

**FZR-187**

Juli 1997

*R. Fülle*

**Jahresbericht 1995/96**

**Abteilung**

**Kommunikation und Datenverarbeitung**

Herausgeber:  
FORSCHUNGSZENTRUM ROSSENDORF  
Postfach 51 01 19  
D-01314 Dresden  
Telefon (03 51) 26 00  
Telefax (03 51) 2 69 04 61

Als Manuskript gedruckt  
Alle Rechte beim Herausgeber

# **Jahresbericht 1995 / 96**

**Abteilung  
Kommunikation und Datenverarbeitung**

Herausgeber: R. Fülle

# Inhaltsverzeichnis

<b>1</b>	<b>Einordnung</b>	<b>4</b>
<b>2</b>	<b>Personelle Besetzung</b>	<b>5</b>
<b>3</b>	<b>Hauptergebnisse 1995 / 96</b>	<b>6</b>
<b>4</b>	<b>Zentrale Server</b>	<b>8</b>
4.1	Ausbau der Zentralen Server . . . . .	8
4.2	Hard- und Software-Ausstattung der Zentralen Server . . . . .	9
4.3	Betrieb der Zentralen Server . . . . .	12
4.4	Nutzung Zentraler Server . . . . .	13
<b>5</b>	<b>Datennetze</b>	<b>16</b>
5.1	Erweiterung des Verkabelungssystems . . . . .	16
5.2	Verbesserung der Performance . . . . .	16
5.3	Einbeziehung von ISDN . . . . .	17
5.4	Nutzung von Funk-LAN . . . . .	17
5.5	Untersuchungen zur Einführung von Breitband-Netzen . . . . .	18
5.6	Zugriff zu externen Netzen . . . . .	18
5.7	Erreichter Stand des Datennetzes . . . . .	19
<b>6</b>	<b>Benutzerservice</b>	<b>22</b>
6.1	Numerische Mathematik / Computeralgebra . . . . .	22
6.2	Fortran-Programmierung . . . . .	23
6.3	Paralleles Rechnen . . . . .	23
6.4	Visualisierung . . . . .	24
6.5	Textverarbeitung . . . . .	25
6.6	Informationsdienste . . . . .	26
6.7	Benutzerseminare . . . . .	27
<b>7</b>	<b>Wissenschaftliche Kooperation</b>	<b>28</b>
<b>8</b>	<b>Konferenzbeiträge und Vorträge</b>	<b>30</b>
<b>9</b>	<b>Ausführliche Berichte zu ausgewählten Themen</b>	<b>31</b>

# Vorwort

Mit Gründung des Forschungszentrums Rossendorf (FZR) im Dezember 1991 galt es zunächst, eine leistungsfähige Informatik-Infrastruktur am Forschungsstandort (FSR) aufzubauen und damit einen Rückstand von ca. 17 Jahren gegenüber dem Weltstand in möglichst kurzer Zeit aufzuholen\*.

Im Berichtszeitraum (1995/96) hatten die Arbeiten der Abteilung Kommunikation und Datenverarbeitung das Ziel, die Informatik-Infrastruktur am Forschungsstandort Rossendorf auszubauen, die Dienste an den Zentralen Servern zu erweitern und die Nutzer sowohl der zentralen als auch der dezentralen Rechentechnik zu unterstützen.

Der FSR verfügte Ende 1996 über mehr als 660 PCs, 90 Workstations und 80 XTerminals, die über ein strukturiertes Ethernet miteinander verbunden waren, Zugang zum Wissenschaftsnetz (WIN) und damit zu internationalen Netzen hatten. Im lokalen Netz des FSR waren mehr als 900 IP-Adressen aktiv. Als Zentrale Server standen 9 Rechner unterschiedlicher Architektur und Leistungsprofile zur Verfügung.

Darüber hinaus galt es, neuen Anforderungen zu entsprechen und innovative Techniken einzuführen. Beispiele hierfür sind die Inbetriebnahme eines Parallelrechners, die Migration zu Breitbandnetzen, die Neugestaltung des File-Services sowie die Erzeugung von Videosequenzen.

Für die Förderung unserer Arbeiten sei den Zuwendungsgebern und dem Vorstand des FZR gedankt. Der Leitung der Hauptabteilung Technische Infrastruktur gilt unser Dank für vielfältige Unterstützung.

Ruprecht Fülle

---

\*Diese Entwicklung ist im Internet-Magazin für High Performance Computing, PRIMEUR, dargestellt: Harms, U.: The IT-development at the Research Centre Rossendorf near Dresden; PRIMEUR, March 1997  
<http://www.hoise.com/primeur/>

# 1 Einordnung

Die Abteilung Kommunikation und Datenverarbeitung (FVTK) versteht sich als Koordinierungs- und Kompetenzzentrum für die Prozesse der Datenverarbeitung und Datenkommunikation am Forschungsstandort Rossendorf. Die Abteilung ist Teil der wissenschaftlich-technischen Infrastruktur und der Hauptabteilung Technische Infrastruktur des FZR zugeordnet.

Die Abteilung gliedert sich entsprechend ihren Aufgaben in die Arbeitsgruppen

- Zentrale Server
- Datennetze
- Benutzerservice.

Sie verfügt über 15 Planstellen, darunter 5 Wissenschaftlerstellen, und über 1 Annexstelle (Doktorand).

Die Abteilung FVTK arbeitet eng mit den Verantwortlichen für Datenverarbeitung in den Organisationseinheiten des FZR sowie des Vereins für Kernverfahrenstechnik und Analytik (VKTA) zusammen. Die Abstimmung der Strategien und wichtiger Aktivitäten auf dem Gebiet der Datenkommunikation und Datenverarbeitung ist Voraussetzung für das sichere Funktionieren des offenen Systems.

Die Abteilung FVTK ist hierbei für die Koordinierung des Gesamtsystems, ferner für die bedarfsgerechte Gestaltung und Betreuung der institutsüberschreitenden Datenkommunikation (Backbone-Netz, aktive Netzkomponenten und Zugang zu externen Netzen) am FSR zuständig.

Sie betreibt die Zentralen Server CONVEX C3220 und C3820, IBM RS/6000 Mod. 550, 4 \* HP 9000/700, seit 1995 den Parallelrechner CONVEX/HP SPP 1000 für ressourcenintensive Aufgaben als Ergänzung der Workstations im High-end-Bereich und seit 1996 das zweite IBM RS/6000 Mod. 591 als dedizierten Server. Gleichzeitig unterstützt die Abteilung die dezentrale Rechentechnik u. a. durch Dienste des File-Serving, wie File-Backup, und durch die Bereitstellung von Netzlaufwerken auf den Zentralen Servern für dezentrale PCs.

Der Benutzerservice der Abteilung umfaßt die Beratung und Unterstützung der Nutzer auf den Gebieten Numerik, Fortran- und C-Programmierung, Optimierung/Parallelisierung, Visualisierung, Informationssysteme (WWW), Textsatzsysteme sowie System- und Netzwerknutzung. Auch die Nutzer dezentraler Rechentechnik sind in diese Dienste einbezogen.

Zu den Aufgaben des Benutzerservices gehören außerdem die Koordinierung und Beschaffung von PC-Software. Hiermit soll der PC-Nutzer entlastet, die Software weitgehend vereinheitlicht und die Möglichkeit von Rabatten erschlossen werden. Die Abteilung verwaltet Campuslizenzen für Workstations der Fabrikate SUN und DEC. Sie betreut Campuslizenzen der Firmen IBM (RS/6000) und HP (9000/700) und gibt hierfür Systemunterstützung.

## 2 Personelle Besetzung

### Abteilungsleitung:

Doz. Dr. habil. Ruprecht Fülle

### Gruppe Zentrale Server

- Leitung der Gruppe, Systemmanagement: Dipl.-Ing. Ursula Fromm
- Dateien: Dipl.-Math. Helga Kirsch
- Technik der Zentralen Server: Dipl.-Ing. (FH) Klaus Kirsch
- Betriebsorganisation: Dipl.-Ing. (FH) Gerhard Nickel
- Operating der Zentralen Server: Andreas Geyer  
Katrín Paul

### Gruppe Datennetze

- Leitung der Gruppe, Netzmanagement: Dr.-Ing. Wolf-Jochen Linnemann
- Netztechniker: Dipl.-Ing. (FH) Uwe Eisold
- Netztechniker: Ing. Rolf Rüdrieh
- Netztechniker: Ing. Dieter Köhler

### Gruppe Benutzerservice

- Leitung der Gruppe: Dr. Manfred Kunicke
- Numerik und Fortran: Dipl.-Math. Erhard Müller
- Paralleles Rechnen: Dr. Manfred Kunicke
- Visualisierung: Dipl.-Math. Horst Böttger  
Dipl.-Math. Nils Schmeißer (Doktorand)
- Textsysteme: Dr. Galina Vinel
- Arbeitsplatz-Software: Ing. Therese Jäger

### 3 Hauptergebnisse 1995 / 96

Die Abteilung Kommunikation und Datenverarbeitung nahm 1995 einen Parallelrechner SPP 1000/CD-4N der Fa. HP/CONVEX mit 4 PA-RISC-Prozessoren in Betrieb. Dem waren detaillierte Untersuchungen zur Parallelverarbeitung an unterschiedlichen Rechnern vorausgegangen. Nunmehr steht mit den Zentralen Servern das gesamte Spektrum des Meta-Computings, das Vektorrechner (CONVEX-C-Serie), Skalarrechner (IBM RS/6000, HP 9000/700) und Parallelrechner (CONVEX/HP SPP 1000) umfaßt, zur Verfügung. Die Skalarrechner wurden 1996 durch eine Power-Workstation IBM RS/6000 Mod. 591, die u. a. als Datenbankrechner für Oracle 7 und damit als Datenbasis u. a. für ein institutsweites Bibliotheksinformationssystem dient, und eine Workstation HP 9000/700 ergänzt. Die Abteilung verfügte Ende 1996 über 9 Zentrale Server. Sie stehen dem gesamten Forschungsstandort über das Datennetz zur Verfügung.

Durch die Installierung eines Bandarchivs mit einer Kapazität von z. Zt. 700 GByte am Server C3820 eröffnen sich seit Ende 1995 neue Dimensionen der Sicherung und Verwaltung von Dateien. Das Dateiarchiv war Ende 1996 bereits zu einem Drittel belegt.

An den Zentralen Servern waren im Dezember 1996 ca. 600 Nutzer eingetragen, von denen 320 ständig aktiv waren. 55 Betreiber dezentraler PCs verwenden Speicherplatz der Zentralen Server als Netzlaufwerke.

Die Graphikdienste haben hinsichtlich ihres wissenschaftlich-technischen Niveaus und ihres Nutzungsumfanges einen bedeutenden Aufschwung genommen: Die Technologie der Videoerstellung wurde weiterentwickelt und bei der Erzeugung von Videos zur Simulation von Prozessen eingesetzt. Die Herstellung von Druckvorlagen und Vortragsfolien am A3-Farbdrucker Canon CLC700 (über Netz) hat sich mit der Bearbeitung und Herstellung von 41.000 Farbdrukken im Jahr 1996 zu einem zentrumsweit intensiv genutzten Dienst entwickelt.

Neben dem systematischen Ausbau des lokalen Datennetzes auf gegenwärtig über 900 aktive IP-Adressen mit standortweit mehr als 660 PCs, 90 Workstations, 80 XTerminals galten die Arbeiten der Einführung innovativer Netztechnologien, u. a. durch Anfertigung einer Studie über die Migration zu ATM, die Einbindung von ISDN und die Nutzung von Funk-LANs. Durch Lastanalysen, Umstrukturierung und Lasttrennung gelang eine Durchsatzverbesserung in stark belasteten Netzabschnitten. Der Anschluß des FZR an das Wissenschaftsnetz (WIN) konnte 1995 von 64 KBit/s auf 128 KBit/s aufgerüstet werden. In Vorbereitung befindet sich die Nutzung einer 2-MBit/s-Teilkapazität im neuen Breitband-WIN (B-WIN).

An den Zentralen Servern wurden zwei Systeme der Computer-Algebra, AXIOM und Maple, zur Nutzung bereitgestellt. Eine neue Aufgabe war die Einführung des parallelen Rechnens durch Erschließen neuer Tools, durch Programmanalysen, Nutzerberatungen und Hilfe bei der Parallelisierung von Programmen. In insgesamt 16 Seminaren wurden die Nutzer mit der Anwendung neuer Software-Produkte bekannt gemacht.

Der Benutzerservice beschaffte für die dezentrale Rechentechnik am Forschungsstandort im Jahr 1995 über 910 Software-Produkte im Wert von insgesamt 740 TDM. 1996 waren es 758 Software-Produkte im Gesamtwert von 680 TDM.

Die Abteilung betreut den WWW-Server des Forschungszentrums. Er erhielt 1995 Proxy-Funktionalität, um den Zugang zu externen Netzen zu entlasten. Die Gestaltung der WWW-Dokumente obliegt nunmehr einer Redaktionsgruppe, der Vertreter der Institute und Zentralabteilungen des FZR angehören.

Mitarbeiter der Abteilung waren in Projekte von Instituten einbezogen oder lösten spezielle Programmieraufgaben.

Die Abteilung beteiligte sich aktiv an der Ausbildung von Studenten und Doktoranden sowie von Bürokauleuten.

Im Berichtszeitraum fand eine Innensanierung des Rechnergebäudes (Gebäude 13c) statt. Die Zentralen Server erhielten u. a. eine neue Klimatechnik und eine DIN/VDE-gemäße Elektroinstallation mit einer neuen Einspeisung.

## 4 Zentrale Server

### 4.1 Ausbau der Zentralen Server

Schwerpunkt für den Zeitraum 1995/96 war die Bereitstellung

- eines Servers mit moderner Parallelrechner-Architektur
- eines Systems für die Archivierung großer Datenmengen sowie
- neuer Dienste, vor allem der Visualisierung.

Der Vektorrechner C3820 fand intensive Nutzung für rechenintensive Prozesse im Stapelbetrieb, während dialogintensive Applikationen zunächst auf der C3220 (hera) verblieben.

Wegen der nach wie vor intensiven Nutzung beider Anlagen wurden ihre Hauptspeicher auf jeweils 512 MByte aufgerüstet.

Ein Cluster, bestehend aus 3 HP-Workstations HP PA-RISC 9000, diente zur Einarbeitung in die Technologien des Parallelrechnens, verlor aber bald bei Erscheinen modernerer Architekturen mit Shared-Memory an Bedeutung.

1994 wurde ein Parallelrechner SPP 1000 der Fa. HP/CONVEX am Standort installiert. Er verfügt über eine Shared-Memory-Architektur und ist mit 4 Prozessoren HP PA-RISC 9000 ausgerüstet.

Die ursprünglich für den Aufbau eines Workstationclusters vorgesehenen HP-Workstations wurden als dedizierte Server eingesetzt, um die CONVEX C3220 zu entlasten:

- Die Workstation HP 9000/715 dient, mit einer Parallax-Karte ausgerüstet, zur Erzeugung von Videosequenzen.
- Auf den beiden Workstations HP 9000/735 laufen zentral bereitgestellte Anwendungen,
  - die Visualisierungssysteme AVS und IDL
  - der Browser Netscape und
  - das Textsystem  $\text{\LaTeX}$ .
- 1996 kam eine weitere Workstation HP 9000/735 hinzu, die u. a. die Software-Distribution, vor allem von HP-Produkten, übernimmt.

Da die am Standort vorhandenen Applikationen nur geringe Chancen für eine Parallelisierung boten, wurde die skalare Technik durch Kauf eines Rechners IBM RS/6000 Mod. 591 im Jahre 1996 aufgewertet. Argumente für die Anschaffung dieses Rechners waren:

- die guten Erfahrungen, die bisher mit dem Modell 550 gesammelt worden waren,
- die Verfügbarkeit eines breiten Spektrums von Applikationen auf dieser Rechnerfamilie
- die Migration rechenintensiver Aufgaben vom Modell 550 auf die Workstation Modell 591 sowie
- die Installation von Oracle 7 für ein campusweites Informationssystem.

Für das File-Archiv wurde eine Lösung der Fa. CONVEX gewählt, die eine weitere, sinnvolle Nutzung von Rechnern der C-Serie versprach. Das System basiert auf einem Roboter der Fa. Odetics. Es nutzt Digital Linear Tapes (DLT), also relativ robuste Speichermedien, mit max. 10 GByte Kapazität je Magnetband. Der Roboter ist mit 6 Laufwerken ausgerüstet, um auch das sogenannte "Family Concept" verwirklichen zu können. Als Cache steht ein Plattenspeicher mit 2 \* 9 GByte zur Verfügung. Das Archivsystem läuft unter UniTree und dem CONVEX-Produkt CVDM (Convex Virtual Disk Manager).

Die Erprobung des Systems am Rechner C3220 zeigte, daß wegen der hohen Belastung der Maschine durch UniTree der Dialogbetrieb spürbar behindert wird. Daher wurden Hard- und Software des Dateiarchivierungssystems an den Batch-Rechner C3820 überführt.

Inzwischen ist das Datenarchiv zu 30 % ausgelastet. Eine Beschreibung der Funktionen des Datenarchivs befindet sich unter den ausführlichen Darstellungen (Abschnitt 9) dieses Berichts.

Zu den Zentralen Servern zählt auch ein A3-Color-Drucker Canon CLC 700, der von einem dedizierten PC (Fiery) gesteuert wird, Druckaufträge über das lokale Netz entgegennimmt und ausführt.

## 4.2 Hard- und Software-Ausstattung der Zentralen Server

(Stand: Dezember 1996)

Seit der Inbetriebnahme des Parallelrechners SPP 1000 steht das gesamte Spektrum des sogenannten Meta-Computings, bestehend aus Skalar-, Vektor- und Parallelrechnern, dem Forschungsstandort zur Nutzung als Zentrale Server zur Verfügung.

### CONVEX C3220 (hera)

Anzahl Prozessoren:	2
Hauptspeicher:	512 MByte
Plattenspeicher:	35 GByte
Peripherie:	9-Spur-MB(1600, 6250 bpi), DAT, PS-Drucker (HP LaserJet III, 4M), Farbdrucker (Tektronix Phaser III, Canon CLC 10, Canon CLC 700)
Betriebssystem:	ConvexOS/V11.0
Compiler:	C/V5.0, Fortran77/V8.0, C++
Grafik:	AVS 3.9, GKS
Textsystem:	L <sup>A</sup> T <sub>E</sub> X
Bibliotheken:	VECLIB, VECLIB8, SCILIB, NAGLIB, CERNLIB, FZRLIB
weitere Software:	Debugger für alle Sprachen

### CONVEY C3820 (rhea)

Anzahl Prozessoren: 2  
Hauptspeicher: 512 MByte  
Plattenspeicher: 17 GByte + 63 GByte (SCSI)  
Peripherie: ACL6/176 (6 DLT), DAT  
Betriebssystem: ConvexOS/V11.0  
Archivierungssysteme: UniTree2.0, CVDM 1.1  
Compiler: C/V5.0, Fortran77/V8.0  
Grafik: AVS5.0

### CONVEY SPP 1000/CD-4N (vesta)

Anzahl Prozessoren: 4  
Hauptspeicher: 512 MByte  
Plattenspeicher: 8 GByte  
Betriebssystem: SPP-UX/V3.2  
Compiler: C/V6.2.1.0, Fortran/V9.3  
Bibliotheken: CERNLIB, LAPACK, NAGLIB, SCILIB, VECLIB, VECLIB8  
weitere Software: ConvexPVM, APC, Cxdb, Cxpa

### IBM RISC RS/6000 Mod. 550 (rita)

Hauptspeicher: 128 MByte  
Plattenspeicher: 12 GByte  
Peripherie: CD-ROM, Exabyte, 8mm-MB  
Betriebssystem: AIX 3.2.5  
Compiler: Fortran V3.1 (für Fortran90 und Fortran77)  
C/V1.2, C++/V1.1, Pascal/V1.1  
Grafik: AVS, GKS, Visualization Data Explorer  
Textsystem:  $\text{\LaTeX}$   
Bibliotheken: OSL, ESSL, NAGLIB, CERNLIB, FZRLIB

### IBM RISC RS/6000 Mod. 591 (rike)

Hauptspeicher: 128 MByte  
Plattenspeicher: 24 GByte  
Peripherie: CD-ROM, Diskette  
Betriebssystem: AIX 4.1.4  
Compiler: Fortran V3.2, CSet++ V3.1.2  
DBS: ORACLE 7.3.2

### **HP 9000/735 (apoll)**

Taktfrequenz: 99 MHz  
Hauptspeicher: 80 MByte  
Plattenspeicher: 7 GByte  
Betriebssystem: HP-UX 9.05  
Compiler: Fortran 9000/V9.16, C/Ansi 9.61  
Grafik: AVS 5.01  
Bibliotheken: BLAS Library, ConvexMLIB

### **HP 9000/735 (minos)**

Taktfrequenz: 99 MHz  
Hauptspeicher: 80 MByte  
Plattenspeicher: 7 GByte  
Betriebssystem: HP-UX 9.05  
Compiler: Fortran 9000/V9.16, C/Ansi 9.61  
Grafik: AVS 5.01  
Bibliotheken: BLAS Library, ConvexMLIB

### **HP 9000/715 (ikon)**

Taktfrequenz: 75 MHz  
Hauptspeicher: 64 MByte  
Plattenspeicher: 7 GByte  
Videosystem: Parallax PowerVideo700 Plus  
Betriebssystem: HP-UX 9.05  
Compiler: Fortran 9000/V9.16, C/Ansi 9.61  
Grafik: AVS 5.01, Parallax Video Development Environment  
Bibliotheken: BLAS Library, ConvexMLIB

### **HP 9000/715 (hermes)**

Taktfrequenz: 75 MHz  
Hauptspeicher: 32 MByte  
Plattenspeicher: 6 GByte  
Betriebssystem: HP-UX 10.20  
Einsatz: HP-UX SW-Depot

Eingeleitet wurde der Kauf eines Parallelrechners CONVEX/HP SPP 2000/S mit 8 Prozessoren HP PA-RISC 8000 und Shared-Memory-Architektur.

### 4.3 Betrieb der Zentralen Server

Die Zentralen Server werden rund um die Uhr betrieben. Operatoren sind nur werktags von 6.00 bis 16.30 Uhr anwesend. Außerhalb dieser Zeit kontrollieren die Systemverantwortlichen den Betrieb der Anlagen über Telefoneinwahl in das lokale Datennetz des FSR.

Abschaltungen wurden kurzzeitig und nur bei Erweiterungen oder Umbauten der Hardware vorgenommen. Störungen des Elektronetzes blieben, soweit sie innerhalb der Autonomiezeit der Unterbrechungsfreien Stromversorgung (30 min) lagen, ohne Auswirkungen auf den Betrieb der Server.

