

Anwendung der Tele-Immersion in Weitverkehrsnetzen

Abschlussbericht, Version 1.1, 25.07.2003

Stephan Olbrich, Hans-Christian Hege, Alexander von Berg, Hubert Busch, Karsten Chmielewski, Arnd Diekmann, Dirk Ditschke, Ralf Einhorn, Oliver Göthel, Sebastian Heidl, Steffen Heinze, Matthias Heyder, Fritz Hüsemann, Nils Jensen, Harald Knipp, Karsten Meier, André Merzky, Siegfried Raasch, Ralf Staudemeyer, Manfred Stolle, Tino Weinkauff, Stefan Zwierlein, Helmut Pralle



Projektleiter: Dr. Stephan Olbrich
Regionales Rechenzentrum für Niedersachsen (RRZN)
Universität Hannover
Hans-Christian Hege
Konrad-Zuse-Zentrum für Informationstechnik Berlin (ZIB)

Teilnehmende
Einrichtungen: RRZN, Universität Hannover
ZIB, Berlin
Institut für Meteorologie und Klimatologie (IMUK), Universität Hannover
Institut für Thermodynamik (IfT), Universität der Bundeswehr Hamburg
Institut für Strömungsmechanik und Elektronisches Rechnen
im Bauwesen (ISEB), Universität Hannover

Projektlaufzeit: 26 Monate (01.01.2001 – 28.02.2003)

Inhaltsverzeichnis

1	Beschreibung des Projekts	3
1.1	Projektziel	3
1.2	Inhaltlicher Hintergrund	3
1.3	Informations- und kommunikationstechnische Beschreibung	7
1.4	Projektgliederung	12
2	Durchführung des Projekts	21
2.1	Personal	21
2.2	Publikationen	22
2.3	Präsentationen	25
3	Ergebnisse des Projekts	31
3.1	Teilprojekt 1 – Gigabit-Netz und Systemtechnik	31
3.2	Teilprojekt 2 – 3D/VR-System	55
3.3	Teilprojekt 3 – Informationsextraktion und Datenvorverarbeitung	71
3.4	Teilprojekt 4 – Simulation atmosphärischer Konvektionsströmungen	77
3.5	Teilprojekt 5 – Molekulardynamische Simulation	87
3.6	Teilprojekt 6 – Simulation von Strömung und Sedimentation in einem Tidehafen	93
4	Zusammenfassung und Ausblick	97
4.1	Zusammenfassung der Ergebnisse	97
4.2	Veränderungen der Rahmenbedingungen und erweiterte Kontexte	98
4.3	Einfluss der Ergebnisse und Zukunftsperspektiven	100
5	Referenzen	101

Das Vorhaben wurde aus Mitteln des Bundesministeriums für Bildung und Forschung (BMBF) durch den Verein zur Förderung eines Deutschen Forschungsnetzes e. V. (DFN-Verein) finanziert.

1 Beschreibung des Projekts

In diesem Kapitel wird das Projekt gemäß dem vom Technischen Ausschuss des DFN-Vereins befürworteten und vom DFN-Vorstand genehmigten Projektantrag vom 25.11.1999 beschrieben.

1.1 Projektziel

1.1.1 Zusammenfassung

Anhand von Modellszenarien soll die praktische Machbarkeit geographisch verteilter Systeme für Simulationsrechnungen zu dreidimensionalen, zeitvarianten Phänomenen im Grand-Challenge-Bereich systematisch untersucht werden. Basis für Interaktionen mit dem Simulationsmodell sind verteilte Tele-Immersion-Umgebungen, die den beteiligten, räumlich getrennten Forschergruppen eine weitgehend synchrone Sicht auf die Ergebnisse der Simulationsrechnungen in Form von 3D-Darstellungen ermöglichen. Unterstützt durch Virtual-Reality-Präsentations- und Interaktionsmethoden können damit eine wissenschaftliche Bewertung sowie die notwendigen Abstimmungen effizient durchgeführt und Entscheidungen über den weiteren Verlauf der Simulationsrechnungen interaktiv umgesetzt werden.

1.1.2 Angestrebte Ergebnisse

Die praktische Machbarkeit tele-immersiver Steuerung interaktiver Rechenprozesse für Grand-Challenge-Modelle wird nachgewiesen. Leistungsparameter der Rechner- und Kommunikationssysteme für geographisch verteilte Szenarien werden ermittelt. Die notwendige Software wird im DFN-Umfeld verfügbar gemacht und via WWW zur Verfügung gestellt.

1.1.3 Szenario für die Pilotnutzung

Für die Visualisierung dreidimensionaler, zeitlich veränderlicher Vorgänge wird ein System-Prototyp entwickelt und in Pilotanwendungen eingesetzt. Als Basis der Entwicklung dient das am RRZN bereits entwickelte System DocShow-VR, das für dynamische 3D-Szenen und kooperatives Arbeiten erweitert wird. Alternativ soll das am ZIB verwendete Visualisierungssystem Amira eingesetzt werden können, um einen größeren Satz von Visualisierungsverfahren anwenden zu können.

1.1.4 Nutzung der Projektergebnisse

Aufbauend auf den Ergebnissen des Projekts soll eine geographisch verteilte Konfiguration von Höchstleistungsressourcen (Höchstleistungsrechner, Datenkonvertierungs-, Speicher-, Visualisierungssysteme einschließlich Peripherie und Netzwerken) eingerichtet und in den Wirkbetrieb überführt werden. Basis der tele-immersiven Mechanismen ist das für Hochleistungsanwendungen entwickelte System DocShow-VR und das für hochqualitative Darstellungen optimierte Amira.

1.2 Inhaltlicher Hintergrund

Die rechnergestützte Simulation von Systemen der Natur-, Ingenieurs-, Bio- und Wirtschaftswissenschaften erlangt ständig wachsende Bedeutung. Im Bereich höchster Anforderungen (Grand Challenges) besteht der Bedarf nach immer leistungsfähigeren Rechnern (Höchstleistungsrechnern). Gleichzeitig entsteht Bedarf nach Exploration der im Allgemeinen in Form großer Datenmengen erzeugten Rechenergebnisse und ihrer Aufbereitung für eine interdisziplinäre wissenschaftliche Interpretation. Hierfür werden zunehmend Visualisierungswerkzeuge als Postprozessoren eingesetzt. Besonders für die Untersuchung von dreidimensionalen, zeitabhängigen Phänomenen ist eine 3D-Darstellung mit Animations- und Navigationsmöglichkeiten außerordentlich nützlich. Ein immersives VR-System bietet potenziell visuelle, haptische und akustische Schnittstellen und ist daher besonders geeignet als Modell für eine Mensch/Computer/Mensch-Kommunikation.

Als Ziel der rechnergestützten Simulation kann ein interaktives System nach *Abbildung 1* betrachtet werden. Dabei wird im Allgemeinen von räumlich verteilten Ressourcen auszugehen sein. Die Verteiltheit bezieht sich sowohl auf die maschinellen Ressourcen (Höchstleistungsrechner, Visualisierungsrechner etc.) als auch auf die am Projekt arbeitenden Forschergruppen.

Die Zielvorstellung enthält daher eine verteilte 3D/VR-Umgebung zur Zusammenarbeit räumlich getrennter Forschergruppen bei der Bearbeitung von Grand-Challenge-Aufgaben auf entfernten Höchstleistungsrechnern.

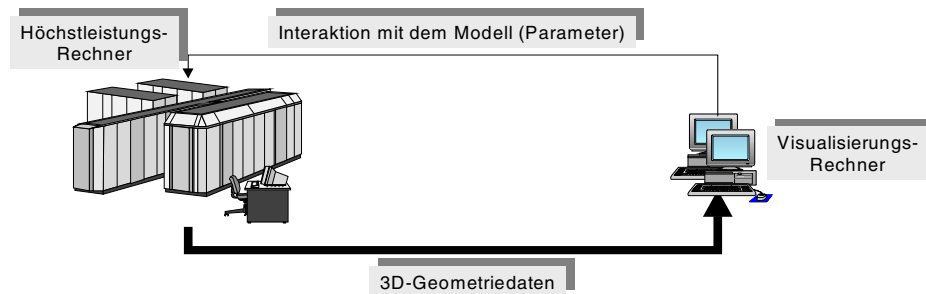


Abbildung 1. Interaktives Simulationssystem.

Szenario

Das Gesamtszenario ist in seinen Einzelheiten hoch komplex und stellt hohe Anforderungen an die Kopplung der beteiligten Ressourcen und Prozesse (Rechner, Peripherie, Netzwerk, Software). *Abbildung 2* zeigt den prinzipiellen Aufbau des Szenarios.

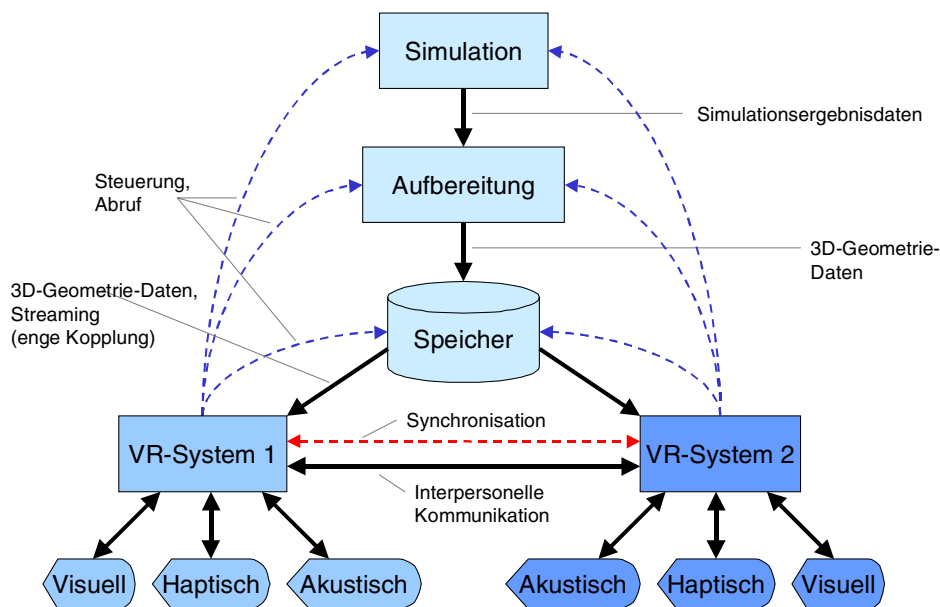


Abbildung 2. Tele-Immersionsszenario.

Die Simulationsrechnung wird auf einem Hoch-/Höchstleistungsrechner <Simulation> ausgeführt, die Ergebnisdaten werden für die Aufbereitung zu 3D-Geometrie-Daten auf einen leistungsfähigen Rechner <Aufbereitung> übertragen. Falls das Datenvolumen zu umfangreich ist, findet bereits auf dem Rechner <Simulation> eine Reduktion der Daten auf niedrig-dimensionale Geometrien statt. Die auf-

bereiteten 3D-Geometrie-Daten werden auf einem Server <Speicher> abgelegt, von dem sie in Form eines Datenstroms zu den VR-Systemen übertragen werden. Die Systeme sind grundsätzlich als verteilte Systeme implementiert und über lokale und Weitverkehrsnetze gekoppelt. Die VR-Systeme sind mit visuellen, akustischen und haptischen E/A-Geräten gekoppelt. Sie stellen vor Ort insbesondere die notwendige Rendering-Leistung (z. B. 3D-Hochleistungsgraphik) und qualitativ hochwertige Display-Systeme (z. B. Stereo-Großbildprojektion) sowie Tracking-Systeme bereit. In der ersten Stufe des Projekts wird eine enge Kopplung zwischen Simulation, Speicher- und VR-System betrachtet. Hierfür werden die notwendigen Prozesse implementiert. In einer weiteren Stufe wird das kooperative Arbeiten von zwei VR-Systemen untersucht.

3D-Graphikdaten

Die 3D-Geometriedaten liegen in einem speziellen Format vor, welches am RRZN entwickelt wurde und hier als DVR-Format bezeichnet wird. In diesem Datenformat werden geometrische 3D-Objekte durch Topologieinformation, Koordinatentripel, Normalenvektoren und weitere Oberflächenattribute (z. B. Farbe) repräsentiert sowie Defaultwerte für virtuelle Kameras und Beleuchtungen spezifiziert. Es wurde eine binäre Codierung im IEEE-Format festgelegt, um den für die Online-Präsentation erforderlichen Verarbeitungsaufwand möglichst gering zu halten und auf diese Weise eine ausgewogene Streaming-Pipeline – Speicher, Server, Kommunikation, Client, Präsentation – zu realisieren. Der Stand der bisherigen Arbeiten wurde auf der IFIP HPN '98 und Terena NordUNET '99 vorgetragen und demonstriert.

Eine Interoperabilität mit dem standardisierten 3D-Datenformat VRML (Virtual Reality Modeling Language, ISO/IEC 14772-1:1997) ist durch den im Software-Paket enthaltenen Konverter¹ gewährleistet. Dieser dient bislang zur Vorverarbeitung der in ASCII-Klartextformat codierten VRML-Dateien, um durch WWW-Zugriff auf die generierten DVR-Dateien eine wesentlich beschleunigte Online-Präsentation von 3D-Szenen zu ermöglichen, da die äußerst rechenaufwändigen VRML-Ausgabe und -Konvertiervorgänge vermieden werden.

Die bisher für interne Zwecke zur DVR-Generierung realisierte Klassenbibliothek soll für die im Rahmen dieses Projekts geplanten Arbeiten als offene Schnittstelle zur direkten Ausgabe von DVR-Daten, z. B. aus Simulationsrechnungen heraus, dienen. Durch Offenlegung dieser Mechanismen kann diese auf leistungsfähige Netz- und Rechner-Plattformen ausgelegte Technologie generell verfügbar gemacht werden.

Eine Kompression dieser Datenstrukturen kommt zunächst nicht in Betracht, da signifikante Kompressionsfaktoren nur mit verlustbehafteten Kompressionsverfahren erzielt werden können. In den hier betrachteten wissenschaftlichen Anwendungen muss zur korrekten und sicheren Ergebnisinterpretation – zumindest wahlweise – eine das Simulationsergebnis akkurat wiedergebende Präsentation möglich sein. Insbesondere während der Entwicklungs- und Testphase, in der die Korrektheit der Simulations- und Geometrieerzeugungsmodule zu überprüfen ist, stellen unkontrollierbare Darstellungsartefakte ein Problem dar. Daher soll zunächst mit unkomprimierten Daten gearbeitet werden. Erreicht die Komplexität der Daten ein für die Handhabung zu hohes Niveau, z. B. wegen nicht realisierbarer Anforderungen bezüglich Datenvolumen, Bitraten und Rendering-Raten, müssen jedoch Polygonreduktion bzw. Level-of-Detail-Techniken eingesetzt werden. Techniken dieser Art können aus Aufwandsgründen innerhalb des Projekts allerdings nicht implementiert werden. Es soll versucht werden, die Lücke durch Nutzung außerhalb des Projekts angesiedelter Ressourcen zu schließen.

Bezug zu den vorher bereits existierenden Projekten

Das beantragte Projekt ist in gewisser Weise komplementär zu dem DFN-Projekt *Tele-Immersion – Kollision Schwarzer Löcher* (TIKSL). Im TIKSL-Projekt wird eine Softwareumgebung geschaffen, die es erlaubt, im Simulationsrechner vorhandene Daten interaktiv zu explorieren und die Simulation online visuell zu kontrollieren sowie zu steuern². Dazu wird ein flexibler Remote-Datenzugriff entwickelt, der dem Anwender volle Freiheit bei der Auswahl und Anwendung von Visualisierungsmethoden

1. <http://www.dfn-expo.de/Technologie/DocShow-VR/>

2. <http://www.zib.de/Visual/projects/TIKSL/>

den sowie der interaktiven Steuerung entfernt laufender Simulationen gibt. Kooperative Aspekte werden im TIKSL-Projekt nur ansatzweise realisiert.

Im vorliegenden Projekt hingegen wird eine Lösung mit eingeschränkteren Möglichkeiten entwickelt, die für viele der heutigen (typischerweise immer noch batch-orientierten) Anwendungen im Hochleistungsrechnen auch einen deutlichen Fortschritt bedeuten. Vorteil der hier entwickelten Lösung ist ihre einfachere Handhabbarkeit. Die Einschränkungen liegen darin, dass statt der Kopplung des Simulationsprogramms mit einem interaktiv nutzbaren Visualisierungssystem, 3D-Geometrien vorproduziert werden und die Visualisierung sich auf eine gemeinsame Sichtbarmachung genau dieser Geometrien (an verschiedenen Standorten) beschränkt. Interaktiv veränderbare Parameter beziehen sich nur auf das Rendering, z. B. die Sichtposition, nicht aber die Visualisierungsverfahren selbst. Für Simulationsrechnungen, die Teilergebnisse in Animationsgeschwindigkeit – d. h. in der Größenordnung von 10 Updates je Sekunde – liefern können, sind auch On-the-fly-Präsentationen und Steuerungsfunktionen (Restart, Parametrisierung der Simulationsrechnung) vorgesehen. Auch kooperativen Aspekten, wie etwa die Darstellung der handelnden Personen im dargestellten Datenraum, soll ein höherer Stellenwert beigemessen werden.

Welche der beiden Lösungen für eine Anwendung besser geeignet ist, hängt von der Fragestellung, der Arbeitsweise und der vorhandenen Software ab. Steht interaktive Datenexploration von Simulationsdaten im Vordergrund, sind die im TIKSL-Projekt erarbeiteten Werkzeuge die geeigneten. Zur Darstellung von Ergebnissen, z.B. aus Serienrechnungen, mittels grafischer Standardverfahren sind die im vorliegenden Projekt zu schaffenden Verfahren im Allgemeinen geeigneter. *Tabelle 1* fasst die Unterschiede zwischen dem vorliegendem Projekt (ATIW) und dem TIKSL-Projekt — aus Anwendersicht — zusammen.

	ATIW	TIKSL
Adaption der Simulationen	i.d.R. einfach	aufwändiger
Steuerung der Anwendung	wiederholte parametrisierte Läufe	interaktive Steuerung durch Remote-Zugriff auf interne Parameter
Prä-Visualisierung	kleines Set einfacher Methoden	sehr flexibel, aber komplexer
Visualisierungssystem	Browser-Plugin (DocShow-VR), andere DVR-kompatible Geometrie-Viewer (Amira)	vielseitige, hochinteraktiv nutzbare Visualisierungsumgebung (Amira)
Offline Visualisierung	Inhärent	nur für explizit gespeicherte Daten nach Ablauf der Simulation
Kooperative Visualisierung	voll integriert	teilweise integriert
Daten-Formate im Netz	3D-Geometrien (DVR)	Rohdaten bzw. extrahierte Simulationsergebnisse
Daten-Protokolle	RTSP, HTTP	GLOBUS-basiert (RPC)
Netzwerk-Protokolle	Gigabit Ethernet	ATM

Tabelle 1. Vergleich der Projekte ATIW und TIKSL.

Erfahrungen aus dem TIKSL-Projekt werden in dem vorliegenden Projekt berücksichtigt. Umgekehrt sollen Ergebnisse aus dem ATIW-Projekt, insbesondere die Schnittstelle zur Datenextraktion und die Bibliothek zur parallelen Geometrieerzeugung sowie die Erweiterungen zur kollaborativen Visualisierung, auch im TIKSL-Projekt verwendet werden.

1.3 Informations- und kommunikationstechnische Beschreibung

1.3.1 Beschreibung der geplanten Anwendungsszenarien

Die Ergebnisse von Simulationsrechnungen, die auf zentralisiert betriebenen Höchstleistungsrechnern durchgeführt werden, sollen mittels netzverteilter, qualitativ hochwertiger Werkzeuge effizient erzielt, interpretiert und vermittelt werden. Aufgrund der unterschiedlichen Anforderungsprofile (siehe auch *Tabelle 2*) werden hier drei Anwendungsszenarien unterschieden:

(A) *Präsentationsszenario:*

Informationssystem zum interaktiven Abruf von Ergebnispräsentationen in Form von navigierbaren 3D-Animationen – „Virtual Reality Movie“ – mittels 3D-Streaming-Verfahren.

(B) *Explorationsszenario:*

Virtuelle Experimentierumgebung mit interaktiver, bidirektionaler Kopplung von Simulations-, Visualisierungs- und Darstellungsmodulen.

(C) *Diskussionsszenario:*

Erweiterung des Präsentationsszenarios um computergestützte Zusammenarbeit – Computer-Supported Cooperative Work (CSCW).

Diese Anwendungsszenarien sind Teile bzw. Spezialfälle des in *Abbildung 2* dargestellten Tele-Immersion-Szenarios. In dem hier vorliegenden Projektvorschlag erfolgt eine Konzentration auf diese drei Szenarien. Das zu erstellende System soll alle drei Szenarien ermöglichen.

Die folgende *Tabelle 2* fasst die Charakteristika der Szenarien zusammen:

	(A) Präsentation	(B) Exploration	(C) Diskussion
Anwender	Breites Publikum; Einzelpersonen	Spezialisten; Einzelperson oder kleine Gruppe	Mehrere, räumlich verteilte Spezialisten (-gruppen)
Szenario	Streaming aus vorbereiteten 3D-Geometrie-Daten bzw. davon abgeleiteten Multimedia-Daten, z. B. Bild, Videofilm, ggf. annotiert, vertont	On-the-fly-Verarbeitung mit Steuerungsmöglichkeit von Simulationsrechnungen (Modell-) und Visualisierungsparametern	Streaming aus vorbereiteten 3D-Geometrie-Daten bzw. davon abgeleiteten Multimedia-Daten, z. B. Bild, Videofilm, ggf. annotiert, vertont
Parametrisierung	Auswahl aus vorgegebenen Konfigurationen	Flexible Steuerung durch Anwender	Auswahl aus vorgegebenen Konfigurationen
Zeitachse	Interaktive Positionierung und Ausspielung in Echtzeit	Restriktionen des zeitlichen Ablaufs durch Simulationsrechnungs- und Visualisierungsmodulen	Interaktive Positionierung durch „Master“ und synchronisierte Ausspielung in Echtzeit
Interpersonelle Kommunikation	Keine	Keine	Synchronisation der Navigation und der Zeitachse; Audio-/ Videokonferenz optional

Tabelle 2. Charakteristika der Anwendungsszenarien.

Im Folgenden werden die Szenarien näher erläutert. Die Abbildungen enthalten schematische Darstellungen der Szenarien. Bei der Realisierung soll die bereits als Prototyp am RRZN/RVS vorliegende 3D/VR-Software DocShow-VR als Ausgangsbasis dienen und entsprechend erweitert werden. DocShow-VR besteht aus einem Netscape-Plugin (DVR) auf Client-Seite und einem Streaming-Server (DVRS). Beide kommunizieren über das Realtime-Streaming-Protokoll (RTSP) und über ein eigenes 3D-Graphik-Transport-Protokoll (DVRP). Diese Komponenten werden in den Abbildungen verwendet. Zusätzlich sind ein HTTP-Server (HTTPD), sowie konfigurierbare Simulations- (SIM) und Visualisierungsprogramme (VIS) enthalten.

Alternativ zum DocShow-VR Client kann Amira eingesetzt werden, um auch spezielle, nur in Amira vorhandene Visualisierungsalgorithmen einsetzen zu können. Dazu wird Amira um Schnittstellen für die o. g. Protokolle erweitert.

(A) Präsentationsszenario

Eine schematische Darstellung zeigt *Abbildung 3*. Der Ablauf des Szenarios ist durch folgende vier Schritte gekennzeichnet (in der Abbildung markiert):

1. Der jeweilige Visualisierungs-Client (VR-System) ruft eine HTML-Seite mit eingebettetem DVRS-Objekt ab (HTTP).
2. Der Visualisierungs-Client baut eine Verbindung zum 3D-Streaming-Server auf (RTSP).
3. Der 3D-Streaming-Server überträgt die 3D-Objekte zum Visualisierungs-Client (DVRP über TCP/IP).
4. Nachdem das erste 3D-Objekt übermittelt wurde, führt der Visualisierungs-Client synchron zur weiteren Übertragung noch weitere Aktionen aus:
 - Er stellt 3D-Objekte entsprechend der aktuellen virtuellen Kamera dar (Rendering).
 - Er modifiziert die virtuelle Kamera gemäß Anforderungen aus den Input-Devices (VR-Navigation).
 - Er steuert den 3D-Streaming-Server entsprechend den Anforderungen aus den Input-Devices (RTSP, VCR-Paradigma).

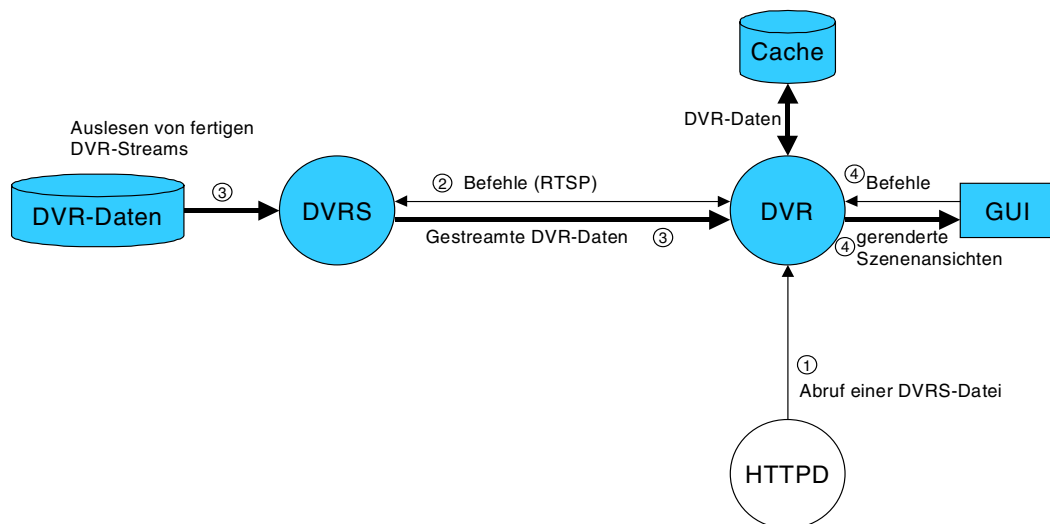


Abbildung 3. Präsentationsszenario.

(B) Explorationsszenario

Eine schematische Darstellung zeigt *Abbildung 4*. Der Ablauf des Szenarios ist durch folgende fünf Schritte gekennzeichnet (in der Abbildung markiert):

1. Der jeweilige Visualisierungs-Client (VR-System) ruft eine HTML-Seite mit eingebettetem DVRS-Objekt ab (HTTP).

2. Der Visualisierungs-Client baut eine Verbindung zum 3D-Streaming-Server auf (RTSP).
3. Der 3D-Streaming-Server interagiert mit einem Simulationsprogramm, welches 3D-Objekte generiert und zurückerliefert.
4. Der 3D-Streaming-Server überträgt die 3D-Objekte zum Visualisierungs-Client (DVRP über TCP/IP).
5. Nachdem das erste 3D-Objekt übermittelt wurde, führt der Visualisierungs-Client synchron zur weiteren Übertragung noch weitere Aktionen aus:
 - Er stellt 3D-Objekte entsprechend der aktuellen virtuellen Kamera dar (Rendering).
 - Er modifiziert die virtuelle Kamera gemäß Anforderungen aus den Input-Devices (VR-Navigation).
 - Er steuert den 3D-Streaming-Server entsprechend den Anforderungen aus den Input-Devices (RTSP, VCR-Paradigma).
 - Er stellt GUI-Elemente zur Modifizierung von Parametern der Simulation dar.

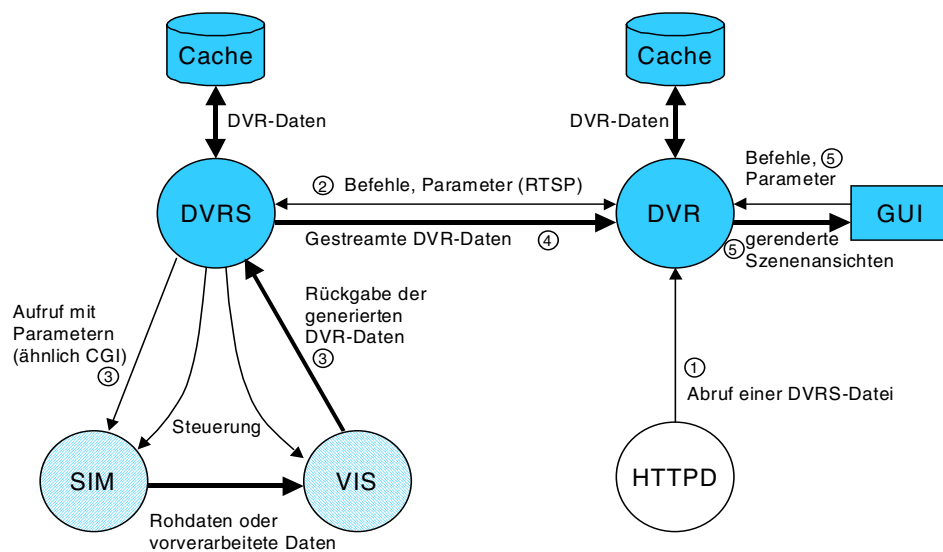


Abbildung 4. Explorationszenario.

(C) Diskussionsszenario

Die Erweiterung des Szenarios A auf die kooperative Betrachtung eines 3D-Films zeigt *Abbildung 4*.

1. Die Visualisierungs-Clients rufen eine HTML-Seite mit eingebettetem DVRS-Objekt ab (HTTP).
2. Ein Visualisierungs-Client baut eine CSCW-Verbindung (Computer-Supported Cooperative Work) zum anderen Visualisierungs-Client auf; daraufhin senden beide Clients jeweils Update-PDUs zur Synchronisation der Navigation des jeweils anderen Clients (virtuelle Kamera, virtueller VCR).
3. Ein Visualisierungs-Client baut eine Verbindung zum 3D-Streaming-Server auf (RTSP), der andere Visualisierungs-Client bekommt eine entsprechende Update-PDU und baut ebenfalls eine RTSP-Verbindung auf.
4. Der 3D-Streaming-Server überträgt 3D-Objekte (DVR) zu beiden Visualisierungs-Clients (DVRP).
5. Nachdem das erste 3D-Objekt übermittelt wurde, führen beide Visualisierungs-Clients jeweils synchron zur weiteren Übertragung noch weitere Aktionen aus:
 - Darstellung von 3D-Objekten entsprechend der aktuellen virtuellen Kamera (Rendering).

- Modifizierung der virtuellen Kamera gemäß den Anforderungen aus den Input-Devices (VR-Navigation).
 - Steuerung des 3D-Streaming-Servers entsprechend den Anforderungen aus den Input-Devices (RTSP, VCR-Paradigma).
6. Optional erfolgt daneben eine synchrone Audio/Video-Kommunikation zwischen beiden VR-Systemen:
- Audio: CD/DAT-Qualität, z. B. über TCP/IP.
 - Video: hochqualitative Übertragung, z. B. über SDI (D1 – ITU-R 601), je nach verfügbarer Übertragungstechnik evtl. komprimiert.
 - Einbettung des Videobildes in die dargestellte 3D-Szene: Rendering nach jedem Frame-Update.

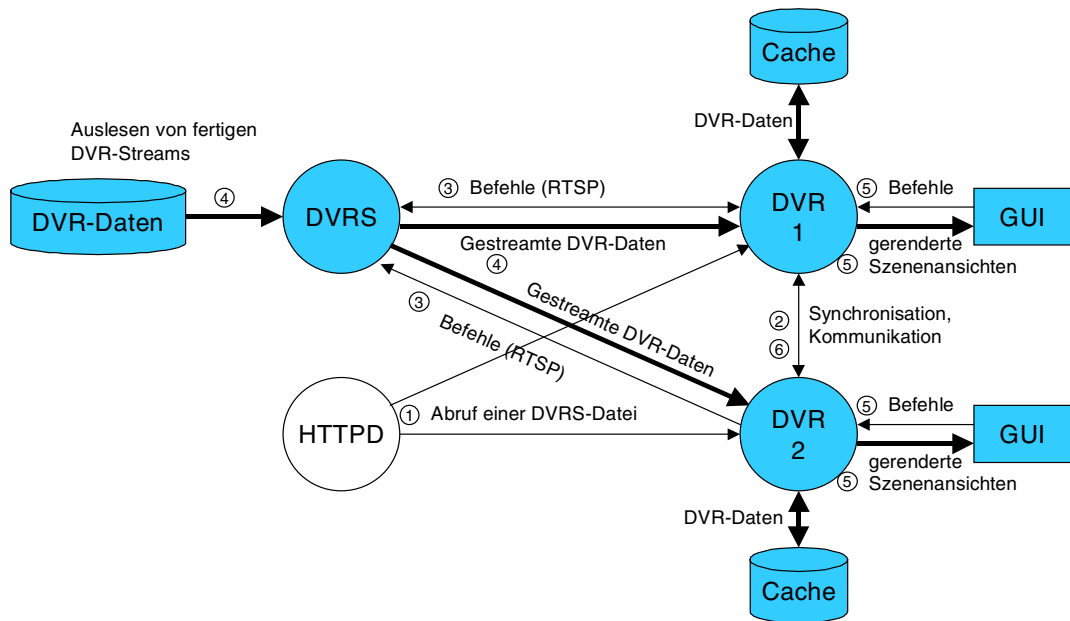


Abbildung 5. Diskussionsszenario.

Optional besteht unabhängig vom verwendeten Szenario die Möglichkeit, über einen Videoausgang der Graphikeinheit des VR-Systems einen Videofilm der betrachteten Szene zu generieren und aufzuzeichnen. Beispielsweise verfügt die in Hannover vorhandene SGI Onyx2 über eine D1-Video-Schnittstelle. So gewonnene Videofilme können dann, nachbearbeitet und vertont, für die Offline-Präsentation von Simulationsergebnissen verwendet werden. Die Nachbearbeitung von Videos soll in den Multimedia-Labors des RRZN und ZIB geschehen, d. h. für CDROM, DVD, Videokassetten, oder für die Online-Präsentation im WWW (Videostreaming) aufbereitet werden.

1.3.2 Bandbreitenbedarf

Der Bandbreitenbedarf für die beschriebenen Szenarien ergibt sich zu einem variablen Teil aus den zu visualisierenden Simulationsdaten und zu einem festen Teil aus der interpersonellen Kommunikation. Der Einfachheit halber betrachten wir hier den Fall, dass die zu visualisierenden Daten auf Plattenspeicher vorliegen, die Prozesskette also keine Online-Simulation enthält.

Die erreichbare Datenrate hängt ab von der Leistungsfähigkeit sämtlicher Übertragungssysteme in der Kette vom Festplattenspeicher bis zur Graphik-Engine. Da die Leistungsfähigkeit von CPU, Hauptspeicher und Systembus in der Regel um Größenordnungen über der vom Festplattensystem, vom Netz und vom Rendering-System liegt, wird der limitierende Faktor einer der letztgenannten sein. Festplattensysteme mit RAID-Level 3 oder 5, die über Fibre-Channel angeschlossen sind, erreichen im lesenden Zugriff eine Datenrate von ca. 400–600 Mbps, wobei die Grenze eines Fiber-Channel-Kontrollers bei ca. 800 Mbps liegt (mit Festplatten älterer Bauart wurden am RRZN Datenraten von

300–400 Mbps gemessen). Durch parallelen Zugriff über mehrere Fibre-Channel-Controller sind aber auch höhere Datenraten erzielbar, bevor sich andere Systemkomponenten limitierend auswirken, z. B. der Systembus (PCI: ca. 1–4 Gbps). Die Rendering-Leistung einer High-End-Graphik-Workstation (SGI Onyx2) liegt im Bereich bis ca. 900 Mbps, abhängig von der 3D-Szene, d. h. vom Datenvolumen, von der Topologie der Dreiecke, von den Beleuchtungsparametern, etc.

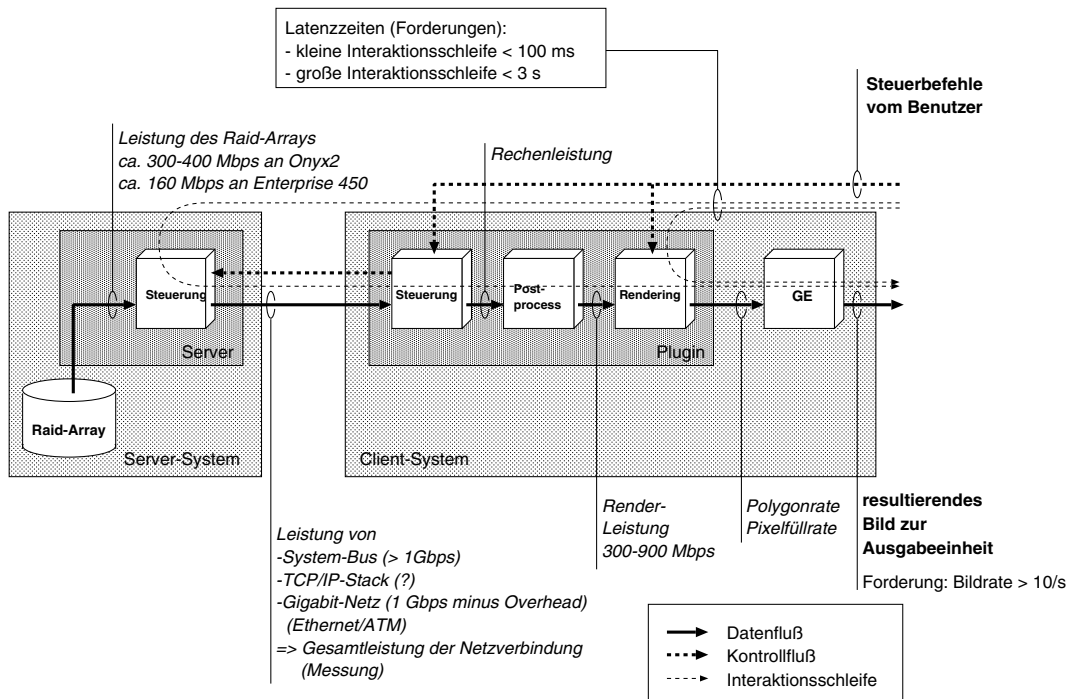
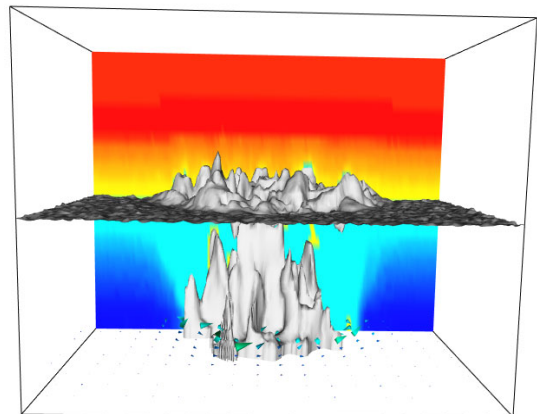


Abbildung 6. Datenraten und begrenzende Faktoren in der Prozesskette.

Einige Beispiele aus der Praxis belegen dies. Es wurden 3D-Szenen aus zwei Anwendungsszenarien ausgewählt (siehe Abbildung 7) und der Rendering-Durchsatz auf einer Graphikworkstation gemessen. Die Ergebnisse sind in Tabelle 3 dargestellt.



(a) Szene „DX-03“:
Ergebnis einer Visualisierung mit dem IBM Data Explorer, aus dem OpenGL-Benchmark „viewperf“: 91.584 Dreiecke mit Normalenvektoren an allen Eckpunkten.



(b) Szene „IMUK-Express-492“:
Visualisierung eines Temperatur- und Strömungsfeldes mit AVSEXPRESS, aus einem IMUK-Ergebnisdatensatz: 109.255 Primitive mit verschiedenen Attributen (Normalen, Farben).

Abbildung 7. 3D-Szenen aus zwei verschiedenen Anwendungsszenarien.

Szene	Topologie	Datenvolumen (DVR-Datensatz)	Rendering- Zeit	Rendering- Datenrate
„DX-03“ (Abb. 7a)	Separate triangles	6.601.928 Byte	53,6 ms	986 Mbit/s
	Triangle strips	2.252.744 Byte	22,1 ms	816 Mbit/s
„IMUK-Express-492“ (Abb. 7b)	Verschiedene Graphikprimitive	9.080.416 Byte	168 ms	432 Mbit/s

Tabelle 3. Ergebnisse der Messung der Rendering-Leistung in den Szenarien lt. Abb. 7. Die Messungen wurden mit „DocShow-VR“ auf einer SGI Onyx2 Infinite Reality (4 CPUs R10000, 195 MHz) unter IRIX 6.5.3 durchgeführt. Die Fenstergröße betrug jeweils 848 x 511 Pixel, Antialiasing war mit 8 Samples/Pixel konfiguriert. Der Durchsatz im 3D-Graphiksystem (Hauptspeicher => Perspektivische Darstellung) wird als Rendering-Datenrate ausgewiesen: Datenvolumen * 8 bit/Byte / Rendering-Zeit.

Der Anteil der interpersonellen Kommunikation hängt hauptsächlich vom verwendeten Videoformat ab, während die Audio- und die Steuerdaten vernachlässigbar sind. Für die Videoübertragung kommen – je nach verfügbarer Netz-Infrastruktur und geforderter Qualität – verschiedene Standards in Frage. Wählt man z. B. unkomprimierte D1-Übertragung, so sind 270 Mbps pro Richtung erforderlich, bei Kompression im Motion-JPEG-Verfahren sind es etwa 25 Mbps. Diese Bandbreite ist zwischen den 3D/VR-Systemen während der gesamten Arbeitssitzung konstant erforderlich, unabhängig davon, ob 3D-Geometrie-Daten vom Server abgerufen werden oder nicht.

Diese Abschätzungen lassen erkennen, dass die Bandbreitenanforderungen der geplanten Szenarien nur mit Gigabit-Netzen erfüllt werden können.

Im Rückkanal fließen lediglich Steuerinformationen mit geringem Volumen. Diese treten nicht kontinuierlich, sondern burst-artig auf. Wichtig ist aber eine geringe Verzögerungszeit, um die Reaktionszeit der interaktiven Navigation nicht zu groß werden zu lassen. Interaktive Systeme sollten eine Reaktionszeit unter 100 ms haben. Diese Anforderung ist mit der geplanten Vernetzung ebenfalls realisierbar.

1.4 Projektgliederung

Das Projekt gliedert sich in mehrere Teilprojekte, die sich in vier aufeinander aufbauende Ebenen einordnen lassen. Die untere Ebene bildet das Netz, die zweite Ebene das 3D/VR-System, die dritte Ebene die Informationsextraktion und Datenvorverarbeitung und die obere Ebene enthält die Anwendungen.

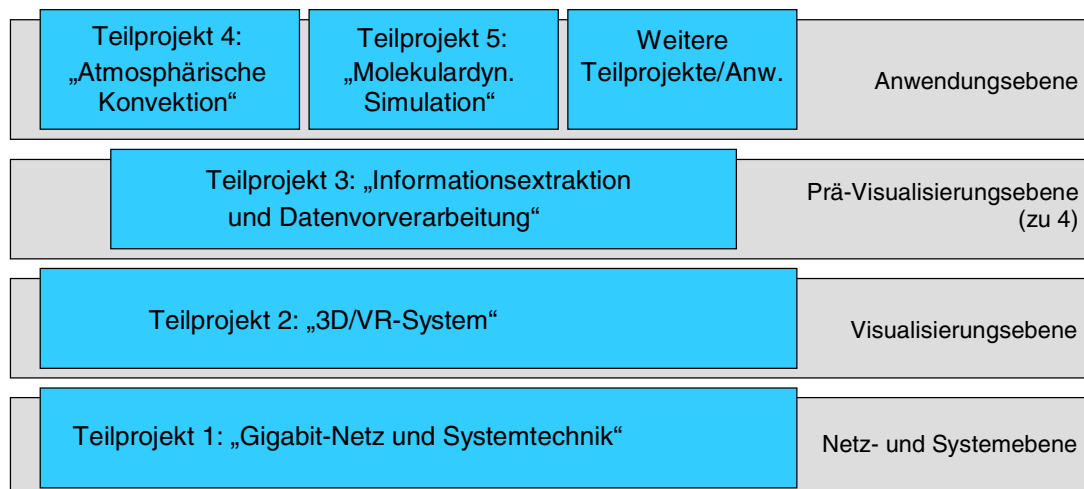


Abbildung 8. Projektgliederung.

In der Anwendungsebene des Projekts liegen mehrere Teilprojekte, die das 3D/VR-System als Pilotanwender nutzen. Dabei wird es sich nicht um neue Projekte, sondern um bereits an Instituten laufende Forschungsvorhaben handeln, die durch die Anwendung des 3D/VR-Systems in besonderer Weise profitieren können. Die Hauptaufgaben in den Teilprojekten der Anwendungsebene sind:

- die Anpassung der bereits bestehenden Simulationsprogramme an die Anforderungen des 3D/VR-Systems,
- die Erstellung von anwendungsspezifischen Visualisierungsmodulen für die Informationsextraktion und die Erzeugung der DVR-Daten.

1.4.1 Teilprojekt 1: „Gigabit-Netz und Systemtechnik“ (Durchführung: RRZN, ZIB)

Das Teilprojekt „Gigabit-Netz und Systemtechnik“ befasst sich mit der Realisierung und der Erprobung der Gigabit-Netze an den Standorten Berlin und Hannover. Die Arbeitspunkte in diesem Teilprojekt sind

- Auswahl und Beschaffung,
- Inbetriebnahme, Konfiguration und Test,
- Überwachung und
- Bewertung

der notwendigen Komponenten. Die Konfiguration der geplanten Testbeds ist in *Abbildung 9* dargestellt. Als Übertragungstechnologie kommen sowohl ATM (622 Mbps) als auch Gigabit-Ethernet in Frage. Nach Untersuchungen am RRZN ist eine Gigabit-Ethernet-Verbindung mit der proprietären „Jumbo Frame Option“ (Spezifikation von Alteon WebSystems, unterstützt auch von SGI und angekündigt von CISCO) besonders performant. Zwischen einer SGI Origin200 und einer SGI Onyx2 konnten damit lokal Datenraten von bis zu 720 Mbps ermittelt werden. Beim Einsatz von „Jumbo Frames“ wird die MTU von 1500 Bytes auf 9000 Bytes erhöht, wodurch die Framerate und damit die Systembelastung sinkt. Ohne „Jumbo Frames“ konnten lediglich ca. 300 Mbps übertragen werden. Der Einsatz von Gigabit Ethernet mit „Jumbo Frames“ wird für dieses Projekt favorisiert, da die Anwendungs-Szenarien entsprechende Bitraten erfordern.

Da der Preis einer solchen Gigabit-Ethernet-Strecke zwischen Hannover und Berlin zum gegenwärtigen Zeitpunkt noch nicht feststeht, wird zunächst von einer Verbindung der beiden Standorte über das bestehende B-WiN/G-WiN ausgegangen. Nach Vorliegen eines Angebots für die Strecke und positivem Votum des DFN-Vereins wird die Realisierung der Gigabit-Ethernet-Strecke Hannover-Berlin in einem zweiten Teilprojekt beantragt. Dabei ist zu beachten, dass die Anwendung im bestehenden B-WiN/G-WiN rein funktional lauffähig ist, die dort verfügbaren Datenraten jedoch für die geplanten Szenarien nicht ausreichen.

