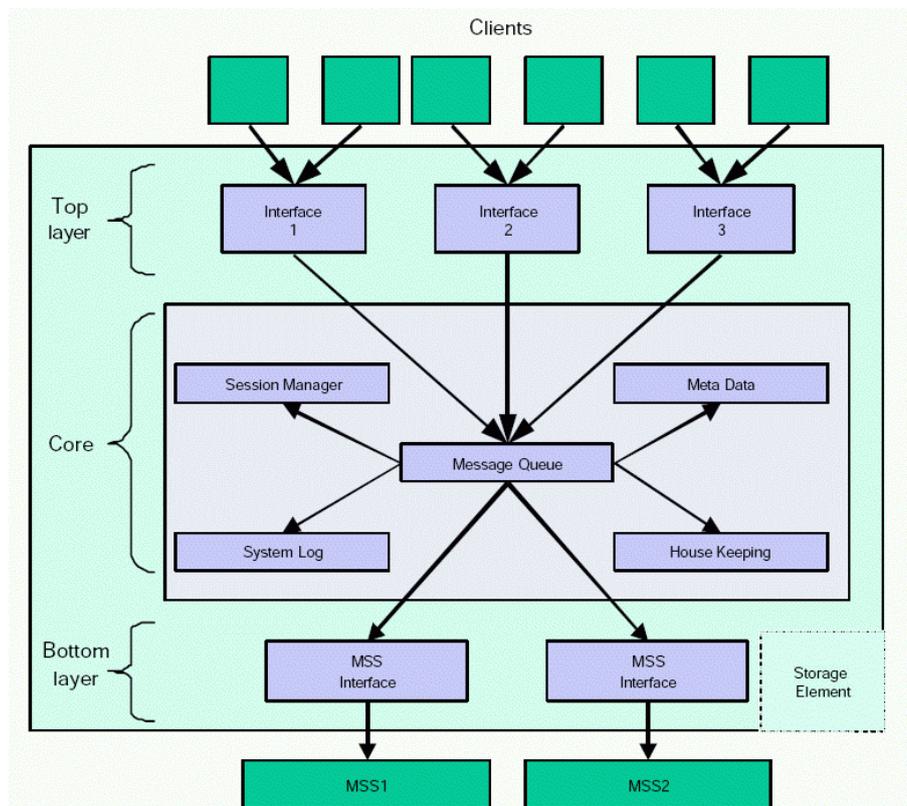


Abbildung 4.1: Schematische Darstellung der schichtartigen Softwarestruktur des „WP5 Storage Element“. Aus [52].

Die unterstützten Netzwerkprotokolle werden über die obere Lage implementiert. Dabei kommen vor allem die Protokolle RFIO [53] und Grid-FTP [50] zum Einsatz. Die mittlere Schicht beinhaltet die Verwaltungsstrukturen, insbesondere die Datenbank, in der die vorhandenen Dateien und deren physikalische Speicherorte abgelegt sind. Die untere Schicht dient dann zur Kommunikation mit den Massenspeichersystemen. Die interne Kommunikation wird über gepufferte Datenverbindungen – benannte Pipes oder als „Named Pipe“ bezeichnet – realisiert.

Der Datenfluss in diesem System wird durch die Abbildung 4.2 veranschaulicht:

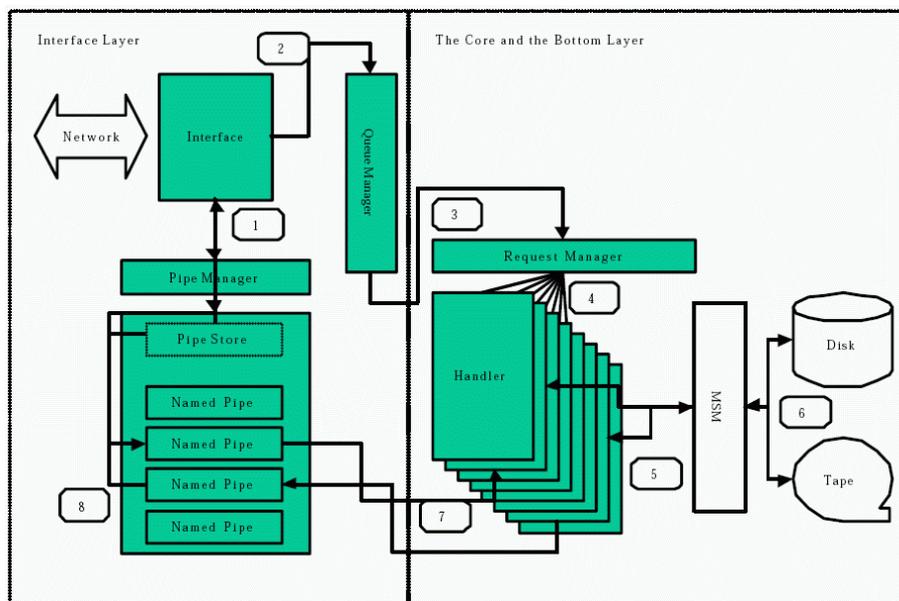


Abbildung 4.2: Illustration des Datenflusses durch das „WP5 Storage Element“. Aus [52].

Zunächst wird durch die Anforderung einer Schnittstelle durch den sogenannten „Pipe Manager“ eine definierte Anzahl von Datenverbindungen erzeugt (Ziffer „1“ in der Illustration 4.2). Durch die Anzahl der Verbindungen wird der Datendurchsatz optimiert. Daneben erzeugt das Interface eine Beschreibung (in XML) für die notwendigen Aufgaben, die von der mittleren und unteren Softwareschicht bearbeitet werden müssen („2“). Diese Aufgaben werden mit einem Algorithmus weitergereicht, der sicherstellt, dass nur genau so viele Aufgaben gleichzeitig abgearbeitet werden, dass das System nicht überlastet wird („3“). Diese Aufgaben werden dann durch sogenannte „Handler“ abgearbeitet („4“), die ihrerseits mit den Massenspeicherverwaltungssystem (MSS) kommunizieren („5“). Das MSS ver-

waltet die Zugriffe auf Festplatten- und Bandspeicher eigenverantwortlich („6“). Schließlich werden die Daten wieder an die Datenverbindungen zurückgeschrieben („7“ und „8“). Eine detailliertere Beschreibung des „WP5 Storage Elements“ befindet sich in [52].

- **Computing Element - kurz CE**

Die Schnittstelle zwischen den Rechenaufträgen der Benutzer und den „Worker Nodes“. Es benutzt ein „Batch System“, um die Liste der Rechenaufträge zu verwalten und abzuarbeiten. Es ist die einzige Komponente des EDG, in der Zugriffe von außen gestattet werden. Es besteht aus zwei Komponenten: dem „Gatekeeper“ und dem „Jobmanager“:

- Der *Gatekeeper* bildet die Schnittstelle zwischen dem Grid und dem lokalen Batch System. Das heißt er nimmt Rechenanforderungen („Jobs“) über Grid-Protokolle entgegen und leitet sie nach Prüfung an das Batch System weiter. Für die Prüfung ist der „Local Centre Authorization Service“ (LCAS) zuständig, welcher aufgrund der übermittelten Grid-Identität und der Zugehörigkeit dieser Identität zu einer virtuellen Organisation unter Zuhilfenahme des „Virtual Organization Membership Service“ (VOMS) über den Zugriff entscheidet. Desweiteren wird der „Local Credential MAPPING Service“ (LCMAPS) benötigt, der nach erlaubtem Zugriff die Zuordnung der Grid-Identität zu einer lokalen Identität übernimmt.
- Der *Jobmanager* stellt die Verbindung zum lokalen Batch System her und übernimmt die Überwachung der Rechenanforderung während der Ausführung.

Die folgende Darstellung 4.3 soll den Zusammenhang zwischen den einzelnen Komponenten verdeutlichen.

- **Worker Node - kurz WN**

Die eigentlichen Arbeitsmaschinen. Hier werden die Benutzeraufträge abgearbeitet.

- **Replica Catalog - kurz RC**

Enthält eine Sammlung von Diensten, die die Verwaltung von Dateikopien (sogenannte „Replica“) ermöglichen. Dabei wird auch der Zusammenhang zwischen physikalischen Dateinamen (SURL) zu globalen, logischen Dateinamen (LFN) – realisiert. Jede virtuelle Organisation benötigt diese Dienste einmal. Sie unterteilen sich in mehrere Dienste, die unter dem Begriff „Data Management Services“ zusammengefasst und in [54] genauer beschrieben werden:

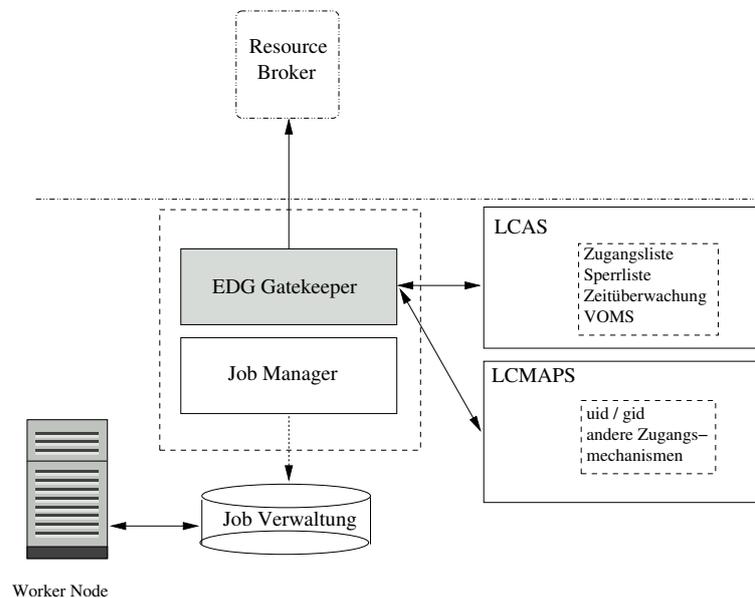


Abbildung 4.3: Die Subsysteme des Computing Elements. Vereinfachte Darstellung nach [54].

- „Replica Location Service“ (RLS)  
Sucht Kopien von Dateien innerhalb des EDG und speichert deren physikalische Dateinamen.
- „Replica Metadata Catalog“ (RMC)  
Fragt logische Dateinamen (LFN) ab und verwaltet diese.
- „Replica Optimization Service“ (ROS)  
Sucht nach der für eine Anforderung besten Dateikopie, so dass diese schnellstmöglich der Anwendung zur Verfügung gestellt werden kann.

Der Benutzer kommuniziert im Normalfall nicht mit den einzelnen Diensten, sondern über Kommandozeilenbefehle oder über andere Programme, die die Schnittstelle zur Anwendungsprogrammierung – Application Programming Interface, auch kurz API genannt – benutzen.

- **Information Index - kurz II**

Eine Datenbank, die den momentanen<sup>4)</sup> Zustand aller Ressourcen abbildet. Diese Informationen wertet der „Resource Broker“ aus. Wird mindestens einmal pro virtueller Organisation benötigt.

<sup>4)</sup>Hierbei ist zu beachten, dass „momentan“ nur als zeitnah zu verstehen ist.

- **Resource Broker - kurz RB**

Ein zentraler Dienst, der zu jeder Rechenanforderung einen möglichst günstigen Ort für deren Ausführung bestimmt. Dabei wird als günstig verstanden, dass die Anforderung möglichst schnell ausgeführt wird und dass möglichst wenig benötigte Daten transportiert werden müssen. Auch dieser Dienst wird mindestens einmal für jede virtuelle Organisation benötigt. In Kapitel 6.4.2 wird der RB genauer beschrieben.

- **Monitor Box - kurz MON\_BOX**

Dieser Dienst ist für die Überwachung und Buchführung eines Rechnerverbundes zuständig und wird einmal pro Institut installiert. Er beheimatet die *Relational Grid Monitoring Architecture* (kurz: *R-GMA*) [55], ein auf Web Services basierender Dienst, der auch zur Verteilung anwendungsspezifischer Daten genutzt werden kann. Sie bildet eine große, relationale Datenbank ab. Der Dienst basiert auf einem Standard des Global Grid Forums, wobei die Architektur dieses Dienstes aus drei Komponenten besteht: Konsumenten (*Consumers*), Produzenten (*Producers*) und Verzeichnisdienste (*Registries*). Dies ist in der folgenden Abbildung 4.4 verdeutlicht. Die Produ-

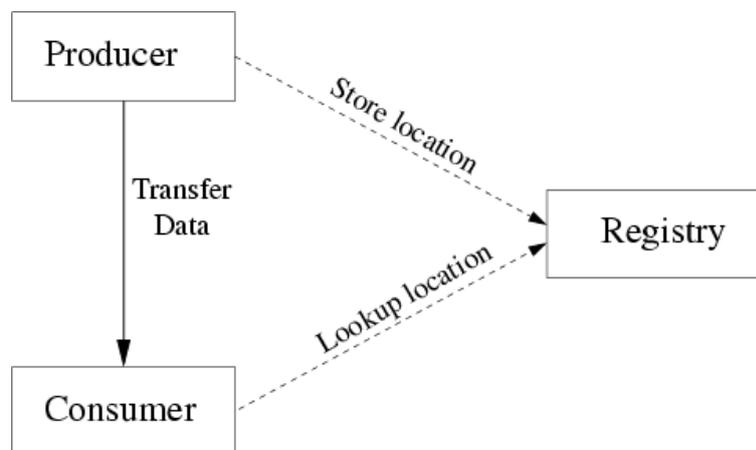


Abbildung 4.4: Die Dienste der „Grid Monitoring Architecture“. Aus [56].

zenten registrieren sich beim Verzeichnisdienst und beschreiben dort die Art und Struktur der Information, die sie im Grid zur Verfügung stellen wollen. Die Konsumenten können nun den Verzeichnisdienst anfragen, um herauszufinden, welche Art von Information existiert und wo diese verfügbar ist. Der Datentransfer findet danach nur noch zwischen Produzent und Konsument statt, in der Abbildung 4.4 als durchgezogene Linie angedeutet.

- **Certificaton Authority - kurz CA**

Eine zentrale Instanz, die die Echtheit von Diensten, Rechnern und Personen mit Hilfe von X.509 Zertifikaten Grid-weit elektronisch bescheinigt. Im Prinzip wird eine CA nur einmal im Grid benötigt, aber aus praktischen Erwägungen hat man diese Instanzen an Staaten gekoppelt. Für Deutschland wurde eine CA am Forschungszentrum Karlsruhe etabliert.

Die folgende Abbildung 4.5 zeigt schematisch die Struktur des Grid, welches aus EDG-Komponenten gebildet wird. Die verschiedenen Möglichkeiten, an einem solchen Grid teilzunehmen, werden durch typische Institutsgrößen angedeutet. Es sind aber auch andere Konfigurationen möglich. Typischerweise werden zentrale Dienste von großen Zentren betrieben.

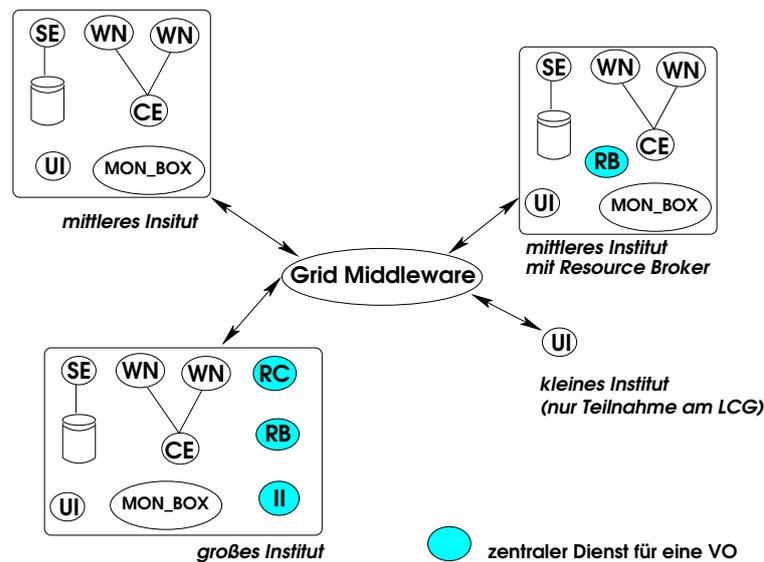


Abbildung 4.5: Schematischer Aufbau eines Grids aus EDG-Komponenten. Der Übersichtlichkeit halber wurden die logischen Verbindungen zwischen den Komponenten nicht gezeichnet.

Eine typisches Institut wie eine Universität ist dabei oben links angedeutet. Es besteht aus einem „User Interface“, einem „Storage Element“ samt angeschlossenen Speicher, einem „Computer Element“ und diverser angeschlossener „Worker Nodes“ sowie einer „Monitor Box“. Da in diesem Szenario bereits vier Rechner nur für Verwaltungsdienste benötigt werden (die nicht zum verteilten Rechnen zur Verfügung stehen), werden kleinere Institute wohl eher die Teilnahme über ein einzelnes „User Interface“ realisieren, wobei diese dann keine Ressourcen für andere Benutzer bereitstellen. Die Bergische Universität hingegen stellt

neben Speicherplatz und Rechenknoten auch noch einige weitere Dienste zur Verfügung (siehe Kapitel 6.2) und ist daher oben rechts angedeutet. Die großen internationalen Forschungszentren wie CERN und DESY<sup>5)</sup> werden unten links angedeutet. Sie stellen oft alle nötigen Dienste für die virtuellen Organisationen zur Verfügung, deren Experimente am jeweiligen Standort beherbergt sind. So stellt CERN alle Dienste für die vier LHC-Experimente ATLAS<sup>6)</sup>, ALICE<sup>7)</sup>, CMS<sup>8)</sup> und LHCb<sup>9)</sup> bereit.

---

<sup>5)</sup>DESY = Deutsches Elektronen Synchrotron. Das DESY hat Standorte in Hamburg und in Zeuthen bei Berlin

<sup>6)</sup>ATLAS = A Toroidal LHC AparatuS

<sup>7)</sup>ALICE = A Large Ion Collider Experiment at CERN

<sup>8)</sup>CMS = The Central Muon Solenoid

<sup>9)</sup>LHCb = The Large Hadron Collider beauty Experiment

# Kapitel 5

## Erstellung einer Grid-basierten Softwareinfrastruktur für AMANDA

### 5.1 Einleitung

Ziel dieses Teils der Arbeit soll es sein, basierend auf dem in Kapitel 4.3.1 vorgestellten Globus-Toolkit bereits frühzeitig ein System zur automatischen Generierung von Simulationsdaten für das AMANDA-Experiment aufzubauen. Als Grundlage wurde das Globus-Toolkit gewählt, da es zum einen den quasi-Standard in der Hochenergiephysik darstellt. Das „European Data Grid“ war zur damaligen Zeit noch nicht in einem für größere und längere Anwendungen brauchbaren Zustand.

Mit diesem Projekt sollten mehrere Ziele verfolgt werden:

- Hauptsächlich sollte der steigende Rechenzeitbedarf der AMANDA-Kollaboration durch Zusammenschlüsse der Ressourcen gedeckt werden.
- Es sollte gezeigt werden, dass auch nicht an Grid-Umgebungen angepasste Software in einer solchen Umgebung erfolgreich eingesetzt werden kann.
- Der Verwaltungsaufwand bei der verteilten Produktion von Simulationsergebnissen sollte reduziert werden, weil unter Verwendung von Grid-Technologien die ganze Produktion zentral gesteuert werden kann.

Um diese Ziele zu erreichen, wurde auf dem Globus-Toolkit basierend eine Software mit folgendem Eigenschaften erstellt:

- Dem Benutzer soll der einheitliche Zugriff auf eine zu erstellende zentrale Software-Datenbank gegeben werden. Die in Kapitel 2.4 vorgestellte Software der AMANDA-Kollaboration soll in dieser Datenbank vorliegen und direkt benutzbar sein.

- Für den Benutzer soll der Zugriff auf die Rechner der verschiedenen beteiligten Institute transparent sein. Das heißt insbesondere, dass die Besonderheiten des jeweils benutzten Rechenknotens verborgen bleiben.
- Die Ein- und Ausgabedateien sollen für den Benutzer von seinem Arbeitsplatz aus automatisch an den verwendeten Rechenknoten übertragen werden. Alternativ soll auch jede andere Ressource, für die der Benutzer die nötigen Zugriffsrechte hat, angegeben werden können.
- Um die Verwaltung größerer Anzahlen von Rechenaufträgen zu erleichtern, soll eine graphische Benutzeroberfläche erstellt werden.
- Neben der Software aus der zentralen Software-Datenbank sollen auch benutzerdefinierte Rechenaufträge abgearbeitet werden können.
- Bei dynamisch gelinkter Software – das ist Software, die zur Ausführung bestimmte dynamische Bibliotheken benötigt – soll überprüft werden, ob diese Software auf dem Zielsystem lauffähig ist und für den Fall, dass dort Bibliotheken fehlen, sollen diese automatisch transferiert werden.

Die von der Kollaboration benutzte Software ist, wie in Kapitel 2.4 bereits vorgestellt, aus mehreren Programmiersprachen zusammengesetzt (C, C++, JAVA, FORTRAN, ...). Sie ist nicht für den Einsatz in einer Grid-Infrastruktur konzipiert. Ein möglicher Ansatz würde darin bestehen, die Software zu modifizieren, so dass sie direkt die entsprechenden Programmierschnittstellen des Globus Toolkits, wie sie unter 4.3.1 vorgestellt wurden, benutzt.

Dieses erscheint zwar wünschenswert, jedoch ist dies insbesondere für den in FORTRAN geschriebenen Teil der Software nicht durchführbar, da es für diese Programmiersprache keine Schnittstellen zu Globus gibt. Auch stellt es einen erheblichen Eingriff in die vorhandene Software dar und würde deren Weiterentwicklung zumindest zwischenzeitlich behindern.

Die hier verfolgte Lösung baut darauf auf, eine Softwareschicht zwischen die von AMANDA verwendete Software einerseits und dem Globus Toolkit andererseits zu legen. Die AMANDA-Software wird dabei in einem „Sandbox“ Mechanismus gestartet, in dem alle benötigten Dateien und Bibliotheken bereitgestellt werden. Dieser Mechanismus sowie der Aufbau und die Funktionsweise der gesamten Software, werden im folgenden Abschnitt 5.2 beschrieben.

## 5.2 Aufbau des Grid-Projektes für AMANDA

Um herauszuarbeiten, welche Vorteile die Verwendung von Grid-basierten Mechanismen bei der gestellten Aufgabe haben, wird zunächst der konventionelle

Ansatz vorgestellt, der in vielen rechenintensiven Bereichen der Wissenschaft vorzufinden ist.