Auch während der umfangreichen Sanierungsarbeiten (März bis Juli 1995) im Rechnerraum und an den Versorgungseinrichtungen der Zentralen Server wurde der Rechner- und Netzbetrieb mit Ausnahme von drei Wochenenden aufrechterhalten. Die beiden großen CONVEX-Server, C3220 und C3820, wurden hierzu provisorisch gemeinsam mit einem Klimagerät in einem staubdichten Bauwerk, das innerhalb des Rechnersaals errichtet worden war, untergebracht und weiter betrieben.

## 4.4 Nutzung Zentraler Server

### Genutzte Rechenleistungen

Rechenleistungen an den Zentralen Servern wurden auch 1995/96 vor allem von den physikalischen und physikalisch-technischen Arbeitsrichtungen des Forschungsstandortes Rossendorf in Anspruch genommen (Tabelle 1).

		FWI <sup>0</sup>	FWK <sup>1</sup>	FWS <sup>2</sup>	Sonstige
1995	C3220 (hera)	8 %	54 %	17 %	21 %
	C3820 (rhea)	3 %	39 %	47 %	11 %
	SPP 1000 (vesta)	< 1 %	54 %	39 %	7 %
	RS/6000 Mod. 550 (rita)	99 %	< 1 %	< 1 %	1 %
1996	C3220 (hera)	11 %	32 %	7 %	50 %
	C3820 (rhea)	< 1 %	44 %	37 %	19 %
	SPP 1000 (vesta)	3 %	41 %	48 %	8 %
	RS/6000 Mod. 550 (rita)	94 %	< 1 %	1 %	5 %
	RS/6000 Mod. 591 (rike)	53 %	< 1 %	24 %	23 %

Tabelle 1: Anteil an der genutzten Prozessorkapazität Zentraler Server

Der Hauptanteil der Prozessorzeiten an den Zentralen Servern wurde für folgende Einsatzgebiete genutzt:

- Materialforschung
  - Molekulardynamische Simulation von Hartstoffschichten
  - Dichtefunktionalrechnungen

<sup>0</sup>Institut für Ionenstrahlphysik und Materialforschung

<sup>1</sup>Institut für Kern- und Hadronenphysik

<sup>2</sup>Institut für Sicherheitsforschung

- Kern- und Hadronenphysik
  - Diagonalisierung großer Matrizen zur Eigenwertberechnung
    - \* für die Berechnung von Einteilchenenergie-Spektren von Quarks
    - \* für Schalenmodellrechnungen in der Nähe von  $N=50$   
(Matrizen bis zum Rang 19.000!)
  - Monte-Carlo-Berechnungen von Kernreaktionen mittlerer Energie
- Nukleare Sicherheit / Entsorgung
  - Monte-Carlo-Rechnungen zur Druckkessel-Versprödung an Reaktoren russischer Bauart
  - Bestimmung von Neutronengruppen-Querschnitten mittels des Codes NJOY
  - 3D-Monte-Carlo-Rechnungen zur Kritikalität von Forschungsreaktoren
  - Rechnungen zur Rekritikalität von Leistungsreaktoren nach schweren Störfällen
  - Berechnung der Kritikalität u. a. bei der Umlagerung von Spaltmaterial
  - Transportrechnungen für Gamma- und Neutronenfelder zur Dosisbestimmung
  - Analyse von Röntgen-Kleinwinkel-Streudaten zur Charakterisierung von mikrostrukturellen Schäden
  - Monte-Carlo-Rechnungen unter Anwendung von PVM zur Modellierung einer Plasma-Neutronenquelle
- Strahlenschutz
  - Ausbreitungs- und Expositionsrechnungen für die Strahlenschutz-Umgebungsüberwachung
- Beschleunigertechnik
  - Simulation von Beschleunigungsprozessen
  - Entwicklung und Anwendung moderner Optimierungsverfahren für komplexe Systeme, u. a. zur Strahloptimierung am Beschleuniger "ELBE"

### **Infrastrukturelle Dienste**

Neben der Nutzung als Compute-Server erfüllen die Zentralen Server infrastrukturelle Aufgaben:

- als File-Archiv (C3820 mit einem Bandroboter ALT Odetics), wobei dessen Volumen von 700 GBytes zu 30 % durch die Nutzer belegt worden ist,
- als File-Server (C3220) für das File-Backup, u. a. für dezentrale Workstations; Ende 1996 waren 23 Workstations in diesen Dienst, der sowohl die zyklische Sicherung aller als auch das tägliche Backup veränderter Dateien umfaßt, einbezogen;

- als Server (C3220) zur Bereitstellung von Netzlaufwerken ("virtuellen Laufwerken") für gegenwärtig 55 dezentrale PCs
- als Datenbankserver (RS/6000 Mod. 591, Datenbank Oracle 7 mit WWW-Server) für Intranet-Informationdienste, z. Zt. für das Pilotprojekt ROBIS (Rossendorfer Bibliotheks-Informationssystem)
- als Distribution-Server (HP 9000/735, "hermes") für HP-Software
- als Fontserver für XTerminals (RS/6000 Mod. 591)
- als WWW-Server (C3220)
- Netnews-Server (C3220)
- Ftp-Server (C3220) und
- als Name-Server (C3220) der Internet-Domain fz-rossendorf.de.

Von den genannten Aufgaben wurden in den Jahren 1995/96 das File-Archiv, der Datenbankservice und die HP-Software-Verteilung neu eingerichtet und bereitgestellt.

### Spezielle Dienste

Neben den genannten Leistungen hat die Abteilung in den Jahren 1995/96 das Spektrum ihrer Dienste erweitert.

- Seit 1995 steht ein hochwertiger A3-Color-Drucker (Canon CLC 700 mit einem Steuerrechner) neben dem älteren A4-Drucker (Tektronix, Sublimationsdrucker) zur Verfügung.  
Standortweit können Dateien über das Netz zum Drucker übertragen werden. Der Operator kontrolliert die Graphikdateien und nimmt nach Rücksprache mit dem Nutzer gegebenenfalls Korrekturen vor.  
Das Drucken von graphischen Darstellungen auf Papier und Folie hat sich zu einem intensiv genutzten Dienst entwickelt:  
1996 wurden insgesamt 41.000 Seiten und Folien gedruckt.
- Zur Erzeugung von Video-Sequenzen auf professionellem Niveau stehen seit 1995 eine entsprechende Gerätekette und leistungsfähige Tools zur Verfügung (vgl. Abschnitte 6.4 und 9).
- Ein Windows-NT-Server wurde eingerichtet, auf dem die Tektronix-Software WinDD verfügbar ist. Sie gestattet, Microsoft-PC-Applikationen von Tektronix-XTerminals aus zu nutzen.
- Ein weiterer PC unter Windows NT dient als zweiter CD-ROM-Server. Er ist mit einem CD-ROM-Tower ausgerüstet und kann z. Zt. im Netz insgesamt 7 CD-ROMs bereitstellen.
- Das Brennen von CD-ROMs wird seit 1996 angeboten. Der CD-ROM-Brenner läßt sich je nach gewünschtem Dateiformat an UNIX- oder MS-DOS-Rechner anschließen. Hauptnutzer sind gegenwärtig die Abteilung Mechanische Konstruktion (Zentralabteilung Forschungs- und Informationstechnik zur Archivierung von CAD-Unterlagen) und die Zentralbibliothek zum Sichern des FZR-Fotoarchivs.

## 5 Datennetze

### 5.1 Erweiterung des Verkabelungssystems

Das strukturierte Ethernet des Forschungsstandortes wurde 1995/96 weiter ausgebaut.

Auf Anforderung der Nutzer wurden vorhandene Inhouse-Verkabelungen ergänzt bzw. modifiziert. Geringfügige Erweiterungen übernahmen dabei Mitarbeiter der Gruppe Datennetze, während umfangreichere Arbeiten, wie der Aufbau ganzer Netzsegmente, Fremdfirmen übertragen wurden.

Im Zuge von Sanierung und Neubau erhielten folgende Gebäude neue Verkabelungssysteme:

- Gebäude 8b (Radiochemisches Labor)
- Gebäude 8d (ehemals Isotopenproduktion)
- Gebäude 93 (PET-Labor)
- Gebäude 17 (Detektorlabor)
- Gebäude 101 ("Alte Schmiede")
- Gebäude 68 (ehemals Nullreaktor)
- Gebäude 87 (ehemals SMA).

Hierbei kam durchweg eine Twisted-Pair-Verkabelung (S/STP-Kabel, Kategorie 5) zum Einsatz, um bei Bedarf auch Breitbandtechnologien auf diesen Systemen betreiben zu können.

Als aktive Komponenten wurden weiterhin Cabletron-MMAC und CISCO-Router verwendet.

Lichtwellenleiterkabel wurden zwischen folgenden Gebäuden neu verlegt:

- Gebäude 92 (PET-Labor) und Gebäude 93 (PET-Labor)
- Gebäude 92 (PET-Labor) und Gebäude 17c (Detektorlabor)
- Gebäude 90 (TZ I) und Gebäude 8d (ehemals Isotopenproduktion)
- Gebäude 7 (Zyklotron) und Gebäude 97b (Tandem).

Sie ersetzen zum Teil ältere Kupferkabel.

Für alle diese Projekte erarbeitete die Abteilung die Aufgabenstellung und legte die technologische Konzeption fest.

### 5.2 Verbesserung der Performance

Untersuchungen der integralen Netzlast im LAN ergaben einen sehr hohen Datenverkehr, besonders im Subnetz 8 (Institut für Ionenstrahlphysik und Institut für Kern- und Hadronenphysik) und im Subnetz 12 (Zentralabteilung Forschungs- und Informationstechnik, Institut für Sicherheitsforschung, Zentralbibliothek). Eine Verringerung der Netzlast gelang durch Trennung der Netzabschnitte mittels Bridges.

Neben den Subnetzen 8 und 12 war auch das Subnetz 4, an dem die Zentralen Server und die übrigen Rechner der Abteilung angeschlossen sind, sehr hoch belastet. Die hohen Datenflüsse kommen durch die Zugriffe auf die Zentralen Server, aber vor allem durch Kommunikation der Zentralen Server untereinander infolge der gemeinsamen Nutzung von Dateien und Kopplung der Server über NIS (Network Information Service) zustande.

Die Durchlässigkeit des Subnetzes 4 wurde 1995 durch Einbau eines Switches (ESXMIM der Fa. Cabletron) mit 8 Anschlüssen erhöht. Die Ankopplung des Parallelrechners SPP 1000 und der beiden Workstations IBM/RS 6000 geschieht über FDDI (Modul FDMIM 04 der Fa. Cabletron). Switch- und FDDI-Modul sind über das schnelle Backplane des gemeinsamen MMAC miteinander verbunden.

### 5.3 Einbeziehung von ISDN

1996 wurde am Forschungsstandort ISDN (Integrated Services Digital Network) für die Datenkommunikation erschlossen. Grundlage hierfür war die TK-Anlage HICOM 390 (Fa. Siemens). Sie verfügte Ende 1996 über 48  $U_{p0}$ -Anschlüsse.

Erste Anwendung von ISDN am Standort war die Anbindung der vom Kerngelände entfernt gelegenen Gebäude 101 und 101.1 an das LAN des Forschungsstandortes. Der Übergang zum LAN erfolgt über den zentralen Router CISCO 4500, der u. a. mit 8 BRI-Anschlüssen ausgerüstet ist. Die Gegenstelle im Gebäude 101 bildet ein Router CISCO 2500. Diese Lösung soll als Prototyp für weitere ähnliche Anbindungen dienen.

Als weitere ISDN-Anwendung wurde 1996 das Telekonferenz-System ProShare (Fa. Intel/Telekom) installiert. Es ist, wie in Abschnitt 9 dargelegt, für die Telekommunikation zwischen dem PET-Labor des FZR und der Klinik für Nuklearmedizin der TU Dresden bestimmt und befindet sich im Probebetrieb.

### 5.4 Nutzung von Funk-LAN

Als alternative Möglichkeit zur Ankopplung entfernter oder mobiler Datenendgeräte im Ethernet wurde das Funk-LAN "ARLAN 600" der Fa. Daimler-Benz Aerospace erprobt und mit Erfolg genutzt. Bei einer Sendeleistung von 100 mW und einer Sendefrequenz von 2,4 GHz lassen sich Reichweiten von 150 m in Gebäuden und 500 m im Freibereich erzielen. Die Übertragungsrate beträgt bis zu 2 MBit/s. In Analogie zu verkabelten Netzen existieren für das ARLAN Funkhubs, Funkbridges, Funkadapter für ISA-PCs und PCMCIA-Adapter für Notebooks.

Im Einsatz befinden sich ein Funkbridge-Paar zur Meßdatenübertragung am Tandembeschleuniger sowie ein Funkhub und ein PCMCIA-Adapter für den Betrieb eines mobilen Notebooks.

## 5.5 Untersuchungen zur Einführung von Breitband-Netzen

Sowohl der wachsende Datenverkehr im LAN als auch die Ankündigung neuartiger Applikationen waren Anlaß zur Auseinandersetzung mit Breitband-Technologien und Möglichkeiten der Migration zu diesen.

Da künftig auch Videodaten aus PET- und Detektorlabor in Echtzeit übertragen werden sollen, stand ATM (Asynchronous Transfer Mode) zur Diskussion. Gleichzeitig war der Parallelbetrieb des 10-MBit/s-Ethernet an den übrigen Netzabschnitten gefordert.

Der Lösungsvorschlag sieht den Aufbau einer ATM-Substruktur mit ATM-Eingangsswitch zum WIN, einem ATM-Backbone-Bereich (155 MBit/s) mit einem weiteren ATM-Switch im PET-Labor und ATM-Anbindungen (155 MBit/s) an die bildgebenden und -verarbeitenden Endstellen vor. Die übrigen Bereiche sollen über Workgroup-Switches mit dem zentralen ATM-Eingangsswitch verbunden werden.

Für Abschnitte, über die in absehbarer Zeit keine Video- oder Audiodaten übertragen werden, ist der schrittweise Ersatz des 10-MBit/s-Ethernet durch Fast Ethernet (100 MBit/s) geplant.

## 5.6 Zugriff zu externen Netzen

Die Übertragungsgeschwindigkeit des Anschlusses zum Wissenschaftsnetz (WIN) (Adresse 450 50 17007) konnte im Jahr 1996 von 64 KBit/s auf 128 KBit/s erhöht werden. Das WIN diente dabei fast ausschließlich als Transportnetz vom/zum Internet. Das FZR hat die Domain-Adresse 149.220.. bzw. fz-rossendorf.de. In geringem Umfange werden über das WIN Datex-P-Dienste, wie Electronic Banking, genutzt. Zugriffe vom/zum EARN spielen keine Rolle mehr.

Wurden im zweiten Halbjahr 1994 im Mittel 1300 MByte je Monat übertragen, so waren es im zweiten Halbjahr 1995 4000 MByte/Monat und im gleichen Zeitraum 1996 6000 MByte. Diese Angaben sind Mittelwerte. Sie verdeutlichen die drastische Zunahme des Datenverkehrs mit externen Partnern und die wachsende Nutzung des Internet insgesamt. Allerdings konzentriert sich der externe Datenverkehr auf die Arbeitszeit: Er erreicht von 8.00 Uhr an bis 20.00 Uhr sein Maximum von 25 % bis 70 % der maximalen Transferleistung.

In Vorbereitung befindet sich ein 2-MBit/s-Anschluß an das Breitband-WIN, ein ATM-Netz mit IP-Protokoll. Der Forschungsstandort wird eine 2-MBit/s-Teilkapazität des 34-MBit/s-Anschlusses an der TU Dresden nutzen. X.25 wird als WIN-Protokoll am Standort nicht mehr zur Verfügung stehen, so daß auch der unmittelbare Zugriff auf Datex-P entfällt. Es ist vorgesehen, Electronic Banking künftig über ISDN abzuwickeln. Die Umstellung auf das 2-MBit/s-B-WIN ist für Mitte 1997 geplant.

## 5.7 Erreichter Stand des Datennetzes

Das lokale Netz (LAN) des Forschungsstandortes verfügte Ende 1996 über mehr als 900 aktive Adressen, u. a. von ca. 660 PCs, 90 Workstations und 80 XTerminals. Das Netz verbindet flächendeckend die Gebäude des Kerngeländes am Forschungsstandort.

Das LAN stellt ein strukturiertes Ethernet dar. Es ist mittels zweier Router (CISCO 4500 und AGS 3+) in Subnetze untergliedert (Abbn. 1 und 2).

Das Backbone-Netz hat die Struktur eines Sterns. Die Gebäude sind über Lichtwellenleiter-Kabel (LWL) (62,5  $\mu$  Multimode, 12 bis 24 Gradientenfasern) verbunden. Entsprechend der technologischen Entwicklung bestehen am Standort unterschiedliche Inhouse-Verkabelungssysteme:

- Thinwire-System:  
Gebäude 4, 4a, 4b, 6, 7, 8a, 9, 13;  
Installierung: 1992/93.
- Twisted-Pair-Systeme:
  - IBM-Typ-1-Kabel:  
Gebäude 94, z. T. Gebäude 13;  
Installierung: 1991/92;
  - S/STP-Kabel, Kategorie 5:  
Gebäude 7, 8b, 8d, 13c, 87, 90, 91, 92, 93, 101;  
Verlegung ab 1993 als verbindliche Norm.

Als aktive Komponenten werden überwiegend, seit 1992 durchgängig Cabletron- und CISCO-Produkte eingesetzt. Lediglich bei einigen älteren Verkabelungen ist noch Hirschmann-Technik im Einsatz.

Besondere Lösungen wurden im Subnetz 4 (Rechenzentrum) gewählt: Die Zentralen Server sind über einen Cabletron-Switch, z. T. auch über FDDI, untereinander und mit dem übrigen LAN verbunden.

Über das Ethernet werden die Protokolle TCP/IP, DECnet, ferner IPX betrieben. Hinzu kommen die Protokolle NetBIOS und NetBEUI für die Einbindung von PCs. Sie stellen die Verbindung von PC- und UNIX-Welt her. Als Netzwerkbetriebssystem ist NFS installiert. Daneben wird das SMB-Protokoll (Server-Message-Block) genutzt, das durch das Software-Produkt SAMBA unterstützt wird. Nunmehr können die Zentralen Server Dienste für am Netz befindliche PCs sowohl über NFS als auch über SMB bereitstellen.

Der Anschluß des LAN zum Wissenschaftsnetz (X.25-Protokoll) hat gegenwärtig eine Übertragungsrate von 128 KBit/s.

Das gesamte LAN wird mit Hilfe des Systems SPECTRUM (Fa. Cabletron) überwacht und gemanagt.



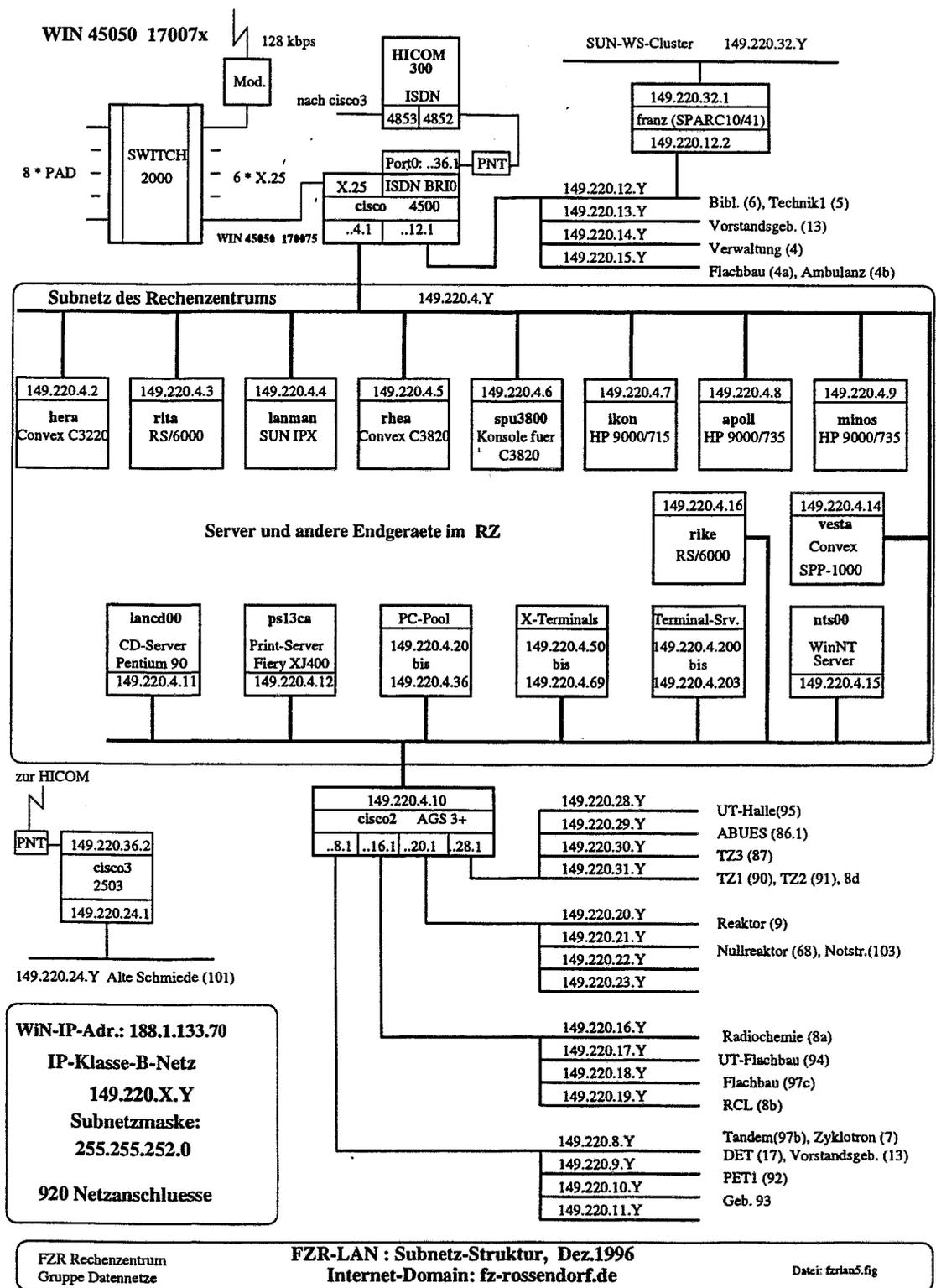


Abb. 2: Subnetzstruktur des lokalen Netzes am Forschungsstandort Rossendorf

## 6 Benutzerservice

Das Aufgabenspektrum der Gruppe Benutzerservice erstreckte sich auch 1995/96 von der Beratung auf den Gebieten Numerische Mathematik, Fortran-Programmierung, Datenvisualisierung, Parallelisierung, Nutzung von Textsystemen bis hin zur Koordinierung und Beschaffung von Arbeitsplatz-Software für den gesamten Forschungsstandort. Ein Teil der inzwischen routinemäßig durch die Gruppe Zentrale Server angebotenen Dienste wurde von der Gruppe Benutzerservice erschlossen und installiert.

