Abschlußbericht des TIKSL-Projekts

Werner Benger[†,‡], Christian Hege[†], André Merzky[†],
Thomas Radke[‡], and Edward Seidel[‡]
[†]Konrad-Zuse-Zentrum (ZIB), Berlin
[‡]Max-Planck-Institut-für-Gravitationsphysik (AEI), Golm

Kontakt und Projektverantwortliche: eseidel@aei-potsdam.mpg.de, hege@zib.de

7. Mai 2001

1 Zielstellung des Projekts

Das TIKSL-Projekt zielte auf eine allgemeine und einfach handhabbare Lösung für das Überwachen und Steuern entfernter Anwendungen im Bereich des verteilten Höchstleistungsrechnens ab. Diese Lösung sollte aus allgemeinen Gesichtspunkten heraus entworfen (designed), und an Hand einer beispielhaften Umgebung (Simulation Cactus, Visualisierung Amira) implementiert werden.

Aus Projektsicht gliedert sich die geleistete Softwareentwicklung in mehrere Teilbereiche:

- Entwicklung eines allgemeinen Konzepts zum Zugriff auf beliebige entfernte Daten
- damit zusammenhängend ein Konzept zur Beschreibung von Daten
- Entwicklung von Zugriffsmöglichkeiten auf dateibasierte entfernte Daten (Remote File-I/O) und auf applikationsinterne Daten (Remote Streaming)
- Entwicklung von Zugriffsmöglichkeiten auf applikationsinterne Parameter zur Steuerung selbiger (Remote Steering)
- Anbindung der geschaffenen Infrastruktur an Cactus und Amira, Bereitstellung vom Visualisierungsmethoden und Datenhaltungstechniken unter Ausnutzung der neuen Techniken

Aus Sicht der Applikationen sollten dabei keine prinzipiellen Unterschiede beim Zugriff auf lokale oder entfernte Daten erkennbar sein. Dies sollte sich in der entwickelten Softwarearchitektur (siehe Abb. 1) und in einem einheitlichen Programmier-Interface für diesen Datenzugriff niederschlagen.

Im folgenden Abschnitt werden die Ergebnisse des Projekts dargestellt. Der Abschnitt 3 beinhaltet eine detailierte Abrechnung der geleisteten Arbeiten. Es schließen sich einige Bemerkungen zu weiteren Aktivitäten der Projektgruppe, Aussichten auf eventuelle weiterführende Arbeiten, und eine Übersicht zu Veröffentlichungen an.