Es sollen vier verschiedene Typen von Endgeräten (Workstations) zum Einsatz kommen:

1. SGI Onyx2 mit zwei Graphikpipes vom Typ Infinite Reality mit Stereogroßbildprojektion und Trackingsystem.
2. SGI Octane mit stereofähigem 21“-Monitor und Crystal Eyes LCD-Shutterbrille.
3. HP Visualize Unix-Workstation mit Graphiksystem FX 6+, stereofähigem 21“-Monitor und Crystal Eyes LCD-Shutterbrille.
4. HP X-Class Windows-NT-Workstation mit Graphiksystem FX 6+, stereofähigem 21“-Monitor und Crystal Eyes LCD-Shutterbrille.

In Berlin werden die Simulationsergebnisse vom Parallelrechner Cray T3E mit einem IP-basierten Protokoll über HIPPI zu einem Server vom Typ SGI Origin200 übertragen. Dieser dient als Zwischenspeicher und 3D-Streamingserver zum Abruf der 3D-Szenensequenzen durch o. g. Klienten.

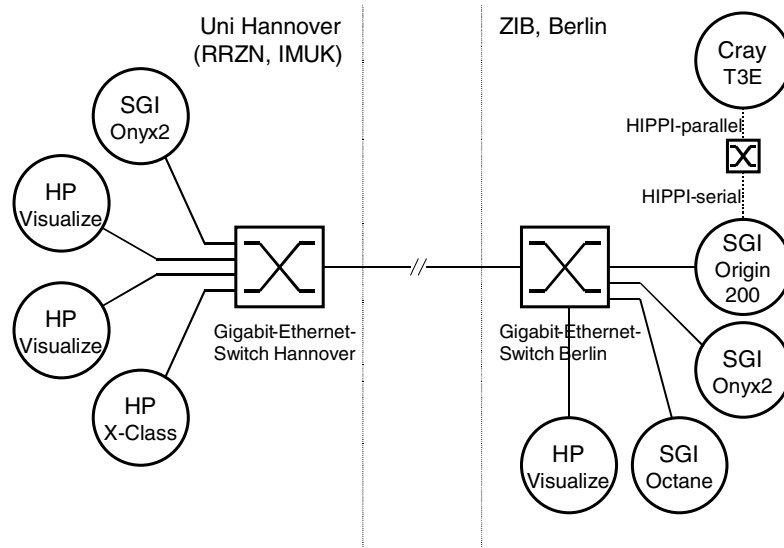


Abbildung 9. Netz- und Rechnerkonfiguration.

Die weitgehend symmetrische Verfügbarkeit von Endgeräten in Hannover und Berlin erlaubt eine Erprobung des Diskussionsszenarios (CSCW). Für dieses Szenario wird eine Videoverbindung zwischen den beiden Standorten benötigt. Diese kann, aufgrund guter Erfahrungen am RRZN und am ZIB, mit vorhandenen Codecs der Firma FORE, die Motion-JPEG-komprimierte Videodaten über ATM übertragen, eingerichtet werden. Die Datenrate beträgt ca. 25 Mbps. Die PVCs können über das B-WiN oder G-WiN geschaltet werden. Die analogen Video-Ausgangssignale der FORE-Codecs sollen über A/D-Wandler in die DIVO-Boards (Digital Video Option) der beiden SGI Onyx2 eingespielt (siehe *Abbildung 10*) oder wahlweise auf Monitoren angezeigt werden.

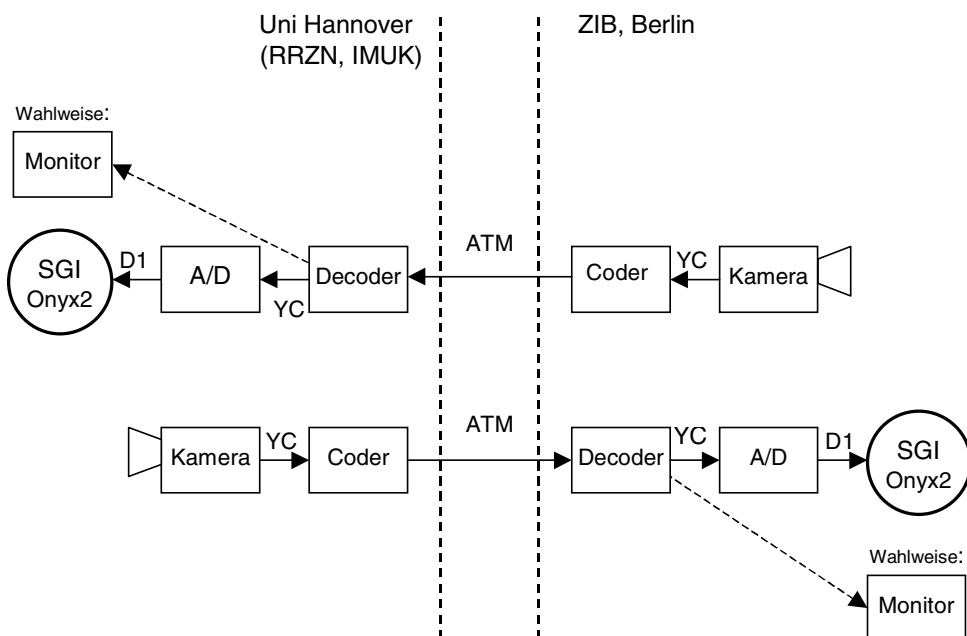


Abbildung 10. Videokonfiguration (Beispiel mit FORE-Videocodecs).

Je nach verfügbarer Netzstrecke und geforderter Darstellungsqualität kommt auch eine unkomprimierte D1-Videoübertragung (270 Mbit/s gemäß ITU-R 601) oder DV (Firewire, IEEE 1394) in Betracht.

1.4.2 Teilprojekt 2: „3D/VR-System“ (Durchführung: RRZN)

Das Teilprojekt „3D/VR-System“ befasst sich mit der Realisierung des in der Beschreibung der Anwendungsszenarien erwähnten interaktiven Simulationssystems, bestehend aus DVR-Server und Client, wobei der Client als Netscape-Plugin (DocShow-VR) oder als HTTP-Client (Amira) realisiert ist. Über den HTTP-Abruf einer DVRS-Datei vom Webserver wird das Plugin aktiviert. Aus der DVRS-Datei erhält es Informationen über die Datenquelle. Dabei sind zwei Fälle zu unterscheiden:

1. Die Quelle besteht aus fertigen DVR-Streams, die im Vorfeld generiert wurden
2. Die Quelle ist ein Simulationsprozess, der den DVR-Stream in Echtzeit generiert

Der Server wird um eine CGI-ähnliche Schnittstelle zum Aufruf von Simulationsprogrammen erweitert, deren Ergebnisse im DVR-Format gestreamt an den Client geschickt werden. Die eigentlichen Simulations- und Visualisierungsprozesse sind nicht Bestandteil dieses Systems, sondern werden als Moduln eingefügt. Sie werden in den Teilprojekten der Anwendungsebene erstellt. Die Simulationsprogramme können mit veränderlichen Parametern aufgerufen werden, deren Größen der Benutzer über GUI-Elemente verändern kann. Die GUI-Elemente werden mit Hilfe von Parameter-Beschreibungen aus der DVRS-Datei generiert.

Zwischen den DVR-Clients wird eine Schnittstelle für CSCW-Funktionalitäten eingerichtet, über die Synchronisation und Kommunikation abgewickelt werden.

Aus den Diagrammen der drei Anwendungsszenarien erhält man die Architektur des Gesamtsystems, die in *Abbildung 11* zu sehen ist.

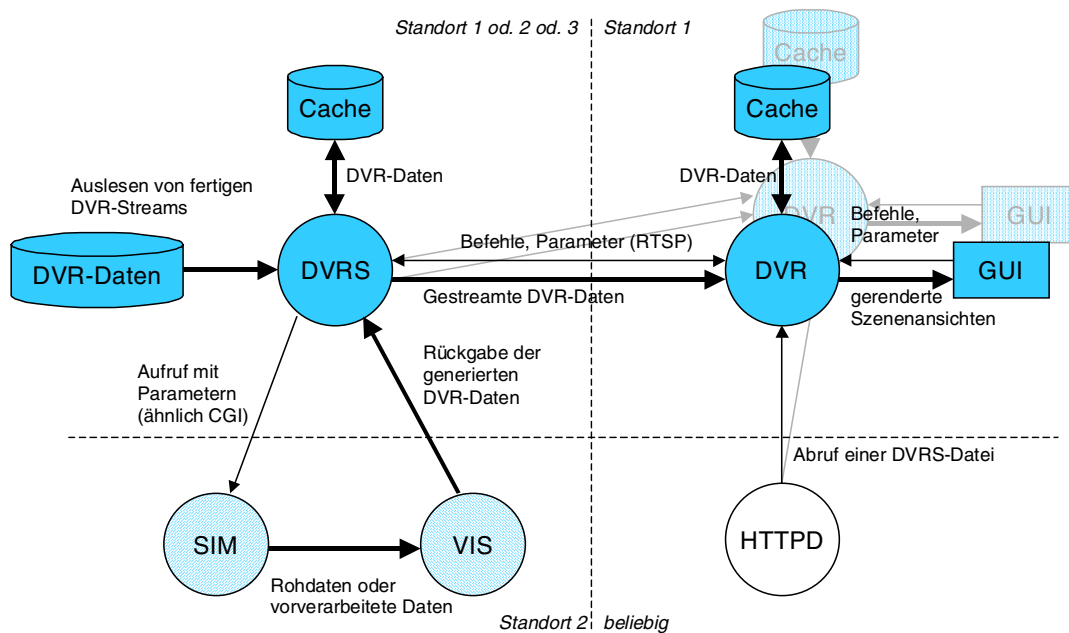


Abbildung 11. Architektur des 3D/VR-Systems.

1.4.3 Teilprojekt 3: „Informationsextraktion und Datenvorverarbeitung“ (Durchführung: ZIB)

Das Volumen der in aktuellen Supercomputer-Simulationen behandelten zeitvarianten Datensätze liegt typischerweise im Terabyte-Bereich. Dies verbietet eine Speicherung der Rohdaten auf Plattenspeichern, ja sogar einen Transfer zu einem anderen Rechner, etwa für Postprocessing-Zwecke. Selbst eine enge Netzkopplung zu einem Postprocessing-System ist bei Vorliegen solcher Datenmengen wegen der damit verbundenen Belastung der Simulationsrechnung durch I/O-Aktivität unpraktisch. Zudem sind Postprocessing-Systeme typischerweise nicht in der Lage, die nötige Hauptspeicherkapazität zu bieten.

zität und die für die Visualisierung derartiger Datenmengen erforderliche Rechenleistung zu erbringen.

Aus diesen Gründen muss die notwendige Information bereits im Hauptspeicher des Supercomputers extrahiert werden. Dies geschieht am besten durch Bibliotheksfunktionen, die dem Anwender zur Verfügung gestellt werden, und die es ihm erlauben, aus den ohnehin im Speicher liegenden Daten die für die Visualisierung notwendigen niedrig-dimensionalen Geometrien zu generieren. Beispiele solcher geometrischen Gebilde sind etwa Teilvolumina, Schnittebenen, Isoflächen, Stromlinien, Streichlinien, Trajektorien, usw.

Die Bibliotheksfunktionen werden in das Simulationsprogramm eingebunden und müssen parallel arbeiten. Durch Verwendung von prozeduralen Zugriffen können generische Varianten geschaffen werden, die von dem jeweils vorliegenden Gittertyp und Datenlayout abstrahieren. Aus Performance-Gründen werden aber auch Funktionen benötigt, die auf den jeweils vorliegenden Gittertyp und das jeweils gewählte Datenlayout zugeschnitten sind. Funktionen letzteren Typs sollen nach Bedarf erstellt werden. Die Bibliothek soll neben der DVR-Ausgabe noch weitere Datenformate unterstützen, so dass alternativ auch andere Viewer verwendet werden können.

Bei der Entwicklung der Funktionsbibliothek wird auf die am ZIB bereits vorliegende umfangreiche Software aufgebaut.

1.4.4 Teilprojekt 4: „Atmosphärische Konvektion“ (Durchführung: IMUK, RRZN, ZIB)

Es handelt sich um Simulationsrechnungen von instationären Strömungsphänomenen im Bereich der atmosphärischen Grenzschicht, z. B. atmosphärische Konvektion und Wirbelschleppen. Dahinter verbergen sich sowohl Probleme aus der Grundlagenforschung als auch aus dem Bereich der angewandten Forschung. So ist die Frage nach den Ursachen spezieller Strömungsorganisationen in atmosphärischer Konvektion immer noch ungelöst. Wirbelschleppen stellen hingegen ein großes praktisches Problem dar. Sie lösen sich von den Tragflügelspitzen landender Flugzeuge und bilden so eine große Gefahr für nachfolgende Flugzeuge. Dieser Gefahr begegnet man zur Zeit durch Festlegung eines zeitlichen Mindestabstandes aufeinanderfolgender Maschinen. Ein genaues Verständnis der Ausbreitung und des Zerfalls dieser Wirbelschleppen kann dazu beitragen, die Landefrequenzen auf großen Verkehrsflughäfen zu erhöhen und ist daher von großem ökonomischen Interesse. Das Institut für Meteorologie und Klimatologie hat eine für Hoch-/Höchstleistungsrechner (Vektorrechner, Parallelrechner) optimierte Simulationssoftware entwickelt. Es handelt sich dabei um die **parallelisierte** Version eines sogenannten **Large-Eddy-Simulationsmodells** (PALM-1). Beispielsweise wird diese Version für den Parallelrechner SGI/Cray T3E auf folgenden Rechnern betrieben:

1. RRZN (Hannover): Entwicklung, Tests.
2. ZIB (Berlin): Produktionsläufe.
3. NIC (Jülich): Skalierungstests für hohe Prozessor-Anzahl (PEs), zukünftig auch Produktionsläufe.

Die derzeit benötigte Rechenzeit beträgt:

- für „große“ Probleme mit 1000 x 1000 x 200 Gitterpunkten (GP) bei Verwendung von 400 PEs: 12 Sekunden je Zeitschritt (mit jeweils 37 GB Ergebnisdaten).
- für „kleine“ Probleme mit 120 x 120 x 80 GP auf 120 PEs: 0,2 Sekunden je Zeitschritt.

Eine typische Anzahl von Zeitschritten lässt sich an einem Wirbelschleppen-Szenario spezifizieren (fällt unter die Rubrik „großes Problem“). Das Phänomen hat in Echtzeit eine charakteristische Dauer von ca. 180 Sekunden. Dies entspricht der Zeitspanne von der Erzeugung der Wirbelschleppen bis zu ihrem endgültigen Zerfall. Die Simulation erfordert bei 400 eingesetzten PEs eine CPU-Zeit von ca. 12 Stunden, d.h. bei einer Rechenschrittweite von 0,05 Sekunden ergeben sich 3600 Zeitschritte, die letztlich zu visualisieren sind. Zur Erreichung des Anfangszustands einer solchen Szenarienrechnung ist eine Echtzeitdauer von ca. 2 Stunden zu simulieren, die derzeit mit einem Rechenzeitbedarf in der

Größenordnung von 500 CPU-Stunden abgeschätzt wird. Dabei wird jedoch nur das Ergebnis des letzten Zeitschritts für die dann folgende Wirbelschleppensimulation weiter verwendet.

Für bisherige Visualisierungen wurden abgespeicherte Ergebnisdatensätzen „kleinerer“ Probleme (160 x 160 x 40 GP) mit Hilfe verschiedener Visualisierungssoftware auf der Basis von UNIRAS, AVS, Express und Visualization Toolkit (VTK – Public-Domain-Software, C++) zwei- oder dreidimensional aufbereitet. 3D-Visualisierungen wurden z. T. über eine mehrstufige Prozesskette durch Konvertierung aus VRML- in DVR-Daten einem 3D-Streaming-Prototypen zugänglich gemacht, mit dem am RRZN/RVS zeitabhängige Abläufe stereoskopisch, dreidimensional navigierbar und animiert präsentiert werden konnten. Aufgrund der hohen Volumina der angestrebten Ergebnisse aktueller Produktionsrechnungen (1000 x 1000 x 200 GP) kommt die Speicherung von Rohdaten eines kompletten Rechenlaufs mit mehreren tausend Zeitschritten nicht in Frage. Allein für eine einzelne, auszugebende, skalare Variable würde der entsprechende Plattenplatzbedarf im Terabyte-Bereich liegen. Es liegt hiermit also der typische Fall vor, bei dem direkt im Hauptspeicher des Supercomputers eine Datenreduktion geschehen muss.

Abhängig von der jeweils gewünschten Visualisierung erfolgt die Realisierung:

(A) Kooperation IMUK mit RRZN

1. Mit Hilfe eigener Erweiterungen des Simulationsprogramms sollen dreidimensionale „Particle Traces“ aus dem Strömungsfeld errechnet und im 3D-Graphikformat DVR abgespeichert oder direkt zum Visualisierungsklient (3D/VR-System des RRZN: „DocShow-VR“ bzw. Weiterentwicklung in Teilprojekt 2) übertragen werden. Zur Generierung der Partikeldaten im DVR-Format wird die am RRZN bereits vorhandene Klassenbibliothek „OutputDVR“ verwendet.
2. Mit Hilfe von Erweiterungen auf Basis der VTK-Software sollen „Slicer“ und „Isosurface“ aus den skalaren Ergebnisgrößen (z. B. Temperatur, Luftfeuchtigkeit) errechnet werden. Da sich die 3D-Geometrien aus den jeweils lokal auf dem Rechenknoten liegenden Daten generieren lassen, ist eine Parallelisierung möglich. Weiterhin muss die VTK-Software um ein Modul zur direkten Erzeugung von DVR-Daten ergänzt werden. Dies geschieht auf der Grundlage der Klassenbibliothek „OutputDVR“.

(B) Kooperation IMUK mit ZIB

Die gegenwärtige Kooperation zur Optimierung des Laufzeitverhaltens und der Skalierbarkeit der Simulation wird fortgeführt. Mit Hilfe von Erweiterungen des Simulationsprogramms auf Basis der am ZIB entwickelten Software (Amira) sollen symbolische Repräsentationen aus den skalaren und vektoriellen Ergebnisgrößen errechnet werden. Das dabei generierte DVR-Format wird mit dem in Teilprojekt 2 entwickelten Visualisierungssystem weiterverarbeitet.

1.4.5 Teilprojekt 5: „Molekulardynamische Simulation“ (Durchführung: IfT, RRZN, ZIB)

Die Verläufe der Wärmekapazitäten einiger fluorierter Kohlenwasserstoffe weisen im Gasgebiet ein ungewöhnliches Verhalten auf, das theoretisch bisher noch nicht verstanden ist. Für Difluormethan nimmt die isochore Wärmekapazität auf einer Isobaren mit abnehmender Temperatur zunächst ab, durchläuft ein Minimum und steigt in der Nähe der Taulinie wieder stark an. Dieser Anstieg ist auch bei tiefen Temperaturen ausgeprägt. Im Vergleich dazu sind die Verläufe für 1,1,1,2-Tetrafluorethan ähnlich, aber deutlich weniger stark ausgeprägt. Insbesondere weisen die Isobaren bei niedrigen Temperaturen keine Minima auf. Es wird vermutet, dass dieser Effekt auf die Bildung von Clustern zurückzuführen ist.

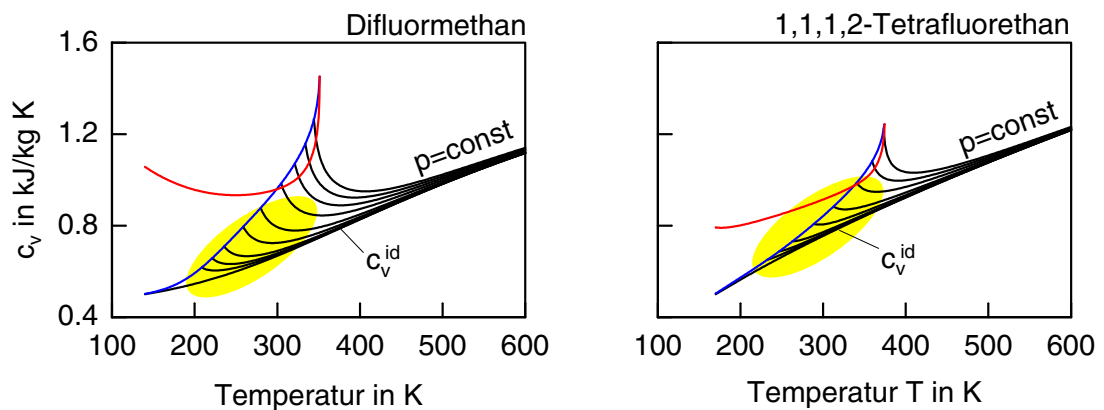


Abbildung 12. Die isochore Wärmekapazität im Gasgebiet von Difluormethan und 1,1,1,2-Tetrafluorethan.

Da Experimente im Gasgebiet nahe der Taulinie aufgrund von unvermeidbaren Adsorptionseffekten an den Messzellenwänden nur schwierig durchzuführen sind, soll der Effekt in diesem Projekt mit molekulardynamischen Simulationen untersucht werden. In einer molekulardynamischen Simulation werden die Newton'schen Bewegungsgleichungen für etwa 100–10000 Moleküle mit einem vorzuzugenden Modell für das Wechselwirkungspotenzial zwischen den Molekülen numerisch gelöst. Mit Methoden der statistischen Mechanik können thermodynamische Eigenschaften (z. B. Druck, innere Energie, Wärmekapazitäten) und Transporteigenschaften (z. B. Diffusionskoeffizient, Viskosität) berechnet werden. Zudem kann das Verhalten auf molekularer Ebene analysiert werden, da Molekülkoordinaten und -geschwindigkeiten direkt zugänglich sind.

Neben der numerischen Auswertung der Simulationen ist eine Visualisierung als „Trickfilm“ zur Deutung der Ergebnisse erforderlich. Insbesondere die Clusterbildung, eines der Hauptuntersuchungsziele, lässt sich numerisch nur z. T. analysieren. Falls sich besondere Clusterstrukturen (z. B. Ketten oder Ringe) ausbilden sollten, wären diese visuell leicht zu identifizieren. Eine Online-Visualisierung wäre auch bei der Beobachtung der Gleichgewichtseinstellung nach dem Start der Simulation von der Startkonfiguration hilfreich.

Molekulardynamische Simulationen sind sehr rechenzeitintensiv. Der größte Anteil der Rechenzeit wird für die Berechnung der Kräfte zwischen den Molekülen in jedem Zeitschritt benötigt. Da die Kraftberechnung gut parallelisiert werden kann, ist dieses Projekt für massiv parallele Großrechner geeignet. Vorgesehen sind Simulationen für vier verschiedene Molekülmodelle. Pro Modell sollen fünf Isothermen mit je fünf bis zehn Zustandspunkten simuliert werden. Da eine Simulation ca. 10 CPU-Stunden erfordert, werden insgesamt 1000–2000 CPU-Stunden benötigt. Der Arbeitsspeicherbedarf beträgt 200 MB. Der Speicherbedarf auf Festplatte beläuft sich auf ca. 2 GB temporär für Zwischenergebnisse und 100 MB für permanent benötigte Dateien.

Die Software wurde auf eine SGI/Cray T3E portiert und effizient parallelisiert. Beispielsweise benötigte ein Rechner mit 16 Prozessoren für ein Partikelsystem mit 1372 Teilchen und 2000 Zeitschritte 45 CPU-Sekunden, es können also ca. 40 Zeitschritte je Sekunde berechnet werden. Eine On-the-fly-Visualisierung bietet sich daher an, wobei üblicherweise nur jeder 5. bis 10. Zeitschritt dargestellt wird. Der Parallelrechner des ZIB würde die benötigte Rechenleistung bereitstellen, um flüssige Animationen sowie noch höhere Partikelanzahl zu ermöglichen. Ein hoher Parallelisierungsgrad bei hoher Teilchenanzahl wurde bereits abgeschätzt.

Die resultierenden Ergebnisdaten – Partikel-Koordinaten und ggf. -Orientierungen – können mit geringfügiger Nachbereitung direkt – z. B. in Form von Kugeln oder Dipol-Symbolen – visualisiert werden. Dies wurde anhand von Test-Ergebnisdaten und daraus generierten vorbereiteten DVR-Graphikdateien mit Hilfe des 3D-Streamingsystems am RRZN erfolgreich demonstriert.

Abhängig von der jeweils gewünschten Visualisierung erfolgt die Realisierung:

(A) Kooperation IFT mit RRZN

Im Projekt soll – unterstützt durch das RRZN – eine Online-Visualisierung auf der Basis des im RRZN entwickelten 3D/VR-Systems „DocShow-VR“ sowie deren Weiterentwicklung in Teilprojekt 2 ermöglicht werden. Dazu muss die Generierung der 3D-Geometrie-Daten in das Simulationsprogramm und die Ausgabe dieser Daten in den Streamingserver-Mechanismus integriert werden.

(B) Kooperation IFT mit ZIB

Das ZIB unterstützt den Anwender im Bedarfsfall bei der Optimierung des Laufzeitverhaltens und der Skalierbarkeit der Simulation, und in der Entwicklung geeigneter Algorithmen zur Feature-Extraktion/Visualisierung.

2 Durchführung des Projekts

2.1 Personal

An dem Projekt waren die folgenden Personen beteiligt:

- Projektleitung
 - Dr. Stephan Olbrich (RRZN)
 - Hans-Christian Hege (ZIB)
- Wissenschaftliche Mitarbeiter
 - Alexander von Berg (RRZN)
 - Hubert Busch (ZIB)
 - Dr. Karsten Chmielewski (RRZN)
 - Arnd Diekmann (RRZN)
 - Dirk Ditschke (ISEB)
 - Ralf Einhorn (RRZN)
 - Oliver Göthel (ISEB)
 - Sebastian Heidl (ZIB)
 - Steffen Heinze (RRZN)
 - Dr. Fritz Hüsemann (RRZN)
 - Nils Jensen (RRZN)
 - Dr. Karsten Meier (IFT)
 - André Merzky (ZIB)
 - Dr. Siegfried Raasch (IMUK)
 - Dr. Manfred Stolle (ZIB)
 - Tino Weinkauff (ZIB)
- Designerin
 - Anne-Kathrin Ittmann (RRZN)
- Studentische Mitarbeiter
 - Matthias Heyder (ZIB)
 - Harald Knipp (ZIB)
 - Ralf Staudemeyer (ZIB)
 - Stefan Zwierlein (ZIB)

2.2 Publikationen

Teilergebnisse des Projekts, entsprechende Vorarbeiten (seit der Projektgenehmigung durch den DFN-Verein) sowie Arbeiten, in denen Projektergebnisse angewandt oder weiterentwickelt wurden, wurden wie folgt publiziert (in umgekehrter chronologischer Reihenfolge):

Veröffentlichungen und Vorträge

1. Meier, K., Holz knecht, C., Kabelac, S., Olbrich, S., Chmielewski, K.: *3D Visualization of Molecular Simulations in High Performance Parallel Computing Environments*. FOMMS 2003 – Foundations of Molecular Modeling and Simulation. 06.-11.07.2003, Keystone, Colorado, USA.
2. Olbrich, S., Jensen, N.: *Lessons Learned in Designing a 3D Interface for Collaborative Inquiry in Scientific Visualization*. HCI 2003 – International Conference on Human-Computer Interaction. Kreta, Griechenland, 22.-27.06.2003.
3. Jensen, N., Seipel, S., Nejd, W., Olbrich, S.: *CoVASE: Collaborative Visualization for Constructivist Learning*. CSCL Conference 2003. Bergen, Norway, 14.-18.06.2003.
4. Olbrich, S., Einhorn, R., Jensen, N., Chmielewski, K.: *Aufbau und Entwicklung von Multimedia-Infrastrukturen und -Services für innovative E-Learning-Anwendungen*. 17. DFN-Arbeitstagung über Kommunikationsnetze, Düsseldorf, 11.-13.06.2003. Wird in GI-Edition – Lecture Notes in Informatics (LNI), Köllen Verlag Bonn, erscheinen.
5. Meier, K., Kabelac, S.: *Decay of Time-correlation Functions and Molecular Transport Mechanisms in the Lennard-Jones Model Fluid from Molecular Dynamics Simulations*, Posterpräsentation, Thermodynamics 2003, 9.-11.4.2003, Cambridge, United Kingdom.
6. Ditschke, D., Göthel, O.: *Parallelisierung und Visualisierung für morphodynamische Berechnungen*, 4. FZK-Kolloquium „Küsten- Morphodynamik und Küstenschutzwerke“, Hannover 2003.
7. Ditschke, D., Markofsky, M., Zielke, W.: *Modellierung der Sedimentation in Tidehäfen*, 4. FZK-Kolloquium „Küsten- Morphodynamik und Küstenschutzwerke“, Hannover 2003.
8. Pralle, H., Hege, H.-C., Olbrich, S.: *Supercomputer und Visualisierung*. RRZN-BI (Benutzerinformation) 363, 03.12.2002.
9. Endebroch, P., et al: *Tag der Forschung – kurze Nachlese*. RRZN-BI 363, 03.12.2002 (incl. Posterbeilage).
10. Ditschke, D. Göthel, O.: *Harbour Sedimentation and Visualisation of High-performance Computing*, Vortrag beim Telemac User-Club, Karlsruhe 2002.
11. Endebroch, P., et al: *Das RRZN beim „Tag der Forschung“*. RRZN-BI 362, 29.10.2002.
12. Jensen, N., Olbrich, S., Pralle, H., Raasch, S.: *An Efficient System for Collaboration in Tele-Immersive Environments*. 4. Eurographics Workshop on Parallel Graphics and Visualization. Blaubeuren, 09.-10.09.2002.
13. Chmielewski, K.: *Software-Dokumentation mit dem DocBook DTD*, Vortrag auf dem HLRN Fachberater-Treffen 26.6.2002.
14. Hege, H.-C.: *Visualization and Supercomputing: From Atoms to Galaxy Clusters*. ISC2002 International Supercomputer Conference, 20.-22. Juni 2002, Heidelberg (eingeladener Vortrag, 22.06.2002).
15. Hege, H.-C.: *Visual Data Analysis*. PARA'02 Conference on Applied Parallel Computing, 15.-18. Juni 2002, Espoo, Finnland (eingeladener Vortrag, 18.06.2002).
16. Olbrich, S., Weinkauff, T., Merzky, A., Knipp, H., Hege, H.-C., Pralle, H.: *Lösungsansätze zur Visualisierung im „High-Performance Computing and Networking“-Kontext*. 16. DFN-Arbeitstagung über Kommunikationsnetze, Düsseldorf, 21.-24.05.2002. In: von Knop, J., Haverkamp, W. (Hrsg.): *Zukunft der Netze – Die Verletzbarkeit meistern*. GI-Edition – Lecture Notes in Informatics (LNI), P-17, Köllen Verlag Bonn, 2002, S. 269-280.

17. Olbrich, S.: *Präsentation auf der CeBIT 2002 – Supercomputing und Virtual Reality im Gigabit-Wissenschaftsnetz G-WiN*. RRZN-BI 358, 30.04.2002.
18. Raasch, S.: *An den Grenzen des Machbaren*. Simulationen turbulenter atmosphärischer Störungen benötigen die weltweit leistungsfähigsten Rechner – Technische Herausforderungen und Anwendungen. In: Uni-Magazin Hannover 1|2–2002.
19. Busch, H., et al: *Teleimmersion – Erfahrungen mit der Gigabit-Strecke zwischen Berlin und Hannover*. DFN-Mitteilungen, Heft 58, März 2002.
20. Olbrich, S., Pralle, H., Raasch, S., Meier, K.: *Tele-immersive Visualisierung mittels 3D-Streamingverfahren im Gigabit-Wissenschaftsnetz*. DFN-Mitteilungen, Heft 57, November 2001.
21. Olbrich, S.: *3D-Streaming – Basistechnologie zur Visualisierung im „High-Performance Computing and Networking“*. Vortrag im Max-Planck-Institut für biophysikalische Chemie (Göttingen), im Rahmen einer Vortragsreihe der Mathematischen Fakultät der Georg-August-Universität Göttingen und der Gesellschaft für wissenschaftliche Datenverarbeitung Göttingen (GWDG), 12.11.2001.
22. Olbrich, S., Raasch, S.: *Simulation und 3D-Visualisierung – Exponat auf dem „Tag der Forschung“ war großer Erfolg*. RRZN-BI 353, 20.11.2001.
23. Olbrich, S., Raasch, S.: *Simulation und 3D-Visualisierung – ein Exponat am „Tag der Forschung“*. RRZN-BI 352, 23.10.2001 (incl. Posterbeilage).
24. Olbrich, S.: *Tele-immersive Hochleistungsvisualisierung auf der Basis von Real-Time Streaming komplexer, virtueller 3D-Szenen-Sequenzen*. Vortrag im Rahmen einer Fachtagung der Informationstechnischen Gesellschaft (ITG) des VDE zum Thema 3D-Bewegtbilddarstellung, Heinrich-Hertz-Institut Berlin, 12.10.2001.
25. Olbrich, S., Pralle, H.: *A Tele-Immersive, Virtual Laboratory Approach based on Real-Time Streaming of 3D Scene Sequences*. Proceedings of ACM Multimedia Conference 2001, Ottawa, Canada, 30.09.–05.10.2001.
26. Olbrich, S., Pralle, H., Raasch, S.: *Using Streaming and Parallelization Techniques for 3D Visualization in a High-Performance Computing and Networking Environment*. In: Hertzberger, B., Hoekstra, A., Williams, R.: *High-Performance Computing and Networking*. 9th International Conference, HPCN Europe 2001, Amsterdam, The Netherlands, 25.–27.06.2001, Proceedings. Lecture Notes in Computer Science, Vol. 2110, Springer-Verlag.
27. Olbrich, S.: *3D-Hochleistungsvisualisierung in einem verteilten System*. Vortrag im Rahmen eines IGC-Meetings am 05.–06.04.2001 im RRZN.
28. Olbrich, S.: *Anwendung der Tele-Immersion in Weitverkehrsnetzen*. RRZN-BI 346, 20.02.2001.
29. Olbrich, S.: *Verteilte Visualisierung von ingenieurwissenschaftlichen Simulationsergebnissen*. 3. Kolloquium des Internationalen Zentrums für Computergestützte Ingenieurwissenschaften (ICCES), Hannover, Leibnizhaus, 26.01.2001. In: Wriggers, P. (Hrsg.): *ICCES-Kolloquium '01 – Geometrische Modellierung / Visualisierung / Software*. ISBN 3-935732-04-X.
30. Olbrich, S.: *Zur Integration von Bewegtbild-Medien in Informationssysteme – vom Video bis hin zu 3D / Virtual Reality im Internet*. Vortrag, Fachhochschule Hannover, Fachbereich Informatik, 03.11.2000.
31. Olbrich, S.: *Ein leistungsfähiges System zur Online-Präsentation von Sequenzen komplexer virtueller 3D-Szenen*. Dissertation, Fachbereich Elektrotechnik und Informationstechnik, Universität Hannover, 02.05.2000.
32. Olbrich, S., Pralle, H., Raasch, S.: *Efficient volume visualization using parallelization and 3D streaming techniques in a high-performance computing and networking scenario*. Technical Report, 26.04.2000.
33. Olbrich, S., von Berg, A., Einhorn, Heineke, S., R., Ittmann, A., Sander-Beuermann, W., Pralle, H.: *DFN-Expo – Entwicklung und Demonstration fortgeschrittener Online-Präsentationstechnologien*. Abschlussbericht, 23.02.2000.

34. Olbrich, S.: *Streaming-Technologie in Anwendungsszenarien der Virtuellen Realität*. DFN-Symposium 2000, ZIB, Berlin, 14.02.2000.
35. Olbrich, S., Pralle, H.: *Multimedia- und Virtual-Reality-Technologie für die wissenschaftliche Visualisierung*. In: Buziek, G. et al (Hrsg.): *Dynamische Visualisierung*. Springer-Verlag, 2000.
36. Olbrich, S.: *3D/VR-Technologie (Tele-Immersion) zur Präsentation wissenschaftlicher Ergebnisse*. Workshop „Visualisierung / Bildgebende Verfahren“, Universität Hannover, 26.10.1999.

Rundfunkbericht

- Deutschlandfunk: Bericht vom DFN-Symposium, 18.02.2000, mit Original-Ton von Herrn Olbrich zum Thema seines Vortrags am 14.02.2000 (siehe oben).

2.3 Präsentationen

Präsentationen

- BMBF/DFN-Stand auf der CeBIT 2002, Hannover, März 2002.
- Am „Tag der Forschung 2001“ der Universität Hannover (04.11.2001) wurden Anwendungsbeispiele aus dem Meteorologie-Bereich demonstriert. Dabei wurde eine mobile 3D-Stereogroßbildprojektion (2 m x 1,5 m) verwendet. Eine 3D-Grafikworkstation (Windows 2000) wurde über ein LWL-Faserpaar (Single-Mode) und ein Medienwandler-Paar (Single-Mode/Multi-Mode) an das Gigabit-Ethernet-Pilotnetz angeschlossen.
- Am „Tag der Forschung 2002“ der Universität Hannover (03.11.2002) wurden Anwendungsbeispiele aus dem Molekulardynamik-Bereich demonstriert.
- Einweihung des Norddeutschen Verbunds für Hoch- und Höchstleistungsrechnen (HLRN), 13.11.2002. (<http://www.hlrn.de/presse/einweihung/> – incl. Videomitschnitt)
- Etliche Demonstrationen im 3D-Präsentationsraum/Multimedia-Labor des RRZN.

Simulation und 3D-Visualisierung – Exponat auf dem „Tag der Forschung“ war großer Erfolg

Im Jahr 2001 fand der Tag der Forschung der Universität Hannover im Rahmen des hannoverschen Festes der Wissenschaften statt, in dem sich vom 02.–04.11.2001 verschiedene Hochschulen und Fachhochschulen präsentierten. Am Sonntag, den 4. November 2001 von 11 bis 17 Uhr zeigten 30 Institute der Universität Hannover im Lichthof des Uni-Hauptgebäudes aktuelle wissenschaftliche Ergebnisse unter dem Motto „Forschung zum Anfassen“. Rund 4000 Besucher ließen sich in die vielfältigen Forschungsprojekte einführen.

Unter anderen als Zwischenergebnis des BMBF-geförderten DFN-Projekts „Anwendung der Tele-Immersion in Weitverkehrsnetzen“ wurden in einem Exponat auf dem Tag der Forschung Anwendungsbeispiele des Instituts für Meteorologie und Klimatologie mit Hilfe eines leistungsfähigen, unter Windows 2000 betriebenen 3D-PCs gezeigt. Der PC mit Dual-Pentium 4 (1,7 GHz) und stereofähiger 3D-Grafikkarte Wildcat 5110 (3Dlabs) griff dabei über eine Lichtwellenleitung (Gigabit-Ethernet) auf einen 3D-Streamingserver im RRZN zu und stellte entsprechend vorbereitete „3D-Filme“ auf einer 3D-Stereoprojektion dar. Die Besucher konnten diese dann mit Hilfe von Polfilterbrillen räumlich betrachten und sich in diesen Szenarien virtuell frei bewegen, z. B. mit Hilfe einer 3D-Maus („Space Mouse“). Darüber hinaus wurde der fachliche Hintergrund auf einer Flash-Animation illustriert:

- <http://www.mml.uni-hannover.de/flash/tdF2001a.swf>



Abbildung 13. Dr. Raasch (Institut für Meteorologie und Klimatologie) und Dr. Olbrich (RRZN) im Gespräch mit Bundesministerin Bulmahn (BMBF) – am 4. November 2001 im Lichthof der Universität Hannover.



Abbildung 14. Mobile 3D-Stereo-Großbildprojektion (2 m x 1,5 m) und 3D-Workstation (Dell Precision 530, Dual-Pentium 4 Xeon, Wildcat II 5110) auf dem gemeinsamen Stand des Instituts für Meteorologie und Klimatologie und des RRZN, Gigabit-Ethernet-Verbindung zum 3D-Streamingserver im RRZN.

CeBIT 2002

Auf der CeBIT 2002 wurde in Hannover vom 13. bis 20. März auf dem Stand des BMBF eine Projektpräsentation unter dem Motto „Supercomputing und Virtual Reality im Gigabit-Wissenschaftsnetz G-WiN“ durchgeführt. Diese Präsentation wurde durch die Mitarbeiter Stephan Olbrich (RRZN) sowie André Merzky und Tino Weinkauff (ZIB) realisiert und durch Mitarbeiter des Instituts für Meteorologie und Klimatologie sowie des Instituts für Thermodynamik unterstützt.

Wir zitieren hier einen Artikel aus der RRZN-BI (Benutzerinformation):

Olbrich, S.: *Präsentation auf der CeBIT 2002 – Supercomputing und Virtual Reality im Gigabit-Wissenschaftsnetz G-WiN*. RRZN-BI 358, 30.04.2002.

In dem vom DFN-Verein aus Mitteln des BMBF (Bundesministerium für Bildung und Forschung) geförderten Projekt „Anwendung der Tele-Immersion in Weitverkehrsnetzen“ werden räumlich verteilte Szenarien zur Kopplung hochkomplexer Simulationen und interaktiver 3D-Visualisierungen entwickelt und erprobt:

- Kooperative Betrachtung,
- Navigierbare 3D-Filme,
- Online-Visualisierung,
- Simulationssteuerung.

Dabei wird im G-WiN eine transparente WDM-Strecke (Wave Division Multiplex) zwischen Berlin (Projektpartner: Konrad-Zuse-Zentrum für Informationstechnik Berlin – ZIB) und Hannover (RRZN/RVS) zur Datenkommunikation genutzt.

Als fortgeschrittene Basistechnologien kommen zum Einsatz:

- Parallelisierte Aufbereitung von Ergebnisdaten in 3D-Szenen,
- 3D-Streamingverfahren über Gigabit-Ethernet / Jumboframes.

Auf der CeBIT 2002 wurden vom 13. bis 20. März auf dem Stand des BMBF Anwendungsbeispiele aus so genannten „Grand Challenge“-Simulationen – Ergebnisse aufwändiger Rechnungen auf massiv-parallelen Supercomputern – über eine 3D-Stereo-Großbildprojektion präsentiert:

- Turbulente atmosphärische und ozeanische Strömungen
(Projektpartner: Institut für Meteorologie und Klimatologie, Universität Hannover),
- Molekulardynamik
(Projektpartner: Institut für Thermodynamik, Universität der Bundeswehr Hamburg).

Für die Präsentation der verteilten 3D-Streaming-Anwendung wurde das Gigabit-Ethernet-Netzwerk bis zum Messestand erweitert und vor Ort das PC-basierte, mobile 3D-Stereoprojektionssystem¹ sowie eine Rückprojektionsscheibe² des RRZN verwendet.

Der größte Teil der Verlängerung zum Messegelände wurde dabei durch eine der Glasfaserstrecken bereitgestellt, die vom RRZN zum Europahaus auf dem ehemaligen Expo-Ost-Gelände führen und künftig zur hochratischen Anbindung der dort befindlichen Universitäts- und Fachhochschulinstiute sowie des Learning Lab Lower Saxony (L3S) dienen sollen. Für die Strecke vom Europahaus zum BMBF-Stand auf dem Messegelände wurde für den CeBIT-Zeitraum ein Single-Mode-Glasfaserpaar gemietet und über Medienwandler (Multimode/Singlemode) des RRZN an die vorhandenen Endgeräte (Europahaus: WDM-System, BMBF-Stand: Gigabit-Ethernet-Interface im PC) angepasst.



Abbildung 15. DFN-Geschäftsführer Dr. Klaus-Eckart Maass, Bundesforschungsministerin Edelgard Bulmahn, Projektleiter Dr. Stephan Olbrich (RRZN) und Messe-Vorstandsmitglied Sepp D. Heckmann (von links nach rechts) betrachten Beispiele für dynamische, komplexe 3D-Visualisierungen, die über eine Gigabit-Ethernet-Verbindung zum BMBF-Stand übertragen und – interaktiv navigierbar – stereoskopisch präsentiert wurden.

(Ende des Zitats aus der RRZN-BI 358)

Zur CeBIT 2002 wurde außerdem gemeinsam mit dem DFN ein Flyer³ erstellt, der am Ausstellungsstand verteilt wurde.

1. Siehe auch in RRZN-BI 351 (September 2001):
Olbrich, S.: „Neuer 3D-PC im Multimedia-/Virtual-Reality-Labor“.
(http://www.rrzn.uni-hannover.de/BIs/Jahrgang2001/BI351/bi351_13.html)
2. Siehe auch in RRZN-BI 312 (Februar 1998):
Olbrich, S.: „3D-Stereopräsentation wissenschaftlicher Arbeitsergebnisse am RRZN“.
(<http://www.rrzn.uni-hannover.de/BIs/Jahrgang98/BI312/bi312-8.html>)
3. Download einer korrigierten bzw. aktualisierten Version:
<http://www.rvs.uni-hannover.de/projekte/tele-immersion/cebit02.pdf>

Schautafel auf dem BMBF-Stand, CeBIT 2002



**Bundesministerium
für Bildung
und Forschung**

Supercomputing und Virtual Reality im Gigabit-Wissenschaftsnetz G-WiN

Im DFN-Projekt *Anwendung der Tele-Immersion in Weitverkehrsnetzen* werden räumlich verteilte Szenarien zur Kopplung hochkomplexer Simulationen und interaktiver 3D-Visualisierung entwickelt und erprobt:

- ▶ Kooperative Betrachtung
- ▶ Online-Visualisierung
- ▶ Navigierbare 3D-Filme
- ▶ Computational Steering

Dabei wird im G-WiN eine transparente WDM-Strecke zwischen Berlin und Hannover mit Datenraten im Gigabit/s-Bereich genutzt. Basis:

- ▶ Gigabit-Ethernet im WAN
- ▶ 3D-Streaming-Technologie

Beispiele für „Grand Challenge“-Simulationen:

- ▶ Turbulente Strömungen
- ▶ Molekulardynamik



3D-Navigation,
Tele-Pointer,
Video-Conferencing,
Annotationen



3D-Daten

WWW,
3D-Streaming,
Supercomputer



Verein zur Förderung eines Deutschen Forschungsnetzes e. V. (DFN-Verein)
Regionales Rechenzentrum für Niedersachsen (RRZN)/
Lehrgebiet Rechnernetze und Verteilte Systeme (RVS), Universität Hannover
Konrad-Zuse-Zentrum für Informationstechnik Berlin (ZIB)
Institut für Meteorologie und Klimatologie (IMUK), Universität Hannover
Institut für Thermodynamik (IfT), Universität der Bundeswehr Hamburg
<http://www.rvs.uni-hannover.de/projekte/tele-immersion/>

BMBF MESSE

Einweihung des Norddeutschen Verbunds für Hoch- und Höchstleistungsrechnen (HLRN)

Anlässlich der feierlichen Inbetriebnahme des HLRN durch Minister der sechs beteiligten Bundesländer wurden am 13.11.2002 in Anwesenheit der Presse Anwendungsszenarien dieses Hochleistungsrechners dargelegt. In einem Vortrag von Dr. Stephan Olbrich wurden der Öffentlichkeit Resultate aus den Teilprojekten 4–6 präsentiert (siehe *Abbildung 16*).