Der Rechnerverbund ist durch ein (meist sehr schnelles) lokales Netzwerk – auch „Local Area Network“, abgekürzt LAN, genannt – verbunden. Ein Rechner agiert dabei als Server und hat die Aufgabe, den Plattenplatz und die Benutzerdatenbank vorzuhalten. Diese Informationen werden über netzwerkbasierende Protokolle wie das sogenannte „Network File System“ (NFS) [57–59] und das „Network Information System“ (NIS) [60] zu den Rechenknoten übertragen.

Das eigentliche Programm wird auf einem der Rechenknoten ausgeführt. Dazu muss es selbst sowie jede benötigte Datei (seien es nun Bibliotheken, Daten, Konfigurationsdateien, ...) über das Netzwerk zu dem Zeitpunkt, an dem diese Dateien benötigt werden, übertragen werden. Änderungen an der Datei oder Ausgaben des Programms müssen zeitgleich zurück zum Server übertragen werden. Abbildung 5.1 soll dies illustrieren.

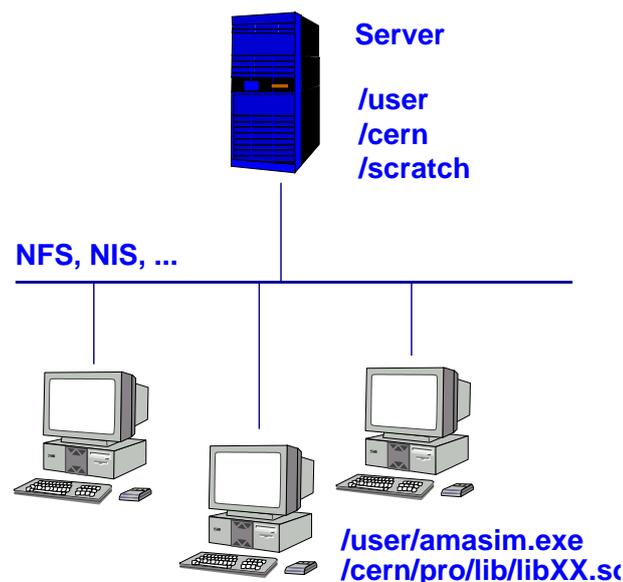


Abbildung 5.1: Ein konventioneller Ansatz zum Aufbau eines verteilten Rechensystems. Nach [61].

Die ständigen Übertragungen von Dateien stellen in schnellen lokalen Netzwerken normalerweise kein größeres Problem dar, zumindest nicht, solange nicht zu viele Rechenknoten gleichzeitig Dateien anfragen.

Möchte man nun aber in verteilten Systemen Rechenaufgaben ausführen, die mit einem Fernnetz – auch „Wide Area Network“, abgekürzt WAN, genannt – verbunden sind, so kann man dieses Lösung nicht übertragen. Zum einen nimmt der Datendurchsatz bei der Nutzung von NFS als Übertragungsprotokoll sehr

schnell mit der Anzahl der Rechenknoten ab. Dieses wurde durch zahlreiche Studien bestätigt. Eine Messung [62] zeigt die folgende Abbildung 5.2. Man erkennt deutlich den starken Abfall der Übertragungsgeschwindigkeit mit zunehmender Anzahl von Rechenknoten. Bei Fernnetzen (WANs) würde der Effekt noch durch hohe Latenzzeiten verstärkt. Dieses würde die Leistung eines Rechnerverbundes stark limitieren.

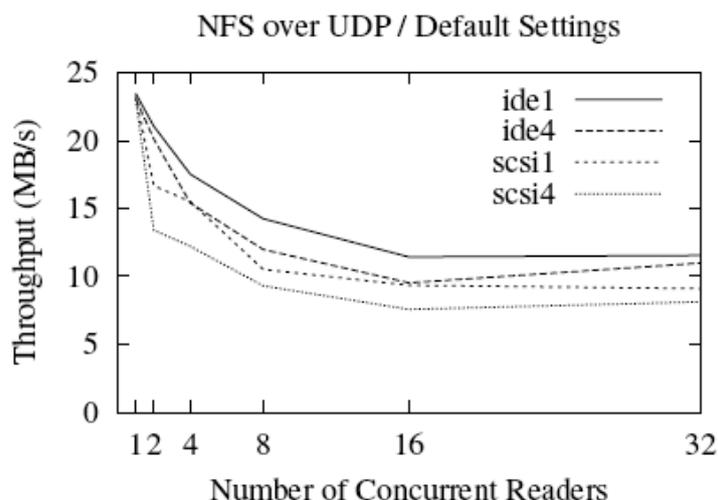


Abbildung 5.2: Übertragungsgeschwindigkeit (in MB/s) bei steigender Anzahl von Rechenknoten. Die Bezeichnungen ide1, ide4, scsi1 und scsi4 geben verschiedene Typen von Festplatten an. Aus [62].

Zum anderen ist eine zentrale Verwaltung von Zugriffsrechten in einem weltweit verteilten Rechnerverbund unwirtschaftlich, fehleranfällig und vor allem sehr unflexibel.

Der hier verfolgte Ansatz setzt auf das Globus-Toolkit und die damit zur Verfügung stehenden Protokolle auf. Die Ideen und die Implementierung wird im folgenden Kapitel genauer beschrieben.

### 5.3 Beschreibung des gewählten Ansatzes

Die hier entwickelte Lösung basiert komplett auf der Globus-Toolkit Version 2 und ist in der Programmiersprache Python realisiert. Sie ist in folgende Teile gegliedert:

1. Jeweils ein Softwarepaket für die in Kapitel 2.4 vorgestellten Teile der AMANDA-Software.
2. Eine Softwareschicht (im Folgenden als „Sandbox“ bezeichnet), die um die unter 1. genannten Pakete liegt und die folgende Aufgaben erfüllen soll:
  - (a) Ablaufsteuerung,
  - (b) Erzeugen einer temporären Umgebung für die AMANDA-Software,
  - (c) Benachrichtigung des Benutzers über den Zustand seiner Rechenanforderung,
  - (d) Transport des Softwarepakets von einem zentralen Ort zum Rechenknoten,
  - (e) Transport von nötigen Konfigurationsdaten zum Rechenknoten,
  - (f) Transport von Ergebnissen zu dem vom Benutzer gewählten Ort,
  - (g) Fehlerbehandlung,
  - (h) Aufräumen der temporären Umgebung nach Abarbeitung der Rechenanforderung.
3. Einer Softwareschicht für benutzereigene Anwendungen, die zusätzlich zu der in 2. genannten Lösung Abhängigkeiten von dynamischen Bibliotheken auflöst.
4. Eine graphische Benutzeroberfläche zur einfachen Bedienung durch den Benutzer.

Die folgenden Abschnitte erläutern nun die einzelnen Programmteile genauer.

### **5.3.1 Softwarepakete für die AMANDA-Software**

Die Hauptaufgabe der Bündelung der AMANDA-Software in Pakete besteht darin, diese Software unabhängig von anderen Softwarepaketen zu machen. Benötigte dynamische Bibliotheken, die nicht unmittelbar zum Umfang eines Betriebssystems gehören, sind hier eingefügt worden.

Insbesondere im Fall vom Programmpaket MMC, welches in JAVA implementiert ist (vergleiche Kapitel 2.4), ist es notwendig, eine sogenannte „JAVA Virtuelle Maschine“ beizufügen. JAVA-Programme sind nicht unmittelbar auf einem Betriebssystem lauffähig, sondern benötigen eine Schnittstelle zum jeweiligen Betriebssystem, welche als „virtuelle Maschine“ bezeichnet wird.

### 5.3.2 Der Sandbox-Mechanismus

Der Sandbox-Mechanismus ist dafür zuständig, die nicht Grid-fähige Software mit einer Softwareschicht zu umgeben, die nach außen hin die Kommunikation durchführt. Dazu wurde zunächst für die in Kapitel 2.4 genannten Softwarepakete im Hinblick darauf analysiert, welche Dateien durch den Benutzer veränderbar sein müssen und welche ins Softwarepaket mit aufgenommen werden konnten. Die veränderlichen Dateien, die zumeist zur Konfiguration der Parameter dienen, werden von der auf dem Rechner des Benutzers laufenden graphischen Benutzeroberfläche (vergleiche Kapitel 5.3.4) erzeugt und durch die Sandbox über das GASS-Protokoll transportiert. Zusätzlich werden Pseudo-Zufallszahlen erzeugt, die dynamisch in die Konfigurationsparameter eingefügt werden. Dieses ist allgemein bei Simulationen notwendig, um nicht immer wieder die gleichen Ereignisse zu berechnen.

Die AMANDA-Software wird von dem zentralen Speicherplatz mit dem GASS [48] (vergleiche Kapitel 4.3.1) oder dem GridFTP-Protokoll [63] übertragen. Das GridFTP-Protokoll hat dabei den Vorteil, bei größeren Datenmengen höhere Übertragungsraten zu ermöglichen. Dieses wird durch eine höhere Flexibilität bei den netzwerkspezifischen Parametern wie TCP Paketgröße [63, 64] erreicht, welches einen höheren Konfigurationsaufwand nötig macht. Daher wird dieses Protokoll in der vorliegenden Anwendung nur zum Transfer großer Datenmengen von zentral gewarteten Datenspeichern benutzt.

Der Ablauf einer Rechenanforderung, wie sie auf dem ausführenden Rechenknoten stattfinden, ergibt sich aus dem folgenden Diagramm 5.3.

Die Information des Benutzers über den Status seiner Anforderungen ist bei verteilten Rechnersystemen wichtig, da dieser nicht interaktiv in den Ablauf eingreifen kann. Daher wird er zunächst über den Beginn seiner Rechenanforderung informiert und erhält damit auch die Information, wo diese Anforderung abgearbeitet wird.

Danach wird jeder einzelne Ablaufschritt auf seine erfolgreiche Ausführung hin überprüft und im Falle von nicht vorgesehenen Ergebnissen wird der Benutzer informiert und die Ausführung an dieser Stelle abgebrochen.

Zunächst wird dabei ein temporärer Speicherplatz erstellt und dorthin das Softwarepaket und die vom Benutzerarbeitsplatz transferierten Dateien kopiert. Nun wird eine Umgebung erstellt, wie es nach Maßgabe des Softwarepakets erwartet worden wäre. Insbesondere wird die Betriebssystemumgebung so konfiguriert, dass die dynamischen Bibliotheken, die zur Ausführung des Softwarepaketes erforderlich sind, von den entsprechenden Teilen des Betriebssystems gefunden werden.

Nach dem Transfer der spezifischen Dateien für diese Rechenanforderung wird eine Überwachungsumgebung gestartet, die den Ablauf der AMANDA-Software

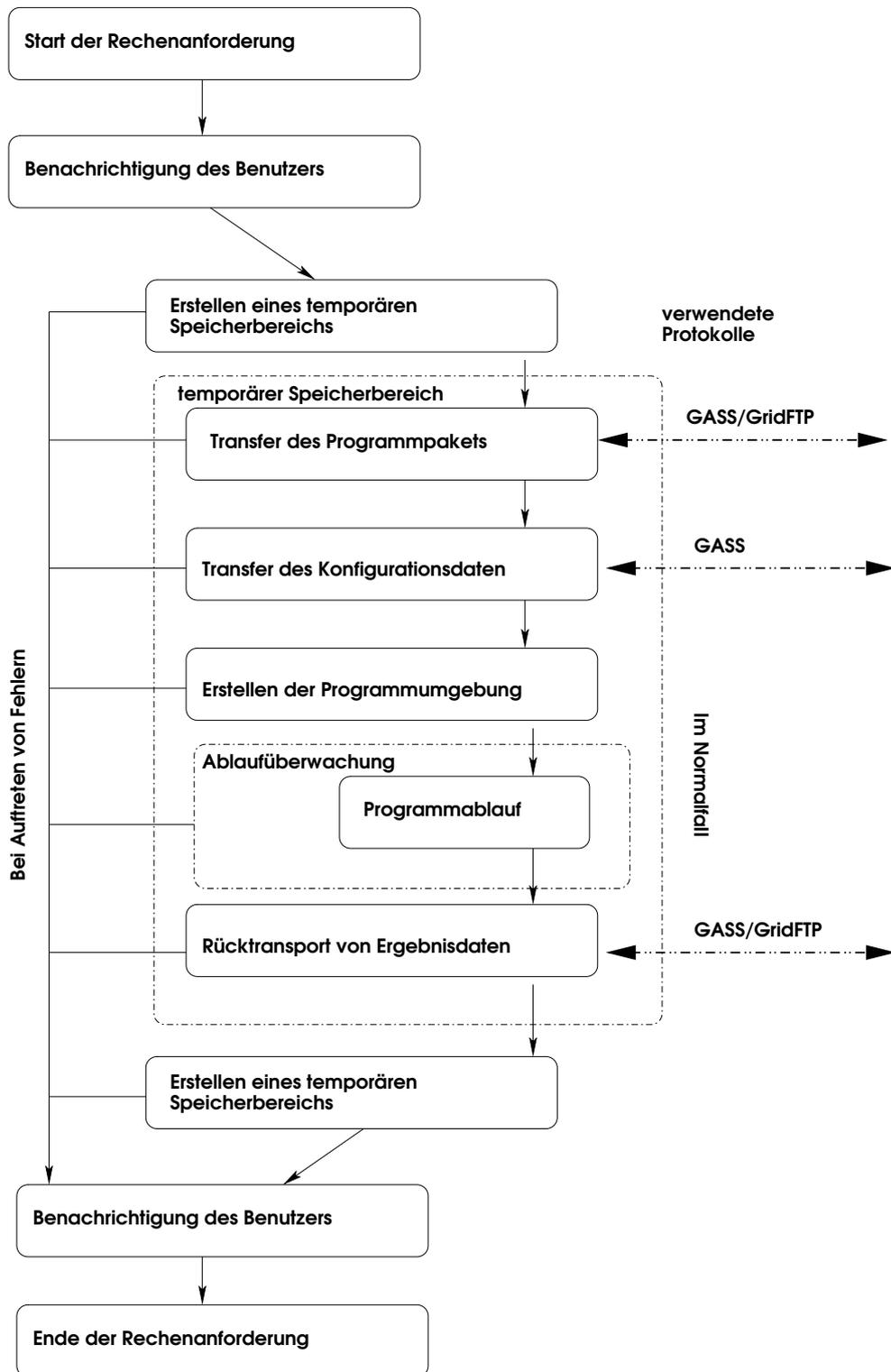


Abbildung 5.3: Ablauf einer Rechenanforderung.

startet und überwacht. Nach einem erfolgreichem Ablauf werden dann die Resultate wieder mit Hilfe des GASS- oder GridFTP-Protokolls zum angegebenen Speicherort transportiert und der Benutzer über den erfolgreichen Ablauf informiert. Im Falle von Problemen während des Ablaufs wird der Benutzer über die von der AMANDA-Software oder vom Betriebssystem ausgegebenen Fehlermeldungen informiert. In jedem Fall wird der temporäre Speicherplatz noch gelöscht.

### 5.3.3 Erweiterung für beliebige Software

Der im vorherigen Abschnitt behandelte Fall ging davon aus, dass die Software zentral gebündelt und auf Abhängigkeiten analysiert wurde (vergleiche Kapitel 5.3.1). Will man nun eine verteilte Rechenumgebung für beliebige Software schaffen, so sind folgende Probleme zusätzlich zu lösen:

1. Abhängigkeiten zu externen dynamischen Bibliotheken können nicht zentral analysiert und die benötigten Bibliotheken bereitgestellt werden.
2. Die benutzereigene Software ist im Normalfall nicht darauf ausgelegt, in verteilten Rechenumgebungen eingesetzt zu werden und entsprechende Protokolle zur Datenübertragung zu benutzen.

Für beide Probleme wurden im Rahmen der vorliegenden Arbeit Lösungen entwickelt, die im Folgenden diskutiert werden.

#### Abhängigkeiten von dynamischen Bibliotheken

Da die auszuführende Software auf dem Rechner des Benutzers ausführbar ist, wird schon hier eine automatische Analyse der von dem Programm benötigten dynamischen Bibliotheken vorgenommen. Dadurch wird eine Liste erstellt, die auf die Sandbox-Softwareschicht zu dem ausführenden Rechnersystem übertragen wird. Um Konflikte mit verschiedenen Versionen derselben Bibliothek zu vermeiden, werden Prüfsummen der Bibliotheken erstellt und mit übertragen. Zusätzlich wird ein sogenannter GASS-Server gestartet, ein Prozess, der es dem Sandbox-Mechanismus auf dem ausführenden System unter anderem ermöglicht, eventuell benötigte, aber dort nicht vorhandene, dynamische Bibliotheken zu übertragen.

Der Sandbox-Mechanismus auf dem ausführenden System überprüft nun vor dem Start der Software, ob alle benötigten dynamischen Bibliotheken in der jeweils richtigen Version vorhanden sind. Bibliotheken, für die dies nicht der Fall ist, werden nachgeladen und das Betriebssystem entsprechend konfiguriert.

## Zugriff auf Dateien

Regelmäßig ist auch der Zugriff auf Dateien vom Rechnersystem des Benutzers erforderlich, um den Ablauf der Rechenanforderung zu gewähren. Dieses wird ähnlich wie im Falle der vorher diskutierten Abhängigkeit von dynamischen Bibliotheken dadurch erreicht, dass der Sandbox-Mechanismus die benötigten Dateien über den GASS-Server vom Rechner des Benutzers kopiert. Über den gleichen Prozess werden die Ergebnisse der Rechenanforderung an den Arbeitsplatz des Benutzers zurück kopiert. Hierzu werden alle Dateien, die vom Benutzerprozess während des Ablaufs neu erstellt wurden – dieses sind neben den Ergebnissen der Berechnungen auch oft Dateien, die den Ablauf protokollieren – zurück kopiert. Zusätzlich kann der Benutzer weitere Dateien über die im folgenden Kapitel 5.3.4 vorgestellte graphische Benutzeroberfläche vorgeben.

Zusammenfassend verdeutlicht Abbildung 5.4 noch einmal den Ablauf dieses Teils des Systems:

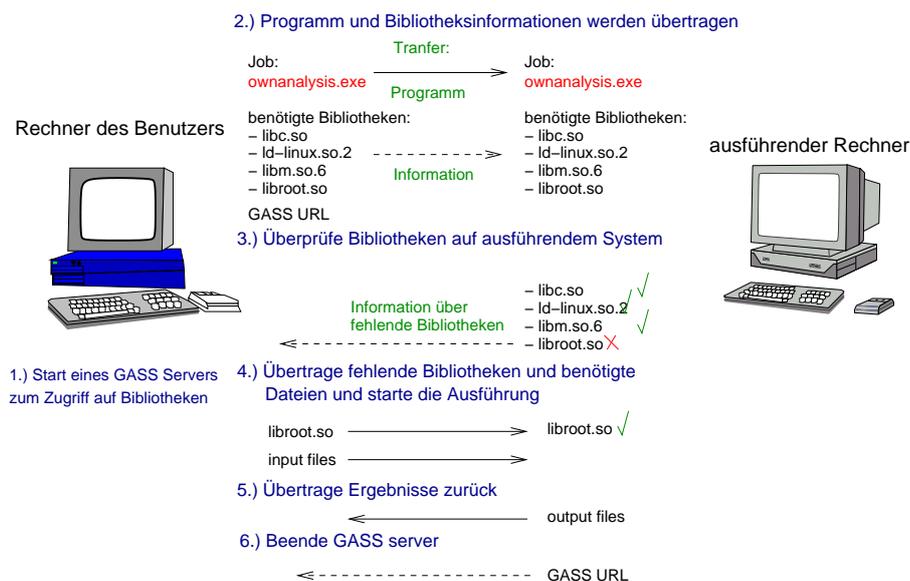


Abbildung 5.4: Ablauf des Mechanismus zum Auflösen von Abhängigkeiten von dynamischen Bibliotheken und Dateien.

### 5.3.4 Die graphische Benutzeroberfläche

Die Bedienung eines Grid-Systems unterscheidet sich erheblich von dem eines normalen Rechnersystem. Das Fehlen eines verteilten Dateisystems und die

Notwendigkeit, für jede Art von Ressource eine weltweit eindeutige Identifizierung durchzuführen, die lange Kennzeichnungen nach sich ziehen (siehe Kapitel 4.3.2), stellt für den normalen Benutzer eine hohe Hemmschwelle dar. Zur leichteren Bedienbarkeit wurde daher im Rahmen dieser Arbeit eine graphische Benutzerschnittstelle erstellt, die im Folgenden beschrieben wird. Diese graphische Schnittstelle ist modular zu den bisher besprochenen Programmteilen, das heißt, sie kann zur Bedienung benutzt werden. Die anderen Programmteile benötigen aber die graphische Oberfläche nicht für ihre Funktionalität.

Abbildung 5.5 zeigt den Hauptbildschirm. In der oberen Hälfte werden die verfügbaren Rechenressourcen angezeigt. Darunter ist die Liste der aktiven Rechenaufträge zu sehen.

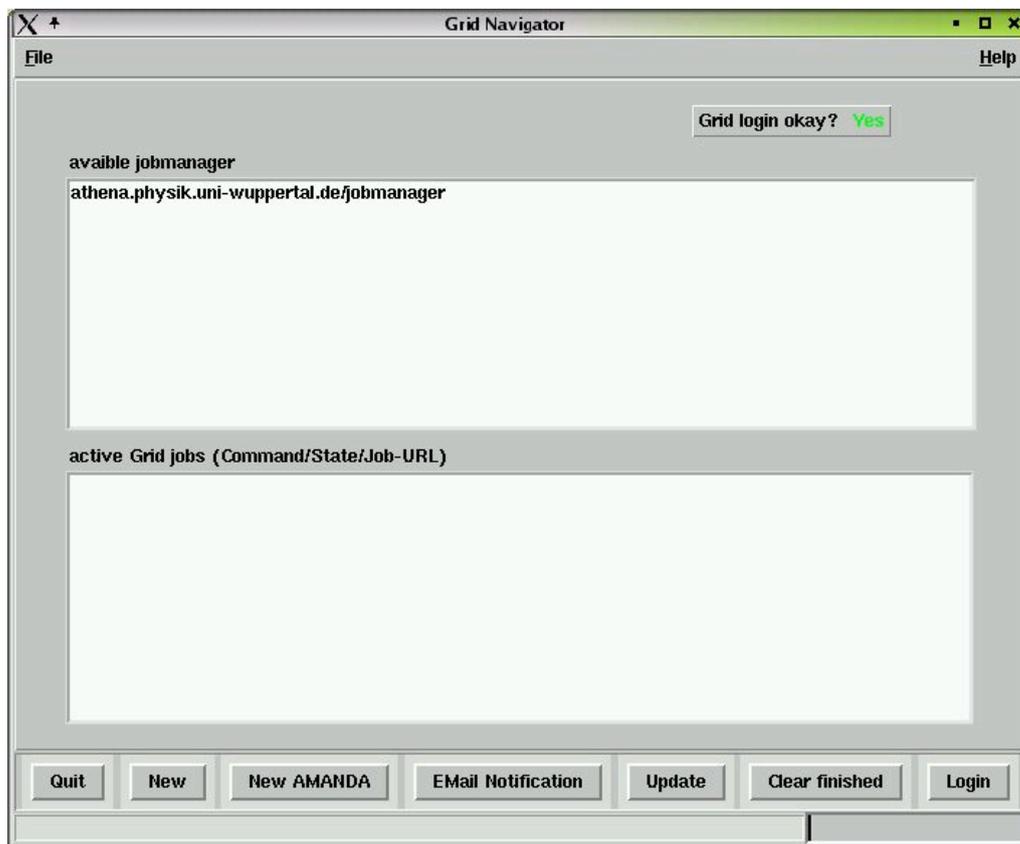


Abbildung 5.5: Der Hauptbildschirm der erstellten graphischen Benutzeroberfläche ohne aktive Rechenaufträge.

