# Abschlussbericht zum GTB-Projekt ViSpaS - Dezember 2000 „Polyatomare Systeme: Visuell gesteuerte Simulation und Analyse komplexer Oberflächenreaktionen“

Auftragsnummer TK 602 - NT 104

## Projektverantwortung:

Prof. Dr. K. Hermann, Prof. Dr. M. Scheffler, Fritz-Haber-Institut, Berlin

## Mitarbeiter:

J. Kühn, F. Mante (Gemeinsames Rechenzentrum, Fritz-Haber-Institut)

A. Hetey, Dr. P. Kratzer, Dr. J. Neugebauer (Abt. Theorie, Fritz-Haber-Institut)

Dr. H. Lederer, A. P. Seitsonen (Rechenzentrum Garching der MPG)

## Kurzbeschreibung des Projekts

Zur Untersuchung von Eigenschaften polyatomarer Systeme werden Molekulardynamik-Simulationsrechnungen auf dem Hochleistungsrechner der Max-Planck-Gesellschaft in Garching durchgeführt, wobei die Online-Steuerung der Rechnung und die Analyse der Ergebnisse am Fritz-Haber-Institut in Berlin grafisch durchgeführt werden (ViSpaS - Visuelle Simulation polyatomarer Systeme). Für eine effiziente Arbeitsweise der Materialwissenschaftler und Chemiker müssen Ausgabe-Daten vom Cray T3E-Rechner in Garching zeitkritisch zur Grafik-Umgebung in Berlin übertragen werden. Hierzu sind zeitweise Übertragungsraten im Bereich von 240-400 Mbit/sec erforderlich, die nicht mehr mit den vorhandenen B-WiN-Datenleitungen zu erreichen sind.

Dieser Bericht enthält nach einer Einführung in die Thematik und Ausführungen zu den hardwaretechnischen Details eine genaue Beschreibung der Umsetzung des beschriebenen Vorhabens.

# Inhaltsverzeichnis

# 0 Einleitung

In dem Projekt "Polyatomare Systeme: Visuell gesteuerte Simulation und Analyse komplexer Oberflächenreaktionen" wurden computertechnische Methoden untersucht, die es erlauben, numerische Ergebnisse, die auf einem massiv parallelen Großrechner erhalten werden, an einem davon entfernten Arbeitsplatzrechner innerhalb einer grafischen Benutzeroberfläche zu analysieren. Zum anderen soll die Steuerung der numerischen Berechnung vom Grafikrechner aus erfolgen. Ein wesentlicher Aspekt bei dem Projekt war die Ausnutzung der schnellen Datenübertragung zwischen den Großrechnern am Rechenzentrum Garching der Max-Planck-Gesellschaft (MPG) und dem Fritz-Haber-Institut der MPG in Berlin, die durch den Aufbau einer Gigabit-Testbed-Übertragungsstrecke ermöglicht werden sollte.

Als Ergebnis der Entwicklungs- und Erprobungsarbeiten während des Berichtszeitraums ist es uns in ersten Schritten gelungen, die Berechnung einer chemischen Reaktion zwischen einem Molekül und einer Oberfläche interaktiv zu steuern und die vom Zentralrechner gelieferten Daten nach ihrer Übertragung auf dem Arbeitsplatzrechner zu visualisieren. Aufgrund verschiedener Schwierigkeiten, die während der Projektdurchführung auftraten, konnte bisher jedoch lediglich ein Testbetrieb der bei dem Projekt entwickelten und eingesetzten Softwarekomponenten verwirklicht werden. Über den Regelbetrieb der Steuerungs- und Visualisierungssoftware und über das Verhalten der Datenübertragungsstrecke im praktischen Betrieb bei einer interaktiven Visualisierung können noch keine zuverlässigen Aussagen getroffen werden. Es wurden jedoch eingehende Tests der erzielbaren Übertragungsraten durchgeführt, die erkennen lassen, dass die durch das Gigabit-Testbed bereitgestellte Übertragungskapazität für die Anforderungen in dem vorliegenden Projekt im Prinzip ausreichend ist.

Die zum Einsatz kommende Software besteht aus mehreren Komponenten, zum einen auf dem Arbeitsplatzrechner, einer SGI Octane am FHI Berlin, zum anderen auf dem Großrechner, einer Cray T3E am Rechenzentrum Garching. Auf dem Arbeitsplatzrechner besteht die benötigte Software aus einer grafischen Benutzeroberfläche zur Steuerung, einem Grafik-Programmpaket zur Visualisierung sowie aus diversen Hilfsprogrammen, die der Interaktion zwischen diesen beiden Softwarekomponenten dienen. Auf der Seite des Großrechners kommt das numerische Simulationsprogramm fhi98md zum Einsatz. Gleichzeitig wird aber auch eine weitere Softwarekomponente, der sog. Datenserver, benötigt, der die Interaktion mit dem Arbeitsplatzrechner ermöglicht und das Abschicken angeforderter Daten vom Großrechner an den Arbeitsplatzrechner übernimmt.

Die Interaktivität innnerhalb des verwirklichten Konzepts ermöglicht es, das numerische Simulationsprogramm vom Arbeitsplatzrechner aus auf dem Großrechner zu starten, die Konvergenz beim Programmlauf zu verfolgen und das Programm nötigenfalls anzuhalten. Ferner ist es möglich, interaktiv eine Auswahl aus verschiedenen physikalischen Größen zu treffen, die der Benutzer visualisieren möchte und die jeweils dazu benötigten Daten vom Großrechner anzufordern. Die zunächst angestrebte weitreichende Interaktivität würde es notwendig machen, den Programmablauf des Simulationsprogramms fhi98md mit den Anforderungen des Benutzers nach einem bestimmten Typ von Ausgabedaten zu synchronisieren. Diese Aufgabe erwies sich als schwierig zu verwirklichen, da bei einem hochgradig parallelisiertem Programm wie dem fhi98md eine Ausgabeanforderung während des Programmlaufs ein Anhalten aller an der Simulation beteiligten Prozessoren erfordern würde.

Um herauszufinden, wie lange die Wartezeiten sind, die durch ein Anhalten des Programms und der direkten Übermittlung von Programmdaten über eine Socketverbindung

über das Gigabit-Testbed verursacht werden, wurden eingehende Tests der Datenübertragungsrate durchgeführt (siehe Abschnitt II.5 "Netz-Durchsatzmessungen"). Es zeigte sich, dass die direkte Beeinflussung des parallelisierten Programms durch Anforderung einer Socket-Ausgabe einen Performance-Verlust zur Folge hat. Dies kann zum Verlust an Stabilität des Programmlaufs führen, da Störungen der Datenübermittlung bei diesem Konzept einen Abbruch des Simulationsprogramms zur Folge haben können. Die erzielbare Datenübertragungsrate, die das Gigabit-Testbed ermöglicht, erwies sich für unsere Zwecke als ausreichend.

Wegen der auftretenden Stabilitätsprobleme erschien es uns ratsam, im Weiteren ein anderes Konzept zu verfolgen, das grundsätzlich ohne kontinuierliche Synchronisation zwischen Erzeugung und Übermittlung der Daten auskommt. Dabei schreibt das fhi98md-Programm die Ausgabedaten zunächst in Dateien. Die vom Benutzer am entfernten Rechner kommenden Anfragen werden von einem von der eigentlichen Simulationsrechnung getrennten Programm, dem Datenserver, entgegengenommen. Dieser übernimmt auch die Verwaltung der Ausgabedaten und konvertiert die Daten in eine Form, die vom Grafikprogramm auf dem Arbeitsplatzrechner empfangen werden kann (siehe Abschnitt II.4.3 "Datenserver und Kommunikationsprotokoll").

Es stellte sich heraus, dass eine häufige Auffrischung des Dateninhalts während des Programmlaufs nur für *einen* vorgegebenen Typ physikalischer Daten (von vielen möglichen), z.B. die elektronische Ladungsdichte, sinnvoll ist. Dies ermöglicht dem Benutzer ein Verfolgen der elektronischen Konvergenz während des Programmlaufs. Daneben wird in größeren Zeitabständen eine weitere Datei vom fhi98md-Programm geschrieben, die einen vollständigen Datensatz enthält. Aus diesem können alle weiteren vom Benutzer angefragten Informationen rekonstruiert werden. In Anbetracht der begrenzten Projektlaufzeit erschien uns dieser Ansatz als am schnellsten realisierbar. Weitere Entwicklungsarbeit wurde auf die Ausgestaltung der grafischen Analyse verwandt, die dem Benutzer der Visualisierungssoftware am Arbeitsplatzrechner einen komfortablen Zugriff auf die benötigten Informationen ermöglicht (siehe Abschnitt II.4.4 "Grafische Kontrollumgebung").

Der im vorliegenden Projekt enthaltene Teilaspekt der Visualiserung wurde mittels des professionellen Programmpakets Amira verwirklicht. Das Programmpaket gestattet eine vielseitige grafische Darstellung und Analyse von räumlich-dreidimensionalen Datensätzen. Die Auswahl der Grafiksoftware erfolgte in vergleichenden Untersuchungen, wie im Abschnitt II.3.2 näher beschrieben wird.

Als Testfall wurde zunächst ein einfaches physikalisch-chemisches Beispiel, die Reaktion eines Arsen-Moleküls mit der Galliumarsenid-Oberfläche, ausgewählt. Im späteren Stadium sollten dann zusätzliche chemische Reaktionen an Oberflächen, beispielsweise die Oxidation von Oberflächen, untersucht werden. Die interaktive Vorgabe von atomaren Geometrien für verschiedene zu untersuchende chemische Reaktionen spielte während der bisherigen Testphase noch keine Rolle. Der Schwerpunkt der zur Verfügung stehenden grafischen Benutzeroberfläche lag auf der Analyse der Simulationsergebnisse. Der interaktive Aspekt der grafischen Benutzeroberfläche (z.B. Bewegen von Atomen durch den Benutzer mit der Maus) konnte bisher noch nicht verwirklicht werden.

Voraussetzung der Rechnerverbindung im Gigabit-Testbed (GTB) ist die Schaltung entsprechender Datenleitungen. Diese wurde, auch für die anderen Teilprojekte des GTB, soweit sie Berliner / Potsdamer Forschungseinrichtungen betreffen, in einer Installationsphase zu Beginn des Projektes durchgeführt. Ein kurzer Bericht hierzu ist im Abschnitt I "Netzinfrastruktur" enthalten.

# I Netzinfrastruktur

Im Rahmen dieses Projekts und des parallelen Projekts TIKSL wurde das in Abb. 1 dargestellte Netz aufgebaut. Das Gemeinsame Rechenzentrum der Berliner Max-Planck-Institute am Fritz-Haber-Institut (GRZ) hat hiervon den Teilbereich Berlin / Potsdam vom ZIB über das FHI bis zum AEI geplant und realisiert.

Hierzu wurden die folgenden projektbezogenen Teilschritte durchgeführt und abgeschlossen:

- Beschaffung der ATM-Hardware zum Anschluß des Albert-Einstein-Instituts (AEI) und des FHI an das Konrad-Zuse-Rechenzentrum (ZIB)
- Inbetriebnahme der Strecke zwischen FHI und AEI über Glasfaser und WDM, Inbetriebnahme der Strecke zum ZIB
- Anschluß der FHI-Gigabit-Workstation am FHI
- Aufbau eines IP-Netzes innerhalb der ATM-Infrastruktur

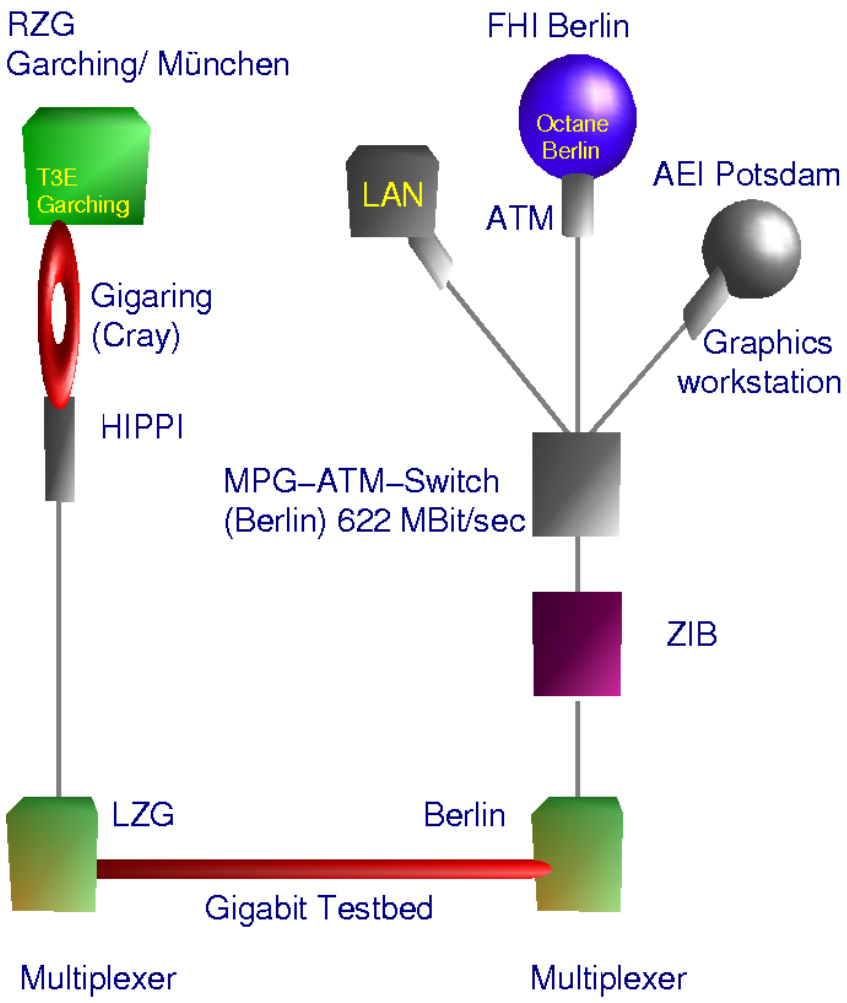


*Abb. 1: Netztopologie*

# I.1 ATM-Hardware

## I.1.1 Beschaffung der ATM-Hardware

Aus organisatorischen Gründen wurden alle ATM-Komponenten für den Anschluß an das ZIB und für die Verbindung zum AEI zentral vom GRZ beschafft. Nach der separaten Beschaffung der Interfaces zum Anschluß der Workstations besteht die ATM-Infrastruktur aus je einem ATM-Switch im GRZ (Berlin-Dahlem) und im AEI (Golm) mit den folgenden Konfigurationen:

| | |
|---|---|
| ATM-Switch GRZ: | Fore ASX 1000 mit<br>622 Mbit/s-Interface zum ZIB<br>622 Mbit/s-Interface zum Anschluss der lokalen Workstation<br>622 Mbit/s-Interface zum AEI<br>4 x 155 Mbit/s-Interface für administrative Aufgaben |
| ATM-Switch AEI: | Fore ASX 200-BX mit<br>622 Mbit/s-Interface zum GRZ<br>2 x 622 Mbit/s-Interfaces zum Anschluss lokaler Workstations<br>4 x 155 Mbit/s-Interface für administrative Aufgaben |