Das Grußwort des Berliner Senators für Wissenschaft, Forschung und Kultur, Dr. Thomas Flierl, wurde über eine H.323-basierte Videokonferenz (vgl. *Abschn. 3.1.1*) live übertragen. Von der gesamten Veranstaltung liegt ein Videomitschnitt vor, der im WWW über den Real-Videoserver des RRZN abgerufen werden kann¹. Am Rande sei erwähnt, dass die medien- und kommunikationstechnische Infrastruktur des Multimedia-Seminarraums des L3S, in dem die Veranstaltung stattfand, unter der Federführung des RRZN konzipiert und realisiert wurde. Dabei flossen u. a. die Erfahrungen und Anforderungen aus diesem Teleimmersionsprojekt sowie aus den Pilotinfrastrukturen des RRZN bezüglich Gigabit-Ethernet, Video- und Audiotechnik, Bild- und Tonregie, Videoconferencing, 3D-Rückprojektionstechnik und netzverteilter 3D-Visualisierung ein.



Abbildung 16. HLRN-Einweihung am 13.11.2002 im Learning Lab Lower Saxony (L3S): Vortrag und 3D-Demonstration von Dr. Stephan Olbrich (linkes Bild), Inbetriebnahme durch Minister der beteiligten Bundesländer (rechtes Bild).

1. <http://www.hlrn.de/presse/einweihung/>

3 Ergebnisse des Projekts

3.1 Teilprojekt 1 – Gigabit-Netz und Systemtechnik

3.1.1 Video-Verbindung

Im Projektantrag war zunächst die Verwendung von hochqualitativer Videoübertragung mit den bei den Projektteilnehmern in Berlin und Hannover vorhandenen FORE Streamrunner Codecs vorgesehen. Hiermit ist unter Verwendung von Motion-JPEG-Kompression bei voller PAL-Auflösung (720 x 576 Pixel) und 25 Bildern pro Sekunde eine Übertragungsrate von etwa 25 Mbit/s erforderlich. Parallel wird ein nicht komprimiertes Audiosignal übertragen. Mit der zugrunde liegenden ATM-Übertragung ergibt sich annähernd TV-Qualität bei einer nicht wahrnehmbaren Verzögerung.

Neben der eingeschränkten bzw. kostenintensiven Verfügbarkeit des ATM-Dienstes besteht ein weiterer Nachteil in der Beschränkung auf Punkt-zu-Punkt-Verbindungen.

Aus diesen Gründen wurde für das Projekt nach einer Alternative zur ursprünglich geplanten ATM/M-JPEG-Lösung gesucht.

Allgemeine Kriterien

Da sich – wie sich auch hier zeigt – die zur Verfügung stehenden Konferenz-Systeme abhängig von Entwicklungsstand und Netzinfrastruktur ändern können, betrachten wir einen „Black-Box“-Ansatz: Das verwendete System soll sich in die Arbeitsumgebung möglichst nahtlos als Modul integrieren lassen. Die im Projektantrag angesprochene Möglichkeit der Einbindung des Videobildes in 3D-Szenen bleibt somit bestehen. Als Schnittstellen dienen die analogen Audio- und Videosignale.

Grundsätzlich ist IP-basierte Video-Übertragung derzeit aufgrund fehlender Quality-of-Service (QoS) Mechanismen nur auf Basis eines Best-Effort-Ansatzes realisierbar – weder die Übertragungszeit noch die Verlustrate der UDP-Pakete können garantiert werden.

Alternativen

Aufgrund der angestrebten hohen Übertragungsqualität kommen Software-Lösungen nicht in Betracht. Hardware-basierte MPEG-1- und -2-Lösungen für IP sind verfügbar, konnten bisher aber lediglich in lokalen Netzen störungsfrei eingesetzt werden und weisen eine Latenz von über einer Sekunde auf, was für die hier geplanten interaktiven Szenarien zu hoch ist.

Die Wahl fiel daher auf H.323-Set-Top-Geräte. Die geringe Verzögerungszeit ermöglicht auch ungeübten Konferenz-Teilnehmern eine gewohnte Kommunikation. Die bei Datenraten bis zu 768 kbit/s erreichte Bild- und Tonqualität ist akzeptabel. Neben der Reduzierung der Auflösung auf das CIF-Format (352 x 288 Pixel, d. h. ½ PAL-Auflösung) muss auch eine Verringerung der Bildrate auf unter 25 Bilder pro Sekunde in Kauf genommen werden. Diese wird – wie auch die Bildschärfe – dynamisch angepasst.

Für das Projekt sollten zunächst homogene Endgeräte eingesetzt werden, in diesem Fall Polyspan Viewstations. Diese wurden im RRZN erfolgreich getestet. Die Geräte zeichneten sich durch eine leichte Handhabung und eine robuste Übertragung aus. Neben dem Gerät ist lediglich ein Ausgabegerät für Bild und Ton notwendig. Hierfür wird üblicherweise ein TV-Monitor genutzt, das Bild kann aber bei Bedarf auch auf entsprechende Hardware der Visualisierungsrechner ausgegeben werden und somit in 3D-Szenen integriert werden.

Neben der Verwendung des IP-Netzes ermöglicht das H.323-Verfahren aber auch die Durchführung von Mehrpunkt-Konferenzen. Eine hierfür nötige „kleine“ Multipoint Conference Unit (MCU) sowie auch ein Gatekeeper (GK) wurden im RRZN beschafft und installiert. Hiermit wurden auch Konferenzen zwischen allen vier Standorten der Projektteilnehmer (ZIB, IMUK, IfT, RRZN) ermöglicht.

In Hannover (RRZN) und Berlin (ZIB) wurden H.323-Geräte (Polyspan Viewstation) beschafft. Je eine Viewstation wurden vom RRZN an das Institut für Meteorologie und Klimatologie (Universität Hannover) sowie das Institut für Thermodynamik (Universität der Bundeswehr Hamburg) für den

gesamten Projektzeitraum ausgeliehen. Zwischen Hannover und Hamburg sowie zwischen Hannover und Berlin wurden Videokonferenzen erfolgreich durchgeführt. Insbesondere im HLRN-Kontext wurden regelmäßige (wöchentliche) Videokonferenzen eingeführt (siehe *Abbildung 17*), daneben wurde diese Infrastruktur für Sitzungen des NALWR (Niedersächsischer Arbeitskreis der Leiter der Wissenschaftlichen Rechenzentren) sowie für Projektbesprechungen verwendet. Darüber hinaus wurde in einer vom RRZN mit herausgegebenen Ausschreibung des Niedersächsischen Wissenschaftsministeriums im Jahr 2001 begonnen, die niedersächsischen Hochschulen mit homogenen, H.323-basierten Videokonferenz-Geräten (Polyspan Viewstation) auszustatten. Begleitend wurde dabei eine „größere“ MCU/GK-Konfiguration im RRZN installiert, um in diesem niedersächsischen Rahmen – sowie auch hier angesiedelten Kooperationsprojekten wie z. B. dem *Learning Lab Lower Saxony (L3S)* im internationalen Kontext (Stanford University / USA, Stockholm, Uppsala / Schweden) – entsprechende Multipunkt-Videokonferenzen durchführen zu können.

Eine Benutzungsanleitung zur Polyspan Viewstation wurde in Form von WWW-Seiten erstellt:

- <http://www.mml.uni-hannover.de/viewstation/>



(a) im Multimedia-Seminarraum des RRZN.

(b) aus der Sicht des Regiebereichs.

Abbildung 17. HLRN-Videokonferenz im Multimedia-Seminarraum des RRZN.
Auf der Projektion sind die Kollegen des ZIB zu sehen.

3.1.2 Datentransport vom Simulationsrechner zum 3D-Streaming-Server

Der Datentransport vom Simulationsrechner zum 3D-Streaming-Server wurde in mehreren Schritten entwickelt und dabei optimiert (siehe auch *Abschn. 3.4.1*):

1. Erzeugung von Teilgeometrien (3D-Visualisierungen der Simulationsgebiete auf den Einzelprozessoren der parallelen Rechnung) als lokale Dateien im DVR-Format, Zusammenfassung mit dem UNIX-Befehl „cat“ und schließlich Aufruf des FTP-Kommandos per Shell-Script.
2. Zusammenfassung der Teilgeometrien im Speicher des Parallelrechners (mittels MPI) und Übertragung aus dem Speicher heraus mit dem FTP-Protokoll (mittels libftp). Beide Funktionen wurden zwecks einfacherer Nutzung in einer hierfür am RRZN entwickelten Visualisierungsbibliothek libDVRP gekapselt. Die Public-Domain-Bibliothek libftp musste zur Nutzung für die Cray T3E aufgrund speziellem Byte-Ordering bzw. Datentyp-Layout modifiziert werden.
3. Ergänzung der Dateierzeugung bzw. des Datentransports aus der libDVRP heraus durch eine „RTSP“-Option (RTSP: Real-Time Streaming Protocol). Dies bedeutet, dass die Serien von Grafikdateien nun nicht mehr einzeln mit dem FTP-Protokoll zum 3D-Streamingserver übertragen werden, sondern in einem Streamingmodus als kontinuierlicher TCP/IP-Datenfluss über das am RRZN für den 3D-Streaming-Viewer entwickelte DVRP-Protokoll transportiert werden. Zur Unterstützung dieses alternativen Transportprotokolls wurde eine Bibliothek librtsp entwickelt und in die libDVRP eingebettet. Außerdem erforderte diese Maßnahme die Unterstützung eines RECORDING-Modus in der 3D-Streamingserver-Software.

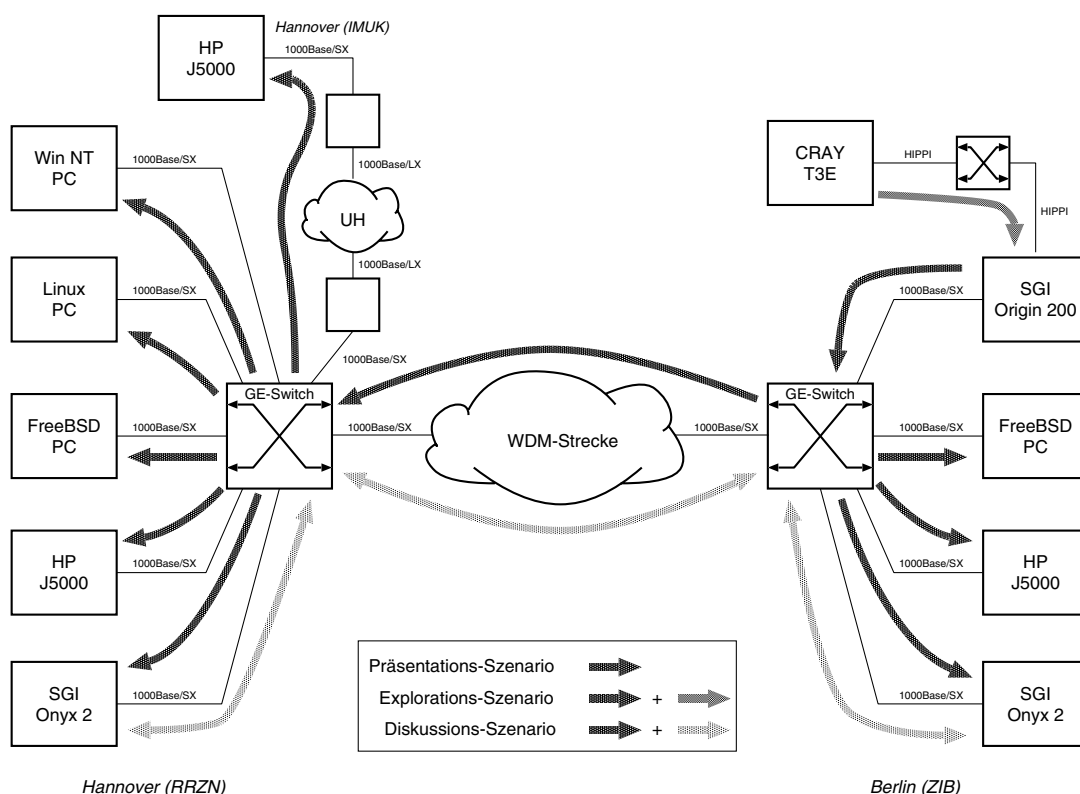
Ergänzend sei auch auf die Ausführungen in *Abschn. 3.2.4* und *Abschn. 3.4.1* verwiesen.

Die komplette Verwendung eigener Quelltexte in C ermöglicht nun auch das Setzen von TCP/IP-Parametern, wie z. B. der Nodelay-Option (um die Latenz zu reduzieren, was insbesondere beim RTSP-Kanal aufgrund der häufigen Request-Response-Beziehungen wichtig ist) oder der Socketbuffergröße (was bei großen Produkten aus Round-trip-delay und Datenraten zunehmend wichtig ist, um das Transportmedium überhaupt ausnutzen zu können).

3.1.3 Aufbau und Tuning des Gigabit-Netzes in Hannover

Aufbau

Als leistungsfähige Infrastruktur-Plattform für das DocShow-VR-Streaming dient ein Gigabit-Ethernet-basiertes LAN. Die ursprünglich geplante Konfiguration zeigt *Abbildung 18*.



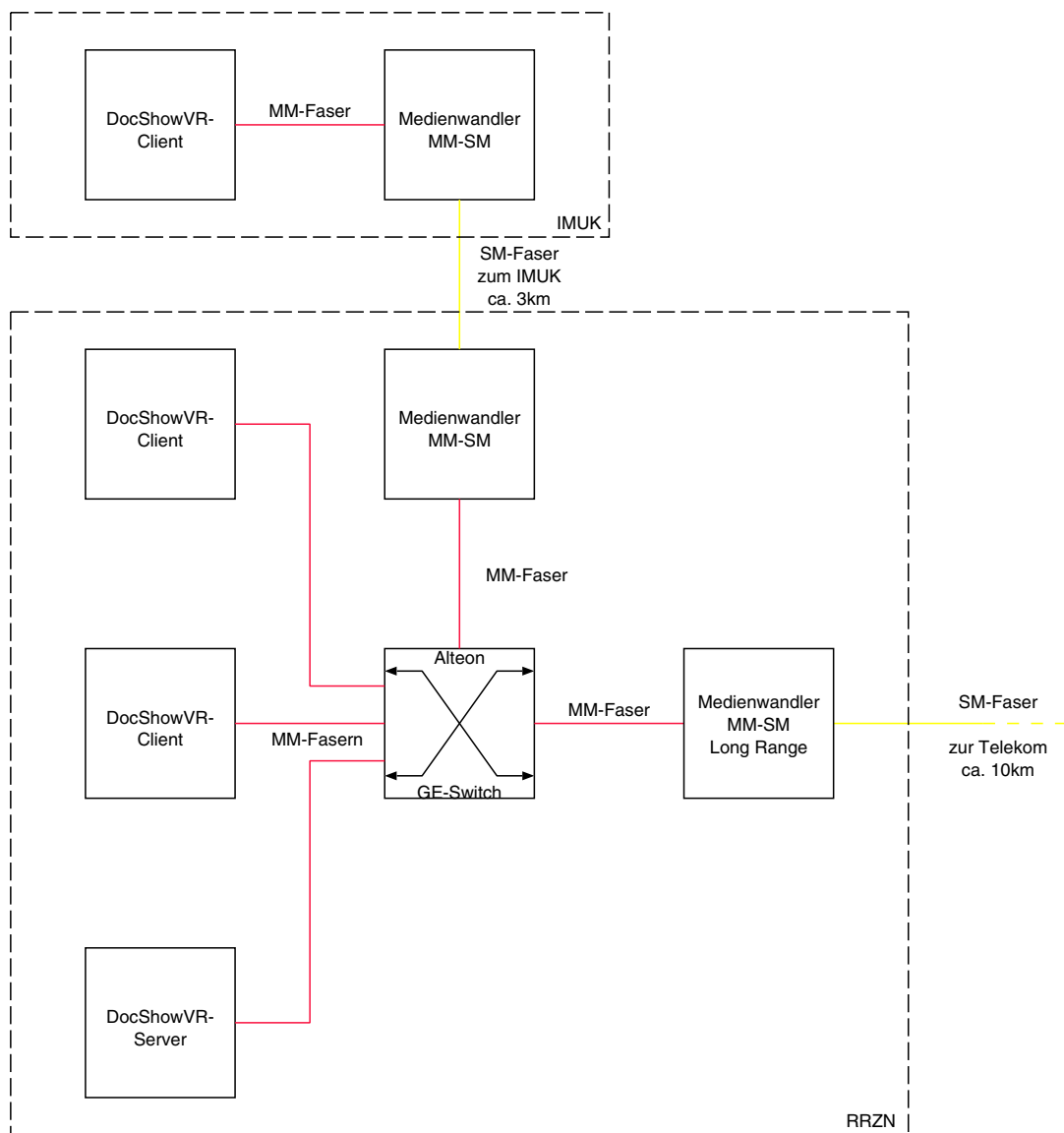
Das Netz wurde zunächst aus zwei identischen Gigabit-Ethernet-Switches vom Typ Alteon ACE-Switch 180 gebildet, von denen je einer am RRZN und am ZIB installiert wurden. An diese Switches wurden die 3D-Grafikrechner und die Streaming-Server angeschlossen. Die Switches werden miteinander über einen transparenten 2,4-Gbit/s-WDM-Kanal von der Deutschen Telekom über Gigabit-Ethernet gekoppelt (siehe auch *Abschn. 3.1.5*).

Im Gigabit-Ethernet-Netz des RRZN in Hannover wurden eine Reihe von unterschiedlichen Rechnern mit Gigabit-Ethernet-Interfaces ausgestattet und an den Alteon-Switch angeschlossen.

Als die wichtigsten sind hier zu nennen:

- SGI Onyx2 mit 4 Prozessoren und 2 GB Hauptspeicher
- SGI Origin 200 mit 2 Prozessoren und 2 GB Hauptspeicher
- HP J5000 bzw. J6000 mit 2 Prozessoren und 4 GB Hauptspeicher
- Sun E450 mit 2 Prozessoren und 2 GB Hauptspeicher

Dazu wurden Gigabit-Ethernet-Karten mit optischer Schnittstelle 1000Base/SX von den Herstellern SGI, Alteon, HP, Sun und Syskonnect verwendet. Ein Rechner im ca. 3 km entfernten Institut für Meteorologie und Klimatologie wurde über eine dedizierte Single-Mode-Faser ebenfalls angeschlossen. Die Umsetzung von den bei 1000Base/SX gebräuchlichen Multi-Mode-Fasern auf Single-Mode-Fasern erfolgt mit Hilfe zweier Medien-Konverter der Firma Transition Networks (siehe *Abbildung 19*). Mit Hilfe von Single-Mode-Fasern können je nach verwendeter Wellenlänge Entfernungen von vielen Kilometern überbrückt werden, während mit Multi-Mode-Fasern bei 850 nm Wellenlänge die Entfernung auf 550 m begrenzt ist.



Jumbo-Frames

Die ersten Tests der Gigabit-Ethernet-Links zeigten enttäuschende Ergebnisse: Datenraten oberhalb von 400 Mbps waren praktisch nicht zu erreichen. Die Ursache hierfür liegt nicht in einer zu geringen I/O-Leistung, sondern in einer zu geringen IP-Paket-Leistung aller heutigen Rechner. Da nach dem Gigabit-Ethernet-Standard die maximale Rahmen-Größe weiterhin – wie auch bei Standard-Ethernet und Fast-Ethernet – 1.518 Byte beträgt, müssen 10-mal so viele Rahmen pro Sekunde übertragen werden als bei Fast-Ethernet, um die erhöhte Datenrate ausnutzen zu können.

Jeder ankommende Ethernet-Rahmen erzeugt einen *packet received interrupt*, der das Betriebssystem zur Verarbeitung veranlasst. Jede Unterbrechung beansprucht einige Prozessortakte. Würde eine Gigabit-Ethernet-Verbindung unter voller Last gefahren, so müsste der betroffene Rechner mehr als 80.000 Interrupts pro Sekunde¹ verarbeiten (siehe auch *Abbildung 20*). Das ist erheblich mehr als heute übliche Rechner verarbeiten können.

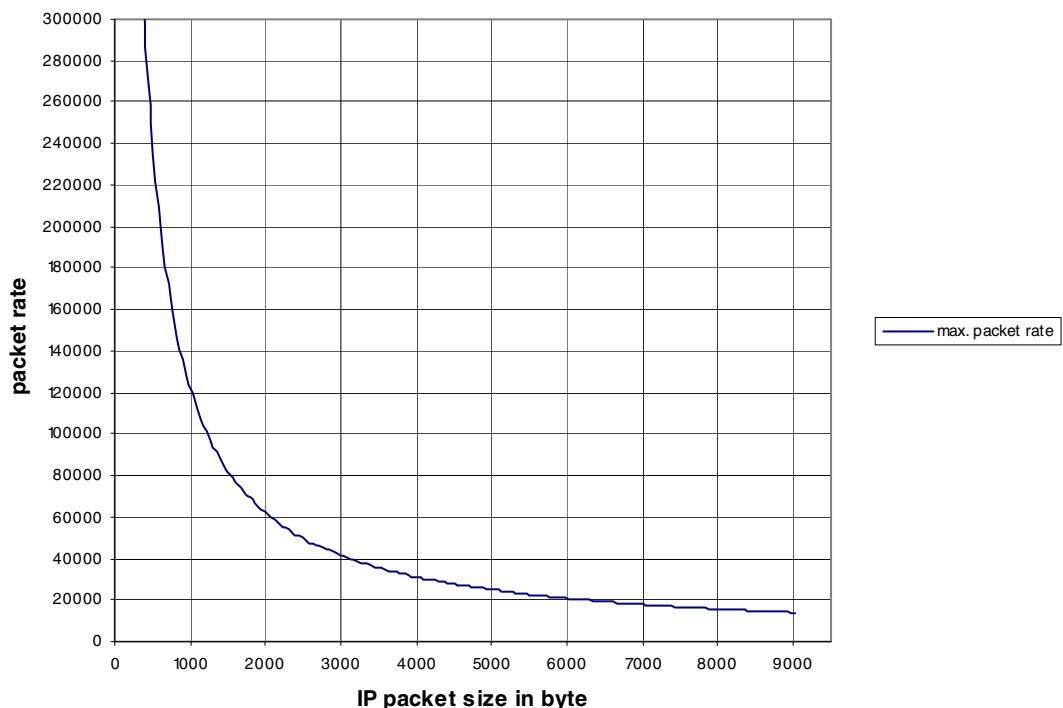


Abbildung 20.

in Abhängigkeit von der IP-Paket-Größe.

$$\begin{aligned} \text{Paketrate} &= 1.000.000.000 \text{ bit pro s} / ((\text{IP-Paketlänge} + \text{Ethernet-Header} + \text{FCS} + \text{Präambel} + \text{Interframe Gap}) * 8) \\ &= 1.000.000.000 \text{ bit pro s} / ((\text{IP-Paketlänge} + 14 \text{ Byte} + 4 \text{ Byte} + 8 \text{ Byte} + 12 \text{ Byte}) * 8) \\ &= 1.000.000.000 \text{ bit pro s} / (\text{IP-Paketlänge in Bit} + 304 \text{ Bit}) \end{aligned}$$

Ein Ethernet-Rahmen setzt sich aus einem Ethernet-Header, der Nutzlast und vier Fehlerkorrektur-bytes zusammen. Viele Operationen des Protokoll-Stacks werden auf den Header angewandt. Die Analyse und der Aufbau des Paket-Headers kostet, unabhängig von dessen Größe, für jedes Paket gleich viel Zeit. Deshalb erzeugen kleinere Rahmen eine relativ höhere Belastung als wenige große. Im Detail ist die zur Auslastung von Gigabit-Ethernet notwendige Paketrate in Abhängigkeit von der IP-Paket-Größe bereits oben dargestellt.

1. Max. Durchsatz / Rahmengröße = max. Anzahl von Rahmen pro Sekunde;
 $1.000.000.000 \text{ bit/s} / 1500 \text{ byte/Rahmen} / 8 \text{ bit/byte} = 83.333 \text{ Rahmen pro Sekunde}$

Ein weiterer Punkt der geringen Effizienz der Nutzung der Gigabit-Ethernet-Leistung besteht darin, dass jedes abgehende oder ankommende Paket eine Speicheroperation verursacht. *Memory-Pages* sind meist in Blöcken von 4 Kbyte, 8 Kbyte oder 16 Kbyte organisiert. Damit kann ein Standard-Ethernet-Rahmen mit seinen 1.500 byte Nutzdaten die Speicheroperationen nicht effizient ausnutzen. So erzeugt z. B. ein Transfer von 8.000 byte sechs Speicheroperationen, obwohl theoretisch bei 4 Kbyte großen Blöcken nur zwei notwendig wären. Für die Nutzlast wäre ein Vielfaches dieser Seitengröße günstig!

Der beliebigen Vergrößerung der Nutzlast sind jedoch enge Grenzen gesetzt. Die von Ethernet eingesetzte Fehlerkorrektur FCS (*Frame Check Sequence*) nutzt das Fehlerkorrekturverfahren CRC32 (Cyclic Redundance Check). Dieses ist nur effizient bis zu einer Nutzlast von 11.450 byte.

Der Hersteller Alteon hat eine Abweichung vom Gigabit-Ethernet-Standard propagiert, bei der eine Erhöhung der Rahmen-Größe auf 9.018 Bytes die Belastung der Rechner verringert. Alteon hat seine Produkte mit diesem Feature, genannt *Jumbo-Frames*, ausgestattet und viele Hersteller sind diesem Beispiel inzwischen gefolgt. Ein positiver Nebeneffekt der Jumbo-Frames ist, dass der Protokoll-Overhead geringer ist und sich dadurch die Nutzdatenrate erhöht (siehe *Abbildung 21*).

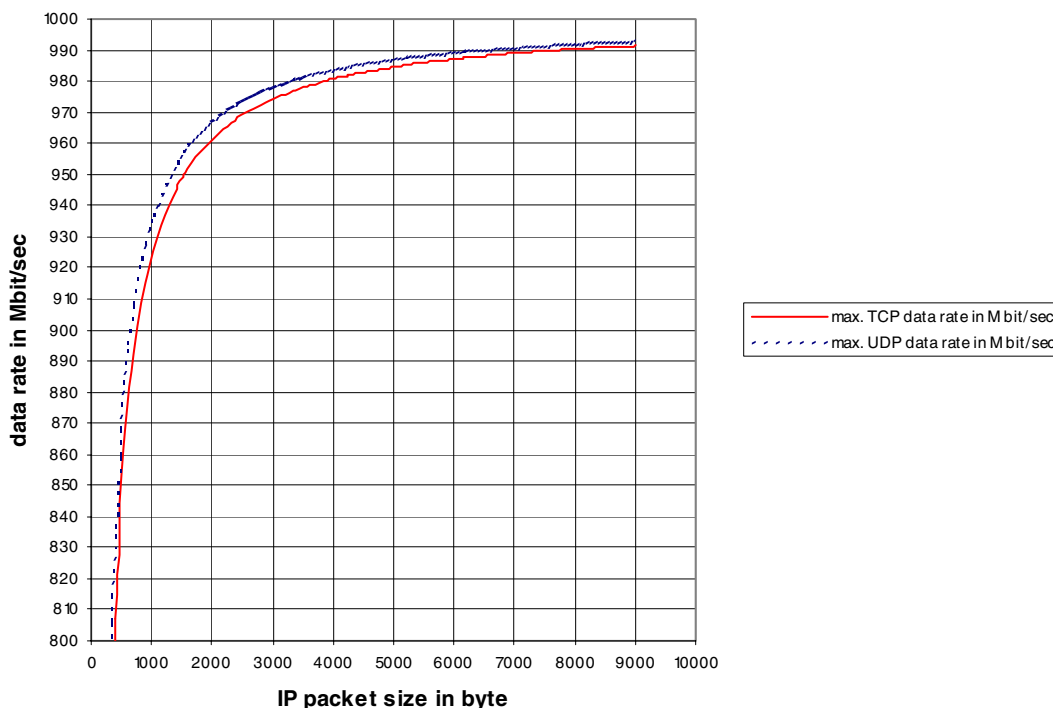


Abbildung 21. Theoretische maximal mögliche Datenraten für TCP und UDP über Gigabit-Ethernet in Abhängigkeit von der IP-Paket-Größe.

$$\begin{aligned} \text{TCP-Datenrate} &= \\ &= (\text{IP-Paketgröße} - \text{TCP-Header} - \text{IP-Header}) * \text{Paketrate} = \\ &= (\text{IP-Paketgröße} - 40) \text{ Byte} * \text{Paketrate in } 1/\text{s} \end{aligned}$$

$$\begin{aligned} \text{UDP-Datenrate} &= \\ &= (\text{IP-Paketgröße} - \text{UDP-Header} - \text{IP-Header}) * \text{Paketrate} = \\ &= (\text{IP-Paketgröße} - 28) \text{ Byte} * \text{Paketrate in } 1/\text{s} \end{aligned}$$

Jumbo-Frames sind allerdings nicht in jedem Szenario sinnvoll und erfordern besondere Anpassungen des TCP-Stacks, um wirklich Verbesserungen zu bringen. So profitieren Anwendungen, die viele kleine Nachrichten austauschen, nicht von Jumbo-Frames. Nur für *Bulk-Data-Transfers*, bei denen eine große Datenmenge am Stück übertragen werden muss, sind Jumbo-Frames sinnvoll. Die hier betrachtete Streaming-Anwendung ist in diesem Sinne ein idealer Kandidat für Jumbo-Frames, da pro Stream mehrere Gigabyte an Daten anfallen, die mit etwa konstanter Datenrate übertragen werden.

TCP-Fenstergröße

Die TCP-Fenstergröße bestimmt die Anzahl der Bytes, die ein Sender abschicken darf, bevor eine Bestätigung vom Empfänger erwartet wird. Durch sie erfolgt die Flusssteuerung. Wird sie zu klein gewählt, so legt der Sender Pausen ein, wenn entsprechend viele Daten gesendet wurden, ohne eine Bestätigung erhalten zu haben. Die benötigte Größe lässt sich aus Datenrate und Verzögerung ermitteln, wobei die Datenrate beim Gigabit-Ethernet-Netz mit 1 Gbit/s fest vorgegeben ist. Die Verzögerung setzt sich aus der Laufzeit der Daten über das Netz und der Verarbeitungszeit bei Sender und Empfänger zusammen.

Die Standard-Fenstergröße ist systemseitig oft mit 16 Kbyte festgelegt. Für Verbindungen wie z. B. der zwischen Hannover und Berlin mit einer *Roundtrip*-Zeit von ca. 4,5 ms zeigt die Rechnung¹, dass die Standard-Fenstergröße bei Standard-Ethernet noch gut ausreicht, für Gigabit-Ethernet allerdings viel zu klein ist.

Unter dem Betriebssystem IRIX lässt sich die eingestellte maximale Fenstergröße für das Gigabit-Ethernet-Interface durch

```
ifconfig -v eg0
```

anzeigen. Anpassungen werden durch

```
ifconfig eg0 rspace <n> sspace <n>
```

vorgenommen.

Messungen

Die Messungen im lokalen Testnetz in Hannover wurden mit der Messsoftware Netperf (Version 2.1p13) durchgeführt. Dabei wird eine TCP-Verbindung zwischen zwei Rechnern 10 Sekunden lang belastet und die Datenrate ermittelt. Die Übertragung erfolgt von Speicher zu Speicher, so dass keine Verfälschung durch Festplattenzugriffe auftritt. Die Messungen erfolgten pro Rechnerpaar einmal mit Standard-Gigabit-Ethernet und einmal mit aktivierten Jumbo-Frames.

Für eine Messreihe wurde die Größe der Socketpuffer konstant gehalten und die Sende- und Empfangspuffergröße in 512-byte-Schritten von 512 byte auf das 1,5-Fache der Socketpuffergröße erhöht. Für jeden Schritt wurden drei Messungen durchgeführt und gemittelt. Fünf Messreihen mit Socketpuffergrößen von 64, 128, 256, 512 und 1024 KByte bilden ein Diagramm.

Jeder Einzelwert wurde folgendermaßen ermittelt:

Sender:

```
netperf -H <zielrechner> -l 10 -s Socketbuffer,Socketbuffer \
-S Socketbuffer,Socketbuffer \
-m Sendeblockgröße -M Empfängerblockgröße
```

Empfänger:

```
netserver
```

Ergebnisse

Die Ergebnisse zeigten, dass alle Messungen zwar reproduzierbar sind, jedoch starke Unterschiede zwischen den Rechnerkombinationen bestanden. Abhängigkeiten bestanden sowohl von Hardwarekomponenten wie Gigabit-Ethernet-Karte, Hauptspeicher, Anzahl und Leistung der Prozessoren, als auch von der Art und Version des Betriebssystems. Bei sehr kleinen Sende- und Empfangspuffern (< 64 Kbyte) geht die Datenrate stark zurück, da die CPU mehr gefordert wird, während bei hohen Puffergrößen eine Konvergenz gegen einen Grenzwert einsetzt, der das Maximum einer Messreihe darstellt. Dieser Wert hängt von der jeweils eingestellten Größe des Socketpuffers ab. Bei einer Socketpuffergröße von 256 Kbyte wird ein Optimum dieses Grenzwerts erreicht, während bei 64 Kbyte in allen Fällen die geringsten Datenraten beobachtet wurden.

-
1. TCP-WindowSize = verfügbare Datenrate * Roundtripverzögerung
Standard Ethernet: WS = 10 Mbit/s * 0,0045 s = 5,49 Kbyte
Gigabit Ethernet: WS = 1000 Mbit/s * 0,0045 s = 549 Kbyte

Exemplarisch werden auf *Abbildung 22* bis *Abbildung 25* die Ergebnisse verschiedener Messungen dargestellt, bei denen jeweils Daten von einer Servermaschine (SGI Origin200) zu einer Workstation (SGI Onyx2 bzw. HP J6000) übertragen wurden.

Abbildung 22 zeigt die Messergebnisse für eine Übertragung von einem Server SGI Origin200 zu einer Workstation SGI Onyx2 (Ausstattung siehe Seite 33) mit Standard-Gigabit-Ethernet, d. h. 1.500 Byte MTU. *Abbildung 23* zeigt die gleichen Messreihen mit Jumbo-Frames. *Abbildung 24* und *Abbildung 25* zeigen die entsprechenden Messergebnisse für eine Übertragung zwischen der SGI Origin200 und einer Workstation HP J6000.

Die gezeigten Messungen erfolgten alle auf Maschinen, die direkt an den Switch angeschlossen waren, d. h. mit wenigen Metern Kabellänge. Bei Messungen über die 3 km lange Strecke zum IMUK wurden keine messbaren Abweichungen der Ergebnisse festgestellt.

Mit Jumbo-Frames erhöhen sich die erzielbaren Datenraten beinahe um den Faktor zwei, und man kommt dicht an den theoretischen Grenzwert von 991 Mbit/s¹. Allerdings zeigen viele Kurven auch starke, reproduzierbare Einbrüche an einzelnen Messpunkten, bei denen der Abfall der Datenrate leicht 200 Mbit/s betragen kann. Im Verlauf des Projekts wurde die Streaming-Software instrumentiert, und es wurden Konfigurationsmöglichkeiten implementiert (z. B. einstellbare Socketbuffer-Size auf Client- und auf Server-Seite), um sicherzustellen, dass das Client-Server-System an einem optimierten Arbeitspunkt betrieben wird.

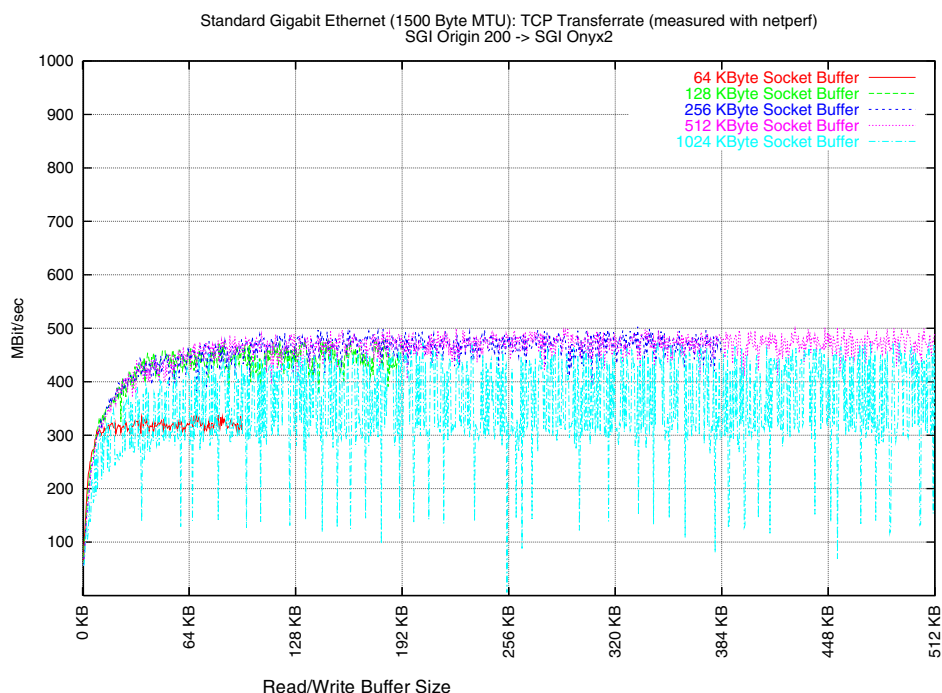


Abbildung 22. Messungen zwischen SGI Origin200 und SGI Onyx2 mit Standard-Gigabit-Ethernet.

1. Mit Jumbo-Frames von 9.018 bytes ergibt sich eine Rate von 13.861 Ethernet-Rahmen pro Sekunde ($1.000.000.000 \text{ bit/s} / (9018 \text{ byte} * 8 \text{ bit/byte})$). Nach Abzug der Ethernet- und der Übertragungsinformationen (siehe *Abbildung 20*) und der IP- und TCP-Header (siehe *Abbildung 21*) ergibt sich eine effektive Größe der Nutzdaten eines Jumbo-Frames von 8.940 byte. Multipliziert mit der Frame-Rate resultiert daraus eine effektive Datenrate von 123.917.340 byte/s oder 991 Mbit/s.

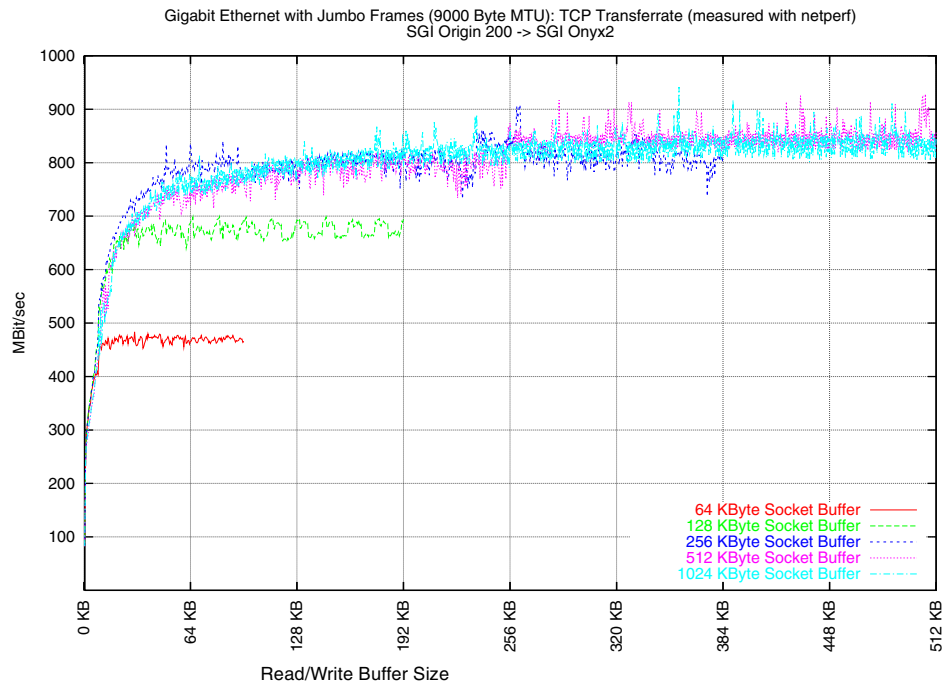


Abbildung 23. Messungen zwischen SGI Origin 200 und SGI Onyx2 mit Jumbo-Frames.

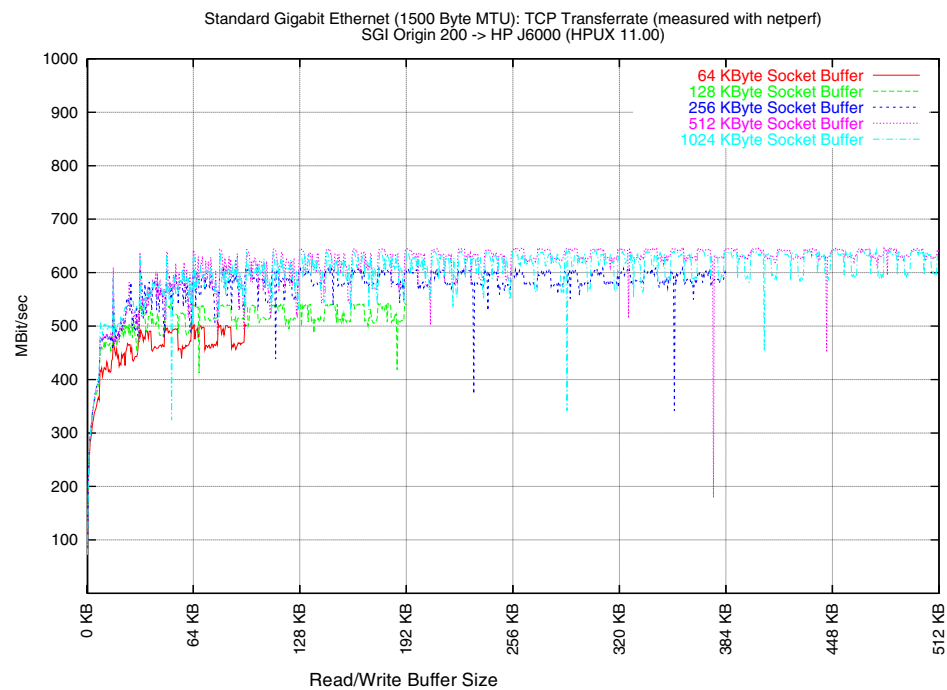


Abbildung 24. Messungen zwischen SGI Origin 200 und HP J6000 mit Standard-Gigabit-Ethernet.

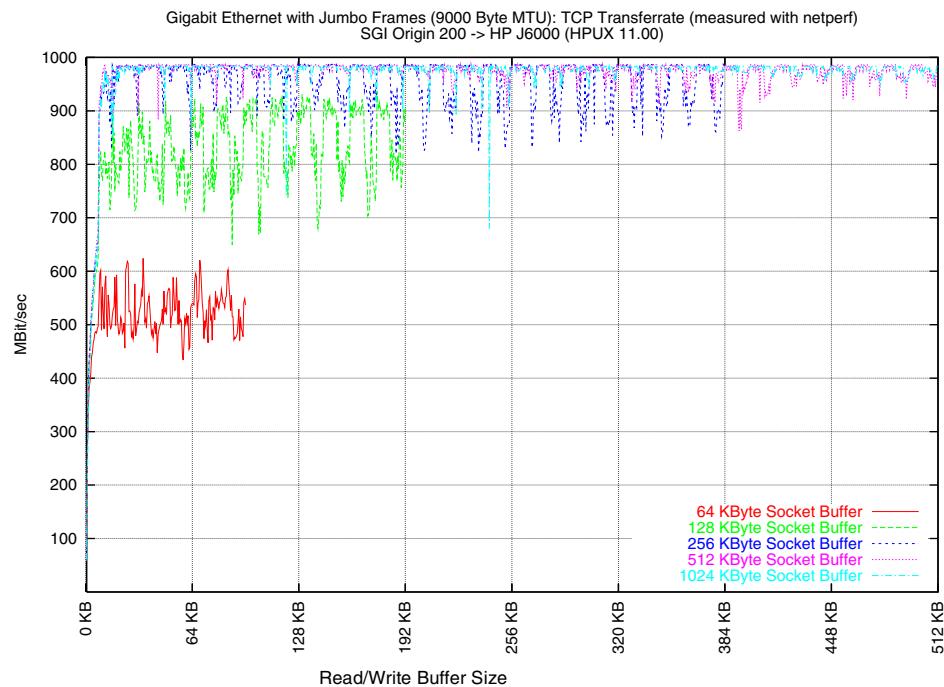


Abbildung 25. Messungen zwischen SGI Origin 200 und HP J6000 mit Jumbo-Frames.

Die Leihmaschinen der Firma HP wurden im Jahr 2001 durch neu beschaffte 3D-Grafikworkstations auf Basis von Windows 2000 ersetzt. Sowohl im RRZN als auch im IMUK handelt es sich dabei um Rechner der Firma Dell (Precision 530, Dual-Pentium 4, 1,7 GHz, stereofähige 3D-Grafikkarte 3Dlabs Wildcat 5110, Gigabit-Ethernet-Interface Sysconnect SK-9843).

Weitere, baugleiche Rechner wurden Anfang 2002 im Learning Lab Lower Saxony (L3S) im ehemaligen Deutschen Pavillon auf dem Expo-2000-Gelände in Betrieb genommen. Das LAN des L3S wurde an das Gigabit-Ethernet-Pilotnetz des RRZN angeschlossen. Im Laufe des Jahres 2002 wurde nach Rücksprache mit dem DFN-Verein auch der HLRN-Rechner [10] (IBM pSeries 690 in Berlin und Hannover, je 384 Prozessoren) in das private Gigabit-Netz integriert. In Hannover wurde der HLRN-Rechner über ein Gigabit-Ethernet-Interface mit dem Alteon-Switch verbunden, an dem auch die Irix- und Windows-basierten Grafik-Workstations und Server angeschlossen sind, über die die 3D-Streaminganwendungen entwickelt und erprobt werden. Weitere Verbindungen wurden zum Institut für Meteorologie und Klimatologie (Hannover, Nienburger Straße) sowie zum L3S geschaltet, um Nutzung und Demonstration auch „vor Ort“ zu ermöglichen.

Anfang 2003 wurde am RRZN eine SGI Onyx3800 (18 CPUs R14000, 600 MHz, 4 InfiniteReality4-Grafikpipes, 34 GB Hauptspeicher, 7 TB Plattenspeicher) in Betrieb genommen und ebenfalls in das Gigabit-Netz integriert.

Generell wurden alle Gigabit-Ethernet-Komponenten mit Jumbo-Frames konfiguriert und betrieben.

3.1.4 Aufbau und Tuning des Gigabit-Netzes in Berlin

Aufbau und Tuning des Gigabit-Netzes im ZIB in Berlin erfolgten analog zu den entsprechenden Schritten im RRZN in Hannover, insofern sind in diesem Abschnitt im Wesentlichen nur die abweichenden Ereignisse beschrieben. Die ursprünglich geplante Konfiguration zeigt *Abbildung 18* (*Abschn. 3.1.3*).

Die interne Gigabit-Ethernet-Infrastruktur am ZIB wurde um den vom RRZN gestellten Gigabit-Ethernet-Switch ebenfalls vom Typ Alteon ACE Switch 180 aufgebaut. Neben den ursprünglich vorgesehenen Systemen SGI Onyx 2 und HP J5000 wurden im ZIB zusätzlich die neu beschaffte SGI

Onyx3000 als Hochleistungs-Graphikrechner, zwei PCs (unter Linux) und schließlich im Laufe des Jahres 2002, ebenso wie in Hannover, auch der HLRN-Rechner [10] (IBM pSeries 690 mit 384 Prozessoren) mit Gigabit-Ethernet-Verbindungen mit dem Alteon-Switch verbunden. Die einzelnen Systeme dienten nicht nur den Anwendungen der Visualisierung, sondern auch ausführlichen Leitungsmessungen und Protokolltests.