Die weiteren Schaltflächen haben folgende Bedeutung (von links nach rechts):

- **Quit**  
Beenden der Applikation,
- **New**  
starten neuer benutzerdefinierter Rechenaufträge, beziehungsweise
- **New AMANDA**  
vordefinierte AMANDA-Rechenaufträge,
- **E-Mail Notification**  
ändern der Benachrichtigungsoptionen,
- **Update**  
überprüfen des Status aller Rechenaufträge,
- **Clear finished**  
beendete Rechenaufträge aus der Liste entfernen und
- **Login**  
sich gegenüber dem Grid authentifizieren.

Die einzelnen Rechenaufträge starten nun eigene Dialogfelder, die sich gerade im Fall der vordefinierten AMANDA-Software im Normalfall auf die Spezifikation der Parameter beschränken. Allerdings ist auch der Fall vorgesehen, dass der Benutzer seine eigene Konfigurationsdatei vorgeben möchte. Programme, die als Eingabe die Ausgabe anderer Programmpakete erwarten – Kapitel 2.4 gibt einen Überblick über die AMANDA-Softwarepakete – sind entsprechend vorkonfiguriert. Abbildung 5.6 zeigt beispielhaft die Eingabemaske für das an AMANDA angepasste CORSIKA-Paket. Man erkennt, dass man für eine Simulation mit Standardparametern lediglich eine Ordnungsnummer und die gewünschte Anzahl der Ereignisse einstellen muss.

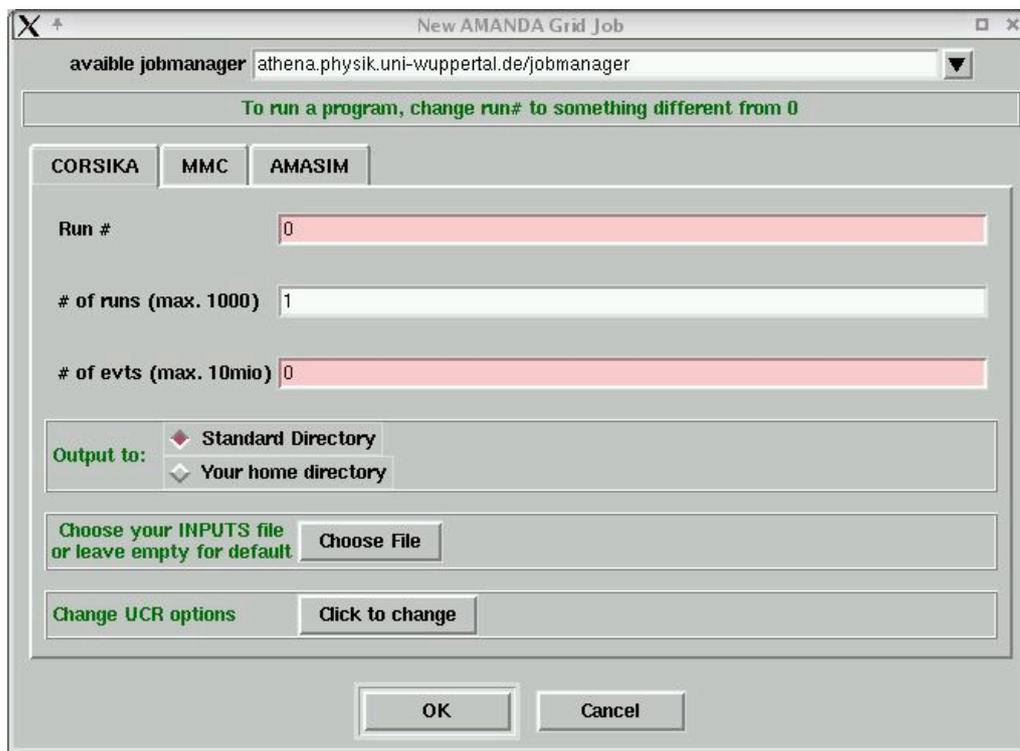


Abbildung 5.6: Eingabemaske für das CORSIKA-Programmpaket.

### 5.3.5 Zusammenfassung

Im Gegensatz zu dem in Abbildung 5.1 gezeigten Ansatz, bilden die Grundfunktionen eines Grids, wie sie unter anderem das Globus-Toolkit zur Verfügung stellen, bereits die Grundlage für ein fortgeschrittenes, verteiltes Rechnersystem. Abbildung 5.7 zeigt den hier verfolgten und realisierten Ansatz auf. Die roten Linien markieren dabei mögliche Grenzen von teilnehmenden Partnern (hier allgemein als „Institute“ bezeichnet).

Das so erstellte System wurde im Rahmen dieser Arbeit an folgenden Instituten installiert:

- Bergische Universität Wuppertal - Fachgruppe Physik
- Bergische Universität Wuppertal - Rechenzentrum
- Rheinisch-Westfälische Technische Hochschule Aachen - Fachgruppe Physik
- Deutsches Elektronen Synchrotron (DESY) - Standort Zeuthen

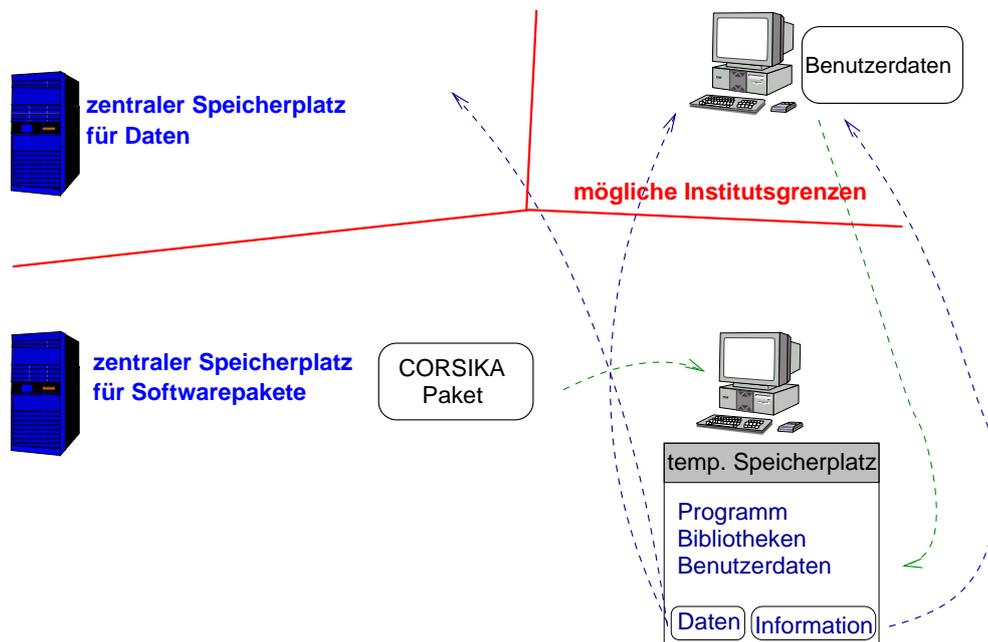


Abbildung 5.7: Hier realisierter Ansatz zum Aufbau eines verteilten Rechnersystems im Grid.

Obwohl nur als Machbarkeitsstudie gedacht, wurde mit der im Rahmen dieser Arbeit erstellten Lösung der Bedarf an Simulationsdaten für einige Diplomarbeiten gedeckt, so unter anderem [14, 65]. Damit wurde zum ersten Mal die Simulationssoftware für ein laufendes Experiment in einem allgemeinen Grid-System integriert.

Die entwickelte graphische Benutzeroberfläche in Zusammenhang mit einem zentral gewarteten Softwarearchiv erwies sich als sehr hilfreich gerade für Diplomanden, um ohne große Einarbeitungszeit den Bedarf an Simulationsdaten zu decken. Auch die Überwachung der Rechenanforderungen wird erheblich erleichtert. Da der einzelne Benutzer keinen direkten Zugriff auf den ausführenden Rechenknoten hat, wird ihm hier eine Übersicht über seine Rechenanforderungen gegeben und Probleme lassen sich leicht identifizieren.

Das zu Grunde liegende Globus-System erwies sich als stabil und verarbeitete auch eine große Anzahl von Rechenaufträgen. Als nachteilig erwies sich, dass es in einem aus dem Globus-Toolkit v2 aufgebautem Grid keine Verteilungsdienste gibt (vergleiche die Funktion des „Resource Brokers“ im EDG, wie sie in Kapitel 4.3.2 beschrieben wird). Damit fällt es dem Benutzer zu, die Auswahl eines Rechnersystems zu treffen. In moderneren Grid-Architekturen wie dem EDG ist dies nicht mehr notwendig.

Da die in der Hochenergiephysik eingesetzten Grid-Initiativen weiterhin das Globus Toolkit als Grundlage haben, ist die hier entwickelte Lösung ohne weitere Anpassung auch in diesen Umgebungen einsetzbar.

Mit Hilfe der hier vorgestellten Lösung wurden auch einige Ereignisse simuliert, die die Grundlage für die nachfolgende Studie bilden.

## 5.4 Flüsse von atmosphärischen Leptonen in AMANDA

In [66] ist eine Messung der atmosphärischen Myonflusses mit Hilfe des Luftschauersimulationsprogrammes CORSIKA vorgestellt. Diese Messung dient dem Verständnis der verschiedenen primären Wechselwirkungsmodelle, die in CORSIKA implementiert sind [67, 68]. Der gemessene Fluss kann als Kalibration für Experimente mit großen Untergrund-Detektoren dienen.

Dazu wurden für vier primäre Wechselwirkungsmodelle (QGSJET, SIBYLL, VENUS und HDPM) eine große Anzahl von hochenergetischen Myonen und Neutrinos simuliert.

### 5.4.1 Hintergrund

Für die Suche nach hochenergetischen, extraterrestrischen Neutrinos werden große Nachweisvolumen von typischerweise  $1 \text{ km}^3$  benötigt. Um die Cherenkov-Photonen nachweisen zu können werden dabei transparente Medien wie Wasser oder Eis benutzt (vergleiche auch Kapitel 2.2 und 2.3 für den Fall des AMANDA-Detektors). Für ein Experiment, welches 1 500 - 2 000 m unter dem Untergrund aufgebaut ist, ist der Hintergrund von atmosphärischen Myonen (sogenannte „downgoing muons“) ungefähr  $10^4$  stärker als das Myonsignal welches von Neutrinos ausgelöst wird, die die Erde passiert haben (sogenannte „upgoing muons“). Der erwartete Myonfluss aus extraterrestrischen Neutrinos ist sogar einen Faktor  $10^6$  kleiner. Daher benötigt man eine gute Unterdrückung von „downgoing muons“.

### 5.4.2 Simulation

Es wurden insgesamt knapp 2,5 Milliarden Luftschauer mit CORSIKA v6.201 in einer Rechenzeit<sup>1)</sup> von etwas über 329 Tagen generiert. Die Anzahl der gene-

---

<sup>1)</sup>Die Rechenzeit wurde normiert auf einen AMD Opteron Prozessor mit 1,8GHz Taktfrequenz.

<b>Wechselwirkungsmodell</b>	<b>CPU (sek.)</b>	<b>Anzahl Luftschauer [in Mio.]</b>
QGSJET	627.368	612
SIBYLL	956.449	887
VENUS	26.548.064	842
HDPM	297.308	141
<b>Summe</b>	<b>28.429.189</b>	<b>2.482</b>

Tabelle 5.1: Übersicht über die generierten Luftschauer.

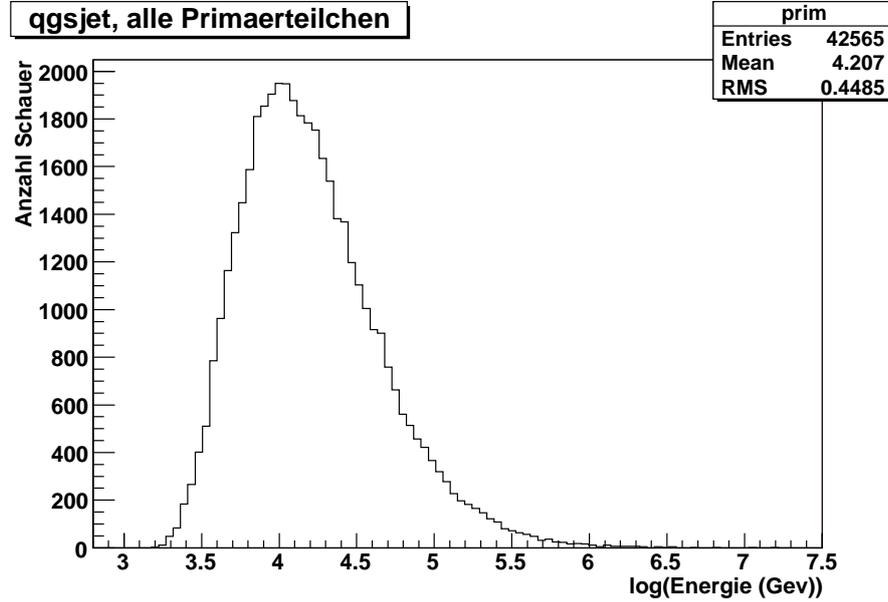
rierten Luftschauer je Wechselwirkungsmodell sowie die verwendete Rechenzeit sind in der Tabelle 5.1 angegeben.

Die Parameter für CORSIKA sind auf AMANDA und dessen Standort am geographischen Südpol angepasst und in der Tabelle 5.2 wiedergegeben.

Parameter	Wert	Beschreibung
PRMPAR	14	particle type of prim. particle
ESLOPE	-2.7	slope of primary energy spectrum
ERANGE	1200. 1.E11	energy range of primary particle
THETAP	0. 89.999	range of zenith angle (degree)
PHIP	0. 360.	range of azimuth angle (degree)
OBSLEV	2834.E2	observation level (in cm)
ELMFLG	T F	em. interaction flags (NKG,EGS)
RADNKG	2.E5	outer radius for NKG lat.dens.determ.
ARRANG	0.	rotation of array to north
FIXHEI	0. 0	first interaction height & target
FIXCHI	0.	starting altitude (g/cm**2)
MAGNET	16.4 -53.4	magnetic field south pole
HADFLG	0 1 0 1 0 2	flags hadr.interact. & fragmentation
ECUTS	273. 273. .003 .003	energy cuts for particles
MUADDI	T	additional info for muons
MUMULT	T	muon multiple scattering angle
LONGI	F 20. F F	longit.distr. & step size & fit
MAXPRT	0	max. number of printed events
ECTMAP	100	cut on gamma factor for printout
STEPFC	1.0	mult. scattering step length fact.
ATMOD	13	october atmosphere
DETCFG	1.	detector information (l/d)
LOCUT	T 1.58	enables skew angle cutoff
RANPRI	2	random primary (0: no; 1: ws; 2: jh)
SPRIC	T	separate primary energy cutoffs
FSEED	T	enable random generator seed recovery
DSLOPE	0.	slope correction
SCURV	T 6.4E8 1.73E5	curved surf., radius of Earth, depth

Tabelle 5.2: Parameter der CORSIKA-Simulation. Hinzu kommt das jeweils gewählte primäre Wechselwirkungsmodell. Erläuterungen finden sich in [68].

Abbildung 5.8 zeigt die Anzahl von Schauern als Funktion der Energie der Primärteilchen, die Myonen mit einer Energie über 1,2 TeV erzeugen. Man erkennt den steilen Anstieg im Energiespektrum. Das bedeutet, dass schon ein kleiner Anteil von niederenergetischen Primärteilen eine signifikante Veränderung des Flusses der Sekundärteilchen verursachen würde. Daher wurde für Primär- und Sekundärteilchen ein identischer Schnitt auf 1,2 TeV angewendet.

Abbildung 5.8: Myonantwort für alle Primärteilchen mit  $E_\mu > 1,2 \text{ TeV}$ 

### 5.4.3 Messung der Spektren von Myonen, Myon- und Elektron-neutrinos

Das generierte Myonspektrum wurde mit

$$\frac{dN}{dE} \propto E^{-\gamma} \left( \frac{1}{1 + \frac{1,2 \cdot E_\mu \cdot \cos \theta}{115 \text{ GeV}}} + \frac{0,054}{1 + \frac{1,1 \cdot E_\mu \cdot \cos \theta}{850 \text{ GeV}}} \right) \quad (5.1)$$

gefittet [69], wobei der spektrale Index  $\gamma$  als freier Parameter angenommen wurde. Da die Gleichung 5.1 die Energie des Myons am Produktionsort beschreibt, wurden zwei Korrekturen auf die Myondaten an der Erdoberfläche angewandt.

Zum einen wurde dem Energieverlust zwischen dem Produktionsort und dem Beobachtungspunkt abgeschätzt mit

$$\frac{dE}{dx} = - \left( 0,25 + 3,5 \cdot 10^{-4} E \right) \quad (5.2)$$

mit: x in Meter Wasseräquivalent, E in GeV.

Zum anderen wurde jedes Myon mit

$$P^{-1} = \exp\left(\frac{l}{c\tau} \cdot \frac{M}{E}\right) \quad (5.3)$$

mit  $P$ : Zerfallswahrscheinlichkeit für einen Zerfall;

$\tau$ : die Lebensdauer,

$M$ : die Masse,

$E$ : die Energie der Myons;

$c$ : Lichtgeschwindigkeit;

$l$ : Flugdauer

gewichtet, um den Zerfall zu berücksichtigen.

Abbildung 5.9 zeigt die Fits für die vier betrachteten Wechselwirkungsmodelle. Man erkennt, dass die angegebenen Fitfunktionen die Daten mit dieser hohen Statistik scheinbar nicht exakt beschreiben. Auch die hohen Werte von  $\chi^2/Ndf$  drücken dies aus.

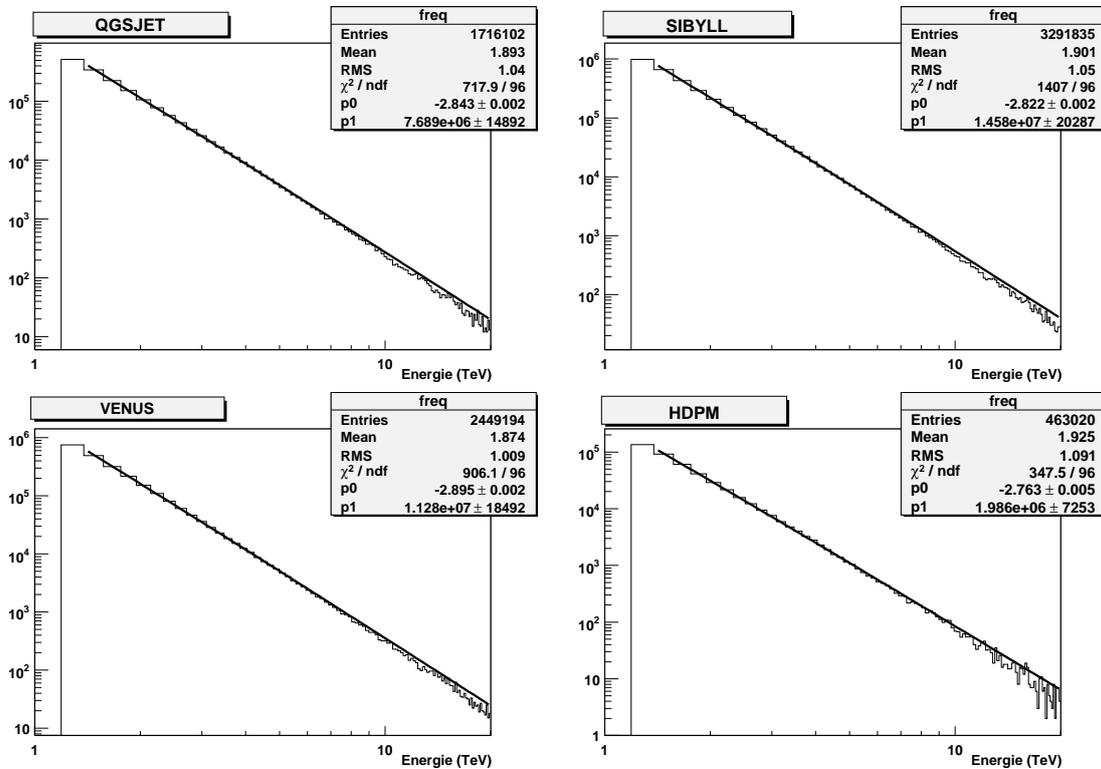


Abbildung 5.9: Fit der Myon-Spektren für vier Wechselwirkungsmodelle.

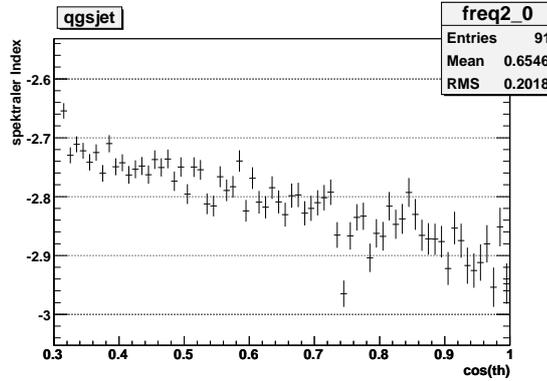


Abbildung 5.10: Fit des Myon-Spektrums in Abhängigkeit von  $\cos(\theta)$  beispielhaft für das QGSJET-Wechselwirkungsmodell.