Alle an den Zentralen Servern neu bereitgestellten Software-Produkte wurden in Benutzerseminaren vorgestellt. Zu jedem Seminar erhielten die Nutzer Unterlagen mit Arbeitsanleitungen.

Im Zeitraum 1995/96 fanden insgesamt 16 Benutzerseminare statt. Ihre Themen sind im Abschnitt 6.7 zusammengestellt.

### 6.1 Numerische Mathematik / Computeralgebra

Auf dem Gebiet der numerischen Mathematik (einschließlich Programmierung) wurden Leistungen zu folgenden Aktivitäten erbracht:

- Verbesserung und Erweiterungen am Programmsystem RAPI zur Berechnung und graphischen Darstellung von Strahlenexpositionen
- Bereitstellung von Algorithmen zur Entfaltung von Tiefenprofilen, die mittels Positronenstrahlen verschiedener Energien gewonnen wurden,
- Tests und Bereitstellung von numerischen Algorithmen für 2-dimensionale Tschebyscheff-Interpolationen und -Quadraturen
- Erweiterungen am Programmsystem für die Mehrphasenregression.

Für C-Anwender wurden die Numerical Recipes-Routinen in C an den Zentralen Servern C3220 und C3820 bereitgestellt.

Für Ad-hoc-Rechnungen wurden auf Wunsch von Nutzern die Computeralgebra-Systeme AXIOM und Maple erschlossen.

Maple V, Rel. 4 ist inzwischen am Zentralen Server IBM RS/6000 Mod. 591 (rike) installiert. Im Nutzauftrag entstand ein Maple-Programmpaket zur Berechnung von Clebsch-Gordan-Koeffizienten sowie von 6j- und 9j-Symbolen.

Begonnen haben Testrechnungen mit einem weiteren Computeralgebra-System, Mathematica.

Die Bereitstellung mehrerer Computeralgebra-Systeme am Forschungsstandort entspricht den Wünschen vor allem junger Nutzer, die an den Umgang mit einem bestimmten System gewöhnt sind. Zudem haben die Systeme unterschiedliche Stärken und Schwächen, wie im Abschnitt 9 dargestellt ist.

## 6.2 Fortran-Programmierung

Fortran stellt nach wie vor die am häufigsten verwendete Programmiersprache am Forschungsstandort dar. Neben Fortran 77 stehen an den Rechnern IBM RS/6000 auch Fortran 90 in der Implementierung der Numerical Algorithm Group (NAG) und AIX XL FORTRAN V. 3.2 zur Nutzung zur Verfügung.

Die Fortran-Arbeitsumgebung wurde um den Syntax-Checker `ftnchek` erweitert. Er ist an den Zentralen Servern C3220, C3820, IBM RS/6000 Mod. 550 und IBM RS/6000 Mod. 591 installiert.

Am Parallelrechner SPP 1000 wurde die Fortran-Arbeitsumgebung erschlossen und bereitgestellt.

Eingerichtet wurden außerdem

- die CERN-Bibliothek in der Version 96a an den Zentralen Servern IBM RS/6000 und HP 9000 sowie SPP 1000
- die NAG Fortran Library, Mark 17 an den Zentralen Servern IBM RS/6000 und SPP 1000.

Die verfügbaren Rechnersysteme und ihre Fortran-Compiler wurden hinsichtlich ihrer Leistungsfähigkeit mittels folgender Programme getestet:

- LINPACK-Benchmark
- Matrizeninversion
- FZR-Meßjob.

Für mehrere Fortran-Programme wurde der Convex-Performance Analyzer (CXpa) eingesetzt, um die Programmeinheiten zu ermitteln, die die meiste Zeit beanspruchen. Durch die Optimierung dieser Einheiten konnte die Effizienz der ausgewählten Fortran-Programme verbessert werden.

## 6.3 Paralleles Rechnen

In Vorbereitung des parallelen Rechnens waren bereits Studien an einem Cluster von Rechnern HP 9000/700 durchgeführt worden.

Zu sondierenden Rechnungen für die Anschaffung eines Parallelrechners wurde ein Bezugsjob definiert, der die rechenintensivsten Programme mit dem gegenwärtig höchsten Verbrauch von Rechenkapazität an den Zentralen Servern enthält (siehe auch Abschnitt 9). Wie sich dabei zeigte, sind die z. T. über viele Jahre gewachsenen, an keiner speziellen Rechnerarchitektur orientierten Anwendungen nur in sehr beschränktem Maße parallelisierbar.

Um die Architektur moderner Parallelrechner maximal nutzen zu können, muß bei neu entstehenden Programmen die Parallelisierung bereits in der Phase der Programmkonzipierung Berücksichtigung finden. In dieser Phase ist die Unterstützung durch Software-Experten der Abteilung notwendig.

Ein anderer Weg, Parallelarchitekturen zu nutzen, führt über die Anwendung parallelisierter Algorithmen, die in Bibliotheken enthalten sind, bzw. über die Nutzung parallelisierender Compiler.

Für die Bearbeitung "grob granulierter" Probleme, d. h. von Aufgaben mit langen parallel ausführbaren Abschnitten und geringer Kommunikation zwischen diesen, ist noch immer die Aufgabenverteilung auf der Basis eines Load-levelings auf mehreren gekoppelten Servern attraktiv.

Das Problem erhält eine weitere Dimension, wenn Server unterschiedlicher Architektur und Leistung zu einem virtuellen Cluster vereinigt werden. Entsprechende Untersuchungen fanden im Rahmen einer Studie statt, über die im Abschnitt 9 berichtet wird.

Konkrete Unterstützung erhielt das Institut für Sicherheitsforschung bei der Parallelisierung eines Nutzerprogramms.

## 6.4 Visualisierung

Die Visualisierung von Daten hat weiterhin an Bedeutung gewonnen und sich zu einem wichtigen Dienst der Abteilung entwickelt. Er wurde auf hohem Niveau fortgeführt und um entscheidende Neuerungen erweitert.

Über das Drucken von Dokumenten und Folien am A3-Color-Drucker Canon CLC 700 in nahezu photorealistischer Qualität war bereits im Abschnitt 4.4 berichtet worden.

An den Zentralen Servern werden die High-end-Visualisierungssysteme AVS (Advanced Visualization System, Fa. UNIRAS) und IDL (Interactive Data Language) zur Nutzung bereitgestellt und gepflegt. Das äußerst leistungsfähige, aber ressourcenintensive AVS (Version V5.02) bildet die Grundlage für die Erzeugung von Videosequenzen. Darüber hinaus wird es für sehr anspruchsvolle Anwendungen genutzt. 1997/98 wird eine PC-Version (V6) erscheinen, mit der das AVS eine breitere Anwendung auch am Forschungsstandort finden wird.

Die Nutzung von IDL hat am Standort spürbar zugenommen. Das FZR verfügte Ende 1996 über 11 Lizenzen der Version V.4.0.

1995/96 kam GSharp als weiteres Visualisierungssystem hinzu. Es wurde in Studien und Testanwendungen erprobt. GSharp (Fa. UNIRAS) ist ein gegenüber AVS vereinfachtes, kommandoorientiertes System. Es eignet sich gut für die Erzeugung von 1- und 2-D-Graphiken. Es verfügt über Driver für hochauflösende Farbdrucker.

Auch die Anwender weniger anspruchsvoller Systeme, wie gnuplot, erhielten durch die Abteilung Unterstützung.

Spitzentechnologie der Visualisierung am Campus ist die Erzeugung von Videos auf professionellem Qualitätsniveau. Sie hat zusammen mit der Simulation wissenschaftlich-technischer Prozesse rasch eine hohe Bedeutung erlangt. Anforderungen kamen vor allem aus dem Institut für Ionenstrahlphysik und Materialforschung.

Hierzu steht seit 1996 eine Workstation HP 9000/715 (ikon), die mit einer PARALLAX-Karte bestückt ist, zur Verfügung. Die mitgelieferte Software mußte im Rahmen zweier Diplomarbeiten vervollständigt werden. Dies betraf vor allem die Erzeugung von M-JPEG-Files und deren Management. Seit Herbst 1995 existiert in der Abteilung eine komplette, professionell arbeitende Linie zur Herstellung von Videosequenzen. Mit dieser wurden 1996 Erfahrungen gesammelt und erste Videos hergestellt.

Die weiteren Arbeiten galten der Verfeinerung dieser Technologie. Es wurden Werkzeuge zur automatisierten Herstellung von Bildwechselfolgen entwickelt sowie Studien zur Problematik Szene-Kamera-Rotation-Positionsveränderungen durchgeführt. Die Bildwechseltechniken beinhalten Stillbilder, Überblendungen, Mischen von Bildern und Rolltexte. Eingeschlossen sind die Erzeugung formatierter Texttafeln aus ASCII-Text und die Konvertierung der resultierenden Sequenzen in M-JPEG-Files.

Insgesamt wurden bisher drei Nutzervideos hergestellt.

## 6.5 Textverarbeitung

Als anspruchsvolles Textsatzsystem steht  $\LaTeX$ ( $\TeX$ ) seit 1995 in seiner neuen erweiterten Version  $\LaTeX_{2\epsilon}$  ( $\text{te}\TeX$ -Distribution) an den Zentralen Servern zur Verfügung. Zur  $\text{te}\TeX$ -Distribution gehören auch Dokumentklassen und Pakete zum Publizieren in führenden wissenschaftlichen Verlagen, z. B.

- American Mathematical Society  $\mathcal{AMS}$
- Elsevier Science Publishers B.V.,

und in USA-Zeitschriften, wie

- American Physical Society (*Physical Review A, B, C, D, E or Letters*)
- American Institute of Physics (18 Zeitschriften), Optical Society of America.

Für das Publizieren in den Zeitschriften des Springer-Verlags wurden spezielle  $\LaTeX$ - und  $\TeX$ -Formatfiles mit Springer-Makropaketen installiert.

Auch die kyrillische  $\LaTeX_{2\epsilon}$ -Version ist als zusätzlich installiertes spezielles Formatfile verfügbar.

Im Zeitraum 1995/96 wurden folgende  $\LaTeX$ - und Graphik-Werkzeuge gepflegt, an weiteren Zentralen Servern installiert, bzw. um neue Komponenten ergänzt:

- `latex2html` – Konverter, der aus einem  $\LaTeX$ - ein analoges HTML-File mit Ausnahme einiger  $\LaTeX$ -Konstruktionen erzeugt
- `Ghostscript` – X11-basierter PostScript-Previewer V4.0 mit Funktionserweiterungen
- `Ghostview` – erweiterte Bedienoberfläche für Ghostscript
- `xtem` – eine komfortable Benutzeroberfläche für die  $\LaTeX$ ( $\TeX$ )-Umgebung

- LyX– Benutzeroberfläche, die das WYSIWYG–Prinzip bei der Erstellung von  $\LaTeX$ –Dokumenten realisiert,
- xv – X11-basierter Previewer zur Darstellung und Konvertierung von Abbildungen in verschiedenen Graphikformaten
- xfig – Programm zur interaktiven Erzeugung graphischer Abbildungen unter X11
- xmgr – Graphikprogramm zum Plotten von Datenfiles und math. Funktionen
- mpeg\_play – X11-basierter Previewer für Videos
- ImageMagick – Softwarepaket zur Darstellung und Konvertierung von Abbildungen in verschiedenen Graphikformaten.

Aufgaben der  $\LaTeX$ –Betreuung im FZR waren weiterhin die primär auf den Nutzer ausgerichteten Aktivitäten. Von nicht minderer Bedeutung und mit spürbarem Aufwand verbunden ist das Bemühen, den Überblick über das sich sehr stürmisch und vielseitig entwickelnde Gebiet der Textverarbeitung zu bewahren, und eine FZR–spezifische, nutzergerechte Auswahl von bereitzustellender Software zu treffen.

## 6.6 Informationsdienste

Das FZR ist seit 1994 im Internet durch einen eigenen WWW–Server (CERN, Version 3.0) an der C3220 (hera) präsent. Bei der Pflege des Servers und der Dokumente wurde eine Arbeitsteilung eingeführt:

- Für den Betrieb und die Pflege des WWW–Servers ist die Abteilung FVTK verantwortlich.
- Die inhaltliche Gestaltung obliegt einer Redaktionsgruppe aus Vertretern der Organisationseinheiten im FZR.

Die Statistik des WWW–Servers registrierte im zweiten Halbjahr 1996 monatlich 14.206 Zugriffe durch 1744 Anforderer (Mittelwert).

Als Browser hat sich am Standort neben Mosaic und lynx netscape durchgesetzt.

## 6.7 Benutzerseminare

Datum	Nr.	Referent	Thema
07.02.95	58	R. Fülle U. Fromm	Aufbau eines WWW-Servers für das FZR NetNews an der hera
21.02.95	59	E. Müller	AXIOM für Einsteiger (Hinweise zur Nutzung und zu Einsatzmöglichkeiten des Computeralgebra-Systems)
22.03.95	60	E. Müller	AXIOM für Einsteiger (Hinweise zur Nutzung und zu Einsatzmöglichkeiten des Computeralgebra-Systems) (Forts.)
27.06.95	61	R. Fülle U. Fromm M. Kunicke	Abschluß Rekonstruktion Rechnergebäude 13c Das neue Convex OS, Version 11.0 WWW
11.07.95	62	V. Kisperth (CONVEX) M. Gramlich (ATL Odetics)	Datenmanagementlösungen für wissenschaftliche Einrichtungen
12.09.95	63	M. Kunicke	Leistungsmessungen mit dem FZR-Meßjob Installation des Zentralen Servers SPP 1000/CD-4N
10.10.95	64	H. Böttger	Videoerstellung aus Computeranimationen (gegenwärtiger Stand)
14.11.95	65	E. Müller	Fortran an der vesta (SPP 1000-Rechner des FZR)
19.12.95 10 10	66	U. Hoffmann (Digital Dresden) Dr. J. Krebs (Digital München) R. Mehlhorn	Entwicklung der 64-Bit-Prozessortechnik von Digital: Aktuelle Server- und Workstation-Produkte Entwicklung des Alpha-Chips Entwicklungen für den Netzwerkbereich
30.01.96	67	H. Böttger	Zentrale Dienste zur Erstellung von Videosequenzen (gegenwärtiger Stand)
12.03.96	68	H. Böttger	Graphische Darstellungen mit Gsharp - Vorstellung eines neuen Visualisierungssystems
21.05.96	69	G. Vinel	L <sup>A</sup> T <sub>E</sub> X2e - Einführung
10.09.96	70	H. Böttger	Die Gestaltung von Videovorspann und -abspann, Vorstellung neuer Werkzeuge (mit Vorführungen und Videobeiträgen)
24.09.96	71	M. Schlett	Versenden und Empfangen von E-Mail, Programme und Techniken für Windows und UNIX
22.10.96	72	H. Kirsch	UniTree- Ein Archivierungssystem für große Datenmengen
05.11.96	73	E. Müller	Einführung in das Computeralgebra-System Maple

## 7 Wissenschaftliche Kooperation

Die Abteilung ist um fachliche Kontakte zu gleichen Einrichtungen im In- und Ausland bemüht. Mitarbeiter der Abteilung sind aktive Mitglieder folgender Gremien und Arbeitsgruppen:

- DFN-Verein:  
Dr. R. Fülle (im Auftrage des FZR)
- Arbeitskreis EDV der WBL:  
Dr. R. Fülle (stellv. Vorsitzender)
- Koordinierungsausschuß für Datenverarbeitung (KODA) der HGF:  
Dr. R. Fülle (als Gast)
- Arbeitsgruppe Graphische Datenverarbeitung der HGF:  
H. Böttger (als Gast)
- DANTE (Deutsche L<sup>A</sup>T<sub>E</sub>X-Nutzergruppe):  
Dr. G. Vinel
- DIGI (Deutsche Interessengemeinschaft Internet):  
Dr. W.-J. Linnemann, Dr. M. Schlett
- Deutsche CONVEX-Nutzergruppe:  
Dr. M. Kunicke (Vorsitzender)
- Europäische CONVEX-Nutzergruppe:  
Dr. M. Kunicke
- NAG User Association:  
E. Müller

Die Zusammenarbeit mit dem Vereinigten Institut für Kernforschung (VIK) Dubna (Rußland), Laboratorium für Rechentechnik, wurde fortgesetzt. 1996 arbeiteten drei Mitarbeiter des VIK als Gastwissenschaftler an Aufgaben der Abteilung mit:

- Victor Krylov:  
Gemeinsame Erarbeitung einer Pilotanwendung für die Video-Animation von wissenschaftlichen Daten;  
September – Dezember 1996
- Albert Rybakin:  
Entwurf und Einrichtung eines Gateways für einen ORACLE-WWW-Server;  
Oktober – Dezember 1996
- Michael Popov:  
Aufbau eines Fontservers für das FZR;  
November – Dezember 1996

Verstärkt beteiligte sich die Abteilung an der Ausbildung von Studenten:

- Vorlesungen und Übungen an der TU Dresden,  
Fakultät Mathematik und Naturwissenschaften:  
“Rechentechische Hilfsmittel des Physikers“  
(Doz. Dr. R. Fülle)
- Seminarvorträge an der TU Dresden,  
Fakultät Mathematik und Naturwissenschaften sowie  
Fakultät Elektrotechnik:  
“Einführung in das Parallele Rechnen & Message-passing“,  
“Shared-memory-Programmierung“  
(Dipl.-Math. N. Schmeißer)
- Betreuung von Diplom-, Beleg- und Praktikumsarbeiten
  - Entwicklung von Visualisierungen und deren Umsetzung in Computeranimationen und Videosequenzen zur Darstellung evolutionärer physikalischer Prozesse mit Hilfe des AVS;  
Diplomarbeit von Dirk Engwicht, TU Dresden, Fakultät Informatik (1995),  
Betreuer im FZR: H. Böttger, Dr. K.-H. Heinig, Dr. M. Kunicke
  - Erweiterung der Funktionalität des Visualisierungssystems AVS durch Entwicklung neuer Module zur Lösung wissenschaftlicher Visualisierungsaufgaben am FSR;  
Diplomarbeit von Thomas Frei, TU Dresden, Fakultät Informatik (1995),  
Betreuer im FZR: H. Böttger, Dr. K.-H. Heinig, Dr. M. Kunicke
  - Analyse eines Workstationclusters und eines Transputersystems;  
Großer Beleg von Thomas Loose, TU Dresden, Fakultät Informatik (1995),  
Betreuer im FZR: Dr. M. Kunicke
  - Zufallszahlengenerator auf dem PowerXplorer;  
Beleg von Andreas Hähne, TU Dresden, Fakultät Informatik (1995),  
Betreuer im FZR: Dr. M. Kunicke
  - Die Nutzung moderner Kommunikationstechnologien am Forschungsstandort Rossendorf – Untersuchung von Varianten am Beispiel der medizinischen Kommunikations- und Datenverarbeitungsprozesse;  
Diplomarbeit von Thomas Bach, TU Dresden, Fakultät Informatik (1996),  
Betreuer im FZR: Doz. Dr. R. Fülle
  - Analyse und Test der Kommunikationsbibliothek MPI;  
Großer Beleg von Andreas Hähne, TU Dresden, Fakultät Informatik (1996),  
Betreuer im FZR: Dr. M. Kunicke

Die Abteilung betreute ferner 3 Auszubildende (Bürokauffrauen) bei der Einarbeitung in WordPerfect und dBase.

## 8 Konferenzbeiträge und Vorträge

Fülle, R.:

Technologien und Anwendungen von Datennetzen,  
eine Einführung;

TUD, Klinik für Nuklearmedizin, Dresden, November 1995

Böttger, H.:

Video-Vorspann und -Abspann mit AVS;

AVS/UNIRAS-Benutzertreffen, Neuss, Oktober 1996

Fülle, R.:

Neue Netze;

Arbeitskreis EDV der WBL, Berlin, Oktober 1996

Linnemann, W.-J.:

Telearbeitsplätze – Datenschutzprobleme bei der Einwahl in Firmennetze;

37. Sitzung der Datenschutzbeauftragten der HGF, DLR Oberpfaffenhofen, Oktober 1996

## 9 Ausführliche Berichte zu ausgewählten Themen

- Helga Kirsch: Einsatz von UniTree zur Archivierung großer Datenmengen
- Erhard Müller: Computeralgebra-Systeme an Zentralen Servern des FZR
- Manfred Kunicke: Leistungsmessungen mit dem FZR-Meßjob
- Nils Schmeißer: PVM am FZ Rossendorf
- Horst Böttger: Neue Hilfsmittel für die Herstellung von Videos
- Galina Vinel:  $\text{\LaTeX}2_{\epsilon}$  – die wichtigsten Merkmale
- Thomas Bach,  
Ruprecht Fülle,  
Reiner Hliscs: Erschließen des Telekonferenz-Systems ProShare

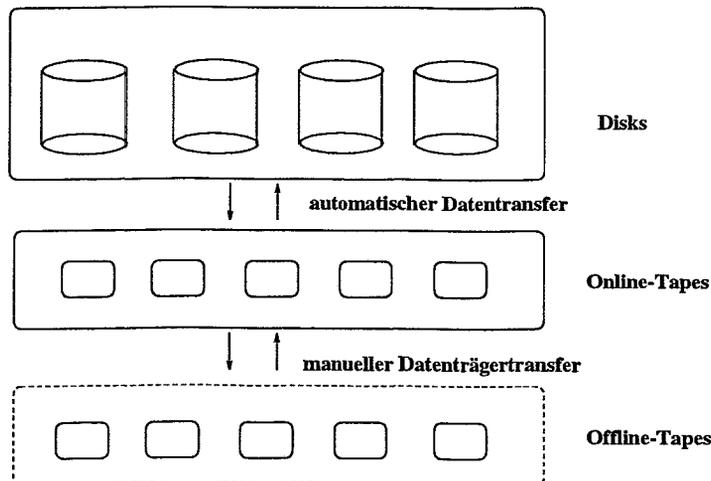
# Einsatz von UniTree zur Archivierung großer Datenmengen

Helga Kirsch

## 1 Allgemeines

Zur Archivierung großer Datenmengen gibt es spezielle Archivierungssysteme, die mit einer hierarchischen Speicherstruktur arbeiten. Ein solches System besitzt die Fähigkeit, Dateien von einem Speichermedium zu einem anderen zu transportieren. Dadurch vermittelt es dem Nutzer den Eindruck eines "unendlich" großen Filesystems. Magnetplatten (Disk-Cache) stellen die höchste Ebene dar, darunter liegt die Ebene der Online-Bänder, während Offline-Bänder die niedrigste Ebene repräsentieren. Online-Bänder befinden sich in einem automatischen Bandarchiv, das durch einen Roboter gesteuert wird.

hierarchische Speicherstruktur



Als Archivierungssoftware steht im Rechenzentrum des FZR seit 1996 UniTree am Zentralen Server CONVEX C3820 (rhea) zur Verfügung. Das zugehörige Robotersystem ist ein Produkt der Firma Odetics Company vom Typ ACL6/176 mit 6 DLT-Laufwerken (DLT - digital linear tape) und einem Fassungsvermögen von 176 DLT-Bändern mit einer Kapazität von 10 GByte pro Band.