Parallel zu den lokalen Messungen in Hannover wurden auch entsprechende Messungen lokal in Berlin durchgeführt, die Ergebnisse ließen sich reproduzieren. Insbesondere beeindruckten die hohen Durchsatzleistungen bei Verwendung von Jumbo-Frames, sofern die jeweils angeschlossenen Rechner genügend leistungsstark waren.

Der 3D-Streaming-Server (SGI Origin200) vom RRZN Hannover wurde nicht, wie ursprünglich geplant, im Jahr 2002 in Berlin installiert, da zunächst Referenzmessungen zwischen den Rechnern Origin200 (RRZN) und Onyx2 (ZIB) und zwischen den Rechnern Origin200 (RRZN) und Onyx3000 (ZIB) durchgeführt werden sollten (vgl. *Abschn. 3.1.5*). Gerade diese Messergebnisse sind wegen der Gigabit-Ethernet-Weitverkehrsstrecke außerordentlich interessant, dabei insbesondere die Abhängigkeiten der Datenraten von den verwendeten Buffergrößen. Jedoch konnten Anfang 2002 die noch Ende 2001 erfolgreich durchgeführten Messungen nicht reproduziert werden, da inzwischen (auch auf Grund von gemeldeten Sicherheitsschwachstellen) auf den Systemen Veränderungen an der Software vorgenommen werden mussten. Längere Zeit gab es keine stabile Gigabit-Netzanbindung zur Onyx3000 im ZIB, so dass sich der Abschluss dieser Messungen verzögerte.

Erst Anfang 2003 konnten diese Probleme im Zuge einer Änderung in der Netzkonfiguration gelöst werden. Da ebenfalls Anfang 2003 auch im RRZN ein Onyx3000-Rechner in Betrieb genommen wurde, bestanden inzwischen Möglichkeiten für Vergleichsmessungen im RRZN selbst ohne einen Rechnertransport nach Berlin durchführen zu müssen.

Während der Projektlaufzeit wurde bei den beiden Projektpartnern RRZN und ZIB ein neues leistungsfähiges verteiltes Hochleistungsrechnersystem vom Typ IBM pSeries 690 installiert und in Betrieb genommen. Bereits unmittelbar nach Installation der ersten Teilsysteme wurden noch im April 2002 die jeweiligen lokalen Gigabit-Ethernet-Netze der IBM-Teilsysteme zwischen Berlin und Hannover über die Gigabit-Ethernet-Verbindung des Projekts miteinander verbunden, und es konnten Messungen durchgeführt werden. Im Einzelnen wird dies im *Abschn. 3.1.5.4* näher ausgeführt. Dieser IBM-Hochleistungsrechner ist für Berlin auch Ersatz für den bislang betriebenen Hochleistungsrechner Cray T3E. Insofern übernahm das IBM-System (teilweise in Berlin, teilweise in Hannover) auch alle Aufgaben des Projekts, die ursprünglich für die Cray T3E vorgesehen waren.

Der Berliner SGI-Rechner Onyx3000 wurde in 2002 mit 16 GB Hauptspeicher ausgestattet und die SGI Onyx2 abgebaut. Weiterhin teilte sich die Berliner Onyx3000 den Alteon-Switch mit dem Berliner Teil des HLRN-Systems, bis Anfang 2003 sämtliche Gigabit-Ethernet-Anschlüsse über die Produktionsnetze (Cisco-Switch) geführt wurden. Auch beim HLRN wurden Jumbo-Frames konfiguriert, um das Gigabit-Ethernet möglichst gut auszulasten.

3.1.5 Aufbau und Erprobung der Verbindung Hannover–Berlin

Im Rahmen des GWiN-Vertrages hatte der DFN-Verein mit der Deutschen Telekom / T-Systems eine Vereinbarung über die Bereitstellung eines transparenten WDM-Kanals¹ mit Übertragungsgeschwindigkeiten zwischen 0,1 und 2,5 Gbit/s zwischen den beiden GWiN-Knoten 1. Ordnung, die im RRZN für weite Teile Niedersachsens und im ZIB für Berlin, Brandenburg und Mecklenburg-Vorpommern betrieben werden, getroffen. RRZN und ZIB beteiligten sich an den Kosten dieser Verbindung.

Technisch stellte die Deutsche Telekom / T-Systems diesen Wellenlängenkanal nur zwischen den jeweiligen Fernmeldeämtern in Hannover und Berlin bereit. Die Streckenlänge beträgt 294 km. Die fehlende Verbindung zwischen den Fernmeldeämtern und den DFN-Knotenräumen wurde über jeweils ca. 10 km lange passive Glasfaser überbrückt. In den Rechenzentren wird die über Singlemode gelieferte Glasfaser mittels eines *Gigabit Singlemode-Multimode Medienkonverters* der Firma *Transition Networks* auf Multimode konvertiert. Endpunkte der insgesamt 314 km langen Lichtwellenleiter-

1. WDM – Wave Division Multiplex

strecke sind in Hannover und Berlin die beiden Alteon-Switche. Sämtliche am Netz befindlichen Rechner in Hannover und Berlin bilden mehrere IP-Subnetze mit privaten Adressen.

Diese Netzwerk-Verbindung, die ursprünglich zum 01.07.2001 bereitgestellt werden sollte, wurde erst am 15.08.2001 – nachdem Deutsche Telekom / T-Systems die Ursache der bis dahin noch bestehenden Probleme in den WDM-Komponenten identifiziert hatte und Konfigurationsfehler behoben hatte – unter erheblicher Mithilfe von Mitarbeitern in RRZN und ZIB erfolgreich in Betrieb genommen.

In den folgenden Abschnitten werden die verschiedenen durchgeführten Messungen erläutert und deren Ergebnisse diskutiert:

- TCP/IP-Messungen,
- Tests mit dem Scheduled Transfer Protocol (STP),
- Verlängerung eines Fibre Channel,
- Einbindung des verteilten HLRN-Systems,
- TCP/IP-Performance-Messungen zum Projektende.

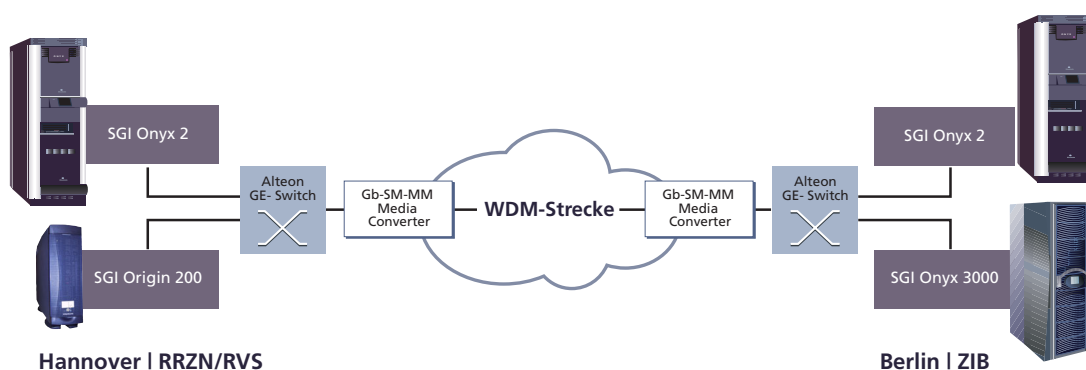


Abbildung 26. Netzkonfiguration zu den im Text erläuterten Messungen.

3.1.5.1 TCP/IP-Messungen

Nachdem sichergestellt war, dass die beteiligten Rechner Jumbo-Frames aktiviert hatten und die maximalen Socketbuffergrößen auf 1 Mbyte vergrößert wurden, wurden mit Hilfe folgender Rechner (siehe auch *Abbildung 26*) umfangreiche Netzmessungen mit variablen TCP-Fenstergrößen und variablen Socketbuffergrößen durchgeführt:

Hannover (RRZN):

- SGI Origin200 mit 2 Prozessoren R12000 (270 MHz) mit 2 GB Hauptspeicher,
- SGI Onyx2 mit 4 Prozessoren R10000 (195 MHz) und 2,3 GB Hauptspeicher.

Berlin (ZIB):

- SGI Onyx2 mit 6 Prozessoren R10000 (195 MHz) und 3 GB Hauptspeicher,
- SGI Onyx3000 mit 16 Prozessoren R14000 (500 MHz) und 12 GB Hauptspeicher.

Die Langzeitmessungen zwischen Berlin (ZIB) und Hannover (RRZN) wurden mit der ursprünglich von Hewlett-Packard entwickelten Messsoftware *netperf* (Version 2.1p3) vorgenommen. Für Stichproben und künstliche Belastung der Gigabit-Ethernet-Strecke wurde *ttcp* eingesetzt. Auf den SGI-Rechnern ließen sich unter dem Betriebssystem IRIX durch den Befehl

```
netstat -I eg0
```

die Rahmengröße des Gigabit-Ethernet-Interfaces anzeigen. Zur Einstellung der Rahmengröße auf Jumbo-Frames müssen Modifikationen im Unix-Kernel vorgenommen werden.

Die Round-Trip-Zeit wurde durch das Senden von zehn *ICMP-Paketen* durch den Befehl

```
ping -c 10 <zielrechner>
```

ermittelt. Dabei wurden Messungen an einem Standort und über die entfernte Leitung bei hoher und bei geringer Last vorgenommen. Die hohe Last wurde durch einen kontinuierlichen Speicher-zu-Speicher-Transfer von Berlin nach Hannover erzeugt.

Folgende Werte wurden dabei ermittelt:

Onyx2 (Berlin) -> Origin200 (Hannover):

- geringe Last: round-trip min/avg/max = 4.798/5.574/9.280 ms
- hohe Last: round-trip min/avg/max = 4.205/27.518/42.588 ms

Onyx3000 -> Onyx2 (lokal, Berlin):

- geringe Last: round-trip min/avg/max = 0.244/0.477/0.706 ms
- hohe Last: round-trip min/avg/max = 0.266/0.491/0.705 ms

Die zusätzliche Netzbelastung führt lokal nicht zu einer größeren Round-Trip-Zeit, jedoch über die entfernte Strecke ergaben sich signifikante Unterschiede.

Mit Hilfe der ermittelten Latenzzeiten ließen sich nun optimale TCP-Fenstergrößen für lokale und entfernte Verbindungen, jeweils bei geringer und bei hoher Last, ermitteln:

lokal: $WS = 1000 \text{ Mbit/s} * 0.0005 \text{ s} = 0.0625 \text{ Mbyte} = 65536 \text{ byte}$
 fern (geringe Last): $WS = 1000 \text{ Mbit/s} * 0.01 \text{ s} = 1.250 \text{ Mbyte} = 1310720 \text{ byte}$
 fern (hohe Last): $WS = 1000 \text{ Mbit/s} * 0.43 \text{ s} = 53.75 \text{ Mbyte} = 56360960 \text{ byte}$

Da jede Verbindung ein eigenes Fenster benötigt und übermäßig große Fenster nicht zu besseren Werten führen, muss ein ausgewogenes Verhältnis zwischen benötigter Fenstergröße und Speicherbedarf gefunden werden.

Die ersten Messungen zwischen SGI Origin200 (RRZN) und der SGI Onyx3000 (ZIB) – beide mit Betriebssystem IRIX – wurden mit *ttcp* durchgeführt:

- Auf der Onyx3000: `ttcp -r -s -l1048576 -b1048576`
- Auf der Origin200: `ttcp -t -s -l1048576 -b1048576 -n1000 onyx3-ge`
- Auf beiden Maschinen: `ifconfig eg0 rspace 1048576 sspace 1048576`

Mit aktivierten Jumbo-Frames wurden dabei Datenraten von ca. 980 Mbit/s erzielt.

Ergebnisse

Unterschiede zwischen Rechnerkonfigurationen machen die Reproduzierbarkeit und Interpretation von Ergebnissen schwierig. Dies betrifft sowohl die Hardwarekomponenten, wie den Typ der Gigabit-Ethernet-Karte, Hauptspeicher, Anzahl und Leistung der Prozessoren als auch Softwarekomponenten, wie Version des Betriebssystems, eingespielte Systemkorrekturen, Treiberversionen und die genaue Systemkonfiguration. Auch die momentane Belastung des Systems, Art der belastenden Services (NFS-Server, diverse Cron-Jobs, speicherintensive bzw. rechenintensive Anwendungen) spielen eine wesentliche Rolle.

Bei kleinen Sende- und Empfangspuffergrößen geht die Datenrate stark zurück, da die CPU mehr gefordert wird. Bei hohen Puffergrößen konvergieren die Messergebnisse gegen einen Grenzwert. Dieser stellt innerhalb einer Messreihe den maximal erreichbaren Messwert dar. Dieser Grenzwert wird allerdings erst bei einer Puffergröße von 1.024 Kbyte erreicht. Größere Werte des Socketbuffers führen zu keinen besseren Ergebnissen.

In *Abbildung 27* und *Abbildung 28* sind exemplarisch die Ergebnisse zweier Messungen dargestellt.

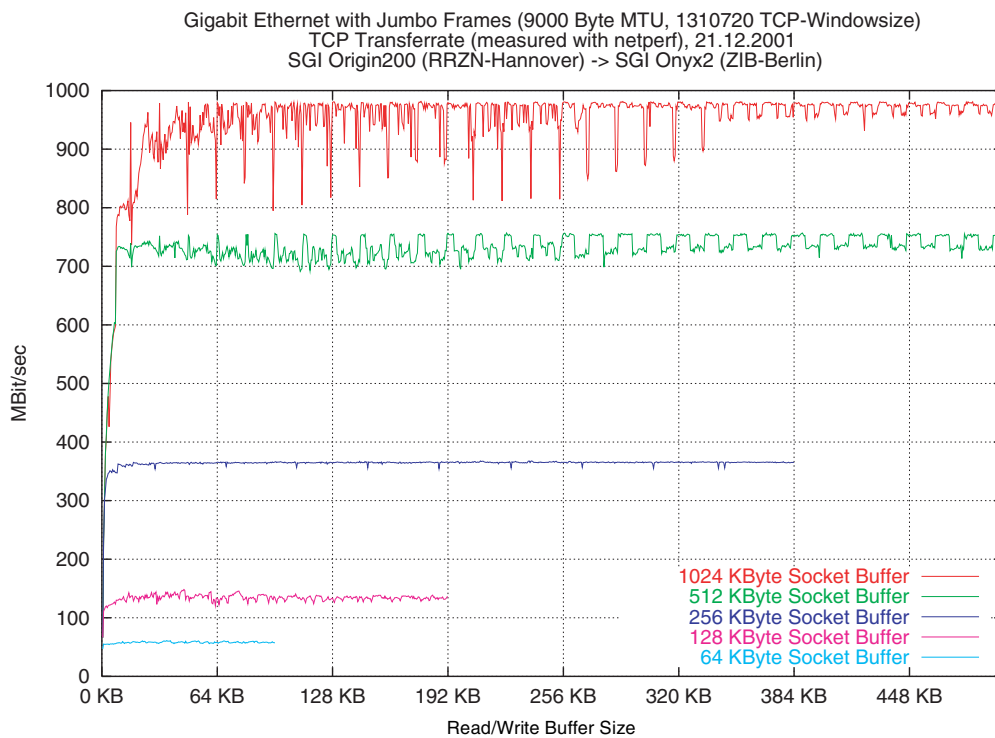


Abbildung 27. Ergebnisse der Messungen (netperf) bei einer Übertragung von SGI O200 (RRZN) zu Onyx2 (ZIB).

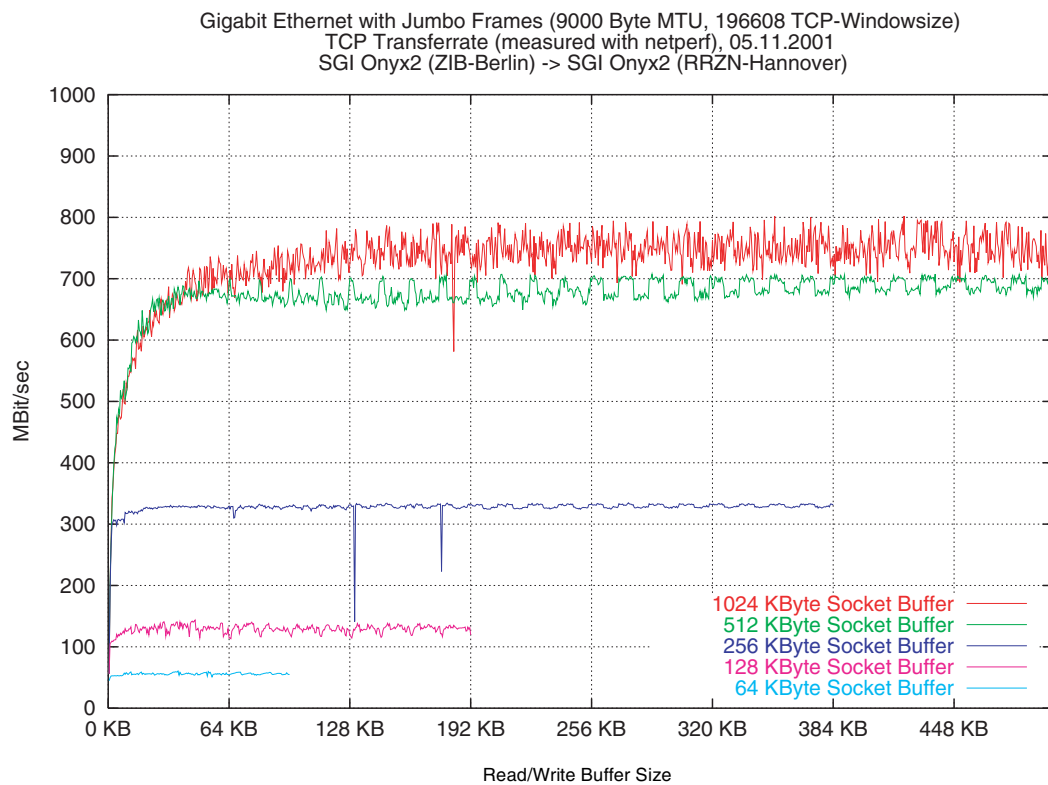


Abbildung 28. Ergebnisse der Messungen bei einer Übertragung von SGI Onyx2 (ZIB) zu SGI Onyx2 (RRZN).

Die Messungen zwischen der SGI Origin200 (RRZN) und der SGI Onyx2 (ZIB) bei einer Fenstergröße von 1.310.720 byte führten zu den besten Ergebnissen. Die Werte brechen jedoch drastisch ein, sobald die entfernte Strecke anderweitig belastet wird. Auch die zur Messzeit geringe Systemaktivität führte zu stärkeren Ausreißern bei den Messwerten. Diese Abfälle der Datenrate können leicht 200 Mbit/s betragen. Die erreichten Spitzenwerte liegen allerdings mit **980 Mbit/s** (122 Mbyte/s) sehr dicht an dem theoretischen Maximum von 991 Mbit/s.

Zwischen der SGI Onyx2 im ZIB und der SGI Onyx2 im RRZN wurden weit schlechtere Werte erzielt. Die Ergebnisse haben sich trotz einer Vergrößerung der Fenstergröße nicht verbessert. Da dasselbe Problem auch bei späteren Messungen mit der Onyx3000 auftrat, handelte es sich dabei offensichtlich um ein Softwareproblem, welches durch eine Systemkorrektur verursacht wurde. Dies zeigt, wie geringe Systemveränderungen zu wesentlich anderen Ergebnissen führen können.

3.1.5.2 Tests mit dem Scheduled Transfer Protocol (STP)

Im Rahmen einer Diplomarbeit wurden am Konrad-Zuse-Zentrum Möglichkeiten untersucht, die Leistung von Gigabit-Ethernet-Netzwerken (GbE) besser auszunutzen. Insbesondere stand hier die Frage nach einer möglichst geringen Belastung der an der Datenübertragung beteiligten Maschinen bei dennoch hoher Übertragungsleistung im Vordergrund.

Bei der Verwendung von Standardprotokollen, wie dem Transmission Control Protocol (TCP), zur Datenübertragung über Hochgeschwindigkeitsnetzwerke, werden die Endsysteme durch die Ausführung verschiedener Protokollfunktionen, wie der Fehlerkorrektur oder der Erkennung und Vermeidung von Stausituationen, signifikant belastet.

Die Ursache hierfür ist zum einen im Ursprung des TCP zu suchen. Es wurde zu einer Zeit entwickelt (1981), als Netzwerkverbindungen im Allgemeinen fehlerträchtig und langsam (im Vergleich zur Datenübertragung innerhalb des Computers) waren. Daher war es notwendig, Mechanismen zur Fehlerkorrektur und zur Vermeidung insbesondere der Überlastung des Empfängers zu etablieren. Der zweite Grund für die starke Belastung der Kommunikationspartner liegt in der Verwendung von GbE als Übertragungstechnik. Bei der Entwicklung von Gigabit-Ethernet wurde aus Gründen der Abwärtskompatibilität zu Fast-Ethernet (100 Mbit/s) und zum ursprünglichen 10-Mbit/s-Ethernet die Größe der Übertragungseinheiten (Maximum Transmission Unit – MTU, bzw. Frame-Größe) unverändert bei 1.500 Bytes belassen. Durch die Erhöhung der Übertragungsgeschwindigkeit auf 1 Gbit/s kann es zu einer Ankunftsrate von ca. 83.000 Frames pro Sekunde kommen, die das Betriebssystem verarbeiten muss. Somit stehen lediglich 12 µs für die Verarbeitung eines Frames zur Verfügung.

Ein Ansatz zur Verringerung der Verarbeitungslast ist die Erhöhung der Frame-Größe (Jumbo-Frames), auf die bereits oben eingegangen wurde. Darüber hinaus bieten moderne Netzwerkadapter die Möglichkeit, Interrupts zusammenzufassen (Interrupt-Coalescing, Interrupt-Avoidance) und so die Belastung des Systems durch die Verarbeitung der Interrupts zu senken.

Die Verwendung des Scheduled Transfer Protocols (STP) ist ein weiterer Ansatz der effizienten Ausnutzung der Leistung von GbE. Hier ist die Idee, TCP durch ein Ressourcen-schonendes Protokoll zu ersetzen, das für Hochgeschwindigkeitsnetzwerke entwickelt wurde. STP entstand ursprünglich im Rahmen einer anderen Netzwerktechnologie, dem Gigabyte System Network (GSN¹).

Hier stand die Entlastung des Empfängers der Daten im Vordergrund. Jede Übertragung wird durch eine kleine Kontrollnachricht vom Sender angekündigt, so dass der Empfänger sie einplanen (*schedulen*) kann. Nach Bereitstellung der Datenpuffer gibt der Empfänger den Transfer durch eine weitere Kontrollnachricht frei. So werden Stau- und Überlastsituationen beim Empfänger vermieden. Um Anwendungen mit hohen Anforderungen an die Latenzzeit ebenfalls zufrieden zu stellen, wurden auch Möglichkeiten vorgesehen, einen Empfangspuffer wieder zu verwenden und so den Austausch von Kontrollnachrichten vor dem Transfer zu vermeiden.

Für den effizienten Einsatz von STP ist es erforderlich, dass die Netzwerkhardware einen Teil der Protokollfunktionen übernimmt. So unterstützen die aktuellen Implementierung für Linux und Irix zwar die Benutzung des Protokolls mit beliebigen Netzwerkkarten, um einen echten Performance-Gewinn

1. <http://www.sgi.com/peripherals/networking/gsn.html>

zu erzielen, sind aber GbE-Adapter mit einem bestimmten Chipsatz erforderlich, der die Programmierung der Karte ermöglicht. Im Weiteren werden Details der Linux-Implementierung erläutert, da eine Realisierung für Irix im Testzeitraum nicht zur Verfügung stand.

Für die Tests wurden zwei PCs mit je einer 3C985B-SX GbE-Karte von 3Com ausgestattet. Die Rechner waren mit einem Pentium-III-Prozessor mit 800 MHz und 256 MB Hauptspeicher ausgestattet. Auf den 3Com-Karten wird der so genannte Tigon2-Chipsatz verwendet, der für die Hardware-Unterstützung von STP unter Linux obligatorisch ist. Für die Tests ist ein modifizierter Linux-Kernel der Version 2.4.4 verwendet worden, der um die STP-Funktionalität erweitert wurde. Die Benutzung von STP unter Linux erfolgt, analog zu TCP über das Socket-Interface. Die Messungen wurden mit `net-perf` durchgeführt, das für die Benutzung von STP nur leicht modifiziert werden musste.

Abbildung 29 zeigt zwei verschiedene Versuchsanordnungen. Auf der linken Seite sind die PCs direkt miteinander verbunden. In diesem Szenario werden die besten Messergebnisse erwartet. Der rechte Teil der Abbildung zeigt die Testumgebung, die für die Untersuchungen über die dedizierte Leitung zwischen ZIB und RRZN umgesetzt wurde.

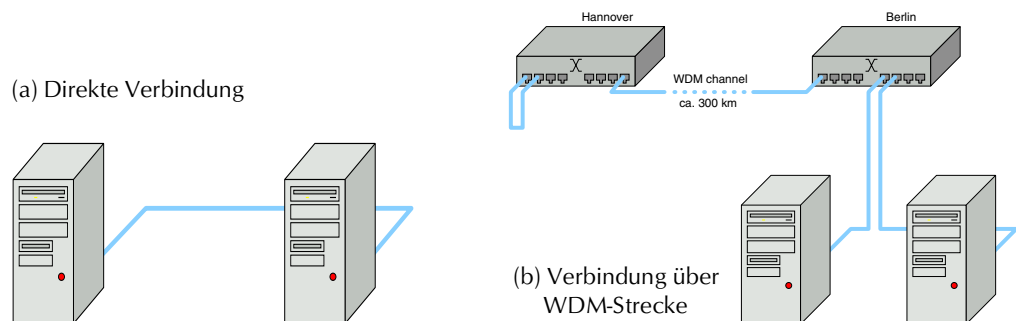


Abbildung 29. Versuchsanordnungen.

Zu beachten ist hier, dass beide PCs dafür in Berlin bleiben konnten. Die Daten wurden durch eine geschickte Konfiguration der GbE-Switches zuerst vom einen PC nach Hannover geleitet und von dort durch eine Direktverbindung zweier Switch-Ports wieder über die Leitung nach Berlin zum zweiten PC geschickt. Dies war durch die Einrichtung zweier getrennter VLANs im Berliner und im Hannoveraner Switch möglich, die lediglich über die direkte Leitung am Switch in Hannover verbunden waren. Tatsächlich wurden weitere Tests durchgeführt, bei denen beide PCs direkt über einen Switch verbunden waren. Es zeigte sich jedoch praktisch kein Unterschied zu den Messungen bei der direkten Verbindung (Back-To-Back) der Rechner.

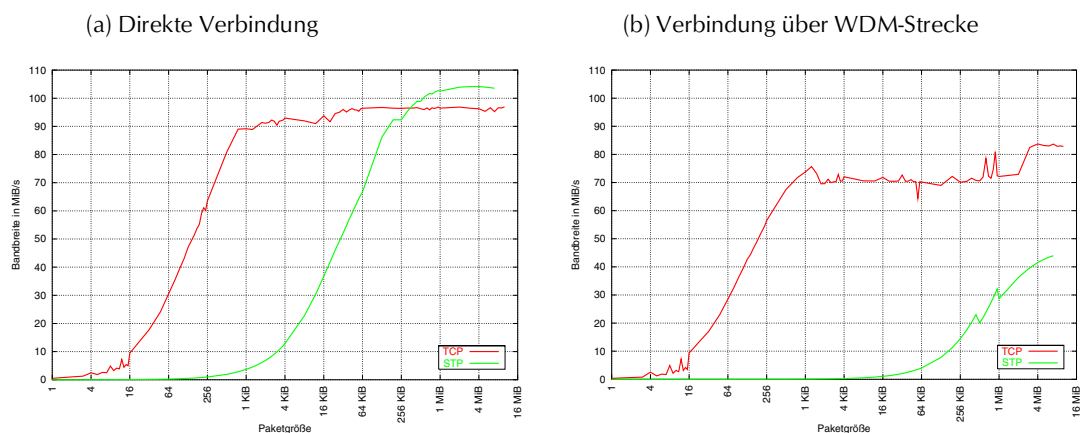


Abbildung 30. Ergebnisse der Messungen der Datenrate.

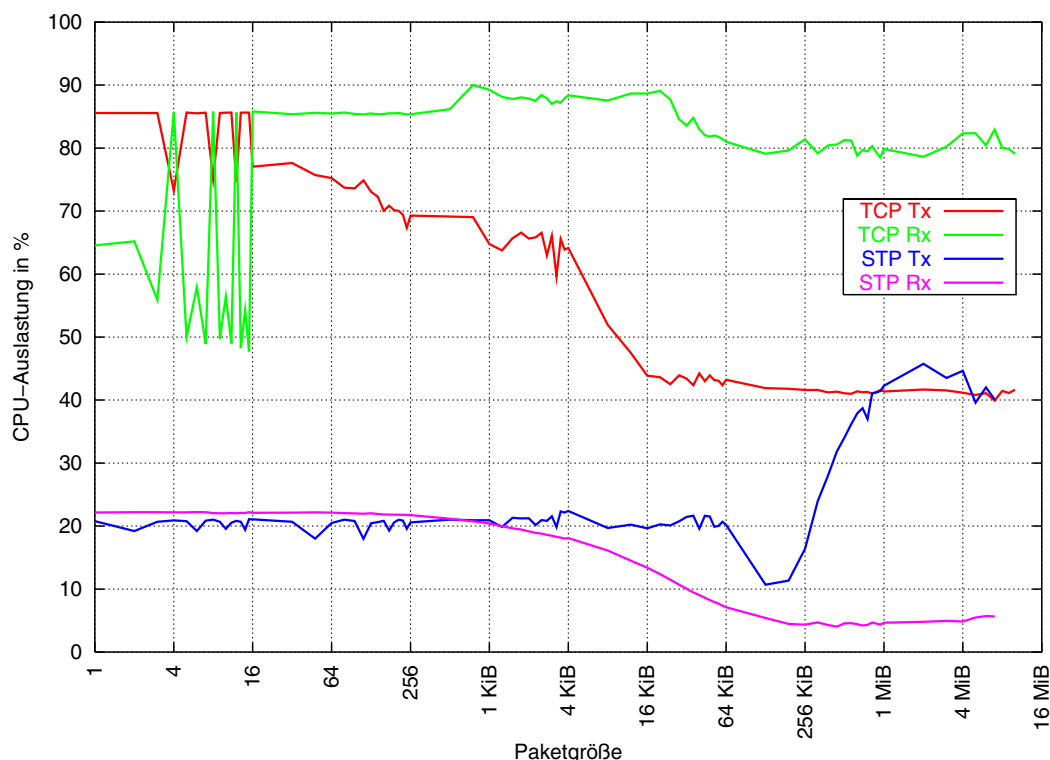


Abbildung 31. Ergebnisse der Messungen der CPU-Lastung.

In den *Abbildungen 30* und *31* sind zwei Messungen der Bandbreite und eine Messung der CPU-Auslastung während der Übertragung dargestellt. Es sind jeweils Messwerte für TCP und STP abgebildet. Der erste Graph zeigt die erreichte Bandbreite bei direkter Verbindung der beiden Testsysteme. Der darauf folgende Graph stellt die Bandbreite bei der Messung über die WDM-Strecke dar und im letzten Graphen wird die CPU-Lastung, jeweils für Sender und Empfänger getrennt, während der ersten (lokalen) Messung gezeigt.

In der lokalen Messung erweist sich eine höhere Leistung von TCP gegenüber STP bis zu einer Paketgröße von ca. 256 kB. Bei größeren Nachrichten kann STP zeigen, dass es besser in der Lage ist, die Leistung von GbE auszunutzen. Es werden Maximalwerte von 95 MB/s bei TCP und 104 MB/s bei STP erreicht. Der Grund für das späte Ansteigen der Kurve von STP liegt in einem Implementierungsdetail. Durch die unflexible Umsetzung müssen auch bei kleinen Paketgrößen zunächst über Kontrollnachrichten Puffer beim Empfänger reserviert werden. Der zusätzliche Zeitaufwand für diese Extrübertragung wird erst ab einer Paketgröße von ca. 256 kB vernachlässigbar. Im zweiten Graphen zeigt sich eine weitere Schwachstelle in der STP-Implementierung für Linux. Über die WDM-Strecke wird maximal die Hälfte der Bandbreite erreicht, die mit TCP gemessen werden konnte. Für die Messung mit TCP wurde hier die Größe des Übertragungsfensters (TCP-Window-Size, siehe oben) erhöht, um die größere Signallaufzeit auszugleichen, die durch die Entfernung von ca. 600 km bedingt ist. STP kennt ein ähnliches Konstrukt wie die TCP-Window-Size, die Blockgröße. Leider honorierte die Linux-Implementierung von STP diese Einstellung nicht und so wurden nur enttäuschende Werte über die WDM-Strecke erreicht. Es zeigt sich jedoch auch bei TCP, dass nicht die volle Leistung erzielt wird. Der Maximalwert der Datenrate liegt hier bei ca. 84 MB/s.

Die CPU-Lastung während der lokalen Kommunikation ist in *Abbildung 31* dargestellt. Es lässt sich erkennen, dass die Benutzung von STP sowohl den Sender als auch den Empfänger signifikant weniger belastet, als die Datenübertragung mit TCP. Besonders beim Empfang von großen Nachrichten geht die Auslastung der CPU bei STP auf bis zu 5 % zurück, während der Empfänger bei TCP zu 80 %

mit der Kommunikation beschäftigt ist. An dieser Stelle kommt die Unterstützung von STP durch die Netzwerkkarten zum Tragen. Der Sender der Daten wird bei STP unterhalb einer Paketgröße von 1 MB ebenfalls weniger beansprucht als bei TCP. Da jedoch keine Funktionalität zum Senden in die Netzwerkkarten integriert wurde steigt die CPU-Belastung bei großen Datenpaketen aufgrund des hohen Datenaufkommens wieder an. Die Senkung der Prozessorauslastung des Senders bei TCP gründet sich auf die Verwendung von Jumbo-Frames, die den Aufwand für die Erzeugung von IP-Paketen ab einer Paketgröße von 4 KB deutlich senken. Jedoch ist auch der Sender bei TCP deutlich stärker belastet als bei STP.

Die Untersuchungen zum Scheduled Transfer Protocol haben gezeigt, dass die Kommunikation mit TCP über Gigabit-Ethernet deutliches Potential für Optimierungen, sowohl im Bereich der Bandbreite, als auch im Bereich der CPU-Belastung durch die Datenübertragung, hat. Die Unzulänglichkeiten der STP-Implementierung für Linux verhinderten leider einen detaillierteren Vergleich der beiden Protokolle, aber es ist deutlich zu erkennen, dass STP die Leistung von GE besser ausnutzen kann.

3.1.5.3 Verlängerung eines Fibre Channel

Für größere Rechnersysteme sind heute für den Platten- und Magnetband-Hintergrundspeicher SAN (Storage Area Network) -Speicher die flexible Lösung. Fibre Channel (FC) Protokolle auf Glasfaserleitungen stellen üblicherweise die notwendigen Datenverbindungen dazu her. Für das verteilte HLRN-System sind jeweils SAN-Systeme vorgesehen. Von großem praktischen Wert ist die Eigenschaft, beide 300 km voneinander entfernten Teil-SANs über die Gigabit-Ethernet-Strecke als **ein** großes SAN zu betreiben. Historisch gesehen lag die Längenbegrenzung eines Fibre Channel zunächst unter 1 km, bei Verwendung von dedizierten Singlemode-Strecken um die 10 km.

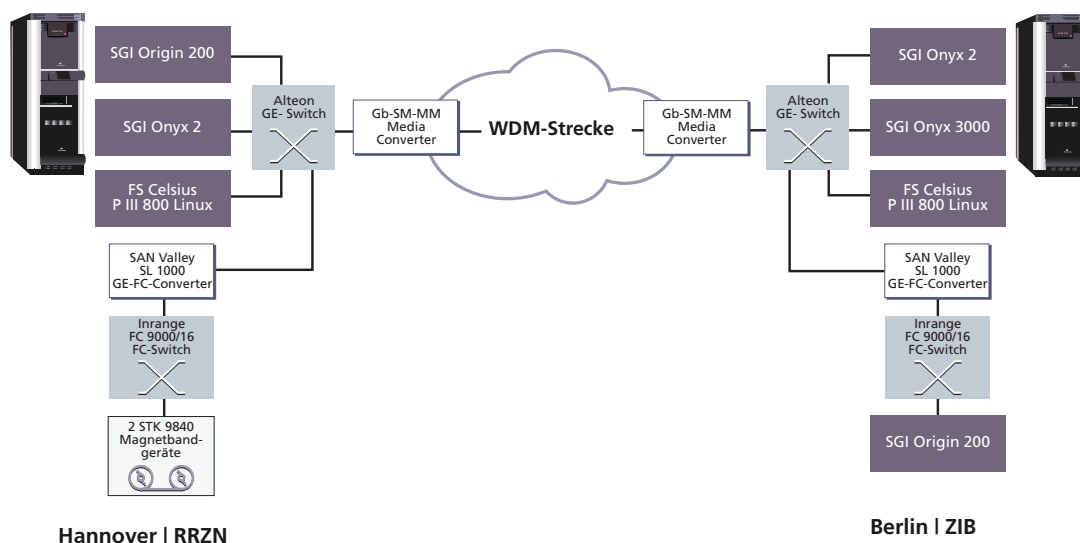


Abbildung 32. Verlängerung eines Fibre Channel, Netzkonfiguration.

Die Firma INRANGE Technologies GmbH erklärte sich bereit, gemeinsam mit dem ZIB und dem RRZN die Machbarkeit der Übertragung eines Fibre Channel über diese große Entfernung zu verifizieren und stellte die notwendigen Geräte (2 Fibre Channel — Gigabit Ethernet Converter vom Typ SANValley SL 1000 sowie 2 Fibre Channel Switches vom Typ Inrange FC 9000/16) sowie das notwendige Know-how für zwei Tage zur Verfügung. Dieser FC-GE-Converter verfügt über einen Durchsatz von mehr als 8 GByte/s bei 4 FC-Verbindungen (bis 100 MByte/s full duplex) und 4 GE-Verbindungen mit gewissen *Quality-of-Service*-Unterstützungen. Der Converter SL 1000 verwendet als Transportprotokoll UDP und nutzt ebenfalls die Eigenschaft der Jumbo-Frames, allerdings mit 3.000 Byte pro Paket.

Da die neuen Rechner in ZIB und RRZN mit ihrem SAN noch nicht installiert waren, haben wir als Testkonfiguration eine FC-Verbindung zwischen einer SGI O200 und zwei Magnetbandstationen vom Typ STK 9840 gewählt. Damit beschränkte sich der Test auf Transfers mit einer Bandbreite in der Größenordnung von 30 MByte/s. Der normale Betrieb innerhalb des ZIB sieht vor, dass diese beiden Magnetbandstationen über eine direkte (kurze) FC-Verbindung bedient werden. In mehreren Schritten wurden die von der Fa. INRANGE bereit gestellten Geräte integriert: Zunächst lokal die beiden FC-Switche, dann ebenfalls lokal Auftrennen der FC-Verbindung zwischen den Switchen durch Einfügen der FC-GE-Converter und damit Transport auf IP-Ebene über Gigabit-Ethernet und schließlich Betrieb der Magnetbandstationen einschließlich eines Converters und eines FC-Switches im RRZN und damit Betrieb der FC-Kopplung über 300 km hinweg. Die folgende Tabelle zeigt die gemessenen Transferraten für das Lesen von beiden Magnetbandgeräten und das Schreiben auf beide Magnetbandgeräte.

	Lese-Datenrate	Schreib-Datenrate
Lokale direkte Anbindung	28,8 MByte/s	26,4 MByte/s
Lokale Anbindung über FC-Switch	27,6 MByte/s	19,6 MByte/s
Lokale Anbindung mit Umsetzung auf IP über Gigabit-Ethernet	27,6 MByte/s	19,6 MByte/s
Entfernte Anbindung auf dedizierter Leitung	27,0 MByte/s	6,0 MByte/s
Entfernte Anbindung mit konkurrierendem Verkehr (97 MByte/s)	26,6 MByte/s	5,6 MByte/s

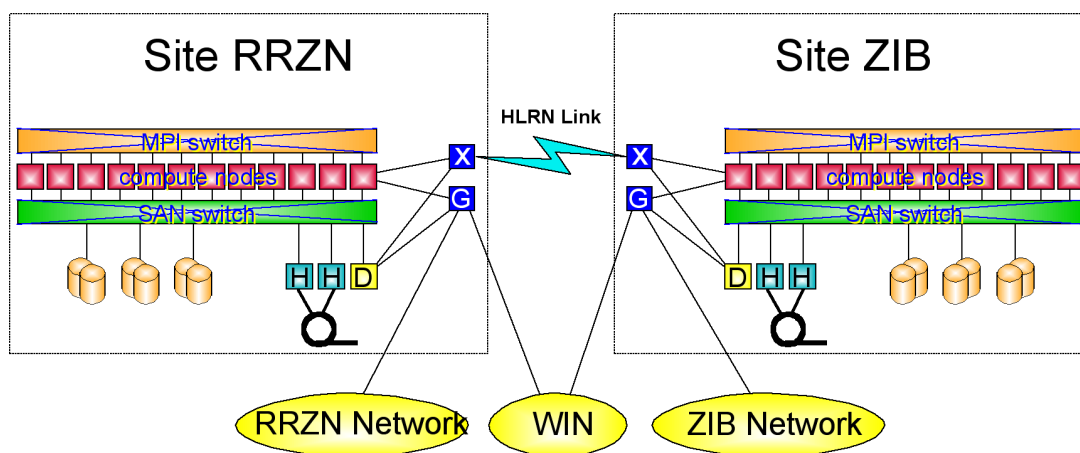
Beim Lesen der Magnetbänder führt die Konvertierung über IP-UDP Pakete auf dem Gigabit-Ethernet lediglich zu einer Reduktion der Bandbreite von 8 %. Erstaunlich ist die gemessene Gesamttransferate beim Test der entfernten Anbindung einschließlich eines konkurrierenden Verkehrs von 97 MByte/s zwischen zwei SGI-Rechnern: Mit **989 Mbit/s** werden die theoretisch möglichen 991 Mbit/s praktisch erreicht! Der drastische Abfall des Durchsatzes beim Beschreiben der Magnetbänder erklärt sich durch einen entstandenen Start-Stop-Betrieb der Magnetbandstationen; hier reichte die knappe Testzeit nicht aus, dies durch Veränderung einiger Parameter zu verbessern. Für die SAN-Kopplung des HLRN-Systems ist diese Eigenschaft nicht relevant.

3.1.5.4 Einbindung des verteilten HLRN-Systems

Während der Projektlaufzeit wurde bei den beiden Projektpartnern RRZN und ZIB ein neues leistungsfähiges verteiltes Hochleistungsrechnersystem installiert und in Betrieb genommen: Dabei handelt es sich um insgesamt 24 Systeme IBM pSeries 690 mit je 32 IBM Power 4 Prozessoren (insgesamt 768 Prozessoren). Jeweils 12 dieser Systeme befinden sich, lokal eng über ein firmenproprietäres Netzwerk „Colony-Switch“ als ein Cluster verbunden, an beiden Standorten.

Beide Systeme zusammen bilden gemeinsam einen „virtuellen Supercomputer“, das „HLRN-System“ (Hochleistungsrechner Nord). Zum virtuellen Supercomputer gehören neben gemeinsamen administrativen Regelungen (einheitliche Benutzerverwaltung, gemeinsamer Wissenschaftlicher Ausschuss für die Vergabe von Projektmitteln, abgestimmte Software-Umgebung, einheitliche Benutzerdokumentation) auch die enge Netzverbindung über die vom Projekt bereitgestellte Hochgeschwindigkeitsverbindung.

Bereits unmittelbar nach Installation der ersten Systeme wurden noch im April 2002 die jeweiligen lokalen Gigabit-Ethernet-Netze zwischen den einzelnen IBM-Systemen mit dem jeweiligen lokalen Alteon-Switch verbunden. Zunächst diente diese Verbindung administrativen Aufgaben und der Vereinfachung der Installation der Software, seit Juni 2002 nutzen auch die Benutzer diese Verbindung und können zum Beispiel auf gemeinsame Plattenbereiche zurückgreifen.



HLRN System - V3.0
HLRN und IBM Confidential

Abbildung 33. Konfiguration des HLRN-Systems.

Netze

Beim HLRN-System sind unterschiedliche Netztechniken installiert:

- lokal zwischen den einzelnen Systemen das firmenproprietäre Netzwerk „Colony-Switch“ (auch SP-II-Switch) sowohl mit proprietären Netzprotokollen als auch mit TCP/IP für lokale Verbindungen nutzbar,
- Gigabit-Ethernet für die Anbindungen an die jeweiligen lokalen Netzwerke von RRZN und ZIB sowie an das GWiN und an den dedizierten Link RRZN-ZIB,
- Fast-Ethernet für die Anbindungen sämtlicher Komponenten (Rechner, Platten-Controller, sonstige Teilkomponenten) und an den dedizierten Link RRZN-ZIB für Zwecke der Administration.

Das Gigabit-Ethernet des HLRN-Systems erstreckt sich von jeweils zwei speziellen Rechnern („Login-Nodes“) über das CISCO-Etherchannel-Protokoll¹ zum GWiN und Link. Damit werden alle nichtlokalen Paket-Ströme sämtlicher Rechner über die Login-Nodes gebündelt, über GbE mittels Trunking zum CISCO-Switch gesendet, dort wiederum gebündelt und schließlich über GbE zum Alteon-Switch des Projektes gesendet. Damit teilen sich dann pro Standort die SGI-Grafikrechner und -Server, weitere angeschlossene Workstations und PCs sowie das HLRN-System die effektive Bandbreite.

Messungen

Ähnlich wie in den vorigen Abschnitten beschrieben wurden Tests der TCP-Datenrate zwischen einzelnen p690-Knoten in Berlin und Hannover vorgenommen. Es handelte sich dabei um Memory-to-Memory-Transfers, welche lediglich die Zeit messen, die nötig ist, um eine bestimmte Menge an Daten zu übertragen. Der Zeitraum der gesamten Messung betrug etwa 24 Stunden. Parameter der durchgeführten Messung sind die verwendete TCP-Socketbuffergröße sowie die Größe der in den read/write-Systemcalls übermittelten Datenpakete.

Ergebnisse

Die maximale Datenrate von 991 Mbit/s wird fast erreicht. Durch die leistungsfähigen CPUs sowie den verwendeten 64-bit-PCI-BUS der p690-Knoten ist die Skalierung des Netzverkehrs sehr gut – d. h. dass es schon bei sehr kleinen Datenpaketen zum Erreichen der maximal möglichen Datenrate kommt.

1. Etherchanneling/Bonding/Trunking: Einbinden verschiedener physikalischer Paket-Ströme im Netzwerk auf ein physikalisches Medium oder Verteilen eines Stromes auf verschiedene physikalischen Medien.