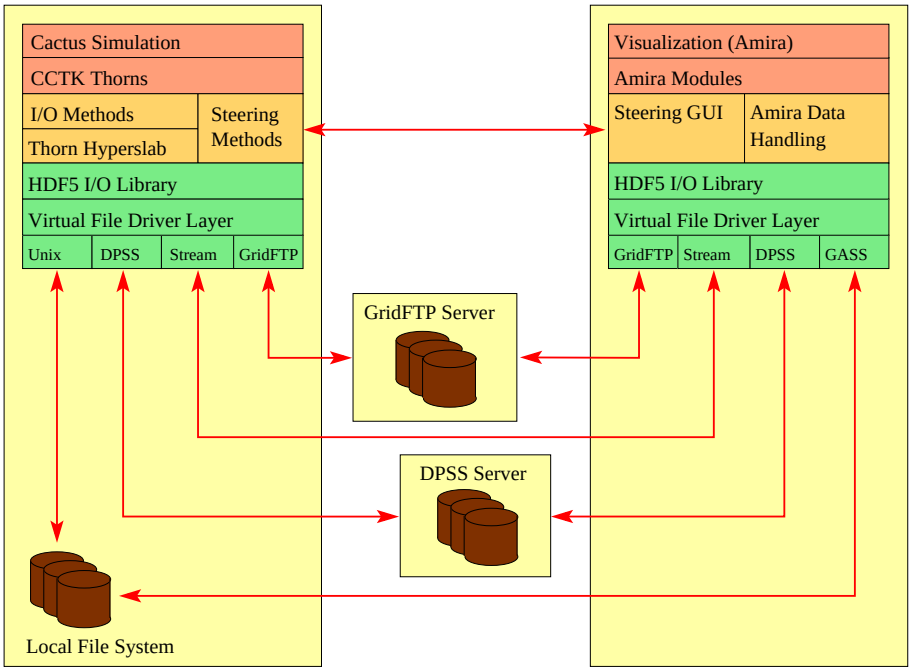

Abbildung 1: *Schema der im Projekt entwickelten Software-Architektur.*

2 Überblick über die Ergebnisse des TIKSL-Projekts

Das TIKSL-Projekt mit seinem Ziel, große numerische Simulationen auf Supercomputern zu neuen, interaktiven Fähigkeiten zu verhelfen, war sehr erfolgreich. Die Projektgruppe und deren Kooperationspartner erfahren weltweite Anerkennung für ihre innovativen Entwicklungen im Bereich des Grid-Computings. Die Mitarbeiter des Projekts wurden mehrfach zu Vorträgen oder Demonstrationen auf verschiedensten internationalen Veranstaltungen eingeladen, um über ihre Entwicklungen und Forschungen zu berichten. Viel wichtiger noch: die im Projekt entwickelten Werkzeuge und Lösungen sind derzeit im Einsatz und werden *routinemäßig* von einer wachsenden Gemeinschaft von Wissenschaftlern und Entwicklern genutzt, und ermöglichen Arbeiten, die vorher nicht oder nur schwierig durchzuführen waren.

Das generelle Ziel des Projekts war es, Techniken, die die Observation, Manipulation und Steuerung entfernter Anwendungen ermöglichen, zu entwerfen, zu entwickeln und in realistischen Umgebungen zu implementieren. Insbesondere wurde die Realisierung von kooperativen Umgebungen in Hochgeschwindigkeitsnetzen, wie etwa im folgenden Scenario, angestrebt:

Ein Nutzer am AEI startet eine Simulation auf einem entfernten Supercomputer, wie etwa der Cray-T3E in Garching. Der Nutzer kann sich dann zu dieser Simulation verbinden, und sich über deren Programmparameter oder auch physikalischen Parameter informieren, und diese bei Bedarf steuern. Ein zweiter Nutzer, etwa am ZIB, kann sich ebenfalls zu dieser laufenden Simulation verbinden, und damit weitere Ressourcen in die Nutzung der Umgebung einbeziehen. Beide Nutzer können die Fortschritte der Simulation mittels (verschiedener) Visualisierungsumgebungen beobachten, die an die Simulation koppeln können. Beide Beobachter können weiterhin sowohl mit der Simulation als auch miteinander über verschiedene verteilte Werkzeuge intera-

gieren. Während dieser Arbeit werden große zeitabhängige Datensätze in variabler 3D-Auflösung über das Netz übertragen. Diese Daten sind auch noch nach der Simulation über das gleiche Netzwerkinterface verfügbar.
Die Werkzeuge für dieses Scenario wurden im Projekt entwickelt, implementiert, getestet und demonstriert, und werden nunmehr in der täglichen Arbeit der beteiligten internationalen Forschungsgruppen genutzt. Das Projekt selber hat dabei zu einer ungewöhnlich guten Kooperation verschiedenster Partner weltweit beigetragen, und andere externe Projekte mit verwandten Themen beeinflusst. Umgekehrt sind verschiedene externe Entwicklungen und Technologien in die Arbeit des TIKSL-Projekts eingeflossen. Daher sollten die Ergebnisse des Projekts nicht allein als Verdienst der Projektgruppe betrachtet werden, sondern als das Ergebniss einer sehr großen Kooperation. Insbesondere sind hier das Argonne National Lab[1], das Lawrence Berkeley Lab[2], das National Center for Supercomputing Applications[3] und die Washington University Saint Louis[4] zu nennen, wie auch die Internationalen Grid-Foren (Global Grid Forum - GGF[5] European Grid Forum - EGrid[6]).

Auf Grund der weiten Kooperationen des TIKSL-Projekts wurden auch Verbindungen zu verschiedenen Internationalen Netzwerkprojekten geschaffen. Dies betrifft, neben dem DFN, Teleglobe, Canarie, NSF, StarTap, die Deutsche Telekom und I-Net/I-Grid. Diese haben das Projekt bei verschiedensten internationalen Demonstrationen großzügig unterstützt.

Im nächsten Abschnitt werden die im Projektplan aufgelisteten Meilensteine detailliert ausgewertet. Um diesen Abschnitt abzurunden, folgt hier bereits eine Zusammenfassung der Resultate.