Abbildung 5.10 zeigt den Fit des spektralen Indexes in Abhängigkeit von  $\cos(\theta)$  beispielhaft bei dem QGSJET-Wechselwirkungsmodell. Man erkennt einen deutlichen Abfall zu höheren  $\cos(\theta)$ . Daher liegt die Vermutung nahe, dass die in der Formel 5.1 enthaltenen Konstanten oder die angenommenen Energieverluste nach Formel 5.2 und 5.3 eine  $\cos(\theta)$ -Abhängigkeit besitzen. Untersuchungen dazu finden sich in [70]. Um aber einen Vergleich mit den Ergebnissen aus [66] zu gewährleisten, werden im Folgenden die Werte aus dem Fit nach Abbildung 5.9 verwendet.

Die Neutrino-Spektren wurden gefittet mit [71]:

- Myon-Neutrinos:

$$\frac{dN}{dE} \propto E^{-\gamma} \left( \frac{1}{1 + \frac{6 \cdot E}{E_{\pi}(\theta)}} + \frac{0,213}{1 + \frac{144 \cdot E}{E_{K^{\pm}}(\theta)}} \right) \quad (5.4)$$

wobei analog zu [66]  $E_{\pi}(\theta)$  mit 202 GeV (der Wert für den mittleren Winkel von  $51^{\circ}$ ) und  $E_{K^{\pm}}(\theta)$  mit 1500 GeV angenommen wurde.

- Elektron-Neutrinos:

$$\frac{dN}{dE} \propto E^{-\gamma} \left( \frac{0,05}{1 + \frac{15 \cdot E}{E_{K^{\pm}}(\theta)}} + \frac{0,185}{1 + \frac{15 \cdot E}{E_{K_2^0}(\theta)}} + \frac{11,4 \cdot E^{\zeta(\theta)}}{1 + \frac{121 \cdot E}{E_{\pi}(\theta)}} \right) \quad (5.5)$$

wobei  $E_{\pi}(\theta)$  und  $E_{K^{\pm}}(\theta)$  wie zuvor betrachtet werden,  $\zeta = a + b \lg E$  mit  $a = -0,355$  und  $b = -0,23$  und  $E_{K_2^0}(\theta) = 324$  GeV.

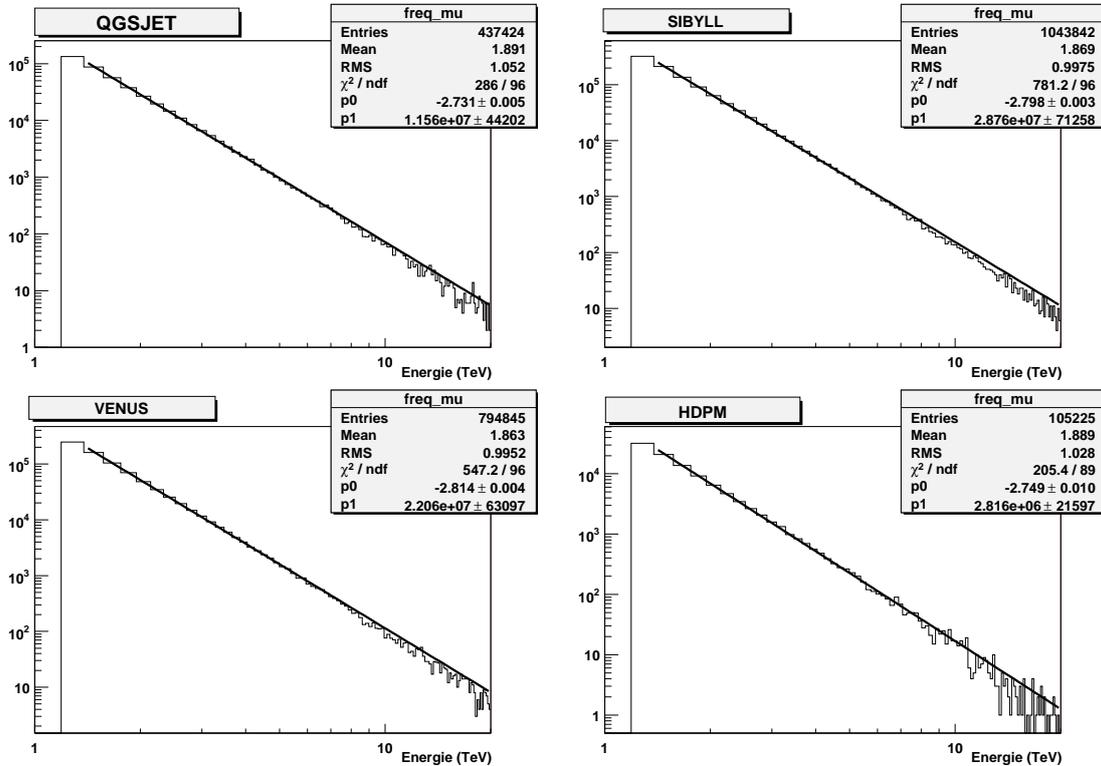


Abbildung 5.11: Fit der Myonneutrino-Spektren für vier Wechselwirkungsmodelle.

Abbildungen 5.11 und 5.12 zeigen die Fits nach Formeln 5.4 und 5.5 an die Simulationsdaten.

Abbildung 5.13 zeigt einen Vergleich der Muonmultiplizitäten für die verschiedenen Wechselwirkungsmodelle. Der rechte Plot ist dabei normiert. Im Vergleich zu den Ergebnissen aus [66] zeigt sich, dass die Abhängigkeit der Muonmultiplizitäten vom jeweiligen Wechselwirkungsmodell abgenommen hat, aber immer noch erkennbar sind. Vor allem die absoluten Häufigkeiten unterscheiden sich deutlich.

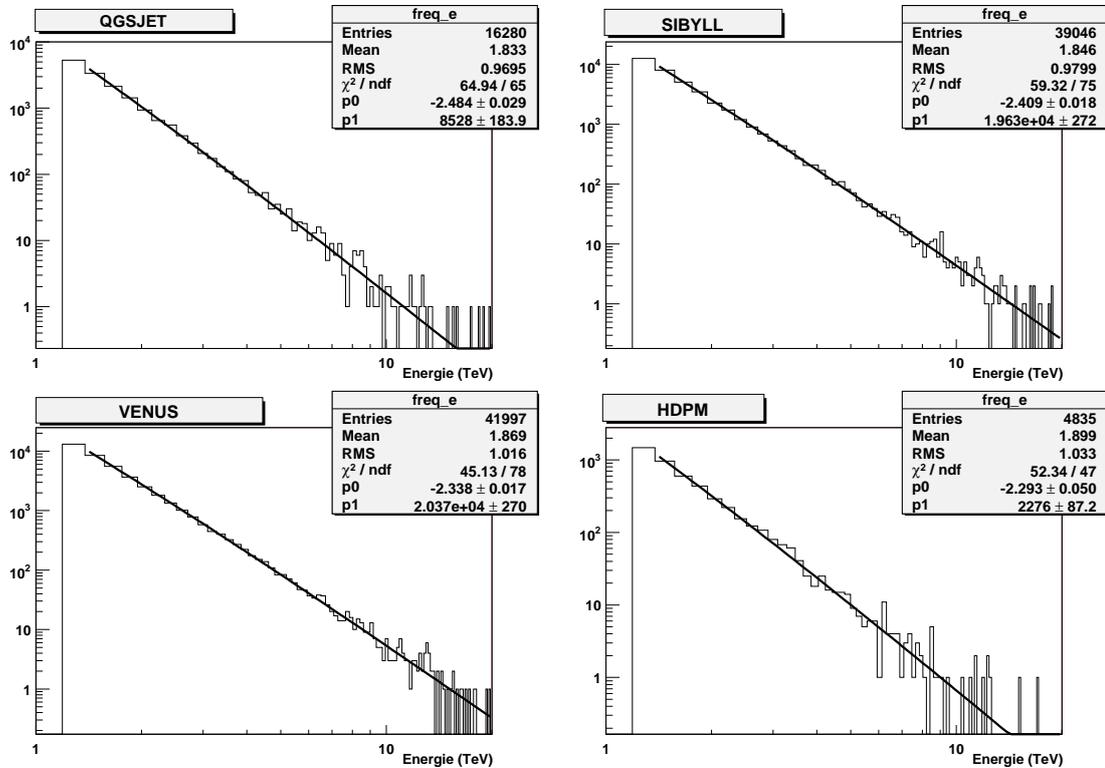


Abbildung 5.12: Fit der Elektronneutrino-Spektren für vier Wechselwirkungsmodelle.

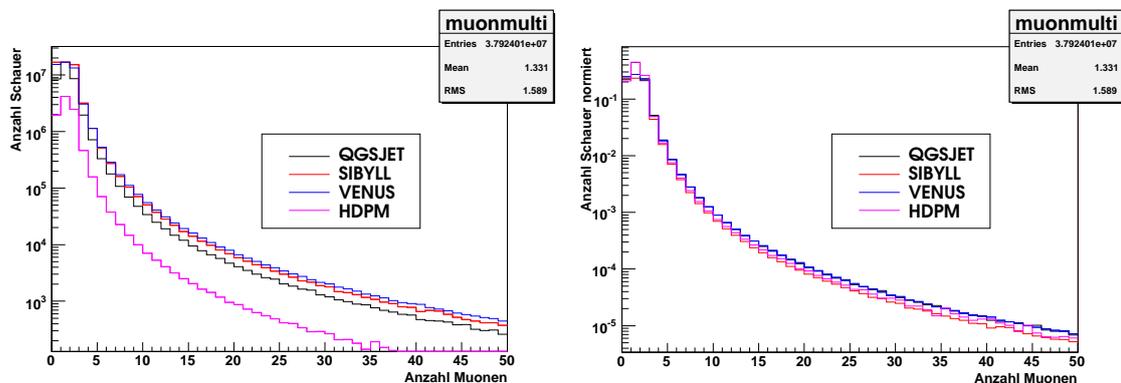


Abbildung 5.13: Vergleich der Myonmultiplizität in Abhängigkeit vom Wechselwirkungsmodell.

### 5.4.4 Zusammenfassung

Tabelle 5.3 vergleicht zwischen den hier untersuchten vier Wechselwirkungsmodellen QGSJET, SIBYLL, VENUS und HDPM die Messwerte für die spektralen Indizes von Myonen, Myon- und Elektronneutrinos sowie den prozentualen Anteil von Myonen sowie Myon- und Elektronneutrinos pro Primärteilchen.

Im Vergleich zur Messung in [66] zeigen die Modelle höhere spektrale Indizes für Myonneutrinos, während die Indizes für Elektronneutrinos gesunken sind. Das HDPM-Modell zeichnet sich auch hier wieder durch den kleinsten spektralen Index für Elektronneutrinos aus, während QGSJET den kleinsten Index für Myonneutrinos ergibt. Dieser Wert sowie der des HDPM-Modells stimmen mit der Messung des Frejus-Experiments ( $2,66 \pm 0,05(stat.) \pm 0,03(syst)$ ) [72] überein. Die spektralen Indizes für Myonen sind insgesamt höher und streuen etwas stärker, stimmen aber – bis auf das VENUS-Modell – im Rahmen der Fehler mit den Messungen der Experimente Frejus ( $2,78 \pm 0,07$ ), LVD ( $2,78 \pm 0,05$ ), und MACRO ( $2,76 \pm 0,05$ ) überein. Die Vergleich zwischen den Wechselwirkungsmodellen der prozentualen Anteile der Leptonen pro Primärteilchen stimmen tendenziell mit den vorherigen Messungen überein. Insgesamt ist der Leptonanteil gestiegen.

Durch die deutlich höhere Statistik konnten die statistischen Fehler entsprechend verkleinert werden.

Wechselwirkungsmodell	QGSJET	SIBYLL	VENUS	HDPM
$N_{\mu}/N_{pri}$ in %	$0,2943 \pm 0,0002$	$0,4011 \pm 0,0002$	$0,2950 \pm 0,0002$	$0,3572 \pm 0,0005$
$N_{\nu(e)}/N_{pri}$ in %	$0,00279 \pm 0,00002$	$0,00476 \pm 0,00002$	$0,00506 \pm 0,0002$	$0,00373 \pm 0,00005$
$N_{\nu(\mu)}/N_{pri}$ in %	$0,0750 \pm 0,0001$	$0,1272 \pm 0,0001$	$0,0957 \pm 0,0001$	$0,0812 \pm 0,0002$
$\gamma_{\mu}$	$2,843 \pm 0,002$	$2,822 \pm 0,002$	$2,895 \pm 0,002$	$2,763 \pm 0,005$
$\gamma_{\nu(e)}$	$2,484 \pm 0,029$	$2,409 \pm 0,018$	$2,338 \pm 0,017$	$2,293 \pm 0,050$
$\gamma_{\nu(\mu)}$	$2,731 \pm 0,005$	$2,798 \pm 0,003$	$2,814 \pm 0,004$	$2,749 \pm 0,010$

Tabelle 5.3: Der prozentuale Anteil der Myonen, Elektron- und Myonneutrinos pro Primärteilchen; der spektrale Index des Myonen-, Elektronneutrino- und Myonneutrinospektrums.

# Kapitel 6

## Integration des DØ-Experiments in das European Data Grid

### 6.1 Einleitung

Im September 2003 hat das DØ-Experiment eine Initiative zum Wiederaufbereiten der bisher genommenen Daten gestartet. Dabei sollten neue Erkenntnisse bei der Beschreibung des Detektors und die zwischenzeitlich verbesserte Rekonstruktionssoftware zum Einsatz kommen. Insgesamt wurden während dieser Initiative 519.212.822 Ereignisse neu berechnet [73], davon 97.619.114 außerhalb des Fermilabs. Ein Teil dieser Ereignisse wurde dabei im Rahmen dieser Arbeit innerhalb des European Data Grid verarbeitet.

Der hier verfolgte Ansatz ähnelt dem für AMANDA, welcher im vorherigen Kapitel 5 vorgestellt wurde. Auch hier wurde davon abgesehen, die bestehende experimentspezifische Software zu verändern. Es wurden entsprechende Umgebungen geschaffen, in der alle für die DØ-Software notwendigen dynamischen Bibliotheken und Dateien vorhanden sind. Eine Besonderheit nimmt dabei das Datenmanagement-System von DØ mit dem Namen „SAM“<sup>1)</sup> ein [74]. In diesem System sind die Daten des DØ-Experimentes zusammen mit Informationen (Metadaten) über die einzelnen Datensätze gespeichert. Das hauptsächliche Konzept des SAM-Systems besteht daraus, alle Daten über Metadaten zu beschreiben und damit zugänglich zu machen und zum anderen den Zugriff auf diese Daten zu optimieren, da die Reihenfolge bei Daten eines physikalischen Experiments keine Rolle spielt.

Da jeglicher Zugriff auf die DØ-Daten über das SAM-System erfolgen muss, war die Integration von SAM in die bestehende Infrastruktur des European Data Grid Hauptziel der vorliegenden Arbeit.

---

<sup>1)</sup>SAM = Sequential data Access via Metadata

Der generelle Prozessablauf einer Rechenanforderung mit diesen Rahmenbedingungen wurde wie folgt implementiert:

1. Von der zentralen DØ-Verwaltung werden einzelne Datensätze zur Wiederaufbereitung zugewiesen. Jeder dieser Datensätze enthält eine Liste von ungefähr siebzig Dateien, die jeweils eine Größe von ungefähr 700 Megabyte haben. Die Rechenzeit, die für jede dieser Dateien benötigt wird, beträgt ungefähr zwölf Stunden (auf einem Pentium Rechner mit einem Gigahertz Taktfrequenz).
2. Für die zugewiesenen Datensätze werden nun aus dem Datenmanagement-System SAM alle zugehörigen Dateien abgefragt und auf einem Rechner innerhalb des EDG-Systems gespeichert.
3. Die Rechenanforderungen werden nun in das EDG-System übergeben und werden dort verarbeitet.
4. Die wiederaufbereiteten Daten werden aus dem EDG-System entnommen und zurück in das DØ-Metadatensystem SAM zurückgespielt.

Für die Integration von SAM in das EDG und die Anpassung der DØ-Software waren einige Entwicklungsschritte nötig, die im Folgenden beschrieben werden. Durch diesen Praxis-Test des EDG wurden etliche Schwachstellen entdeckt und konnten beseitigt werden. Die Erfahrungen werden auch in den folgenden Unterkapiteln vorgestellt.

## 6.2 Aufbau einer EDG-Infrastruktur

Wie in Kapitel 4.3.2 beschrieben, ist zur Teilnahme am European Data Grid eine Installation von mehreren Rechnern mit der spezifischen Software nötig.

Daher wurde an der Bergischen Universität ein Cluster von zwölf Rechnern unter Zuhilfenahme eines Clusterverwaltungssystems namens „Local ConFiGuration system“ (kurz LCFG) [75] mit verschiedenen Diensten installiert:

- Ein sogenannter „LCFGng Server“, der die Konfigurationsdaten aller nachfolgend genannten Dienste speichert,
- ein „User Interface“ zusammen mit einer sogenannten „Monitor Box“ für die R-GMA-Datenbank (vergleiche Kapitel 4.3.2),
- ein „Storage Element“,
- ein „Computing Element“,

- ein „Resource Broker“,
- ein Server für die sogenannten „Data Management Services“ (vergleiche Kapitel 4.3.2),
- sechs „Worker Nodes“ als eigentliche Rechenknoten.

Diese Rechner wurden im EDG zertifiziert und konnten somit von weltweit allen Benutzern im EDG genutzt werden. Da das European Data Grid mittlerweile beendet ist, ist eine vergleichbare Auswahl an Diensten nunmehr im Nachfolgeprojekt LCG<sup>2)</sup> etabliert worden und wird fortlaufend genutzt.

Angeschlossen wurden die Rechner – vor allem das „Storage Element“ – an die im Institut verfügbare Infrastruktur, insbesondere wurde die vorhandene Speicherkapazität auf zentralen Servern mitbenutzt. So sind im Moment 1,4 TB an Festplattenspeicher für Anwendungen im LCG verfügbar gemacht.

Der Server für die „Data Management Services“ (oft auch vereinfacht als „RLS-Server“ bezeichnet) ist der weltweit einzige für die Benutzer innerhalb von DØ. Ihm kommt daher eine zentrale Bedeutung im Grid zu, da ohne diesen keine Dateien im Grid lokalisiert werden können, die Benutzern der virtuellen Organisation dzero zugeordnet sind.

Für R-GMA wird eine sogenannte „Archiver Box“ betrieben. Dies ist Teil der weltweit verteilten Datenbank im R-GMA und entlastet damit signifikant die zentrale Datenbank, die am Rutherford Appleton Laboratory betrieben wird. Auch diesem Dienst kommt eine zentrale Bedeutung zu, da es im LCG nur wenige dieser Dienste gibt (Anfang 2005 wurde an drei Instituten weltweit dieser Dienst betrieben).

## 6.3 Anpassung der DØ-Software

### 6.3.1 DØ-spezifische Infrastruktur

Wie eingangs erwähnt, ist die Integration des DØ-eigenen Datenmanagementsystems SAM eines der Ziele der vorliegenden Arbeit.

Folgende Ziele sollten dadurch erreicht werden:

1. Die zu bearbeitenden Daten sollen dem EDG-Datenverwaltungssystem [76] zugänglich gemacht werden.
2. Die DØ-Software soll auf den EDG-Rechenknoten („Worker Nodes“) nutzbar sein.

---

<sup>2)</sup>LCG = LHC Computing Grid

3. Der Ablauf der Prozesse auf dem EDG sollen überwacht und die Ergebnisse der Überwachung sollen gespeichert werden.
4. Die Ergebnisse der Datenwiederaufbereitung sollen zurück in das SAM-System geschrieben werden.

Schritt eins und vier erfordert einen Austauschkanal zwischen den Datenmanagementsystemen von EDG und DØ. Dieses wurde dadurch erreicht, dass ein Speicherplatzbereich auf einem nachgelagerten Server eingerichtet wurde, auf den sowohl von einem Rechner mit einem installierten SAM-System (im Folgenden immer als „SAM-Station“ bezeichnet) und von einem „Storage Element“ und einem „User Interface“ von EDG aus zugegriffen werden kann. Abbildung 6.1 illustriert dieses.

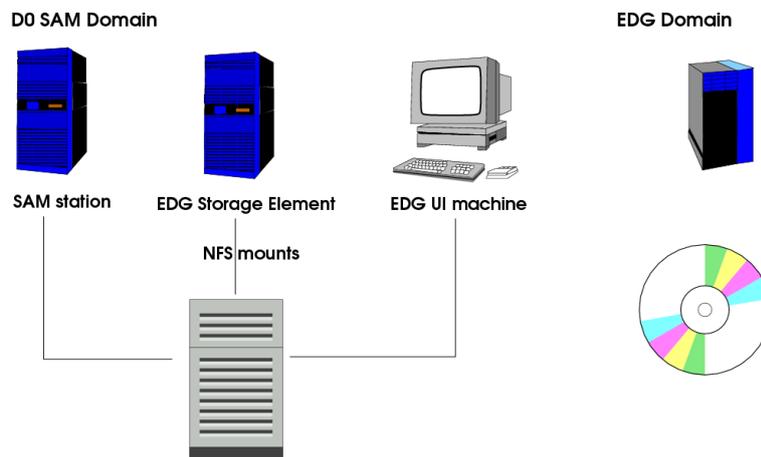


Abbildung 6.1: Die Realisierung des Datenaustausches zwischen den Datenmanagementsystemen von DØ und EDG.

Dabei bleiben die beiden Systeme so weit wie möglich getrennt. Von Nachteil ist allerdings, dass die so gefundene Lösung manuelles Eingreifen bei jedem Datenaustausch erfordert. Sowohl SAM als auch die EDG-Software erwarten beide ein lokales Dateisystem, welches nicht durch andere Softwareprodukte verändert wird. Dieses hat es erforderlich gemacht, auch das EDG „User Interface“ zu integrieren, worüber dann die Dateien im EDG bekannt gemacht werden.