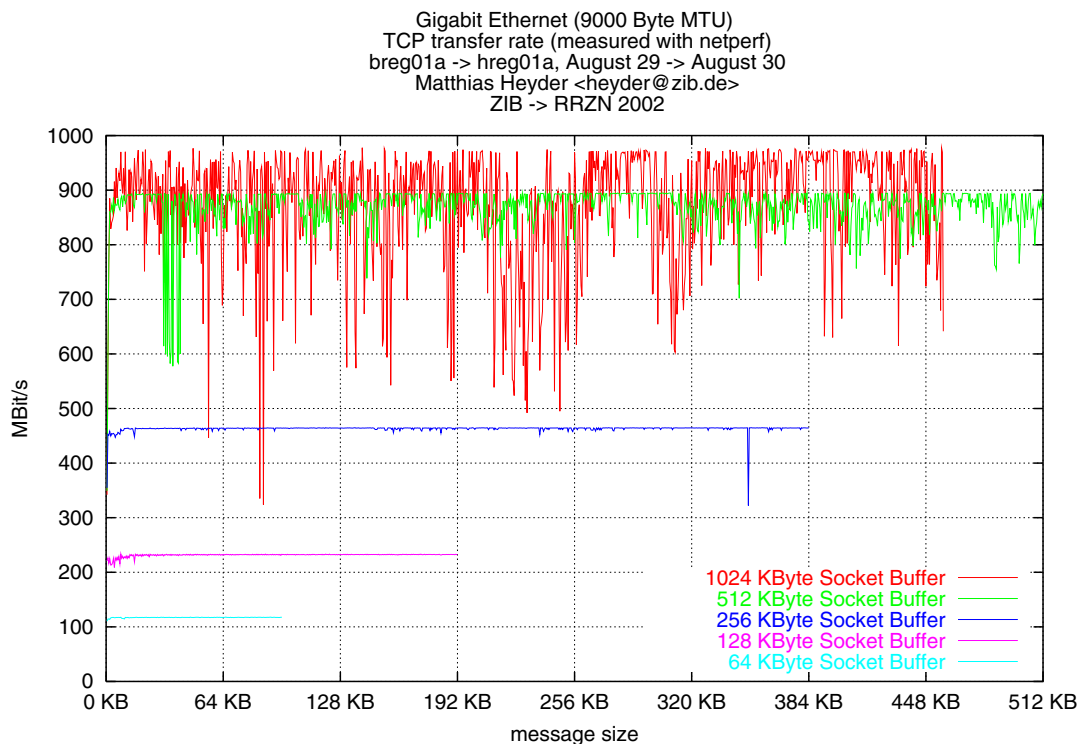


Abbildung 34. Messungen der Datenrate zwischen den HLRN-Komplexen in Berlin und Hannover.

3.1.5.5 TCP-Performance-Messungen zum Projektende

Anfang des Jahres 2003 wurden beide Enden der Gigabit-Weitverkehrsverbindung unmittelbar an die jeweilige Produktionsnetzinfrastruktur in Berlin und Hannover angeschlossen. Dabei wurden die entsprechenden CISCO-Switches auf beiden Seiten auf Jumbo-Frames-Support konfiguriert. Auch die Gigabit-Interfaces der an dem Pilotnetz beteiligten Rechner wurden nach und nach aus der indirekten Betriebsweise über den Alteon-Gigabit-Ethernet-Switch herausgelöst und in die Produktionsnetze überführt, allerdings zunächst weiterhin in privaten IP-Adressbereichen mit zwischen HLRN und anderen Workstations bzw. PCs aufgeteilten VLANs. Außerdem wurde auch im RRZN eine SGI Onyx3800 in Betrieb genommen und in die Messserien aufgenommen.

Die im Folgenden abgebildeten Ergebnisse beruhen auf Messungen, die während des laufenden HLRN-Produktionsbetriebs durchgeführt wurden, teilweise bei eingeschalteter Dateisystem-Spiegelungsfunktion.

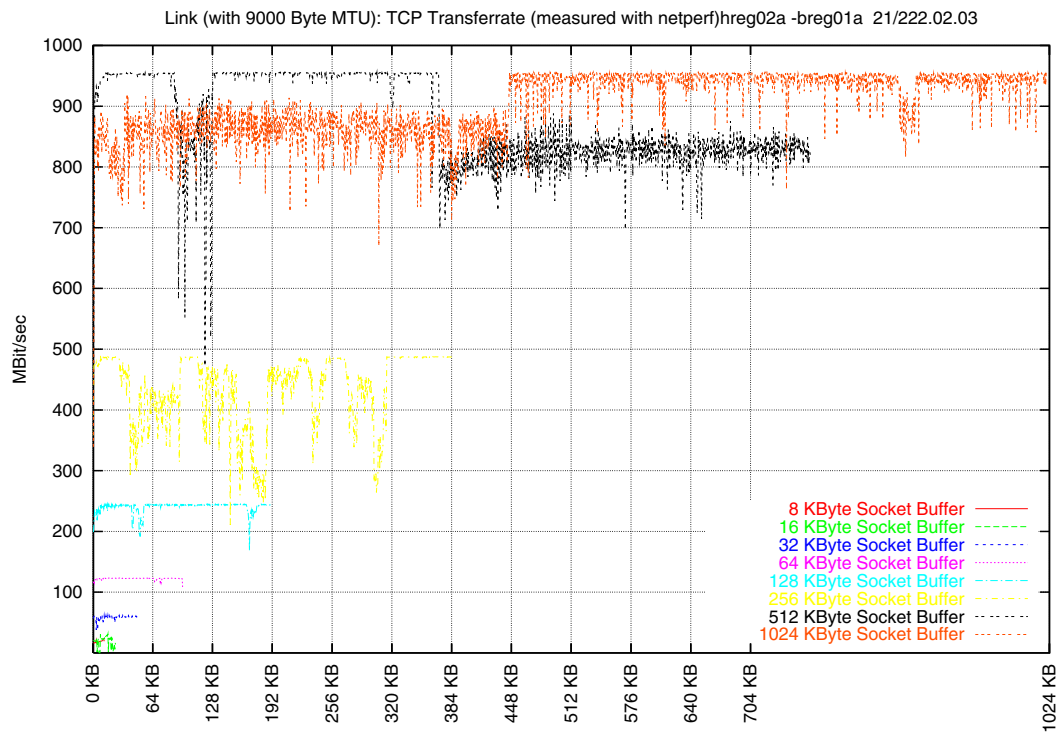


Abbildung 35. Messungen der Datenrate zwischen den HLRN-Komplexen in Berlin und Hannover.

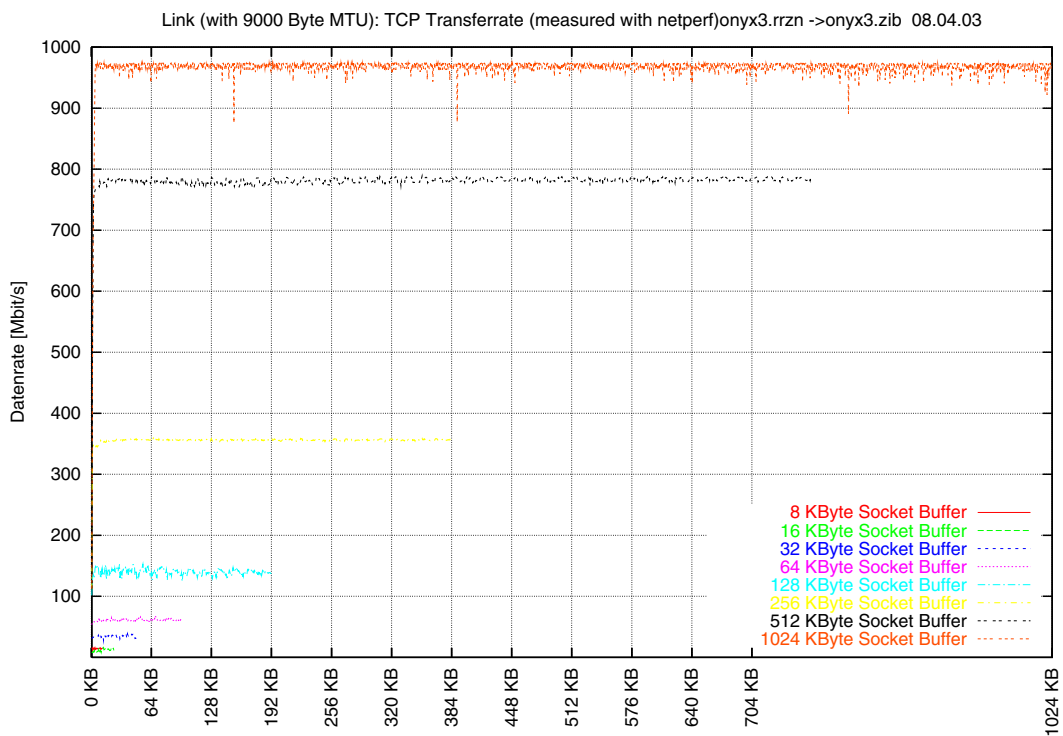


Abbildung 36. Messungen der Datenrate zwischen SGI Onyx3800 in Hannover und SGI Onyx3800 in Berlin.

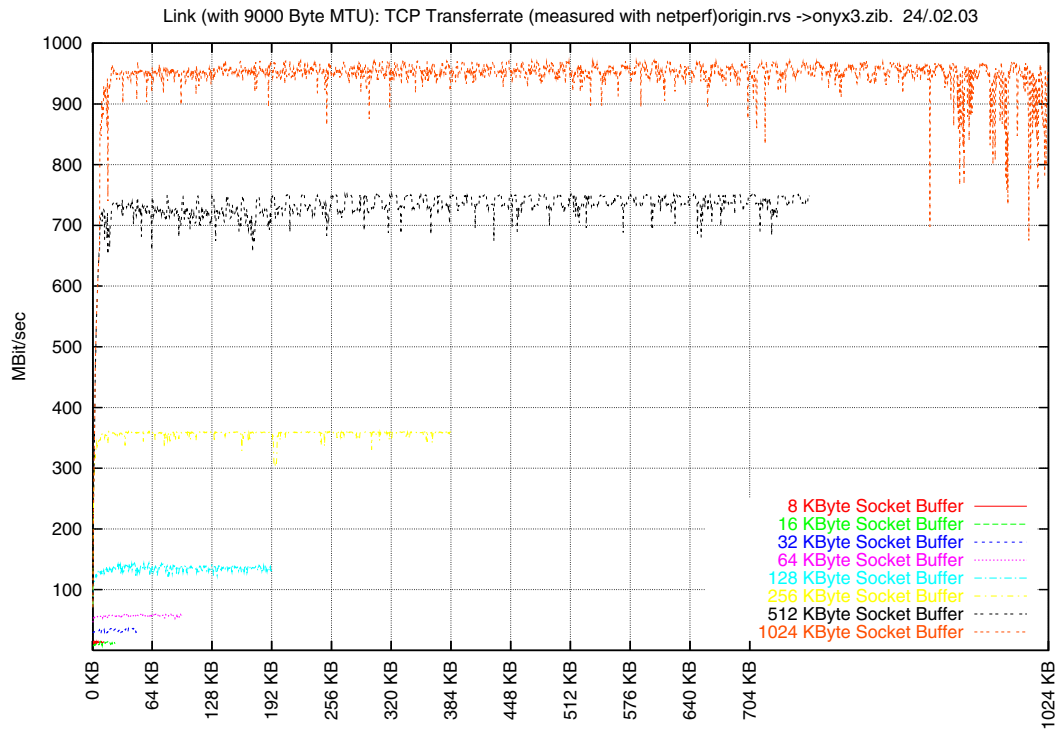


Abbildung 37. Messungen der Datenrate zwischen SGI Origin200 in Hannover und SGI Onyx3800 in Berlin.

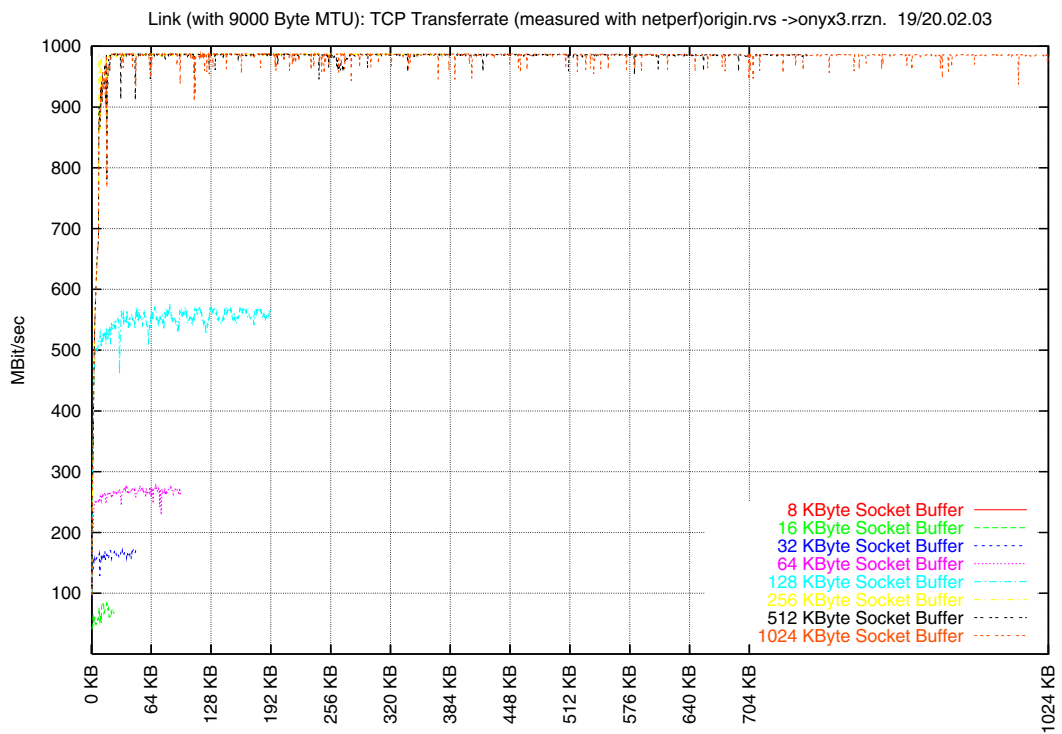


Abbildung 38. Messungen der Datenrate zwischen SGI Origin200 in Hannover und SGI Onyx3800 in Hannover.

3.2 Teilprojekt 2 – 3D/VR-System

3.2.1 Bereitstellung des 3D-Streaming auf breiterer Plattform

Ausgehend von der ursprünglichen prototypischen Implementierung des 3D-Streamingverfahrens unter SGI Irix wurden Portierungen für weitere Betriebssysteme durchgeführt. Inzwischen liegen Versionen der Viewer-Software für die folgenden Plattformen vor:

- HP/UX 10.x / 11.x,
- Linux,
- SGI Irix 6.5.x,
- Sun Solaris 2.x,
- Windows 95 / 98 / NT / 2000 / XP.

Die Viewer-Software wurde als Plugin für WWW-Browser implementiert. Diese Plugin-Schnittstelle wurde ursprünglich von Netscape spezifiziert. Aufgrund bestimmter Unterschiede in der Unterstützung dieses Interfaces war es zunächst problematisch, eine Kompatibilität des 3D-Viewers zu anderen Browsern als Netscape Communicator zu bewerkstelligen. Inzwischen wurden diese Probleme jedoch weitgehend gelöst, so dass eine Nutzung auch im Zusammenhang mit weiteren WWW-Browsern erfolgen kann. Auf der Windows-Plattform wurden z. B. die folgenden Browser erfolgreich getestet:

- Internet Explorer,
- Netscape Communicator,
- Opera.

Gleichzeitig wurde die Palette der unterstützten Browser unter der Linux-Plattform erweitert: Neben Netscape kann das DocShow-Plugin nun auch mit den Internet-Browsern Opera und KDE-Konqueror verwendet werden. Ferner wurde die Kompatibilität für alle gängigen Fenstermanager sicher gestellt.

Die im Rahmen der Portierungsarbeiten zu lösenden Probleme betrafen – neben der Vereinheitlichung der Sourcen und Aufbau eines CVS-Repositorys, incl. Web-Interface – im Wesentlichen die folgenden Aspekte, die sich als plattformabhängig erwiesen haben:

Multithreading

Datentransport und 3D-Rendering laufen zeitlich überlappt ab, um einen Nutzen aus gegebenenfalls vorhandenen Mehrprozessor-Architekturen ziehen zu können. Da die Nutzung von Thread-Bibliotheken auf den Unix-Plattformen verschiedenen deutlichen Einschränkungen unterworfen ist, wurde dort generell auf das Abspalten von Prozessen umgestellt, wobei der Datenaustausch über Shared-Memory-Bereiche geschieht. Unter Microsoft Windows werden Threads verwendet.

Shared Memory und Semaphore

Die Verwendung von separaten Prozessen für Datentransport und 3D-Rendering erfordert eine effiziente Datenübertragung zwischen diesen. Da im Gegensatz zu Threads Shared-Memory-Bereiche nicht standardmäßig verfügbar sind, müssen diese explizit angefordert werden. Dafür wurde der Unix-Systemcall *mmap()* ausgewählt, der eine Vererbung an Kindprozesse ermöglicht, die mit *fork()* erzeugt werden. Es stellte sich heraus, dass auf allen Unix-Plattformen eine unterschiedliche Parametrisierung erforderlich ist. Die jeweiligen plattformspezifischen Lösungen wurden in einer separaten Funktion gekapselt. Dies trifft auch für die Realisierung der Semaphore zu, die benötigt werden, um kritische Bereiche im Zusammenhang mit dem Zugriff auf den gemeinsamen Speicherbereich durch gleichzeitig laufende Threads bzw. Prozesse zu entflechten.

TCP/IP

Die Windows-Socketlibrary weist zum Teil ein etwas unterschiedliches Verhalten im Vergleich zum sonst üblichen Socket-API unter Unix-Betriebssystemen auf. Aufgrund der Tatsache, dass die TCP/IP-Read- und Write-Funktionen bereits gekapselt implementiert waren, war es relativ leicht, diese an die relevanten Unterschiede entsprechend anzupassen.

Nutzeroberfläche

Da nur eine relativ begrenzte Anzahl zusätzlicher Menüs und Dialoge benötigt wird, wurde entschieden, weiterhin zwei unterschiedliche Implementierungsstrategien für Unix- und Windows-

Plattformen zu verfolgen. Die unter Unix begonnene Version basiert auf X-Windows und OSF-Motif, welches erfreulicherweise inzwischen auf sämtlichen relevanten Unix-Varianten, auch im Zusammenhang mit WWW-Browsern und deren Plugins, unterstützt wird. Unter den Windows-Betriebssystemen wurde direkt Win32 verwendet.

3.2.2 Implementierung der Visualisierungsfunktionalität in Amira

Aufgrund der technischen Entwicklungen der letzten Jahre wurde von der Planung für die in diesem Arbeitspunkt vorgesehene Implementierung eines DVR-Moduls für Amira abgewichen.

Zum einen hat es sich als nicht nutzbringend herausgestellt, ein zum DocShow-VR-Client (Netscape-Plugin) kompatibles Amira-Modul zu schreiben. Hierzu müsste der Quelltext des Netscape-Plugins angepasst und im Weiteren müssten sowohl Plugin als auch Amira-Modul parallel auf dem gleichen Stand gehalten werden. Der hierbei entstehende Mehrwert ist jedoch gering, da die Basisfunktionalität sich nicht erweitern würde. Desweiteren läuft der vorhandene DocShow-VR-Client auf allen unterstützten Plattformen als Netscape-Plugin stabil (siehe *Abschn. 3.2.1*), so dass der Aufwand zur Konvertierung in ein Amira-Modul mittlerweile nicht mehr so lohnenswert wie zur Antragstellung erscheint.

Zum anderen wurde mit dem Streaming-HDF5-Protokoll [4][9] eine Möglichkeit gefunden, nicht nur Geometrien, sondern auch Rohdaten zu übertragen. Somit können die mittels der PaDEV aus einem Simulationsprogramm gewonnenen Rohdaten mittlerweile in Amira visualisiert werden. Hierfür wurde in die PaDEV ein Streaming-HDF5-Modul integriert. HDF5 ist ein Binärformat für wissenschaftliche Daten, welches weit verbreitet und standardisiert ist. Die Repräsentation von Roh- und Geometriedaten innerhalb von HDF5 erfolgt mittels der am ZIB entwickelten Bibliothek F5 [7]. Die von der PaDEV (Server) gesendeten Daten werden von einem Amira-Modul (Client) entgegengenommen, welches insbesondere mittels der Fiber-Bundles von Werner Benger (ZIB) implementiert wurde [3]. Damit lassen sich nicht nur zeitabhängige Daten auf festen Gittern übertragen, sondern auch einer eventuellen Änderung der Gitterstruktur über die Zeit wird Rechnung getragen.

Desweiteren wurde eine Technik in die PaDEV integriert, die einen Fernzugriff (Steering) auf alle Algorithmen der PaDEV zulässt. Hierbei erzeugt die PaDEV ein TCL-Skript, welches in Amira eine Benutzeroberfläche generiert. Dort können dann Änderungen an Parametern vorgenommen und an die PaDEV zurückgeschickt werden. Diese Steeringtechnik wurde so flexibel implementiert, dass sie sich ohne weiteres auch auf andere Clients neben Amira erweitern lässt, sofern diese textkodierte Parametersätze entgegennehmen können.

Abbildung 39 zeigt die Steering-Oberfläche des Marching-Cubes-Algorithmus der PaDEV in Amira. Neben dem Isowert lassen sich auch Modifikationsparameter für das Grid beeinflussen. Somit können Sub-Volumina ausgewählt und Sub-Sampling eingestellt werden.

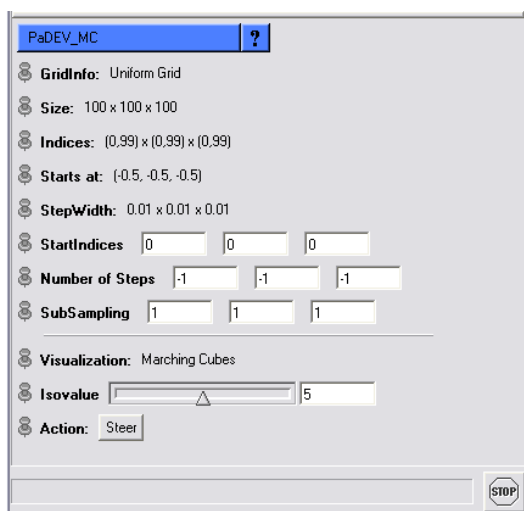


Abbildung 39. Steering-Oberfläche des Marching-Cubes-Algorithmus der PaDEV in Amira.

3.2.3 Weiterentwicklung, Evaluierung

3.2.3.1 Weiterentwicklung

Die in Anwendungsversuchen gewonnenen Erfahrungen und Hinweise der Nutzer führten zur Weiterentwicklung der Streaming-Client- und -Server-Software (z. B. Funktionalitätserweiterungen, Stabilitätsverbesserungen, Modifikationen der Nutzeroberfläche, Leistungsoptimierungen), Ergänzungen in den verwendeten DVRP- und RTSP-Protokollen (siehe auch *Abschn. 3.4.1*) sowie zur Weiterentwicklung der libDVRP-Software (3D-Generator-Client).

3D-Streaming-Client (3D-Viewer-Plugin „DocShow-VR“)

- Verbesserung der Stabilität der Streamingfunktionen (PLAY, STOP, PAUSE, OPTIONS).
- Konfigurationsparameter zur Spezifikation der TCP/IP-Socketbuffer-Größe.
- Zusätzliche optimierte Renderingroutinen zur effizienten Unterstützung weiterer Geometrie- und Attribut-Kombinationen.
- Erweiterte Optionen zur Darstellung von Renderingprimitiven.

Hierzu sind insbesondere die Kugel-Primitive zu nennen. Für die Strömungsvisualisierung können diese z. B. verwendet werden, um Partikelanimationen darzustellen. Auf Client-Seite können statt einer Kugeldarstellung verschiedene andere Grundkörper ausgewählt werden, z. B. platonische Körper oder auch Punkte mit variablem Durchmesser. Dies ist wichtig, um einen Kompromiss aus Qualität der Darstellung und Flüssigkeit der Bewegung treffen zu können.

Darüber hinaus können an diese Partikel nun auch Linienzüge angehängt werden, wobei optional die Farbe an den Stützstellen variiert werden kann. Diese erweiterten Grafikprimitive werden auch durch die weiterentwickelte libDVRP (s. u.) unterstützt. Die Linienzüge können z. B. zur Visualisierung von Bahnlinien eines Strömungsfeldes eingesetzt werden. In einem Menü kann von einer simplen emissiven Darstellung auf ein aufwändigeres Renderingverfahren „Illuminated Stream Lines“ [22] (siehe auch *Abbildung 53*) umgeschaltet werden. Dieses wurde in OpenGL implementiert.

Bei hoher Dichte von Bahnlinien, wie sie etwa in der atmosphärischen Konvektion auftreten, kann die Wahrnehmung der relativen Positionierung einzelner Bahnlinien beeinträchtigt sein. Zur Entflechtung dichter Szenerien und Verbesserung der räumlichen Trennung wurde aus diesem Grund der innovative Ansatz der „Haloed Lines“ (vgl. [17]) umgesetzt. Im Verbund mit der bereits vorhandenen Technik „Illuminated Stream Lines“ und einer zusätzlichen „Transparency“-Option wird eine hervorragende Wahrnehmbarkeit von Bahn- bzw. Stromlinien erzielt.

Zur Verdeutlichung der beschriebenen Rendering-Optionen sei auf die Anwendungsmöglichkeiten im Teilprojekt „Atmosphärische Konvektion“ und zugehörige Renderings in *Abbildung 40* verwiesen.

3D-Streaming-Server

- Funktionale Erweiterungen sowie Effizienz- und Stabilitätsverbesserungen.
- Instrumentierung, d. h. Integration von Messfunktionen, Ausgabe auf Log-Datei.
- Automatisierung der Erzeugung von Metadaten (*.html, *.dvr).
- Konfigurationsparameter zur Spezifikation der TCP/IP-Socketbuffer-Größe.
- Konfigurationsparameter zur Auswahl der TCP/IP-Nodelay-Option.

DVRP/RTSP-Protokollerweiterungen

- Optionen im Sphere-Datenelement (DVRP):
Sphere-Set mit variablem Radius und Einfärbung, Line-Strip-Ergänzung.
- Unterstützung von SET_PARAMETER-Befehlen (RTSP), z. B. zur Wahl einer alternativen Auflösung, auch während laufender 3D-Streams.
- Unterstützung von GET_PARAMETER-Befehlen (RTSP), z. B. zur Abfrage der vom Streamingserver zuletzt gemessenen Bitrate und Anzeige in der Statuszeile des WWW-Browsers.

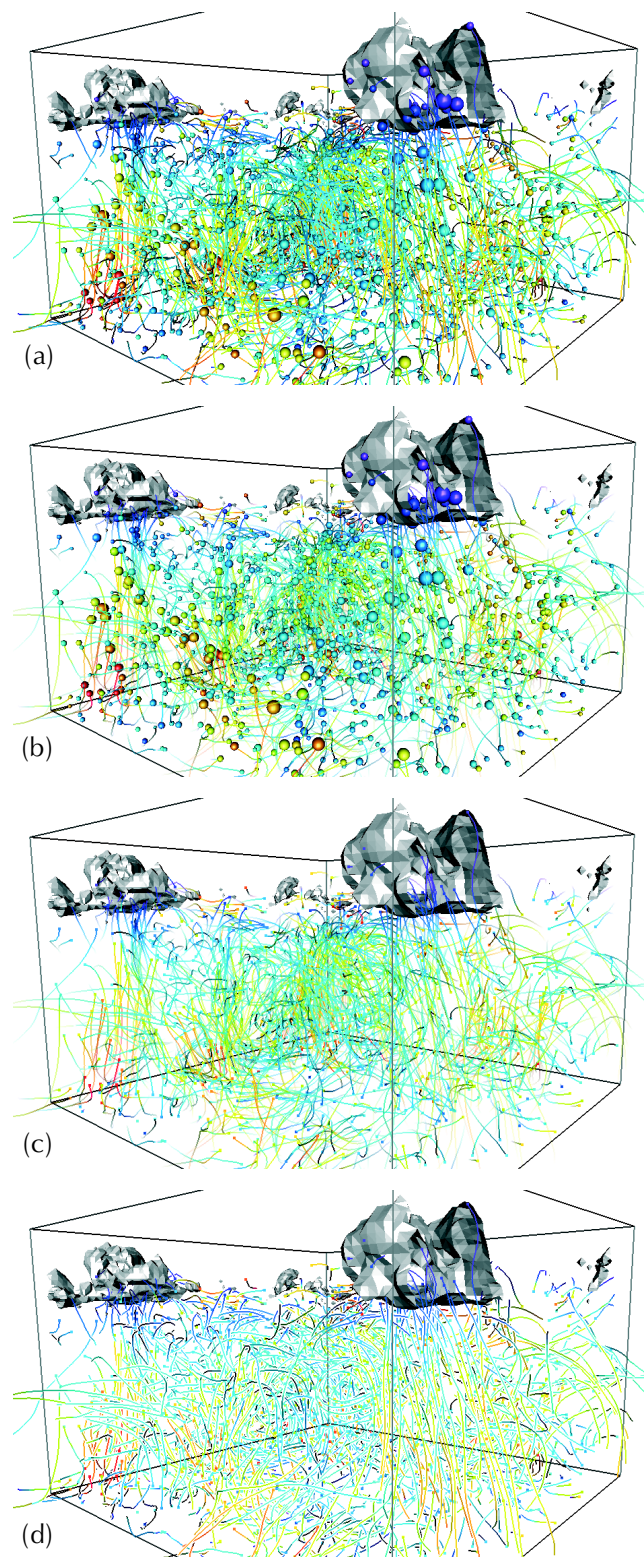


Abbildung 40. Beispiele für den Einsatz verschiedener Renderingverfahren zur Strömungsvisualisierung. Anwendung: „Atmosphärische Konvektion“ (Luftfeuchtigkeit als Isosurface visualisiert). (a) Partikelverlagerung mit Kugeln und Bahnlinien als „Illuminated Lines“ dargestellt, (b) Bahnlinien mit nach hinten zunehmender Transparenz dargestellt, (c) Partikel als Punkte dargestellt, (d) Bahnlinien als „Haloed Lines“ dargestellt.

3D-Generator-Client (libDVRP-Software)

Bei der Weiterentwicklung der libDVRP (parallele Datenextraktion und Geometriegenerierung) wurden folgende Zielsetzungen verfolgt:

- Erweiterung der Funktionalität,
- Vereinfachung der Schnittstelle,
- Portierung auf weitere Rechnerarchitekturen.

Zusätzliche geometrische Grundkörper (Polygone, Kegel, Zylinder) sowie Dipol-Partikel erweitern die bisherige Funktionalität der libDVRP, deren Schnittstellen bislang im Wesentlichen auf höherem Abstraktionsniveau lagen (z. B. Übergabe von Daten auf regulärem Gitter), um daraus intern polygonale Objekte (z. B. Isosurface, Slicer) zu erzeugen. Sie realisieren konkrete Anforderungen aus Anwendungsprogrammen, insbesondere der molekulardynamischen Simulation (*Abschn. 3.5*) und der Simulation von Strömungsverhältnissen in Hafeneinfahrten (*Abschn. 3.6*). Darüber hinaus werden sie zukünftig auch im Zuge weiterer Kooperationen wichtig.

Die libDVRP wurde während der Projektlaufzeit stetig um neue Komponenten ergänzt. Die hinzuge tretene Funktionalität geht zum Teil über die im Projektplan aufgeführten Punkte hinaus. Im Detail handelt es sich um:

- *Darstellung von Punkten, Linien und Polygonen*
Die libDVRP verfügt als zusätzliche geometrische Basisobjekte über Punktmengen und Linienzüge. Außerdem wurde eine Schnittstelle zur Darstellung von Polygonen und polygonalen Netzen implementiert.
- *Dipol-Partikel*
Dipol-Partikel — Teilchen mit Polarisierung — wurden originär in der libDVRP als eigenständige Objekte umgesetzt und werden von der molekulardynamischen Simulation genutzt. Ein Beispiel ist in *Abbildung 41* gezeigt.
- *Molekülvisualisierung*
Die erfolgreiche Kooperation mit dem Institut für Thermodynamik wurde auf weiter führende Anwendungen ausgedehnt und hat zu dem Entwurf einer Schnittstelle zur Darstellung von Molekülgruppen geführt. Ein Beispiel ist in *Abbildung 42* gezeigt.
- *Feldlinien*
Zur Visualisierung von Vektorfeldern wurde die Darstellungsmöglichkeit durch Feldlinien geschaffen. Diese liegt zurzeit allerdings lediglich in einer nicht-parallelen Implementierung vor. Eine parallelisierte Version ist in Vorbereitung. Ein Beispiel ist in *Abbildung 43* gezeigt.
- *Schnittebenen*
Die Schnittstelle zur Erzeugung von Schnittebenen wurde hinsichtlich der Farbgebung vereinfacht.

Die libDVRP wird zurzeit vor allem von Anwendungen genutzt, die in der Programmiersprache Fortran 90/95 entwickelt wurden (siehe Abschnitte 3.4 und 3.5). Um die Benutzung der libDVRP für diese Anwendergruppe zu vereinfachen, wurde eine Schnittstelle zu der in C programmierten libDVRP in ein DVRP-Fortran-95-Modul gekapselt. Dadurch konnten Plattformabhängigkeiten und typische Fehlerquellen der Fortran-C-Schnittstelle, z. B. bei der Feldübergabe, eliminiert werden.

Im Zuge geplanter Anwendungsprojekte sowie erweiterter Anforderungen der bereits bestehenden Projekte (Abschnitt 3.4 und 3.5) ergab sich die Notwendigkeit, die libDVRP auf bisher nicht unterstützte Rechnerarchitekturen zu portieren. Als wichtigste Plattform ist zu der SGI/Cray T3E der HLRN-Rechner (IBM Regatta p690, AIX 5.1) hinzugekommen. Daneben werden nun außerdem die Plattformen Sun Enterprise (Solaris), HP-Parallelrechner (HP-UX), SGI Onyx (IRIX) sowie Linux-PC unterstützt. Die Vielzahl der unterstützten Plattformen machten Maßnahmen zur Minimierung des Portierungsaufwandes erforderlich, die zugleich die zukünftige Wartung/Ergänzung der libDVRP auf den genannten Plattformen vereinfachen werden.

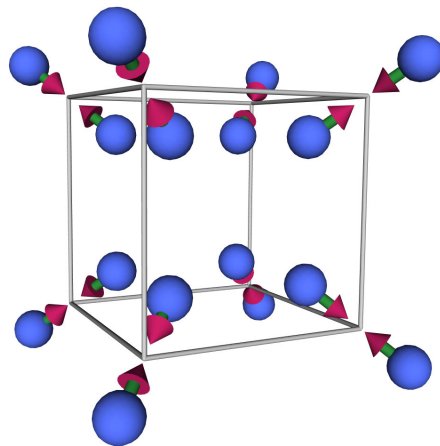


Abbildung 41. Dipol-Partikel erweitern die Funktionalität der libDVRP. Gezeigt ist eine Ansicht aus einem der Demonstrationsprogramme, die Bestandteil der libDVRP-Dokumentation sind.

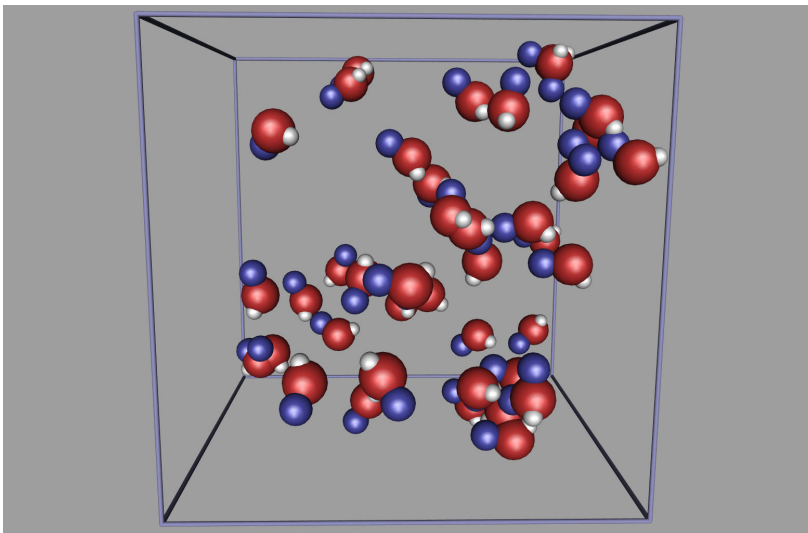


Abbildung 42. Erweiterung der libDVRP: Visualisierung von Molekülen.

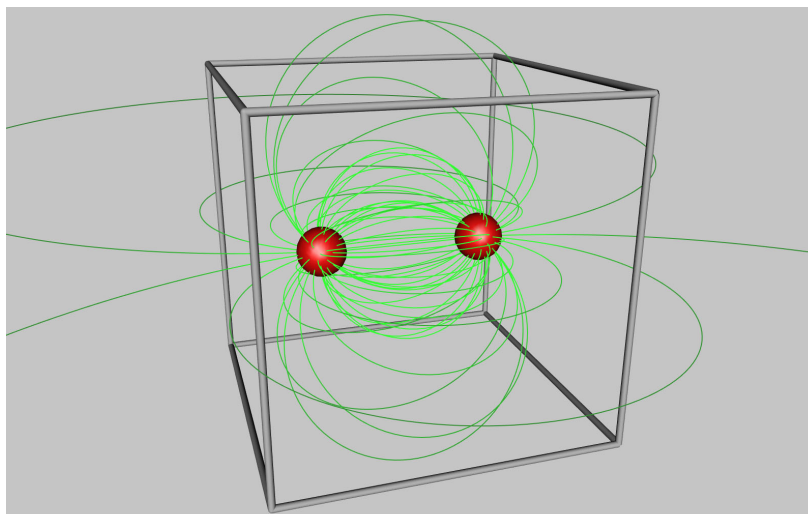


Abbildung 43. Visualisierung von Feldlinien am Beispiel eines elektrischen Dipolfeldes.

3.2.3.2 Evaluierung

Zur Messung der Durchsatz- und Renderingraten in einer praktischen 3D-Streaming-Anwendung wurde ein Testdatensatz aus der Anwendung „Ozeanische Konvektion“ verwendet. Er enthielt die letzten 142 Szenen (Nr. 800–941, siehe auch *Abbildung 44*) einer Temperatur- und Strömungsfeld-Visualisierung im DVR-Format mit einem Gesamtdatenvolumen von 890.662.096 byte („mov“), bzw. in einer überarbeiteten Version 397.220.816 byte („mov_opt“ mit Triangle-Strip-Optimierung).

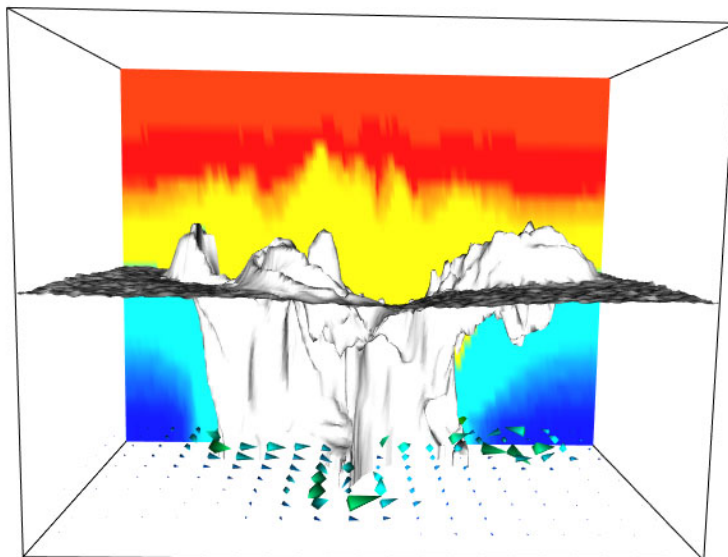


Abbildung 44. Rendering der 3D-Szene Nr. 941 aus dem Datensatz „Ozeanische Konvektion“ (Temperaturfeld: Isosurface und Colored Slicer, Strömungsfeld: Arrows).

Im Datensatz **mov** hat diese Szene (DVR-Format) ein Datenvolumen von 6.263.624 byte, in **mov_opt** (Triangle-Strip-Optimierung) 2.821.200 byte.

Die 3D-Szene enthält in beiden Fällen 86.059 Grafikprimitive unterschiedlicher Art.. Dargestellt ist eine Auflösung von 640 x 480 pixel (mit Antialiasing, 8 samples/pixel), wie sie auch in den Durchsatz- und Renderingmessungen eingestellt wurde.

Der Verlauf der Dateigrößen über die verwendete Szenensequenz ist in *Abbildung 45* dargestellt. Die auf der Servermaschine SGI Origin200 (s. u.) gemessenen Lesedatenraten (jeweils bei 3 Versuchen) sind für „mov“ in *Abbildung 46* und für „mov_opt“ in *Abbildung 47* visualisiert. Die jeweils erste Messung spiegelt die tatsächliche Plattendatenrate wider, in den folgenden Messungen kommen die Daten dagegen aus dem Dateisystem-Cache des Betriebssystems.

Socketbuffer- und Read-/Write-Buffergrößen wurden auf 1 MB eingestellt, Jumbo-Frames wurden aktiviert (MTU=9000), und die Gigabit-Ethernet-Interfaces der beteiligten SGI-Rechner wurden mit ifconfig auf rspace=10 MB und sspace=10 MB konfiguriert (Receive- und Sendspace). Auf Client- und Serverseite wurde die TCP-Option NODELAY eingeschaltet, um kurze Latenzen zu erreichen.

Die verwendeten Rechner hatten folgende Eigenschaften:

- RRZN: SGI Origin200 (Irix 6.5.17f), 2 CPUs R12000, 270 MHz, 2 GB Hauptspeicher, Gigabit-Ethernet-Interface eg0.
- RRZN: SGI Onyx3800 (Irix 6.5.18m), 18 CPUs R14000, 600 MHz, 34 GB Hauptspeicher, Gigabit-Ethernet-Interfaces tg0/tg1, Grafik: 4 Infinite Reality 4 mit je 1 RM (30 bit/pixel, 8 samples/pixel, 1600 x 1024 pixel, 80 Hz).
- ZIB: SGI Onyx3800 (Irix 6.5.19m), 20 CPUs R14000, 500 MHz, 20 GB Hauptspeicher, Gigabit-Ethernet-Interface eg0, Grafik: 3 Infinite Reality 4 mit je 4 RM (30 bit/pixel, 8 samples/pixel, 1280 x 1024 pixel, 60 Hz), 1 IR3- und 2 IR2-Grafiksubsysteme.

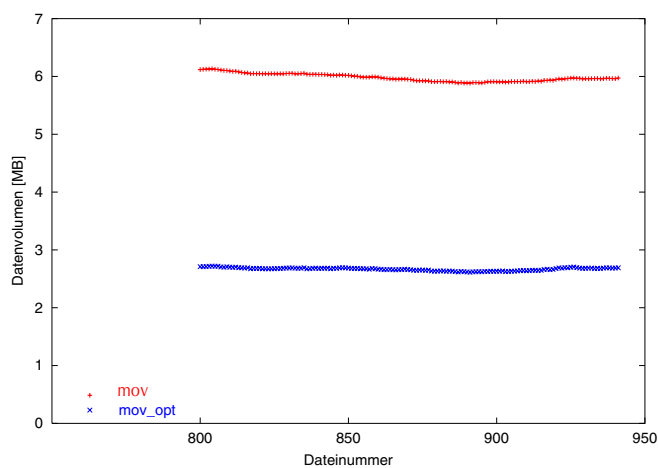


Abbildung 45. Datenvolumen der in den beiden Testdatensätzen enthaltenen jeweils 142 3D-Szenen. (1 MB = 2^{20} byte)

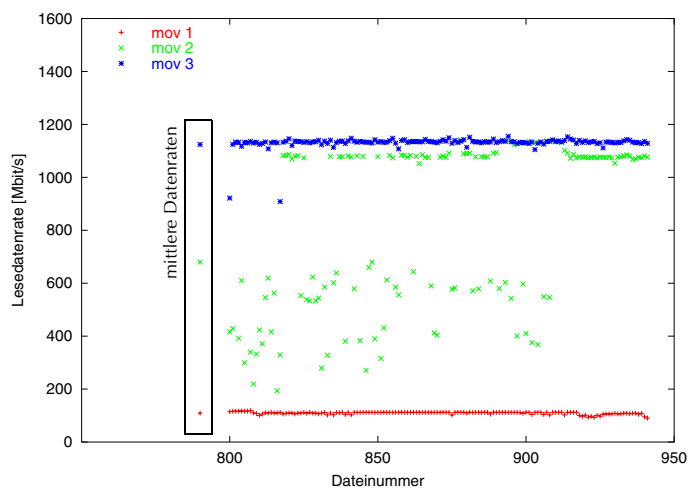


Abbildung 46. Lese-Datenrate der 3D-Szenen (DVR-Format) des Testdatensatzes „mov“. (3 aufeinander folgende Versuche)

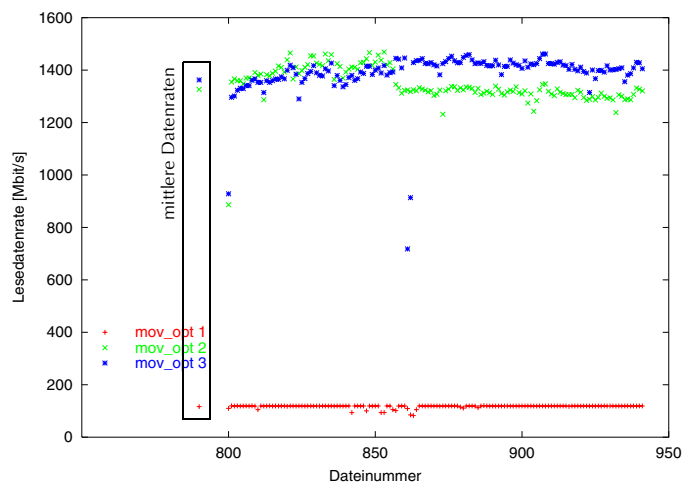


Abbildung 47. Lese-Datenrate der 3D-Szenen (DVR-Format) des Testdatensatzes „mov_opt“. (3 aufeinander folgende Versuche)

Mit unterschiedlichen Renderingoptionen wurden jeweils drei Ausspielvorgänge der oben beschriebenen Szenensequenz vermessen. Die Tests wurden sowohl im lokalen Gigabit-Ethernet-Netzwerk (RRZN, zwischen Origin200 und Onyx3800: *Tabelle 4* und *Tabelle 5*) als auch im Weitverkehrsnetz (RRZN–ZIB, zwischen Origin200 und Onyx3800: *Tabelle 6* und *Tabelle 7*) durchgeführt. Bemerkenswert ist die nahezu vollständige Sättigung des Gigabit-Netzes, sowohl im LAN als auch im WAN. Allerdings limitiert die Renderingleistung der verwendeten Grafikkarte (als Rendering-Subsystem wurde jeweils eine IR4 verwendet, im RRZN mit 1 RM, im ZIB mit 4 RM) den Gesamtdatendurchsatz je nach aktivierten Rendering-Optionen erheblich. Dies ist insbesondere bei stereoskopischer Darstellung (Quadbuffer-Modus erfordert zwei sequenzielle Rendering-Durchläufe, optimierbar durch paralleles Multipipe-Rendering), aber auch bereits bei aktiviertem „two-sided lighting“ zu beobachten. Weiterhin ergibt sich durch die jeweilige Framerate des Displaygenerators eine Limitierung, da sich die jeweilige Updatezeit stets als ganzzahliges Vielfaches des Kehrwertes der Framerate ergibt.