- Wir haben zur Lösung der Aufgaben des TIKSL-Projekts eine Vielzahl von Modellen und Ansätzen zur Beschreibung wissenschaftlicher Daten betrachtet. Anstatt ein weiteres, neues Format zu schaffen, entschieden wir uns dafür, auf Basis des etablierten HDF5-Formates eine flexible I/O-Schicht zu implementieren, und durch diese Schicht die im Projekt benötigte Netzwerkfunktionalität bereitzustellen. Diese Lösung garantiert nicht nur die Verfügbarkeit von schnellen, generischen I/O-Routinen auf Simulationsseite (inklusive parallelem I/O), sondern eben auch den effizienten Zugriff auf entfernte Dateien oder Speicherbereiche von Seiten einer entfernten Anwendung, wie einer Visualisierung. Die geschaffene I/O-Schicht wurde wiederum in die offizielle HDF5-Distribution des NCSA aufgenommen (Streaming HDF5), und steht damit beliebigen Anwendungen zur Verfügung.

- Auf Basis des Streaming HDF5 und des bereits vorhandenen HTTP-Interfaces in Cactus wurde eine Vielzahl von Interaktionsmechanismen zur Simulation implementiert. Diese erlauben es dem Nutzer, von entfernten Anwendungen aus auf die parallele Simulation einzuwirken. Weiterhin werden diese Schnittstellen genutzt, um die Simulation auf die sich ständig ändernde Umgebung des Grids anzupassen und flexibel darauf zu reagieren.

[1] `http://www.anl.gov/`
[2] `http://www.lbnl.gov/`
[3] `http://www.ncsa.uiuc.edu/`
[4] `http://www.washu.edu/`
[5] `http://www.gridforum.org/`
[6] `http://www.egrid.org/`

- Um die Fähigkeiten des Streaming-HDF5 zu demonstrieren, und um für die beteiligten Wissenschaftler einen zusätzlichen Mehrwert durch die Nutzung dieser neuen Technologien zu geben, wurden im Projekt passende Visualisierungstools entwickelt. Dies geschah zum einen innerhalb der Simulation durch die Implementierung verschiedener Techniken zur Extraktion visualisierbarer Daten, zum anderen durch die Erweiterung der verwendeten Visualisierungsumgebungen um die entsprechenden Schnittstellen und Visualisierungsalgorithmen.

Die im Projekt entwickelten Technologien sind auf den Projektseiten im WWW[7] verfügbar und detailliert beschrieben. Wir möchten den Leser dazu auffordern, diese Seiten und die dortige Dokumentation zu beachten, und die dort ebenfalls verfügbare Online-Demo für einen ersten Eindruck zu nutzen.

3 Detaillierte Auswertung der Meilensteine des Projekts

Im folgenden werden die im Projektplan festgelegten Meilensteine des Projekts genannt und deren Erfüllung ausgewertet. Auf eventuell notwendig gewordene Abweichungen vom Projektplan wird dabei besonders eingegangen.

– **Einrichtung einer Hochgeschwindigkeits-Netzinfrastruktur**

  Dieses Ziel wurde, aufbauend auf den Vorarbeiten anderer Testbed-Projekte, im ersten Quartal erreicht. Leider kam es während der Projektlaufzeit oft zu netzbedingten Problemen, was aber in einem Testbed nicht anders zu erwarten ist.

– **Leistung von Netzwerk- und Technologiesupport für die beteiligten Arbeitsgruppen**

  Diese Leistung wurde während der gesamten Projektlaufzeit erbracht.

– **Definition, Evaluation und Implementation eines generischen Dateninterfaces in Cactus, welches am HDF5-API orientiert ist**

  Dieser Arbeitspunkt hat sich (wie zu erwarten) als der mit Abstand aufwendigste herausgestellt. Verschiedene geplante Ansätze haben sich nicht realisieren lassen (I/O über verteiltes Shared Memory, Eingliederung eines Requestprotokolls in die unteren Softwareschichten von HDF5, oder über einen HDF5-Wrapper). In enger Zusammenarbeit mit der HDF5-Entwicklergruppe am NCSA wurde letztendlich eine Lösung implementiert, die das HDF5-API direkt benutzt, und sämtliche Netzwerkfunktionalität in den untersten Softwareschichten in einem sogenannten Virtual File Driver (VFD) realisiert. Dieses von der HDF5-Gruppe neu geschaffene Konzept erlaubt das flexible Einbinden verschiedenster

[7] `http://www.zib.de/Visual/projects/TIKSL/`

I/O-Schichten in die HDF5-Bibliothek, zur Laufzeit der Programme. Diese VFDs implementieren dann die notwendige Funktionalität (Verbindungsaufbau, Datenverteilung, Verschlüsselung...), bleiben aber (außer bei der Initialisierung) vor dem Benutzer verborgen.