## 2 Arbeitsweise von UniTree

UniTree steuert die Verlagerung der Files von einer Ebene zur anderen mittels einer Vielzahl spezieller Prozesse, um folgende Vorgänge zu realisieren:

**Migration:** Kopiervorgang vom Disk-Cache zu niedrigerer Ebene in regelmäßigen Zeitintervallen oder bei Erreichen eines bestimmten Füllstandes des Disk-Caches.

Das File bleibt im Disk-Cache noch erhalten.

**Purging:** Löschen von migrierten Files aus dem Disk-Cache bei Erreichen eines vom Administrator festgelegten oberen Füllstandes (high water mark), bis unterer Füllstand (low water mark) erreicht ist. Files, die am längsten nicht genutzt worden sind, werden zuerst gelöscht.

**Staging:** Kopieren eines Files von einer niedrigeren Ebene in den Disk-Cache.

UniTree verfügt über eine spezielle Einrichtung zur zeitweiligen Aufbewahrung gelöschter Files, den sogenannten Trash Can. Dadurch ist ein Wiederherstellen versehentlich gelöschter Files schnell und unproblematisch möglich.

UniTree unterstützt die Zuordnung von Files zu verschiedenen Familien (Family-Konzept), z. B. Nutzergruppen oder Aufgaben. Files einer Familie werden auf Bänder gespeichert, die nur für diese Familie reserviert sind.

Durch UniTree archivierte Files sind nur im Zusammenhang mit ihren Metadaten, die durch einen Nameserver, eine Tape Header Data Base und eine Tape Header Search Table verwaltet werden, wiederauffindbar. Deshalb können von UniTree erzeugte Banddateien nur in dem UniTree-System gelesen werden, in dem sie archiviert wurden. Ein Transfer von solchen Bändern zu einem anderen Rechner ist nicht möglich.

### 3 UniTree an der rhea

Voraussetzung für die Arbeit eines Nutzers mit UniTree ist ein Account an den Zentralen Servern hera/rhea. Für den Nutzer gibt es zwei Wege, um seine Files von UniTree archivieren zu lassen:

Er kann sie von seiner Workstation oder seinem PC mittels ftp zum Disk-Cache des UniTree übertragen, oder er kopiert sie in ein spezielles Filesystem auf der rhea, das durch den virtuellen Diskmanager (CVDM – Convex Virtual Disk Manager) verwaltet wird.

Archivierte Files können nur über den Archivierungsweg zurückgeholt werden.

In UniTree steht ein spezielles ftp zur Verfügung, das neben den bekannten ftp-Kommandos noch einige zusätzliche Kommandos bietet.

Die Verbindung zum ftp des UniTree-Servers erfolgt über den Port 1024 mit

<b>ftp rhea 1024</b>	von einer Workstation,
<b>ftp -p 1024 rhea</b>	von einem PC unter Windows/Windows95,
<b>ftp</b>	von einem PC unter WindowsNT und dann vom ftp-Prompter aus <b>open rhea 1024</b> .

Nach Eingabe des Nutzernamens und des Paßworts befindet sich der Nutzer in seinem UniTree-Homedirectory, das zur Aufnahme der zu archivierenden Files dient. Sie können mittels der bekannten ftp-Kommandos übertragen werden. Die Migration der Files ins Bandarchiv erfolgt nach 5 Minuten automatisch. Außerdem bleiben die migrierten Files im Disk-Cache solange erhalten, bis die Kriterien des Purging für sie erfüllt sind. Der Nutzer kann sich durch die Anzeige des Direktorys darüber informieren, ob sich seine Files noch im Disk-Cache (Anzeige DC) oder nur noch im Bandarchiv (Anzeige AR) befinden.

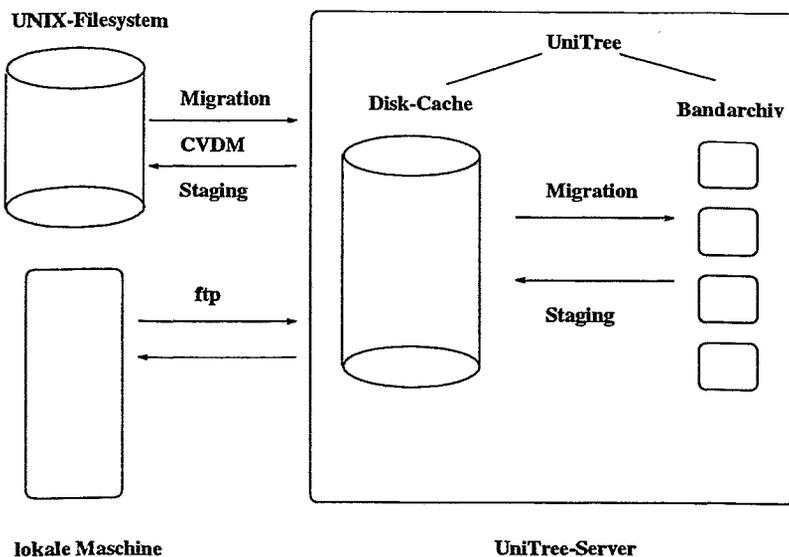
Im UniTree-Homedirectory des Nutzers wird von UniTree automatisch ein Subdirectory `.trash` angelegt. Löscht der Nutzer ein File auf dem UniTree-Server, wird es zunächst in das `.trash`-Directory kopiert, wobei der Filename aus Gründen der Eindeutigkeit durch Datum, Uhrzeit und einen Zähler ergänzt wird.

Die Verweildauer eines Files im Trash Can wird vom Administrator global festgelegt und kann vom Nutzer verändert werden. Nach Ablauf dieser Verweildauer wird das File aus dem Trash Can automatisch gelöscht.

#### 4 CVDM – Convex Virtual Disk Manager

Ein zweiter Weg zur Archivierung von Files führt über das UNIX-Filesystem `/archiv` auf der `rhea`. Der Nutzer kann seine Files in diesem Filesystem mit den üblichen UNIX-Kommandos behandeln. Die eigentliche Archivierung, nämlich die Verlagerung der Files aus diesem UNIX-Filesystem (primärer Speicher) zum UniTree-Server (sekundärer Speicher), wird vom virtuellen Disk-Manager gesteuert.

Die bereits von UniTree bekannten Begriffe Migration, Staging und Purging haben hier ähnliche Bedeutung, beziehen sich jedoch auf die Verlagerung der Files aus dem Primär- zum Sekundär-Speicher und umgekehrt bzw. auf das Löschen von Files aus dem UNIX-Filesystem nach erfolgreicher Migration. Sobald sich die Files im Disk-Cache des UniTree-Servers befinden, unterliegen sie der UniTree-Steuerung.



Im CVDM existiert ebenfalls ein Trash-Can-Mechanismus, der dadurch aktiviert wird, daß der Systemadministrator ein Directory `.trash` anlegt. Sobald ein Nutzer zum ersten Mal ein migriertes File löscht, wird im `.trash`-Directory ein Subdirectory mit dem Nutzernamen angelegt, in das dann jeweils die gelöschten Files gelangen.

Die praktische Erfahrung mit CVDM zeigt, daß es empfehlenswert ist, zum Archivieren gesamter Directories `tar`-Files zu verwenden. Dadurch wird die Zugriffszeit beim Zurückholen aus dem Archiv deutlich verringert.

	UniTree	CVDM
Voraussetzung	Account an hera/rhea	
Verbindung herstellen – von einer Workstation – von einem PC	ftp rhea 1024 ftp -p 1024 rhea bzw. ftp und open rhea 1024	telnet   rlogin rhea tn rhea (systemabhängig)
Files befinden sich in	/home/username	/archiv/username
Trash Can	/home/username/.trash	/archiv/.trash/username
Verweilzeit im Trash Can	24 Stunden oder individuell einstellbar	24 Stunden
Filetransfer	ftp-Kommandos	UNIX-Kommandos
Migration	automatisch nach 5 Minuten	bei Erreichen der high water mark oder mit migout

# Computeralgebra-Systeme an Zentralen Servern des FZR

Erhard Müller

## 1 Einführung

Schon seit einigen Jahren nutzen einzelne Abteilungen des FZR verschiedene Computeralgebra-Systeme, wie Mathematica, Maple, Mathcad und Matlab, zur Lösung ihrer speziellen mathematischen Probleme. Die an den Workstations oder Personalcomputern der Abteilungen installierten Systeme stehen nicht für die allgemeine Nutzung im FZR zur Verfügung. Um die vielen Einsatzmöglichkeiten der Computeralgebra-Systeme (s. 2.) allen Mitarbeitern des FZR zu erschließen, wurden 1995 und 1996 die Systeme AXIOM und Maple an Zentralen Servern eingerichtet. Bei der Auswahl der Systeme war entscheidend, daß AXIOM einen bequemen Zugriff zu der an mehreren Zentralen Servern installierten NAG Fortran Library ermöglicht und daß andererseits Maple an den meisten Hochschulen, Universitäten und größeren Forschungseinrichtungen seit mehreren Jahren genutzt und als sehr anwenderfreundlich eingeschätzt wird.

An den Zentralen Servern sind folgende AXIOM- und Maple-Versionen installiert:

AXIOM, Release 2.1    an rita (IBM RS/6000 Mod. 550),  
Maple V, Release 4    an rike (IBM RS/6000 Mod. 591).

## 2 Einsatzmöglichkeiten von Computeralgebra-Systemen:

- symbolische Lösung mathematischer Probleme
- numerische Lösung mathematischer Probleme mit hoher Genauigkeit
- Erzeugen von 2D- und 3D-Grafik, z. T. auch von Animationen
- Nutzung systemeigener Programmiersprachen und umfangreicher mathematischer Bibliotheken zur Erstellung eigener Programme in der jeweiligen Systemsprache
- Erzeugen von Fortran- und C-Quellcode,  $\LaTeX$ - und PostScript-Output.

Die Computeralgebra-Systeme wurden vorrangig für das symbolische Rechnen und das numerische Rechnen mit hoher Genauigkeit entwickelt. Ihre Einsatzmöglichkeiten auf dem Gebiet der Grafik sind für viele Anwendungen ausreichend. Große Grafik-Systeme, wie AVS oder IDL, haben allerdings ein breiteres Leistungsspektrum.

Die modernen Computeralgebra-Systeme beinhalten umfangreiche Online-Dokumentationen, die der Anwender bequem einsehen kann.

### 3 Nutzung von AXIOM

AXIOM wurde unter Federführung der Firma IBM als Nachfolgesystem von Scratchpad entwickelt. Die erste Version von AXIOM kam 1991 auf den Software-Markt. Das Copyright für AXIOM wurde der Firma NAG (The Numerical Algorithms Group Ltd) übertragen.

Im FZR wurde AXIOM bisher vorrangig genutzt für:

- Differentiation und Integration,
- Matrizenprobleme, u. a. für Ermittlung von Eigenwerten und Eigenvektoren,
- grafische Darstellungen, z. B. für Dichteverteilungen von Protonen und Neutronen mit hohem Drehimpuls im Kern.

Ab Release 2.0 von AXIOM besteht die Möglichkeit des Zugriffs auf viele Routinen der NAG Fortran Library, für die das HyperDoc-System von AXIOM vorbereitete Schablonen (Templates) bereitstellt. Der Anwender kann diese auf einfache Weise seiner Problemstellung anpassen. In der jetzigen aktuellen Version 2.1 von AXIOM existieren für mehr als 150 Routinen aus 14 Kapiteln der Fortran-Bibliothek derartige Schablonen.

Der Zugriff auf eine Schablone für ein spezielles Unterprogramm geschieht auf folgende Weise:

- Anklicken von "Access the Link from HyperDoc" im NAG-Link-Menü
- Anklicken des betreffenden Kapitels, z. B. D01, Quadraturen
- Anklicken der speziellen Routine, z. B. D01FCF (Routine für mehrdimensionale adaptive Quadratur über einem Hyper-Rechteck mit konstanten endlichen Grenzen).

Die angezeigte Schablone enthält eine kurze Beschreibung der Funktion der Routine und i. allg. bereits ausgefüllte Eingabefelder für die Parameter. Die vorbereiteten Beispiele sind identisch mit den Beispielen, die in der Dokumentation zur NAG Fortran Library aufgeführt sind.

Für Routinen ohne Schablonen im HyperDoc-System besteht die Möglichkeit des Direktauf-rufs mittels des AXIOM-Interpreters (s. Anwendungsbeispiele in [1]).

### 4 Nutzung von Maple V

Maple ist wie AXIOM ein leistungsfähiges interaktives Computeralgebra-System, das sich vorrangig für die symbolische und auch numerische Lösung von mathematischen Problemen eignet. Bezüglich der Grafikerunterstützung ist hervorzuheben, daß Maple auch Animationen für 2D- und 3D-Grafiken ermöglicht. Neben der systemeigenen mathematischen Bibliothek mit mehr als 2500 Funktionen kann ein Nutzer zusätzlich die Maple Share Library nutzen, die von Anwendern entwickelte Routinen und Funktionspakete aus verschiedenen Anwendungsgebieten enthält.

Maple wurde an der Universität Waterloo, Kanada und an der ETH Zürich entwickelt. Seit 1985 werden Maple-Implementationen für die wichtigsten Hardwareplattformen und Betriebssysteme ausgeliefert.

Beim Aufruf von Maple wird der Maple-Kern (ca. 10 % des Maple-Systems) geladen. Er enthält neben dem Sprachinterpreter die wichtigsten Algorithmen für Integer- und rationale Arithmetik und polynomiale Berechnungen sowie Ein- und Ausgaberroutinen. Viele weitere oft benötigte Algorithmen befinden sich in der Maple Main Library. Sie werden bei Bedarf automatisch geladen. Zusätzliche Bibliotheksrountinen sowie Funktionen und Pakete der Maple Share Library müssen vom Anwender explizit mit Hilfe des with-Kommandos geladen werden. Um z. B. auf Routinen des Statistikpaketes und der Maple Share Library zugreifen zu können, müssen die Kommandos "with(stats);" und "with(share);" abgesetzt werden.

Im FZR wurde Maple u. a. zur Lösung von Transportproblemen im relativistischen Geschwindigkeitsbereich, zur Berechnung von Magnetspulen (Variationsproblem) und für die Lösung von Problemstellungen der Quantenmechanik genutzt.

Die anstehenden Aufgaben konnten durch die Möglichkeiten des symbolischen Rechnens (z. B. zur Differentiation, Integration und zur Berechnung von Eigenwerten) sowie des numerischen Rechnens schnell und mit hoher Genauigkeit gelöst werden.

Für quantentheoretische Berechnungen wurde ein eigenes Maple-Paket zur Bestimmung von Clebsch-Gordan-Koeffizienten, Wigner-6j-, Wigner-9j-Symbolen und Racah-W-Koeffizienten entwickelt und bereitgestellt. Dabei wurde ein seit Jahren im FZR bewährtes Fortran-Quellprogramm in die Maple-Sprache umgeschrieben und in den internen Maple-Kode überführt (s. [2]). Die Maple Share Library enthält ein ähnliches Paket, auf dessen Routinen nach Absetzen der Kommandos "with(share);" und "readshare(Wigner,science);" zugegriffen werden kann. Mit Hilfe dieses Paketes können neben Wigner-6j- und Wigner-3j-Symbolen (anstelle von Clebsch-Gordan-Koeffizienten) auch  $d_{lmn}(\beta)$ -Koeffizienten berechnet werden.

## 5 Vergleichende Betrachtungen zu den Computeralgebra-Systemen AXIOM und Maple

Es ist problematisch, nach relativ kurzer Zeit seit Installierung der Systeme einen exakten Vergleich hinsichtlich Leistungsfähigkeit und Nutzerfreundlichkeit der verschiedenen Computeralgebra-Systeme zu liefern.

AXIOM zeichnet sich neben einem sehr breiten Einsatzspektrum besonders dadurch aus, daß es sich eng an die Entwicklung der modernen Algebra anlehnt und über eine streng typisierte Sprache verfügt, wobei der Sprachinterpreter so ausgestattet ist, daß für den Anwender Typdeklarationen weitgehend minimiert werden. Noch erforderliche Typvereinbarungen werden von manchen Anwendern als hinderlich angesehen.

Für AXIOM-Installationen auf Rechnern, an denen auch die NAG Fortran Library verfügbar ist, ist der bequeme Zugang von AXIOM zu den bewährten Routinen der Fortran-Bibliothek sehr vorteilhaft. Dadurch wird der Leistungsumfang des Systems (auch verglichen mit anderen Computeralgebra-Systemen) wesentlich erweitert.

Bei Maple benötigt der Anwender i. allg. weniger Einarbeitungszeit als bei AXIOM. Das Überführen von Programmen aus problemorientierten Sprachen, z. B. Fortran, C, Pascal, in die Maple-Sprache kann, geringe Programmiererfahrungen für diese Sprachen vorausgesetzt, in relativ kurzer Zeit bewältigt werden.

Hinsichtlich der 2D- und 3D-Grafik ist von Vorteil, daß Animationen in beschränktem Umfang realisierbar sind.

Ein Vorteil von Maple ist auch darin zu sehen, daß es an den meisten Universitäten und Hochschulen in Deutschland (und weltweit) als Werkzeug zur Verfügung steht und dadurch viele junge Wissenschaftler in Forschungszentren wie dem FZR über Maple-Kenntnisse verfügen. Dieses gilt auch für das Computeralgebra-System Mathematica. Eine Installation von Mathematica auf den Zentralen Servern des FZR ist geplant. Durchgeführte Tests zeigten, daß Mathematica hinsichtlich der Integration noch leistungsfähiger als AXIOM und Maple ist.

## Literatur

- [1] Müller, E.: AXIOM für Einsteiger, 1. Fortsetzung;  
Benutzer-Seminar der Abteilung Kommunikation und Datenverarbeitung, 22.03.1995
  
- [2] Müller, E.: Einführung in das Computeralgebra-System Maple;  
Benutzer-Seminar der Abteilung Kommunikation und Datenverarbeitung, 05.11.1996  
und  
WWW-Seiten des FZR

# Leistungsmessungen mit dem FZR-Meßjob

Manfred Kunicke

Die Anforderungen an die Leistungsfähigkeit der zentralen Rechner server steigen kontinuierlich. Um eine gewisse Sicherheit bei der Beschaffung neuer Rechner zu erreichen, muß man das Preis-Leistungs-Verhältnis für die als zentrale Rechner server in Frage kommenden Rechner abschätzen können. Es bedarf also einer Leistungsmessung im Vorfeld einer Beschaffung.

Die Rechenleistung von Rechnern unterschiedlicher Architekturen ist von der Art der zu rechnenden Programme abhängig, d. h. das Preis-Leistungs-Verhältnis hängt vom Spektrum der Kundenjobs ab. Es wäre also falsch, die von den Firmen angegebenen Spitzenrechenleistungen als Basis für die Abschätzung von Preis-Leistungs-Verhältnissen zu nehmen. Verschiedene verbreitete Benchmarks, z. B. der LINPACK-Benchmark, kommen den Anforderungen schon näher. Besser ist es jedoch, einen Meßjob zur Leistungsmessung zu benutzen, der dem jeweiligen Kundenjobspektrum entspricht. Aus diesem Grunde wurde für das FZR ein eigener Meßjob zusammengestellt, der

- repräsentativ für die Last an den Zentralen Servern und
- übertragbar auf verschiedene Rechnersysteme ist.

Ausgangspunkt für die Zusammenstellung des Meßjobs war eine Lastanalyse am Zentralen Server CONVEX C3820 (rhea) im Zeitraum August 1993 bis November 1994:

Jobklasse	Anzahl der Aufgaben	Rechenzeitanteil
express	2.160.009	3,9 %
short	2.328	4,8 %
long	2.923	16,3 %
verylong	657	25,2 %
unlimited	250	49,8 %

Ausgehend von dieser Lastanalyse, wurden für den FZR-Meßjob vier Kundenjobs ausgewählt, die

- repräsentativ für die Last am Zentralen Server rhea und
- skalierbar in der benötigten Rechenzeit sind (der Meßjob sollte nicht zu lange rechnen) sowie
- als Fortranquellprogramme vorliegen und nur allgemein verbreitete Bibliotheken benutzen (als Bedingung für die Übertragbarkeit auf verschiedene Rechnersysteme).

Jobname	Anteil an der Gesamtrechenzeit	Wichtung	Rechenzeit als Teil des Meßjobs auf rhea (C3810, fc -O2)
COUL	11,8 %	0,28	77 s
SIM1	9,2 %	0,22	324 s
BUU	16,4 %	0,33	69 s
TRAMOR	4,6 %	0,11	210 s
Meßjob	42,0 %	1,00	148,0 s (gewichtet)

Bei Leistungsmessungen an verschiedenen Rechnersystemen geht die gemessene Rechenzeit jedes der vier Teiljobs des Meßjobs mit der ihm zugeordneten Wichtung in die Berechnung der Rechenleistung ein, um das Jobprofil auf den Zentralen Rechenservern des FZR richtig widerzuspiegeln.

Leistungskennzahlen sind als relative Werte aussagekräftiger als Absolutwerte. Zur Bestimmung der Leistung von Rechnersystemen setzen wir die Rechenleistung, die ein Prozessor des Zentralen Servers C3820 (rhea) mit dem Meßjob erbringt, auf "1". Die für den Meßjob gemessene und berechnete relative Leistung RL ergibt sich zu

$$RL = 1 / \text{Summe}_i ( W_i / LE_i )$$

mit

$W_i$  - Wichtung des i-ten Teiljobs,

$LE_i$  - gemessene relative Leistung des Rechnersystem mit dem i-ten Teiljob.

#### Ergebnisse der Messungen an FZR-Rechnern

Rechnersystem	Meßjob	COUL	SIM1	BUU	TRAMOR
C3810	1,00	1,00	1,00	1,00	1,00
SPP 1000	0,95	1,55	2,01	0,54	2,22
DEC 4000/710	1,23	0,98	2,29	1,03	1,58
IBM RS/6000 Mod. 550	0,61	0,36	0,96	0,66	0,79

# PVM am FZ Rossendorf

Nils Schmeißer

## 1 Die parallele virtuelle Maschine

PVM (*Parallel Virtual Machine*) ist ein System, bestehend aus Kommunikationsbibliotheken und einem Laufzeitsystem. Es verbindet in einem Netzwerk integrierte Rechner zu einem echten MIMD (Multiple Instruction Multiple Data) Parallelrechner mit einer heterogenen verteilten Architektur [1]. Die dazu benutzbare Klasse von Rechnern ist nicht auf Workstations eingeschränkt, sondern umfaßt auch Supercomputer und PCs. Insbesondere liefern fast alle namhaften Supercomputer-Hersteller eine PVM-Implementation für ihre Maschinen. PVM hat sich zum Quasi-Standard entwickelt.