## I.1.2 Inbetriebnahme der ATM-Strecke Berlin-Dahlem bis Golm

Die MPG hatte sich schon frühzeitig um eine eigene Verbindung zwischen dem GRZ und dem neuen Campus in Golm bemüht. Im Rahmen des Anschlusses des Standortes Potsdam-Babelsberg der Universität Potsdam an das BRAIN wurde auch ein "MPG-eigenes" Faserpaar reserviert. Verhandlungen über die Weiterführung dieser Fasern bis nach Golm scheiterten aus finanziellen Gründen. In Verhandlungen mit der Universität Potsdam, die über ein dediziertes Faserpaar zwischen den Standorten Babelsberg und Neues Palais verfügte, wurde statt dessen vereinbart, die Strecke aus Investitionsmitteln der MPG mit Wellenlängen-Multiplexern zu versehen, die eine vierfache Ausnutzung der Verbindung ermöglichen. Eine "Farbe", die 622 Mbit/s-fähig ist, bleibt für MPG-Zwecke reserviert (ein weiterer Nutzer ist der DFN-Verein). Die Universität Potsdam stellt der MPG ein Faserpaar zwischen ihren Standorten Neues Palais und Golm zur Verfügung. Im Rahmen der Campus-Erschließung Golm hat die MPG eigene Fasern zum UNI-Standort Golm installiert.

Angesichts der technisch und administrativ komplexen Streckensituation (*MPG-Fasern* GRZ - FU-Bibliothek, *FU-Fasern* Bibliothek - ZEDAT, *LIT-Fasern* ZEDAT - Rathaus Zehlendorf, *LIT-Fasern* Rathaus Zehlendorf - UNI Potsdam/Babelsberg, *WDM-Strecke* UNI Potsdam/Babelsberg - UNI Potsdam/Neues Palais, *UNI-Potsdam-Fasern* Neues Palais - Golm, *MPG-Fasern* UNI Potsdam/Golm - MPG/Golm) konnte die ATM-Verbindung zwischen den Switches am GRZ und am AEI überraschend zügig in Betrieb genommen werden. Anfangs auftretende Störungen in der Verbindung konnten nicht eindeutig zugeordnet werden (Abstimmung der WDMs bzw. Bauprobleme auf dem MPG-Campus Golm). Seit Mai 1999 läuft die Verbindung weitgehend zuverlässig, die einzigen aufgetretenen Störungen konnten eindeutig auf Probleme der Stromversorgung im örtlichen Bereich Babelsberg der Universität Potsdam zurückgeführt werden.

Die Anbindung an das ZIB erfolgt über eine relativ übersichtliche Strecke (GRZ - FU Bibliothek - ZEDAT - ZIB). Nach Installation des ATM-Switches am ZIB konnte der Kontakt zum Switch des GRZ ohne weitere Probleme hergestellt werden.

## I.1.3 Anschluss der Gigabit-Workstation am FHI

Nach Lieferung des ATM-Interfaces für die SGI-Workstation am FHI war der Anschluss an den lokalen ATM-Switch (im gleichen Gebäude) problemlos.

## I.1.4 Aufbau eines IP-Netzes innerhalb der ATM-Infrastruktur

Nach Herstellung der ATM-Konnektivität zwischen IPP-Garching, ZIB, AEI-Golm und FHI wurden IP-Verbindungen zwischen diesen Standorten aufgebaut. Wegen des Einsatzes von ATM-Switches unterschiedlicher Hersteller bei den verschiedenen Institutionen einerseits und unterschiedlicher Erfahrungen mit ATM-Protokollen andererseits erwies es sich als durchaus schwierig, alle Standorte in einem homogenen IP-Netz zusammenzufassen. Dies führte dazu, daß verschiedene Varianten von Übertragungs- und Signalisierungsverfahren nacheinander ausprobiert wurden. Dieser Umstand war für die Verbindung zwischen den Knoten IPP-Garching, ZIB und FHI völlig problemlos, da es sich hierbei um Leitungsabschnitte handelte, die ausschließlich dem Testbed dienten.

Eine gänzlich andere Situation bestand auf dem Teilstück zwischen FHI und AEI-Golm. Diese Strecke diente von Anfang an auch der B-WiN-Anbindung der drei Max-Planck-Institute am Standort Golm, die aus technischen Gründen nicht wie ursprünglich geplant auf PVC-Basis, sondern übergangsweise als LAN-Emulation realisiert werden mußte. Bei der Installation der unterschiedlichen Varianten der Testbed-Verbindungen mußte jeweils darauf geachtet werden, daß der laufende Betrieb nicht oder, wenn unvermeidbar, nur kurzfristig nachts unterbrochen wurde.

Unter diesem Aspekt erwies sich der für das Testbed durchaus zu rechtfertigende Trial-and-Error-Ansatz bezüglich der Wahl eines geeigneten Übertragungs- bzw. Signalisierungsverfahrens als störend. Speziell in der Zeit Oktober/November 1999 wurden die Verfahren beinahe im 48-Stunden-Takt geändert. Wenn es bereits in der Planungsphase eine verbindliche Festlegung darüber gegeben hätte, mit welchem Verfahren IP auf ATM übertragen werden sollte, wären solche Probleme vermieden worden.

Seit Mitte November 1999 sind alle Standorte so miteinander verbunden, daß die Anwendungen diese Infrastruktur nutzen können. Nach einer erneuten Änderung der Signalisierung im Februar 2000 konnte auch eine AEI-Anwendung auf der CeBit in Hannover unter Verwendung der Testbed-Strecken demonstriert werden. Von diesem Zeitpunkt an sind keine weiteren Änderungen an der Konfiguration vorgenommen worden.

# I.2 Visualisierungs-Workstation SGI Octane

## I.2.1 Beschaffung der Visualisierungs-Workstation im FHI

Die Auswahl und Beschaffung der lokalen Grafik-Workstation für die Visualisierung wurde mit drei weiteren Beschaffungen im Rahmen des Gigabit-Projekts koordiniert. Die beteiligten Institute waren Albert-Einstein-Institut Potsdam (AEI), Konrad-Zuse-Institut für Informationstechnik Berlin (ZIB), Rechenzentrum Garching der Max-Planck-Gesellschaft (RZG) und Fritz-Haber-Institut Berlin (FHI). Herr Hege (ZIB) übernahm hierbei die Federführung. Probleme ergaben sich zunächst bei der Unterstützung des ATM-Interfaces durch die Hersteller (keine getesteten Treiber bzw. fehlende OS-Unterstützung), wobei nach Verhandlungen mit verschiedenen Firmen (HP, Sun, SGI, Compaq) im April 1999 vier SGI-Workstations Octane beschafft wurden. Die Beschaffung beinhaltete die ATM-Interfaces (Firma Fore Systems) einschließlich Treiber-Software und Upgrades auf die neue Generation der SGI-Grafikkarten (Z-CON). Im FHI wurde eine vorläufige Workstation Ende Mai 1999 geliefert, Ende August 1999 wurde die R10000-IP28 CPU gegen R12000-IP30 ausgetauscht und die ATM-OC12-Karte geliefert. Die ATM-Karte konnte erfolgreich installiert und die Verbindung zum Switch des ZIB aufgebaut werden.

Erst Anfang Dezember 2000 wurde von der Firma SGI die neue Grafikkarte geliefert. Ein Update des Betriebssystems, das Einspielen der Patches und das Auswechseln der Hardware misslang vorerst trotz telefonischer Rückfrage bei SGI. In einem weiteren Versuch konnte die neue Grafik-Hardware eingesetzt werden. Es musste jedoch festgestellt werden, dass die neue Grafikkarte den mitgelieferten Bildschirm nur unzureichend unterstützt.

## I.2.2 ATM-Verbindung SGI Octane Berlin - CRAY Garching

Die Verbindung über das ATM-Netz nach Garching funktionierte nach anfänglichen Problemen gut. Ausführliche Tests zum Datendurchsatz vom Molekulardynamik-Programm fhi98md werden im Abschnitt II.5 beschrieben.

Weitere Tests der Verbindung zwischen Visualisierungs-Workstation Berlin (VWB, Computername "rodin") und numerischem Großrechner (Cray T3E, Computername "pc") sowie Verbindungen zum und vom Konrad-Zuse-Institut Berlin wurden von M. Schünemann (ZIB) im Rahmen des Projektes Meta (Metacomputing durch Kopplung von Hochleistungsrechnern, URL siehe Referenzen) mit dem Programm *netperf* durchgeführt.

Der Verbindungsaufbau zwischen FHI und Garching kann aufgrund von Routing-Problemen scheitern. In diesem Fall muss die Verbindung von Garching aus aktiviert werden, wie ein Hinweis von A. Merzky (ZIB) ergab.
(login an der Cray über B-WiN-Verbindung, ping an die GTB-Adresse der VWS am FHI: "**`ping -i 90 rodin.gtb.clip.dfn.de`**")

# II Anwendung

## II.1 Physikalisch-chemische Problemstellung, computertechnische Umsetzung

Der Einsatz von Parallelrechnern zur Durchführung von Elektronenstruktur-Berechnungen erlaubt es, in relativ kurzer Zeit eine große Anzahl von Daten über das interessierende physikalische oder chemische System zu erhalten. Einerseits enthalten diese Daten wertvolle Informationen über das untersuchte System, deren Kenntnis ein schnelleres Fortschreiten bei dessen Erforschung ermöglicht, andererseits ist eine sinnvolle Interpretation dieser Vielzahl von Daten nur über eine visuelle Auswertung möglich. Die hohe Übertragungsrate des Gigabit-Testbeds ermöglicht es, die erforderliche aufwendige Visualisierung von stark strukturierten Datenfeldern (z.B. elektronische Ladungsdichten und Wellenfunktionen) zeitgleich mit ihrer Berechnung durchzuführen. Die interessierenden Größen lassen sich am sinnvollsten durch Isoflächen- oder Isoliniengrafiken darstellen. Optional sollen atomare Strukturmodelle diesen Grafiken überlagert werden, um eine optische Zuordnung der erkennbaren Strukturen zu ermöglichen.

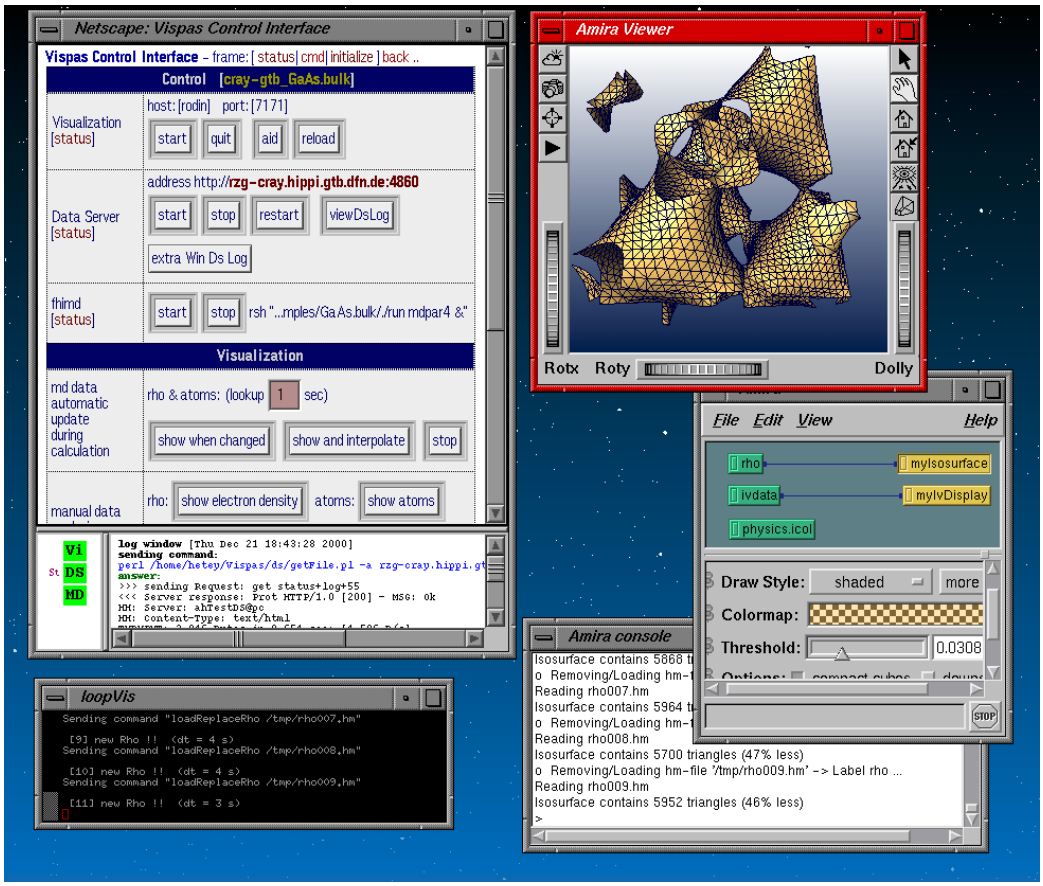


*Abb. 2: Screenshot Visualisierungs-Workstation Berlin: links Kontrollumgebung, rechts Visualisierung mit Software Amira*

Das Programm für die Elektronenstrukturberechnung (fhi98md) ist ein Fortran90-Programm, das auf einer massiv-parallelen *distributed-memory*-Maschine läuft. Es muss

während des Programmlaufs eine Steuerungsmöglichkeit bestehen, um auszuwählen, welche Daten zur Visualisierung übermittelt werden sollen. Dazu läuft ein separater Prozess (*Datenserver*) auf der Cray T3E, der das Konvertieren und Versenden der Ausgabedaten im gewünschten Format sowie das Empfangen und die Weitergabe möglicher Steuersignale während des Programmlaufs von fhi98md übernehmen kann (siehe Abschnitt II.4.3 "Datenserver und Kommunikationsprotokoll").