– **Implementierung von Kontroll- und Steuermechanismen für Cactus**

Das modular aufgebaute Cactus wurde um ein Konzept zur Steuerung von Parametern erweitert. Dessen Schnittstellen erlauben es, ausgewählte Parameter aller Module (Thorns) zur Laufzeit der Simulation zu verändern. Die zur Steuerung notwendigen Informationen können dabei über die vorhandenen Netzanbindungen von Cactus bezogen werden: entweder über den HTTP-Port, oder aber über das Streaming-HDF5.

– **Portierung der Umgebung auf alle relevanten Plattformen**

Streamed-HDF5: HDF5 selber ist auf einer Vielzahl von Plattformen verfügbar. Der geschaffene VFD kann auf all diesen ebenfalls eingesetzt werden. Dies betrifft neben verschiedensten Unix-basierten Systemen auch die für die Projektgruppen relevanten Cray-T3E-Systeme, aber auch Windows-Systeme.

Cactus: Cactus läuft auf allen für die Arbeitsgruppe interessanten Workstations, Großrechnern und Clustern, inklusive Cray-T3E, Origin2000, Linux-Clustern, Windows-NT-Clustern, Hitachi (LRZ), und sogar IA64-Clustern.

Amira: Amira wurde von der Amira-Entwicklergruppe während der Projektlaufzeit auf verschiedene Systeme portiert. Neben der ursprünglichen SGI-Irix-Version sind nunmehr auch Versionen für HP-UX, Sun-OS/Solaris, Linux und Windows-Systeme verfügbar.

– **Die geschaffen Lösung ist für andere Projekte verfügbar**

Dies ist durch die Aufnahme des Streaming-HDF5-VFD in die offizielle HDF5-Distribution auf hervorragende Weise gewährleistet. Jeder Nutzer von HDF5 kann durch das einfache Setzen von Konfigurationsparametern ein einsatzfähiges Streaming-HDF5 erhalten.

Weiterhin kommen die Erweiterungen in Cactus einer stark wachsenden Gemeinschaft von Physikern zugute, die Cactus in verschiedensten numerischen Simulationen einsetzen. Cactus entwickelt sich mehr und mehr zu einem 'Grid-aware Simulation Toolkit'.

Auch die Erweiterungen in Amira sind für Amira-Nutzer verfügbar, und erlauben es, Amiras Visualisierungsmöglichkeiten auf Basis von HDF5-Input (Dateien oder Streams) zu nutzen.

– **Dokumentation**

Die Dokumentation der geschaffenen Umgebung wurde in mehreren Teilen realisiert: Der HDF5-VFD ist in der HDF5-Distribution dokumentiert. Die Cactus- und Amira-Erweiterungen werden durch HOWTOs beschrieben, die jeweils eine Einführung in den Aufbau und die Nutzung der verteilten Umgebung geben. Es finden gegenwärtig Einführungen von externen Nutzern in die Nutzung der Software statt (EGrid Testbed, AEI-Wissenschaftler).

– **Erweitern des Cactus-Codes um integrierte Visualisierung**

Die Möglichkeit, komplette zeitabhängige und voll aufgelöste 3D-Datensätze aus Cactus zur Visualisierung zu streamen, erübrigt eigentlich prinzipiell die Integration von komplexen Visualisierungsalgorithmen in die Simulation. Diese würden unter Umständen die Simulation ausbremsen. Oft jedoch sind die eigentlich interessanten Information *viel* kleiner als der komplette Datensatz. Dies trifft besonders dann zu, wenn aus dem Datensatz einfache Geometrien gewonnen oder Schnitte durchgeführt werden. In Cactus wurden daher eine Reihe von Visualisierungsalgorithmen implementiert. Diese sind:

- **Isosurfacer:**
  Aus den 3D-Daten werden in Cactus Isosurfaces berechnet. Der Algorithmus dazu kann parallele Architekturen ausnutzen, und verschiedene Isolevel gleichzeitig erzeugen.

- **Geodäten:**
  Ein Cactus-Thorn erlaubt es, Lichtstrahlen in der gegenwärtig simulieren Raumzeit zu berechnen. Diese Geodäten werden als 2D-Listen ausgegeben.