PVM wurde am Forschungszentrum Rossendorf (FZR) bereits 1993/94 auf einem Cluster aus drei HP 9000 installiert. Mitte 1996 wurde der Cluster auf alle am Standort verfügbaren Zentralserver erweitert. Damit steht jetzt eine virtuelle parallele Maschine mit der folgenden Konfiguration zur Verfügung:

	Architektur	Rechenleistung/CPU <sup>0</sup>
CONVEX C3220	Vektorrechner mit 2 CPUs	100MFlop/s
CONVEX C3820	Vektorrechner mit 2 CPUs	480MFlop/s
2 HP 9000/735	1 Prozessor RISC Workstation	198MFlop/s
HP 9000/715	1 Prozessor RISC Workstation	26MFlop/s
IBM RS/6000 Mod. 550	1 Prozessor RISC Workstation	84MFlop/s
IBM RS/6000 Mod. 591	1 Prozessor RISC Workstation	274MFlop/s
CONVEX SPP 1000	4 Prozessor SHM Parall. Rechner	200MFlop/s
PC Pentium 166	1 Prozessor IBM PC komp.	62MFlop/s
	Virtuelle 14-Prozessor-Maschine	2.8GFlop/s

Die einzelnen Maschinen nennen wir auch Knoten des Clusters.

Zunächst wurde die Funktionsfähigkeit des Clusters mit dem in Abschnitt 2 vorgestellten Standardbeispiel aus der Mathematik getestet. Wie Testrechnungen über mehr als 2 Tage zeigten, läuft der Cluster stabil.

Der Cluster soll in erster Linie die Gesamtlaufzeit "time-to-solution" einer Anwendung durch parallele Abarbeitung verkürzen. Dabei müssen zwei Dinge beachtet werden:

1. Das Programm muß an die Architektur angepaßt werden.
2. Die Maschinen sollten zur Laufzeit der Applikation möglichst exklusiv zur Verfügung stehen.

Auf die Erfüllung des zweiten Punktes muß oft verzichtet werden.

<sup>0</sup>Die Angabe der Rechenleistung stützt sich auf Herstellerangaben. Für die IBM RS/6000 Mod. 591 wurde eine eigene Skalarprodukt-Benchmark verwendet, da keine Angaben von IBM verfügbar waren. Die LINPACK-Angaben nach Dongarra 1995 lassen eine realistischere Schätzung der Gesamtleistung auf 700MFlop/s zu.

Im vorliegenden Fall haben wir einen Cluster aus skalaren Maschinen (HP 9000, IBM RS/6000, IBM-PC) und parallelen Maschinen (CONVEX C-Serie und SPP 1000). Die Sicht auf die Architektur aller dieser Maschinen wird durch die darauf laufenden Betriebssysteme transparent gehalten. Wir haben dadurch keine Möglichkeit, außer unter Verwendung von direkten Systemaufrufen, auf den parallelen Maschinen eine Teilaufgabe (Thread) gezielt auf einem Prozessor zu starten.

PVM verteilt Threads "round-robin", d. h. die Knoten des Clusters erhalten solange reihum einen Thread zur Abarbeitung, bis alle zu startenden Threads laufen. Machen wir uns dies an einem Beispiel klar: Unser Cluster bestehe aus den beiden RS/6000 und der SPP 1000 (6 physische Prozessoren). Starten wir jetzt 6 Threads (Aufruf: "pvm> spawn -6 thread"), so stellen wir fest, daß jeder Knoten zwei Threads erhält. Das ist natürlich eine ungünstige Verteilung, da sich zwei Prozessoren der SPP 1000 im Leerlauf befinden und die CPUs der RS/6000-Maschinen je zwei Threads bearbeiten müssen. Starten wir nur 5 Threads, so kann es durchaus sein, daß die RS/6000 Mod. 550 zwei Threads und die RS/6000 Mod. 591 nur einen erhält. Auch das ist ungünstig, da die RS/6000 Mod. 591 eine etwa 3-mal höhere Rechenleistung als die RS/6000 Mod. 550 erbringt. Wir müssen also eine Möglichkeit finden, die Verteilung der Threads so zu steuern, daß wir eine (sub-)optimale Auslastung der virtuellen Maschine erhalten.

Im Konzept der Lastverteilung (Scheduling) kommen insbesondere die Verfahren des *Load-leveling* (a-priori-Erstellung eines Schedules) und des *Load-balancing* (dynamische Lastverteilung) zum Einsatz. Wir werden im folgenden die Möglichkeiten, die PVM hierfür bietet, darstellen und einen Load-leveler angeben.

### Maschinenkonfiguration

Die Erstellung eines Schedules setzt vorab stets die Kenntnis bestimmter Informationen über das System voraus. Das sind in einem heterogenen System die Anzahl, Architektur sowie die Leistungsfähigkeit der Knoten. Eigentlich müssen wir auch die Kommunikationsleistung zwischen den Knoten mit betrachten, wir wollen aber zunächst darauf verzichten und verweisen für genauere Betrachtungen zu diesem Thema auf [5], [8].

Da PVM neben den Konzepten des Nachrichtenaustausches auch Mechanismen zur Steuerung der virtuellen Maschine implementiert, können wir einen großen Teil der benötigten Informationen während der Laufzeit vom System erfragen. Zunächst bekommen wir von PVM durch die Funktion "pvm\_config" die aktuelle Konfiguration der virtuellen Maschine geliefert. Wir erhalten die Anzahl der Knoten und die Anzahl der voneinander verschiedenen Architekturen im Cluster, also ein gewisses Maß für die Heterogenität. Zu jedem Knoten erhalten wir neben seinem Namen und seiner Architektur auch eine Leistungsbewertung.

Die Leistungsbewertung wird allerdings nicht dynamisch bestimmt, sondern – falls nicht anders definiert<sup>1</sup> – auf den Standardwert 1000 gesetzt. Dieser Wert ist statisch. Das ist normalerweise hinreichend. Wenn jedoch die Auslastung der Maschinen starken Schwankungen unterworfen ist, z. B. durch laufenden Betrieb als zentrale Server in einem Rechenzentrum, so erscheint eine Ermittlung der Leistungszahlen während der Laufzeit sinnvoller. Für die Bestimmung der Leistung verweisen wir z. B. auf die Publikationen von J. Dongarra und schlagen vor, die LINPACK-Werte als Leistungszahlen zu verwenden. PVM nutzt diese Leistungsbewertung im Moment nicht [1].

## Load-leveler

Wir geben in diesem Abschnitt zunächst einen Load-leveler an, der ohne Rücksicht auf die Leistungsfähigkeit der einzelnen Knoten zunächst dafür sorgt, daß die Knoten nicht überlastet werden. Im Anschluß diskutieren wir einen Rahmen für einen Load-leveler unter Einbeziehung der Knotenleistungsbewertung.

Wir beginnen mit der Bestimmung der Architektur der virtuellen Maschine:

```
int nhosts,narch;
struct pvmhostinfo *hosts;
pvm_config(&nhosts,&narch,&hosts);
```

Nutzen wir unsere Kenntnisse über die einzelnen Maschinen und ihre Architekturen, so können wir einen Load-leveler wie folgt angeben:

```
int nproc;
int *ntasks,*tids;

nproc=0;
ntasks=new(int,nhosts);
tids=new(int,1024);
for (i=0;i<nhosts;i++) {
    if (!(strcmp(hosts[i].hi_name,"vesta")))
        ntasks[i]=4;
    else if (    (!(strcmp(hosts[i].hi_name,"hera")))
              || (!(strcmp(hosts[i].hi_name,"rhea"))))
        ntasks[i]=2;
    else
        ntasks[i]=1;
    pvm_spawn("thread",NULL,PvmTaskHost,
              hosts[i].hi_name,ntasks[i],tids);
}
```

---

<sup>1</sup>PVM kann in einer Datei über die Architektur der Maschine informiert werden. In dieser Datei gibt es für jeden Knoten einen Eintrag, der zusätzlich zum Knotennamen auch Informationen zur Architektur und eine Leistungsbewertung enthalten kann.

PVM bietet mit der Funktion "*pvm\_spawn*" neben round-robin (PvmTaskDefault) die Möglichkeit, eine frei wählbare Anzahl Threads dediziert auf

- einem bestimmten Knoten (PvmTaskHost) oder
- einer bestimmten Architektur (PvmTaskArch)

zu starten<sup>2</sup>. Übersteigt die Anzahl der zu startenden Threads die Anzahl der physisch vorhandenen Prozessoren, so wird das Verfahren wiederholt. Damit erreichen wir nicht notwendig eine optimale Auslastung (vgl. Abschnitt 1). Spätestens ab diesem Zeitpunkt müssen auch Informationen über die Leistung der Knoten mit ausgewertet werden.

Wie bereits festgestellt, reicht zur Erzeugung eines optimalen Schedules eine Leistungsbeurteilung der Knoten allein nicht aus. Wir schlagen deshalb vor, noch weitere Informationen in eine Konfigurationstabelle aufzunehmen bzw. diese während der Laufzeit zu ermitteln:

1. Architektur: Finden wir unter den Threads eine Menge, die untereinander ein relativ hohes Kommunikationsaufkommen haben, so läßt die Architektur eines Knotens (SISD, SIMD, MISD, MIMD) eine Empfehlung zu, ob der Knoten geeignet ist, diese Threads aufzunehmen.
2. Anzahl der Prozessoren eines Knotens: Diese Zahl läßt einen Rückschluß auf den Typ des Knotens zu (skalarer Rechner oder Parallelrechner).
3. Prozessortyp: Wenn wir eine Unterscheidung zwischen Prozessortypen treffen, so können wir daraus Rückschlüsse auf die Verteilung bestimmter Aufgabentypen ziehen (haben wir z. B. einen Intel- und einen RISC-Prozessor im System und haben wir zwei Aufgaben, von denen eine hohe Fließkommaleistung erfordert, so werden wir diese auf der RISC-CPU bearbeiten lassen, da hier Fließkomma-Operationen wesentlich schneller als auf dem Intel-Prozessor laufen. Selbst bei gleicher Fließkommaleistung der Prozessoren würden wir diese Verteilung anstreben, da Ganzzahl-Operationen auf dem Intel-Prozessor dann schneller ablaufen).
4. (Skalare) Rechenleistung: Diese Zahl erlaubt eine allgemeine Bewertung der Leistungsfähigkeit im Fließkommabereich. Sie wird in MFlop/s (Millionen Fließkommaoperationen pro Sekunde) angegeben.
5. Verbindungsstruktur: Neben der Architektur ist auch die Verbindungsstruktur von Bedeutung. Von ihr ist ein Maß für die Kommunikationsleistung zwischen bzw. innerhalb der Knoten ableitbar. Wir unterscheiden Netzwerke (Ethernert, FDDI etc.), Schaltnetzwerk (Crossbar-switch) und gemeinsamen Speicher (Übertragungsrate, Bandbreite).
6. System: Von Bedeutung kann außerdem das verwendete Betriebssystem sein (Linux, 386BSD, ConvexOS, HP-UX, AIX etc.)

---

<sup>2</sup>Wir können unter PVM nicht sicherstellen, daß auf einem Parallelrechner mit  $n$  physischen Prozessoren jeder höchstens einen von  $m$  Threads ( $m \leq n$ ) zugeordnet bekommt. Das Scheduling wird hier vom Betriebssystem des Parallelrechners übernommen.

Die Merkmale 4 und 5 können wir automatisch bestimmen, die Größe 2 eventuell (abhängig vom Betriebssystem), alle anderen Informationen müssen wir von Hand eintragen.

Weiterhin erscheint es als sehr sinnvoll, eine Kommunikationsleistungsmatrix zwischen den Knoten der virtuellen Maschine anzugeben. Die Bestimmung der Kommunikationsleistung kann auch während der Laufzeit erfolgen. Den Load-leveler generieren wir in mehreren Stufen:

1. Bestimmung der Konfigurationstabelle
2. Bestimmung eines Schedules
3. Start/Verteilung der Threads.

Für die Stufen 1 und 3 können wir relativ einfach einen Rahmen vorgeben, die zweite Stufe bedarf einer genaueren Untersuchung, die unabhängig von PVM erfolgen kann.

## 2 Tests

### Mandelbrot-Menge

Als Standardbeispiel für die parallele Abarbeitung eines Problems kann die Berechnung eines Abbildes der Mandelbrot-Menge gesehen werden. Hierbei handelt es sich um eine Iteration  $x_{n+1} = x_n^2 + c$ ;  $x_i, c \in C$ , bei der die Konstante  $c$  über einem Intervall aus  $C$  variiert wird. Das gewonnene Bild widerspiegelt Informationen über die Divergenzgeschwindigkeit.

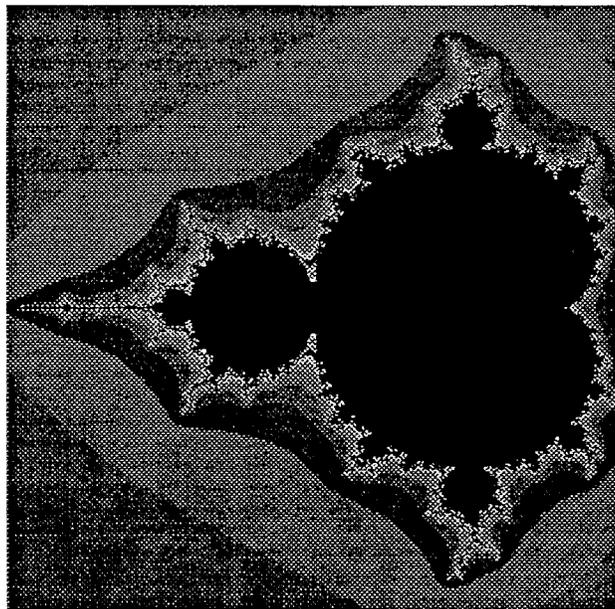


Abb. 1: Abbild der Mandelbrot-Menge

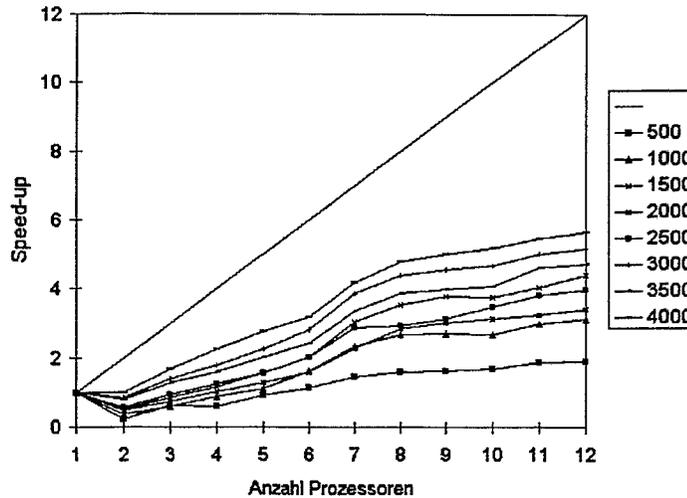


Abb. 2: Speed-up bei unterschiedlicher Problemgröße  
(Die Problemgröße ist gleich der Anzahl der berechneten Punkte)

Der Iterationsprozeß wird für eine diskrete Menge von Punkten der Gaußschen Zahlenebene durchlaufen, wobei er für alle Punkte unabhängig abgearbeitet werden kann. Daraus folgt, daß der Prozeß für alle Punkte parallel abgearbeitet werden kann. Um den Kommunikations-Overhead möglichst gering zu halten, wird die Aufgabe nicht punktweise, sondern zeilenweise verteilt. Ein Master schickt dazu an einen Slave die zur Berechnung einer Zeile nötigen Daten. Der Slave schickt, sobald er die Aufgabe beendet hat, das Ergebnis an den Master, der ihn dann umgehend mit einer neuen Aufgabe versorgt. Auf diese Weise wird gleichzeitig eine dynamische Lastverteilung (Load-balancing) vorgenommen. Das Ergebnis auf dem FZR-Cluster ist in Abb. 2 zu sehen. Aufgezeichnet wurde das Speed-up ( $Speed-up := \frac{t_1}{t_n}$ ,  $t_n = \text{Time-to-solution}$  für  $n$  Prozessoren). Hier ist schon erkennbar, daß die Kommunikation eine entscheidende Rolle spielt. Erst für große Probleme erhält man eine sinnvolle Beschleunigung.

### Gauß-Elimination

Als zweites Testproblem wurde der lineare Gleichungssystemlöser nach Gauß gewählt, der das Problem  $A \cdot x = b$  wie folgt löst:

```

k := 1 → n - 1 (Schleife 1)
  falls  $|a_{k,k}| < \epsilon$  tausche Zeile k mit Zeile t ( $|a_{t,k}| \geq \epsilon$ )
  i := k + 1 → n (Schleife 2)
    p :=  $\frac{a_{i,k}}{a_{k,k}}$ 
    j := k + 1 → n (Schleife 3)
       $a_{i,j} := a_{i,j} - p \cdot a_{k,j}$ 
     $b_i := b_i - p \cdot b_k$ 
i := n → 1 (Schleife 4)
   $x_i := b_i$ 
  j := i + 1 → n (Schleife 5)
     $x_i := x_i - a_{i,j} \cdot x_j$ 
   $x_i := \frac{x_i}{a_{i,i}}$ 

```

Nach der Analyse der Datenabhängigkeiten erkennt man, daß sich die inneren Schleifen (2 und 3) parallelisieren lassen. Sinnvoll ist es natürlich nur, Schleife 2 zu parallelisieren. Damit können wir folgende Überlegung anstellen:

Wir betrachten eine parallele Maschine mit der Rechenleistung  $S$ ;  $[S] = 1MFlop/s$  pro Knoten. Wird Schleife 2 verteilt, so berechnet jeder benutzte Knoten im Schritt  $k$  ein Skalarprodukt aus Vektoren der Dimension  $n - k + 1$  und benötigt dafür die Zeit

$t_l = \frac{1}{S} \cdot 2 \cdot (n - k + 1) + t_{ld/st}$ , mit  $t_{ld/st} \approx 4 \cdot (n - k + 1) \cdot \frac{1}{B_{mem}}$  (wir vernachlässigen die Berechnung von  $p$ ). Mit der Übertragungsbandbreite  $B$ ;  $[B] = 1MByte/s$  läßt sich die zur Berechnung einer Zeile nötige Kommunikation innerhalb von  $t_c = (3 \cdot (n - k + 1) - 1) \cdot s_{fp} \cdot \frac{1}{B}$  abwickeln (die Länge einer Gleitkommazahl ist  $s_{fp} = 4Byte$ , IEEE 764 float).

Bemerkung: Beim Senden des Ergebnisses kann auf das neue Element  $a'_{i,k}$  verzichtet werden, da der Wert a priori feststeht ( $= 0$ ). Es ergibt sich folgendes Kriterium für die Effizienz der Parallelisierung des Gaußschen Algorithmus:

**Satz:** Um bei der Parallelisierung des Gaußschen Algorithmus eine Beschleunigung erzielen zu können, muß für die Kommunikationsbandbreite  $B$  in jedem Schritt der Berechnung gelten:

$$B > \frac{(3 \cdot (n - k + 1) - 1) \cdot s_{fp}}{\left(\frac{2}{S} + \frac{4}{B_{mem}}\right) \cdot (n - k + 1)}$$

Unter Hinzunahme von  $a'_{i,k}$  zum Ergebnis ergibt sich eine etwas nach oben verschobene Abschätzung:

$$B > \frac{3 \cdot s_{fp}}{\frac{2}{S} + \frac{4}{B_{mem}}}$$

Interessant an dieser Aussage ist, daß  $B$  unabhängig von der Problemgröße ist. Wir können außerdem den Term  $\frac{2}{S} + \frac{4}{B_{mem}}$  mit dem reziproken LINPACK-Benchmark identifizieren und erhalten  $B > 12 \cdot |S_{linpack}|$ .

Da die Rechenleistung der Knoten im FZR-Cluster  $\geq 22MFlop/s$  ist, bedeutet das, daß die Kommunikationsbandbreite  $B \geq 264MByte/s$  sein muß. Folglich ist die Parallelisierung auf einem PVM-Cluster, bei dem die Knoten via Ethernet ( $B \leq 1.25MByte/s$ ) verbunden sind, zwecklos. Jedoch ist auf der CONVEX SPP 1000, die eine Verbindungsstruktur mit  $B \approx 400MByte/s$  verwendet, eine Parallelisierung sinnvoll.

Da der kritische Faktor in unserem Beispiel die Skalarproduktberechnung ist, können die getroffenen Aussagen auf alle expliziten, nicht die Struktur der Matrix  $A$  ausnutzenden Lösungsverfahren ausgedehnt werden.

### 3 Konfiguration

PVM ist im FZR auf den Zentralen Servern *hera*, *rhea*, *vesta*, *apoll*, *minos*, *ikon*, *rita*, *rike* und *hebe* (vgl. Abschnitt 4.2) installiert (z. Zt. Version 3.3.1). Um es für einen Nutzer verfügbar zu machen, steht ein Konfigurationsscript bereit, mit dem zunächst ein Konfigurationsprofil erstellt und dann die Konfiguration erzeugt wird. Das Script sorgt dafür, daß die Heterogenität der Maschine verborgen bleibt.

Vorraussetzung ist eine vorhandene Zugangsberechtigung für den Nutzer auf alle Maschinen des Clusters. Weiterhin muß ein von allen Rechnern aus zugreifbares Verzeichnis existieren (am einfachsten läßt sich dies durch ein im Network File System (NFS) integriertes Verzeichnis lösen).

#### Starten und Herunterfahren der Maschine

Der Start der virtuellen Maschine kann von einem der oben angeführten Knoten erfolgen. Dazu wird in einer Shell "\$PVM\_ROOT/lib/pvm \$PVM\_ROOT/hostfile" aufgerufen. Die Datei \$PVM\_ROOT/hostfile enthält die Namen aller zum Cluster gehörenden Knoten. Die Knotenleistungsbewertungen wurden für den Test nicht verwendet. Nach erfolgreichem Start meldet sich die PVM-Konsole mit dem Prompt "pvm>". Der Befehl "conf" veranlaßt PVM zur Ausgabe der aktuellen Konfiguration.