Durchsatzmessungen im Gigabit-LAN (RRZN)

„mov“ (142 frames)	Rendering off		Rendering on			
			Two-sided off		Two-sided on	
Mono	979,9 Mbps	19,5 fps	941,1 Mbps	18,8 fps	796,4 Mbps	15,9 fps
Stereo			553,8 Mbps	11,0 fps	421,7 Mbps	8,4 fps

Tabelle 4. Mittlere Datenrate (in Mbit/s) und Bildrate (in frames/s) der Sequenz „mov“ (Ozeanische Konvektion) bei verschiedenen Rendering-Optionen (mono/stereo, two-sided lighting off/on) bzw. ohne Rendering in einer verteilten 3D-Streaming-Anwendung zwischen einer Origin200 (RRZN) und einer Onyx3800 (RRZN). Renderingrate und -zeit von Nr. 941: 1.449 Mbps / 34,6 ms bzw. 970,2 Mbps / 51,6 ms (two-sided).

„mov_opt“ (142 frames)	Rendering off		Rendering on			
			Two-sided off		Two-sided on	
Mono	980,8 Mbps	43,8 fps	595,8 Mbps	26,6 fps	460,8 Mbps	20,6 fps
Stereo			377,2 Mbps	16,9 fps	279,3 Mbps	12,5 fps

Tabelle 5. Mittlere Datenrate (in Mbit/s) und Bildrate (in frames/s) der Sequenz „mov_opt“ (Ozeanische Konvektion) bei verschiedenen Rendering-Optionen (mono/stereo, two-sided lighting off/on) bzw. ohne Rendering in einer verteilten 3D-Streaming-Anwendung zwischen einer Origin200 (RRZN) und einer Onyx3800 (RRZN). Renderingrate und -zeit von Nr. 941: 1.271 Mbps / 17,8 ms bzw. 684,2 Mbps / 33,0 ms (two-sided).

Durchsatzmessungen im Gigabit-WAN (RRZN – ZIB)

„mov“ (142 frames)	Rendering off		Rendering on			
			Two-sided off		Two-sided on	
Mono	971,6 Mbps	17,9 fps	851,6 Mbps	17,0 fps	741,7 Mbps	14,8 fps
Stereo			515,0 Mbps	10,3 fps	428,0 Mbps	8,5 fps

Tabelle 6. Mittlere Datenrate (in Mbit/s) und Bildrate (in frames/s) der Sequenz „mov“ (Ozeanische Konvektion) bei verschiedenen Rendering-Optionen (mono/stereo, two-sided lighting off/on) bzw. ohne Rendering in einer verteilten 3D-Streaming-Anwendung zwischen einer Origin200 (RRZN) und einer Onyx3800 (ZIB). Renderingrate und -zeit von Nr. 941: 1.431 Mbps / 35 ms bzw. 969 Mbps / 51,7 ms (two-sided).

„mov_opt“ (142 frames)	Rendering off		Rendering on			
			Two-sided off		Two-sided on	
Mono	948,6 Mbps	42,4 fps	529,6 Mbps	23,6 fps	446,8 Mbps	20,0 fps
Stereo			285,8 Mbps	12,8 fps	247 Mbps	11,0 fps

Tabelle 7. Mittlere Datenrate (in Mbit/s) und Bildrate (in frames/s) der Sequenz „mov_opt“ (Ozeanische Konvektion) bei verschiedenen Rendering-Optionen (mono/stereo, two-sided lighting off/on) bzw. ohne Rendering in einer verteilten 3D-Streaming-Anwendung zwischen einer Origin200 (RRZN) und einer Onyx3800 (ZIB). Renderingrate und -zeit von Nr. 941: 1.133,8 Mbps / 19,9 ms bzw. 673,8 Mbps / 33,5 ms (two-sided).

Abhängigkeit von der Socketbuffergröße im Vergleich zwischen LAN- und WAN-Konfiguration

Mit dem Parameter TCP/IP-Socketbuffergröße bzw. Socket-Read/Write-Buffergröße (Bestandteil der 3D-Viewer-Konfiguration, die in der Datei dvrcnf.txt vom Anwender verändert werden kann bzw. in einer Konfigurationsdatei des 3D-Streamingsservers) wurde analog verfahren (Tabelle 8).

TCP/IP-Socket-und Read/Write-Buffergröße	Gigabit-Ethernet-LAN (RRZN)				Gigabit-Ethernet-WAN (RRZN-ZIB)			
	ohne Rendering		Rendering: mono, two-sided off		ohne Rendering		Rendering: mono, two-sided off	
64 kB	323,1	6,4	319,3	6,4	82,5	1,6	82,4	1,6
128 kB	527,8	10,5	515,3	10,3	170,3	3,4	167,5	3,3
256 kB	854,6	17,0	781,8	15,6	346,8	6,9	337,9	6,7
512 kB	968,9	19,3	902,8	18,0	678,1	13,5	650,7	13,0
1 MB	979,9	19,5	941,1	18,8	971,7	19,4	851,6	17,0

Tabelle 8. Mittlere Datenrate (jeweils in der linken Spalte, in Mbit/s) und Bildrate (jeweils in der rechten Spalte, in frames/s) der Sequenz „mov“ (Ozeanische Konvektion), gemessen auf Server-Seite: Maximum bei drei aufeinander folgenden Versuchen.
(1 kB = 1024 byte, 1 MB = 1024 kB)

An dem Verlauf der Datenrate ist deutlich erkennbar, dass genügend hohe Buffergrößen zur Erzielung einer guten Ressourcenausnutzung in dieser Anwendung – insbesondere aufgrund der hohen Window-size-Anforderung bei Verwendung der WAN-Kopplung (hier: Gigabit-Ethernet zwischen Berlin und Hannover) – zwingend erforderlich sind.

Die Anforderung an die ausreichende Dimensionierung der TCP/IP-Socketbuffergröße (hier: mindestens ca. 1 MB für Gigabit-Ethernet zwischen Berlin und Hannover) gilt natürlich auch für beliebige andere Anwendungen bzw. Middleware, die über TCP/IP kommuniziert. Zu nennen sind beispielsweise: Message-Passing (z. B. MPI), Dateitransfer (z. B. rcp) oder der transparente Zugriff auf entfernte Dateisysteme (z. B. nfs). Auch hier gilt, dass die TCP/IP-Socketbuffergröße sowie die in diesem Zusammenhang noch zu erwähnende TCP/IP-Nodelay-Option und die Socket-Read/Write-Buffergrößen geeignet eingestellt werden bzw. überhaupt erst einstellbar zur Verfügung gestellt werden können, wie dies beim 3D-Streamingverfahren bereits der Fall ist (siehe Abschn. 3.1.2 und Abschn. 3.2.3.1).

mov.html:

```
<html>
<body bgcolor=#ffffff>
<h3>Konvektion</h3>
<embed src="mov.dvrs" width="640" height="480"
  headlightintensity="200" noscene="1"
  noprogressive="1" notransparency="1">
</body>
</html>
```

mov.dvrs:

```
SCENES=1
MED=TCP
ADR=rtsp://origin-ge.rvs.uni-hannover.de/olbrich/imuk_express/mov.***.dvr
MAXBYTES=7000000
FRAMES=942
FRAMES_P_SEC=80
SCENEEND
```

Abbildung 48. Listings der HTML- und DVRS-Dateien, die in den beschriebenen Tests („mov“) verwendet wurden.

3.2.4 Schnittstelle zwischen 3D-Generator, 3D-Server und 3D-Viewer

Es wurden allgemein anwendbare Online-Visualisierungs- und Steering-Funktionalitäten entwickelt, die in dem verteilten 3D-Streaming-System an drei Instanzen Ergänzungen in den existierenden Software-Modulen erforderten (siehe auch *Abschn. 3.4.1*):

- 3D-Generator (libDVRP),
- 3D-Server (3D-Streamingserver) und
- 3D-Viewer (DVR-Plugin für WWW-Browser).

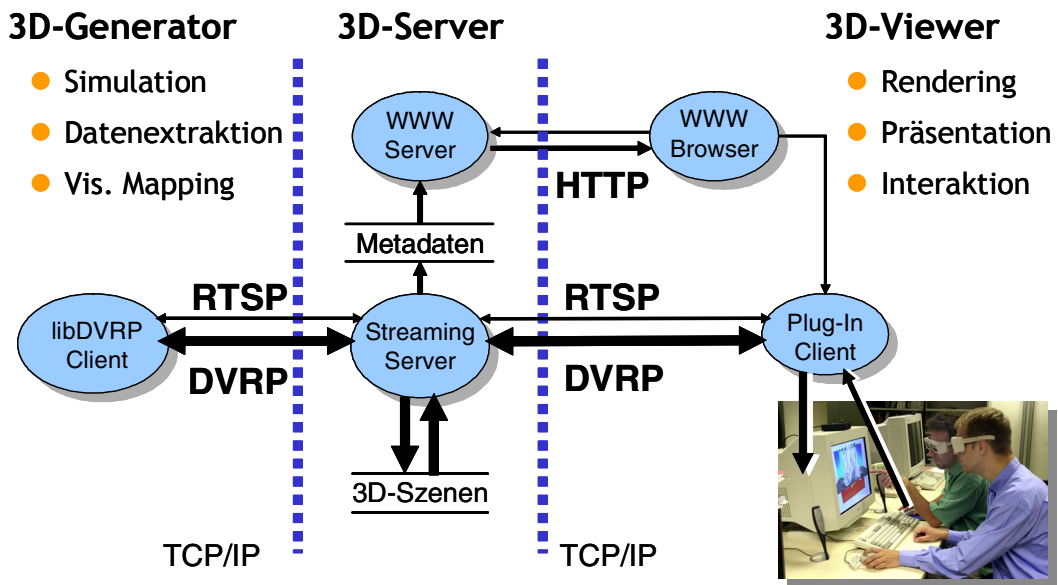


Abbildung 49. Zusammenspiel der netzverteilten Komponenten 3D-Generator, 3D-Server und 3D-Viewer.

Das Hauptziel war die Bereitstellung eines interaktiven Rückkanals, um vom 3D-Viewer aus auf steuerbare Parameter der Simulationsrechnung Einfluss nehmen zu können. Diese Funktionalität wurde letztlich auf der Grundlage des RTSP-Protokolls sowie durch Nutzung des DVRP-Rückkanals realisiert.

Medien-Datenflüsse	
• <i>Codierung:</i>	3D-Szenen (vorverarbeitet für OpenGL-Rendering)
• <i>Video recording:</i>	3D-Generator → 3D-Streamingserver
• <i>Video on demand:</i>	3D-Streamingserver → 3D-Viewer
• <i>Live streaming:</i>	3D-Generator → 3D-Streamingserver → 3D-Viewer
Interaktion mit Datenquelle	
• <i>VCR paradigm:</i>	3D-Streamingserver ← 3D-Viewer
• <i>Interactive Steering:</i>	3D-Generator ← 3D-Streamingserver ← 3D-Viewer
Medienspezifische Skalierungsoptionen	
• <i>Bildrate:</i>	Szenen-Update rate
• <i>Bildauflösung:</i>	Modell-Komplexität (z. B. Polygonanzahl)
• <i>Bildqualität (SNR):</i>	Präzision der 3D-Koordinaten und Attribute

Abbildung 50. 3D-Streaming – Analogie-Betrachtung zum Video-Streaming im Internet.

3D-Generator

Seitens der Simulationsrechnung wird die Steeringfunktionalität durch Bereitstellung zusätzlicher Funktionen in der Bibliothek libDVRP unterstützt. Die von Fortran aus aufrufbaren Methoden heißen:

DVRP_STEERING_INIT

Hiermit werden steuerbare Parameter spezifiziert:

- Name,
- Datentyp (zunächst: int, float, double),
- Zulässiger Wertebereich,
- Referenz auf Variable (damit wird gleichzeitig der Initialwert vorgegeben).

DVRP_STEERING_UPDATE

An dieser Stelle wird die Werte-Aktualisierung durchgeführt, incl. der Verteilung der Werte an sämtliche der in der parallelen Simulationsrechnung beteiligten PEs.

Diese Funktionen stellen intern Protokollelemente bereit, die durch Ergänzungen der RTSP- und DVRP-Datenflüsse (vgl. *Abschn. 3.1.2*) spezifiziert wurden. Als Neuerung wurde die libDVRP im letzten Projekthalbjahr um einen Mechanismus zur ferngesteuerten Simulationsunterbrechung bereichert, wie nachfolgend beschrieben.

3D-Server

Der 3D-Server hat nicht nur die Aufgabe, 3D-Szenen auszuspielen, sondern soll diese nun auch vom 3D-Generator empfangen und speichern können, online an einen dazu passenden 3D-Viewer weiterleiten können sowie einen interaktiven Steering-Rückkanal verwalten können. Dazu mussten mehrere Ergänzungen durchgeführt werden:

RTSP-Protokollelemente

Die Protokollelemente GET_PARAMETER und SET_PARAMETER gestatten nun auch die Spezifikation von steuerbaren Parametern, wie sie im 3D-Generator mit Hilfe der libDVRP-Funktion DVRP_STEERING_INIT vorgegeben werden können.

DVRP-Rückkanal

Der Rückkanal des DVRP-Datenflusses wird verwendet, um die Parameter-Änderungen interaktiv zu übertragen.

DVRS- und HTML-Metadaten-Erzeugung

Der 3D-Streaming-Server erzeugt nach einem bestimmten Schema automatisch die benötigten Metadaten, die vom WWW-Browser benötigt werden, um das 3D-Viewer-Plugin entsprechend zu initialisieren. Diese werden sowohl beim Aufbau der RTSP/DVRP-Verbindung erzeugt (dabei zunächst mit undefinierten Werten für Bildanzahl etc.) als auch noch einmal bei Beendigung.

Vermittlung zwischen Generator- und Viewer-Clients

Auf der Grundlage der Location-Metadaten, die beim jeweiligen RTSP-Setup mitgeteilt werden, übernimmt der 3D-Streamingserver eine Vermittlungsfunktionalität, die zwei Ziele verfolgt:

- Online-Visualisierung – sobald von einem Generator-Client eine neue 3D-Szene empfangen wurde, wird diese an die dazu passenden Viewer-Clients gesendet.
- Interactive Steering – Spezifikationen von Steering-Parametern müssen vom Generator-Client an einen dazu passenden Viewer-Client übermittelt werden, ebenso müssen Mitteilungen über Parameter-Änderungen von einem Viewer-Client an den dazu passenden Generator-Client vermittelt werden.

Da für die verschiedenen Clients auf 3D-Serverseite voneinander unabhängige Server-Prozesse geforkt (als Kindprozess erzeugt) werden, war es erforderlich, den Hauptprozess mit jedem zusätzlichen Server-Prozess durch bidirektionale Pipes zu verbinden, die zur Koordinierung sämtlicher jeweils existierender Server-Prozesse dienen. Der Hauptprozess übernimmt dabei die Vermittlung zwischen den geforkten Server-Prozessen. Es wurde ein einfaches Protokoll zur Interprozess-Kommunikation entwickelt. Über die bereits genannten Pipes werden Protokolldateneinheiten übermittelt, die zurzeit vor allem zur Aktualisierung der jeweiligen Szenen-Anzahl bzw. der Steering-Parameter dienen.

3D-Viewer

Der 3D-Viewer wurde um ein zusätzliches Menü „Steering“ ergänzt. Durch Aktivierung dieses Kontextmenüs wird ein Dialogfenster (dargestellt in *Abbildung 51*) geöffnet, das alle Steuerparameter aufführt, einschließlich verschiedener Nutzerschnittstellen zur Werteänderung (Slider, Textfelder). Das Aussehen dieses Steering-Dialogs hängt letztlich von der Spezifikation auf 3D-Generator-Seite ab. Zusätzlich wurde eine Spezifikationsmöglichkeit über die Metadaten (DVR-Datei) geschaffen.

Ferngesteuerte Simulationssteuerung

Die beschriebene Funktionalität zur ferngesteuerten Simulationssteuerung („Steering“) wurde mit dem Ziel der Vereinfachung für den Anwender im letzten Projekthalbjahr überarbeitet und in wichtigen Punkten erweitert. Gleichzeitig wurden bereits existierende Komponenten einer intensiven Erprobung auf den hauptsächlichen Zielplattformen — IBM Regatta p690 (HLRN), SGI Onyx 3000, Sun Enterprise — unterzogen und die gewonnenen Erfahrungen in die Software integriert. Die Änderungen haben Eingriffe an allen Systemkomponenten, d. h. der libDVRP, dem Streaming-Server und dem DocShow-VR-Viewer erforderlich gemacht.

Im Einzelnen gliedern sich die Erweiterungen wie folgt:

- *Ferngesteuertes Anhalten/Fortsetzen der Simulation*
Die libDVRP stellt für ein Simulationsprogramm mit Steering-Funktionalität nunmehr automatisch einen Mechanismus zum ferngesteuerten Anhalten bzw. Fortsetzen per Knopfdruck bereit (vgl. *Abbildung 51*). Obwohl in der Regel die Steuerung von Parametern bei laufendem Simulationsprogramm geschieht, bietet sich die Option zur Simulationsunterbrechung an, um beispielsweise eine konsistente Einstellung von mehreren Parametern vorzunehmen. Bei der Fortsetzung werden dann automatisch die gesetzten Werte gültig. Ebenso kann eine Simulationspause zur Veränderung von visuellen Parametern der 3D-Szenerie genutzt werden, ohne dass die Simulation weiterläuft. Im Explorationsszenario steht während der Simulationsunterbrechung z. B. Zeit für eingeschobene Diskussionen ohne Voranschreiten der Simulation zur Verfügung.
- *Logarithmische Skalen*
Häufig erstreckt sich der Wertebereich von Simulationsparametern über viele Größenordnungen. Um dennoch eine feinfühligere Justierung zu ermöglichen, bietet das Steering-Fenster die Möglichkeit, den Einstellbereich auf eine logarithmische Skala umzustellen.
- *Aktualisierung des 3D-Fensters*
Für die Simulationssteuerung ist es wichtig, die im Viewer angezeigte 3D-Szene mit dem jeweils aktuellen Simulationsschritt in der Modellrechnung synchronisieren zu können. Modifikationen am Streaming-Server sowie am Streaming-Fenster des Viewers haben diese optionale Synchronisation bewerkstelligt. Das angepasste Streaming-Fenster, eingeblendet in *Abbildung 53*, informiert über den Fortschritt sowohl der angezeigten 3D-Szene als auch der Simulationsrechnung. Die Transmission der jeweils aktuellen am Streaming-Server eintreffenden Simulations-Sequenznummer zum Viewer-Client geschieht mittels eines neuen DVRP-Protokollelements.
- *Dynamische Speicherverwaltung*
Die maximale Dateigröße einer DVR-Sequenz, die bislang als Teil der Meta-Informationen übermittelt wurde, ist für eine laufende Simulationsrechnung nicht a priori bekannt. Daher wurde eine dynamische Speicherverwaltung implementiert, mit deren Hilfe sich der Viewer automatisch auf die Größe der eintreffenden DVR-Dateien einstellt. Derselbe Mechanismus erlaubt es dem Streaming-Server zudem, bereits zu Anfang einer Simulationsrechnung abrufbereite Meta-Daten ohne Angabe der zu erwartenden Dateigröße zu generieren. Nach Abschluss der Simulation werden Meta-Daten im bisherigen Format zur Wahrung der Kompatibilität erzeugt.

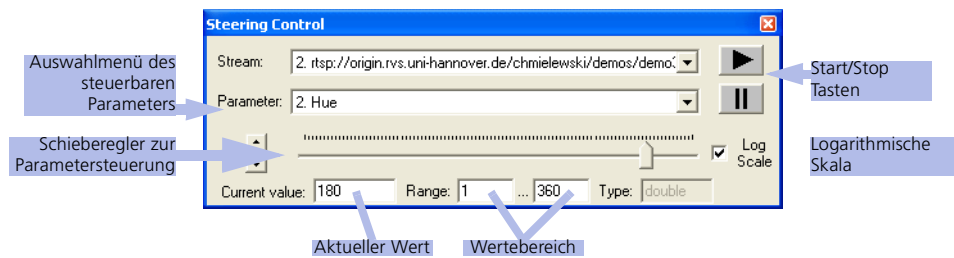


Abbildung 51. Simulationssteuerung: Das Steering-Fenster.

3.2.5 CSCW-Erweiterungen (Computer-Supported Collaborative Working)

Es wurden Protokolle und Verfahren entwickelt, die zur Inter-Client-Kommunikation genutzt werden können. Hierzu gehört die Implementierung von Software-Erweiterungen in der Viewer-Software zur Bereitstellung folgender Funktionen:

- Synchronisierung der Interaktion bzw. Navigation,
- Videoconferencing und Einblendung eines Videofensters.

Diese werden im Folgenden kurz beschrieben.

Synchronisierung der Interaktion bzw. Navigation

Das Protokoll zur Synchronisierung von zwei (oder mehr) Viewer-Systemen war vor Projektstart ein textorientiertes, welches zum funktionalen Test eines Prototypen für SGI-Irix entwickelt wurde. Im Rahmen des Projekts wurde ein binäres Protokoll neu spezifiziert (Abbildung 52). Die Inter-Client-Kommunikation über TCP/IP wurde plattformunabhängig implementiert, so dass eine Interoperabilität zwischen beliebigen Windows- und Unix-Systemen gegeben ist. Zur Unterstützung dreidimensionaler Zeigefunktionen wurde außerdem ein 3D-Telepointer eingeführt, der mit der Synchronisierungsfunktion ferngesteuert werden kann. Das Erscheinungsbild dieses Pointers kann individuell modelliert werden. Es wird mit einer konfigurierbaren, im Plugin-Verzeichnis angelegten DVR-Datei – die z. B. aus einer VRML-Datei konvertiert werden kann – spezifiziert.

Videoconferencing und Einblendung eines Videofensters

Zur Audio-/Videoübertragung wurde festgelegt, H.323-basierte Hardware-Codecs einzusetzen (siehe auch Abschn. 3.1.1). Sowohl in Berlin als auch in Hannover wurden dazu mehrere Videoconferencing-Geräte des Typs Polyspan Viewstation SP beschafft und auch bei den Anwendungspartnern installiert.

In die Viewer-Software wurden Funktionen integriert, die es gestatten, ein Video-Interface („Grabber“, z. B. Win-TV-Karte im PC oder DIVO in SGI-Onyx) im Zusammenhang mit der 3D-Betrachtung zu verwenden. Dabei wird ein konfigurierbares Fenster in die 3D-Szene einblendet, das unabhängig von der eigentlichen Visualisierung dreidimensional verschoben, gedreht und skaliert werden kann und auf dem eine Videotextur (mittels OpenGL) dargestellt wird. Diese wiederum entspricht dem Videobild, das bei einem Raumkonferenzsystem üblicherweise auf einem Fernseher wiedergegeben wird und nun über ein Video-Interface in die 3D-Grafikwerkstation eingespeist wird. Im Zusammenhang mit der oben beschriebenen Synchronisierungsfunktion wird somit eine tele-immersive Visualisierung, also eine erweiterte Form des „Computer-Supported Collaborative Working“ (CSCW), ermöglicht (siehe auch Abbildung 53).

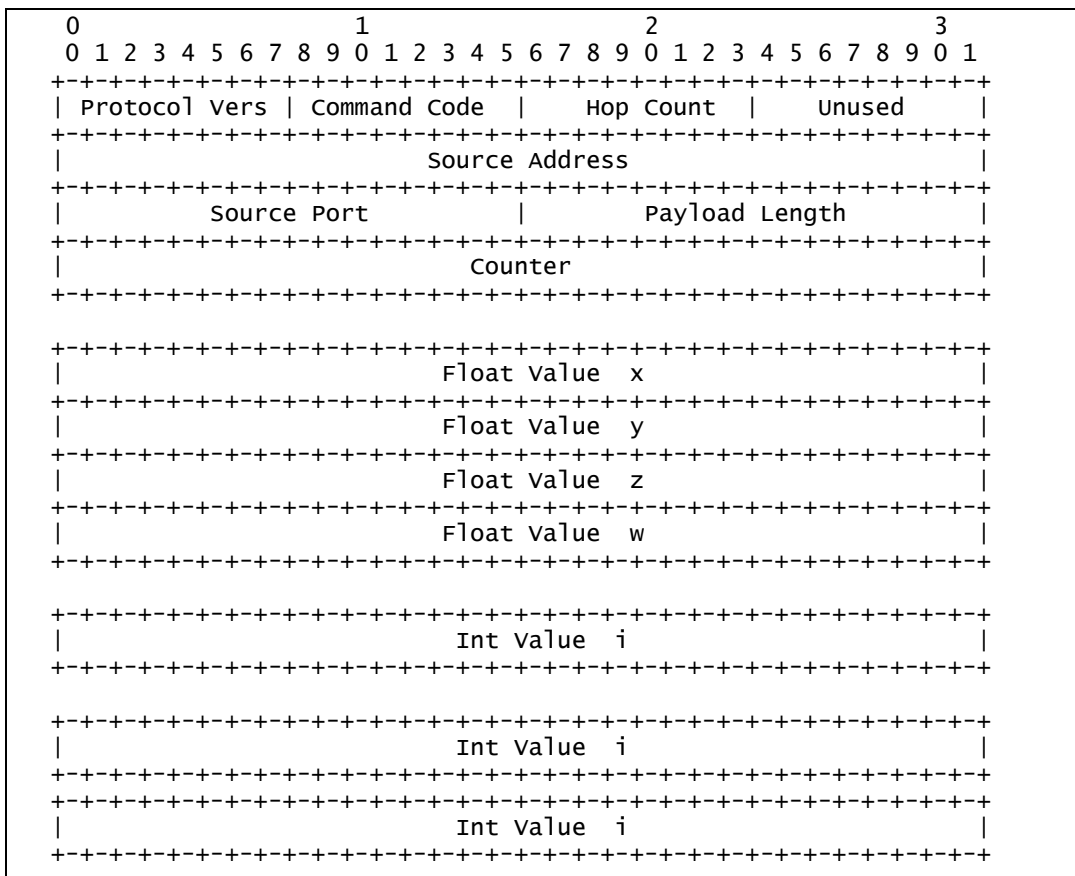


Abbildung 52. CSCW-Protokollelemente. Die Bedeutung der angegebenen Werte – z. B. x, y, z, w – hängt vom jeweiligen „Command Code“ ab. Die Anzahl dieser optionalen Daten ist im „Counter“ codiert.

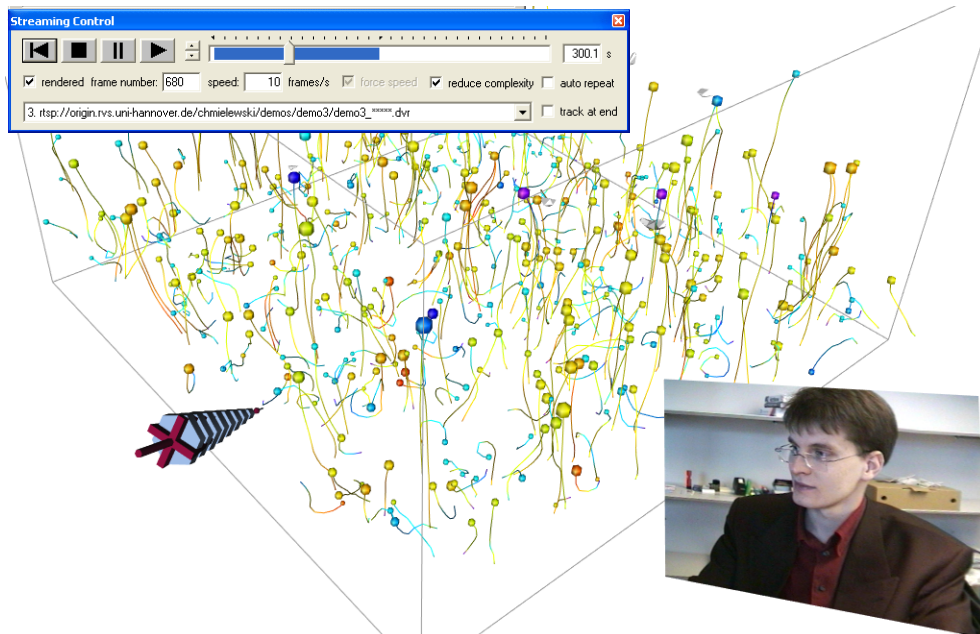


Abbildung 53. Visualisierung turbulenter atmosphärischer Phänomene mittels 3D-Animation von „Illuminated Stream Lines“ [22], ergänzt durch Elemente für „Computer-Supported Collaborative Working“: 3D-Tele-Pointer, Virtual Video Wall (aus [11]).

3.2.6 Software-Dokumentation und -Distribution über WWW

Auf dem RVS-Webserver wurde eine Projekt-Homepage erstellt:

- <http://www.rvs.uni-hannover.de/projekte/tele-immersion/>

Darin ist die Dokumentation und Distribution der entwickelten Software abgelegt. Ein Download der Viewer-Software (3D-Streaming-Client: Plugin für WWW-Browser) befindet sich in:

- <http://www.rvs.uni-hannover.de/projekte/tele-immersion/download/>

Zu internen Zwecken wurde auch ein „Intranet“-Bereich – SSL-geschützt – aufgebaut, zu dem nur Projektteilnehmer einen Zugang bekommen:

- <https://www.rvs.uni-hannover.de/projekte/tele-immersion/intern/>

Für die libDVRP wurde eine englischsprachige Dokumentation erstellt. Sie umfasst sowohl die vollständige Schnittstellendefinition aus der Sicht des Anwendungsprogrammierers als auch eine Einführungslektion, in der die grundlegende Verwendung der libDVRP anhand von leichtverständlichen Programmbeispielen erläutert wird. Aus Gründen der Benutzerfreundlichkeit basiert die Dokumentation auf dem verbreiteten DocBook-System [20], das die Erzeugung unterschiedlicher Ausgabeformate (HTML, PDF, Word, etc.) gestattet. Für die libDVRP-Dokumentation wurde das DocBook-System erstmalig am RRZN bzw. ZIB eingesetzt. Die in einer Evaluationsphase gewonnenen positiven Erfahrungen wurden auf einer Tagung der HLRN-Fachberater [5] vorgestellt und haben maßgeblich die Entscheidung, die komplette HLRN-Dokumentation auf der Grundlage von DocBook zu verfassen, beeinflusst.

Parallel zu der Weiterentwicklung der libDVRP wurden die Dokumentationsseiten gepflegt:

- <http://www.rvs.uni-hannover.de/projekte/tele-immersion/documentation/libdvrp.html>
- <http://www.rvs.uni-hannover.de/projekte/tele-immersion/documentation/dvrp.pdf>

Diese Dokumente erläutern in aktueller Fassung die Fortran-90/95-Schnittstelle im Detail. Die Benutzung wird durch eine Vielzahl an Demonstrationsprogrammen erleichtert, die jeweils spezifische Aspekte der libDVRP zum Inhalt haben. Der Einstieg in die Programmierung mit der libDVRP wird durch die Verfügbarkeit der Quelltexte sowie der jeweiligen DVR Ergebnissequenzen erleichtert.

Zur libDVRP wurde eine englischsprachige Homepage entworfen. Sie ist von den Projektseiten aus zu erreichen und bietet Dokumentation und Viewer-Software zum Lesen bzw. Herunterladen an.

3.3 Teilprojekt 3 – Informationsextraktion und Datenvorverarbeitung

3.3.1 Architektur

Bei der Planung der Bibliothek zur parallelen Datenextraktion und Visualisierung wurden existierende Ansätze (AVS [2], pV3 [16], Pooma [15]) begutachtet, und Erfahrungen aus vorherigen Projekten wurden in diesen Prozess mit einbezogen.

Die letztendlich definierte Architektur versucht einerseits allzu komplexe Konstruktionen und Datenlayouts zu vermeiden (diese erschweren Handhabung und Wartbarkeit des Systems erheblich). Andererseits wird auf eine extrem einfache Erweiterbarkeit großen Wert gelegt. Eine weitere Randbedingung ist, dass der Runtime-Overhead der Bibliothek verhältnismäßig klein ist und möglichst nur linear zur Anzahl der genutzten PEs wächst.

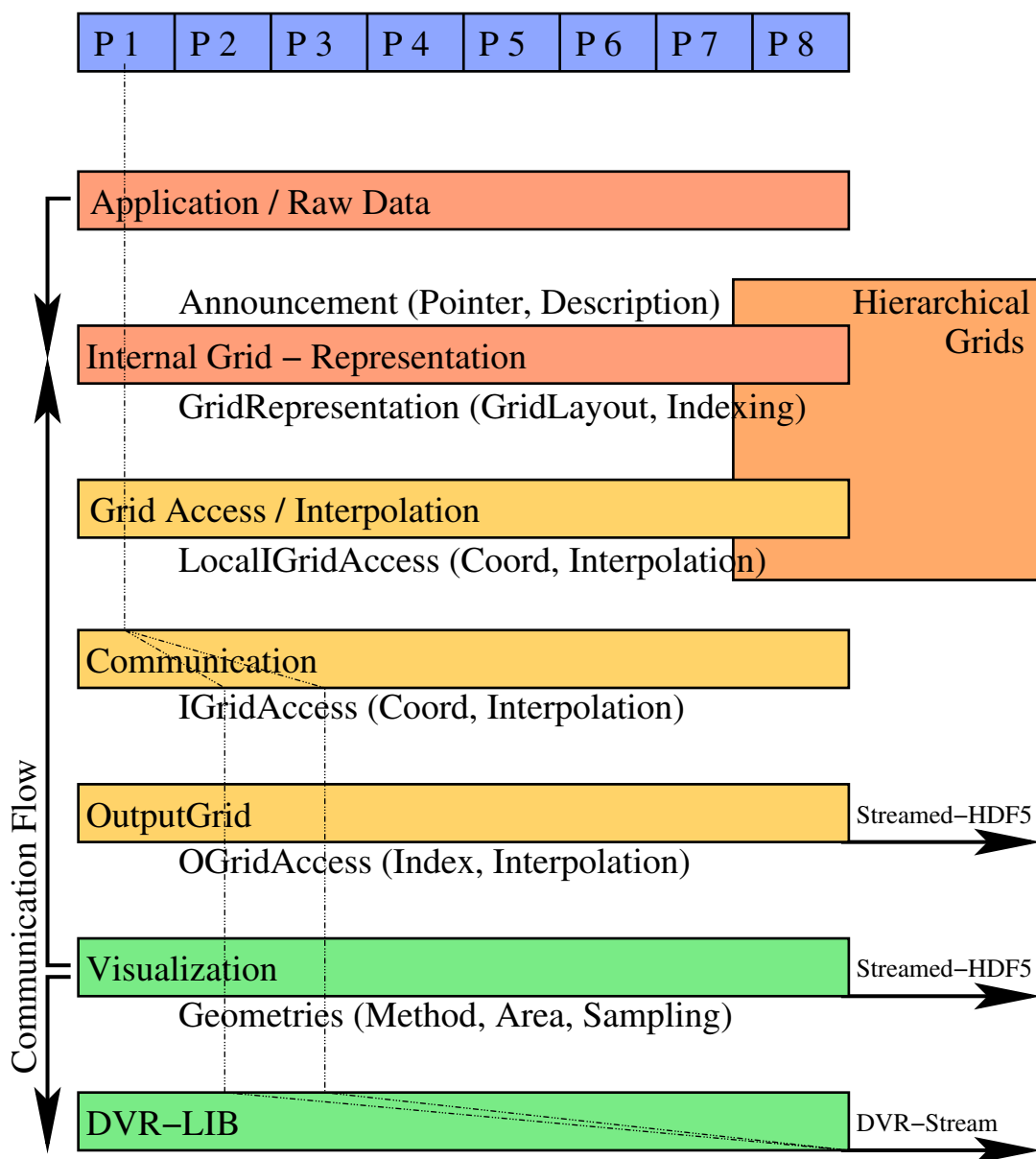


Abbildung 54. Schema der im Projekt entwickelten Architektur zur parallelen Extraktion und Visualisierung von Simulationsdaten.

Im Interesse des einfachen Datenlayouts wurde auf die direkte Einbeziehung von hierarchischen Gittern in das Datenkonzept verzichtet. Hierarchien von Gittern können jedoch über Attribute zu einfachen Gittern nachträglich erstellt werden. Für das Projekt ist dieser Punkt jedoch nicht von Interesse.

Des Weiteren wurde auf ein zu theoretisches Konzept zur Datenbeschreibung (Fiber-Bundles) vorerst verzichtet, da dieses in anderen Projekten Probleme in der Handhabbarkeit zeigt. Statt dessen wurde eine einfache und in der Visualisierung bewährte Trennung von Gitter und Daten gewählt.

Der Kern der Architektur besteht in der Abbildung des simulationsinternen Datengitters auf eine Gitterstruktur, auf welche die verteilte Visualisierungsschicht schnell und, wenn nötig, global zugreifen kann. Im Interesse der Effizienz wurde diese Abbildung über C++-Templates implementiert. Diese Abbildungsschicht beinhaltet ebenfalls die Möglichkeit zur Interpolation der Simulationsdaten an beliebigen Punkten im Datenraum (funktionaler Datenzugriff). Das Design dieser Softwareschicht orientiert sich stark an den Ideen des Pooma-Projektes [15], ohne jedoch diese Bibliothek direkt zu nutzen.

Diese Architektur benötigt im Wesentlichen keinen zusätzlichen Speicherplatz, sondern greift direkt auf die Applikationsdaten zu. Abhängig von den gewählten Visualisierungsmethoden können jedoch auch Kopien ausgewählter Speicherbereiche angelegt werden.

Die Kommunikation der Daten für nicht-lokale Visualisierungsalgorithmen erfolgt auf MPI-Basis unabhängig von der Kommunikationsinfrastruktur der Simulation. Diese muss jedoch die räumliche Verteilung der Daten und deren Speicherlayout beschreiben. Dies erfolgt über ein bereits definiertes C++-API, welches einfach in anderen Sprachen (C, Fortran) abgebildet werden kann.

Die Kommunikation der Bibliothek mit den Visualisierungsclients kann auf mehreren Schichten unabhängig voneinander erfolgen, sowohl über das am ZIB entwickelte F5 bzw. Streaming-HDF5, als auch über das am RRZN entwickelte DVR-Protokoll. Desweiteren können die Parameter der Visualisierungsalgorithmen vom Client beeinflusst werden (Steering). Dieser Prozess wurde bereits in *Abschn. 3.2.2* beschrieben.

3.3.2 Visualisierungs- und Datenverarbeitungstechniken der PaDEV

Verschiedene Algorithmen zur Verarbeitung von Simulationsdaten wurden als integraler Bestandteil in die PaDEV eingebaut. Hierbei wurden insbesondere Techniken berücksichtigt, die das anfallende Datenvolumen schnell und mit nur geringem Overhead verarbeiten können, sowie ihrerseits ein relativ geringes Datenvolumen erzeugen.

Sub-Volumina und Sub-Sampling

Oftmals sind in Simulationen bestimmte Teilbereiche der Gesamtdaten von besonderem Interesse. Die PaDEV ermöglicht es dem Endanwender, solche Sub-Volumina zu definieren und fortan werden nur noch Simulationsdaten aus diesen Bereichen zur Visualisierung herangezogen.

Ähnlich verhält es sich mit der Gitterauflösung. Benötigt eine Simulation für ihre Rechenprozesse eine außerordentlich hohe Auflösung, so fällt bei der Visualisierung oftmals eine große Menge an Geometriedaten an. Während dies selten ein Problem für die Übertragung zum Visualisierungsclient darstellt, so ist es doch sehr wohl ein Problem für die flüssige Darstellung auf dem Client. Wenn der Beitrag eines Geometrieprimativs (Dreieck, Liniensegment, etc.) im Vergleich zur gesamten 3D-Szene sehr klein ist, kann man also von einer zu hohen Geometrieauflösung sprechen. Die PaDEV bietet hier eine einfache Lösung: Bei der Interpretation der Gitterstruktur (und somit bei der Visualisierung) wird ein implizites Sub-Sampling durchgeführt. Es wird also nur jeder n -te Datenpunkt betrachtet und ausgewertet. Dies führt nicht nur zu einer schnelleren Geometrieextraktion, sondern auch zu einem geringeren Aufkommen an Geometriedaten, welches wiederum vom Client schneller verarbeitet werden kann.

Die PaDEV bietet dem Endanwender getrennte Einstellungsmöglichkeiten für die Definition von Sub-Volumina und Sub-Sampling. Somit kann man einen beliebigen Datenausschnitt in verschiedenen Auflösungsstufen betrachten.

Statistik

In der PaDEV wurden auch einfache Methoden zur statistischen Auswertung von Simulationsdaten implementiert. Hierbei handelt es sich um die Berechnung von Mittelwerten, Standardabweichungen sowie Minima und Maxima. Die Reduktion des Datenaufkommens ist bei diesen Techniken enorm. Jedoch ist die Aussagekraft dieser Werte meist schwächer als die einer 3D-Szene.

Marching Cubes

Zur Darstellung von dreidimensionalen Skalarfeldern werden häufig Visualisierungstechniken verwendet, die zu einem vorgegebenen Schwellwert eine Isofläche erzeugen. Der Marching-Cubes-Algorithmus [12] ist eine hierfür oftmals verwendete, sehr schnelle Technik.

In die PaDEV wurden zwei derartige Marching-Cubes-Algorithmen integriert, die sich in Laufzeit und Geometrieausgabe unterscheiden.

Zum einen handelt es sich dabei um das aus [12] bekannte Verfahren, welches an dieser Stelle nicht näher erläutert werden soll.

Zum anderen wurde ein erweiterter Marching-Cubes-Algorithmus implementiert, welcher eine topologisch exakte Triangulierung liefert. Dieser basiert auf einem Ansatz von Holger Theisel, wie er in [18] beschrieben wird. Zusätzlich zum allgemein bekannten Marching-Cubes-Verfahren [12] werden hierbei so genannte „innere Punkte“ einer Datenzelle ausgewertet und zu einem „inneren Ring“ zusammengefasst (*Abbildung 55b*). Nun wird die Konnektivität zwischen dem inneren Ring und jedem äußeren Ring (welche nach [12] extrahiert werden) geprüft und eine entsprechende Triangulierung durchgeführt. Im Gegensatz zum allgemein bekannten Marching-Cubes-Verfahren liefert diese Technik in allen Fällen eine topologisch exakte Triangulierung der Isoflächenrepräsentation.

Dieses Verfahren ist aufgrund der Berechnung und Auswertung des inneren Rings im Laufzeitverhalten langsamer als ein Algorithmus nach [12]. *Abbildung 57* zeigt dies deutlich.

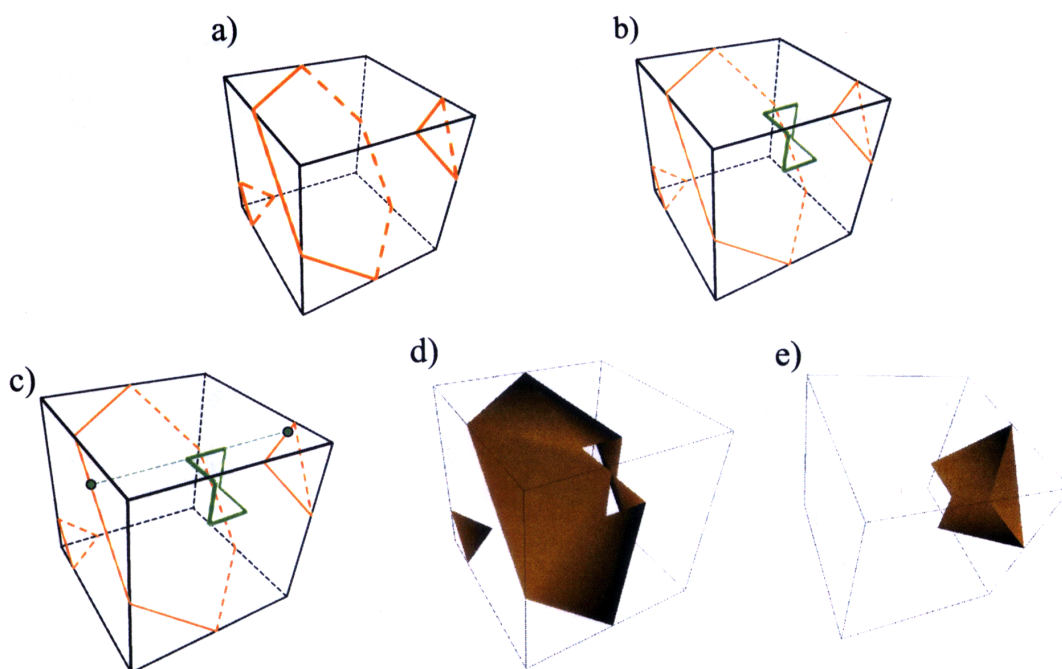


Abbildung 55. Illustration zu einem topologisch exakten Marching-Cubes-Algorithmus (aus [18]);

- a) zeigt die nach [12] extrahierten äußeren Ringe;
- b) Berechnung des inneren Ringes;
- c) Überprüfung der Konnektivität zwischen innerem und äußeren Ringen durch Schnittpunktberechnung zwischen den Linien des inneren Ringes und den Flächen der Datenzelle – in diesem Beispiel ist der innere Ring mit 2 äußeren Ringen verbunden;
- d), e) Triangulierung der Gebiete zwischen innerem und äußeren Ringen.

Stromlinien

Zur Visualisierung von dreidimensionalen Vektorfeldern wurde eine Technik zur Extraktion von Stromlinien implementiert. Hierzu wurde ein Runge-Kutta-Verfahren der Ordnung 8 (5,3) verwendet, welches eine besonders hohe Genauigkeit bietet. Das Verfahren wurde in [8] beschrieben.

Für eine aussagekräftige Stromlinienvisualisierung ist die Wahl der Startpunkte entscheidend. Deswegen wurde ein von Tino Weinkauf entwickeltes Verfahren zur automatischen Wahl der Startpunkte in der Nähe turbulenter Regionen implementiert (beschrieben in [21]).

Abbildung 56 zeigt den deutlichen Vorteil dieses Verfahrens.

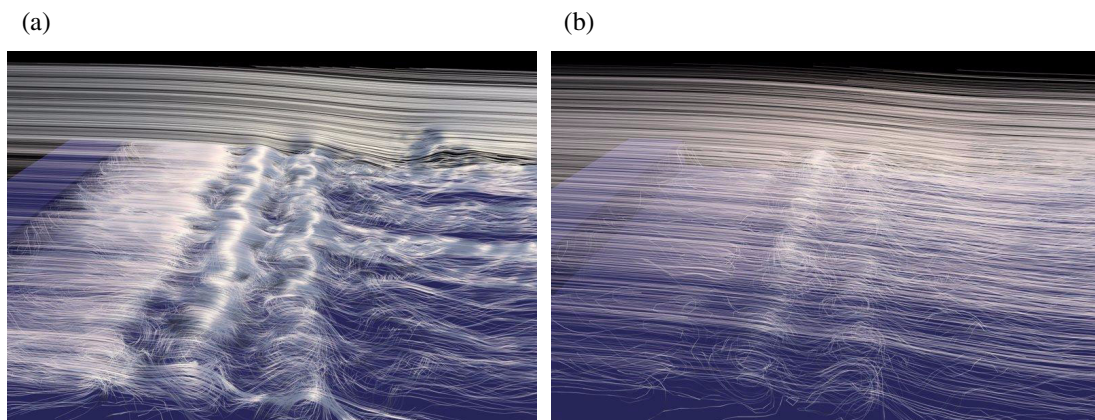


Abbildung 56. Stromlinienvisualisierung einer Stufenströmung. Die Daten wurden freundlicherweise von Bernd R. Noack (Hermann-Föttinger-Institut für Strömungsmechanik, TU Berlin) zur Verfügung gestellt.
 (a) Die Startpunkte für die Stromlinien wurden in turbulenten Bereichen angesiedelt.
 (b) Die Startpunkte für die Stromlinien wurden homogen verteilt.