Über eine grafische Benutzeroberfläche auf der Workstation am FHI kann der Benutzer die Parameter einstellen, Programme starten und die Daten visualisieren. Der Benutzer sieht auf dem Bildschirm der Workstation sowohl die grafische Benutzeroberfläche (Kontrollumgebung) als auch die Visualisierungssoftware (vgl. Abb. 2).

Die Funktionalität der Visualisierungssoftware wird im Abschnitt II.3, die einzelnen Komponenten der Kontrollumgebung im Abschnitt II.4.1 beschrieben.

## II.2 Ablauf einer Beispielsitzung mit fhi98md

Ziel der Programmentwicklung war die Schaffung einer Arbeitsumgebung für einen theoretischen Physiker / Chemiker, der mit dem Programm fhi98md arbeiten will. Die Arbeitsumgebung soll dabei den Transfer sowie das Visualisieren der Daten erleichtern. Eine Hilfe zum "Aufsetzen" einer Rechnung sowie die Verwaltung der Rechenaufträge wurden zunächst nicht integriert, da es dafür bereits eine befriedigende Lösung am FHI gibt (Software EZWave).

- **Ausgangspunkt**: Der Benutzer befindet sich vor der Visualisierungs-Workstation in Berlin (VWB) und möchte mit dem fhi98md-Programm auf der Cray T3E in Garching rechnen.
- **Starten der grafischen Benutzeroberfläche** zur Steuerung der Sitzung auf der VWB.
  Im Web-Browser Netscape klickt man auf den Link der ViSpaS-Projekt-Seite. Man kann bereits gespeicherte Parameter über eine Auswahlliste oder die Voreinstellungen laden (siehe Abb. 10 und 11 in Abschnitt II.4.4). Die Parameter werden in ein html-Formular geladen. Der Benutzer kann nun einzelne Parameter ändern und dann den Knopf "Initialize" klicken. Es erscheint die Kontrollumgebung (siehe Abb. 12). Jetzt kann der Benutzer den Datenserver, die Visualisierungssoftware und das fhi98md-Programm starten. Alternativ kann das fhi98md-Programm durch das queueing-System auf der Cray gestartet werden.
- **Starten des Datenservers**
  Der Datenserver wird über einen remote-shell-Aufruf gestartet. Es wird ein Link zum aktuellen fhi98md-Arbeitsverzeichnis aufgebaut.
- **Starten des Visualisierungsprogramms** und Initialisierung mit entsprechendem Skript.
  Der Benutzer kann nun an der Steueroberfläche die zu visualisierende Größe aus einem Menü auswählen, zum Beispiel die elektronische Ladungsdichte. Diese Wahl wird mit den entprechenden Parametern an den Datenserver auf der Cray T3E weitergegeben. Die vom Datenserver gelieferten Daten werden dann zur Visualisierung an die VWB weitergeleitet.
- **Visualisierung**
  Zuerst werden typischerweise die Ladungsdichte in der Initialisierungsphase von fhi98md sowie Näherungswerte für die Energieeigenwerte der Wellenfunktionen ausgegeben. Falls vom Benutzer die Ladungsdichte zur Visualisierung ausgewählt wurde,

wandelt der Datenserver diese Daten in ein entsprechendes Grafik-Format um und sendet sie über das Gigabit-Testbed an die VWB, wo sie von der Visualisierungssoftware gelesen und visualisiert werden. Der Benutzer kann bei der Visualisierung der Elektronendichte zwischen verschiedenen Modi wählen: einmalige Übertragung, automatische Übertragung bei Änderung der Elektronendichte, automatische Übertragung mit Interpolation. Im letzteren Fall werden die Daten gepuffert, bis mindestens zwei Elektronendichten aus aufeinanderfolgenden Iterationen vorliegen. Zur Glättung des zeitlichen Verlaufs von Veränderungen werden die Daten linear zwischen den beiden Datensätzen interpoliert, in das hypermesh-Format umgewandelt und dann von der Visualisierungssoftware angezeigt.

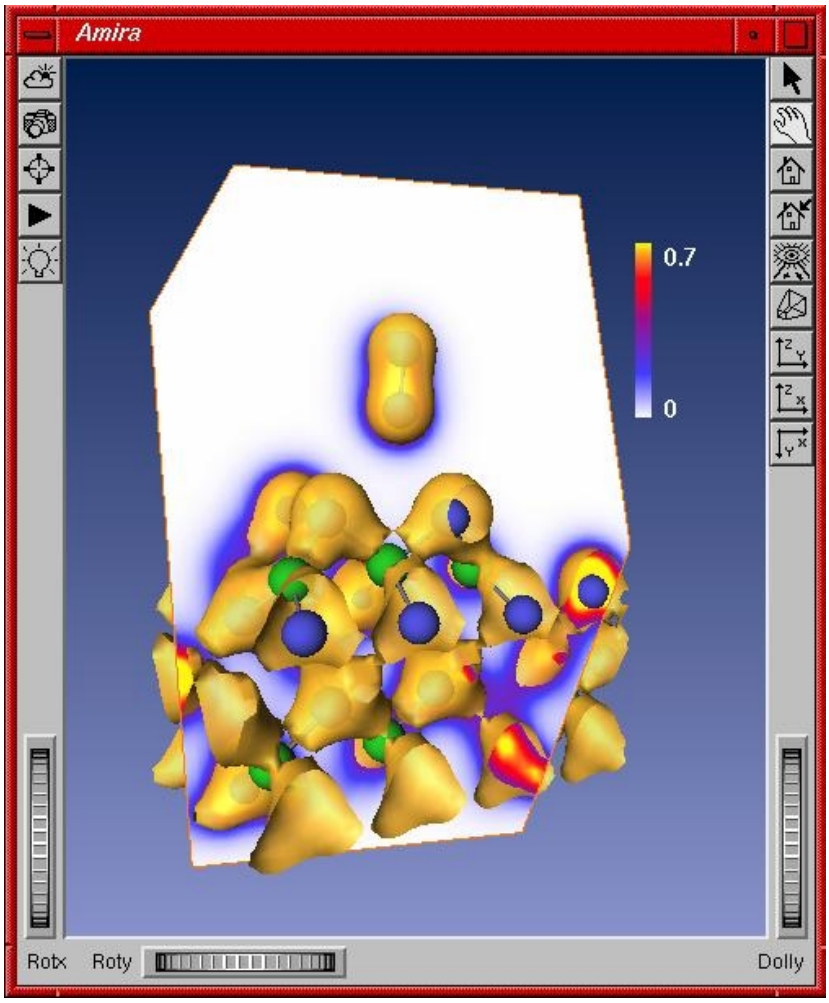


*Abb. 3: Amira Viewer: exemplarische Ansicht der zu visualisierenden Größen: Atome im balls-and-sticks-model (blau und grün) mit Elektronendichte (gelb) und Contour-Plot*

Jetzt entscheidet sich der Benutzer für eine andere Visualisierungsmöglichkeit, beispielsweise dafür, eine bestimmte Wellenfunktion visualisieren zu lassen. Die Auswahl der Wellenfunktion erfolgt grafisch, indem der Benutzer unter den angezeigten Energieeigenwerten einen durch Anklicken mit der Maus auswählt. Hierfür öffnet die Kontrollumgebung ein Zusatz-Fenster mit den Auswahlmöglichkeiten. Die Liste der aktuellen Energieeigenwerte wird davor vom Datenserver übermittelt und weiterverarbeitet (siehe auch ausführliche Beschreibung im Abschnitt II.4.1.1).

Wenn die Elektronendichte konvergiert ist, d.h. sich nicht mehr sichtbar verändert, wird der Benutzer zur detaillierten Analyse übergehen. Er möchte zum Beispiel die Wellenfunktion in einer bestimmten Schnittebene genauer inspizieren. Die Auswahl der Schnittebene und die Visualisierung durch zweidimensionale Isolinien-Darstellung erfolgt innerhalb der Visualisierungssoftware durch das Zuschalten entsprechender bereitgestellter Objekte.

- Vor **Beenden** der Sitzung kann der Benutzer über Mausklick die grafische Kontrollumgebung veranlassen, das Programm fhi98md zu beenden. Hier können auch der Datenserver und die Visualisierungssoftware beendet werden.

# II.3 Programmentwicklung Visualisierung

Die Visualisierung der physikalischen Größen ist ein zentrales Ziel in diesem Projekt. Im vorliegenden Abschnitt werden die Visualisierungssoftware sowie die Aufgaben, die sie übernehmen soll, vorgestellt.

## II.3.1 Aufgaben der Visualisierungssoftware

Die Visualisierungssoftware soll folgende Aufgaben übernehmen:

- Darstellung der atomaren Geometrie als '*balls-and-sticks*'-Modell
  Hierbei werden die Atome als Kugeln dargestellt, die Verbindungen zwischen den Atomen ("Bindungen") als Zylinder.
- Darstellung der kontinuierlichen physikalischen Größe als
  a) dreidimensionale Isofläche
  b) dreidimensionale Isofläche und zusätzlich zweidimensionale farbcodierte Isolinien-Abbildung in einer frei wählbaren Ebene
  c) Isolinien-Darstellung in einer frei wählbaren Ebene
- Elementare grafische Funktionen wie
  a) Wahl der Funktionswerte, für die die Isocontourfläche angezeigt werden soll
  b) Wahl von Kamerastandpunkt, Perspektive, Colormaps, ...
- Exportieren der Bilder in Bild-Dateien (Snapshots) zum nachträglichen Erstellen einer Film (Animation)

Der Aufruf von Visualisierungs-Funktionen soll von außerhalb der Visualisierungssoftware möglich sein (*remote control*)

## II.3.2 Auswahl und Tests von Visualisierungssoftware

Im Hinblick auf den speziellen Einsatz (interaktive Visualisierung und grafische Steuerung von Reaktionen an Oberflächen) wurden verschiedene Grafiksoftwarepakete zur Visualisierung begutachtet und getestet:

- IBM Data-Explorer
- NAG Iris-Explorer
- Amira der Firma Indeed - Visual Concepts

Die Visualisierungssoftware IBM Data-Explorer steht seit Mai 1999 kostenlos als "Source Code" zur Verfügung. Sie konnte erfolgreich installiert und getestet werden, erwies sich jedoch in der Benutzung als recht unhandlich und lief instabil. Insbesondere war der Umgang mit den visualisierten Daten relativ umständlich und nicht so intuitiv wie bei den anderen getesteten Programmen.

Der im FHI bereits in einer anderen Arbeitsgruppe vorhandene NAG Iris-Explorer erlaubt eine relativ einfache Programmierung durch Verknüpfung einzelner Module über Pipes, die die auftretenden Datenströme gut veranschaulichen. Er wurde zu Beginn des Projekts zur Visualisierung eingesetzt (siehe Zwischenbericht ViSpaS September 1999). Er bietet eine Fülle von Modulen, die interaktiv miteinander verknüpft werden können. Die Module werden als eigene Prozesse gestartet, und der Datentransfer läuft über UNIX-Sockets. Bei großen Datenmengen läuft das Programm merklich langsamer als Amira (siehe unten). Es kam des öfteren vor, dass einzelne Module abstürzten; der Iris-Explorer lief zwar weiter, aber Programmablauf und Datenfluss waren gestört. Die Maps (Netze von Modulen) können in der Skriptsprache Scheme gespeichert werden. Der Iris-Explorer bietet keine einfache Möglichkeit der Steuerung von einer anderen Anwendung aus.