9 hosts, 4 data formats

HOST	DTID	ARCH	SPEED
hera	40000	CNVXN	1000
rhea	80000	CNVXN	1000
apoll	c0000	HPPA	1000
minos	100000	HPPA	1000
ikon	140000	HPPA	1000
vesta	180000	CSPP	1000
rita	1c0000	RS6K	1000
rike	200000	RS6K	1000
hebe	240000	LINUX	1000

Abb. 3: Konfiguration des FZR-Clusters<sup>3</sup>

Eine Anwendung wird mit dem Befehl "spawn <anwendungsname>" gestartet. Bei der Arbeit in der PVM-Konsole lassen sich allerdings Interaktionen zwischen Anwendung und Nutzer nur sehr schwer realisieren. Mit dem Kommando "quit" wird die PVM-Konsole beendet, und man befindet sich wieder in der Shell. Der virtuelle Parallelrechner läuft jetzt im Hintergrund, d. h. PVM-Applikationen können wie normale Programme aufgerufen und abgearbeitet werden. Auch die Interaktion zur Anwendung ist ohne Probleme möglich.

---

<sup>3</sup>Ein Eintrag in der PVM-Konfigurationsdatei besteht aus einem Knotennamen (HOST), einem Identifikator für den PVM-Daemon auf diesen Knoten (DTID), einem Bezeichner für die Architektur des Knotens (ARCH) sowie einem Leistungsindex (SPEED).

Zum Herunterfahren des Clusters ruft man zunächst wieder die PVM-Konsole auf (hierzu genügt diesmal der Aufruf "pvm") und gibt dann den Befehl "halt" ein. PVM stoppt alle, zum Laufzeitsystem gehörenden Komponenten und terminiert mit der Ausschrift "libpvm [txxxxxx]: mxfer() EOF on pvmd sock".

Für eine ausführliche Beschreibung von PVM verweisen wir auf [1].

## Erzeugen einer parallelen Applikation

Um die volle Leistungsfähigkeit komfortabel nutzen zu können, muß etwas Arbeit in den Übersetzungsprozeß investiert werden. Als sehr mächtiges Werkzeug, welches diese Arbeit erleichtert, sollte hierzu *make* angewandt werden. Im Prinzip können folgenden Schritte aber auch per Hand ausgeführt werden:

Für jeweils einen Knoten einer Maschinenarchitektur

1. Einloggen auf diesem Knoten
2. Übersetzen aller Module und Linken der Executables
3. Kopieren der Executables in das entsprechende PVM-Verzeichnis

Bei Verwendung von *make* genügt der Aufruf von "make" auf einem beliebigen Knoten der Maschine. Wir geben jetzt ein Makefile an, das den Übersetzungsprozeß für eine PVM-Applikation beschreibt (zum generellen Aufbau und der Funktionsweise von *make* siehe z. B. [3]).

```
#### Setzen von Variablen (notwendig)
#--- Bestimmen des aktuellen Verzeichnisses
DIR      = $(PWD)
#--- das Verzeichnis der PVM Include-Dateien
PVMIDIR  = $(PVM_ROOT)/include
#--- das Verzeichnis der PVM Bibliotheken
PVMLDIR  = $(PVM_ROOT)/lib/$(ARCH)
#--- die PVM Bibliotheken (fuer C "pvm3", fuer Fortran "fpvm3")
PVMLIBS  = -lpvm3 -lm
#--- das Verzeichnis fuer die Executables
PVMEDIR  = $(PVM_ROOT)/bin/$(ARCH)
#--- Setzen der zu bindenden Bibliotheken (notwendig, auch $(ALIB))
LIBS     = -L$(PVMLDIR) $(PVMLIBS) $(ALIB)
#--- Setzen des Include-Suchpfades (notwendig)
IDIR     = -I$(PVMIDIR)
#### Setzen weiterer Variablen (optional)
#--- Optimierungsstufe des Compilers
OPTI     = -O2
#--- allg. Compileroptionen
OPTIONS  = $(OPTI)
#--- der zu verwendende Compiler (z.B. auch c++)
CC       = gcc
```

```

# Die folgenden Abschnitte bis ~~~ sollten so uebernommen werden
# ~~~~~
# Die executables muessen auf den angegebenen Knoten erzeugt
# werden dazu wird make mit dem Knotennamen und eventuell
# weiteren Parametern z. B. weiteren Bibliotheken ("ALIB=...")
# aufgerufen

all:
    make node "NODE=vesta"\
        "ALIB=-L/usr/convex/all/spp1 /lib/libail.sl";\
    make node "NODE=hera";\
    make node "NODE=rita";\
    make node "NODE=minos"

# Diese Regel beschreibt, wie die Applikation auf den einzelnen
# Knoten erzeugt wird
# 1. Saubermachen
# 2. einloggen auf dem Knoten <NODE>
# 3. Anlegen eines Verzeichnisses mit dem Namen der zu-
#    gehoerigen Architektur
# 3. Aufruf von make (diesmal lokal fuer diesen Knoten)

node:    clean
        rsh $(NODE) 'cd $(DIR);\
        mkdir $(ARCH);\
        make bin "ARCH='$(PVM_ROOT)/lib/pvmgetarch'" "ALIB=$(ALIB)''

# Diese Variante wird gewaehlt, wenn auf allen Architekturen der
# gleiche Code laufen soll. Sollen verschiedene Codes erzeugt
# werden (Maschinenabhaengigkeit, so kann "bin" durch irgendein
# beliebiges Ziel ersetzt werden, z.B. $(NODE)
# ~~~~~
# So wird sauber gemacht (die Option -f ist wichtig, da clean in
# jedem Fall gerufen wird

clean:
    rm -f *.o

# Das ist die "Wurzel"-Regel zur Erzeugung der Applikation auf
# einem Knoten. Diese Regel sorgt gleichzeitig dafuer, dass die
# executables in die richtigen Verzeichnisse transportiert werden

bin:    $(PVMEDIR)/master $(PVMEDIR)/slave

```

```

# Zum Linken ist die Angabe von $(LIBS) unerlaesslich

$(PVMEDIR)/master: $(ARCH)/master.o
    $(CC) -o $(PVMEDIR)/master $(OPTIONS) $(ARCH)/master.o $(LIBS)

# Beim Uebersetzen eines Moduls sollte die Variable $(IDIR) an
# den Compiler weitergeleitet werden
# das Modul wird im Verzeichnis ./$$(ARCH) abgelegt

$(ARCH)/master.o: master.c
    $(CC) -c $(OPTIONS) $(IDIR) master.c

$(PVMEDIR)/slave: slave.c
    $(CC) -o $(PVMEDIR)/slave $(OPTIONS) $(IDIR) slave.c $(LIBS)

```

**Bemerkung:** Bei Abarbeitung auf verschiedenen Architekturen muß auf jeden Fall überprüft werden, ob die ausführbaren Programme numerisch gleiche Ergebnisse liefern.

## Anwendungen

Neben den verschiedenen Testproblemen verweisen wir hier auf eine Anwendung aus dem Institut für Sicherheitsforschung, Abt. Teilchen- und Strahlungstransport. Es handelt sich dabei um einen Code, der eine Monte-Carlo-Methode zur Simulation des Transports schneller Ionen (Fast Ion Transport – FIT) innerhalb des “Gasdynamic Trap“-Experiments am *Budker Institute Novosibirsk* verwendet. Der Code wurde am Institut für Sicherheitsforschung entwickelt ([6]) und dort bereits 1995 auf einem Parsytec PowerXplorer unter Parix parallelisiert ([7]). Das FIT-Programm wurde dann erfolgreich auf die Nutzung von PVM umgestellt und läuft zur Zeit auf der CONVEX SPP 1000. Ein Lauf auf dem gesamten Cluster wird angestrebt.

Eine weitere Beispielapplikation stammt aus dem Bereich der virtuellen Realität und stellt ein virtuelles Beschleunigerlabor zur Verfügung.

## Literatur

- [1] Geist, A., u. a.:  
PVM – Parallel Virtual Machine, A User’s Guide and Tutorial for Networked Parallel Computing;  
The MIT Press, 1994
- [2] Kunicke, M.:  
PVM – ein virtueller Parallelrechner; Benutzerseminar 04.10.1994;  
Abt. FVTK, FZR
- [3] Feldmann, S. I.:  
Make – A Program for Maintaining Computer Programs;  
ConvexOS Tutorial Papers; Eight Edition, Convex Computer Corporation

- [4] Barret:  
Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods;  
SIAM; Philadelphia, 1994
- [5] Schmeißer, N.:  
Parallelrechenseminar am IWR der TU Dresden, WS96;  
<http://www.fz-rossendorf.de/FVTK/MITARB/schmei/publications/parallel>;  
Abt. FVTK, FZR, 1996
- [6] Kumpf, H., St. Krahl, K. Noack, G. Otto, S. Collatz:  
Computer Simulation of a Plasma Neutron Source;  
Institute for Safety Research, Annual Report 1993, FZR, 1993
- [7] Noack, K., G. Otto:  
Parallel Particle Transport Simulations on a PowerXplorer;  
Institute for Safety Research, Annual Report 1995, FZR, 1995
- [8] Coulouris, G., J. Dollimore, T. Kindberg:  
Distributed Systems – Concepts and Design;  
Addison–Wesley, 1994

# Neue Hilfsmittel für die Herstellung von Videos

Horst Böttger

## 1 Hilfsmittel für die Komplettierung von Videos

Videsequenzen stellen heute eine der fortschrittlichsten Präsentationstechniken dar. Sie ermöglichen es, die komplexen Resultate von Computersimulationen in Form von Animationen adäquat darzustellen, wo statische Graphiken nicht mehr ausreichen. Die eigentlichen Animationen eines Videos müssen in der Regel komplettiert werden durch Vorspann, Abspann und Einfügungen. Das sind

- Texte (Titel, Untertitel, Aufzählungen, Erklärungen, Bemerkungen)
- Bilder (Logos, Skizzen, Standszenen, Schlüsselbilder)
- Dunkelfolgen und Pausen (Schwarzbilder)
- Aufblenden von Startszenen und Abblenden von Endszenen der Animation.

Technisch gesehen sind dies Standbilder (Sequenzen konstanten Inhalts) und Bildwechselfolgen (Sequenzen der Übergänge zwischen Bildern, mit oder ohne Beteiligung von Schwarzbildern). Dabei ist die visuelle Gestaltung der Bildwechselfolgen von besonderer Bedeutung, wirkt sie doch unmittelbar auf den Zuschauer ein.

Beim Entwurf geeigneter Werkzeuge gingen wir von folgender Zielstellung aus:

- Systematisierung der mit dem AVS realisierbaren Bildwechseltechniken
- Einbettung aller zu präsentierenden Standbilder in Bildwechselfolgen, die vom Betrachter akzeptiert werden können, Vermeidung abrupter Sprünge
- Ausnutzung aller Möglichkeiten des AVS hinsichtlich Visualisierung und Scriptsteuerung
- Reduktion des Arbeitsaufwandes und der Zeitdauer für die Videokomplettierung
- Vereinfachung der Rechnerkommunikation (weniger Interaktionen)
- einfachste Bedienung ohne Spezialkenntnisse (z. B. AVS).

Die geschaffenen Werkzeuge umfassen mehrere Shell-Scripte (**vinit**, **vgenscr**, **vgentxt**, **vgenseq**, **vgenjpg**, **vclear**) und ein in Fortran geschriebenes Programm (**genscr**). Sie ermöglichen dem Anwender, in einfacher Weise Sequenzen von Standbildern und Dunkelfolgen sowie Bildwechselfolgen durch Überblenden, Mischen und Rollen zu erzeugen. Dazu sind folgende Schritte erforderlich:

- Festlegen der zu erzeugenden Sequenzen, der jeweiligen Bildwechselart und zugehörigen Zeitdauer ("Drehbuch")
- Aufrufen des Scripts `vinit`, der alle benötigten Dateien (Programme, Scripte, Daten, AVS-Netzwerke) in das aktuelle Verzeichnis kopiert
- Bereitstellen aller benötigten Bilddateien und Texttafeln (hierzu stehen spezielle Werkzeuge zur Verfügung, z. B. können Texttafeln für das Rollen vollautomatisch aus ASCII-Daten erzeugt werden)
- Bereitstellen eines Parameterfiles (ASCII-File), in welchem die gewünschten Sequenzen in einer UNIX-ähnlichen Syntax zu codieren sind. Im Header sind neben einigen allgemeinen Angaben auch die Bildgröße in Pixeln und die vorgesehene Bildrate (Frames/sec) anzugeben.

Es stehen folgende Kommandos zur Verfügung; jedes Kommando beschreibt eine in sich geschlossene Sequenz:

- `blend` `<opt>` `<nsec>` `<data>` für Überblendungen
- `merge` `<opt>` `<nsec>` `<data>` für Mischen
- `rollx` `<opt>` `<nsec>` `<data>` für Rollbilder aus Texttafeln
- `rolla` `<ascii>` für Rollbilder aus ASCII-Text
- `sleep` `<nsec>` für Dunkelfolgen.

`<opt>` sind zusätzliche Optionen, `<nsec>` ist jeweils die Anzeigedauer für jede in `<data>` referierte Bilddatei. `<ascii>` ist ein ASCII-File, das einen zum Rollen vorgesehenen Text enthält; im Header sind Angaben zu Fonts, Formatierung, Zentrierung und Hintergrundfarbe optional. Die Dauer jedes Bildwechsels ist auf 1.2 sec festgelegt. Am Anfang und Ende jeder Sequenz wird eine Dunkelfolge von je 0.6 sec hinzugefügt (für spätere Umschnitte). Es können beliebig viele Bilder in einem Kommando codiert werden, die dann innerhalb der Sequenz in gleichartiger Weise dargestellt werden.

Die Abbn. 1 und 2 vermitteln den zeitlichen Ablauf der verschiedenen Bildanzeigen und -wechsel für jeweils drei Bilder (`file1`, `file2`, `file3`) sowie die Abspielzeiten, die die gewünschten Sequenzen später benötigen werden (in den Formeln ist  $m=3$  zu nehmen).

- Aufrufen des Programms `genscr` mit Angabe des Parameterfiles (hier: `param`) durch den Script `vgenscr`:

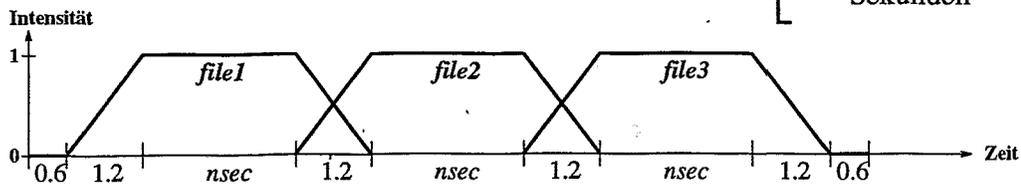
`vgenscr param`

Der Script aktiviert das Programm `genscr`. Dieses liest das Parameterfile `param` und analysiert es. Falls keine syntaktischen Fehler vorliegen, erzeugt `genscr` mehrere Scripte für den AVS-Kommandointerpreter (CLI), unter deren Steuerung alle weiteren Arbeiten zur Generierung der gewünschten Sequenzen vollautomatisch ablaufen, ohne daß der Nutzer AVS-Kenntnisse haben muß. Die Namen dieser Scriptfiles sind `introtxt`, `introgen` und `introjpg`.

Das Programm informiert den Nutzer über die Ergebnisse der internen Syntextests, der lexikalischen Analyse, über Fehler und über die weiteren Schritte, die im konkreten Fall nötig sind. Es handelt sich dabei um maximal drei weitere Scriptaufrufe:

`blend [nsec] file1 file2 file3`

$[m \cdot nsec + (m+2) \cdot 1.2]$   
Sekunden



`blend -black typ [nsec] file1 file2 file3`

$[m \cdot nsec + (2 \cdot m + 1) \cdot 1.2]$   
Sekunden

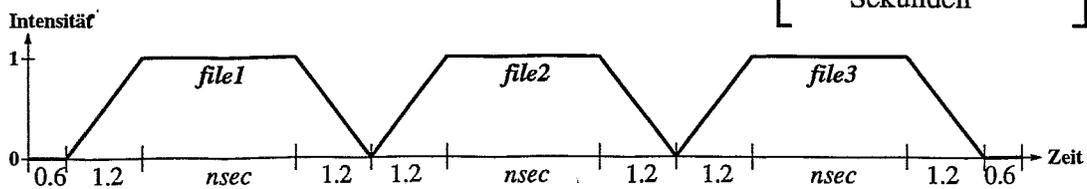
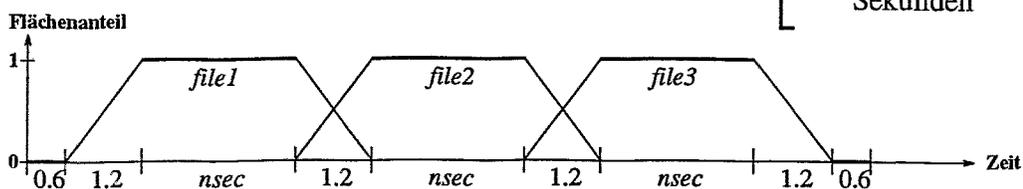


Abb. 1 : Bildwechsel mit blend

`merge [-masc maske] [nsec] file1 file2 file3`

$[m \cdot nsec + (m+2) \cdot 1.2]$   
Sekunden



`rollx [nsec] file1 file2 file3`

$[(m+1) \cdot nsec + 1.2]$   
Sekunden

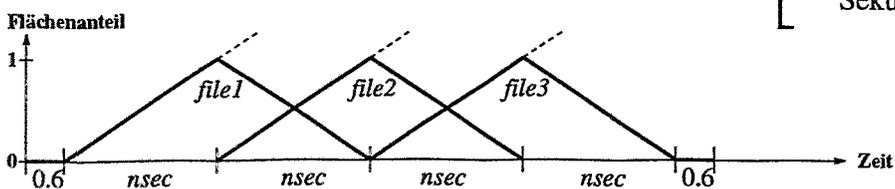


Abb. 2 : Bildwechsel mit merge und scroll

- Aufruf des Generators für die Texttafeln, die zur Bildung von Rolltextsequenzen aus in ASCII geschriebenen Texten erzeugt werden (entfällt, wenn in **param** keine Kommandos **rolla** enthalten sind):

#### **vgentxt**

Der Script aktiviert den Script **introtxt**, dieser ruft das AVS, lädt das Netzwerk **introtxt.net**, erzeugt durch laufende Modifizierung einzelner Modulparameter die Texttafeln als Bildfiles und löscht das AVS wieder.

- Aufruf des Generators für die komplette Folge von Sequenzen, wie sie im Parameterfile codiert sind:

#### **vgenseq**

Der Script aktiviert den Script **introgen**, dieser ruft das AVS, lädt das Netzwerk **introgen.net**, erzeugt durch laufende Modifizierung und Aktivierung einzelner Netzwerkeile (für Standbild, Überblenden, Mischen, Rollen und Schwarzbild) die komplette Folge von Bildfiles als komprimiertes Sequenz-File (spezielles AVS-Format) und löscht das AVS wieder. Die Anzahl der erzeugten Bildfiles kann mehrere Tausend betragen, demzufolge kann der Prozeß mehrere Stunden dauern.

- Umwandlung in das JPEG-Format:

#### **vgenjpg**

Der Script aktiviert den Script **introjpg**, dieser ruft das AVS, lädt das Netzwerk **introjpg.net**, erzeugt aus den einzelnen Bilddaten des Sequenzfiles eine indizierte Folge von JPEG-Bilddateien und löscht das AVS wieder. Der Prozeß hinterläßt eine Liste, in der die Namen aller erzeugten JPEG-Bilddateien gespeichert sind. Diese Filenamenliste dient als Instanz der zu erzeugenden Sequenzen für Vorspann, Abspann und Einfügungen. Die Umwandlung ist so programmiert, daß aufeinanderfolgende identische Bildfiles nicht mehrfach abgespeichert werden.

Eine gleichartige Liste erhält man bei der Umwandlung der eigentlichen Animationsfolge, die das Kernstück des angestrebten Videos darstellt, in JPEG (auch hierfür steht ein spezielles Werkzeug zur Verfügung).

Das Zusammenschneiden des endgültigen Videos wird – quasi schon vor dessen Herstellung – vollzogen durch das entsprechende Zusammenmischen der beiden Filenamenlisten zu einer Gesamtliste **flist** mittels Texteditor.

- Erzeugen des MJPG-Files für die Echtzeitaufzeichnung mit der Parallaxkarte:

```
create_mjpg flist <mjpg file>
```

Das erzeugte File enthält die Einzelbilder in der in **flist** angegebenen Reihenfolge.

Die Abbildung 3 gibt einen Überblick über die einzelnen Produktionsschritte, die nach der Erzeugung der Steuerscripte erforderlich sind.

Die Aufzeichnung des Videos erfolgt an einem speziellen Server, der die Parallaxkarte und zugehörige Software enthält, mit einem S-VHS-Videorecorder, wobei die Bildrate (Frames/sec) interaktiv gesteuert werden kann.