3.3.3 Evaluierung

Um die Anpassbarkeit der PaDEV zu gewährleisten, war die Einführung logischer Abstraktionsschichten erforderlich. Mit Hilfe dieser Abstraktionsschichten ist es nunmehr möglich, die meisten Visualisierungsalgorithmen so zu implementieren, dass sie ohne Kenntnis der eigentlichen Struktur des Datengitters auf die Daten zugreifen können. Dies ermöglicht eine hohe Wiederverwertbarkeit der Visualisierungsalgorithmen. Allerdings kann dies auch zu einem gewissen Performanceverlust führen. *Abbildung 58* zeigt dies deutlich. Dennoch ist eine andere Art der Implementierung nicht akzeptabel, da sonst jeder Visualisierungsalgorithmus an jede mögliche Gitterstruktur angepasst werden müsste.

Trotz der natürlichen Komplexität einer Bibliothek wie der PaDEV, die Algorithmen auf verschiedensten Gitterstrukturen zum Laufen bringen muss, ist es gelungen, die Implementierung übersichtlich und verständlich zu halten. Es kommen zwar C++-Templates mit mehreren Templateparametern zum Einsatz, jedoch sind diese alle ausreichend dokumentiert. Um die Wartbarkeit der PaDEV zu erhöhen, wurden spezielle Kommentare im Quelltext eingeführt, die mit Hilfe von Quelltextdokumentationsprogrammen wie z. B. doxygen [5] ausgewertet und zu einer Quelltextdokumentation zusammengefasst werden können.

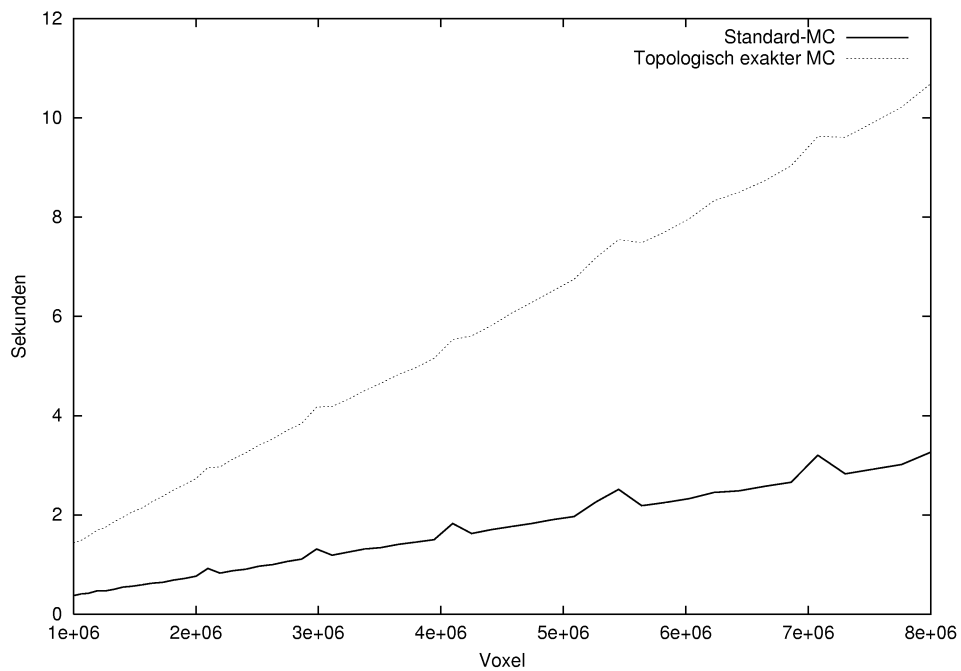


Abbildung 57. Performance der PaDEV: Laufzeitvergleich zwischen dem alten und dem neuen MC-Algorithmus. Beide liefen mit Abstraktionsschicht. Es zeigt sich, dass der neue Algorithmus 3–4-mal schneller als der alte ist. Man beachte, dass es sich um 1 Million bis 8 Millionen Zellen handelt, die untersucht werden müssen. [Dual-XEON 1.7 GHz, 1 GB RAM, Windows XP Pro, INTEL C++ Compiler 5.0]

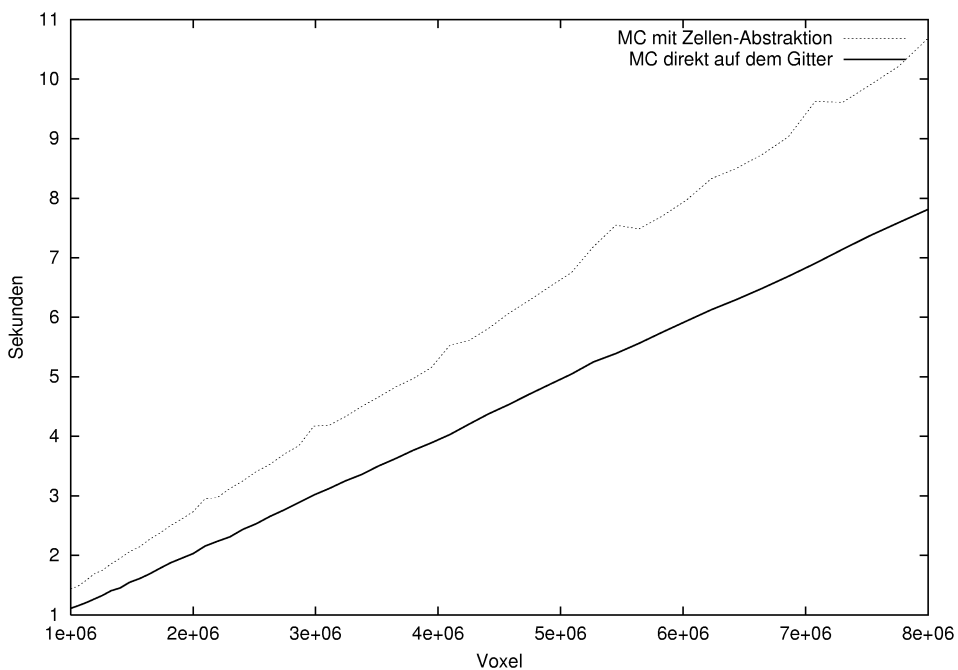


Abbildung 58. Performance der PaDEV: Laufzeitvergleich eines MC-Algorithmus, der sowohl mit als auch ohne Abstraktionsschicht implementiert wurde. Die Implementierung, welche direkt auf dem Datengitter arbeitet, ist schneller. [Dual-XEON 1.7 GHz, 1 GB RAM, Windows XP Pro, INTEL C++ Compiler 5.0]

3.4 Teilprojekt 4 – Simulation atmosphärischer Konvektionsströmungen

3.4.1 Bereitstellung der IMUK-Anwendungsumgebung für das Präsentationsszenario

Bei der Unterstützung der Anwendung des 3D-Streamingverfahrens zur Visualisierung von Ergebnissen parallelisierter Simulationsrechnungen zur „Atmosphärischen Konvektion“ wurde – neben der Weiterentwicklung und Verteilung bzw. Installation der 3D-Viewer-Software (siehe auch Abschn. 3.2.3) – insbesondere die Generierung von 3D-Szenen im DVR-Format effizienter und universeller nutzbar gemacht. Es wurde in den folgenden Schritten vorgegangen:

VRML/DVR-Konverter (C++)

Mittels vorhandener bzw. weiterer prinzipiell verwendbarer Software zur 3D-Visualisierung von Rechenergebnissen – z. B. AVS, VTK – besteht bereits die Möglichkeit, 3D-Szenen im VRML-Dateiformat auszugeben. Zur weiteren Aufbereitung und Nutzung des 3D-Streamingverfahrens kommt daher eine Konvertierung von VRML in das DVR-Format mit den am RRZN entwickelten Konvertern *wrl1toDVR* bzw. *wrl2toDVR* in Betracht. Bei der Analyse der entsprechenden VRML-Dateien hat sich gezeigt, dass häufig Redundanzen bezüglich der Geometrie- und Attributspezifikationen existieren. Die am RRZN entwickelte C++-Bibliothek *OutputDVR*, die in den o. g. Konvertern zur DVR-Ausgabe dient, wurde daher hinsichtlich der automatischen Erkennung und Eliminierung redundanter Attribute (z. B. Normalenvektoren, Farbspezifikationen) und Polygon-Geometrien (z. B. Unterstützung optimierter Topologien) verfeinert. Zum Teil ergibt sich dabei eine Reduktion des DVR-Datenvolumens um den Faktor 2 bis 3, bzw. bei gleicher Datenrate eine entsprechende Erhöhung der Bildrate.

VTK / vtkDVRExporter / OutputDVR (C++)

Die Speicherung von Rohdaten und anschließende Aufbereitung beinhaltet einen relativ großen Overhead, der besonders bei hochauflösenden, instationären Simulationsrechnungen nicht tolerierbar ist. Daher wurde versucht, auf der Basis einer frei verfügbaren Visualisierungsbibliothek (VTK [19]) aus einer Simulationsrechnung heraus 3D-Szenen im DVR-Format direkt zu generieren. Dazu wurde die VTK-Bibliothek auf der Cray T3E installiert und Wrapper-Routinen in C entwickelt, die eine Schnittstelle zwischen den Programmiersprachen Fortran90 (Simulationsrechnung) und C++ (VTK) bieten. Zur Vermeidung eines weiteren Engpasses bei der Generierung der DVR-Dateien wurde statt einer Konvertierung von VRML- in DVR-Dateien eine Exporter-Klasse in VTK integriert, die eine direkte Ausgabe im DVR-Format gestattet. Dabei wurde von der bereits vorhandenen *OutputDVR*-Klasse Gebrauch gemacht, die für die Nutzung auf einer Cray T3E entsprechend angepasst wurde. Da es sich um eine parallele Simulationsrechnung handelt, werden entsprechend der Anzahl verwendeter Prozessoren mehrere DVR-Dateien erzeugt. Diese Teilgeometrien wurden dann jeweils nach einem Barrier-Call von einem einzelnen Prozessor zusammengefügt. Letzteres geschah mit einem Unix-Kommandozeilenbefehl: `cat *.dvr > all.dvr`. Diese Vorgehensweise war zwar im Prinzip anwendbar, erwies sich jedoch als nicht genügend effizient [14].

DVRP-Library / 3D-Streaming-Client (C)

Die in den vorangegangenen Versuchen gewonnenen Erfahrungen führten zur Entwicklung einer eigenständigen Bibliothek (*libDVRP*), die aufgrund bestimmter auf der Cray T3E mit C++ beobachteter Probleme in C – direkt von Fortran90 aufrufbar – implementiert wurde und ohne VTK bzw. *OutputDVR* auskommt. Die benötigten Visualisierungsfunktionen (Isosurface, Orthogonal Slicer, Particles) wurden neu erstellt. Das Einsammeln der Teilgeometrien wird direkt im Hauptspeicher durchgeführt. Diese auf MPI-Send- und -Receive-Befehlen basierenden Vorgänge sowie auch die Erzeugung von DVR-Daten – wahlweise in lokalen Dateien, auf einem entfernten FTP-Server (mittels *libftp*) oder über RTSP/DVRP auf einem entfernten 3D-Streamingserver (in einem neu entwickelten RECORDING-Betriebsmodus) – wurden in entsprechenden *libDVRP*-Routinen gekapselt. Obwohl in der bisherigen Netzkonfiguration zwischen Cray T3E (Parallelrechner) und SGI Origin200 (3D-Streamingserver) nur über 100-Mbit-Ethernet kommuniziert wird, stellte sich bereits ein deutlicher Performance-Vorteil dieses Ansatzes heraus, insbesondere bei der Nutzung

der RTSP/DVRP-Variante.

Problematisch ist jedoch weiterhin die Tatsache, dass die Datenspeicherung bzw. der Datentransport zum Streamingserver auf einem einzelnen Prozessor erfolgt, der währenddessen die gesamte parallele Simulationsrechnung blockiert (siehe auch *Abbildung 59*). Daher wurde in libDVRP eine Option integriert, die durch Reservierung eines zusätzlichen Prozessors einen asynchronen Schreibvorgang ermöglicht (DVRP_SPLIT). Bei hoher Prozessoranzahl fällt die Allokierung eines weiteren Prozessors kaum ins Gewicht, der Laufzeitvorteil ist dagegen zum Teil deutlich [13] (siehe auch *Abb. 62, S. 82*).

Erweiterungen des 3D-Streaming-Servers (C)

Zur Unterstützung eines Betriebsmodus RECORDING wurde der Streamingserver entsprechend ergänzt. Damit ist es nun möglich, innerhalb einer Verbindung eine Sequenz von mehreren DVR-Dateien als Datenstrom (DVRP) zu übertragen und zu speichern. Die Erstellung von DVRS- und HTML-Dateien, die zum Abruf von 3D-Animationen (Betriebsmodus PLAYING) erforderlich sind, wurde inzwischen automatisiert. Bei Beendigung einer RECORDING-Session erzeugt der 3D-Streamingserver diese Metadaten automatisch.

Grafische Darstellung der parallelen Simulation und Visualisierung [13]

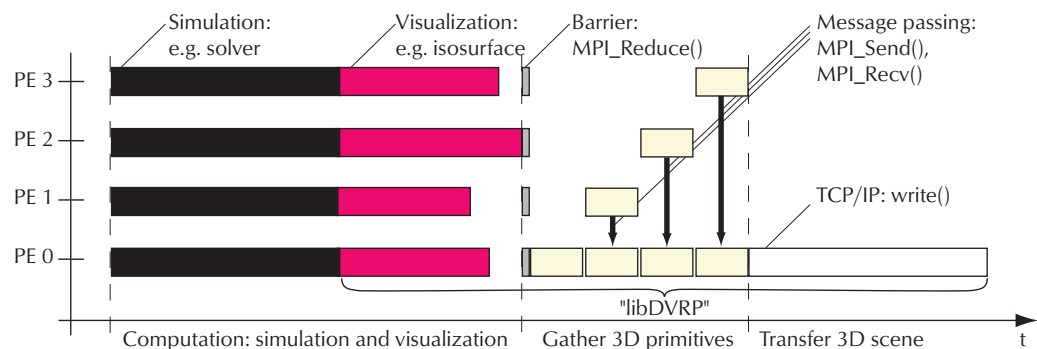


Abbildung 59. Prinzipieller Ablauf der eng gekoppelten Simulation und Visualisierung mit libDVRP.

Hier erfolgt der Datentransport zum Streamingserver auf PE 0, dabei blockieren die PEs 1-3.

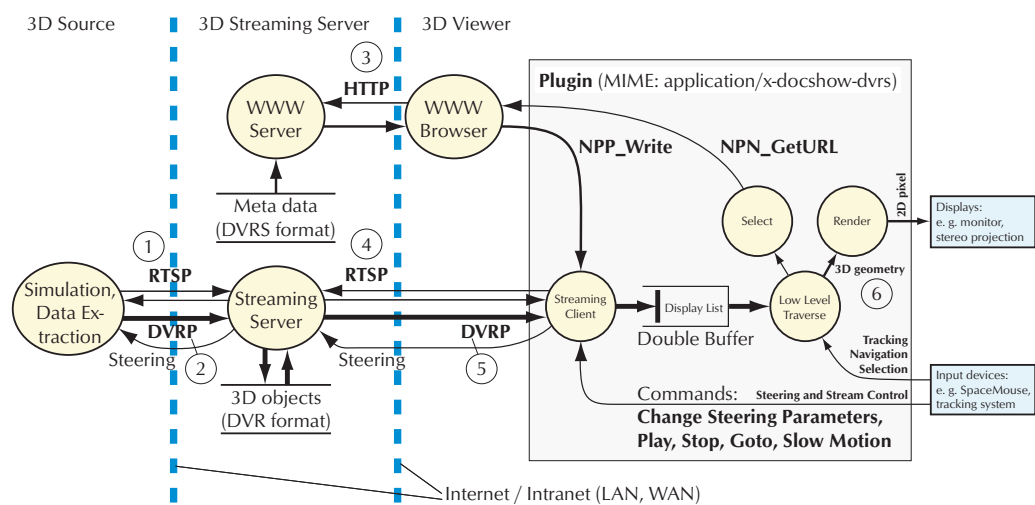


Abbildung 60. 3D-Streamingsystem zur verteilten Simulation und Visualisierung in einer Virtual-Reality-Umgebung.

Effizienz-Analyse – Datentransfer in libDVRP: blockierend vs. asynchron (Pipelining-Option)*Verwendete Symbole*

- T_S : Zeitaufwand zur Ausführung eines sequentiell implementierten Simulations- und Visualisierungsschritts.
- T_0 : Zeitaufwand zum Transport der Ergebnisse eines Simulations- und Visualisierungsschritts zum Streamingserver (hängt von Datenvolumen und Datenrate ab).
- k : Konstante zur Charakterisierung der Anwendung. $k = \frac{T_S}{T_0}$ (1)

N : Prozessor-Anzahl.

$T_P(N)$: Zykluszeit der parallelisierten Simulations-, Visualisierungs- und Transfer-Schritte.

$S_P(N)$: Speedup bei der Parallelisierung. $S_P(N) = \frac{T_S + T_0}{T_P(N)}$ (2)

$E_P(N)$: Effizienz der Parallelisierung. $E_P(N) = \frac{S_P(N)}{N}$ (3)

Vereinfachende Annahme

Simulation und Visualisierung skalieren linear für beliebige N .

D. h. ohne den Transport von Ergebnissen zum Streamingserver $T_0 = 0$ gelte:

$$T_P(N) = \frac{T_S}{N} \Rightarrow S_P(N) = N \Rightarrow E_P(N) = 1. \quad (4)$$

*Zwei Parallelisierungsstrategien bezüglich des Datentransfers**1. Blockierender Datentransport*

Der Transfer wird auf einem Prozessor durchgeführt, dabei blockieren alle anderen Prozessoren. Es gilt:

$$\text{Zykluszeit: } T_P(N) = \frac{T_S}{N} + T_0. \quad (5)$$

$$\text{Speedup: } S_P(N) = \frac{T_S + T_0}{\frac{T_S}{N} + T_0} \Rightarrow S_P(N) = \frac{k+1}{\frac{k}{N} + 1}. \quad (6)$$

$$\text{Effizienz: } E_P(N) = \frac{T_S + T_0}{T_S + N \cdot T_0} \Rightarrow E_P(N) = \frac{k+1}{k+N}. \quad (7)$$

2. Asynchroner Datentransport (Pipelining)

Der Transfer wird auf einem zusätzlichen Prozessor durchgeführt, und zwar zeitlich überlappt mit dem jeweils nachfolgenden Simulations- und Visualisierungsschritt. Es gilt:

$$\text{Zykluszeit: } T_P(N) = \max(T_0, \frac{T_S}{N-1}) \quad (8)$$

Hier sind zwei Fälle zu unterscheiden:

(a) $T_0 \leq \frac{T_S}{N-1}$ bzw. $k \geq N-1$. In diesem Fall gilt: $T_P(N) = \frac{T_S}{N-1}$. (9)

$$\text{Speedup: } S_P(N) = \frac{(T_S + T_0) \cdot (N-1)}{T_S} \Rightarrow S_P(N) = \frac{(k+1) \cdot (N-1)}{k}. \quad (10)$$

$$\text{Effizienz: } E_p(N) = \frac{(T_S + T_0) \cdot (N - 1)}{N \cdot T_S} \Rightarrow E_p(N) = \left(1 + \frac{1}{k}\right) \cdot \left(1 - \frac{1}{N}\right). \quad (11)$$

$$(b) \quad T_0 > \frac{T_S}{N-1} \text{ bzw. } k < N - 1. \text{ In diesem Fall gilt: } T_p(N) = T_0. \quad (12)$$

$$\text{Speedup: } S_p(N) = \frac{T_S + T_0}{T_0} \Rightarrow S_p(N) = k + 1. \quad (13)$$

$$\text{Effizienz: } E_p(N) = \frac{T_S + T_0}{N \cdot T_0} \Rightarrow E_p(N) = \frac{k + 1}{N}. \quad (14)$$

Verlauf von Speedup und Effizienz in einer Beispielanwendung

In *Abbildung 61* sind Speedup und Effizienz in Abhängigkeit von der Prozessoranzahl für die beiden betrachteten Transfer-Alternativen (ohne / mit Pipelining) mit $k = 32$ grafisch dargestellt. Die Illustration beruht auf den Zusammenhängen gemäß Gleichungen 6, 7, 10, 11, 13 und 14.

Im Vergleich zwischen den jeweiligen beiden Kurven in *Abbildung 61* ist festzustellen, dass ein Vorteil der Pipelining-Strategie erst oberhalb einer bestimmten Prozessoranzahl existiert. Durch Gleichsetzung der Gleichungen 2 und 6 bzw. 3 und 7 ergibt sich als Bedingung für diesen „Break-Even-Point“:

$$N \cdot (N - 1) = k. \quad (15)$$

Für den in *Abbildung 61* dargestellten Fall mit $k = 32$ gilt:

$$N \geq 7. \quad (16)$$

Vorteile der Pipelining-Strategie

1. Bessere Skalierbarkeit der Parallelisierung.

Zum Beispiel sinkt die Effizienz – verglichen mit der blockierenden Version – erst bei ca. doppelt so hoher Prozessoranzahl unter 50 %.

$$- \text{ ohne Pipelining: } E_p(N) \geq 0,5 \Rightarrow N \leq k + 2. \text{ (aus Gleichung 7)} \quad (17)$$

$$- \text{ mit Pipelining: } E_p(N) \geq 0,5 \Rightarrow N \leq 2k + 2. \text{ (aus Gleichung 14)} \quad (18)$$

2. Geringere Burstiness, d. h. Glättung des Datentransports.

Mit $k \leq N - 2$ führt das Pipelining-Verfahren zu nahezu kontinuierlicher Datenübertragung.

3. Geringere Anforderung an die Datenrate für den jeweiligen Datentransport.

Zur Erzielung einer Effizienz von mindestens 50 % – verglichen mit der blockierenden Version – bei gleich großer Prozessoranzahl eine nur ca. halb so hohe Datenrate erforderlich.

$$- \text{ ohne Pipelining: } E_p(N) \geq 0,5 \Rightarrow k \geq N - 2. \quad (19)$$

$$- \text{ mit Pipelining: } E_p(N) \geq 0,5 \Rightarrow k \geq \frac{1}{2}(N - 2) \quad (20)$$

Da T_S in beiden Fällen identisch ist, kann k (gemäß Gleichung 1) bei gleichbleibendem jeweiligen Datenvolumen als nahezu proportional zur Datenrate angenommen werden.

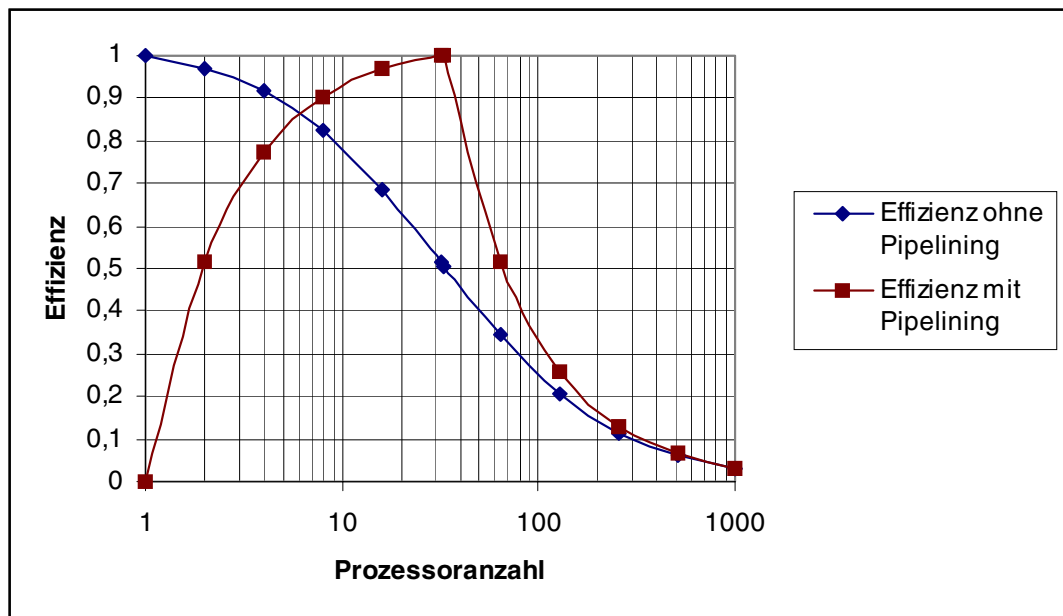
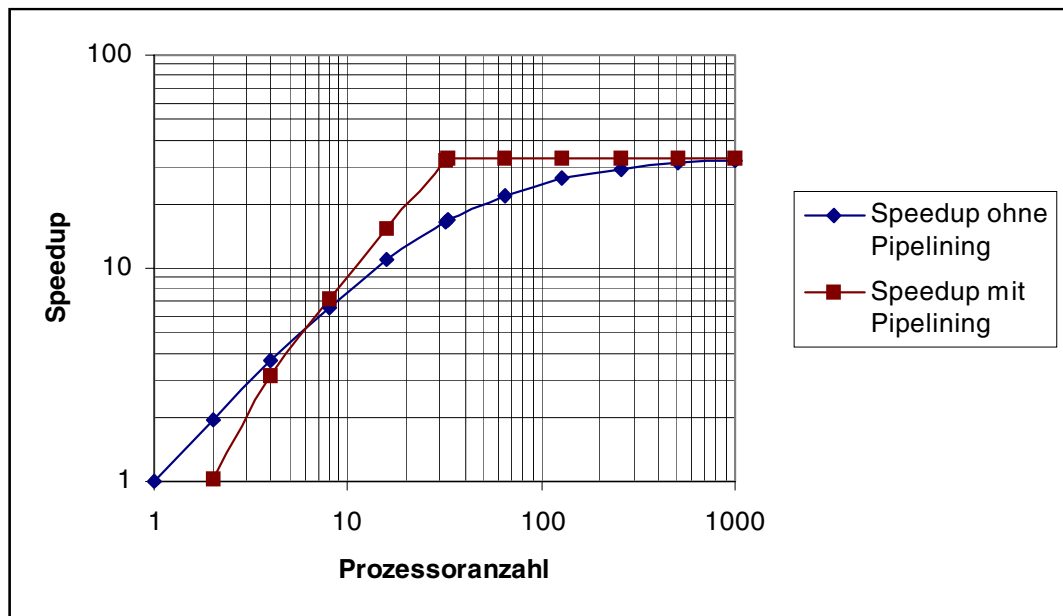


Abbildung 61. Speedup und Effizienz in Abhängigkeit von der Prozessoranzahl für die beiden betrachteten Transfer-Alternativen (ohne / mit Pipelining) in einer Beispielanwendung mit $k = 32$.

Praktische Anwendung der Pipelining-Option (DVRP_Split) [13]

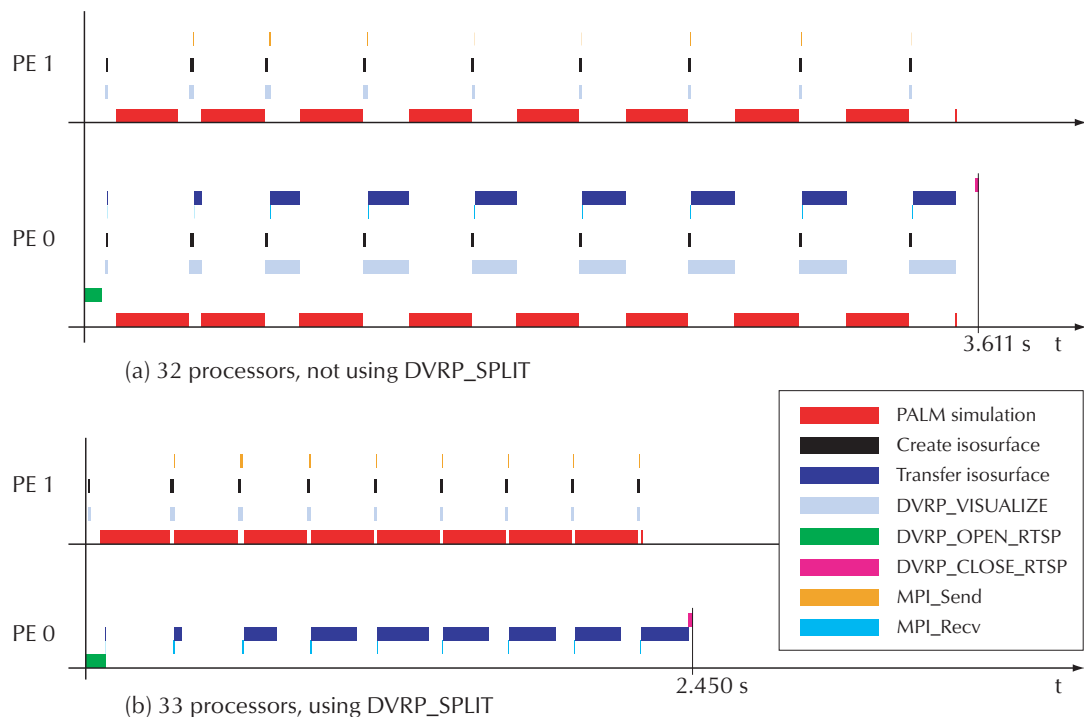


Abbildung 62. Time limes, showing atmospheric simulation (159x31x32 grid points), isosurface visualization and transfer (from Cray T3E to SGI Origin 200, connected via 100 Mbps Ethernet) steps, in a (small) case study on a Cray T3E. Only the first 2 PEs are shown, the other PEs behave similar to PE 1.

- Improvement by using `DVRP_SPLIT`: 32 %.
- 3D data volume (9 DVR files): 3.363.592 bytes.

3.4.2 Bereitstellung der IMUK-Anwendungsumgebung für das Explorationsszenario

Die in *Abschn. 3.2.4* beschriebenen Maßnahmen zur Unterstützung von „Online-Visualisierung“ und „Interactive Steering“ wurden den Anwendungspartnern bereit gestellt. Im Rahmen der Anwendung „Atmosphärische Konvektion“ wurde vom IMUK die Online-Visualisierung – d. h. interaktive 3D-Präsentation entsprechend dem Fortschritt der Simulationsrechnung – genutzt.

Die Funktionalität zur ferngesteuerten Simulationssteuerung, dem Steering, wurde beispielhaft anhand der atmosphärischen Konvektion demonstriert und getestet.

Dazu wurde in einem ersten Schritt die Schnittstelle zu der libDVRP komplett auf das neue Fortran 95 Modul umgestellt. In einem zweiten Schritt wurden die allgemeinen Steering-Routinen in Palm integriert und zugleich typische Simulationsparameter zum Steering eingerichtet. Im Zuge der Anpassungen wurde zudem die Palm-Software auf die SGI Onyx 3000 Plattform portiert. Damit standen zur Evaluation des Steering-Szenarios mit Palm neben der IBM Regatta p690 (HLRN) zwei unterschiedliche Plattformen zur Verfügung.

Die Erprobungsphase hat die praktische Nutzbarkeit unter Beweis gestellt und zudem folgende zusätzliche Neuerungen motiviert (siehe auch *Abschn. 3.2.4*):

- ferngesteuertes Anhalten/Fortsetzen der Simulation,
- logarithmische Skalen zur Parameteränderung.

Es können sowohl Parameter mit Einfluss auf die visuelle Darstellung als auch simulationspezifische Parameter interaktiv gesteuert werden. Beispielsweise hat sich die Möglichkeit, die Position von Schnittebenen während der laufenden Simulation zu kontrollieren, als sehr sinnvoll erwiesen.

Die beispielhafte Implementierung der Steering-Funktionalität wird inzwischen aktiv von den Mitarbeitern des Instituts für Meteorologie und Klimatologie in Simulationsläufen genutzt.

3.4.3 Test- und Produktionsläufe

Zum Studium von Phänomenen der „atmosphärischen Konvektion“ wurden vom IMUK etliche verschiedenartige Szenarien simuliert und visualisiert. Aufgrund der Erfahrungen während der Pilotnutzung haben sich u. a. Ergänzungswünsche im Bereich der Streamline-Visualisierung ergeben. Dazu wurde die libDVRP-Software zur Erzeugung von variabel eingefärbten Liniensegmenten ergänzt. Als Konsequenz musste der 3D-Viewer zur Unterstützung dieser Grafikprimitive angepasst werden. Außerdem wurden Optionen für „Illuminated Lines“-Rendering dieser Liniensegmente, aber auch Transparenz und „Haloed Lines“-Rendering sowie Wahlmöglichkeiten zum alternativen Rendering von Kugeln integriert (Points mit wählbarer Größe, platonische Körper).

Der große Nutzen der im Projektkontext bereit gestellten Infrastruktur und Software-Module für dieses Institut lässt sich auch daraus ablesen, dass inzwischen im IMUK eine zweite 3D-Grafikworkstation beschafft wurde, die über eine separate LWL an den Gigabit-Ethernet-Switch des GE-Pilotnetzes angeschlossen werden soll. Das RRZN stellte die dazu erforderliche Leitung zur Verfügung und beschaffte das notwendige weitere Medienwandler-Paar (Single-Mode/Multi-Mode).

Anwendungsbeispiele

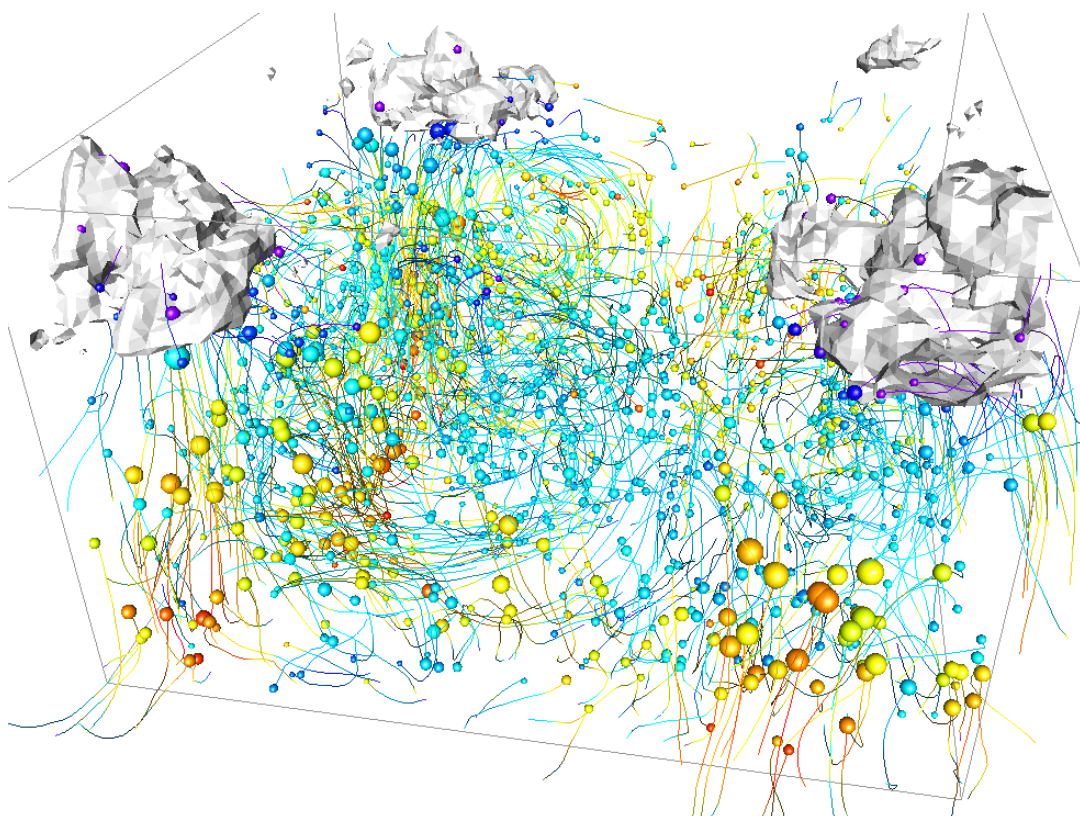


Abbildung 63. Anwendungsbeispiel aus dem Bereich der atmosphärischen Turbulenz (Strömungsdynamik, konvektive Grenzschicht) – Visualisierung instationärer Strömungen mittels Partikelverlagerung und Stromlinien (Einfärbung gemäß Temperatur), Visualisierung der Luftfeuchtigkeit (Wolken) mittels Isosurface.

Die folgenden Abbildungen zeigen Momentanaufnahmen turbulenter Grenzschichtströmungen.

In *Abbildung 64* handelt es sich um den Abfluss kalter Luft an einem Hang. Die Strömung fließt von rechts nach links und wird hier durch eine Isooberfläche der Temperatur veranschaulicht, die erkennen lässt, dass die Strömung durch sich brechende Wellen gekennzeichnet ist.

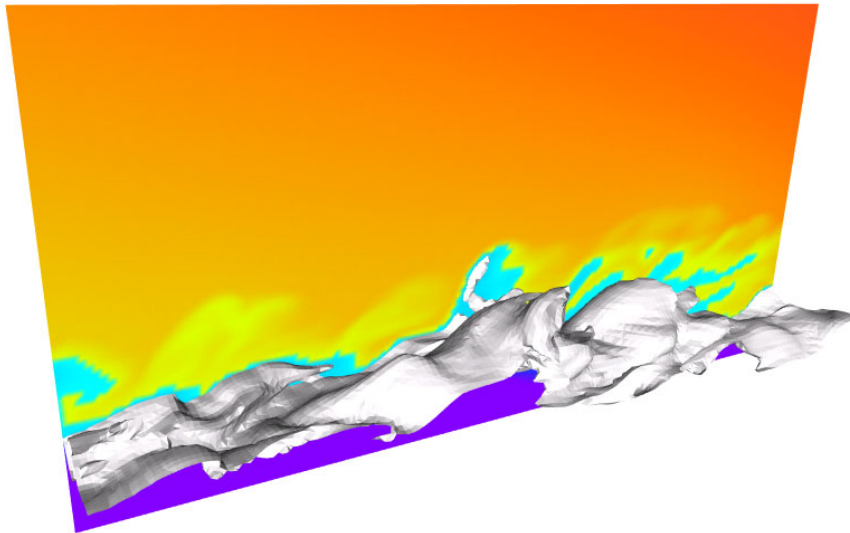


Abbildung 64. Abfluss kalter Luft an einem Hang.

Abbildung 65 zeigt eine so genannte konvektive Grenzschicht, die sich durch Aufheizung der Luft von der Oberfläche her ausbildet. Die Strömung wird in diesem Fall durch passiv transportierte Partikel und ihre Trajektorien visualisiert. Durch die zusätzliche Koppelung der Partikeldurchmesser an die jeweilige Aufwindgeschwindigkeit und der Partikelfarbe an die jeweils lokalen Temperaturverhältnisse werden einzelne Strömungsstrukturen besonders hervorgehoben.

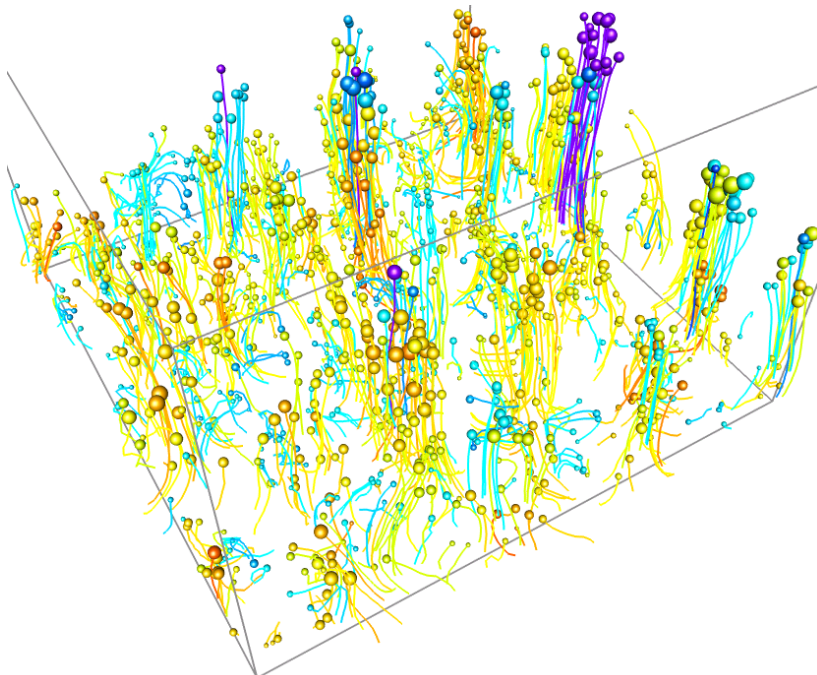


Abbildung 65. Konvektive Grenzschicht.

Die Palette der Anwendungen im Bereich Meteorologie und Klimatologie wurde um die Simulation ozeanischer Konvektionsströmungen ergänzt. *Abbildung 66* visualisiert gleichzeitig Isooberflächen sowie Strömungsbahnen und wurde anlässlich der CeBIT 2002 auf dem BMBF/DFN-Stand vorgeführt. Im Gegensatz zu früheren Visualisierungen wurden die Resultate direkt mit Hilfe der libDVRP aus der Simulation gewonnen. Isooberflächen können zur Verbesserung der Übersichtlichkeit halbtransparent dargestellt werden.

Die Betrachtung der dynamischen Szene profitiert von der erweiterten Möglichkeit, beim Erstellen einer Szenerie, eine Kamerabewegung, z. B. eine Rotation, vorzugeben.

Die Erfahrung aus bisherigen Produktionsläufen hat gezeigt, dass die resultierende Komplexität der Isooberflächen selbst die Grenzen von Hochleistungs-Hardware ausschöpft. Für zukünftige Anwendungen ist aus diesem Grund für die Zukunft die Implementierung von Algorithmen zur Reduktion der Oberflächenkomplexität vorgesehen.

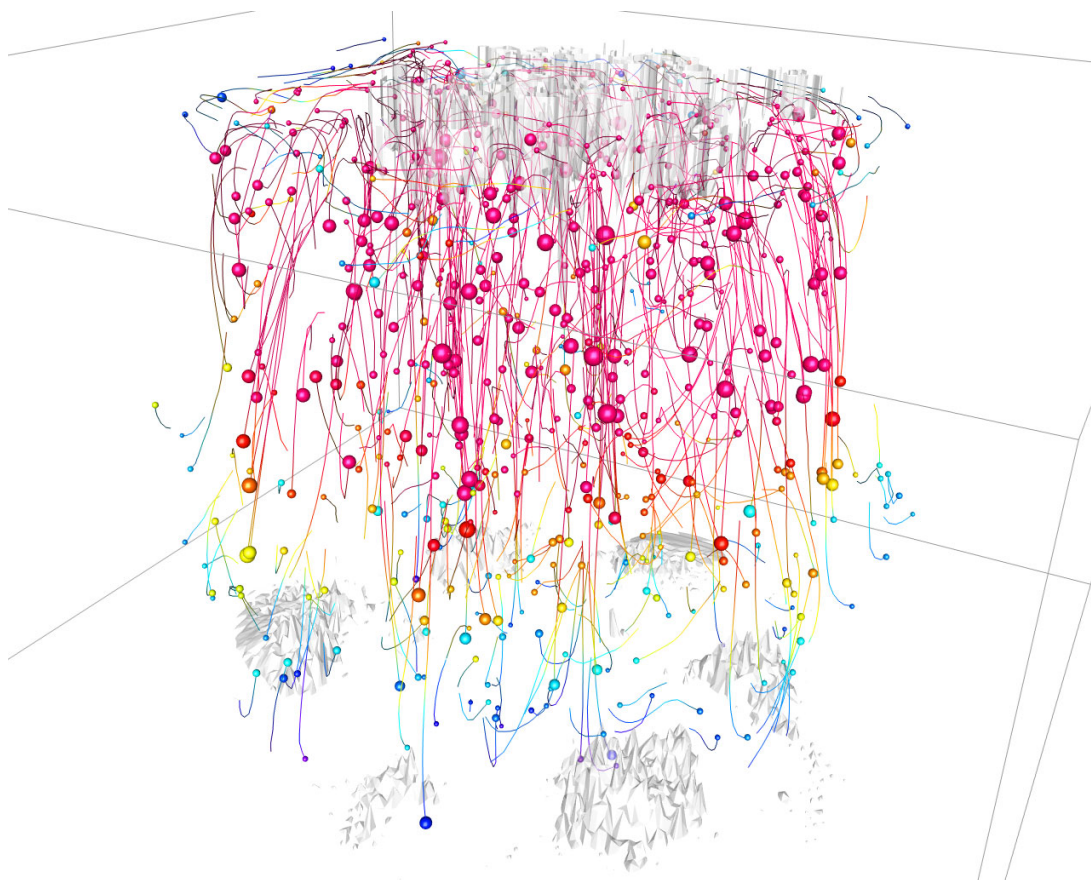


Abbildung 66. Ozeanische Konvektion: Gleichzeitige Darstellung von Isooberflächen (Temperaturfeld) und Bahnlinien (Strömungsfeld).

3.5 Teilprojekt 5 – Molekulardynamische Simulation

3.5.1 Postprocessing binärer Ergebnisrohdaten

Bei der Unterstützung der Anwendung des 3D-Streamingverfahrens zur Visualisierung von Ergebnissen parallelisierter Simulationsrechnungen zur „Molekulardynamik“ wurde – neben der Weiterentwicklung und Verteilung bzw. Installation der 3D-Viewer-Software (siehe auch *Abschn. 3.2.3*) – insbesondere die Generierung von 3D-Szenen im DVR-Format unterstützt.

Ein erster Ansatz bestand in der Speicherung von binären Rohdaten und anschließenden Aufbereitung in 3D-Szenen. Die Aufbereitung wurde in zwei Schritten durchgeführt:

- Erzeugung von Molekülsymbolen im VRML-Format. Je nach Anwendung wurden eingefärbte Kugeln, Zylinder und Kegel kombiniert, um Ort, Größe und Orientierung der simulierten Atome bzw. Moleküle abstrakt darzustellen.
- Konvertierung von VRML in DVR (siehe auch *Abschn. 3.4.1*).

3.5.2 Integration der Dipolsymbol-Generierung

Zur Unterstützung einer engeren Kopplung von Simulation und Visualisierung, sowie auch zur Vorbereitung des Explorationsszenarios (Steering-Funktionalität, siehe auch *Abb. 60, S. 78*), wurde die in Teilprojekt 4 (siehe *Abschn. 3.4.1*) entwickelte Bibliothek libDVRP entsprechend ergänzt und in die molekulardynamische Simulationsrechnung (Fortran 90) integriert. Somit ist es nun ebenso wie in der Anwendung „Atmosphärische Konvektion“ auch hier möglich, 3D-Animationen aus einer auf der Cray T3E laufenden Simulation heraus über das Netz auf den 3D-Streamingserver SGI Origin200 zu schreiben und unmittelbar danach zu sichten.

Die libDVRP wurde um Dipol-Partikel erweitert. Gleichzeitig benutzt die Integration der erweiterten Funktionalität aufseiten der molekulardynamischen Simulation bereits das neue DVRP-Fortran 95-Modul als Schnittstelle. Anwendungsbeispiele sind in *Abbildung 71* bis *Abbildung 70* gezeigt. Während des letzten Projekthalbjahres wurde die libDVRP um einen nativen Typ zur Molekül-Visualisierung erweitert. *Abbildung 72* zeigt ein erstes Beispiel aus der molekulardynamischen Simulation von flüssigem Wasser.