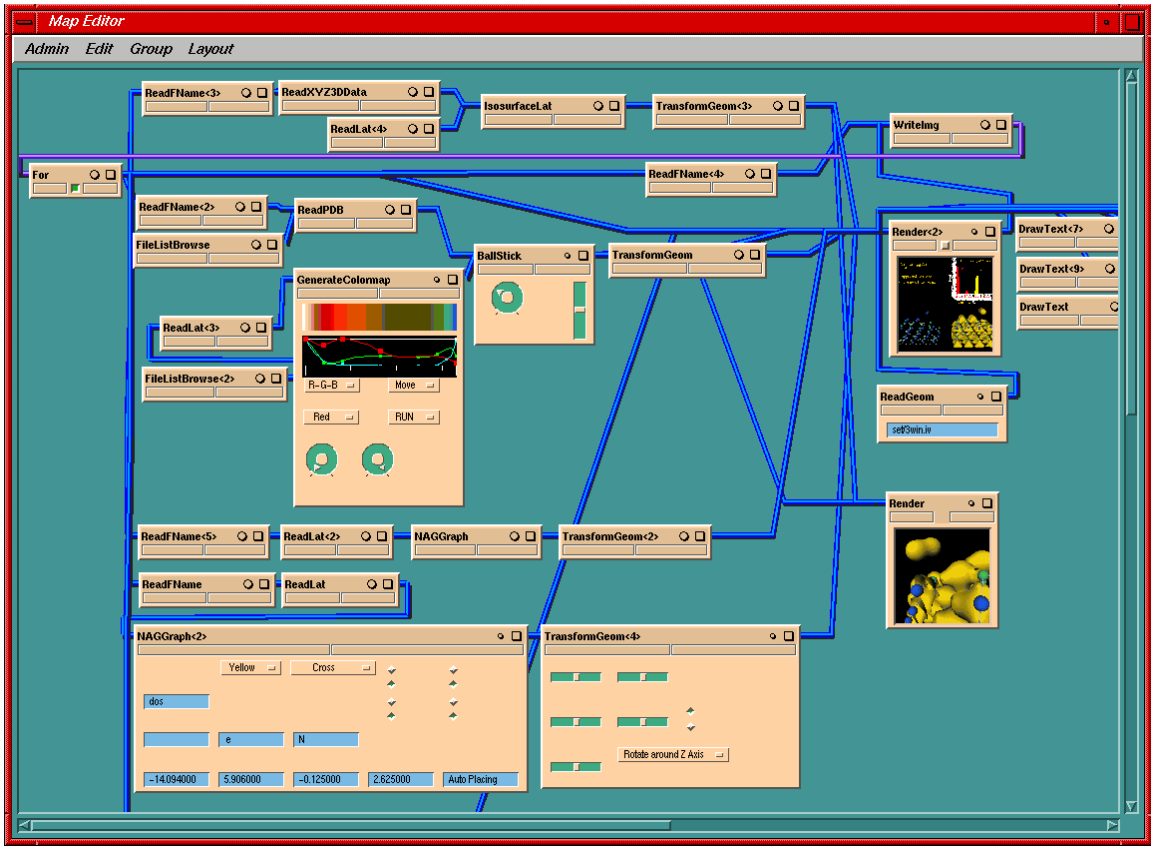


*Abb. 4: Screenshot Map Editor des Iris-Explorers: die einzelnen Module sind über Pipes miteinander verbunden*

Weitere Tests wurden mit der Visualisierungssoftware Amira durchgeführt. Amira basiert bei der Visualisierung wie auch der Iris-Explorer auf OpenInventor (einem objektorientierten 3D-Toolkit, das auf OpenGL aufsetzt) und ist bezüglich der Datenformate ähnlich. Es erwies sich als sehr fruchtbar für dieses Projekt, dass die Software auch für die Visualisierung und Interaktion in dem anderen DFN-Gigabit-Projekt TIKSL von ZIB und AEI benutzt wurde

und es einen guten Kontakt zu einigen Amira-Entwicklern am Konrad-Zuse-Institut Berlin gab.

Bei Amira werden Objekte miteinander verbunden ähnlich den Modulen beim Iris-Explorer. Im Unterschied zum Iris-Explorer findet der Datenfluss zwischen den Objekten im Arbeitsspeicher des Rechners statt, woraus eine deutlich bessere Performance und Stabilität resultieren. Die Objekte enthalten eine größere Funktionalität, was sich in diesem Fall in einem einfacheren Netzwerk der Objekte widerspiegelt (siehe Abb. 5 rechts oben).

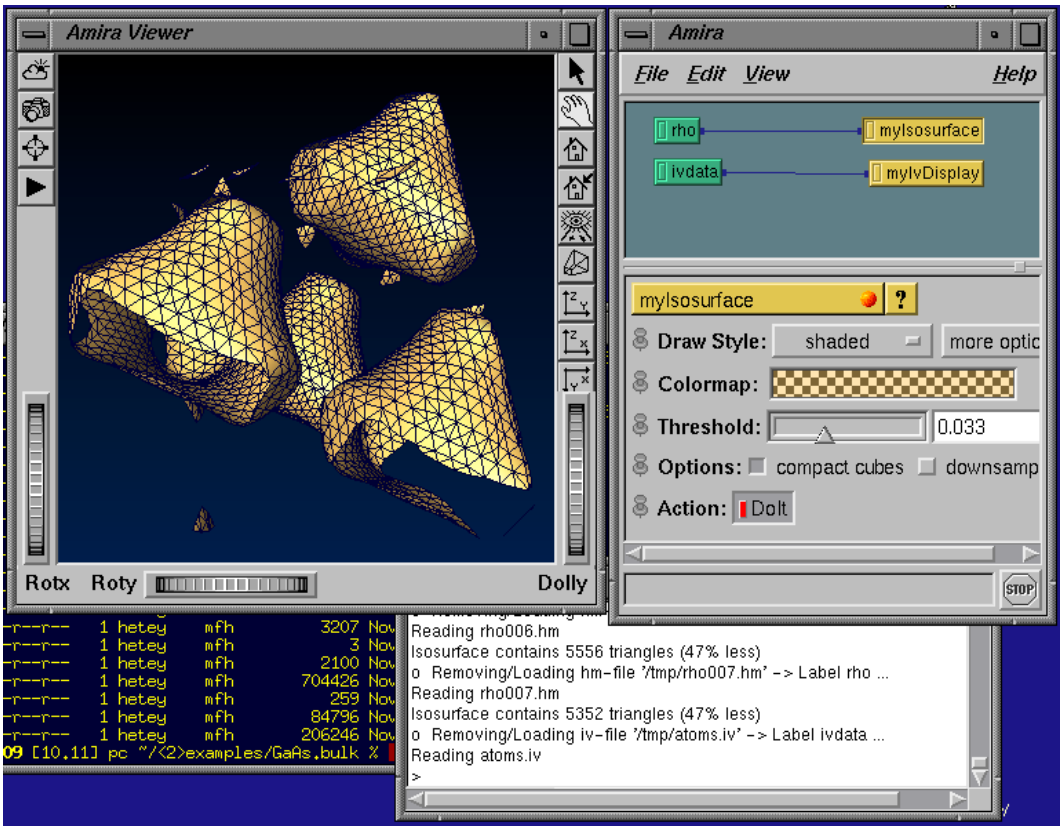


*Abb. 5: Screenshot Amira: links ist der 3D-Viewer, rechts oben der Object-Pool mit der Working Area und unten das tcl-Konsolenfenster; der 3D-Viewer stellt gerade die Elektronendichte von GaAs dar.*

Amira kann über die Skriptsprache tcl (*tool command language*) gesteuert werden. Alle Aktionen, die der Benutzer interaktiv ausführt, können auch im Konsolenfenster (Abb. 5 rechts unten) eingegeben bzw. in einer tcl-Funktion zusammengefasst werden. Man kann diese Funktionen im Konsolenfenster eingeben oder von einer anderen Anwendung aus aufrufen. Diese Möglichkeit wurde in diesem Projekt genutzt, um Amira von der Kontrollumgebung fernzusteuern.

Die Testergebnisse führten zur Auswahl der Software Amira zur Visualisierung innerhalb des Projekts.

### II.3.3 Initialisierung von Amira

Beim Starten von Amira über die Kontrollumgebung wird ein tcl-Skript geladen, in dem die Voreinstellungen der Amira-Objekte gespeichert wurden. Es werden ein Objekt zur Visualisierung von OpenInventor-Dateien sowie ein Objekt zur Visualisierung von Iso-

Oberflächen geladen. Durch den Amira-Befehl "**`app -listen`**" wird Amira in den Zustand versetzt, von außen tcl-Befehle empfangen zu können.

Es wurden für dieses Projekt eigene tcl-Funktionen entwickelt, die in dem geladenen tcl-Skript definiert sind und von der Kontrollumgebung aufgerufen werden können, zum Beispiel:

- loadReplaceRho
  Ein 3D-Datensatz im hypermesh-Format (Elektronendichte, Wellenfunktion, STM) wird geladen und ersetzt einen bereits zuvor geladenen. Aktualisierung des 3D-Viewer-Fensters.
- loadReplaceIv
  Ein 3D-Datensatz im OpenInventor-Format (Atome im *balls-and-sticks-model*) wird geladen und ersetzt einen bereits zuvor geladenen. Aktualisierung des 3D-Viewer-Fensters.
- takeSnapShot
  Der Bildinhalt des 3D-Viewer-Fensters wird als Datei im JPEG- oder TIFF-Format gespeichert. Ein Dateinamen-Zähler wird hochgezählt.

In weiteren Funktionen können Einstellungen von Amira oder Parameter der tcl-Funktionen geändert werden. Über das Setzen eines Flags kann bei jeder Aktualisierung des 3D-Viewers ein Bild gespeichert werden. Diese Bilder können dann anschließend mit anderen Programmen in eine Animation (MPEG-Movie oder animated GIF) für eine separate Präsentation umgewandelt werden.

## II.4 Programmentwicklung Netzkomponente

Im diesem Abschnitt werden die Funktionalität der grafischen Kontrollumgebung, deren Umsetzung sowie die Komponenten der Datenkommunikation beschrieben.

### II.4.1 Funktionalität der grafischen Kontrollumgebung

Im folgenden werden die Details der Funktionalität der grafischen Kontrollumgebung aufgelistet.

- **Starten / Beenden** von Datenserver, fhi98md, Visualisierungssoftware.
- **Atomkonfiguration**
  Die Konfiguration der Atome wird im *balls-and-sticks-model* angezeigt. Die Daten werden vom Datenserver im OpenInventor-Format gesendet.
- **Elektronische Ladungsdichte**

  a) manuelle Aktualisierung der Elektronendichte (und der Atomkonfiguration)

  b) automatische Aktualisierung der Elektronendichte. Es öffnet sich ein Fenster (*xterm*), in dem ein Skript läuft, welches mit Hilfe einer HTTP-HEAD-Anfrage auf das Vorhandensein einer neuen Elektronendichte überprüft und diese gegebenenfalls überträgt und zur Visualisierungssoftware weiterleitet.

  c) automatische Aktualisierung mit Interpolation. Wie bei der automatischen Aktualisierung, jedoch kann der Benutzer entscheiden, die Veränderung der Elektronendichte zwischen einer Aktualisierung und der darauf folgenden in einer vorgegebenen Zahl

von Teilschritten ablaufen zu lassen. Die für die Teilschritte benötigten Daten werden durch lineare Interpolation aus zwei aufeinanderfolgenden, vom Datenserver übermittelten Datensätze gewonnen.

- **Simulation von Rastertunnelmikroskopbildern** (gewichtete Summe der Betragsquadrate von mehreren Wellenfunktionen)
  Nach der Tersoff-Hamann-Theorie des Rastertunnelmikroskops ist der Tunnelstrom proportional zur Zustandsdichte innerhalb eines bestimmten Energieintervalls. In einem separaten Fenster können die Ober- und Untergrenze dieses Energieintervalls angegeben werden (siehe auch Beschreibung im Abschnitt II.4.1.1). Als Hilfestellung kann die Zustandsdichte als x-y-Diagramm dargestellt werden (siehe weiter unten). Die ausgewählten Parameter werden zum Datenserver gesandt. Der Datenserver bedient sich dann des Konvertierungsprogramms *contour*, um die Zustandsdichte in dem benötigten Energieintervall aus den gesamten bei der Simulation erzeugten Daten zu extrahieren. Darauf wandelt der Datenserver die Daten der Zustandsdichte in ein grafikfähiges Format um und übermittelt sie an die Visualisierungssoftware Amira. Der Benutzer kann mit Hilfe von Amira Ebenen (OrthoSlice, ObliqueSlice) durch den 3D-Datensatz legen, um ein zweidimensionales STM-Bild zu erhalten (siehe Abb. 6).

- **Wellenfunktion** eines Zustandes an einem Punkt des reziproken Raumes (komplexwertig); dargestellt wird jedoch eine reelle Größe.
  Es wird mit Hilfe der vom Datenserver übermittelten fhi98md-Ausgabe-Datei (fort.6) die Anzahl der in der Simulationsrechnung benutzten k-Punkte ermittelt. Daraus kann ein k-Punkt in einem separaten Fenster ausgewählt werden. Daraufhin werden die für den gewählten k-Punkt passenden Energiewerte in einer Auswahlbox angezeigt (Abb. 7 links). Wenn der Benutzer die gewünschten Daten ausgewählt hat, wird eine Anfrage an den Datenserver geschickt, der das Konvertierungsprogramm *contour* startet. Die zurückgesandten Daten werden an die Visualisierungssoftware weitergeleitet.