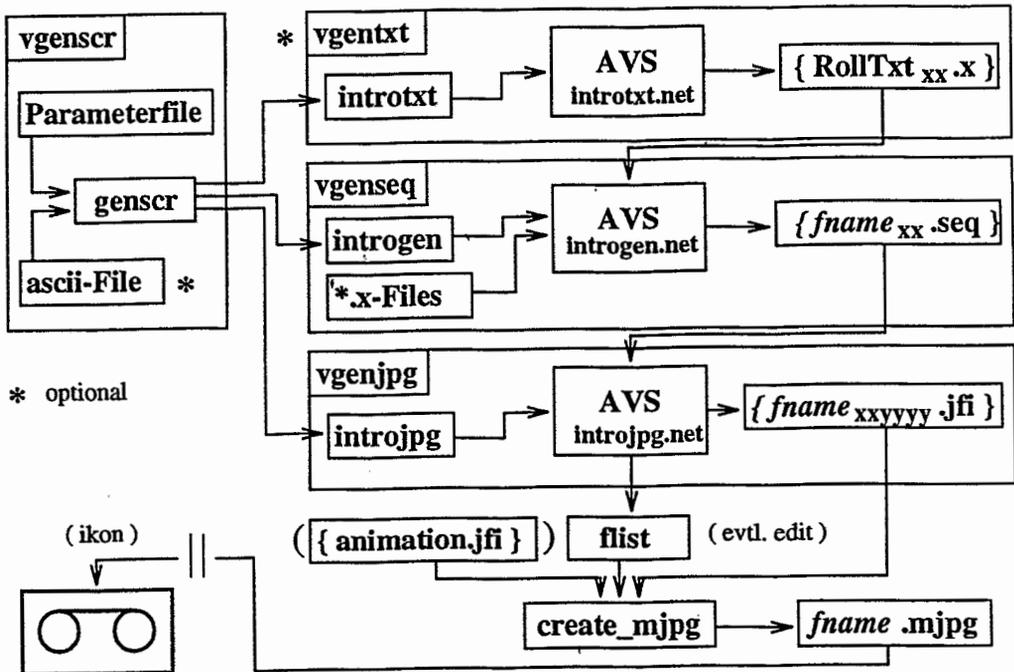


Abb. 3 : Produktionsschritte Videovorspann

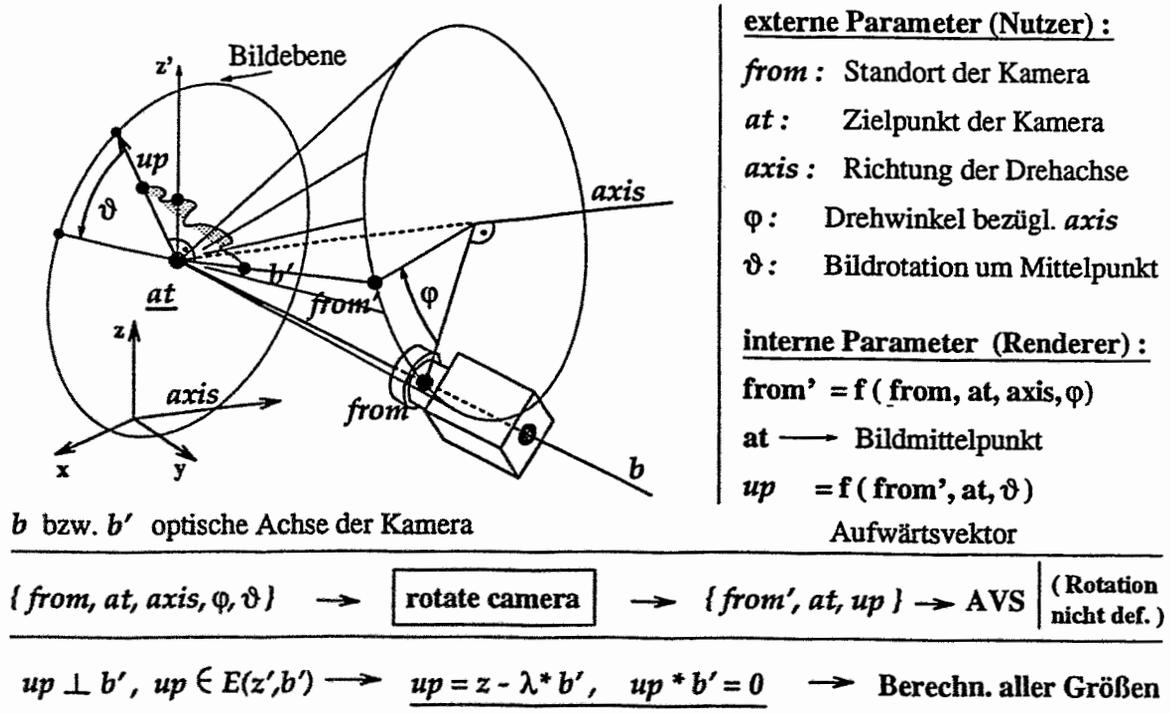


Abb. 4 : Kameraparameter

Bei der Herstellung mehrerer Videos wurden die vorgestellten Werkzeuge mit hoher Effektivität eingesetzt. Mit vergleichsweise geringem Aufwand wurden Videovorspann, -abspann und Einsetzungen erstellt, die in ihrer ausgewogenen Präsentation von Standbildern, Überblendungen, Rolltexten und Dunkelpausen sowie harmonischen Übergängen zu und von den Objektszenen eine hohe Akzeptanz beim Betrachter aufweisen.

## 2 Hilfsmittel für die Animation durch Kamerabewegungen

Neben der Präsentation echter Animationen (Bewegungsvorgänge im Objekt) kommt der Animation starrer (statischer) Objekte durch die Bewegung der Kamera eine große Bedeutung zu. Diese Vorgehensweise ist aus der Welt der virtuellen Realität längst bekannt, wo die Darstellung unbewegter Szenerien oftmals das eigentliche Anliegen ist. Durch die Bewegung der Kamera wird beim Zuschauer eine scheinbare Bewegung der Objekte impliziert. Die Betrachtung verschiedener Seiten oder des Inneren der Objekte ermöglicht die Wahrnehmung von Details oder von Zuständen, die sonst überhaupt nicht darstellbar wären.

Geeignete Objekte sind nicht nur unbewegliche 3D-Szenerien, sondern auch Schnappschüsse oder stationäre Zustände von beweglichen Objekten bei echten Animationen.

Das AVS gestattet zwar, die Kameraposition interaktiv beliebig zu verändern und stellt mit dem Modul zur Keyframe-Animation sogar Hilfsmittel bereit, aus wenigen, interaktiv festgelegten Kamerapositionen eine komplette Flugbahn zu generieren. Doch die Erfahrungen zeigen, daß schon die Erzeugung einer einfachen, glatten Kreisbahn erhebliche Schwierigkeiten bereitet. Die Interaktionen liefern dem Nutzer kaum Informationen über die jeweiligen räumlichen Beziehungen zurück. Trotz umfangreichen Probierens sind die Resultate dann meist nicht befriedigend, insbesondere bei Bewegungen, die über ein simples Hin und Her hinausgehen. Um eine hohe Aussagefähigkeit derartiger Animationen zu erreichen, um die Kamera an interessante Details gezielt und optimal heranzuführen und um die Realisierung der Flugbahnen effektiv zu gestalten, ist es notwendig,

- die Raumbeziehungen zwischen Kamera und Objekten genau zu definieren
- die Kamerabewegungen durch anschauliche, leicht faßbare Parameter zu beschreiben und den Anwender von der Notwendigkeit zu entlasten, sich mit dem Formelapparat für die Transformationen zu befassen
- Hilfsmittel zur Erzeugung von derartigen Bewegungen im dreidimensionalen Raum (im folgenden Flugbahnen genannt) sowie
- Hilfsmittel für die realitätsnahe Gestaltung der Dynamik innerhalb der Flugbahnen (Start, Stop, Beschleunigung, Bremsung, usw.) bereitzustellen.

Die Erfüllung dieser Forderungen auf der Basis des AVS stellt eine echte Erweiterung von dessen Funktionalität dar. In Anbetracht der Erweiterungsfähigkeit war es folgerichtig, einen neuen Weg der AVS-Anwendung zu beschreiten: die Programmierung eigener AVS-Module. Dies bedeutet eine neue Qualität bei der Nutzung des AVS.

Die erstellten bzw. vorgesehenen neuen Module lassen sich in drei Kategorien einordnen:

#### A) Modul `rotate_camera`

Das Modul bildet das Interface zwischen den vom Nutzer gewählten externen Parametern und ihren AVS-internen Entsprechungen, wobei alle erforderlichen Koordinatentransformationen eingeschlossen sind. Er ist über die Modul-Widgets oder den Netzwerk-Datenfluß steuerbar.

Folgende Parameter sind zur Steuerung des Moduls vorgesehen:

Zur eindeutigen Visualisierung einer 3D-Szenerie sind zunächst die Kameraposition (**from**) und der Zielpunkt der Kamera (**at**) notwendig. Beide sind Punkte im dreidimensionalen Raum. Der Differenzvektor dieser beiden Raumpunkte, die Blickrichtung, ist mit der optischen Achse der Kamera identisch. Die beiden Parameter legen die Abbildungsgeometrie fest und lassen nur noch eine Drehung um den Bildmittelpunkt offen. Dieser Freiheitsgrad wird aufgebraucht durch die Festlegung desjenigen zur optischen Achse senkrechten Raumvektors, der in der Bildebene senkrecht nach oben gerichtet erscheint (Aufwärtsvektor **up**), vgl. Abb. 4. Die dazu nötigen Berechnungen werden intern im Modul nach plausiblen Regeln ausgeführt; der Nutzer braucht sich um die Berechnung des Vektors **up** nicht zu kümmern. Mit den aufgeführten Parametern sind somit alle denkbaren Kamerabewegungen realisierbar.

Um jedoch einfache Kamerabewegungen ohne explizite Konstruktion von Flugbahnen bequem steuern zu können, wurde der Begriff der externen Kamerarotation definiert. Sie ist gegeben durch eine beliebige räumliche Achse **axis** und einen Drehwinkel  $\varphi$ . Mit variablem  $\varphi$  führt die Kamera eine gebundene Rotation aus, wobei die optische Achse den Mantel eines Kreiskegels mit der Achse **axis** und der Spitze **at** überstreicht. Zu jedem  $\varphi$  gehört intern eine transformierte Position **from'**, die dann zur Berechnung von **up** verwendet wird. Ein weiterer Parameter  $\vartheta$  dient schließlich zur Drehung des Vektors **up** innerhalb der Bildebene, was einer Kameradrehung um die optische Achse entspricht. Damit können besondere Effekte erzielt werden, z. B. Kurvenneigungen.

Die Parameter und ihre Relationen sind in Abb. 4 dargestellt.

Bei der Anwendung des Moduls `rotate_camera` können wir also zwei Typen von Kamerabewegungen erkennen, je nachdem, welche externen Parameter beim Durchlaufen des Moduls im Netzwerk konstant gehalten werden bzw. variabel (d. h. zeitlich veränderlich) sind:

- Kamerarotation:  
**from** und **at** konstant,  
**axis** konstant oder variabel,  
 $\varphi$  variabel,  
 $\vartheta$  konstant oder variabel.
- Kameraflug:  
**from** und **at** variabel,  
**axis** nicht definiert,  
 $\varphi$  gleich Null,  
 $\vartheta$  konstant oder variabel.

Natürlich können beide Typen kombiniert werden, was aber zu sehr komplexen Bewegungen führt und damit die Klarheit der Wahrnehmung beeinträchtigt.

Die konstant gehaltenen Kameraparameter sind mittels der entsprechenden Modul-Widgets im Steuerpaneel des Netzwerkes interaktiv festzulegen, die variablen Parameter müssen mittels Modulen im Netzwerk aus den vom Animator generierten diskreten Werten der Zeitvariablen ( $t$ ) berechnet und dem Modul `rotate_camera` über die betreffenden Datenports zugeleitet werden. Dieses Modul gibt die intern gebildeten Resultate als Geometriedaten aus, die zusammen mit den Objektdaten an den Renderer weitergegeben werden.

Ein weiterer Parameter, der für die hier behandelte Problematik jedoch ohne Bedeutung ist, soll nicht unerwähnt bleiben: der Skalierungsfaktor *scale* für die Bildgröße. Er entspricht der Brennweite der Kamera und wird vom AVS bei eingeschalteter Zentralperspektive, wie sie bei Kameraflügen üblich ist, nicht ausgewertet.

#### B) Module für Operationen mit skalaren Daten

Die Datenströme im AVS betreffen in der Regel die zu visualisierenden Daten, meist mehrdimensionale Felder. Skalare Daten treten nur als Steuerparameter von Modulen in Erscheinung. Sie werden, wie oben erwähnt, mit den betreffenden Widgets oder durch das Netzwerk gesteuert und über die Datenports den Empfängern zugeleitet.

Beim Kameraflug sind die skalaren Komponenten der vektoriellen Parameter *from* und (in der Regel) *at* zeitabhängig. Das ergibt bis zu sechs Größen, die in jedem Zeitschritt formelmäßig aus  $t$  berechnet werden müssen. Ein derartiges Netzwerk generiert einen einzelnen Bahnabschnitt.

Eine komplette Flugbahn besteht in der Regel aus mehreren Abschnitten. Wegen der nötigen Komposition kommen als Bahnabschnitte möglichst einfache, überschaubare Kurven in Frage: die Abschnitte von Geraden, Kreisen, Ellipsen, Parabeln, aber auch von Kurven höherer Ordnung wie Konchoiden, Strophoiden, Kardioiden, Zykloiden und Spiralen, sowie echte Raumkurven, bei denen alle drei Koordinaten voneinander unabhängige Zeitfunktionen sind.

Zur Berechnung der Formeln bzw. ihrer Terme innerhalb des Netzwerkes wurden sechs Module erstellt, die in Abb. 5 aufgeführt und näher erläutert sind. Sie stellen Formelrohlinge dar, deren zahlreiche Auswahlmöglichkeiten und Koeffizienten eine effektive Berechnung der Bahnkoordinaten (d. h. Berechnung mit möglichst wenigen Modulen) erlauben. Jeder Koeffizient kann als Konstante oder als Variable fungieren.

Erste Anwendungen der neuen Module für Testsequenzen haben gezeigt, daß die Erzeugung von Kamera-Flugbahnen durch analytische Ausdrücke sehr effektiv ist und die Animationen durch die Kamerabewegungen einen großen Gewinn bei der Darstellung statischer Szenarien erbringen. Die Arbeiten hierzu sind mit dem Ende des Berichtszeitraumes noch nicht abgeschlossen.

Die Abb. 6 zeigt den Entwurf einer Kameraflugbahn, die aus 28 einzelnen, zusammengefügten Bahnabschnitten besteht.

#### C) Module zur Harmonisierung von Kamerabewegungen

Hierzu sind Module konzipiert worden, mit denen die Dynamik der Kamerabewegungen gesteuert werden kann (Start, Stop, Beschleunigung, Bremsung, Geschwindigkeitstrend). Die Arbeiten werden im Jahr 1997 fortgeführt.

( Operationen mit skalaren Daten  $\rightarrow$  Formeln )

float 1 term  $z = U(a*x + b) * c + d$

float 2 term  $z = (U(a*x + b) \begin{matrix} + \\ * \\ / \end{matrix} V(c*y + d)) * e + f$

float 1 func  $z = F(a*x + b) * c + d$

float 2 func  $z = ((F(a*x+b)*c+d)OP(G(e*y+f)*g+h))+k$

float 4 oper  $z = (\pm x1 O1 \pm x2) O3 (\pm x3 O2 \pm x4)$

float math1  $z = F(x)$  oder  $x O1 y$  oder  $x O1 const$

U(t), V(t) =

t	t	1/t	t <sup>2</sup>
---	---	-----	----------------

F(t), G(t) =

t	t	1/t	1-1/t
int	frac	sign	pow
sqrt	ln	log	exp
10 <sup>↑</sup>	sin	cos	tan

OP:  $F \begin{matrix} + \\ * \\ / \end{matrix} G \mid \underline{\underline{G(F)}}$

$\begin{matrix} O1 \\ O2 \\ O3 \end{matrix} \} = \begin{matrix} + & - & * & / \end{matrix}$

input: x y x1 x2 x3 x4      output: z

Parameter: a b c d e f g h k      feststellbar / steuerbar

Abb. 5 : Module zur Bahnbeschreibung / 1

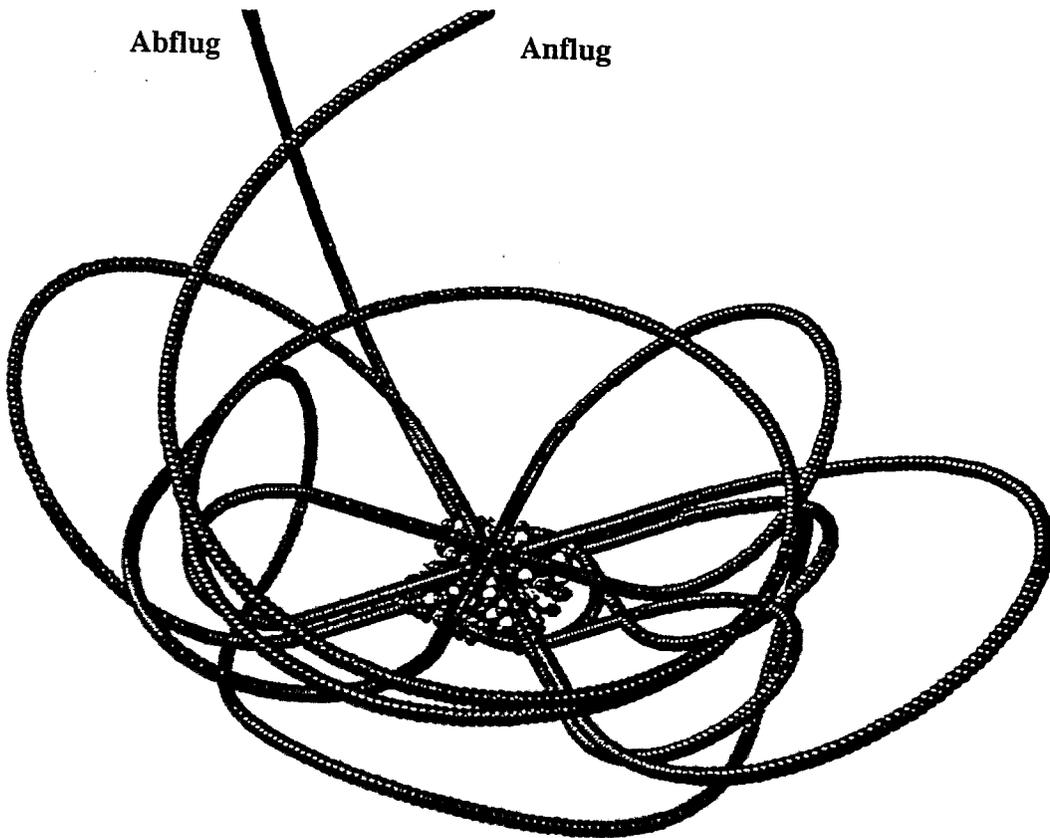


Abb. 6 : Beispiel eines Kamera-Orbits

# L<sup>A</sup>T<sub>E</sub>X<sub>2 $\epsilon$</sub> - die wichtigsten Merkmale

Galina Vinel

Im FZR ist L<sup>A</sup>T<sub>E</sub>X seit 1992 auf den Zentralen Servern verfügbar und wird von der Gruppe Benutzerservice der Abteilung Kommunikation und Datenverarbeitung betreut [1]. Dazu gehören entsprechende Informations- und Konsultationsaktivitäten sowie die Installierung von Neuentwicklungen.

## Geschichtlicher Hintergrund zu L<sup>A</sup>T<sub>E</sub>X<sub>2 $\epsilon$</sub>

1989 diskutierten namhafte Entwickler von L<sup>A</sup>T<sub>E</sub>X-Erweiterungen (Styles, Utilities) gemeinsam mit dem Autor des L<sup>A</sup>T<sub>E</sub>X-Makropakets, Leslie Lamport, Grundlinien der künftigen L<sup>A</sup>T<sub>E</sub>X-Entwicklung. Als Ergebnis entstand das L<sup>A</sup>T<sub>E</sub>X3-Projekt<sup>4</sup> unter der Leitung von Frank Mittelbach (Universität Mainz).

Ziel dieses Projekts ist die komplette Überarbeitung und Erweiterung des o.g. Makropakets. Insbesondere sollen die bisherigen Einschränkungen bei der Schriftenauswahl und bei der Bearbeitung nichtenglischer und mehrsprachiger Texte wegfallen. Der neue, verbesserte Kern dieses Makropakets soll durch verschiedene Pakete ergänzt werden, die spezielle Funktionen (z. B. für Tabellen, Bilder usw.) zur Verfügung stellen.

L<sup>A</sup>T<sub>E</sub>X<sub>2 $\epsilon$</sub>  wurde von dieser Projektgruppe als Vorstufe zu L<sup>A</sup>T<sub>E</sub>X3 herausgegeben [2, 3]. Es steht seit Mitte 1994 als offizielle L<sup>A</sup>T<sub>E</sub>X-Version auf den internationalen T<sub>E</sub>X-Servern (CTAN-Archive [4]) zur Verfügung. Zweimal im Jahr (Juni, Dezember) wird die neueste L<sup>A</sup>T<sub>E</sub>X<sub>2 $\epsilon$</sub> -Version verteilt. Das originale Makropaket von L. Lamport heißt jetzt L<sup>A</sup>T<sub>E</sub>X2.09.

Parallel zur L<sup>A</sup>T<sub>E</sub>X-Entwicklung wurde von einer Arbeitsgruppe der TUG (*T<sub>E</sub>X Users Group/USA*) das TDS-Konzept (*TeX Directory Structure*) zur Standardisierung von L<sup>A</sup>T<sub>E</sub>X-Installationen ausgearbeitet. Mittels der Kpathsea library von Karl Berry wird das rekursive Suchen nach einer bestimmten Datei im L<sup>A</sup>T<sub>E</sub>X-Verzeichnisbaum automatisiert.

Die erfolgreichste L<sup>A</sup>T<sub>E</sub>X<sub>2 $\epsilon$</sub> -Distribution, die auf TDS und der Kpathsea library aufbaut, wurde von Thomas Esser für verschiedene UNIX-Systeme bereitgestellt und heißt *teT<sub>E</sub>X* [5].

Im FZR ist *teT<sub>E</sub>X* in der Version L<sup>A</sup>T<sub>E</sub>X<sub>2 $\epsilon$</sub>  <1996/12/01> an den Zentralen Servern

CONVEX C3200 (*hera*), HP 9000/735 (*minos*),

IBM RS/6000 Mod. 550 (*rita*), IBM RS/6000 Mod. 591 (*rike*)

sowie an den Rechnern

DEC 4000/710 (*alpha*) und HP 9000/735 (*anka*)

verfügbar<sup>5</sup>.

Im vorliegenden Beitrag sollen die wichtigsten Unterschiede zwischen der L<sup>A</sup>T<sub>E</sub>X<sub>2 $\epsilon$</sub> - und der L<sup>A</sup>T<sub>E</sub>X2.09-Version angeführt, einige der neuen Möglichkeiten durch Beispiele demonstriert und Entwicklungstendenzen genannt werden.

## Klassen und Pakete

Einer der wichtigen Unterschiede zwischen L<sup>A</sup>T<sub>E</sub>X<sub>2 $\epsilon$</sub>  und L<sup>A</sup>T<sub>E</sub>X2.09 besteht im Ersetzen der Anweisung `\documentstyle` durch die neuen `\documentclass`- und `\usepackage`-Anweisungen. Die Dokumentklasse (Endung `.cls`) wird mit der `\documentclass`-Anweisung festgelegt. L<sup>A</sup>T<sub>E</sub>X<sub>2 $\epsilon$</sub>  stellt standardmäßig die Dokumentklassen `article`, `book`, `report`, `letter`, `slides`

<sup>4</sup><http://www.tex.ac.uk/CTAN/latex/latex3.html>

<sup>5</sup>Für PCs (Netzlaufwerke) stehen L<sup>A</sup>T<sub>E</sub>X-Versionen (*emtex*) am Server CONVEX C3200 (*hera*) im Verzeichnis `/pcsw/tex` zur Verfügung.

und proc bereit. Die optionalen Styles (mit der Endung .sty) heißen jetzt Packages (Pakete) und werden durch eine \usepackage-Anweisung eingebunden [2, 3].