- **Ereignishorizonte:**
  Die für die Simulation Schwarzer Löcher besonders interessanten Ereignisshorizonte werden als Superpositionen von sphärischen harmonischen Funktionen berechnet.

- **2D-Slices:**
  Besonders für einfache 2D-Darstellungen über das HTTP-Interface ist das Anfertigen von Schnitten durch einen 3D-Datensatz sinnvoll. In Cactus werden diese Schnitte zu Bildern im JPEG-Format gewandelt, und können zu den Beobachtern gesendet werden.

- **1D-Slice:**
  Analog werden 1D-Schnitte (Linien) angefertigt, die über ein schmalbandiges Interface (wie HTTP) effizient Informationen über die Simulationsdaten vermittlen können.

– **Multiple parallele I/O-Kanäle, Multiple Clients**

Die geschaffene Lösung erlaubt mehreren verschiedenen Visualisierungs-Clients und Web Browsern, sich gleichzeitig zu einer laufenden Simulation zu verbinden. Weiterhin kann

eine Cactus-Simulation mit einer anderen über Streaming-HDF5 in Kontakt treten.

Die gegenwärtige Implementation erlaubt jedoch nicht die Nutzung mehrerer paralleler Datenströme zwischen der Simulation und *einem* Client. Diese Funktionalität wird in Zukunft über einen neuen VFD implementiert werden, der auf der GSI-FTP-Lösung der Globus-Gruppe basiert. Dies ist im Abschnitt 4 näher beschrieben.

– **Simultane entfernte Visualisierung an verschiedenen Standorten**

Dies ist, wie eben beschrieben, implementiert. Verschiedene Visualisierungsclients können die 3D-Daten erhalten, und auf diesen lokal beliebige Visualisierungsmethoden anwenden.

– **Kooperative Werkzeuge, Einbeziehung von Videokonferenzen in die Umgebung**

Die geschaffenen Werkzeuge erlauben das kooperative Betrachten und Steuern laufender Simulationen über eine breite Palette verfügbarer Interfaces, wie z.B. Web-Browser.

Die ursprünglich geplante Integration von Videokonferenzsystemen in die verteilte Umgebung wurde in einem experimentellen Stadium realisiert. Es stellte sich jedoch heraus, das eine ATM-basierte Lösung zu aufwendig und inflexibel ist. Eine einfachere Lösung wurde später aus Zeitgründen, aus Mangel an Austattung und Erfahrung, und wegen mangelnder Nachfrage nicht mehr realisiert. Das Augenmerk der beteiligten Arbeitsgruppen richtet sich gegenwärtig auf das AccessGrid-Projekt, welches eine passende Lösung in diesem Bereich anbietet.

– **Generische HDF5-Funktionalität in Amira**

In Amira wurde ein Modul implementiert, welches das Lesen von zeitabhängigen HDF5-basierten Daten über verschiedene VFDs erlaubt. Diese Modul erkennt die typischen, in Cactus generierten Datentypen.

– **Design und Implementierung eines prozeduralen Interfaces zum Zugriff auf entfernte Daten**

Dieses Interface ist durch die Verfügbarkeit des expliziten Zugriffs auf den entfernten Datensatz im Wesentlichen überflüssig geworden. Der hierbei zu erzielende Performance-Gewinn stand in keinerlei Verhältnis zum erwarteten Aufwand, da diese Lösung zwingend die Implementation eines zusätzlichen, externen Request-Protokolles in HDF5 erfordert hätte. Weiterhin wird die durch den prozeduralen Zugriff erwartete Flexibilität durch das ebenfalls flexible HDF5-Interface und dessen Kombination mit den Cactus-Steuerinterfaces, die auch die I/O-Funktionen betreffen, gewährleistet.

– **Datenserver für den Zugriff auf entfernte HDF5-Dateien**

  Dieser Datenserver wurde experimentell auf Basis eines DPSS-Systemes implementiert. Das *D*istributed *P*arallel *S*torage *S*ystem erlaubt es, Datensätze auf mehrere Server zu verteilen, und auf diese über einen DPSS-Datenserver parallel zuzugreifen. Dies wurde von uns am Stand des DFN auf der CeBit'2000 demonstriert. Eine zukünftig angestrebte Lösung wird jedoch auf einem einfacher zu handhabenden und flexibler einsetzbaren GSI-FTP-Server der Globus-Gruppe basieren. Dieser stellt im Wesentlichen einen WU-FTP-Server dar, der um verschiedene Grid-Fähigkeiten (Security, Parallele Streams) erweitert wird. Experimente hierzu wurden im Rahmen der SC2000 durchgeführt.