- **Zustandsdichte (DOS, *density of states*)**
  Es öffnet sich ein separates Fenster, in dem die nötigen Parameter eingestellt werden können (vgl. Abb. 7). Vom Datenserver wird eine fhi98md-Ausgabe-Datei (fort.6) geladen, die lokal weiterverarbeitet wird. Die Visualisierung der zweidimensionalen Funktion wird von den Programmen *xmgr* bzw. *xmgrace* übernommen (siehe Abb. 8)

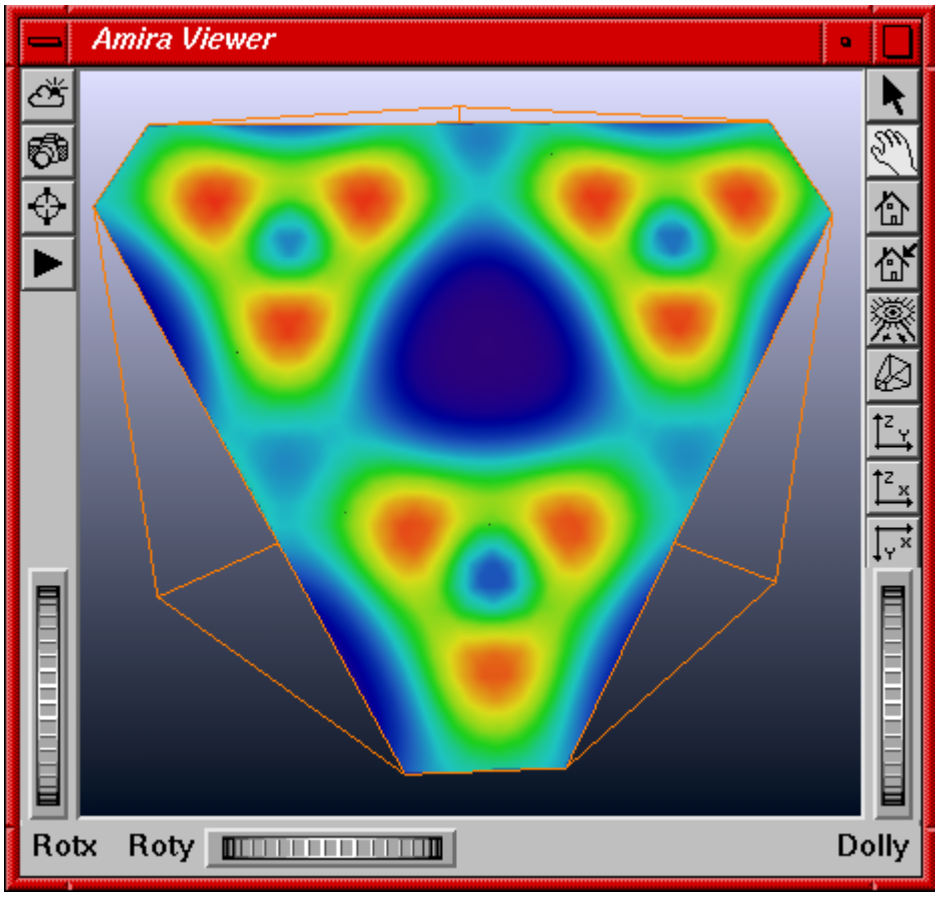


*Abb. 6: simuliertes Bild eines Rastertunnelmikroskopes*

### II.4.1.1 Visualisierung von Wellenfunktionen, STM-Bildern

Zur Visualisierung von Wellenfunktionen bzw. zum Generieren von STM-Bildern (STM: *Scanning Tunneling Microscope* - Rastertunnelmikroskop) wird das Konvertierungsprogramm *contour* benutzt, das in der Abteilung Theorie des FHI bereits existierte und für dieses Projekt angepasst und erweitert wurde. So wurde ein Perl-Skript geschrieben, das die Eingabe-Daten für die gewünschte Konvertierung aus der fhi98md-Ausgabe-Datei fort.6 sowie den vom Datenserver übermittelten Parametern generiert. Das Konvertierungsprogramm läuft auf der Cray, da die Wellenfunktionen aus der binär geschriebenen Ausgabe-Datei fort.71 des fhi98md-Programms eingelesen werden müssen.

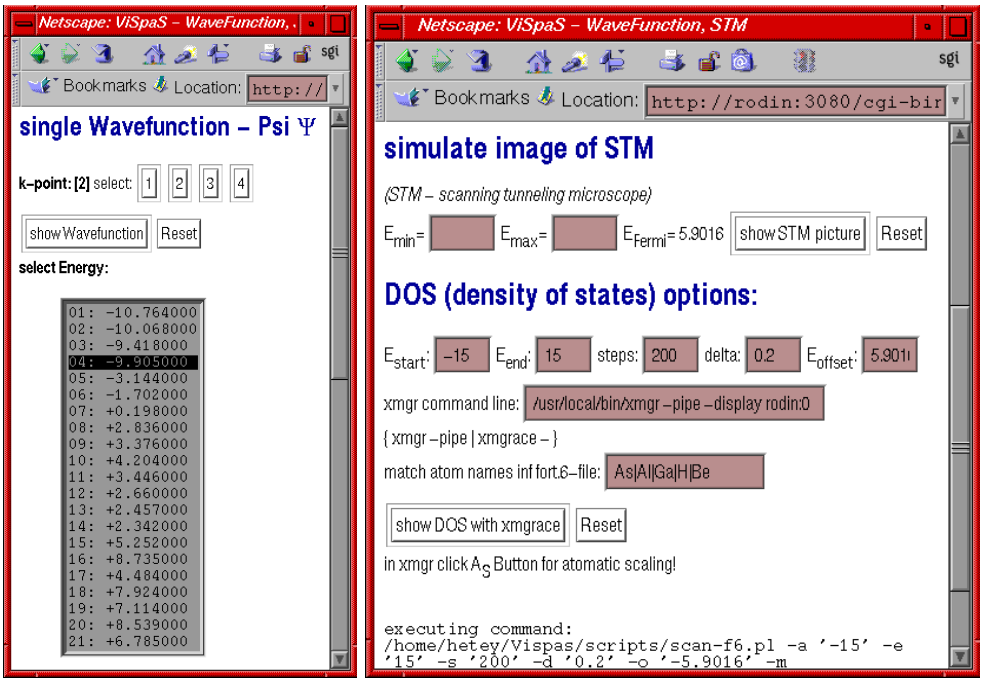


*Abb. 7: Fenster zum Auswählen der Optionen zum Visualisieren von Wellenfunktion, STM-Bild und Zustandsdichte*

Von der Kontrollumgebung kann mit dem Knopf "wavefunction, STM, DOS" (vgl. Abb. 12) ein weiteres Fenster geöffnet werden, das oben die Einstellungen für die Visualisierung einer Wellenfunktion (Abb. 7 links) und unten die Einstellungen zum Erzeugen eines STM-Bildes bzw. dem Anzeigen der Zustandsdichte beinhaltet (Abb. 7 rechts).

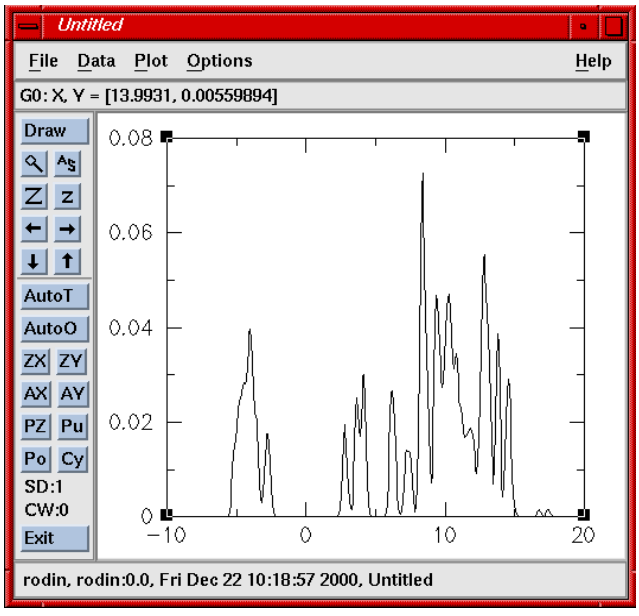


*Abb. 8: Zustandsdichte (DOS) visualisiert mit dem Programm xmgr*

Zum Visualisieren der Wellenfunktion und des STM-Bildes wird das oben erwähnte Programm *contour* benutzt, das mit den übermittelten Parametern vom Datenserver aufgerufen wird. Die Zustandsdichte wird aus einer zuvor übermittelten Ausgabe-Datei lokal

erzeugt und mit einer zusätzlichen Visualisierungssoftware angezeigt, wobei hier wahlweise *xmgr* oder *xmgrace* benutzt werden kann.

## II.4.2 Analyse der Datenkommunikation

Zur Kommunikation wurde eine Zusatzschicht, bestehend aus dem Datenserver und einer grafischen Benutzeroberfläche, der Kontrollumgebung, hinzugefügt, die für die Kommunikation zwischen fhi98md-Programm und Visualisierungssoftware zuständig ist (vgl. Abb. 9).

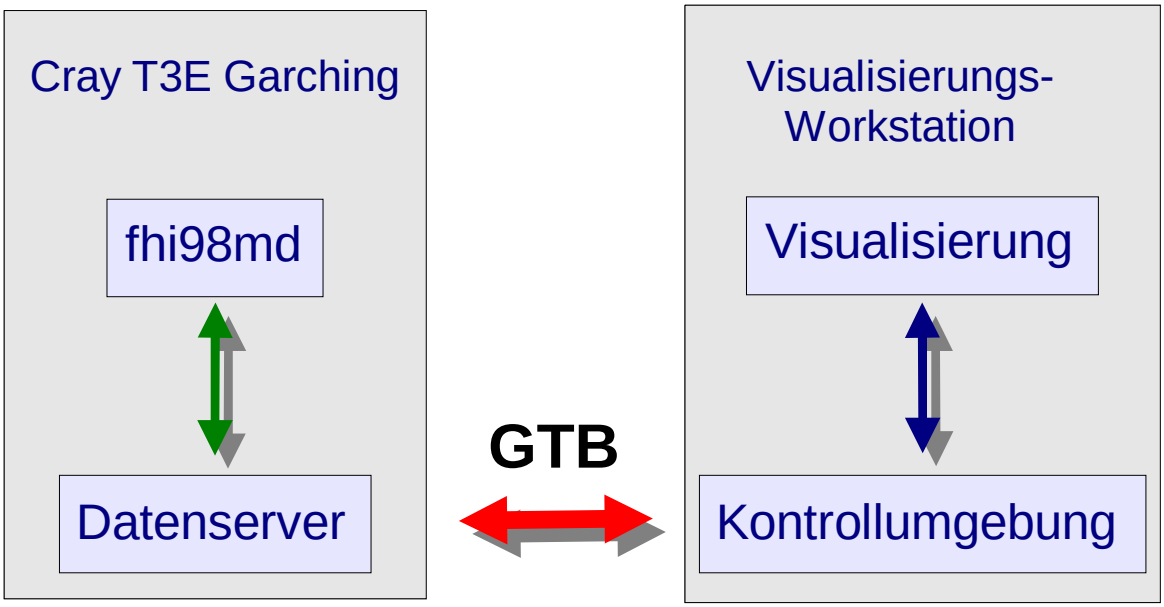


*Abb. 9: Kommunikation und Datenfluss*

Die Kontrollumgebung ist eine grafische Benutzerschnittstelle in Form eines Web-Browsers, hinter der sich eine Vielzahl von Funktionen zur Steuerung verschiedener Komponenten verbirgt. Sie soll es dem Benutzer ermöglichen, die Parameter für die laufende fhi98md-Rechnung und den Datenserver einzustellen. Im weiteren kann man hiermit verschiedene Modi der Visualisierungssoftware mit verschiedenen Parametern aktivieren.

Der Datenserver konvertiert die vom fhi98md-Programm generierten Daten und übernimmt die Kommunikation mit der Visualisierungs-Workstation in Berlin. Die Kommunikation mit der Kontrollumgebung findet auf Socket-Basis zwischen bestimmten Ports über das Gigabit-Testbed statt. Zur Kommunikation wurde ein Protokoll entwickelt, welches das Auslesen und Schreiben der Daten und Statusinformationen ermöglicht (siehe Abschnitt II.4.3).

## II.4.3 Datenserver und Kommunikationsprotokoll

Zur Kommunikation zwischen dem Datenserver und der Kontrollumgebung wurde ein einfaches Protokoll entwickelt, das in Anlehnung an das *Hypertext Transfer Protocol* (HTTP) einen einfachen Zugriff auf Daten und Meta-Information ermöglicht. Dies erwies sich als sehr praktisch, da man beim Entwickeln von Server und Client jeweils auf bereits funktionierende HTTP-Komponenten zurückgreifen konnte.

Anfangs war eine enge Kopplung zwischen Datenserver und fhi98md-Programm geplant (siehe ViSpaS Zwischenbericht März 2000). Dies erwies sich jedoch wegen der starken Beeinträchtigung der Stabilität des fhi98md-Programms durch das Netz als nicht praktikabel. Eine genauere Beschreibung hierzu findet man im Abschnitt II.5.

Der Datenserver kommuniziert mit der Kontrollumgebung über einen Internet-Socket. Bei dem hier vorgestellten Protokoll handelt es sich um eine Untermenge von HTTP, die an die Anwendung angepasst wurde. Die einzelnen Elemente des Protokolls werden im folgenden näher erläutert:

- **PING**
  einfache Anfrage, ob der Datenserver läuft und auf entsprechendem Port 'lauscht'
- **GET *filename***
  Übertragung einer Datei *filename*
- **HEAD *filename***
  Übertragung der Meta-Informationen wie Datum, Länge ... der Datei *filename*
- **GET status** {ds, fhimd}
  Anfrage, die genauere Informationen über den Status des fhi98md-Programms bzw. den Datenserver liefert
- **GET data** {rho, wavefunct, atoms, stm} ?format=
  Anfrage nach Daten der Elektronendichte, Wellenfunktion ...
  optionale Parameter: Format
  Der Datenserver startet das Konvertierungsprogramm *contour* mit den übermittelten Parametern.