### 6.3.2 Anpassung der DØ-Software

Die Software, die zur Wiederaufbereitung von Daten verfügbar sein muss, besteht im Wesentlichen aus vier Teilen:

- Die DØ-Software an sich, wie sie von der Kollaboration erstellt wird. Diese besteht aus Programmen, dynamischen Bibliotheken und einer umgebenden Struktur, die unter anderem auch die Dateien zur Konfiguration enthält.
- Das sogenannte „mc\_runjob“ Paket, ein Programmpaket, welches für die Ablaufsteuerung zuständig ist.
- Das „pyxml“ Programmpaket, wie es unter [77] erhältlich ist. Dieses Paket wird von „mc\_runjob“ benötigt. Obwohl es für die Durchführung der Datenwiederaufarbeitung im Rahmen dieser Initiative nicht gebraucht wurde, hätte die Entfernung der Abhängigkeiten aus dem „mc\_runjob“ Paket einen erheblichen Mehraufwand bedeutet.
- Die Programmiersprache „Python“ in der Version 2.1. Diese wird auch von „mc\_runjob“ benötigt. Das EDG-System liefert Python in den Versionen 1.5.2 und 2.2.2 aus, aber diese haben sich als inkompatibel mit den Anforderungen von „mc\_runjob“ erwiesen.

Die eigentliche DØ-Software musste nur geringfügig angepasst werden. Dabei wurden die Optionen des „mc\_runjob“-Paketes so angepasst, dass der Programmablauf auch außerhalb einer dedizierten DØ-Softwareumgebung lauffähig ist. Desweiteren mussten einige der mit der DØ-Software ausgelieferten dynamischen Bibliotheken gelöscht werden, die Konflikte mit denen auf dem EDG-System vorhandenen Bibliotheken verursacht hätte. Das betrifft vor allem die C-Bibliothek `libc.so`.

Obwohl EDG ein System zur Verbreitung experimentspezifischer Software über sogenannte RPM<sup>3)</sup>-Pakete [78] anbietet, wurde bei der vorliegenden Arbeit ein anderer Ansatz gewählt. Ausschlaggebend waren die häufigen (teilweise wöchentlichen) Änderungen an der DØ-Software, die gerade zu Beginn der Initiative zur Wiederaufbereitung notwendig waren. Diese Änderungen hätten in dem vom EDG vorgesehenen Mechanismus nicht kurzfristig verteilt werden können. Daher wurden die benötigten Software-Pakete manuell in Archive verpackt und auf einem „Storage Element“ abgelegt und im „Replica Location Service“ (RLS - vergleiche Kapitel 4.3.2) registriert. Diese Lösung hat zwar den Nachteil, dass für jeden einzelnen Prozess die Softwarepakete von einem „Storage Element“ auf den betreffenden ausführenden Rechner kopiert werden müssen, aber die Flexibilität auf kurzfristige Änderungen bleibt so erhalten.

Durch eine Beschränkung in der DØ-Software hat es sich zudem als nötig herausgestellt, den gesamten Programmablauf in einem Verzeichnis unterhalb der

---

<sup>3)</sup>RPM = RedHat Package Manager

/tmp-Struktur ablaufen zu lassen, anstatt in dem vom EDG-System vorgegebenen Verzeichnis. Der Grund liegt darin, dass an einigen Stellen in der DØ-Software die absolute Pfadlänge auf 100 Zeichen beschränkt ist. Hingegen können die Pfade innerhalb des EDG-Systems sehr lang werden, zumindest kann man nicht garantieren, dass die absolute Pfadlänge unter 100 Zeichen beträgt. So werden hier Pfade wie zum Beispiel /tmp/1423.d0reco angelegt, wodurch diese Beschränkung umgangen wird. Der Nachteil besteht allerdings darin, dass die automatischen Routinen des EDG, die nach Beendigung eines Rechenauftrags die übrig gebliebenen Dateien löschen, diese dort nicht finden. Ohne eine Veränderung an der DØ-Software ist dieses Problem nicht lösbar und wurde daher an die entsprechenden Stellen innerhalb der DØ-Kollaboration gemeldet.

### 6.3.3 Überwachung und Buchführung

In einem verteilten System ist die Überwachung und Buchführung – im englischen „Monitoring and Bookkeeping“ genannt – ein sehr wichtiger Bestandteil. Das EDG-System bietet allerdings lediglich im Bereich Überwachung an, den Zustand einer Rechenanforderung zu erfragen. Dieser Zustand wird recht allgemein mit den englischsprachigen Begriffen „running“, „waiting“, „scheduled“, „done“ und „aborted“ beschrieben<sup>4)</sup>. Abbildung 6.2 zeigt die möglichen Übergänge von einem dieser Zustände in andere. Im Bereich Buchführung bietet das EDG-System überhaupt keine Lösung an. Hier muss also, wie im Rahmen dieser Arbeit geschehen und nachfolgend beschrieben, der Benutzer eine Lösung entwickeln.

In der hier vorliegenden Arbeit wurde das R-GMA-System<sup>5)</sup> (siehe Kapitel 4.3.2) als Grundlage zur Entwicklung eines Überwachungs- und Buchführungsinfrastruktur verwendet. Es erlaubt innerhalb des EDG-Rechnerverbundes das Verbreiten von beliebigen Informationen. Diese Informationen können auch auf entfernten Maschinen innerhalb des Verbundes ausgelesen oder in eine Datenbank zur späteren Verwendung gespeichert werden.

R-GMA verwendet ein SQL-Modell<sup>6)</sup> zum Zugriff und zur Speicherung von Daten. Daher werden alle Informationen in Tabellen abgelegt. Es wurden vier solcher Tabellen erstellt:

#### 1. „**submission table**“

Diese Tabelle beinhaltet Informationen über die Übergabe der Rechenaufträge an den „Resource Broker“. Es werden folgende Informationen abgespeichert:

---

<sup>4)</sup>running=ausführend, waiting=wartend, scheduled=(hier:) zugewiesen, done=fertig, aborted=abgebrochen

<sup>5)</sup>R-GMA = Relational Grid Monitoring Architecture

<sup>6)</sup>SQL = Structured Query Language, deutsch: strukturierte Abfragesprache

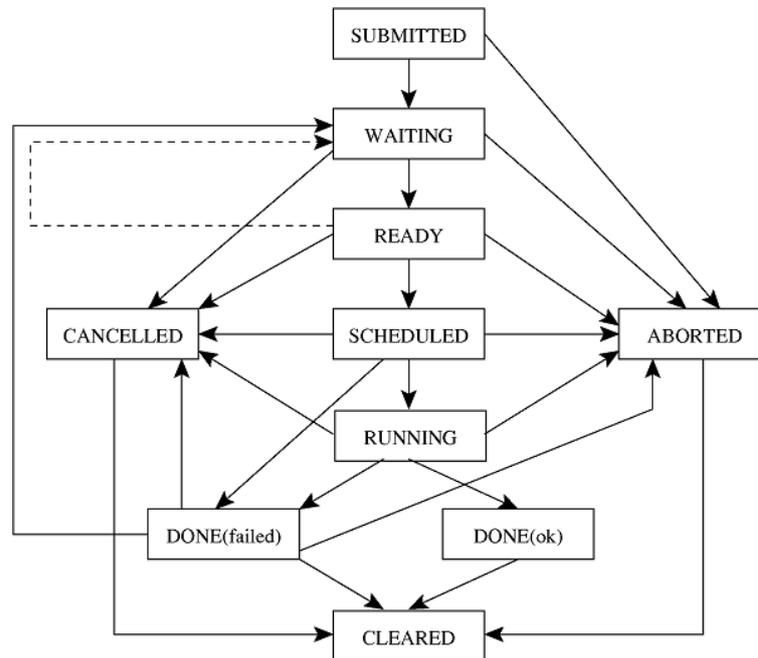


Abbildung 6.2: Mögliche Zustände von Rechenanforderungen im European Data Grid. Aus [79].

- Die Identifizierung des Rechenauftrages, wie sie vom EDG-System zurückgeliefert wird,
- die Uhrzeit und
- die Identifizierung des „User Interface“, von dem aus der Rechenauftrag abgeschickt wurde.

## 2. „job start table“

Hier werden Informationen zum Startzeitpunkt auf einer „Worker Node“ abgespeichert. Und zwar im Einzelnen:

- Ein eindeutiger Zufallszahlenwert, der den einzelnen Ablauf kennzeichnet<sup>7)</sup>,
- die Identifizierung des Rechenauftrages, wie sie vom EDG-System generiert wurde,
- die Uhrzeit,
- der Rechnernamen der „Worker Node“,

<sup>7)</sup>Dies ist notwendig, da bei einem Abbruch der Rechenanforderung das EDG-System den Ablauf noch einmal anstößt.

- der logische Dateiname (LFN, siehe Kapitel 4.3.2), über den die nötigen Dateien definiert wurden,
- die Identifizierung von SAM sowie
- einige Informationen über das ausführende System.

### 3. „job end table“

Diese Tabelle wird unmittelbar vor Beendigung des Rechenauftrags auf der „Worker Node“ beschrieben und enthält:

- Den in der „job start table“ vergebenen eindeutigen Zufallszahlenwert,
- die Uhrzeit,
- die Zeit, die die Rechenanforderung auf der Maschine verbraucht hat (oft als „wall time“ bezeichnet) und die Zeit, die der Prozessor für diese Anforderung gerechnet hat,
- die Anzahl der DØ-Ereignisse, die diese Rechenanforderung bearbeitet hat,
- den physikalischen Dateinamen der Ergebnisdateien inklusive des Rechenernamens des „Storage Elementes“, auf dem die Namen geschrieben wurden sowie die GUID (siehe Kapitel 4.3.2) dieser Datei. Mit Hilfe dieser GUID können die Dateien und eventuell vorhandene Kopien davon im EDG-System wiedergefunden werden.

### 4. „command table“

Die Tabelle dient der Fehlersuche. Kritische Teile werden durch ein umliegendes Programm auf mögliche Fehler untersucht. Bei normalem Ablauf wird nichts weiter unternommen. Bei einem fehlerhaften Programmablauf wird in diese Tabelle ein Eintrag geschrieben, der folgende Informationen beinhaltet:

- Den in der „job start table“ vergebenen eindeutigen Zufallszahlenwert,
- einen weiteren eindeutigen Zufallszahlenwert, der nur eine interne Funktion hat,
- die aktuelle Uhrzeit, bei der das betrachtete Programm gestartet wurde,
- den Programmnamen des fehlgeschlagenen Programms samt den Argumenten, mit denen es gestartet wurde,

- den sogenannten „Exit status“<sup>8)</sup> des Programms,
- die letzten 255 Zeichen der Programmausgabe.

Durch diese Tabelle wird es möglich, die Teile innerhalb des Programmablaufs zu identifizieren, die für den Abbruch der Rechenanforderung verantwortlich waren. Die Verbindung mit den Informationen aus den anderen Tabellen bietet die Möglichkeit, weitere Informationen wie zum Beispiel den Namen des betroffenen Rechners, auf dem die Rechenanforderung fehlgeschlagen ist, zu erhalten.

Zusammengenommen ergibt dieses System ein Werkzeug, um Fehler im EDG-System zu suchen und zu beseitigen. Vor allem bietet es die Möglichkeit zwischen sporadisch auftretenden Fehlern in der DØ-Software und Fehlern, die aufgrund falscher Konfiguration einer „Worker Node“ auftreten, zu unterscheiden.

#### 6.3.4 Ablaufsteuerung einer DØ-Rechenanforderung

Im Rahmen dieser Arbeit wurde ein Programm zur Ablaufsteuerung von DØ-Rechenanforderungen in der Programmiersprache Python entwickelt. Das Programm dient zur Einrichtung der Laufzeitumgebung; in ihm ist die oben vorgestellte Überwachung und Buchführungsroutine enthalten; es startet die DØ-Software und überwacht dessen Ergebnisse.

Im Einzelnen werden folgende Schritte durchlaufen (diejenigen Schritte, die unter der oben vorgestellten Überwachung stattfinden, sind mit einem Stern \* markiert):

1. Die Startzeit wird notiert.
2. Ein temporärer Speicherplatz wird angelegt, in dem später die DØ-Software abgelegt wird.
3. Die anforderungsspezifischen Dateien werden angefordert und ebenfalls in dem temporären Speicherplatz abgelegt.
4. Ein „job start table“-Objekt wird instantiiert, mit den gewonnenen Daten beschrieben und über R-GMA publiziert.
5. Für jedes der vier Softwarepakete, die benötigt werden, wird

---

<sup>8)</sup>Der „Exit status“ ist ein Zahlenwert, mit dem ein Programmator dem überliegenden Betriebssystem mitteilt, ob ein Programm erfolgreich abgelaufen ist oder nicht. Ein Wert von Null bedeutet erfolgreicher Ablauf, für die Werte ungleich Null gibt es keine standardisierte Bedeutung.

- (a) das Softwarepaket von der Datenverwaltung des EDG-Systems angefordert\* und
  - (b) im temporären Speicherplatz abgelegt\*.
6. Einige symbolische Links werden angelegt, die für den erfolgreichen Ablauf benötigt werden.
  7. Der m4 Preprozessor [80] wird gestartet\*, um eine Dateivorlage mit den anforderungsspezifischen Werten zu füllen. Diese Datei wird vom mc\_runjob Paket benötigt.
  8. mc\_runjob wird mit der entstehenden Datei gestartet\*. Dadurch wird die DØ-Rekonstruktionssoftware angestoßen.
  9. Die entstehenden Ausgabedateien mit den Ergebnissen der Berechnung werden archiviert\*.
  10. Dieses Archiv wird unter Zuhilfenahme des EDG-Datenverwaltungssystems abgelegt\*.
  11. Der physikalische Speicherort dieser Datei wird abgefragt\*.
  12. Die „job end table“ wird instantiiert, mit den gewonnenen Daten beschrieben und in R-GMA publiziert.
  13. Der temporäre Speicherbereich wird wieder freigegeben.
  14. Das Programm wird beendet.

## 6.4 Ergebnisdiskussion

Das European Data Grid hat bewiesen, dass es die notwendige Funktionalität für den Produktionsbetrieb eines laufenden Experimentes zur Verfügung stellt. Die Rechenanforderungen wurden von einer zentralen Stelle – im hier vorgestellten Fall vom NIKHEF<sup>9)</sup> aus – über die angeschlossenen Rechnersysteme weltweit verbreitet und verarbeitet. Das Datenverwaltungs- und das Informationssystem (R-GMA) erlaubte es, ein Steuerungs- und Überwachungssystem aufzubauen.

---

<sup>9)</sup>NIKHEF = Nationaal Instituut voor Kernfysica en Hoge-Energiefysica

### 6.4.1 Stabilitäts- und Funktionalitätsprüfungen

Im Rahmen der hier vorgestellten Arbeit konnten einige Komponenten des European Data Grids identifiziert werden, die Instabilitäten aufwiesen oder bei denen eine grundlegende Funktionalität nicht zur Verfügung stand. Diese werden nun im Einzelnen vorgestellt:

- **„Storage Element“**

Wie in Kapitel 4.3.2 erläutert, existieren zwei Implementationen für ein „Storage Element“ - das „WP5 Storage Element“ und das „Storage Element Classic“. Es hat sich durch die im Rahmen der DØ-Initiative durchgeführten Tests herausgestellt, dass das „WP5 Storage Element“, obwohl technisch deutlich überlegen, instabil ist und die meiste Zeit nicht funktioniert hat. Im Einzelnen traten folgende Probleme auf:

- Die Konfiguration ist so fehleranfällig, dass ein signifikanter Anteil der im EDG vorhandenen Systeme falsch konfiguriert war. Selbst einfache Kopieranweisungen schlugen dadurch fehl.
- Dateien konnten nicht mehr gelöscht werden. Dies hatte zur Folge, dass die DØ-Initiative, die einige Terabyte an Daten produziert hat, diese nur auf Systemen mit „Storage Element Classic“ ablegen konnte.
- Es findet keine Fehlerbehandlung statt. So sind viele Operationen als erfolgreich gemeldet worden, obwohl sie in Wirklichkeit nicht oder unvollständig stattgefunden haben.
- Das vorhandene virtuelle Dateisystem, welches die auf dem „Storage Element“ gespeicherten Daten abbilden soll, geriet immer wieder in einen Zustand, in dem es einen alten Datenbestand wiedergab. So waren angeblich Dateien vorhanden, die gar nicht mehr gespeichert waren und Dateien fehlten, die tatsächlich dort vorhanden waren.

- **Datenverwaltung**

Im Allgemeinen haben die Werkzeuge zur Datenverwaltung den Anforderungen entsprochen. Trotzdem haben sich einige Kritikpunkte ergeben:

- Es fehlt ein Listen-Befehl analog zu `ls`, um alle Dateien aufzulisten, die einer virtuellen Organisation oder zu einem Suchbegriff gehören. Stattdessen muss der Benutzer direkt mit den grundlegenden Diensten „Local Replica Catalog“ und „Replica Metadata Catalog“ interagieren.
- Das Kopieren von Dateien geschieht asynchron. Das heißt, wenn an einem Cluster eine größere Anzahl von Rechenanforderungen eingehen, die eine Datei benötigen, die nicht auf dem lokalen „Storage

Element“ gespeichert ist, werden so viele Dateikopien erzeugt, wie Rechenanforderungen zeitgleich abgearbeitet werden.

- Es musste eine veränderte Version der Dateiverwaltungsbefehle erstellt werden. Dies wurde nötig, weil die Befehle, die das EDG mitliefert, die Existenz einer Datei überprüfen, bevor sie übertragen wird. Dies ist im Allgemeinen eine gute Eigenschaft, aber es wurde für diese Überprüfung auf Funktionen des Globus-Toolkits (siehe Kapitel 4.3.1) zurückgegriffen, die nur dann funktionieren, wenn die jeweilige „Worker Node“ von außen per Internet direkt erreichbar ist. Viele – vor allem größere Institute wie zum Beispiel das Rutherford Appleton Laboratory in England und das NIKHEF in den Niederlanden – blockieren aber aus Sicherheitsgründen Netzwerkverkehr von außen. Dadurch werden diese Befehle unbrauchbar.

- **Informationssystem**

Das Informationssystem R-GMA hat sich seiner Funktionalität nach als geeignet erwiesen, im Produktionsbetrieb als Grundlage für ein Überwachungssystem zu dienen. Das Hauptproblem in dieser Komponente ist allerdings die anhaltende Instabilität. Während des Rechenbetriebs stürzte R-GMA häufig ab und damit war das gesamte EDG-System unbenutzbar. Dieses hat dazu geführt, dass R-GMA später durch ein anderes Informationssystem ausgetauscht wurde.

Weitere Kritikpunkte, die sich ergaben:

- Es wurden keine Versionskontrollen zwischen den verschiedenen Teilen der R-GMA Software durchgeführt. So konnte es passieren, dass ein Teilnehmer nur durch die Benutzung der Version 2.1 das gesamte EDG-System zum Absturz bringen konnte, als es noch mit Version 2.0 betrieben wurde. Solche Probleme hätten leicht vermieden werden können.
- Wie schon beim „Storage Element“ war auch hier die Konfiguration des Systems sehr anfällig. Auch bei diesem Dienst wurden Fehlkonfigurationen einzelner „Storage Elemente“ im EDG erst durch fehlgeschlagene Rechenanforderungen entdeckt.

- **Resource Broker**

Der „Resource Broker“ ist eine sehr zentrale Komponente des EDG (vergleiche Kapitel 4.3.2). Jedes „User Interface“ ist durch die Konfiguration an genau einen „Resource Broker“ gebunden. Fällt letzterer aus, so sind die Benutzer des betreffenden „User Interface“ vom weiteren Betrieb ausgeschlossen. Das folgende Kapitel 6.4.2 stellt eine Lösung für dieses Problem vor.

- **Bedienbarkeit**

Die Bedienung des EDG-Systems ist geprägt von der Verwendung von Kommandozeilenbefehlen. Die einzelnen Rechenaufträge werden durch weltweit eindeutige Zeichenketten gekennzeichnet wie zum Beispiel:

```
https://grid-rb.physik.uni-wuppertal.de:9000/Z74vhFI8wD5C1RVqIQUGSA
```

Die Handhabung von mehreren Rechenaufträgen wird dadurch erschwert, dass es keine Möglichkeit gibt, alle Rechenaufträge eines Benutzers zu erfragen. Ist die Zeichenkette, die einen Rechenauftrag kennzeichnet, nicht mehr zugänglich, ist der Rechenauftrag insofern verloren, dass die Ergebnisse nicht mehr abgefragt werden können. Zu dieser Problemstellung wurde ebenfalls eine Lösung in Form einer graphischen Benutzeroberfläche erarbeitet, die im folgenden Kapitel 6.4.3 vorgestellt wird.

## 6.4.2 Entwicklung eines ausfallsicheren Ressource Brokers

Der „Resource Broker“ als zentrale Komponente im EDG übernimmt vielfältige Aufgaben. Er dient nicht nur der Verteilung der Rechenanforderungen, sondern ist auch für die Überwachung und die Buchhaltung im EDG verantwortlich. Abbildung 6.3 zeigt schematisch den vereinfachten Aufbau des „Workload Management“-Systems des EDG. „Workload Management“ bezeichnet die auf dem „Resource Broker“ installierte Softwarekomponente.

Im Detail übernimmt der „Resource Broker“ die folgenden Aufgaben:

- Prüfung des Benutzerzertifikats auf Gültigkeit,
- Abgleich des Zertifikats mit der Datenbank der virtuellen Organisation (die Prüfung wird auf dem „Computing Element“ wiederholt, vergleiche Kapitel 4.3.2),
- Analyse der JDL,
- Finden von passenden Ressourcen auf Grundlage der Anforderungen aus der JDL,
- Bereitstellen einer Input-Sandbox, in der alle Daten zwischengespeichert werden, solange bis der Auftrag zum „Computing Element“ übermittelt wurde,
- Bereitstellen einer Output-Sandbox, in der die Auftragsergebnisse gespeichert werden und für den Abruf durch den Benutzer bereit stehen,
- Erzeugen eindeutiger Auftragskennungen (siehe vorheriges Kapitel),

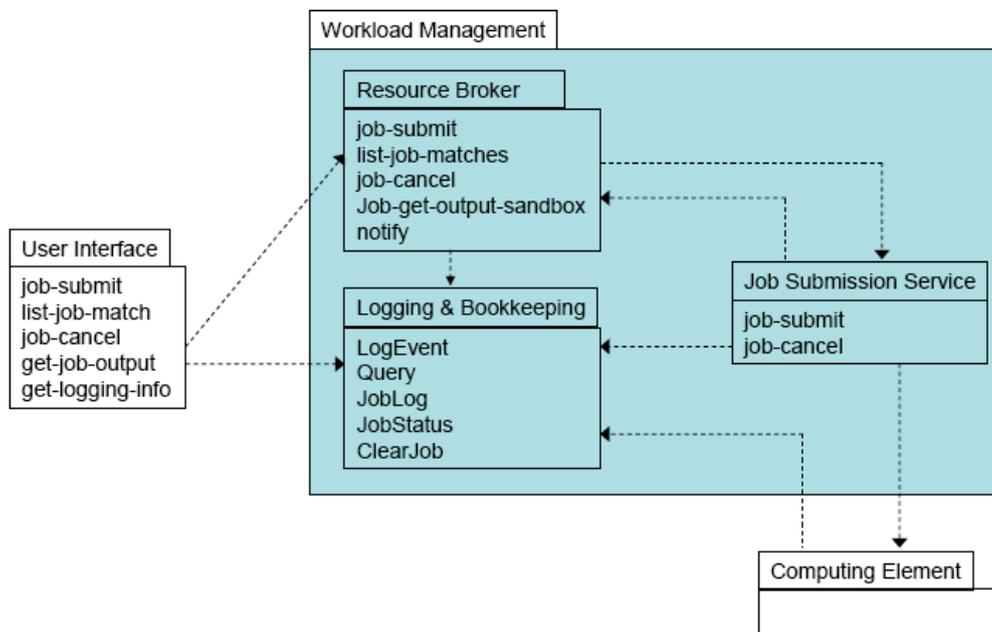


Abbildung 6.3: Schematischer Aufbau des „Workload Management“-Systems des EDG. Nach [81].