3.5.3 Integration von Steuerungsmöglichkeiten

Zur Erprobung der Online-Visualisierungs- und Steering-Funktionalitäten, deren Realisierungen in *Abschn. 3.2.4* beschrieben sind, wurde eine Version des Molekulardynamik-Simulationsprogramms erzeugt, in der die libDVRP-Funktionen *DVRP_STEERING_INIT* und *DVRP_STEERING_UPDATE* benutzt wurden. Dabei wurden mehrere Einflussmöglichkeiten getestet, z. B.: Druckwerte, Temperaturwerte sowie die Auswahl der Nummer eines gezielt einzufärbenden Moleküls.

Es stellte sich heraus, dass unter bestimmten Umständen Parameteränderungen in der Simulationsrechnung nicht durchzugreifen schienen. Dieses Verhalten wurde durch Compileroptimierungen verursacht, da die Werteänderungen in *DVRP_STEERING_UPDATE* so weit gekapselt sind, dass der Compiler mit derartigen Variablenmanipulationen nicht rechnen kann. Mögliche Auswege waren: Abschaltung der Optimierungsoptionen oder Spezifikation der Steeringparameter-Variablen in COMMON-Blöcken.

3.5.4 Integration der DVR-Transportmechanismen

Die in *Abschn. 3.1.2* erläuterten Verfahren zum Transport von 3D-Szenendaten im DVR-Format vom Simulationsrechner zum 3D-Streamingserver wurden in die libDVRP integriert und damit der IFT-Anwendung zugänglich gemacht. Somit werden die molekulardynamischen Simulationsergebnisse in Form von 3D-Objekten unmittelbar aus der Rechnung erzeugt und zum 3D-Streamingserver transportiert. Von da aus sind die resultierenden 3D-Animationen als Online-Visualisierung – d. h. bereits während der Rechnung – oder zu einem späteren Zeitpunkt aus den dort gespeicherten Dateisequenzen abrufbar.

3.5.5 Test- und Produktionsläufe

Die nachfolgenden Beispiele repräsentieren einen Ausschnitt aus den vielfältigen Fragestellungen, die in der molekulardynamischen Simulation untersucht werden.

Lennard-Jones-Fluid

Die Beispiele in *Abbildung 67* zeigen Simulationsszenen für das sphärische Lennard-Jones-Modellfluid, einem einfachen Potenzialmodell für sphärische Moleküle. Dargestellt sind Zustände im Gasgebiet nahe der Taulinie (a), in der Flüssigphase in Tripelpunktsnähe (b) und ein überkritischer Zustand bei doppelter kritischer Dichte und 1,3-facher kritischer Temperatur (c).

Phasengrenzfläche

Die Simulation betrachtet die Phasengrenzflächen zwischen der Flüssigkeit und dem Gas eines Lennard-Jones-Fluids (*Abbildung 68*).

Stockmayer-Fluid

Betrachtet wird ein Stockmayer-Fluid, d. h. ein Lennard-Jones-Potential mit überlagertem Punktdipol (*Abbildung 69*, *Abbildung 70*). Dabei ist die Zielsetzung wiederum die Identifikation von Clustern. Die Darstellung von Dipol-Partikeln wird durch die in *Abschn. 3.5.2* beschriebene Erweiterung der libDVRP ermöglicht.

Cluster-Bildung

Die Simulation betrachtet die Bildung von Clustern im Gasgebiet eines Lennard-Jones-Fluids bei einer Temperatur nahe des Tripelpunkts (*Abbildung 71*). Die Visualisierung hat das Auffinden von Clustern gegenüber einer algorithmischen Analyse vereinfacht.

Molekülsimulation

Simuliert wurde flüssiges Wasser (256 Moleküle) unter Umgebungsbedingungen bei 25 Grad Celsius und einem Druck von 1 bar (*Abbildung 72*). Die Beschreibung der zwischenmolekularen Wechselwirkungen erfolgt mit dem SPC/E Potenzialmodell, die elektrostatischen Wechselwirkungen mit langen Reichweiten wurden mit den rechenzeitintensiven Ewald-Summen berechnet.

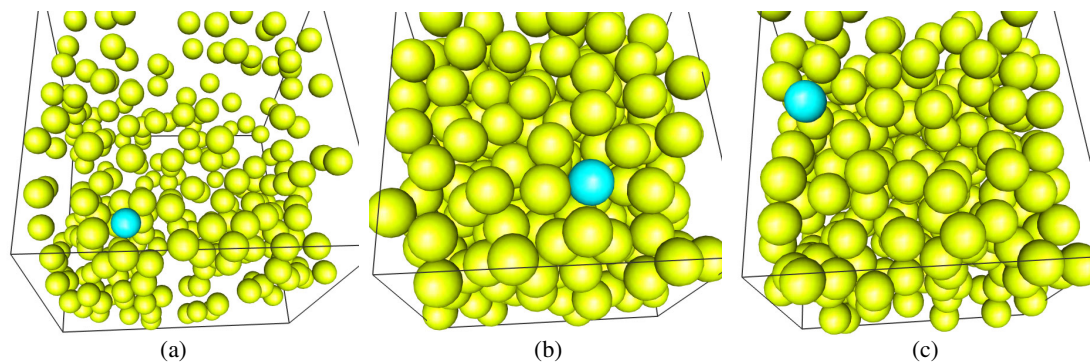


Abbildung 67. Simulation des sphärischen Lennard-Jones-Modellfluids mit verschiedenen thermodynamischen Zuständen:

- (a) im Gasgebiet nahe der Taulinie,
- (b) in der Flüssigphase in Tripelpunktsnähe,
- (c) überkritischer Zustand bei doppelter kritischer Dichte und 1,3-facher kritischer Temperatur.

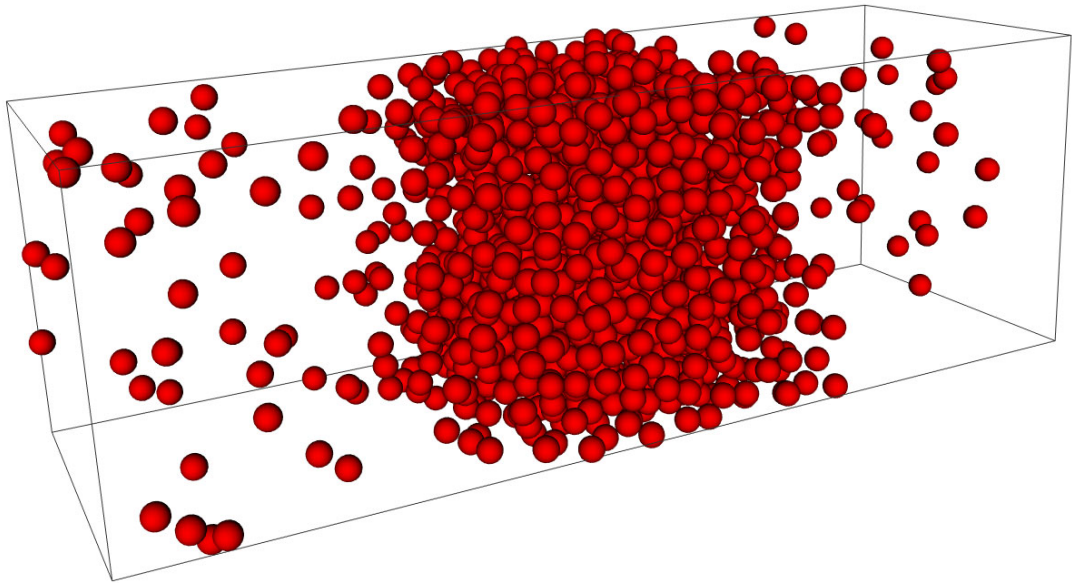


Abbildung 68. Lennard-Jones-Fluid: Phasengrenzflächen im Flüssigkeit-Gas-Gebiet.

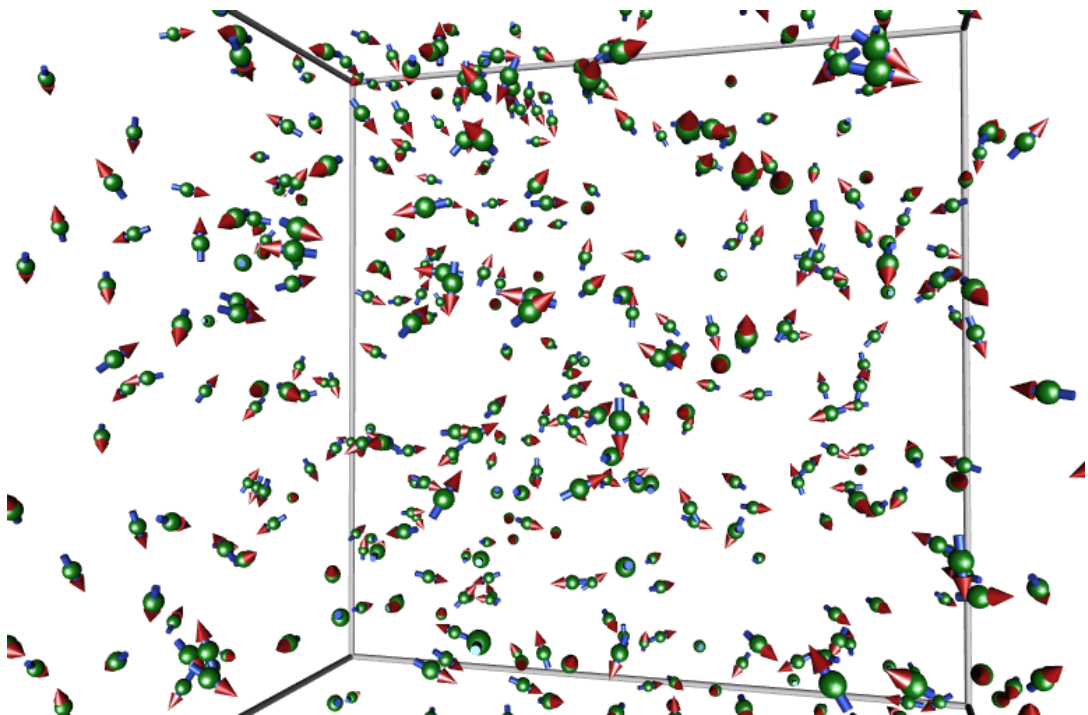


Abbildung 69. Simulation eines gasförmigen Zustands niedriger Dichte für das Stockmayer-Modellfluid, einem Potenzialmodell für polare Stoffe, in dem einem sphärischen Lennard-Jones-Potenzial ein Punktdipol überlagert ist.

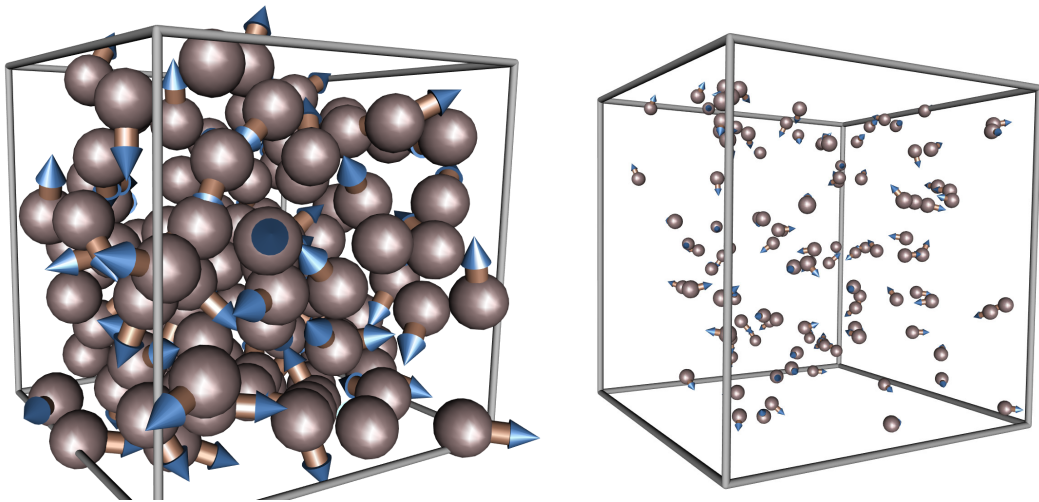


Abbildung 70. Stockmayer-Fluid: Erprobung der Dipol-Partikel im Anwendungsbeispiel der molekulardynamischen Simulation für die Flüssigkeit (linke Seite) bzw. im Gasgebiet (rechte Seite).

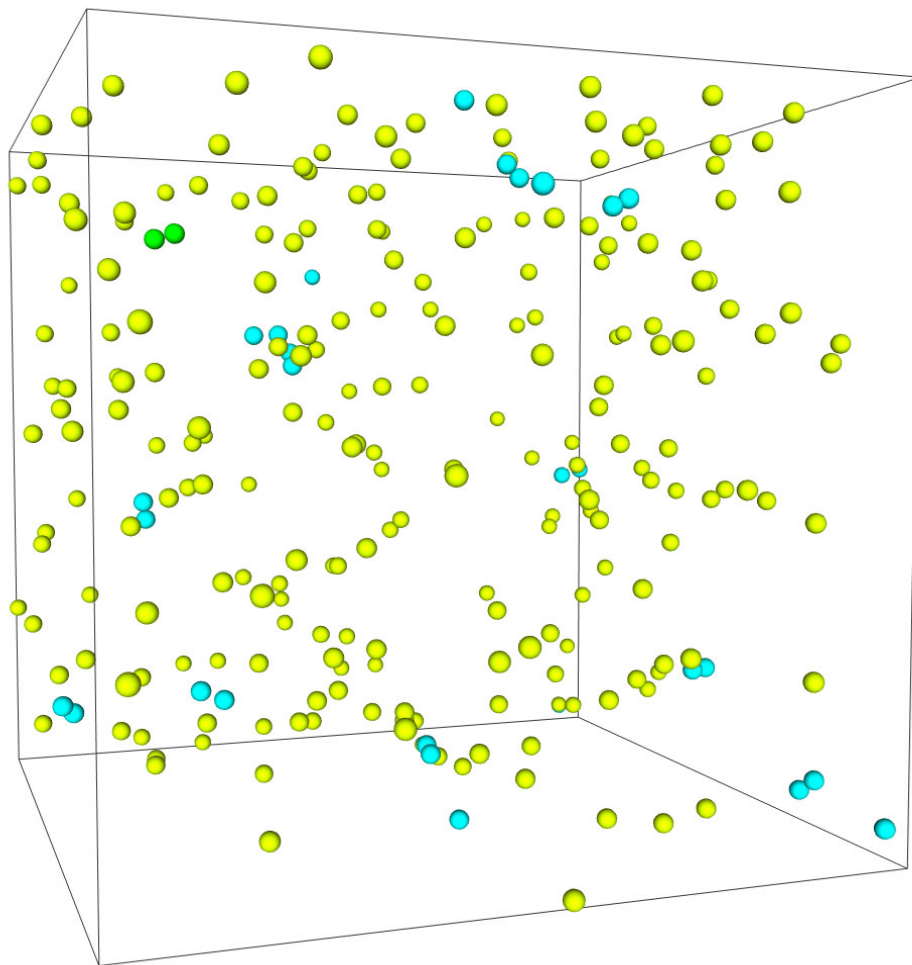


Abbildung 71. Lennard-Jones-Fluid: Clusterbildung im Gasgebiet:
gelb – ungebundene Partikel, blau – stabile Cluster, grün – metastabile Cluster.

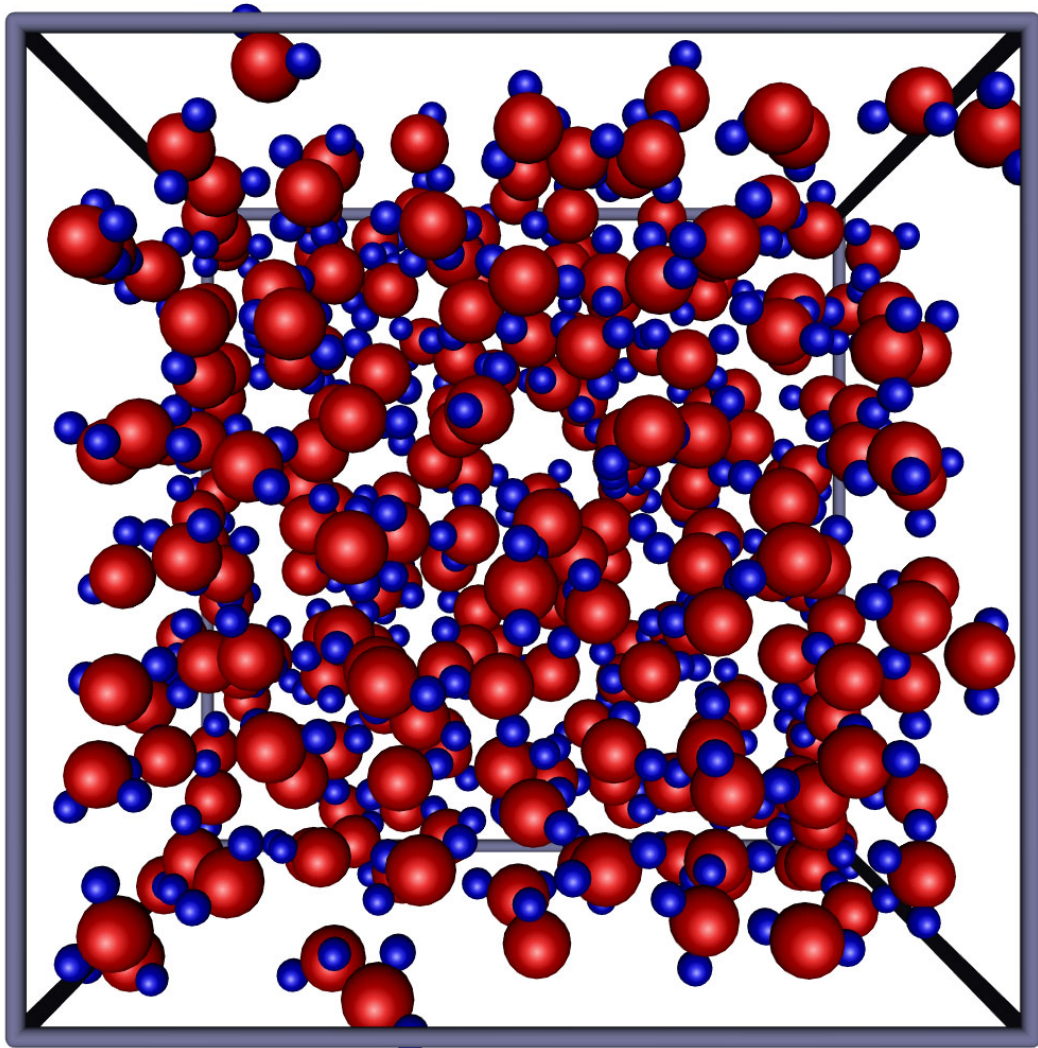


Abbildung 72. Molekulardynamische Simulation von flüssigem Wasser gemäß dem SPC/E-Potenzial-Modell.

3.6 Teilprojekt 6 – Simulation von Strömung und Sedimentation in einem Tidehafen

Der Projektstart wurde vom DFN mit der Auflage verbunden, über die ursprünglich bereits geplanten Anwendungen hinaus (Teilprojekt 4 – Simulation atmosphärischer Konvektionsströmungen / Teilprojekt 5 – Molekulardynamische Simulation) weitere Anwendungen zu finden. Ein weiteres konkretes Anwendungsgebiet für die Nutzung der in diesem Projekt entwickelten Werkzeuge sowie der aufgebauten Infrastrukturen wurde in einer Kooperation mit dem Institut für Strömungsmechanik und Elektronisches Rechnen im Bauwesen¹ (ISEB) an der Universität Hannover identifiziert. Im ISEB besteht intensives Interesse an fortgeschrittenen Visualisierungsmethoden – bis hin zur Virtuellen Realität – im Kontext hochkomplexer, zeitabhängiger Simulationen, die z. B. auf dem HLRN-Komplex gerechnet werden sollen. Die Analyse der dort vorliegenden Anforderungen ergab, dass in dieser Kooperation ähnliche Zielsetzungen verfolgt werden können wie in dem hier bearbeiteten Teleimmersionsprojekt und somit auch die Übertragbarkeit der Ergebnisse gezeigt werden kann.

Während die in den vorangehenden beiden Abschnitten 3.4 und 3.5 beschriebenen Anwendungsprojekte eher dem Bereich der Grundlagenforschung zuzurechnen sind, handelt es sich bei diesem um anwendungsbezogene Forschung und Entwicklung. Sie wird mithilfe der speziellen Simulations-Software *Telemac* durchgeführt, die in der Kooperation direkt an die libDVRP gekoppelt wurde, und analysiert die komplexen Strömungsverhältnisse und Turbulenzen von Hafeneinfahrten. Strömungswirbel verursachen in diesem Bereich Sedimentablagerungen und machen regelmäßig kostspielige Ausbaggermaßnahmen erforderlich. Vorrangiges Ziel ist daher, die Wirbelbildung zu reduzieren, letztlich um die Sedimentablagerung zu minimieren. Im konkreten Anwendungsbeispiel wurde der Einfluss einer so genannten Strömungsumlenkwand untersucht, die im Köhlfleet des Hamburger Hafens platziert wurde. Die Ausführung einer Simulationsrechnung benötigt auf einem aktuellen Arbeitsplatzrechner (PC) etwa dreimal so viel Zeit wie die zu simulierende Zeitspanne. Daher können zurzeit nur relativ kurze Zeiträume (einige Stunden) untersucht werden. Auf dem HLRN-Rechner soll es dagegen künftig möglich werden, Ergebnisse der eigentlich gewünschten Simulation eines kompletten Gezeitenzyklus inklusive Springflut (1 Monat) in wenigen Stunden zu erhalten und mit Hilfe des Visualisierungssystems, das in diesem Projekt entwickelt wurde, währenddessen bereits dreidimensional betrachten zu können. Speziell für dieses Teilprojekt wurde die libDVRP um Möglichkeiten zur Visualisierung von polygonalen Netzen ergänzt.

Die Kooperation mit dem ISEB hat sich als außerordentlich fruchtbar erwiesen, wie diverse Publikationen ([6], [7] und [10]) sowie die Präsentation von Dr. Olbrich anlässlich der Einweihung des HLRN am 13.11.2002 belegen.

Zur Illustration des Anwendungsszenarios sei auf die *Abbildungen* 73 bis 76 verwiesen. Konkret verdeutlicht *Abbildung* 73 in einer Kombination aus modellierter Umgebung und simulierter 3D-Szenerie realitätsnah das von der Simulation umfasste Gebiet im Hamburger Hafen. Durch eine Modellierung der Umgebung des Simulationsgebiets sowie der Lichtsituation in 3D-Studio Max unter Verwendung eines Luftbilds, anschließenden Export ins VRML-Format und Konvertierung in das DVR-Format kann die gezeigte Ansicht als Komposition direkt im DocShow-VR-Viewer, mit den entsprechenden Explorationsmöglichkeiten, visualisiert werden. Demgegenüber zeigt *Abbildung* 74 am Modell den Verlauf des Hafenbeckens unter besonderer Kennzeichnung der am Einlauf des Köhlfleets platzierten Strömungsumlenkwand. Grundlage der numerischen Simulation ist eine Diskretisierung des Hafenbeckens, dargestellt in *Abbildung* 75. Es ist geplant, die Komplexität der Simulation zukünftig durch Verfeinerung des Gitters weiter zu steigern. Eine (stark überhöhte) Ansicht der eigentlichen Simulation ist in *Abbildung* 76 gezeigt. Der virtuelle Betrachter befindet sich in dem Hafenbecken (orange) unterhalb der Wasseroberfläche (blau) und blickt flussaufwärts. Zur Rechten liegt der Einlauf des Köhlfleets mit der Strömungsumlenkwand (rot). Aufgrund des Tidenhubs erfolgt die Strömung, verdeutlicht durch die Bahnlinien einzelner Partikel (grün), ebenfalls flussaufwärts. Bei dieser Anwen-

1. www.hydromech.uni-hannover.de

ung hat sich die 3D-Visualisierung und Exploration als ein ausgezeichnetes Hilfsmittel erwiesen, um das Verhalten der Strömung im kritischen Gebiet um die Strömungsumlenkwand zu studieren.



Abbildung 73. Umgebung des Simulationsgebiets im Hamburger Hafen.

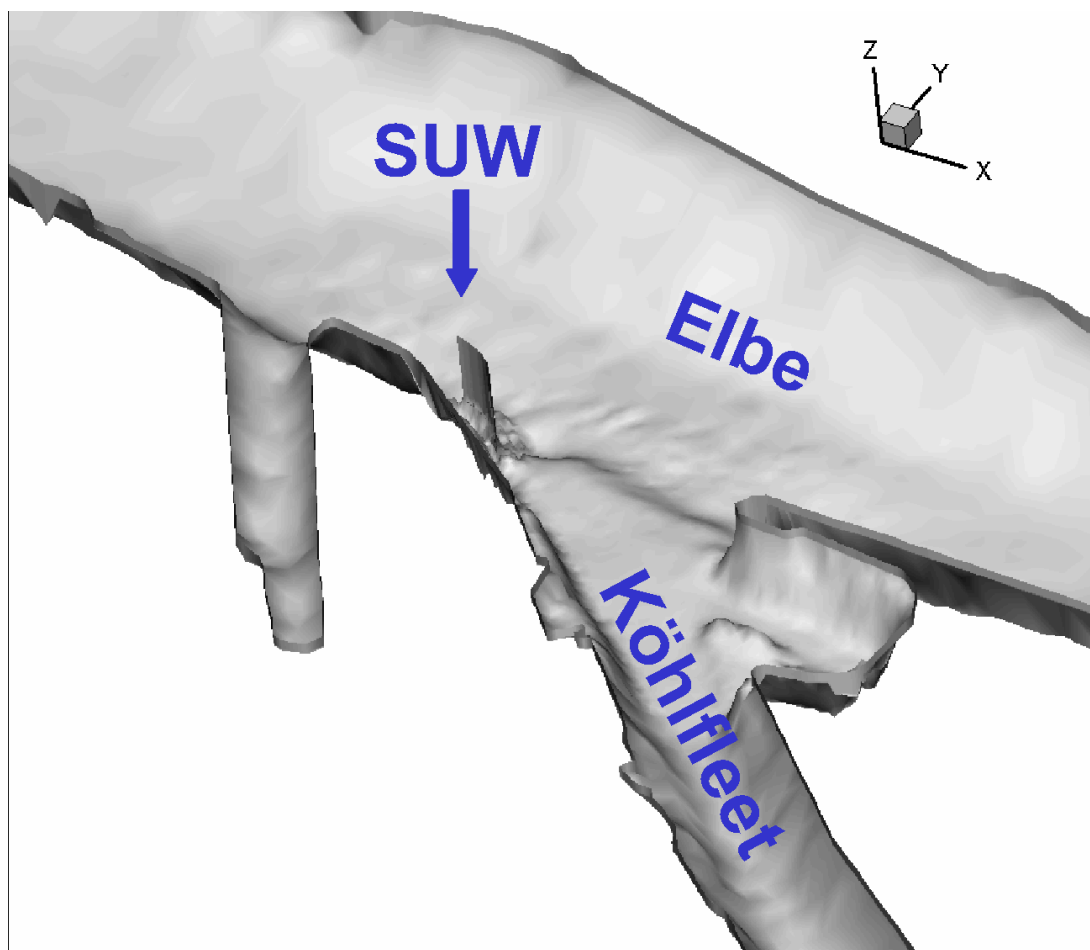


Abbildung 74. Strömungsumlenkwand (SUW) mit Sohlschwelle im Hafenbecken.

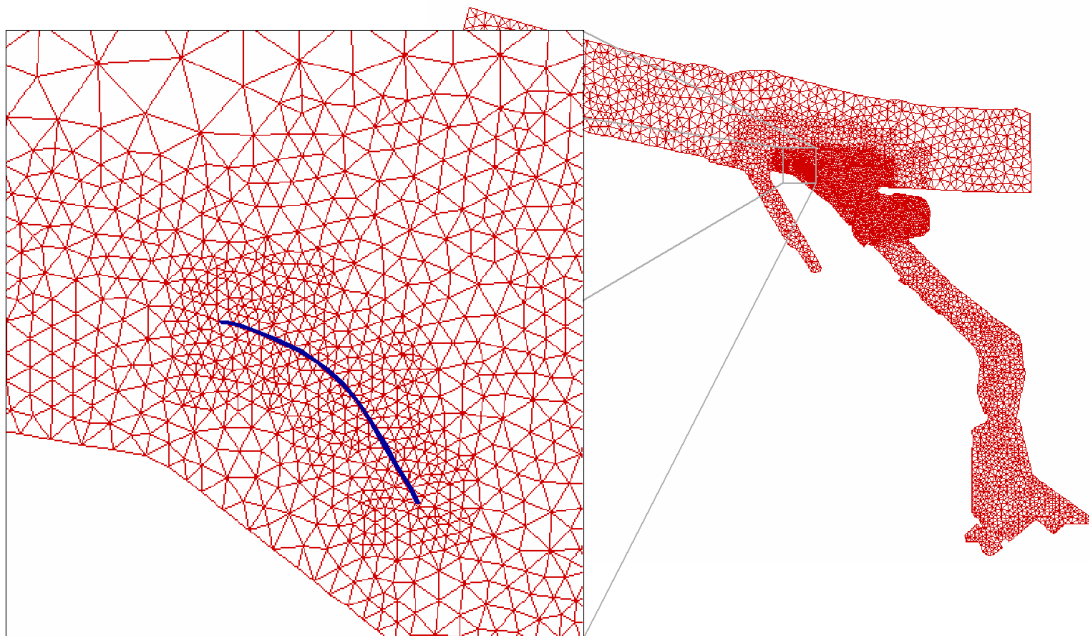


Abbildung 75. Diskretisierung: Finite-Elemente-Modell. Kantenlängen: 1m bis 200m, 10 Ebenen.

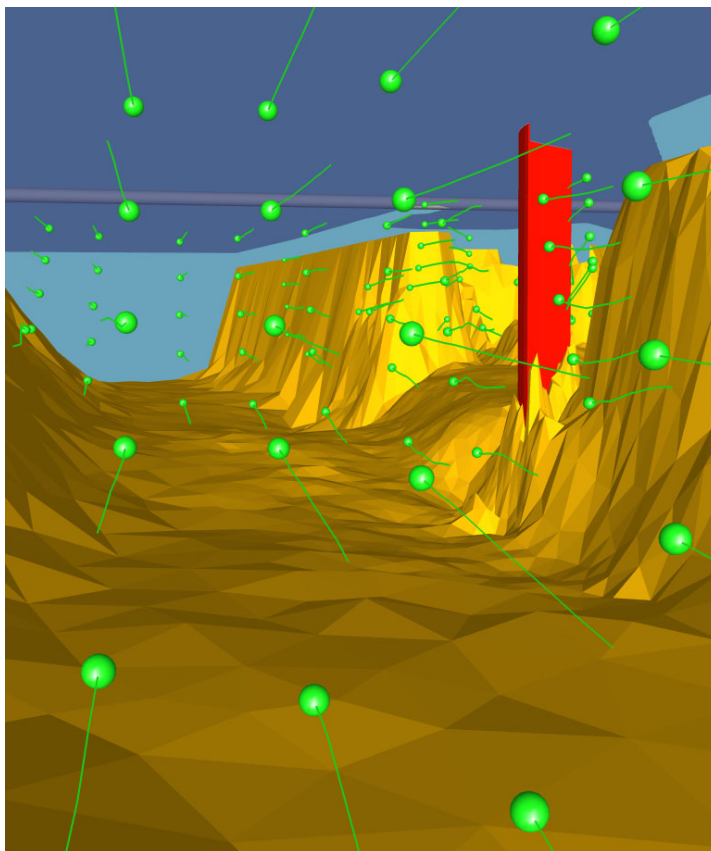


Abbildung 76. 3D-Visualisierung des Strömungsfeldes mittels animierter Partikelverlagerung und Bahnlinien in einer stark überhöhten Darstellung. Erläuterung siehe Text.

4 Zusammenfassung und Ausblick

4.1 Zusammenfassung der Ergebnisse

Sämtliche Arbeiten wurden im Wesentlichen wie geplant durchgeführt.

Das Projekt gliederte sich in „Teil A“ in die folgenden vier Abschnitte bzw. Meilensteine:

1. **Bereitstellung der 3D-Streamingfunktionalität** bezüglich der 3D/VR-Systemkomponenten und der anwendungsspezifischen Software-Integration, um die Anwendung des **Präsentations-szenarios** erproben und anwenden zu können. Dabei wurden batchartig erstellte 3D-Ergebnisszenen als „3D-Film“ abgerufen und mit Virtual-Reality-Methoden – in Raum und Zeitachse navigierbar – dargestellt.
2. **Ergänzung von Maßnahmen für „Online-Visualisierung“ und „Interactive Steering“**, um die Anwendung des **Explorationsszenarios** erproben und anwenden zu können. Dabei wurden 3D-Ergebnisszenen unmittelbar aus Simulationsrechnungen heraus generiert und als „3D-Film“ dargestellt. Mit Hilfe von konfigurierbaren, anwendungsspezifischen Menüs des 3D-Viewers konnten bestimmte Parameter der jeweiligen Simulationsrechnung gesteuert und die entsprechenden Einflüsse interaktiv beobachtet werden.
3. **Ergänzung von Maßnahmen zur Anwendung des Diskussionsszenarios**. Dabei wurden auf Client-Seite der vom RRZN entwickelten Viewer-Software Audio-/Videokomponenten sowie Komponenten zur Synchronisation der Navigation hinzugefügt.
4. **Weiterentwicklung der Visualisierungsbibliotheken**. Wie im Projektantrag vorgesehen, wurde am RRZN dabei der Ansatz verfolgt, spezialisierte Lösungen zu entwickeln, die auf die gegebenen Anwendungsfälle (jeweils als Modell für eine Klasse von Simulationsanwendungen) zugeschnitten sind. Dies ermöglichte mit relativ geringem Aufwand durch den Einsatz verfügbarer Implementierungen von bekannten Visualisierungsalgorithmen unter Ausnutzung der in den Simulationsrechnungen ohnehin bereits vorgenommenen Gebietszerlegung eine hochgradige Parallelisierung.

Zur prototypischen Realisierung und möglichst einfachen Nutzung und Erprobung dieses Lösungsansatzes wurde am RRZN die Programmbibliothek **libDVRP** entwickelt, in der Visualisierungsalgorithmen, Parallelisierung und Datenausgabe auf lokalen Dateien oder auf entfernten Servern (FTP bzw. 3D-Streaming über RTSP/TCP/IP) gekapselt sind.

Dagegen wurde am ZIB eine generische Visualisierungsbibliothek (**PaDEV**) entwickelt, in der in den niedrigen Layern bestimmte Funktionen aus der libDVRP genutzt werden.

In „Teil B“ war das Projekt in die folgenden zwei Abschnitte bzw. Meilensteine aufgeteilt:

1. Bereitstellung der **lokalen Gigabit-Ethernet-Netze** in Berlin (ZIB) und Hannover (RRZN).
2. Inbetriebnahme und Erprobung der **Gigabit-Ethernet-Weitverkehrrstrecke**.

Aufgrund von personellen Engpässen (insbesondere in Hannover – trotz mehrerer Ausschreibungen wurden letztlich fast 2 Personenjahre weniger Drittmittelpersonal als geplant eingesetzt) konnten jedoch bestimmte Projektziele im Detail nicht erreicht werden. Mit der Demonstrierbarkeit der Ergebnisse sowie der großen Akzeptanz bei der Nutzung der entwickelten, hochperformanten Werkzeuge seitens der Nutzer der Anwendungsteilprojekte wird u. E. der volle Erfolg dieses Projekts bestätigt.

4.2 Veränderungen der Rahmenbedingungen und erweiterte Kontexte

Die Rahmenbedingungen im Projektumfeld hatten sich gegenüber der Antragstellung im Jahr 1999 teilweise deutlich geändert bzw. erweitert.

Dies betrifft z. B. die Installation eines verteilten Großrechnerkomplexes HLRN¹, der in 2002 in Betrieb genommen wurde und damit die bisherigen Parallelrechner SGI/Cray T3E in Hannover (Entwicklungsrechner) und Berlin (Hochleistungsrechner) ablöst. Dadurch entstand die Aufgabe, die bislang prototypische, modellhafte Implementierung der Software-Module, die ursprünglich auf der T3E-Architektur entwickelt und auch bereits erfolgreich angewandt wurden (libDVRP), zu portieren. Auch auf weitere Portierungswünsche seitens der Anwendungspartner (z. B. für Hochleistungsrechner im Institut bzw. bei z. T. internationalen Kooperationspartnern) wurde eingegangen.

Ein weiteres Beispiel ist die gewünschte hochperformante Kopplung der HLRN-Komplexe – in Berlin (ZIB) und Hannover (RRZN) jeweils 12 so genannte Regatta-Knoten (IBM pSeries 690) mit je 32 Prozessoren². Vom RRZN und ZIB wurden – nach Absprache mit dem DFN – diese Rechner seit April 2002 über die Gigabit-Ethernet-Strecke dieses Projekts miteinander verbunden, um Erfahrungen zu sammeln, die zur Planung einer leistungsfähigen Netzverbindung für den künftigen Produktionsbetrieb beitragen sollten.

Neben den in *Teilprojekt 4 – Simulation atmosphärischer Konvektionsströmungen*, *Teilprojekt 5 – Molekular-dynamische Simulation* und dem zusätzlich durchgeführten *Teilprojekt 6 – Simulation von Strömung und Sedimentation in einem Tidehafen* behandelten **e-Science**-Anwendungen wurden die Visualisierungsansätze inzwischen in ein grundlegend anderes Anwendungsumfeld adaptiert: Zur Einweihung des „Learning Lab Lower Saxony“ (L3S³) – einem Forschungszentrum für **e-Learning**, das als Einrichtung der Universität Hannover mit bundesweiten Partnern gegründet wurde, im Deutschen Pavillon auf dem ehemaligen Expo-2000-Gelände angesiedelt wurde und in einen internationalen Verbund (WGLN⁴) eingebunden ist – wurde am 28.01.2002 demonstriert, wie *kollaborative 3D-Visualisierung in Lehr-/Lernumgebungen* eingesetzt werden kann (siehe auch *Abbildung 77*). In einem L3S-Projekt, an dem das RRZN beteiligt ist, werden spezielle Werkzeuge ergänzt, und auch die Erprobung in der universitären Lehre ist für das Sommersemester 2003 geplant.

-
1. HLRN – Norddeutscher Verbund für Hoch- und Höchstleistungsrechnen (<http://www.hlrn.de>).
 2. Siehe auch in RRZN-BI 357 (März 2002):
Heerhorst, W.: „Supercomputer für Norddeutschland“.
(<http://www.rrzn.uni-hannover.de/BIs/Jahrgang2002/BI357/bi35701.htm>)
 3. L3S: www.learninglab.de
 4. WGLN = Wallenberg Global Learning Network (<http://www.wgln.org>):
Stanford Learning Lab (USA) / Swedish Learning Lab / Learning Lab Lower Saxony.



Abbildung 77. 3D-Demonstration zur Einweihung im Learning Lab Lower Saxony (L3) am 28.01.2002.
Von links nach rechts: Dr. Siegfried Raasch (Institut für Meteorologie und Klimatologie, Universität Hannover), Dr. Stephan Olbrich (Regionales Rechenzentrum für Niedersachsen, Universität Hannover), Sigmar Gabriel (Ministerpräsident Niedersachsen), Dr. Christoph Strutz (Geschäftsführer Learning Lab Lower Saxony), Edelgard Bulmahn (Bundesministerin für Bildung und Forschung), Thomas Opper mann (Niedersächsischer Minister für Wissenschaft und Kultur).

4.3 Einfluss der Ergebnisse und Zukunftsperspektiven

Der erzielten Ergebnisse sind in Bezug auf die Realisierung leistungsfähiger, netzverteilter Visualisierungsansätze im Kontext „High-Performance Computing and Networking“ von außerordentlich großer Bedeutung. Wesentliche Beiträge zu den aktuellen und zukunftsorientierten Themen GRID-Computing und -Visualisierung sowie Gigabit-Networking wurden geleistet. Die dadurch aufgebaute Erkenntnis- und Entwicklungsbasis ermöglicht nun weitere Schritte, zu denen beispielsweise das DFN-Projekt „Virtueller Supercomputer“ (RRZN/ZIB) gehört, aber auch Planungen der nächsten Leistungsstufe hinsichtlich der Kommunikationsverbindung der beiden Teilkomplexe des Norddeutschen Verbundes für Hoch- und Höchstleistungsrechnen (HLRN) zwischen Berlin und Hannover (10-Gigabit-Ethernet).

Aufgrund des aus unserer Sicht im Wesentlichen erfolgreichen Projektverlaufs und des offensichtlich vorhandenen Zukunftspotenzials des erreichten Entwicklungsstandes wird eine Verlängerung des Projekts angestrebt. Da Personalmittel in nennenswertem Umfang nicht ausgeschöpft wurden (insbesondere in Hannover), wurde das Projekt um 2 Monate kostenneutral verlängert. Allerdings besteht u. E. der Bedarf, das Projekt – mit erweiterten Zielsetzungen – noch darüber hinaus weiter zu fördern, um die inzwischen erkennbaren inhaltlichen Erweiterungen erarbeiten zu können.

Dabei könnten technische Gesichtspunkte vertieft werden, die sich in der bisherigen Projektlaufzeit als zunehmend wichtig herausgestellt haben und die für ein eventuelles Folgeprojekt von Bedeutung sind:

- Portierung und Optimierung auf den Norddeutschen Verbund für Hoch- und Höchstleistungsrechnen (HLRN), einem IBM-Parallelrechner mit je 384 Prozessoren in Berlin (ZIB) und Hannover (RRZN).
- Ergänzung weiterer effizienter, skalierbarer Visualisierungsalgorithmen zur Unterstützung weiterer Datentypen und -strukturen mit hoher Komplexität, sowohl auf Seite der Aufbereitung (libDVRP-Client) als auch für den Viewer-Client des netzverteilten 3D-Streaming-Visualisierungssystems.
- Ergänzung weiterer Server-Softwarekomponenten in der Visualisierungspipeline des „Distributed Virtual Reality“-Systems:
 - Entwurf und Implementierung eines Remote-Rendering-Services. Dafür verwendbare, leistungsfähige Rendering-Hardware ist im RRZN und im ZIB vorhanden (s. u.).
 - Entwurf und Implementierung eines netzverteilten Proxy-Cache-Server-Konzeptes.
- Anpassungen der Client- und Server-Software an aktuelle Hochleistungsvisualisierungssysteme, z. B. SGI Onyx3800/4xIR4 im RRZN und ZIB.
- Unterstützung weiterer aktueller 3D-Präsentations- und Interaktionssysteme, z. B. Holobench, Stereo-HDTV-Recorder, Tracking- und Haptikgeräte im RRZN und ZIB.
- Erschließung weiterer Anwendungsfelder in Pilotprojekten, wie z. B. dem Teilprojekt 6 – Simulation von Strömung und Sedimentation in einem Tidehafen, welches vom RRZN in Kooperation mit dem Institut für Strömungsmechanik und Elektronisches Rechnen im Bauwesen an der Universität Hannover begonnen wurde. Hierzu zählen auch Virtual-Reality-Pilotanwendungen zur Mikrochip-Visualisierung und zur Simulation und Visualisierung produktionstechnischer Anlagen wie z. B. Werkzeugmaschinen.
- Erprobung von 10-Gigabit-Ethernet (HLRN und Hochleistungsvisualisierungsrechner werden voraussichtlich noch im Jahr 2003 10-Gigabit-Ethernet-Interfaces bekommen).

Einige dieser Aspekte erfordern eine intensivere Forschungs- und Entwicklungsarbeit, um ein genügend tiefes Verständnis sowie eine stabile und allgemein verwendbare Realisierung zu erhalten. Daher sind wir an einer Fortsetzung der Arbeiten interessiert und würden gern einen entsprechenden Antrag stellen, sobald ein generelles positives Votum vorliegt und die Art und Weise der Antragstellung geklärt ist.

5 Referenzen

1. Amira Visualization Environment.
<http://amira.zib.de>
2. AVS – Advanced Visualization System.
<http://www.avs.com>
3. Benger, W., Hege, H.-C., Merzky, A., Radke, T., Seidel, E.: *Data Description Via a Generalized Fiber Bundle Data Model*. 2001.
4. Benger, W., Hege, H.-C., Merzky, A., Radke, T., Seidel, E.: *Schwarze Löcher sehen*. DFN-Mitteilungen, Heft 52, 2000.
5. Chmielewski, K.: *Software-Dokumentation mit dem DocBook DTD*
http://www.hlrn.de/support/intern/hlrn_support_2002.06.26B_docbook.pdf
6. Doxygen Documentation System:
<http://www.doxygen.org>
7. Fiber Bundle HDF5 Interface:
<http://www.zib.de/benger/F5/>
8. Hairer, E., Norsett, S. P., Wanner, G.: *Solving ordinary differential equations I — nonstiff problems*, 2nd edition, Springer Series in Computational Mathematics, Springer-Verlag (1993).
9. HDF5-Protokoll:
<http://hdf.ncsa.uiuc.edu/HDF5>
10. HLR Nord – Norddeutscher Verbund für Hoch- und Höchstleistungsrechnen.
<http://www.hlrn.de>
11. Jensen, N., Olbrich, S., Pralle, H., Raasch, S.: *An Efficient System for Collaboration in Tele-Immersive Environments*. Proceedings of 4. Eurographics Workshop on Parallel Graphics and Visualization, Blaubeuren, Germany, 9.–10.09.2002.
12. Lorensen, W. E. and Cline, H. E.: *Marching cubes: a high resolution 3d surface reconstruction algorithm*. Computer Graphics 21(4): 163–169, 1987.
13. Olbrich, S., Pralle, H.: *A Tele-Immersive, Virtual Laboratory Approach based on Real-Time Streaming of 3D Scene Sequences*. Proceedings of ACM Multimedia Conference 2001, Ottawa, Canada, 30.09.–05.10.2001.
14. Olbrich, S., Pralle, H., Raasch, S.: *Using Streaming and Parallelization Techniques for 3D Visualization in a High-Performance Computing and Networking Environment*. In: Hertzberger, B., Hoekstra, A., Williams, R.: *High-Performance Computing and Networking*. 9th International Conference, HPCN Europe 2001, Amsterdam, The Netherlands, 25.–27.06.2001, Proceedings. Lecture Notes in Computer Science, Vol. 2110, Springer-Verlag.
15. Pooma.
<http://www.pooma.com>
16. pV3 – Parallel Visualization Library.
<http://raphael.mit.edu/pv3/pv3.html>
17. Schussman, G., Ma, K.-L., Schissel, D., Todd, E.: *Visualizing DIII-D Tokamak magnetic field lines*. Proceedings of the conference on visualization '00, IEEE Computer Society Press.
18. Theisel, H.: *CAGD and Scientific Visualization*. Habilitation, University of Rostock, July 2001.
19. VTK – Visualization Toolkit:
<http://www.kitware.com>
20. Walsh, N., Muellner, L.: *DocBook: The Definitive Guide*, O'Reilly, 1999.
www.docbook.org
21. Weinkauff, T., Theisel, H.: *Curvature Measures of 3D Vector Fields and their Applications*. Proc. WSCG '2002. The 10th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision, Plzen, Czech Republic, February 4–8, 2002.
22. Zöckler, M., Stalling, D., Hege, H.-C.: *Interactive visualization of 3d-vector fields using illuminated streamlines*. Proceedings of IEEE Visualization '96, San Francisco, pages 107–113, 1996.