Damit wird eine klare Trennung zwischen dem dokumentbestimmenden Style (Klasse) und den zusätzlichen Styles (Pakete), deren Aufgabe nur auf kleine Änderungen im Dokumentaussehen beschränkt ist, erreicht. Die Syntax des L<sup>A</sup>T<sub>E</sub>X<sub>2</sub><sub>ε</sub>-Inputfiles sieht jetzt wie folgt aus:

```

\documentclass[list of globaloptions]{documentclass}
\usepackage[list of package options]{package}           % Angabe optional
\usepackage...                                         % Angabe optional
\begin{document}
  Dokumenttext
\end{document}

```

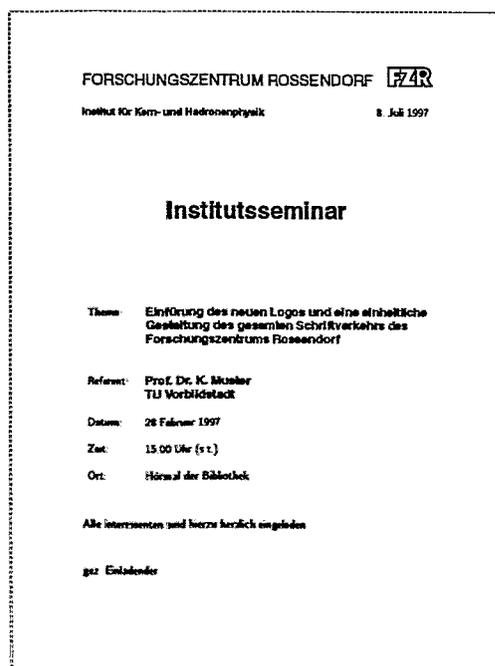
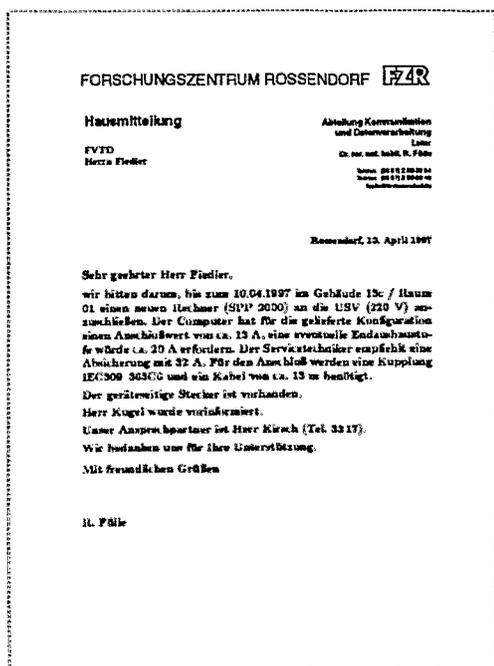
Außer den Standard-Dokumentklassen stehen z. Z. auf den FZR-Rechnern folgende Dokumentklassen zur Verfügung:

- amsart, amsbook, amspro - von  $\mathcal{A}\mathcal{M}\mathcal{S}$  (American Mathematical Society) speziell für Dokumente mit vielen mathematischen Formel
- revtex, elsart - zum Publizieren in USA-Zeitschriften und bei Elsevier Science Publishers B.V.
- seminar - zur Folienerrstellung
- dinbrief, g-brief, myletter - zur Formatierung von Briefen.

Für Briefe mit dem neuen FZR-Briefkopf wurde die Dokumentklasse FZR erstellt. Für interne FZR-Mitteilungen oder für FZR-Faxe können bei dieser Klasse die globalen Optionen fzrhaus oder fzrfax angegeben werden. Die entsprechenden Vorlagen

- fzrbrief.tex, fzrhaus.tex, fzrfax.tex - für die Erstellung von FZR-Briefen, FZR-Mitteilungen, FZR-Faxe
- fzrsem.tex - für die Ankündigung von Institutsseminaren

stehen den Nutzern im Verzeichnis /usr/local/teTeX/texmf/tex/latex/local zur Verfügung. Folgende Beispiele veranschaulichen die Verwendung dieser Vorlagen:



## Schriftenauswahl unter $\text{\LaTeX}2_{\epsilon}$

Eine weitere Neuerung ist NFSS2 (*New Font Selection Scheme*) [2, 3]. Es beseitigt die Schwächen der Schriftenauswahl von  $\text{\LaTeX}$  und ermöglicht neben den *Computer Modern Schriften* die Integration von vielen weiteren Schriftfamilien in  $\text{\LaTeX}$ . So ist es z. B. möglich, mit dem Paket `oldgerman` altdeutsche Schriften (Gothisch, Schwabacher, Fraktur) zu benutzen [6]. Das folgende Beispiel demonstriert gothische Schrift.

```

\documentclass[a5]{article}
\usepackage{oldgerman}
\begin{document}
\gothfamily \Large
DAS LIED VON DER GLOCKE \\
{\rmfamily Friedrich Schiller}
\begin{verse}
Fest gemauert in der Erden \\
Steht die Form, aus Lehm gebrannt. \\
Heute muß die Glocke werden! \\
Frisch, Gesellen, seid zur Hand! \\
\quad Von der Stürne heiß \\
\quad Rinnen muß der Schweiß, \\
Soll das Werk den Meister loben, \\
Doch der Segen kommt von oben. \\
\quad ..... \\
\end{verse}
\end{document}

```

**DAS LIED VON DER GLOCKE**

Friedrich Schiller

Fest gemauert in der Erden  
 Steht die Form, aus Lehm gebrannt.  
 Heute muß die Glocke werden!  
 Frisch, Gesellen, seid zur Hand!  
     Von der Stürne heiß  
     Rinnen muß der Schweiß,  
 Soll das Werk den Meister loben;  
 Doch der Segen kommt von oben.

Mit Paketen des PSNFSS-Systems<sup>6</sup> [3, 7], wie `avant`, `bookman`, `courier`, `helvetic`, `palatino`, `times`, `utopia`, `pifont` u. a., wird die Verwendung von Type1-PostScript-Schriften in  $\text{\LaTeX}$ -Dokumenten vereinfacht. Das Paket `pifont` stellt die Anweisungen für den Zugriff auf Zeichen von ZapfDingbats- und Symbol-Fonts (sog. Pi-Fonts) bereit. Das folgende Beispiel zeigt die Möglichkeiten des `pifont`-Pakets.

```

\ding{38} \ding{113} \ding{236}
\Pisymbol{psy}{210} \Pisymbol{psy}{211}
{\Pifont{psy} w mega} \par
\dingline{35} \par
\begin{dinglist}{43}
\item Der erste Punkt.
\item Der zweite Punkt.
\end{dinglist}

```

Ⓒ ☐ ↷ ® © ω μ ε ρ α  
 ~ ~ ~ ~ ~

☞ Der erste Punkt.

☞ Der zweite Punkt.

## EC-Schriften, Internationalisierung

Ein Problem bei der Anpassung von  $\text{\TeX}$  und  $\text{\LaTeX}$  an nichtenglische Sprachen bestand darin, daß die *Computer Modern Schriften* von  $\text{\TeX}$ , die mit 7 Bit pro Zeichen kodiert sind, nur 128 Zeichen enthalten und keine Umlaute oder andere nationale Sonderzeichen besitzen.  $\text{\LaTeX}2_{\epsilon}$  stellt jetzt die EC-Fonts zur Verfügung, die mit ihren 256 Zeichen (8 Bit pro Zeichen) diesen Nachteil beseitigen.<sup>7</sup>

Ein weiterer Schritt in Richtung Internationalisierung stellt das `babel`-Paket dar. Es unterstützt über 20 verschiedene Sprachen und ermöglicht auch die Verarbeitung von Texten, die in mehr als einer Sprache verfaßt sind. Sowohl nationale Bezeichnungen als auch passende Trennregeln werden durch das `babel`-Paket zur Verfügung gestellt. Jede Sprache, die im Dokument verwendet wird, sollte in der Anweisung `\usepackage[list of options]{babel}` als Option definiert und mittels der entsprechenden Sprachauswahl- oder Umschaltungsan-

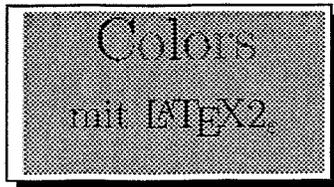
<sup>6</sup>PostScript New Font Selection Scheme

<sup>7</sup>An der Erweiterung der  $\text{\TeX}$ -Schriften entsprechend dem UNICODE (mit 16 bit pro Zeichen kodierten Zeichentabellen) wird unter Leitung von Yannis Haralambous gearbeitet (Projekt OMEGA <http://www.ens.fr/omega/>, <http://www.dante.de/projekte/Projekte.html>).

weisung aktiviert werden.<sup>8</sup>

## Weitere wichtige L<sup>A</sup>T<sub>E</sub>X<sub>2 $\epsilon$</sub> -Pakete

L<sup>A</sup>T<sub>E</sub>X<sub>2 $\epsilon$</sub>  enthält eine Reihe neuer nützlicher Pakete, wie `graphics`, `graphicx` für die Einbindung von Bildmaterial, oder `color`, `colortbl` für die farbige Gestaltung von Dokumenten [8]. Außer vordefinierten Farben stellen o. g. Pakete die Anweisungen für weitere Farbdefinitionen entsprechend dem RGB-, CMYK-Farbmodell oder GRAY-Grauwert bereit. Folgende Beispiele demonstrieren die Möglichkeiten farbiger Boxen- und Tabellengestaltung mit `color`, `colortbl`.



Farbige Boxen

```
(\usepackage{fancybox,shadow},  
\usepackage[dvips]{color})
```

First two columns		Third column
P-type	D-type (dcolumn)	
P-column	and another one	12-34
Total	(wrong)	100-6
Some long text in the first column	bbb	1-2
nan	and some long text in the second column	1-345

Aus der Dokumentation zum `colortbl`-Paket  
(`\usepackage{colortbl,dcolumn,longtable}`)

## Erstellung von russischen Dokumenten

In Zusammenarbeit mit dem VIK Dubna wurde die L<sup>A</sup>T<sub>E</sub>X<sub>2 $\epsilon$</sub> -Version im FZR um das spezielle L<sup>A</sup>T<sub>E</sub>X-Formatfile `3latex` mit dem `wrussian`-Paket und mit russischen Schriften ergänzt. Es erlaubt russische oder gemischte englisch-deutsch-russische Dokumente zu erstellen. Die Inputfiles für solche Dokumente können mit den Editoren `emacs` oder `xedit` direkt mit russischen Alphabet-Zeichen erstellt werden. Dabei müssen natürlich die russische Schriften dem X-Server (z. B. dem X-Terminal) bekannt sein. Das Beispiel rechts daneben zeigt die Titelseite des dazugehörigen Benutzermanuals [9].

## L<sup>A</sup>T<sub>E</sub>X<sub>2 $\epsilon$</sub> -Dokumentation

Über L<sup>A</sup>T<sub>E</sub>X<sub>2 $\epsilon$</sub> -Neuerungen und zu zahlreichen Paketen gibt es eine ausreichende Dokumentation. Diese kann zusammen mit den FVTK-Seminarunterlagen zum Thema L<sup>A</sup>T<sub>E</sub>X über die lokale L<sup>A</sup>T<sub>E</sub>X-Service-Page vom FZR-WWW-Server abgerufen werden (<http://www.fz-rossendorf.de/FVTK/latex/local.html>).

Использование русифицированной версии L<sup>A</sup>T<sub>E</sub>X<sub>2 $\epsilon$</sub>  в операционной системе UNIX.

Анатолий Рапортниченко \*  
([ram@undg1.jinr.dubna.ru](mailto:ram@undg1.jinr.dubna.ru))  
Forschungszentrum Rossendorf e.V.  
Kommunikation und Datenverarbeitung (FVTK)

Аннотация

Данную работу скорее можно рассматривать как подробную инструкцию для пользователя, которому необходимо подготовить хорошо сформатированный документ, содержащий текст либо на двух языках — русском и английском, либо на трех — русском, немецком и английском.

Новая версия системы L<sup>A</sup>T<sub>E</sub>X, система L<sup>A</sup>T<sub>E</sub>X<sub>2 $\epsilon$</sub> , не должна пугать пользователя. Она совместима с предыдущей версией L<sup>A</sup>T<sub>E</sub>X 2.09, и может быть использована для обработки документов, приготовленных в "старом" формате.

Предполагается, что пользователь работает в операционной системе UNIX, в среде X Window и использует для набора текстов редактор Emacs, или xedit. Так же рассмотрен случай когда пользователь подготовил свой документ на какой-нибудь другой машине (например PC, работающей в операционной системе MS DOS), и затем передал его для дальнейшей обработки на UNIX машину.

\*Постоянный адрес: Объединенный Институт Ядерных Исследований, Дубна

<sup>8</sup>Die Bereitstellung von Trennregeln für mehrere Sprachen mittels `babel`-Paket geschieht bei der Installation des sog. L<sup>A</sup>T<sub>E</sub>X-Formatfiles.

## Entwicklungstendenzen

Mit  $\text{T}_{\text{E}}\text{X}$  und  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  formatierte (gesetzte) Texte sehen dank der überragenden Formatierungsfähigkeiten dieser Systeme in der Regel erheblich besser aus als mit herkömmlichen Textverarbeitungsprogrammen (wie *Word* oder *WordPerfect*) gestaltete. Als frei verfügbare, leicht portierbare, sehr flexible Systeme haben sich  $\text{T}_{\text{E}}\text{X}$  und  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  in bestimmten Bereichen fest etabliert, insbesondere dort, wo naturwissenschaftliche Texte verarbeitet werden (an Universitäten, Forschungseinrichtungen und bei den meisten wissenschaftlichen Verlagen). Trotzdem bevorzugen viele Autoren Textverarbeitungssysteme, die erlauben, den Text direkt in seiner formatierten Version zu sehen und zu verarbeiten (Stichwort WYSIWYG).

Die neue graphische Benutzeroberfläche  $\text{L}_{\text{Y}}\text{X}$ , die unter der Koordinierung von Matthias Ettrich entwickelt wurde, bietet einen WYSIWYG-Zugang zu  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  an. Dabei gibt es praktisch keine Einschränkungen in den verfügbaren Fähigkeiten gegenüber dem *echten*  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ . Z. Z. steht  $\text{L}_{\text{Y}}\text{X}$  mit ausreichender Dokumentation [10] den Nutzern auf dem *minos* (HP9000/735) zur Verfügung.

Für Internet-Dokumente, die z. Z. in HTML (*Hypertext Markup Language*) dargestellt werden, ist die SGML (*Standard Generalized Markup Language*) maßgebend. SGML ist als ISO-NORM 8779 festgeschrieben und kann zur Beschreibung von international standardisierten Dokumenten benutzt werden.

$\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  und SGML haben eine Gemeinsamkeit. Beide verwenden das sog. generische Markup-Prinzip zum Setzen von Dokumenten. SGML gibt aber nur die Regeln zur Struktur des Dokuments vor und befaßt sich nicht mit der Formatierung des markierten Dokuments.

Das Einsatzgebiet von  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  wird sich spürbar vergrößern, wenn es sich in die Reihe der SGML-basierten Systeme einordnet und somit als Formatierungs-Maschine für SGML-Systeme eingesetzt werden kann. Das wird ein weiteres Ziel der  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}3$ -Projektgruppe sein. Das ermöglicht dann Autoren, ihre Texte mit vorhandenen SGML-Editoren zu schreiben und anschließend mit Hilfe von  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  zu formatieren. D. h. die ganze Mächtigkeit dieses Systems kann genutzt werden, ohne  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  selbst erlernen zu müssen.

## Literatur

- [1] Vinel, G.:  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  im FZR; FZR-100, Abteilung Kommunikation und Datenverarbeitung; Jahresbericht 1993/1994
- [2]  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}3$  Project Team:  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}2_{\epsilon}$  for authors; Warbrick, J.: Essential  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}++$ ; Dokumentation der  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}2_{\epsilon}$ -Distribution
- [3] Goossens, M., F. Mittelbach, A. Samarin: Der  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ -Begleiter; Addison-Wesley Deutschland 1995; ISBN 3-89319-646-3
- [4] Lammarsch, J., H. Schopmann: CTAN/3, Das  $\text{T}_{\text{E}}\text{X}/\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ -Archiv von DANTE e.V.; Addison-Wesley Deutschland 1996; ISBN 3-89319-794-X
- [5] Esser, Th., D. Hillbrecht, C. Bateman: The  $\text{t}_{\text{E}}\text{X}$ -Guide – Installation und Maintenance; June 1996, Dokumentation der  $\text{t}_{\text{E}}\text{X}$ -Distribution
- [6] Mittelbach, F.: The *oldgerman* package for use with  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}2_{\epsilon}$ ; Dokumentation der  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}2_{\epsilon}$ -Distribution
- [7] Rahtz, S.: A  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  survival guide for Unix systems;  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}3$  Project Team:  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}2_{\epsilon}$  font selection; Dokumentation der  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}2_{\epsilon}$ -Distribution
- [8] Carlisle, D.: Packages in the 'graphics' bundle; The *colortbl* package; Reckdhal, Keith: Using EPS Graphics in  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}2_{\epsilon}$  Documents; und Dokumentation der  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}2_{\epsilon}$ -Distribution
- [9] Рапортиренко, А.: Использование русифицированной версии  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}2_{\epsilon}$  в операционной системе UNIX; FVTK-Dokumentation
- [10] Ettrich, M., u. a.: The  $\text{L}_{\text{Y}}\text{X}$  User's Guide;  $\text{L}_{\text{Y}}\text{X}$ -Dokumentation.

# Erschließen des Telekonferenz-Systems ProShare

Thomas Bach\*, Ruprecht Fülle, Reiner Hliscs\*\*

Die Forschung erkennt zunehmend, daß "Multi-Media"-Techniken nicht allein "Schauwert" besitzen, sondern vor allem greifbaren Nutzen erbringen können.

Beispiele hierfür sind Telekonferenzsysteme:

Mehrere, mindestens zwei kooperierende Teams gruppieren sich jeweils um ein Telekonferenzsystem, das mit Kameras und Mikrophon-Hörern ausgerüstet ist. Sie betrachten und bearbeiten gemeinsam ein Dokument mit bestimmten Tools, sehen und hören sowohl die Gegen- als auch die eigene Seite.

Den Nutzen eines solchen Systems für die Medizin hatte bereits das Münchner Klinikum rechts der Isar im Rahmen eines Pilotprojekts ("Testbed") gemeinsam mit dem DFN-Verein zur Anwendung des Breitband-Wissenschaftsnetzes (B-WIN) [1] überzeugend dokumentiert.

Bei der Zusammenarbeit des FZR mit der Klinik für Nuklearmedizin, Universitätsklinikum "Carl Gustav Carus" der TU Dresden, lag eine analoge Aufgabe vor:

Die im PET-Labor des FZR aufgenommenen Tomogramme sollen gemeinsam mit Ärzten der genannten Klinik ausgewertet werden.

Zu Beginn der gemeinsamen Arbeiten im Jahr 1995 wurden neben aufwendigen, proprietären Systemen bereits Lösungen für PCs und Workstations angeboten, die über dienstintegrierte Netze, wie ISDN oder ATM, aber auch andere Netze, wie Ethernet, verbunden werden können.

1995/96 verfügte die o. g. Klinik noch über keinen leistungsfähigen Anschluß zum Wissenschaftsnetz (WIN), so daß eine ISDN-basierte Lösung in die engere Wahl kam. Von den damals vorgestellten Lösungen der Firmen Bosch, IBM, sietec und Intel/Telekom schien die letztere als "Einstiegslösung" am besten geeignet. Unabhängig von unseren Recherchen entschied sich die Abteilung Medizinische Informatik und Biometrie des Universitätsklinikums für das gleiche System. 1996 begannen die ersten Versuche mit ProShare.

Das Paket "ProShare" beinhaltet 2 PC-Karten (ISDN- und Video-Karte), eine Kamera, einen Mikrophonhörer sowie die zugehörige Software. Benötigt wurde ein PC mit einem Prozessor Intel 486.

Auf dem PC-Bildschirm erscheinen bei der Videokonferenz 4 Fenster:

- Video der Gegenstelle
- Video des lokalen Teams
- "Notizblock" (Fenster für zu bearbeitende Dateien)

---

\*Jetzt Siemens AG München

\*\*Klinik für Nuklearmedizin, Universitätsklinikum "Carl Gustav Carus"

- "Bedienhörer" mit Optionen u. a.
  - zum Wählen von Gegenstellen und zur
  - Aktualisierung der Notizblockinhalte der Partner.

Für die Übertragungs- und Darstellungsparameter besteht eine Vielzahl von Einstellmöglichkeiten.

Die Leistungsfähigkeit des Video-Kanals ist allerdings gering: Bewegungen auf der Gegenstelle erscheinen im Bild ruckartig. Für den vorgesehenen Anwendungsfall spielt dies eine untergeordnete Rolle. Die Tonwiedergabe hingegen ist zufriedenstellend. Offensichtlich verwendet ProShare jeweils einen B-Kanal (64 KBit/s) für Bild und für Ton.

Die Darstellung und Bearbeitung von Bild- und Textdateien geschieht auf dem sogenannten Notizblock. Dieser läßt sich in einen öffentlichen und einen privaten Bereich unterteilen. Die Dateien müssen vor der Darstellung und Bearbeitung in den Notizblock importiert werden. Zugelassen sind neben Textdateien (ASCII) Graphikdateien der Formate PCX, BMP, TIFF, DIB und TGA.

Für den vorgesehenen Einsatzfall ist es wichtig, daß bei Konferenzverbindungen über ISDN die zu importierenden Dateien über das schnellere Ethernet übertragen werden können, ohne die Konferenzverbindung zu beeinflussen.

Zur Bearbeitung der Bild- und Textdateien stehen einfache Tools zur Verfügung, z. B. zum

- Markieren von Bereichen der Dateien in mehreren auswählbaren Farben
- Eintragen von Text in das Bild.

Im FZR läuft ProShare sowohl unter Windows 3.11 als auch unter Windows 95.

Neben ISDN sind auch andere lokale Netzwerke (Ethernet) als Transportmedium für ProShare möglich. Dabei können wahlweise die Protokolle TCP/IP, IPX und NetBIOS zum Einsatz kommen. Die Konferenzteilnehmer können nur über identische Netztypen und -protokolle miteinander kommunizieren.

Die bisher erprobte ProShare-Version (1.x) unterstützt nur die Konferenz zweier Gruppen, was für den vorgesehenen Einsatzfall ausreichend ist.

Inzwischen ist eine überarbeitete Version (2.x) mit neuen Leistungsmerkmalen erschienen. Sie entspricht den Standards ITU H.320 für Videokonferenzen und T120 für Datenkonferenzen. Über ISDN können Multipoint-Konferenzen mit mehr als zwei Teilnehmergruppen unter Steuerung eines Konferenzmanagers durchgeführt werden.

Das System ProShare halten wir nach den bisherigen Ergebnissen und mit den vorliegenden Erweiterungen für eine brauchbare Einstiegslösung in das Teleconferencing.

## Literatur

- [1] Kaufmann, P.: Start der Regionalen Testbeds;  
DFN-Mitteilungen 36-11/94