- Überwachung und Buchführung zur Nachverfolgung von Aufträgen während ihrer Ausführung und einige Zeit danach,
- Übertragung von Rechenaufträgen an das gewählte „Computing Element“.

Um diese Aufgaben zu erfüllen, bedient sich der „Resource Broker“ diversen Kommunikationsschnittstellen. Abbildung 6.4 zeigt eine Übersicht der Schnittstellen und soll die Komplexität dieser Softwarekomponente verdeutlichen.

Um nun die Abhängigkeit eines „User Interface“ von einem funktionierenden „Resource Broker“ zu verringern, gäbe es zwei Ansätze:

1. Eine komplette Neuimplementierung des „Workload Management“-Systems, die dann nicht mehr auf einem klassischen Client-Server-Ansatz besteht.
2. Die Entwicklung einer Software-Zwischenschicht zwischen dem „User Interface“ und dem „Resource Broker“ mit dem Ziel, die Instabilitäten dem Benutzer zu verbergen.

Der erste Ansatz erscheint nicht wirtschaftlich. Daher wurde der zweite Ansatz in einer Diplomarbeit [82] verwirklicht, die im Rahmen dieser Arbeit ange-regt und betreut wurde. Die Architektur, die hier verwendet wurde, besteht aus

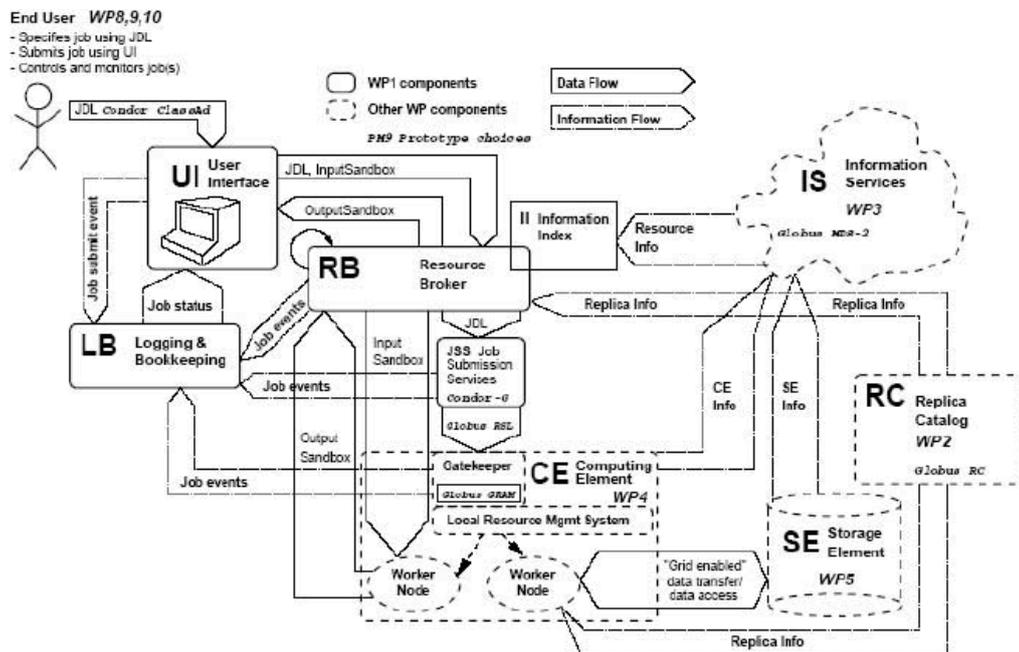


Abbildung 6.4: Kommunikationsschnittstellen des „Resource Brokers“. Nach [81].

einer klaren Trennung zwischen Informationsverteilung und Entscheidungssystem. Abbildung 6.5 zeigt diese Architektur.

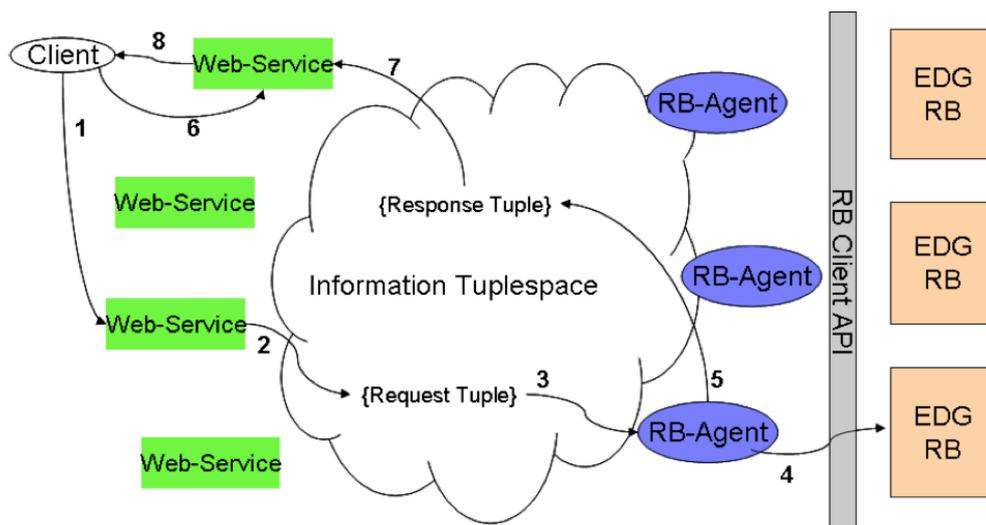


Abbildung 6.5: Architektur des Lastverteilungssystems. Aus [82].

Man erkennt einen sogenannten „Information Tuplespace“ als Kern des Systems. Als Grundlage dient das „Java Intelligent Network Infrastructure“ (Jini). Der Tuplespace dient als zentraler Datenspeicher und ist über mehrere Rechner verteilbar, so dass eine hohe Ausfallsicherheit entsteht. Daneben ermöglicht ein solches System die transparente Lastverteilung auf mehrere „Resource Broker“. Die Hauptaufgabe besteht nun aber darin, auf den Ausfall eines einzelnen „Resource Brokers“ zu reagieren, indem das betroffene System nicht mehr benutzt wird und zukünftige Rechenaufträge des Benutzers auf einen der verbleibenden Systeme verteilt wird. In regelmäßigen Abständen wird dann überprüft, ob der ausgefallene „Resource Broker“ wieder zur Verfügung steht und in diesem Fall wieder in das System integriert werden kann.

Mit diesem System konnten, bei einer entsprechenden Anzahl von zur Verfügung stehender „Resource Broker“ eine nahezu vollständige Ausfallsicherheit in diesem Teil des EDG-Systems realisiert werden. Details dazu sind in [82] zu finden.

### 6.4.3 Entwicklung einer webbasierten graphischen Benutzeroberfläche

Das Problem der Bedienbarkeit wurde mit der Entwicklung einer graphischen Benutzeroberfläche gelöst. Im Gegensatz zu der in Kapitel 5.3.4 vorgestellten Lösung für das Globus Toolkit, wurde diese Benutzeroberfläche komplett als Webservice implementiert, um den Benutzer die Installation von zusätzlicher Software zu ersparen. Dazu wurden sogenannte „Java Servlets“ verwendet, das sind serverseitige, dynamische Softwarekomponenten.

Auch diese Lösung wurde im Rahmen dieser Arbeit angeregt und betreuten Diplomarbeit [83] erstellt. Dabei bildet ein Webdienst quasi den gesamten Funktionsumfang des EDG/LCG-Systems ab, das heisst dass ein Benutzer von jedem Internet-fähigem Rechner nur durch einen Webbrowser das EDG/LCG-System über eine graphische Schnittstelle bedienen kann.

Die Benutzeroberfläche teilt sich im Wesentlichen in drei Bereiche auf:

- eine Rechenanforderungsverwaltung („Job Manager“ genannt),
- eine Dateienverwaltung („Data Manager“) und
- eine Rechteverwaltung.

Abbildung 6.6 zeigt einen Ausschnitt aus der Rechteanforderungsverwaltung der Oberfläche. Zur besseren Bedienbarkeit sind die vielen Möglichkeiten, eine Rechenanforderung für das EDG in einer JDL zu klassifizieren, in verschiedene Unterfenster aufgeteilt worden, die hier nicht alle gezeigt sind.

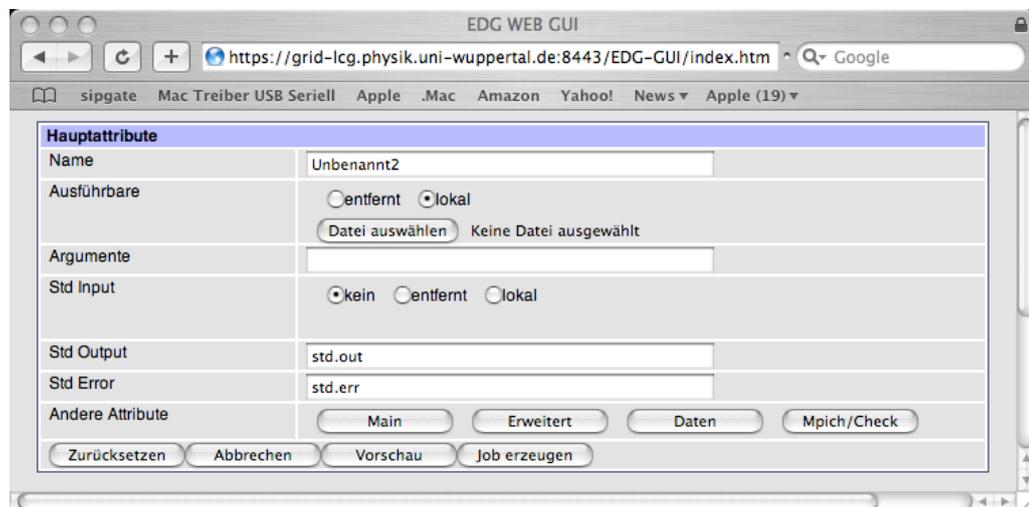


Abbildung 6.6: Ein Ausschnitt aus der Rechenanforderungsverwaltung der graphischen Benutzerschnittstelle zum EDG. Nur die wichtigsten Komponenten werden in diesem Fenster gezeigt.

Die so erstellten Rechenanforderungen können dann in einer Liste verwaltet werden. Sie lassen sich abbrechen, ihr Status lässt sich abfragen und die Ergebnisse können nach Beendigung der Anforderung abgerufen werden. Daneben lassen sich auch Dateien und Dateikopien (sogenannte „Replicas“) in ähnlicher Weise verwalten. Hier können einzelne Replicas oder direkt die gesamte Datei samt aller Kopien gelöscht werden. Es können aber auch Dateien in das EDG-System kopiert und alle möglichen Speicherplätze angezeigt werden, auf die der Benutzer mit seinem Rechten zugreifen kann. Abbildung 6.7 zeigt die Dateiverwaltung. Man erkennt eine Datei, die in drei Kopien vorhanden ist und ein LFN (vergleiche 4.3.2) trägt.

#### 6.4.4 Zusammenfassung

Das European Data Grid war Teil der Kampagne zur Wiederaufbereitung von DØ-Daten.

Die wiederaufbereiteten Ereignisse wurden routinemäßig mit Ergebnissen von Datensätzen verglichen, die ohne Verwendung von Grid-Technologien verarbeitet wurden. Abbildung 6.8 zeigt die  $J/\psi$ - und die  $K_{\text{short}}^0$  Signale in den invariante  $\mu\mu$ - beziehungsweise  $\pi\pi$ -Massenverteilungen. Gezeigt sind zwei Verteilungen, ein Datensatz, der im Rechenzentrum des IN2P3 in Lyon erzeugt und ein Datensatz, der im EDG verarbeitet wurde. Da beide Verteilungen erwartungs-

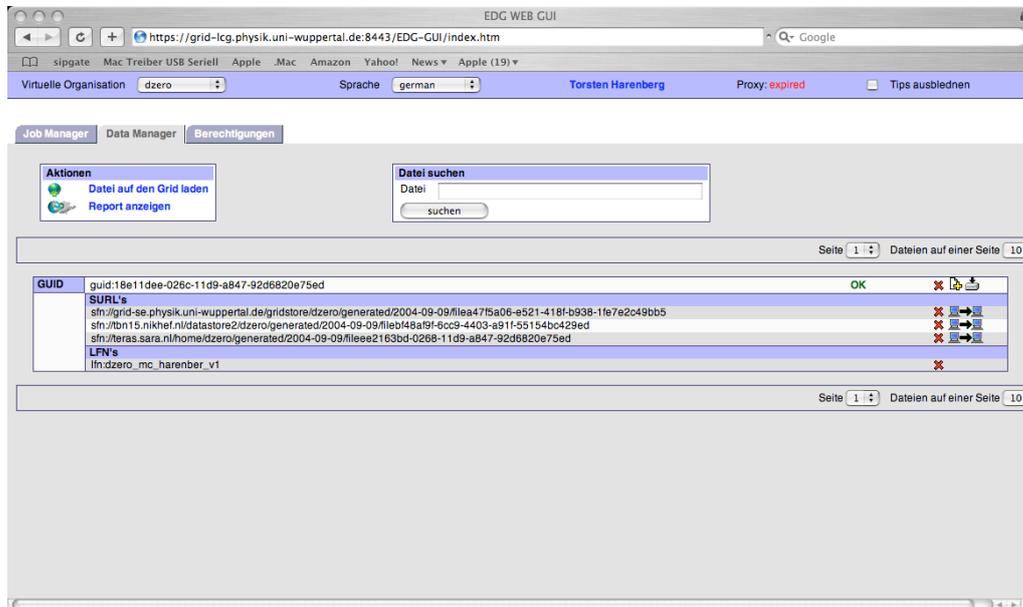


Abbildung 6.7: Dateiverwaltung der graphischen Benutzeroberfläche.

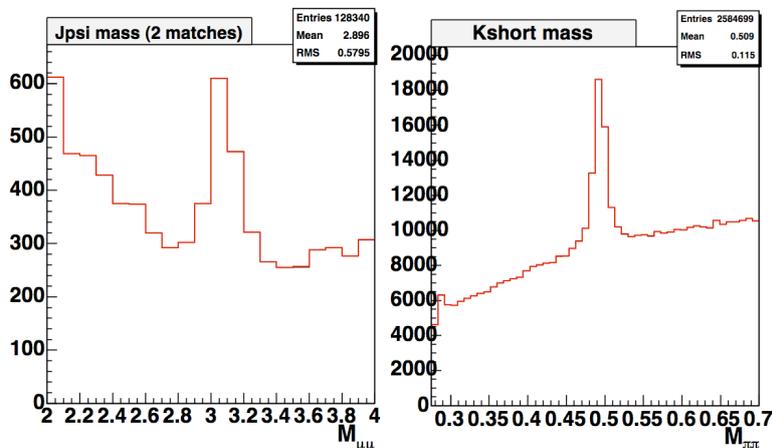


Abbildung 6.8:  $K^0_{\text{short}}$  und  $J/\psi$  Rekonstruktion in zwei Datensätzen. Die beiden Verteilungen (IN2P3 und EDG) liegen quasi übereinander.

gemäß sehr ähnlich sind, sind sie hier nicht mehr zu unterscheiden. Auch alle anderen Verteilungen zeigen ein solches Ergebnis. Damit sind die gewonnenen Ergebnisse nutzbar.

Abbildung 6.9 zeigt einen Vergleich der Rechenzeit zwischen dem IN2P3 und

dem EDG. Man erkennt, dass die „Worker Nodes“ im EDG tendentiell etwas mehr Zeit brauchen, aber der Effekt ist minimal und dadurch zu erklären, dass EDG ein verteiltes, inhomogenes Rechnersystem ist.

Das European Data Grid hat gezeigt, dass es die grundsätzliche Funktionalität besitzt, für ein Experiment der Hochenergiephysik Rechenarbeiten in einer allgemeinen Grid-Umgebung auszuführen. Es hat sich hierbei herausgestellt, dass die Stabilität des Informationssystems der limitierende Faktor war. Es wurde bewiesen, dass eine allgemeine Grid-Umgebung einen signifikanten Beitrag des Rechenzeitbedarfs eines aktuellen Experiments der Hochenergiephysik abdecken kann. Dies gilt selbst dann, wenn dieses Experiment eine Software-Infrastruktur besitzt, die nicht dazu ausgelegt wurde, in verteilten Rechensystemen eingesetzt zu werden.

Das Datenverwaltungssystem von DØ SAM wurde erfolgreich durch Etablierung einer Schnittstelle zum EDG-Datenverwaltungssystem in das European Data Grid integriert. Die Schnittstelle hat sich als leistungsfähig erwiesen, um den Datenverkehr sowohl vom und zum Fermilab als auch innerhalb des EDG zu bewältigen. Da diese Schnittstelle aber von ihrer Übertragungskapazität her begrenzt ist, wird in Wuppertal, aufbauend auf der vorliegenden Arbeit federführend an einer weitergehenden Integrierung des SAM-Systems gearbeitet. Hierbei wird unter anderem gewährleistet, dass die Rechenknoten jeweils selbstständig die zu diesem Zeitpunkt mit größtem Datendurchsatz erreichbare SAM-Station benutzen, um den Datentransfer vom und zum DØ-Datenverwaltungssystem zu benutzen. Damit wird eine höhere Ausfallsicherheit hergestellt und die Skalierbarkeit für eine umfangreichere Benutzung der Grid-Initiativen durch das DØ-Experiment erreicht. Zusätzlich wird es dadurch möglich werden, die Buchhaltung über bereits ausgeführte und laufende Rechenanforderung in SAM zu realisieren, welches eine größere Unabhängigkeit von R-GMA bedeutet. Gerade die Instabilität von R-GMA hat die Anzahl der wiederaufbereiteten Ereignisse limitiert.

Zusammenfassend wurden bei der hier vorgestellten Arbeit insgesamt 6.205.715 von 519.212.822 Events erfolgreich wiederaufbereitet und zum Fermilab geliefert. Obwohl der Anteil mit ungefähr 1,2 % gering erscheint, so ist zu berücksichtigen, dass die Entwicklungsarbeit zur Integration erst während der Laufzeit der Initiative geleistet wurde. Es wurden 1,7 Terabyte an Daten erzeugt und über die beschriebene Schnittstelle an das Fermilab zurück übertragen. Die nächste Initiative des DØ-Experiments für die Wiederaufbereitung von Daten wurde am 14. April 2005 gestartet und wird nun vollständig in verteilten Grid-Systemen stattfinden [73].

Daneben werden fortlaufend mit Hilfe der hier vorgestellten Lösung Simulationsdaten im EDG/LCG erzeugt und dem DØ-Experiment zur Verfügung gestellt. Abbildung 6.10 zeigt die integrierte Anzahl von generierten Simulationser-

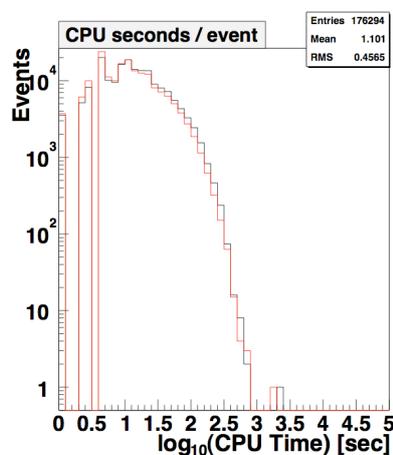


Abbildung 6.9: Vergleich der Rechenzeiten zwischen dem IN2P3 (rotes Histogramm) und dem EDG (schwarzes Histogramm) beim Wiederaufbereiten von DØ-Daten.

eignissen, die auf diese Weise, zentral gesteuert, erzeugt worden sind. Man erkennt die nahezu linearen Anstieg. Die Gesamtanzahl von ungefähr siebzehn Millionen Ereignissen steht im Verhältnis zu ungefähr 53 Millionen Ereignissen, die dem DØ-Experiment zum gleichen Zeitpunkt insgesamt zur Verfügung stehen [84]. Damit ist gezeigt, dass das LCG stabil genug ist, um auch für große Experimente die Grundlage für das verteilte Rechnen zu bilden. Für einige der aufgetretenen Probleme wurden im Rahmen von Diplomarbeiten [82,83] Lösungen erarbeitet, die sich im täglichen Einsatz bewährt haben.

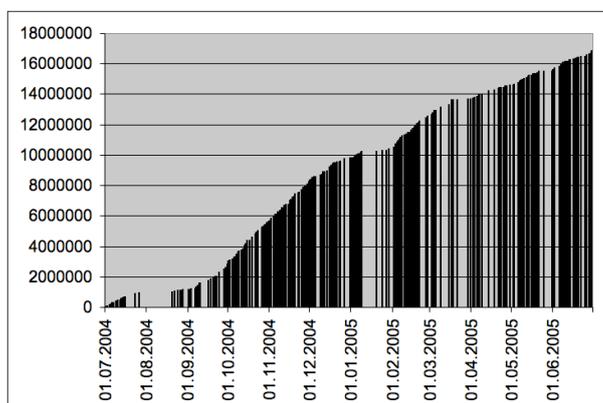


Abbildung 6.10: Integrierte Anzahl von generierten Simulationereignissen.