## II.4.4 Grafische Kontrollumgebung

Die Kontrollumgebung ist eine grafische Oberfläche, die es dem Benutzer ermöglicht, ausgewählte Funktionen einfach zu erreichen und den administrativen Aufwand der Datenkommunikation zu verbergen. Die grafische Oberfläche wird mit Hilfe eines Web-Browsers (z.B. Netscape) realisiert, der über das *Common Gateway Interface* (CGI) eines lokalen Web-Servers mit einem Perl-Skript kommuniziert. Hierbei wurde der Web-Server xitami von der Firma Imatix eingesetzt, ein Freeware-Programm, das es für alle gängigen Betriebssysteme gibt und das im Gegensatz zu der Software *apache* eine Browser-Basierte Administration beinhaltet, die die Installation stark vereinfacht. Dies hat sich bislang als ausreichend flexibel und portabel erwiesen.

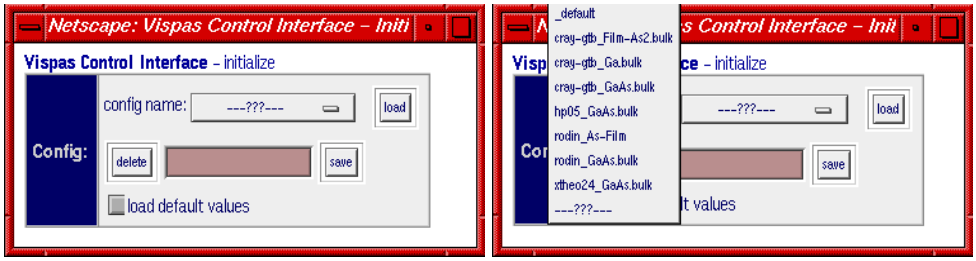


*Abb. 10: Initialisierung der Kontrollumgebung rechts mit Auswahl der gespeicherten Sitzungen*

Zu Beginn einer Sitzung wird die Kontrollumgebung initialisiert. Hier können IP-Adressen, Port-Nummer, Pfade etc. eingestellt bzw. angepasst werden. Diese Einstellungen ändern sich im allgemeinen während einer Sitzung nicht. Um die Einträge von verschiedenen Projekten einfach zu verwalten, kann man die Parameter unter entsprechendem Namen speichern bzw. laden (vgl. Abb. 10).

Nach dem Laden der Projektdaten werden diese in einer html-Form angezeigt, die ein einfaches Editieren ermöglicht. Dabei werden die Variablen aufgrund der Struktur ihrer Namen in der Tabelle thematisch geordnet (vgl. Abb. 11).

Durch das Drücken des "Initialize"-Knopfes gelangt man zur initialisierten Kontrollumgebung, welche aus drei Teilen (html-Frames) besteht, die neben- bzw. übereinander sichtbar sind (siehe Abb. 12).

Im oberen Teil des Command-Fensters (oberer html-Rahmen der Kontrollumgebung in Abb. 12) können die Visualisierungssoftware und der Datenserver gestartet oder beendet werden. Weiter unten können die darzustellenden physikalischen Größen ausgewählt und spezifiziert werden. Das Interface kann einfach um zusätzliche Funktionen erweitert werden. Eine Auflistung der gewünschten Visualisierungsmodi findet sich weiter oben im Abschnitt II.4.1 "Funktionalität der grafischen Kontrollumgebung".

Wurden vom Benutzer die gewünschten Parameter eingestellt und ein entsprechender Button geklickt (z.B. "show atoms" in Abb. 12), so schickt die Kontrollumgebung entsprechende Anfragen an den Datenserver und leitet die erhaltenen Daten an die Visualisierungssoftware weiter. Die Standard- und Fehler-Ausgabe dieser Aktionen sind im Log-Fenster zu sehen (Abb. 12 rechts unten) und für eine Fehleranalyse notwendig.

Eine einfache und schnelle Sicht auf den Status des Datenservers sowie der Visualisierungssoftware gibt das Status-Fenster (links unten in Abb. 12). Hier findet man eine kleine Tabelle mit 3 Icons für *vi* (Visualisierung), *ds* (Datenserver), *md* (fhi98md-Molekulardynamik-Programm). Leuchtet der Hintergrund eines Icons grün (und ist die Schrift fett und groß), so läuft die entsprechende Komponente, ansonsten ist der Hintergrund grau. In Abb. 12 laufen gerade der Datenserver, die Visualisierungssoftware sowie das fhi98md-Programm.

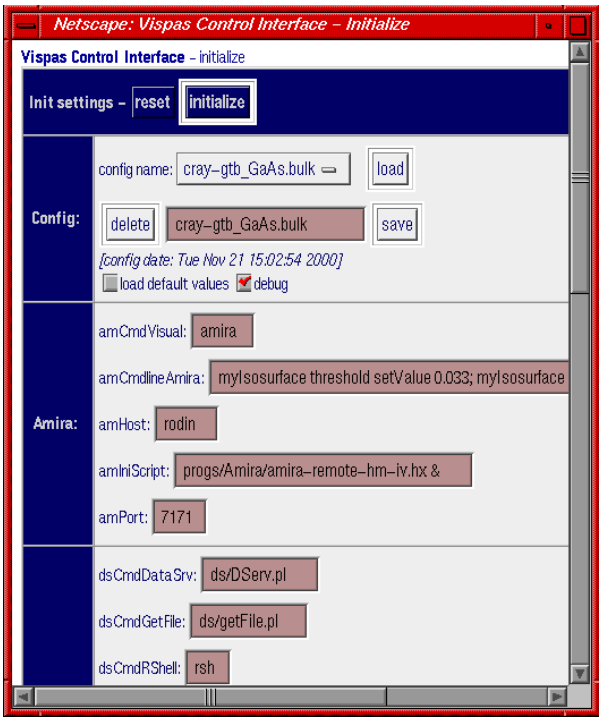


*Abb. 11: Initialisierungsformular der Kontrollumgebung*

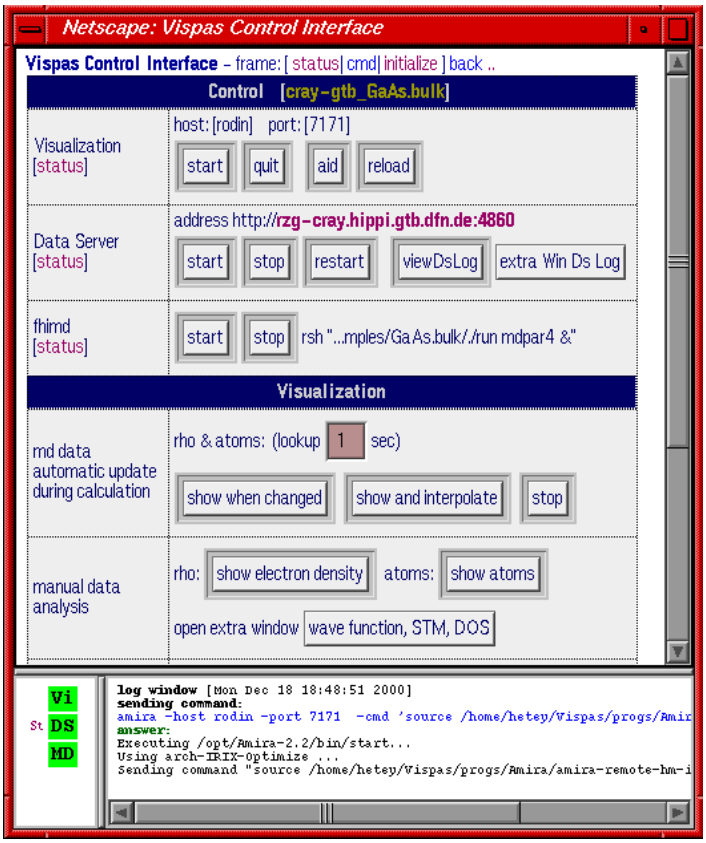


*Abb. 12: Kontrollumgebung mit Command-Fenster (oben), Statusanzeige (links unten) und Log-Fenster (rechts unten)*

# II.5 Netz-Durchsatzmessungen

Für die Entscheidung der Kommunikationsstrategie, die Daten direkt über Sockets oder asynchron mit Hilfe eines Datenservers zu senden, war es wichtig, verschiedene Messungen unter realistischen Betriebsbedingungen durchzuführen. Hier soll der Versuch beschrieben werden, große Datenmengen aus dem fhi98md-Programm auf der Cray in Garching direkt zur Visualisierungs-Workstation in Berlin zu senden. Dazu standen hier Optimierungen hinsichtlich des maximalen Datenduchsatzes im Vordergrund. Die weiter oben besprochenen Problemstellungen der Stabilität und Flexibilität der Software wurden dabei ausgeklammert.

## II.5.1 Voraussetzungen

Die Gigabit-Testbed-Verbindung vom Rechenzentrum Garching (Rechner cray T3E) zum FHI Berlin (Visualisierungs-Workstation Berlin VWB rodin) wurde im Modus CLIP (*classical IP over ATM*) betrieben. Durch den ATM- und IP- Overhead ergibt sich für den 622 MBit-Adapter ein maximaler Durchsatzwert von 538,053 MBit/sec [Abschlussbericht Testbed West, Bericht GIGAnet ;
http://webdoc.sub.gwdg.de/ebook/ah/1999/dfn/GTBWest.pdf].

Für Messungen des Durchsatzes wurde das parallelisierte Programm fhi98md so verändert, dass während einer Rechnung unterschiedlich große Datenmengen über das Gigabit-Testbed geschickt werden können, die auf der anderen Seite (rodin) von einem Programm empfangen werden. Die für die Socket-Kommunikation zuständigen Funktionen sind C-Routinen, die vom fhi98md-Programm aufgerufen werden.

Folgende Parameter haben sich hinsichtlich des maximalen Durchsatzes als optimal erwiesen:

- MTU = 65280 Byte (*maximum transmission unit*) des Fore-ATM-Interfaces; Diese Puffergröße kann auf dem ATM-Interface bis maximal 65535 Byte groß werden [FORE-Runner User's Manual 5.1.6], die benutzte MTU wurde vom HiPPI-Switch vorgegeben.
- TCP-Buffersize = 2097152 Byte - Puffergröße beim Initialisieren der Socketverbindung innerhalb einer C-Routine (getestet wurden 524288 - 4194304 Byte) auf cray und rodin.

Das Programm fhi98md wurde für die hier beschriebenen Messungen des Durchsatzes mit Hilfe des Queueing-Systems (*small queue*) der cray T3E gestartet und lief parallel auf 16 Prozessoren.

Das Einsammeln der verteilten Daten zum dreidimensionalen Datensatz der Elektronen-dichte von den einzelnen Prozessoren erfolgte mit der shmemget-Routine an der Cray vor dem Senden. Die Kommunikation über die Socket-Verbindung übernahm ein ausgewählter Prozessor (*master*), um zu gewährleisten, dass die Durchsatzrate unabhängig von der Anzahl der Prozessoren ist.

## II.5.2 Ergebnisse

### *II.5.2.1 Durchsatzmessungen*

Für die Bestimmung des Durchsatzes wurde vor und nach dem Aufrufen der socket-write-Routine die Zeit mit Hilfe der *gettimeofday*-Routine ausgegeben. Zunächst wird eine "normale" Rechnung ausführlicher erläutert. Im Anschluss werden verschiedene typische Abweichungen besprochen. Als Beispiel wurde hier eine GaAs-Struktur mit 8 Atomen berechnet.

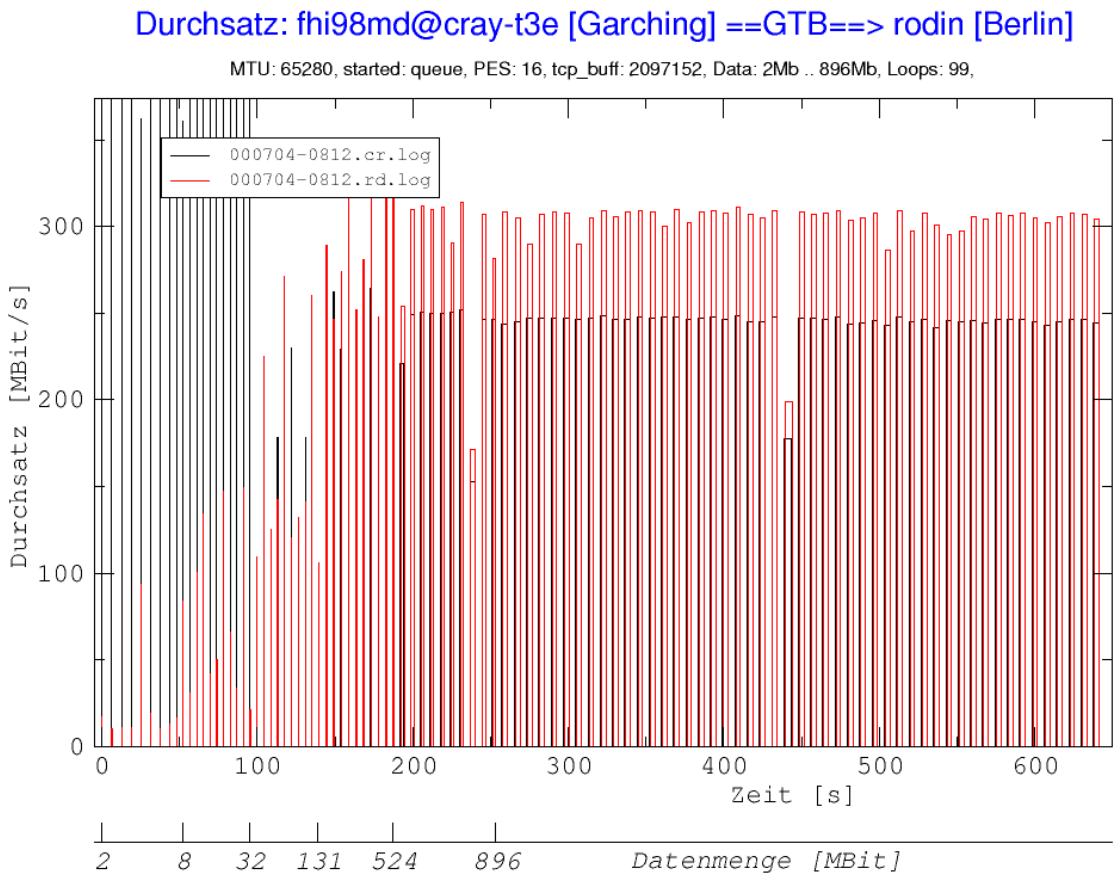


*Abb. 13: Durchsatzraten während einer Rechnung. Der Durchsatz auf der cray ist in rot dargestellt (gestrichelt in schwarz-weiß Abbildung), der auf der VWB rodin in schwarz. Nach jeder Iteration des fhi98md-Programms wird eine neue Elektronendichte berechnet und über das Gigabit-Testbed gesendet. Die Durchsatzrate während des Sendens ist als Balken dargestellt. Die Größe des übermittelten Datenfeldes wurde nach jeweils 10 Iterationen vervierfacht, beginnend mit 2 MBit (entspricht einem Datenfeld von 32000 64-Bit-Realzahlen) bis zu einer maximalen Größe von 896 MBit (14 Mio Realzahlen), wie in der unteren x-Achse zu sehen ist.*

In Abb. 13 sieht man den Durchsatz der übermittelten Daten während einer Rechnung bei verschieden großen Datenfeldern. Auf der rodin geht die Durchsatzrate (rote Linien) bei einer Datenfeld-Größe von 131 MBit (ab der 31. Iteration, entsprechend 150 sec in Abb. 13) in eine Sättigung von etwa 300 MBit/sec (siehe auch Abb. 14), während auf der cray ein Wert von etwa 250 MBit/sec erreicht wird.