– **Demonstration der Allgemeinheit der Lösung durch Anwendung in verschiedenen Applikationen**

  Das im Projekt implementierte Streaming-HDF5 wurde genutzt, um neben Amira auch eine zweite Visualisierungsumgebung um den Zugriff auf entfernte Onlinedaten zu erweitern. Dies wurde am IBM-DataExplorer (DX) demonstriert, dessen Quellen von IBM während der Projektlaufzeit glücklicherweise unter der GPL freigegeben wurden. Der DX erhielt damit in etwa die selben Schnittstellen wir Amira. Ebenso wurde das Visualisierungstool LCA-Vision, welches am NCSA entwickelt wurde, mit Streaming-HDF5-Schnittstellen ausgestattet.

  Weiterhin wurden die in Cactus geschaffenen Schnittstellen und Erweiterungen nicht nur in den geplanten Simulationen genutzt, sondern von verschiedenen Kooperationspartnern auch in weiteren numerischen Simulationsszenarien getestet. So wurden diese Komponenten beispielsweise von der Zeus-Entwicklergruppe um Mike Norman am NCSA genutzt.

– **Optimierung des Datentransfers**

  Die oben beschriebenen in Cactus integrierten Visualisierungsmethoden bieten die Möglichkeit, durch Feature-Extraction das nötige Datenaufkommen sehr stark zu reduzieren. Weiterhin bieten das Streaming-HDF5 und die entsprechenden Cactus-Schnittstellen die Möglichkeit, die Datenauflösung durch die Anwendung von Hyperslabs an die verfügbare Netzwerkanbindung anzupassen, ohne dazu die eigentliche Simulationsauflösung reduzieren zu müssen. Der künftige, auf GSI-FTP basierende VFD wird außerdem die Möglichkeit zur Online-Datenkompression bieten.

  Insbesondere zu erwähnen sind hier auch die Implementation von Transaktionen im HDF5-I/O-Layer. Dies garantiert, dass nicht jeder low-level-I/O-Aufruf einzeln über das Netzt transferiert wird, sondern daß zusammengehörende Aufrufe in einem Paket übertragen werden. Insbesondere in WANs mit nicht vernachlässigbaren Latenzzeiten trägt dies wesentlich zur Performancesteigerung bei.

– **Durchführung von Live-Demonstrationen**

Wie aus dem Abschnitt 6 deutlich wird, wurde dieser Arbeitspunkt glänzend erfüllt.

Zusammenfassend ist zu sagen, dass die im Projektplan gestellten Aufgaben und Meilensteine im Wesentlichen ausgezeichnet erfüllt wurden. Einige Abweichungen von den Projektplänen ergaben sich aus notwendigen Design-Entscheidungen heraus. Nichtsdestotrotz wurden im Projektverlauf auch Themen angerissen, die im Projekt selber aus Ressourcengründen noch nicht zufriedenstellend bearbeitet werden konnten. Dazu zählen insbesondere Arbeiten zu allgemeinen Themen der Handhabung wissenschaftlicher Datensätze (Beschreibung, Metadaten, Replikation, Zugriffsmechanismen...), als auch zur Integration der geschaffenen Umgebung in verbreitete Grid-Umgebungen, obwohl gerade in diesen Punkt sehr viele Anstrengungen investiert wurden, und sehr ermutigende Fortschritte zu verzeichnen sind.

4 Nutzer der TIKSL-Ergebnisse und Verbindung zu anderen internationalen Projekten

Die Arbeiten des TIKSL-Projekts zur Nutzung verteilter Umgebung folgten dem allgemein erkennbaren weltweiten Trend des Grid-Computings. Das Grid-Paradigma schafft eine wohldefinierte systematische Grundlage zur Schaffung von verteilten Werkzeugen und Umgebungen. Aufbauend auf diesen Grundlagen arbeiten weltweite Kooperationen an der Umsetzung in reale Softwarelösungen. Diese gesamte Entwicklung wird durch Standardisierungsgremien, wie dem Global Grid Forum, koordiniert. Einige Gruppen, wie die Globus-Gruppe, nehmen dabei sowohl durch ihre ausgedehnten Aktivitäten, als auch durch ihre zentralen Softwareentwicklungen und grundlegenden thematischen Arbeiten großen Einfluss auf diese Entwicklungen.