# Zusammenfassung und Ausblick

Im Rahmen der vorliegenden Arbeit wurde die Simulations- und Rekonstruktionssoftware von zwei laufenden Experimenten, dem DØ-Experiment am Tevatron und dem AMANDA-Experiment am Südpol, jeweils in einen weltweiten Grid-Rechnerverbund integriert. Dazu wurden Infrastrukturen geschaffen, damit die experimentspezifische Software in einer derartigen Umgebungen funktionieren kann. Außerdem wurden Routinen entwickelt, die die Rechenanforderungen überwachen und über die Ergebnisse Buch führen. Es wurde gezeigt, dass dieser Ansatz ein erfolgversprechender Weg ist, um nicht für den Betrieb in einem weltweiten Rechnerverbund konzipierte Software in solchen Umgebungen zu betreiben.

Auf dieser Grundlage wurde für das AMANDA-Experiment eine sehr große Anzahl von Simulationsereignissen erzeugt, die als Grundlage für diverse Diplomarbeiten sowie eigene Studien dienten. AMANDA war damit das erste laufende Experiment, welches Grid-Technologien in der Produktion von Simulationsdaten im größeren Umfang eingesetzt hat. Die Ergebnisse wurden in [61] veröffentlicht.

Während das auf dem Globus Toolkit basierte Grid noch von einer kleineren Anzahl von Instituten eingesetzt wurde, wurde für das DØ-Experiment ein ähnlicher Ansatz im European Data Grid realisiert, dem damals größten allgemeinen Grid-Rechnerverbund. In dessen Nachfolgeinitiative LCG sind mittlerweile mehr als 10.000 Rechner zusammengefasst.

Dabei wurde die komplette Simulations- und Rekonstruktionssoftware integriert, um damit an einer Kampagne zur Wiederaufbereitung der Daten teilzunehmen.

Da das DØ-Experiment ein eigenes Datenmanagement-System besitzt, musste dieses in das European Data Grid integriert werden. Zu diesem Zweck wurde erfolgreich eine Schnittstelle zwischen den beiden Systemen etabliert, welche die Daten zwischen den Systemen austauscht.

Damit wurden zum ersten Mal in einer allgemeinen Grid-Umgebung Daten von einem bestehenden Experiment verarbeitet. Die dafür nötige Infrastruktur zur Überwachung und zur Buchhaltung wurde entwickelt und hat sich bewährt. Mit ihr konnten während der Kampagne mehrere Instabilitäten im EDG-System aufgedeckt und beseitigt werden. Die R-GMA-Datenbank als Grundlage der Über-

wachung und Buchhaltung hat sich in ihrer Funktionalität bewährt, obwohl ihre Stabilität die beschränkende Größe war.

Für zwei der aufgetretenen Probleme, nämlich der Instabilität des Resource Brokers und der Bedienbarkeit, konnten im Rahmen von Diplomarbeiten Lösungen entwickelt werden.

Insgesamt konnten 1,2 % der DØ-Daten wiederaufbereitet und damit ein signifikanter Beitrag zur Kampagne geleistet werden. Die Ergebnisse und Erfahrungen sind in [85] publiziert. Die entwickelte Infrastruktur wird weiterhin benutzt und zur Zeit zur kontinuierlichen Produktion von Simulationsdaten im LCG eingesetzt. Außerdem wird in Wuppertal federführend eine weitreichendere Schnittstelle zwischen den Datenverarbeitungssystemen von DØ und dem LCG entwickelt, die eine völlig transparente Benutzung des LCG möglich machen werden.

Dieses wird wichtig, da immer mehr nationale Forschungsinstitute ihre großen Rechenkapazitäten nur noch im Grid-Verbund zur Verfügung stellen und keine dedizierten, experimentspezifischen Installationen mehr unterstützen.

Durch die DØ-Aktivität im EDG und LCG wurde die Bergische Universität Wuppertal in die Grid-Aktivitäten im Zusammenhang mit dem Bau des „Large Hadron Colliders“ (LHC) am CERN eingebunden und ist mittlerweile fester Bestandteil in diesen Initiativen. Vor allem durch die Bereitstellung von weltweit genutzter, teilweise sogar ausschließlich hier bereitgestellter Dienste wie die Dateiverwaltungsdienste für DØ (siehe Kapitel 6.2), konnte ein signifikanter Beitrag für diese Kollaborationen geleistet werden.

In Hinblick auf die enormen Anforderungen, die die LHC Experimente an eine Grid-Umgebung stellen, sind die gesammelten Erfahrungen unverzichtbar, um zukünftige Versionen der Grid-Middleware weiter an diese Anforderungen anzupassen. Insbesondere im Hinblick auf die Stabilität müssen weitere Anstrengungen erfolgen. Auch die Benutzerfreundlichkeit muss noch deutlich verbessert werden, um die Akzeptanz dieser verteilten Rechnersysteme zu erhöhen. Hierzu wurden Lösungen erarbeitet, die als Grundlage für zukünftige Entwicklungen dienen können.

# Literaturverzeichnis

- [1] S. L. Glashow. Partial Symmetries of weak Interactions. *Nucl. Phys.*, 22:579–588, 1961.
- [2] S. Weinberg. A Model of Leptons. *Phys. Rev. Lett.*, 19:1264–1266, 1967.
- [3] J. Goldstone, A. Salam, S. Weinberg. Broken Symmetries. *Phys. Rev.*, 127:965–970, 1962.
- [4] S. Eidelman et al. Review of particle physics. *Phys. Lett.*, B592:1, 2004.
- [5] W. Bartel et al. Observation of planar three jet Events in  $e^+ e^-$  Annihilation and Evidence for Gluon Bremsstrahlung. *Phys. Lett.*, B91:142, 1980.
- [6] B. Adeva et al. A Test of QCD based on four jet events from  $Z^0$  decays. *Phys. Lett.*, B248:227–234, 1990.
- [7] P. Abreu et al. The scale dependence of the hadron multiplicity in quark and gluon jets and a precise determination of  $C(A)/C(F)$ . *Phys. Lett.*, B449:383–400, 1999.
- [8] J. Abdallah et al. The measurement of  $\alpha(s)$  from event shapes with the DELPHI detector at the highest LEP energies. *Eur. Phys. J.*, C37:1–23, 2004.
- [9] European Committee for Future Accelerators. *Large Hadron Collider Workshop, Proceedings, Volume I*. CERN 90-10, 1990.
- [10] [http://www-d0.fnal.gov/computing/graphics/d0\\_private/gallery/2003/run172482\\_evt5512259\\_3D4.jpg](http://www-d0.fnal.gov/computing/graphics/d0_private/gallery/2003/run172482_evt5512259_3D4.jpg).
- [11] F. Schmidt. CORSIKA Shower Images.  
<http://www.ast.leeds.ac.uk/~fs/showerimages.html>.
- [12] D. Adams et al. The ATLAS Computing Model, Januar 2005. CERN-LHCC-2004-037/G-085.
- [13] V. Hess, 1912. Über Beobachtungen der durchdringenden Strahlung bei sieben Freiballonfahrten, *physikalische Zeitschrift*, 13:1084-1091.

- [14] H. Geenen, 2002. Energy reconstruction and spectral unfolding of atmospheric leptons with the AMANDA detector, Diplomarbeit, WU D 2002-10.
- [15] T. K. Gaisser, 2000. High Energy Neutrino Astronomy – The Cosmic Ray Connection. astro-ph/0011525v1.
- [16] W. Biermann, T. Stanev et al., 1993. Cosmic rays. IV. The spectrum and Chemical Composition above 10 GeV. *Astronomy and Astrophysics*, 274:902.
- [17] B. Wiebel-Sooth, 1998. Measurement of the allparticle energy spectrum and chemical composition of cosmic rays with HEGRA detector. Doktorarbeit.
- [18] AMANDA-Kollaboration. AMANDA II Public Outreach.  
[http://www.amanda.uci.edu/public\\_info.html](http://www.amanda.uci.edu/public_info.html).
- [19] AMANDA-Kollaboration. AMANDA Software Page.  
<http://amanda.wisc.edu/software>.
- [20] J. Wochele. KASCADE-Grande Homepage.  
[http://www-ik.fzk.de/KASCADE\\_home.html](http://www-ik.fzk.de/KASCADE_home.html).
- [21] P. Steffen. F2000: A standard AMANDA offline event format.  
<http://www-zeuthen.desy.de/~steffenp/f2000>.
- [22] P. Desiati. AMANDA-II MC Mass Production Status Report, März 2002. Vortrag auf dem AMANDA/IceCube Meeting in Berkeley.
- [23] DØ Kollaboration. DØ Presentations.  
<http://www-d0.fnal.gov/Run2Physics/displays/presentations/>.
- [24] Fermilab Visual Media Services.  
<http://www-mediaservices.fnal.gov/>.
- [25] A. M. Patwa. The forward preshower system and a study of the J/psi trigger with the DØ detector. FERMILAB-THESIS-2002-04.
- [26] DØ Kollaboration. DØ Silicon Tracker Technical Design Report.  
[http://d0server1.fnal.gov/projects/silicon/www/tdr\\_final.ps](http://d0server1.fnal.gov/projects/silicon/www/tdr_final.ps).
- [27] DØ Kollaboration. DØ Central Fiber Tracker.  
[http://d0server1.fnal.gov/projects/SciFi/cft\\_description.html](http://d0server1.fnal.gov/projects/SciFi/cft_description.html).
- [28] <http://www-d0.fnal.gov/hardware/cal/geometry/UsefulPlots.html>.
- [29] M. Klute. A Measurement of the  $t\bar{t}$  Production Cross Section in Proton-Antiproton Collisions at  $\sqrt{s}=1.96$  TeV with the Do Detector at the TEVATRON using Final States with a Muon and Jets. Internal report Bonn-IR-2004-02.

- [30] J. Apostolakis. GEANT - Detector Description and Simulation Tool.  
<http://wwwasd.web.cern.ch/wwwasd/geant/>.
- [31] C. Schmitt. Private Konversation.
- [32] I. Foster, C. Kesselman. *The Grid: Blueprint for a New Computing Infrastructure*. Morgan Kaufmann Publishers, November 1998.
- [33] I. Foster, C. Kesselman, S. Tuecke. *The Anatomy of the Grid: Enabling Scalable Virtual Organizations*. International J. Supercomputer Applications, 15(3), 2001.
- [34] A. B. Sculley, W. W. A. Woods. *B2B Exchanges: The Killer Application in the Business-to-Business Internet Revolution*. ISI Publications, 1st Edition, Januar 2000.
- [35] The Globus Alliance. Globus Homepage.  
<http://www.globus.org>.
- [36] Argonne National Laboratory. Mathematics and Computer Science Division Homepage.  
<http://www.mcs.anl.gov>.
- [37] The University of Southern California. Information Sciences Institute Homepage.  
<http://www.isi.edu>.
- [38] The University of Chicago. Distributed Systems Laboratory Homepage.  
<http://dsl.cs.uchicago.edu>.
- [39] The University of Edinburgh. The University of Edinburgh Homepage.  
<http://www.ed.ac.uk>.
- [40] Königlich Technische Hochschule Stockholm. Center for Parallel Computers Homepage.  
<http://www.pdc.kth.se>.
- [41] Zentralinstitute für Angewandte Mathematik am Forschungszentrum Jülich. UNICORE Homepage.  
<http://unicore.sourceforge.net>.
- [42] The University of Virginia. Legion: A Worldwide Virtual Computer.  
<http://legion.virginia.edu>.
- [43] Avaki Corporation. Avaki Homepage.  
<http://www.avaki.com>.

- [44] Department of Computer Sciences The University of Wisconsin. Condor Project Homepage.  
<http://www.cs.wisc.edu/condor>.
- [45] Global Grid Forum. Global Grid Forum Homepage.  
<http://www.ggf.org>.
- [46] I. Foster, C. Kesselman, G. Tsudik, S. Tuecke. A security architecture for computational grids. in ACM Conference on Computers and Security, pages 83-91. ACM Press, 1998.
- [47] S. Fitzgerald, I. Foster, C. Kesselman, G. von Laszewski, W. Smith, S. Tuecke. A Directory Service for Configuring High-Performance Distributed Computations. Sixth IEEE Symp. on High-Performance Distributed Computing, 1997.
- [48] J. Bester, I. Foster, C. Kesselman, J. Tedesco, S. Tuecke. GASS: A Data Movement and Access Service for Wide Area Computing Systems. Sixth Workshop on I/O in Parallel and Distributed Systems, May 1999.
- [49] P. Stelling, I. Foster, C. Kesselman, C. Lee, G. von Laszewski. A Fault Detection Service for Wide Area Distributed Computations. Proceedings of the 7th IEEE Symposium on High Performance Distributed Computing, Seiten 268-278, 1998.
- [50] W. Allcock, J. Bester, J. Bresnahan, A. Chervenak, L. Liming, S. Meder, S. Tuecke. GGF GridFTP Working Group Document, September 2002.
- [51] The European DataGRID project. Job Description Language HowTo. DataGrid-01-TEN-0102-0\_2, Dezember 2001.
- [52] The European DataGRID project. Architecture and Design – WP5 - Mass Storage Management. DataGrid-05-D5.2-0141-3-4, Februar 2002.
- [53] CERN Castor-Kollaboration. CASTOR Man pages.  
<http://castor.web.cern.ch/castor/DOCUMENTATION/MAN/CASTOR/#rfio>.
- [54] The European DataGRID project. Software Architecture Models - EDG 2.1. DataGrid-12-NOTE-0.6, Februar 2004.
- [55] The EGEE project. Information and Monitoring Service (R-GMA) – System Specification. EGEE-JRA1-TEC-490223-R\_GMA\_SPECIFICATION-v2-0, Juli 2004.
- [56] S. Fisher et al. R-GMA: First results after deployment. Proceedings of the 2003 Conference for Computing in High Energy Physics (CHEP 2003), La Jolla, Kalifornien, März 2003.

- 
- [57] Sun Microsystems Inc. Request for Comments (RFC) 1094, NFS: Network File System Protocol Specification, März 1989.
- [58] Sun Microsystems Inc. Request for Comments (RFC) 1813, NFS Version 3 Protocol Specification, Juni 1995.
- [59] Sun Microsystems Inc. et al. Request for Comments (RFC) 3530, Network File System (NFS) version 4 Protocol, April 2003.
- [60] Sun Microsystems Inc. Request for Comments (RFC) 1050, RPC: Remote Procedure Call Protocol Specification, April 1988.
- [61] T. Harenberg, K. H. Becker, W. Rhode, C. Schmitt. AMANDA - first running experiment to use GRID in production. *ECONF*, C0303241:MOAT010, 2003.
- [62] D. Ellard et al. NFS Tricks and Benchmarking Traps. Proceedings of the 2003 USENIX Annual Technical Conference, San Antonio, Texas, Juni 2003.
- [63] The Globus Project. GridFTP Universal Data Transfer for the Grid. Whitepaper, September 2000.
- [64] Information Science Institute at the University of Southern California. Request for Comments (RFC) 793, Transmission Control Protocol DARPA Internet Program Protocol Specification, September 1981.
- [65] K. E. Schinarakis, 2003. Untersuchung von Neutrino- und Myon-Flüssen in der Atmosphäre für den AMANDA-Detektor, Diplomarbeit, WU D 2003-08.
- [66] W. Rhode, D. Chirkin. Fluxes of Atmospheric Leptons in Large Underground Detectors. Proceedings of 26<sup>th</sup> International Cosmic Ray Conference (ICRC 1999), Salt Lake City, Utah, 17 - 25 Aug 1999.
- [67] D. Heck et al. Hadronic interaction models and the air shower simulation program CORSIKA. Prepared for 27<sup>th</sup> International Cosmic Ray Conference (ICRC 2001), Hamburg, Germany, 7-15 Aug 2001.
- [68] D. Heck. The air shower simulation program CORSIKA and hadronic interaction models. *Nachr. Forsch. Zentr. Karlsruhe*, 33:113–120, 2001.
- [69] T.K. Gaisser. *Cosmic Rays and Particle Physics*. Cambridge University Press, 2000.
- [70] Dmitry Chirkin. Fluxes of atmospheric leptons at 600-GeV - 60-TeV. 2004.
- [71] L. V. Volkova. ENERGY SPECTRA AND ANGULAR DISTRIBUTIONS OF ATMOSPHERIC NEUTRINOS. *Sov. J. Nucl. Phys.*, 31:784–790, 1980.

- [72] K. Daum et al. Determination of the atmospheric neutrino spectra with the Frejus detector. WUB-95-03.
- [73] D. Wicke. Data reprocessing on worldwide distributed systems. Proceedings of Computing in High-Energy Physics (CHEP '04), Interlaken, Switzerland, 27 Sep - 1 Oct 2004.
- [74] SAM Homepage.  
<http://d0db.fnal.gov/sam>.
- [75] P. Anderson. The Complete Guide to LCFG. 2001.  
<http://www.lcfg.org/doc/lcfg-guide.pdf>.
- [76] European Data Grid Project. EU Data Grid WP2 Homepage.  
<http://edg-wp2.web.cern.ch/edg-wp2>.
- [77] <http://pyxml.sourceforge.net>.
- [78] R.P. Harrold. RPM Package Manager Homepage.  
<http://www.rpm.org>.
- [79] A. Delgado Peris et. al. LHC Computing Grid: LCG-2 User Guide, Manuals Series, April 2005. CERN-LCG-GDEIS-454439.
- [80] Free Software Foundation. GNU m4 Project Homepage.  
<http://www.gnu.org/software/m4>.
- [81] The European DataGRID project. Definition of the Architecture, technical Plan and Evaluation Criteria for the Resource Co-Allocation Framework and Mechanisms for parallel Job Partitioning. DataGrid-01-D1.4-0127-1\_0, September 2002.
- [82] R. Kahrl. Entwicklung eines Lastverteilungssystem für das LHC Computing Grid, 2004. Diplomarbeit an der Fachhochschule Köln.
- [83] D. Vogel. Entwicklung eines grafischen Benutzerinterface für DataGrid, August 2004. Diplomarbeit an der Fachhochschule Niederrhein.
- [84] DØ Kollaboration. DØ Monte Carlo Production page.  
<http://www-d0.fnal.gov/computing/mcprod/mcc.html>.
- [85] T. Harenberg et al. Do data processing within EDG/LCG. Proceedings of Computing in High-Energy Physics (CHEP '04), Interlaken, Switzerland, 27 Sep - 1 Oct 2004.

# Abbildungsverzeichnis

1.1	Fundamentale Feynman-Graphen der QED . . . . .	5
1.2	Fundamentale Feynman-Graphen der QCD . . . . .	6
1.3	Wirkungsquerschnitte für die Proton-Proton-Streuung . . . . .	8
1.4	Beispiel eines DØ-Ereignisses . . . . .	9
1.6	Zeitliche Entwicklung des Ressourcenbedarf am zentralen ATLAS-Analysezentrum . . . . .	10
1.5	Beispiel eines Luftschauers . . . . .	11
2.1	Das Spektrum der kosmischen Strahlung und der Messbereich verschiedener Experimente . . . . .	13
2.2	Die drei Hauptbeiträge zum Spektrum der kosmischen Strahlung	14
2.3	Schematische Darstellung der Messung des Cherenkov-Lichts. Die Kegelfront bewegt sich durch die Anordnung von optischen Modulen (OM) des AMANDA-Detektors. . . . .	16
2.4	Schematische Darstellung eines optischen Moduls des AMANDA-Detektors . . . . .	17
2.5	Schematische Darstellung des AMANDA-Detektors. . . . .	19
3.1	Luftbildaufnahme des Tevatrons . . . . .	23
3.2	Schematische Darstellung der Beschleunigerstufen . . . . .	24
3.3	Querschnitt des DØ-Detektors . . . . .	26
3.4	Innere Komponenten des DØ-Detektors . . . . .	27
3.5	Schematischer Aufbau des Silizium-Detektors von DØ . . . . .	28
3.6	Überblick über das DØ-Kalorimeter . . . . .	29
3.7	Seitenansicht des DØ-Kalorimeters . . . . .	31
3.8	r-z Ansicht des DØ-Myonsystems . . . . .	32

3.9	Seitenansicht des Myonsystems . . . . .	33
4.1	Schematische Darstellung der schichtartigen Softwarestruktur des „WP5 Storage Element“ . . . . .	43
4.2	Illustration des Datenflusses durch das „WP5 Storage Element“ . . . . .	44
4.3	Die Subsysteme des Computing Elements . . . . .	46
4.4	Die Dienste der „Grid Monitoring Architecture“ . . . . .	47
4.5	Schematischer Aufbau eines Grids aus EDG-Komponenten . . . . .	48
5.1	Ein konventioneller Ansatz zum Aufbau eines verteilten Rechnersystems . . . . .	52
5.2	Übertragungsgeschwindigkeit (in MB/s) bei steigender Anzahl von Rechenknoten . . . . .	53
5.3	Ablauf einer Rechenanforderung. . . . .	56
5.4	Ablauf des Mechanismus zum Auflösen von Abhängigkeiten von dynamischen Bibliotheken und Dateien. . . . .	58
5.5	Der Hauptbildschirm der erstellten graphischen Benutzeroberfläche ohne aktive Rechenaufträge. . . . .	59
5.6	Eingabemaske für das CORSIKA-Programmpaket. . . . .	61
5.7	Hier realisierter Ansatz zum Aufbau eines verteilten Rechnersystems im Grid. . . . .	62
5.8	Myonantwort für alle Primärteilchen mit $E_\mu > 1,2 \text{ TeV}$ . . . . .	66
5.9	Fit der Myon-Spektren für vier Wechselwirkungsmodelle. . . . .	67
5.10	Fit des Myon-Spektrums in Abhängigkeit von $\cos(\theta)$ . . . . .	68
5.11	Fit der Myonneutrino-Spektren für vier Wechselwirkungsmodelle. . . . .	69
5.12	Fit der Elektronenneutrino-Spektren für vier Wechselwirkungsmodelle. . . . .	70
5.13	Vergleich der Myonmultiplizität in Abhängigkeit vom Wechselwirkungsmodell . . . . .	70
6.1	Die Realisierung des Datenaustausches zwischen den Datenmanagementsystemen von DØ und EDG. . . . .	75
6.2	Mögliche Zustände von Rechenanforderungen im European Data Grid . . . . .	78
6.3	Schematischer Aufbau des „Workload Management“-Systems . . . . .	85
6.4	Kommunikationsschnittstellen des „Resource Brokers“ . . . . .	86

---

6.5	Architektur des Lastverteilungssystems . . . . .	86
6.6	Ausschnitt der Rechenanforderungsverwaltung . . . . .	88
6.7	Dateiverwaltung der graphischen Benutzeroberfläche . . . . .	89
6.8	$K_{\text{short}}^0$ und $J/\psi$ Rekonstruktion in zwei Datensätzen . . . . .	89
6.9	Vergleich der Rechenzeiten zwischen dem IN2P3 und dem EDG .	91
6.10	Integrierte Anzahl von generierten Simulationereignissen . . . . .	91

# Tabellenverzeichnis

1.1	Die fundamentalen Kräfte der Natur . . . . .	3
1.2	Fermionen, ihre Wechselwirkungen und Quantenzahlen . . . . .	4
2.1	Übersicht über den Rechenbedarf der AMANDA- Simulations- programme . . . . .	21
3.1	Eigenschaften des Tevatron-Beschleunigers . . . . .	25
3.2	Überblick über die Parameter der Kalorimeters von DØ . . . . .	30
3.3	Testbeam-Ergebnisse zur Parametrisierung der Kalorimetrauflö- sung . . . . .	30
4.1	Überblick über die Grid-Protokolle des Globus-Toolkits v2 . . . . .	40
5.1	Übersicht über die generierten Luftschauer . . . . .	64
5.2	Parameter der CORSIKA-Simulation . . . . .	65
5.3	Der prozentuale Anteil der Myonen, Elektron- und Myoneneutri- nos pro Primärteilchen; der spektrale Index des Myonen-, Elektroneneutri- no- und Myoneneutrinospektrums . . . . .	71

# Glossar

## A

**Application Programming Interface (API)** Ein API ist die Schnittstelle, die ein Betriebssystem oder auch ein anderes Softwaresystem anderen Programmen zur Verfügung stellt, Seite 46.