Das Übersenden eines Datenfeldes ist in der Grafik als ein Rechteck dargestellt, deren Flanken durch die Zeitmessungen vor und nach dem Senden bzw. Lesen bestimmt werden. Der Schreibvorgang nimmt insgesamt etwas mehr Zeit in Anspruch, was sich in einer größeren Breite und daher einer geringeren Höhe ausdrückt; die Datenmenge (Fläche des Rechtecks) bleibt gleich. Beispielswiese haben das rote und schwarze Rechteck in Abb. 13 bei 400 Sekunden eine Zeitdifferenz von 3,64 s - 2,91 s = 0,73 s. Für diese Differenz der Durchsatzraten sind wahrscheinlich Pufferungen in den Rechnern selbst sowie auf der Strecke verantwortlich.

Die hohen Durchsatzwerte bei kleinen Datenpaketen auf der cray (schwarze Linien bis 100 sec in Abb. 3) sind auf Fehler der *gettimeofday*-Routine durch die begrenzte zeitliche Auflösung des Rechners zurückzuführen.

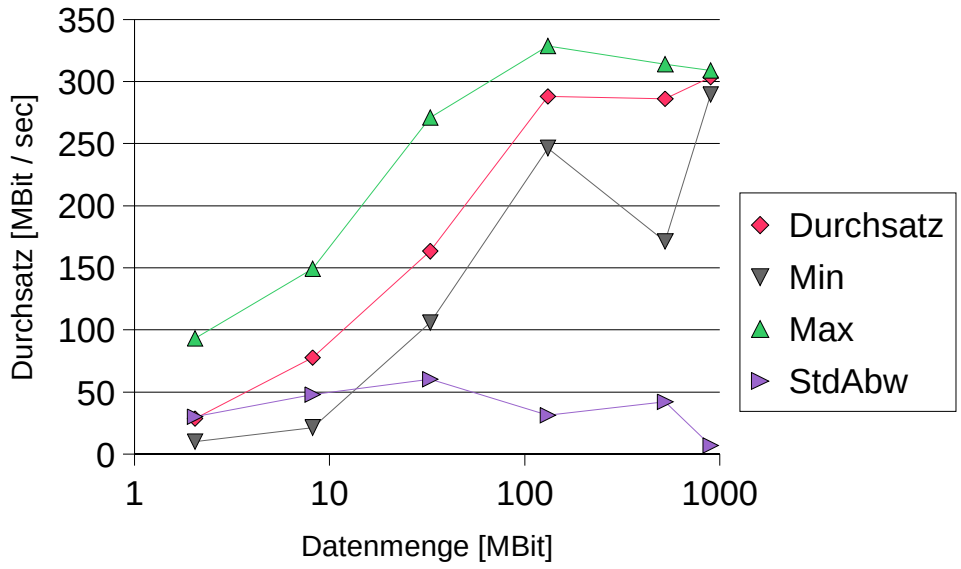


*Abb. 14: Durchsatzrate in Abhängigkeit von der gesandten Datenmenge auf der VWB rodin. Man erkennt gut das Ansteigen des Durchsatzes bei steigender Datenmenge und das Erreichen der Sättigung bei einer Feldgröße von 130 MBit.*

Abb. 15 zeigt die Übertragungsdauer verschieden großer Datenfelder. Die gerade Linie (schwarz) zeigt den linearen Verlauf der theoretischen Durchsatzrate von 538 MBit/sec (siehe oben). Man erkennt, dass mit den oben genannten Parametern größere Datenpakete effizienter übertragen werden, wobei die Durchsatzrate bei den beiden größten Datenmengen noch um mehr als 200 Mbit/sec vom theoretischen Wert abweicht.

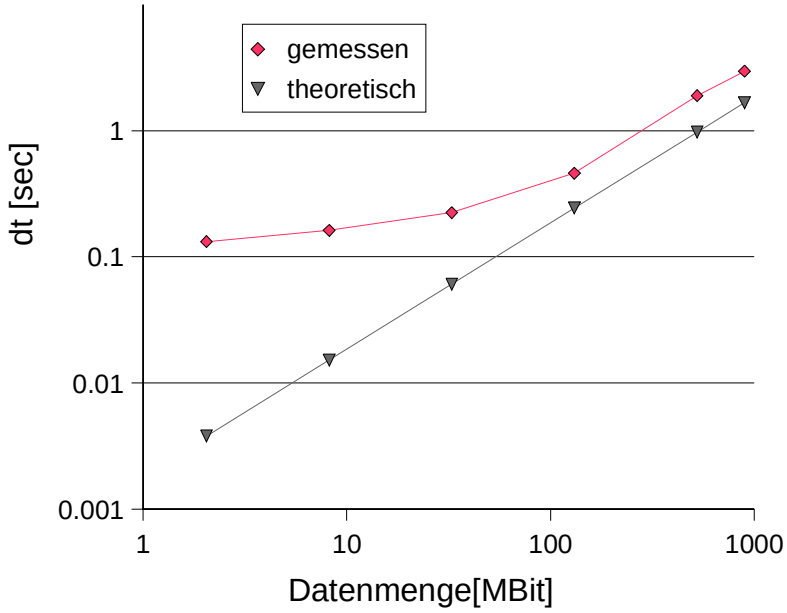


*Abb. 15: Zeit für das Senden eines Datenpaketes (dt) in Abhängigkeit von der Größe der Datenmenge*

### II.5.2.2 Äußere Einflüsse

Die Größe des Durchsatzes hängt von vielen Faktoren ab. Nachfolgend sind fünf Beispiele für unterschiedliche, aber typische Verhaltensmuster des Durchsatzes während einer Messung bei ansonsten gleichen Parametern (16 Prozessoren, Datenmenge 2 - 896 MBit, 99 Iterationen, MTU 65280, gemessen auf der VWB) aufgeführt.

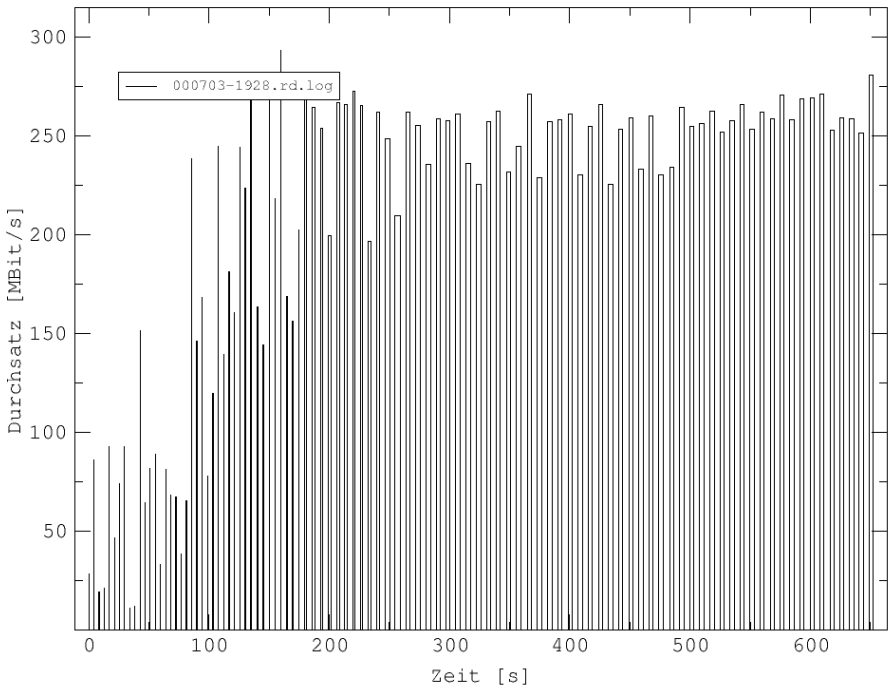


*Abb. 16: Verglichen mit der Messung in Abb. 13 sind hier deutliche zeitliche Schwankungen zu erkennen.*

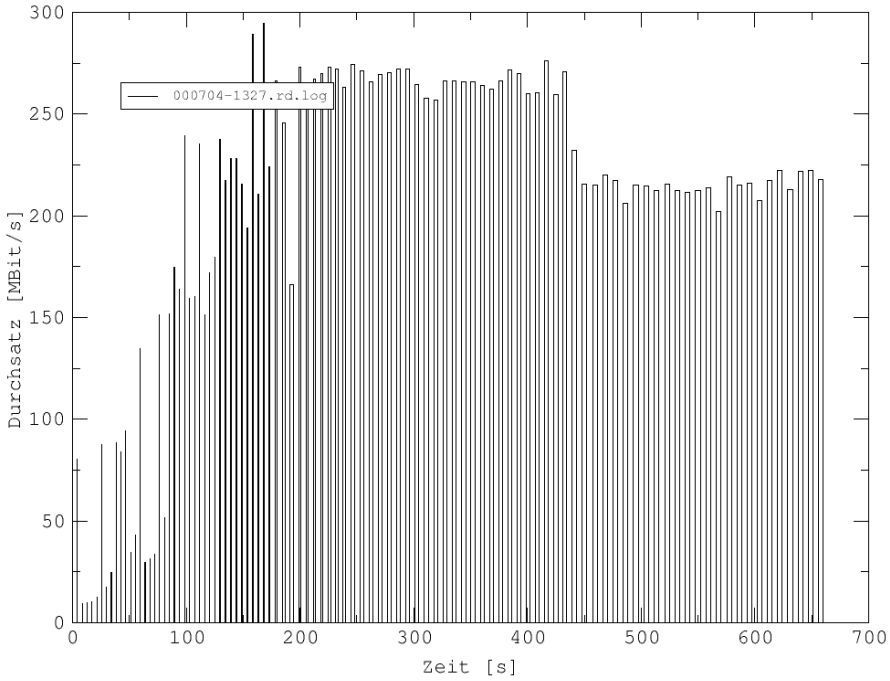


*Abb. 17: Die Durchsatzrate liegt relativ konstant bei 260 MBit/sec und sinkt dann bei 430 sec auf einen relativ konstanten Wert von 215 MBit/sec.*