Die bisherigen Arbeiten all dieser Projektgruppen konzentrierten sich im Wesentlichen auf die Schaffung von Abstraktionsschichten zu den im Netzwerk verteilten Resourcen, und darauf aufbauend auf standardisierte Komponenten zum sicheren Daten- und Informationsaustausch. Genau in dieser Blickrichtung lagen eben auch die Arbeiten des TIKSL-Projekts, so dass sich hier interessante Synergien ergaben. Einige von diesen sollen jetzt kurz dargestellt werden. Im darauf folgenden Abschnitt werden die Aktivitäten der Gruppe in den Grid-Gremien dargestellt.

HDF5-Gruppe

Aufgrund der zentralen Stellung der HDF5-I/O-Bibliothek im TIKSL-Projekt war die Zusammenarbeit mit deren Entwicklern sehr eng. Mehrere Besuche am NCSA haben einerseits dem Projekt geholfen, den Streaming-HDF5-VFD effizient und and die HDF5-Interna angepasst zu implementieren. Andererseits haben aber die Bemühungen der TIKSL-Projektgruppe auch dazu beigetragen, dass in der HDF5-Gruppe auf allgemeine Belange der verteilten Datenhaltung und Kommunikation verstärkt Rücksicht genommen wurde. Nicht zuletzt hat diese Zusammenarbeit die Entwicklung des Konzepts der Virtuellen File Driver in HDF5 bestärkt und hat dessen Design mit beeinflusst.

Ein weiteres Feld der Zusammenarbeit mit dieser Gruppe ergab sich aus dem Problembereich der Datenbeschreibung. HDF5 selber stellt es dem Programmierer/Nutzer vollkommen frei, in

welchen Strukturen Daten gespeichert werden. Dies ermöglicht einerseits natürlich einen extrem flexiblen Einsatz von HDF5 für verschiedenste Datenklassen, birgt aber andererseits die Gefahr, dass zwei HDF5-fähige Programme zueinander inkompatibel werden, wenn sie verschiedene Datenlayouts benutzen. Die HDF5-Gruppe geht nunmehr dazu über, via XML eine Datenbeschreibungssprache in die Bibliothek zu integrieren, die die externe Definition von Datenstrukturen und deren Übernahme in die HDF5-Anwendungen wesentlich vereinfachen wird. Gleichzeitig bemühen sich die Mitglieder der Projektgruppe, in den Gridforen eben solche Datenstrukturstandards zu schaffen (siehe Abschnitt 5)

FlexIO

HDF5 selber ist noch nicht allzulange stabil verfügbar. Die Vorversion HDF4 hingegen ist weltweit anerkannt und wird seit Jahren erfolgreich eingesetzt. Die Limitationen dieser Version bezüglich Datenlayout und Dateistruktur versuchte John Shalf vom NCSA mit einer weiteren Softwareschicht, FlexIO, zu überwinden. Cactus nutzte FlexIO bisher als eines seiner Standard-Formate. Amira kann die von Cactus erzeugten FlexIO-Dateien lesen und verarbeiten. Die Zusammenarbeit mit dem NCSA auf diesem Gebiet erwies sich für die TIKSL-Gruppe als überaus fruchtbar, und führte letztendlich zu der implementierten Lösung.

DICE

Während der Designphase für das Streaming-HDF5 hat die Projektgruppe mehrere Möglichkeiten für den eigentlichen Datentransfer der Online-Daten ins Auge gefasst. Eine anscheinend optimale Lösung schien lange Zeit darin zu bestehen, die in der Simulation vorhandenen Daten über einen global verteilten Shared-Memory-Layer entfernten Teilnehmern zugänglich zu machen. In der Tat hatte die US-Amerikanische DICE-Gruppe auf Basis ihres eigenen Shared-Memory Systems (DGSM) eine solche Lösung bereits für eine frühere HDF-Version (HDF4) implementiert. Die TIKSL-Projektgruppe war daher daran interessiert, diese Lösung zu übernehmen und weiterzuentwickeln. Dies scheiterte aber leider an den rigorosen Lizenzbedingungen der militärisch finanzierten DICE-Gruppe. Weitere Bemühungen unsererseits, einen Ersatz für die DGSM-Bibliothek unter der GPL oder einer ähnlichen Lizenz zu finden, schlugen leider fehl. Eine solche Bibliothek zu implementieren überstieg wiederum die Resourcen des Projekts, zumal nur sehr ungenaue Abschätzungen zum Performanceverhalten einer Lösung möglich waren.