## B

**Batch System** Ein Batch-System arbeitet Rechenaufträge unabhängig vom direkten Einfluss des Auftraggebers ab. In der Regel stehen einem Batch-System dazu mehrere Rechner zur Verfügung, auf denen es die Rechenaufträge bearbeiten kann. Das System bestimmt selbst, wann und auf welchem Rechner der Auftrag ausgeführt wird. Dadurch wird die Auslastung des Rechnersystems optimiert, Seite 45.

## C

**Certification Authority (CA)** Eine zentrale Instanz, die die Echtheit von Diensten, Rechnern und Personen Grid-weit elektronisch bescheinigt. Im EDG beruhen die Zertifikate auf dem X.509-Standard, Seite 48.

**Common Object Request Broker Architecture (CORBA)** Die Common Object Request Broker Architecture kurz CORBA ist eine objektorientierte Middleware, die plattformübergreifende Protokolle und Dienste definiert und von der Object Management Group (OMG) entwickelt wird. CORBA ermöglicht das Erstellen verteilter Anwendungen in heterogenen Umgebungen, Seite 37.

**Computing Element (CE)** Ein Rechner im EDG, der als Vorrechner für ein lokales Batch-System dient. Während der Resource Broker die Jobs an die einzelnen Institute verteilt, ist das CE für die Verteilung auf die

einzelnen Worker Nodes zuständig. Daneben übernimmt das CE die Kommunikation der Worker Nodes, Seite 45.

## D

**dynamische Bibliothek** Eine Bibliothek oder Bücherei (englisch: library) oder Programmbibliothek bezeichnet in der Programmierung eine Sammlung von Programmfunktionalitäten für zusammengehörende Aufgaben. Bibliotheken sind im Unterschied zu Programmen keine eigenständigen Einheiten sondern Hilfsmodule, die Programmen zur Verfügung stehen. Dynamische Bibliotheken werden erst bei Bedarf in den Arbeitsspeicher geladen und durch einen sogenannten Lader mit dem ausführbaren Programm verbunden. Dadurch muss eine Bibliothek, die von mehreren Programmen genutzt wird, nur einmal im Speicher gehalten werden, Seite 51.

## E

**Electronic Data Interchange (EDI)** Electronic Data Interchange (EDI) bezeichnet als Sammelbegriff alle elektronischen Verfahren zum asynchronen und vollautomatischen Versand von strukturierten Nachrichten zwischen Anwendungssystemen unterschiedlicher Institutionen, Seite 37.

**European Data Grid (EDG)** Das „European Data Grid“ war ein von der Europäischen Union gefördertes Projekt mit der Zielsetzung, die weltweit verteilte Rechen- und Speicherkapazitäten, die vor allem im Bereich der Hochenergiephysik vorhanden sind, gemeinschaftlich nutzbar zu machen. Die Software, die im Rahmen dieses Projektes erstellt wurde, wird nun im Rahmen des Projektes „Enabling Grids for E-science in Europe“ (EGEE) weiterentwickelt und im Rahmen des „LHC Computing Grid“-Projektes angewandt, Seite 40.

**Extensible Markup Language (XML)** Die Extensible Markup Language ist ein Standard zur Erstellung maschinen- und menschenlesbarer Dokumente in Form einer Baumstruktur. XML definiert dabei die Regeln für den Aufbau solcher Dokumente. Für einen konkreten Anwendungsfall („XML-Anwendung“) müssen die Details der jeweiligen Dokumente spezifiziert werden. Dies betrifft insbesondere die Festlegung der Strukturelemente und ihre Anordnung innerhalb des Dokumentenbaums. XML ist damit ein Standard zur Definition von beliebigen, in ihrer Grundstruktur jedoch stark verwandten Auszeichnungssprachen, Seite 44.

**F**

**Formula Translation (FORTRAN)** Eine höhere Programmiersprache, die häufig im Bereich numerischer Berechnungen eingesetzt wird. Wurde bereits im Jahr 1954 bei IBM entwickelt und seit 1957 eingesetzt. Die neueste Version stammt aus dem Jahr 2003, Seite 20.

**G**

**Grid Unique Identifier (GUID)** Eine eindeutige Zeichenkette, über die Dateien im EDG angesprochen werden. Wird automatisiert bei der Registrierung einer Datei vergeben. Jede Dateikopie trägt die selbe GUID. Beispiel: guid:579fa133-5751-4e23-bcd3-b0db36436902, Seite 42.

**GZK-Cutoff** Der GZK-Cutoff ist die nach den Physikern Greisen, Zatsepin und Kuzmin benannte Grenze für die Energie kosmischer Strahlung. Teilchen oberhalb dieser Energiegrenze können im Mittel nur etwa 5 Mpc durch das All propagieren, bevor sie mit den Photonen der kosmischen Hintergrundstrahlung via



oder



wechselwirken und damit einen Großteil ihrer Energie verlieren. Die Schwelle für obige Reaktionen liegen bei einigen  $10^{19}$  eV. Die experimentellen Resultate bezüglich der höchstenergetischsten kosmischen Strahlung sind widersprüchlich, Seite 12.

**I**

**Information Index (II)** Eine Datenbank, die den momentanen Zustand aller Ressourcen abbildet. Diese Informationen wertet der „Resource Broker“ aus. Wird mindestens einmal pro virtueller Organisation benötigt, Seite 46.

**J**

**JAVA** Java ist eine objektorientierte, plattformunabhängige Programmiersprache, die von Sun Microsystems zuerst unter dem Namen „Oak“ entwickelt wurde. Üblicherweise benötigen Java-Program-

me zur Ausführung eine spezielle Umgebung (Java Virtual Machine). Als Vorteil muss nur diese Umgebung an verschiedene Computer und Betriebssysteme angepasst werden. Sobald dies geschehen ist laufen die Java-Programme., Seite 37.

**JAVA Virtuelle Maschine** Die Java Virtual Machine (abgekürzt Java VM oder JVM) ist eine Laufzeitumgebung und virtuelle Maschine für Software. Die Java Virtual Machine führt den so genannten Java-Bytecode aus. Bytecode entsteht durch compilieren von Programmen, die in aller Regel in der Programmiersprache Java verfasst wurden. Der Sinn der Java Virtual Machine und des Bytecodes ist es, plattformunabhängige Programme zu ermöglichen. Die JVM dient dabei als Schnittstelle zur Maschine und zum Betriebssystem. Nachteilig dabei ist, dass zur Ausführung des Bytecodes die JVM benötigt wird, Seite 54.

**Job Description Language (JDL)** Eine standardisierte Beschreibung von Rechenanforderungen im EDG/LCG. Dabei werden Zuweisungen nach folgender Syntax

```
Attribut = wert;
```

verwendet, Seite 41.

## L

**Lichtjahr** Eine sehr gebräuchliche Einheit in der Astronomie zur Angabe von Entfernungen (es ist keine Zeiteinheit). Ein Lichtjahr, deutschsprachig abgekürzt mit Lj, ist definiert als die Wegstrecke, die Licht oder Strahlung generell im Vakuum in einem Jahr zurücklegt. Die Vakuumlichtgeschwindigkeit beträgt 299.792.458 Meter pro Sekunde, somit entspricht ein Lichtjahr ungefähr 9.4605 Billionen Kilometer, Seite 14.

**Lightweight Directory Access Protocol (LDAP)** Das Lightweight Directory Access Protocol (LDAP) ist in der Computertechnik ein Netzwerkprotokoll, das die Abfrage und die Modifikation von Informationen eines Verzeichnisdienstes (eine im Netzwerk verteilte hierarchische Datenbank) erlaubt, Seite 39.

**Logical File Name (LFN)** Ein benutzerdefinierter Name für eine Datei im EDG, Seite 42.

## M

**Metadaten** Als Metadaten oder Metainformationen bezeichnet man allgemein Daten, die Informationen über andere Daten enthalten. Bei den beschriebenen Daten handelt es sich oft um größere Datensammlungen (Dokumente) wie Bücher, Datenbanken oder Dateien. Eine allgemein gültige Unterscheidung zwischen Metadaten und normalen Daten existiert allerdings nicht. So werden auch Angaben von Eigenschaften eines Objektes (beispielsweise Personennamen) als Metadaten bezeichnet, Seite 72.

## N

**Named Pipe** Die Pipe (englisch für Rohr, Röhre) bezeichnet eine gepufferte Datenverbindung zwischen zwei Prozessen nach dem First In - First Out (FIFO)-Prinzip. Unter Unix definiert eine Pipe einen unidirektionalen Datenstrom von einem Prozess zum anderen. Sie ist unter Unix eines der mächtigsten Werkzeuge, um die sequentielle Abarbeitung von Befehlen auf einem bestimmten Datenbestand zu ermöglichen. Bei einer anonymen Pipe ist die Kommunikation dabei auf zwei Prozesse gleichen Ursprungs beschränkt. Eine Named Pipe, auch FIFO genannt, ist eine Pipe, die von zwei Prozessen zur Laufzeit über einen Dateinamen zum Lesen oder Schreiben geöffnet werden kann. Bei einer Named Pipe müssen die Prozesse keinen gemeinsamen Ursprung haben, die Prozesse müssen lediglich zum Zugriff auf die Pipe autorisiert sein und den Namen der Pipe kennen, Seite 44.

**Network File System (NFS)** Das Network File System - abgekürzt NFS - ist ein von Sun Microsystems entwickeltes Protokoll, das den Zugriff auf Dateien über ein Netzwerk ermöglicht. Dabei werden die Dateien nicht übertragen, sondern die Benutzer können auf Dateien, die sich auf einem entfernten Rechner befinden, so zugreifen, als wenn sie auf ihrer lokalen Festplatte abgespeichert wären. Bei diesem UNIX-Netzwerkprotokoll handelt es sich um einen Internet-Standard, der auch als verteiltes Dateisystem (oft auch englisch als „Distributed File System“ genannt) bezeichnet wird, Seite 52.

**Network Information Service (NIS)** Network Information Service oder NIS ist ein Verzeichnisdienst zur Verteilung von Konfigurationsdaten wie Benutzernamen oder Rechnernamen in einem Computernetzwerk. NIS entstand bei Sun Microsystems unter dem Namen „Yellow Pages“

(YP), der aus markenrechtlichen Gründen aufgegeben werden musste. NIS/ YP kann als zentrales Verzeichnis für Benutzeraccounts, Rechnernamen und anderen brauchbaren Daten in einem Computernetzwerk verwendet werden, Seite 52.

**Nuklid** Ein Nuklid ist ein Atom, welches durch die Protonen- und Neutronenzahl sowie durch seinen Energiezustand gekennzeichnet ist. Es unterscheidet sich von einem Isotop durch die Elektronenhülle, die das Isotop umgibt. Ein Nuklid kann wie das Isotop stabil oder instabil sein. Instabile Nuklide werden auch als Radionuklide bezeichnet, Seite 12.

## P

**Photomultiplier** Photomultiplier sind spezielle Elektronenröhren um schwache Lichtsignale (bis hin zu einzelnen Photonen) zu verstärken. Die Funktionsweise entspricht der eines Sekundärelektronenvervielfachers mit einer zusätzlichen Photokathode. Die Photonen treffen auf die Photokathode und schießen so Elektronen aus deren Oberfläche. Die freigesetzten Photoelektronen werden in einem elektrischen Feld beschleunigt und treffen auf weitere Elektroden (sogenannte Dynoden), aus deren Oberfläche jedes auftreffende Elektron mehrere Sekundärelektronen herausschlägt. Somit nimmt die Anzahl der Elektronen kaskadenartig zu. Der Verstärkungsfaktor wächst exponentiell mit der Anzahl der Dynoden. Der erzeugte Stromstoß wird als Ausgangssignal ausgekoppelt.

Anwendungen sind beispielsweise Detektoren für Elementarteilchen (zum Beispiel Gamma-Spektrometer). Häufig werden sie in Verbindung mit Szintillationszählern in Großdetektoren eingesetzt. In anderen Fällen geht es darum, die schwachen Lichtblitze, welche durch hochenergetische kosmische Strahlung in der Hochatmosphäre entstehen, nachzuweisen (Cherenkov-Leuchten) und auf diese Weise Informationen über kosmische Strahlungsquellen zu gewinnen. Auch zur Messung schwacher Lichtsignale werden Photomultiplier eingesetzt, Seite 17.

## R

**Remote File I/O (RFIO)** Ein vom CERN entwickeltes Protokoll zum Zugriff auf entfernten Massenspeicher. Wird vor allem beim Zugriff auf große Bandspeicher verwendet, Seite 44.

**Replica Catalog (RC)** Enthält eine Sammlung von Diensten, die die Verwaltung von Dateikopien (sogenannte „Replica“) ermöglicht. Dabei wird auch der Zusammenhang zwischen physikalischen Dateinamen (SURL) zu globalen, logischen Dateinamen (LFN) – realisiert. Jede virtuelle Organisation benötigt diese Dienste einmal. Sie unterteilen sich in mehrere Dienste:

- „Replica Location Service“ (RLS)  
Sucht Kopien von Dateien innerhalb des EDG und speichert deren physikalische Dateinamen.
- „Replica Metadata Catalog“ (RMC)  
Fragt logische Dateinamen (LFN) ab und verwaltet diese.
- „Replica Optimization Service“ (ROS)  
Sucht nach der für eine Anforderung besten Dateikopie, so dass diese schnellstmöglich der Anwendung zur Verfügung gestellt werden kann.

Diese werden unter dem Begriff „Data Management Services“ zusammengefasst, Seite 45.

**Resource Broker (RB)** Ein Rechner der einen zentralen Dienst beherbergt, der zu jeder Rechenanforderung einen möglichst günstigen Ort für deren Ausführung bestimmt. Dabei wird als günstig verstanden, dass die Anforderung möglichst schnell ausgeführt wird und das möglichst wenig benötigte Daten transportiert werden müssen. Wird mindestens einmal für jede virtuelle Organisation benötigt, Seite 47.

**Resource Specification Language (RSL)** Die Prozessbeschreibungssprache, in der das Globus Toolkit Anforderungen für die Ausführung eines Prozesses erwartet. In ihr sind z.B. der Name des Programms, evtl. Argumente, Anforderungen aus ausführenden Ressourcen wie Speicherbedarf, etc. festgelegt. In neueren Versionen des Globus Toolkits werden diese Information in einem XML-Format erwartet. Im Gegensatz dazu verwendet das EDG eine eigene Sprache namens JDL, Seite 39.

## S

**Site File Name (SFN)** Bezeichnet den physikalischen Speicherplatz einer Dateikopie im EDG, beinhaltet aber im Gegensatz zum TURL und SURL weder das Zugriffsprotokoll noch das Verwaltungsprotokoll. Es besteht daher aus dem vollständigen Rechnernamen und dem vollständigen Pfad auf die Datei.

Beispiel: `gppse05.gridpp.rl.ac.uk/storage/atlas/generated/2004-12-13/pythia_5000.tar`, Seite 42.

**Storage Element (SE)** Das Storage Element ist die Schnittstelle des EDG-Systems zu Massenspeichersystemen, die auf ein einheitliches Protokoll abgebildet werden. Dazu wird `gsiftp` verwendet. Es existieren mehrere Arten von Storage Elementen, die sich in ihrer Funktionalität unterscheiden, Seite 41.

**Storage Uniform Ressource Locator (SURL)** Bezeichnet den physikalischen Speicherplatz einer Dateikopie im EDG und die Information, mit welcher Schnittstelle die Dateikopie verwaltet wird. Im EDG ist dies immer SRM. Der SURL beinhaltet das Verwaltungsprotokoll, den vollständigen Rechnernamen und den vollständigen Pfad der Dateikopie. Beispiel: `srm://gppse05.gridpp.rl.ac.uk/storage/atlas/generated/2004-12-13/pythia_5000.tar`, Seite 42.

**Structured Query Language (SQL)** Structured Query Language (SQL) – deutsch: strukturierte Abfragesprache – ist eine Abfragesprache für relationale Datenbanken. Sie hat eine relativ einfache Syntax, die an die englische Umgangssprache angelehnt ist, und stellt eine Reihe von Befehlen zur Definition von Datenstrukturen nach der relationalen Algebra zur Manipulation von Datenbeständen (Anfügen, Bearbeiten und Löschen von Datensätzen) und zur Abfrage von Daten zur Verfügung. Durch ihre Rolle als Quasi-Standard ist SQL von großer Bedeutung, da eine weitgehende Unabhängigkeit von der benutzten Software erzielt werden kann, Seite 77.

## T

**Transfer Universal Resource Locator (TURL)** Bezeichnet den physikalischen Speicherplatz einer Dateikopie im EDG und dessen Zugriffsprotokoll. Es besteht daher aus dem Site File Name und einem Zugriffsprotokoll, welches den Zugang zu der Dateikopie ermöglicht. Beispiel: `gsiftp://gppse01.gridpp.rc.ac.uk/flatfiles/01/data/d0/d073888b9c936f8790a3cd58ea60625624`, Seite 43.

## U

**User Interface (UI)** Die Schnittstelle des EDG-Systems zum Benutzer. Es handelt sich dabei um einen Rechner, auf dem neben dem Betriebssystem ein Satz von Software installiert wurde, der alle zur Bedienung

des EDG-Systems nötigen API-Schnittstellen und Befehle enthält. Auf dem UI selbst werden aber keine Rechenanforderungen abgearbeitet, Seite 41.

## V

**virtuelles privates Netzwerk (VPN)** Ein Virtuelles Privates Netzwerk (VPN) ist ein Computernetz, das zum Transport privater Daten ein öffentliches Netzwerk (zum Beispiel das Internet) nutzt. Teilnehmer eines VPN können Daten wie in einem internen LAN austauschen. Die einzelnen Teilnehmer selbst müssen hierzu nicht direkt verbunden sein. Die Verbindung über das öffentliche Netzwerk wird üblicherweise verschlüsselt. Der Begriff „Privat“ impliziert jedoch nicht, wie vielfach angenommen, dass es sich um eine verschlüsselte Übertragung handelt. Eine Verbindung der Netzwerke wird über einen Tunnel zwischen VPN-Client und VPN-Server ermöglicht. Meist wird der Tunnel dabei gesichert, aber auch ein ungesicherter Klartexttunnel ist ein VPN, Seite 37.

## W

**Webservice** Ein Webservice ist ein Dienst, der mit Hilfe von XML auf der Basis von Internet-Netzwerkprotokollen erbracht wird. Die Dienste, die Webservices erbringen, sind nicht für menschliche Benutzer, sondern für Softwarekomponenten, die Informationen sammeln, bestimmt. Webservices bilden die drei wichtigsten Teile der Zusammenarbeit zwischen Client und Server ab: das Zusammenfinden, Binden und den Datenaustausch, Seite 37.

**Worker Node (WN)** Eine Worker Node ist ein Rechner, der Rechenanforderungen im EDG abarbeitet. Dabei ist jede WN an ein Computing Element angebunden, welches unter anderem die Kommunikation übernimmt. Dabei bedient sich dieses einem „Batch System“, Seite 45.

## X

**X.509** X.509 ist ein ITU-T-Standard für eine Public Key-Infrastruktur und ist derzeit der wichtigste Standard für digitale Zertifikate. Es setzt ein striktes hierarchisches System von vertrauenswürdigen Zertifizierungsstellen (englisch: certification authority) voraus, die Zertifikate

erteilen können. Nahezu alle Webbrowser beinhalten eine vorkonfigurierte Liste von vertrauenswürdigen CAs, deren ausgestellten Zertifikaten der Browser vertraut, Seite 48.

# Danksagung

Viele Personen haben dazu beigetragen, diese Arbeit zu verwirklichen und denen ich dafür sehr dankbar bin.

Insbesondere meinem Doktorvater Herrn Prof. Dr. Becks möchte ich sehr dafür danken, mir die Möglichkeit gegeben zu haben, dieses interessante Thema zu verwirklichen. Auch Herrn Prof. Dr. Mättig möchte ich für die stets gute Zusammenarbeit und für die Unterstützung bedanken, die mir zu Teil wurde.

Herrn Prof. Dr. Dr. Rhode gilt mein besonderer Dank für die Möglichkeit, einen Teil der Arbeit in der Astroteilchenphysik verwirklichen zu können und für die stetige Bereitschaft, mich in allen Problemen zu unterstützen.

Bei allen Mitarbeitern der Gruppen Hochenergiephysik und Astroteilchenphysik möchte ich mich für die gute Zusammenarbeit und das freundschaftliche Arbeitsklima bedanken. Besonders hervorheben möchte ich hier Herrn Dr. Andreas Kootz, Herrn Dr. Peter Gerlach, Herrn Dr. Christian Schmitt. Ihnen möchte ich insbesondere für die fruchtbaren Diskussionen danken, die so manches Mal den entscheidenden Hinweis zur Lösung eines Problems gegeben haben. Herrn Karl-Heinz Becker gebührt mein besonderer Dank bei der Zusammenarbeit bei allen Rechnerproblemen.

Nicht zuletzt gilt mein Dank meiner Familie und meiner Lebensgefährtin für die Geduld, die sie immer wieder aufgebracht haben.