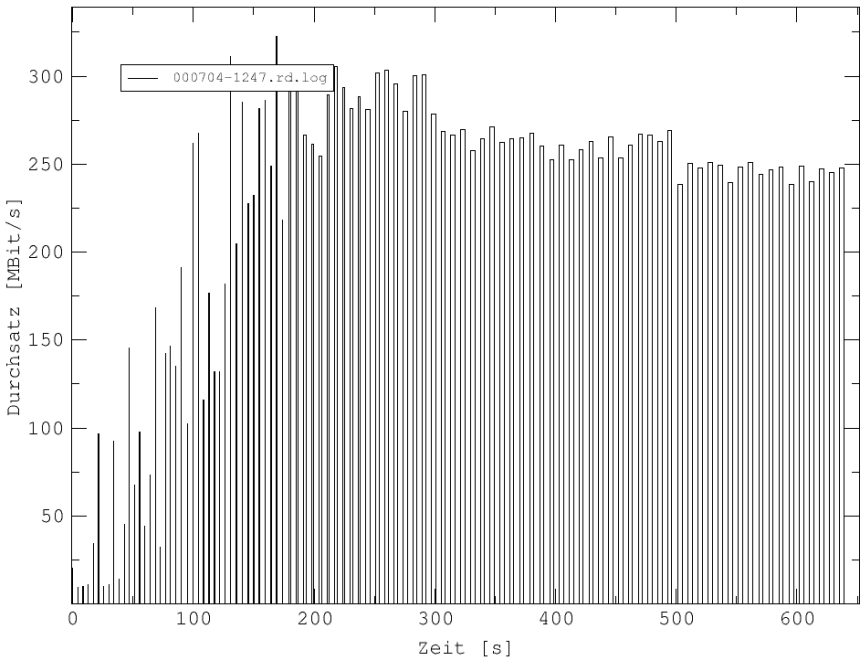


*Abb. 18: Die Durchsatzrate sinkt relativ gleichmäßig von 300 auf 240 MBit/sec.*

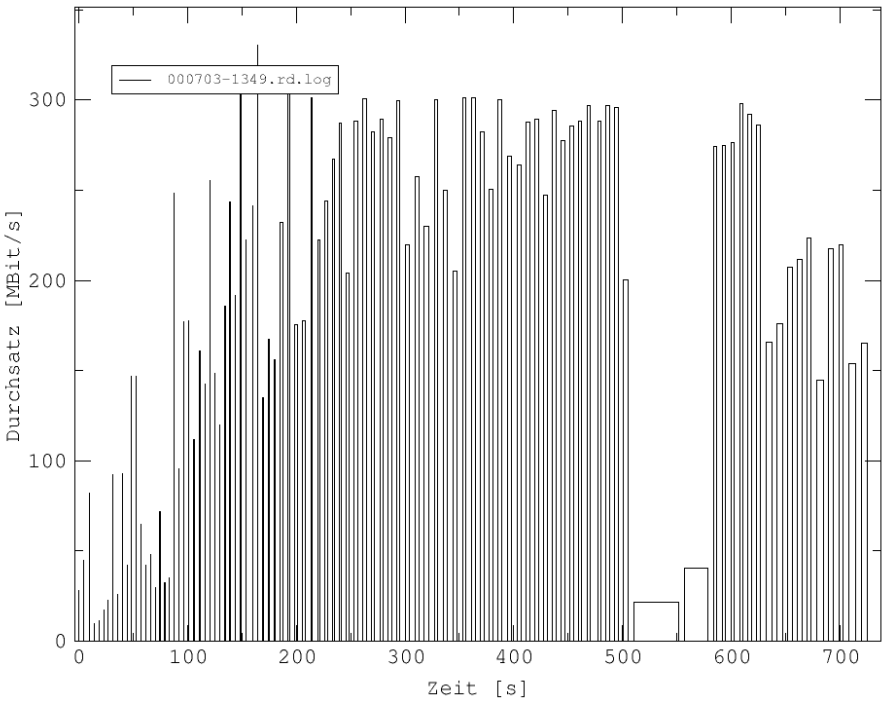


*Abb. 19: Die Durchsatzrate zeigt einen starken Einbruch bei 500 sec.*

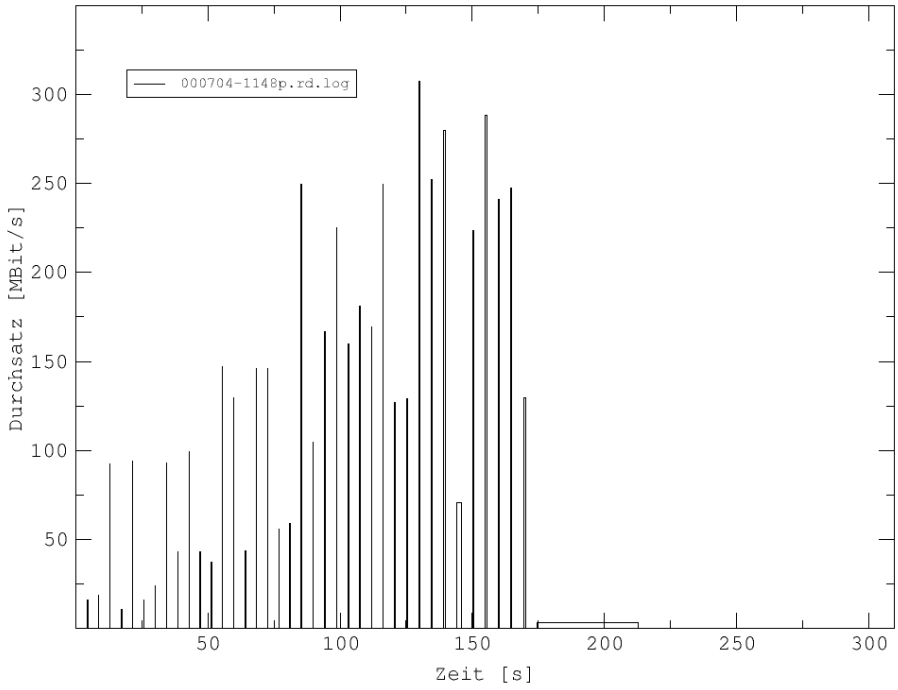


*Abb. 20: Es ist die Durchsatzrate während einer Rechnung zu sehen, die nach 40 Iterationen nicht reproduzierbar abbrach.*

### II.5.3 Fazit Durchsatzmessung

Die Messungen der Durchsatzraten zeigen, dass es möglich ist, Daten direkt vom fhi98md-Programm über das Gigabit-Testbed zur Visualisierung an die Grafik-Workstation zu senden, wobei ein Durchsatz von 300 MBit/sec erreicht werden konnte. Diese Durchsatzrate entspricht 0.3 - 10 Datensätzen / sec und erscheint für den interaktiven Betrieb akzeptabel.

Die starken äußeren Einflüsse, insbesondere die häufigen, vom Netz verursachten Abstürze des fhi98md-Programms, machen diese Art der Datenübertragung jedoch nicht praktikabel, da ein unvorhergesehener Abbruch des Programms nicht akzeptabel ist. Es erwies sich daher als zwingend, die zu visualisierenden Daten zu puffern und mit einem separaten Prozess / separaten Programm (siehe Abschnitt II.4.3) über das Gigabit-Testbed zu übermitteln.

# III Zusammenfassung, Ausblick

Im Projekt konnten wesentliche Teile der Zielsetzung erreicht werden. Es wurde eine Arbeitsumgebung geschaffen, die eine effiziente Analyse der Ergebnisse einer Molekulardynamik-Simulationsrechnung auf einem entfernten Großrechner ermöglicht. Die bei einer Online-Visualisierung anfallenden Datenmengen wurden näher untersucht und es zeigte sich, dass die über das Gigabit-Testbed bereitgestellte Übertragungskapazität ausreichend ist.

Eine Erweiterung der Arbeitsumgebung durch zusätzliche Komponenten ist möglich. So wären eine visuelle Hilfe beim "Aufsetzen" von Rechnungen und eine Kontrolle der Rechenaufträge denkbar.

Das Portieren der entwickelten Softwarekomponenten auf eine andere Plattform ist im Prinzip möglich. Die einzelnen Komponenten sind nicht auf ein bestimmtes Betriebssystem begrenzt, sie müssten gegebenenfalls an die lokalen Bibliotheken angepasst werden.

Der Einsatz der Software unter geänderten Netzbedingungen im Zusammenhang mit dem Übergang zum G-WiN muss gesondert untersucht werden. Da das Kommunikationskonzept auf dem Internet-Protokoll (IP) basiert, sind bei einer Umstellung keine größeren Schwierigkeiten zu erwarten.

# IV Anhang

## IV.1 Verwendete Programme

Die eingesetzten Programme unterteilen sich in zwei Gruppen. Der eine Teil wird auf der lokalen Workstation eingesetzt (VWB), wo auch die Visualisierung stattfindet, in diesem Fall die SGI Octane. Der andere Teil befindet sich auf dem Rechner, wo das Molekulardynamik-Programm laufen soll, hier die Cray T3E.

### IV.1.1 Programme Steuer- und Visualisierungsumgebung (VWB)

Die hier aufgeführte Software wird zur grafischen Analyse bzw. für die Kontrollumgebung auf der Visual Workstation benötigt.

- Web-Browser Netscape 4.x (mit Unterstützung von Frames und JavaScript)
  Der Web-Browser dient zur Anzeige der grafischen Kontrollumgebung. Hier werden die vom CGI-Skript generierten html-Seiten angezeigt. JavaScript wird zum Öffnen von Fenstern verwendet, ist aber nicht unbedingt notwendig.
- Web-Server mit Common Gateway Interface (CGI)
  Der Web-Server stellt die Schnittstelle zwischen dem CGI-Skript und dem Web-Browser dar. Die im Browser ausgewählten Funktionen werden vom CGI-Skript vispasGUI.cgi ausgeführt.
- Perl
  Die Skriptsprache Perl ist mittlerweile Bestandteil bei allen Unix-Derivaten und ist für alle Betriebssysteme verfügbar. Benötigt wird ein Perl in der Version 5.x (benutzt wurde 5.003_02) mit dem Modul CGI.pm (standardmäßig enthalten).
- Perl-Skripte
  - vispasGUI.cgi
    CGI-Skript, das die html-Ausgabe für die Kontrollumgebung erzeugt und die ausgewählten Funktionen aufruft bzw. weiterleitet.
  - scanf6.pl
    Skript zum Parsen der fhi98md-Ausgabe-Datei fort.6 und zur Extraktion von Energieeigenwerten (für DOS), Atomkoordinaten etc.
  - atomconv.pl
    Skript zur Umwandlung der Atomkoordinaten, die mit scan-f6.pl extrahiert wurden, in andere 3D-Formate (PDB => OpenInventor, VRML) umgewandelt. Hiermit werden die Atomkonfigurationen erzeugt.
  - getFile.pl
    Programm zur Kommunikation mit dem Datenserver - Senden der Anfragen und Download der Daten via HTTP.
- Amira
  Visualisierungsprogramm mit verschiedenen Modulen, unter anderem zur Visualisierung von 3D-Datensätzen, OpenInventor-Dateien, mit integriertem tcl steuerbar.

- amira-remote-hm-iv-hx
  tcl-Skript mit Konfigurationsinformationen für Amira sowie tcl-Funktionen, um Amira-Funktionen zusammenzufassen und von außen zu steuern.
- C-Programme
  - ipolhm
    Interpolation zwischen 2 hypermesh-Dateien (3D-Daten-Files) zur Film-Animation in beliebiger Anzahl von Schritten. Nach jedem Interpolationsschritt werden die Daten der Visualisierung übergeben.
  - getfile
    Programm zur Kommunikation mit dem Datenserver. C-Version von getFile.pl (siehe oben)

## IV.1.2 Programme Numerik-Rechner CRAY

Die hier aufgeführte Software wird auf dem Rechner benutzt, auf dem das Programm fhi98md läuft.

- fhi98md (parallelisierte Version)
  *Ab-initio* Dichtefunktional-Theorie-Programm zur Berechnung komplexer quanten-chemischer Systeme unter Verwendung von Pseudopotentialen in einer Ebenen-Wellen-Basis.
- Perl
  siehe oben
- Perl-Skripte
  - DServ.pl
    Datenserver zum Aufbereiten, Konvertieren und Ausliefern der fhi98md-Daten; Kommunikation über ein HTTP-ähnliches Protokoll
  - atomconv.pl
    siehe oben
  - mkContourInp.pl
    Skript zum Erzeugen einer Eingabe-Datei für das Konvertierungsprogramm contour (siehe unten); Eingabe fhi98md-Ausgabe-Datei fort.6 und optionale Parameter.
- Fortran-Programme
  - contour
    Konvertierungsprogramm zur Ausgabe der Wellenfunktion, der Elektronendichte und zur Simulation von STM-Bildern. *contour* liest fhi98md-Ausgabe-Datei fort.71 und eine Konfigurations-Datei ein (erzeugt mit mkContourInp.pl).
  - convrho
    Konvertierungsprogramm zur Ausgabe der Elektronendichte aus der fhi98md-Ausgabe-Datei fort.72.

## IV.2 Abkürzungen

- AEI - Max-Planck-Institut für Gravitationsphysik Albert-Einstein-Institut Potsdam
- ATM - Asynchronous Transfer Mode
- B-WiN, G-WiN - Breitband bzw. Gigabit-Wissenschaftsnetz
- CGI - Common Gateway Interface
- CLIP - Classical IP over ATM
- DOS - Density of States (Zustandsdichte)
- FHI – Fritz-Haber-Institut Berlin
- GRZ - Gemeinsames Rechenzentrum der Berliner Max-Planck-Institute
- GTB - Gigabit Testbed
- GUI - Graphical User Interface
- HiPPI - High Performance Parallel Interface
- LAN - Local Area Network
- LZG - Leibniz-Rechenzentrum München
- MTU - Maximal Transfer Unit
- RZG - Rechenzentrum der Max-Planck-Gesellschaft
- TIKSL - Tele Immersion: Kollision schwarzer Löcher (MPI-AEI, ZIB, MPI-RZG)
- ViSpaS – Visuelle Simulation polyatomarer Systeme
- VWB – Visualisierungs-Workstation Berlin
- ZIB - Konrad-Zuse-Zentrum für Informationstechnik Berlin

## IV.3 Referenzen, URLs

Unter den hier aufgeführten Referenzen findet man Informationen zu der eingesetzten Software, die Software selber zum Download oder weitere Informationen über dieses oder ähnliche Projekte.


- Abschlussbericht Testbed West, Bericht GIGAnet
  http://webdoc.sub.gwdg.de/ebook/ah/1999/dfn/GTBWest.pdf
- Amira: amira.zib.de
- Deutsches Forschungsnetz: www.dfn.de
- fhi98md Homepage: www.fhi-berlin.mpg.de/th/fhimd/
- FHI ViSpaS Projekt-Seite: www.fhi-berlin.de/th/projects/vispas
- Meta - Projektseite Metacomputing Projekt im GTB Süd
  http://www.rzg.mpg.de/rzg/projects/gigabit/Meta/Meta.html
- Perl: www.perl.org
- TCL: www.scriptics.com
- Xitami Web-Server: www.imatix.com
- xmgrace - http://plasma-gate.weizman.ac.il/Grace/
- xmgr - http://plasma-gate.weizman.ac.il/Xmgr/