DPSS

Für den Zugriff auf entfernte dateibasierte Daten boten sich dagegen von Anfang an mehrere Möglichkeiten. Neben den Globus-Lösungen (siehe unten) war dies insbesondere der DPSS-Ansatz der DIDC-Gruppe am Lawrence Berkeley National Laboratory. Das DPSS-System als hochverfügbares, verteiltes, paralleles und schnelles Speichersystem bietet anspruchsvollen und datenintensiven wissenschaftlichen Anwendungen eine hervorragende Möglichkeit zur Datenhaltung in verteilten Umgebungen. Im TIKSL-Projekt wurde daher eine erste Lösung auf Basis des DPSS implementiert und demonstriert. Diese Lösung erwies sich jedoch als sehr aufwendig zu warten und lieferte nicht die erhoffte Performance. Durch engen Kontakt zu den DPSS-Entwicklern wurde versucht, diese Engpässe zu beheben, als auch die Software auf weiteren Architekturen (neben IRIX und Linux) einzusetzen, was jedoch nur teilweise gelang. Für weitere

Designentscheidungen als auch für prinzipielle Überlegungen bezüglich verteilter Datenhaltung im Grid war diese Zusammenarbeit aber von grundlegender Bedeutung.

GASS / Globus-DataGrid / GSI-FTP

Das Team um Ian Foster (Argonne National Lab) und Carl Kesselman (Information Science Institut) entwickelt mit Globus eine sehr breit angelegte und schichtweise (und damit flexibel) aufgebaute Infrastructur (Middleware) für Grid-Umgebungen. Das Globus-Toolkit stellt dabei eine Reihe von Bibliotheken und Services bereit, die von Entwicklern und Nutzern in eigene Projekte eingebunden werden können. Unter diesen Globus-Diensten befassen sich einige auch mit der Handhabung verteilter und entfernter Daten. Die ursprüngliche GASS-Bibliothek (GASS: Global Access to Secondary Storage) stellt dabei eine einfache und effiziente Möglichkeit zur Verfügung, entfernte Dateien 'on demand' auf dem lokalen Rechner zwischenzuspeichern und zu bearbeiten. Auf GASS basierend wurde von der HDF5-Gruppe eine erste HDF5-Version implementiert, die diesen Zugriff auf ferne Daten nutzte. Der Transport nutzt dabei wiederum andere Globus-Dienste, die insbesondere Sicherheit und Datentransport betreffen.

GASS selber ist in seinen Fähigkeiten allerdings limitiert: es bietet keinen partiellen Datenzugriff, benutzt ein proprietäres Protokoll, lässt keine globalen Locks zu, und ist überhaupt relativ inflexibel. Eine Umstellung des GASS-Transferprotokolls auf HTTP hat diese Situation nicht wesentlich geändert.

Auf das Betreiben mehrere Entwicklergruppen hin (darunter die Globus-Gruppe und die TIKSL-Gruppe) wurde eine wesentlich allgemeinere und flexiblere Architektur entworfen, die für sogenannte DataGrid-Umgebungen viel besser geeignet sein sollte. Diese Globus-Datagrid basiert ebenfalls auf standardisierten Protokollen (FTP), die jedoch um partiellen Dateizugriff, parallele Streams, Third-Party-Transfers und Sicherheitsmechanismen erweitert werden (GSI-FTP - diese Erweiterungen sind im FTP-RFC vorgesehen oder standardisiert!). Darauf aufbauend werden auf höherer Abstraktionsebene Dienste implementiert, die für Entwickler und Nutzer den Zugriff auf entfernte und verteilte Daten vereinfachen und beschleunigen. Zu diesen Diensten gehören insbesondere verteilte Replikationsmechanismen, aber auch Anbindungen an die Grid-Informationssysteme sowie Scheduling- und Caching-Mechanismen. Diese Arbeiten werden nunmehr eng mit den Grid-Foren koordiniert, wobei auch die TIKSL-Gruppe aktiv ist (siehe Abschnitt 5.

MPICH-G

Eine wichtige Middleware-Komponente im Globus-Toolkit stellt MPICH-G dar – eine MPI-Implementierung, die auf den Sicherheits- und Transportmechanismen der Globus-Middleware aufbaut. In mehreren Experimenten mit verteilten massiv-parallelen Simulationen wurden hier von der TIKSL-Gruppe Erfahrungen gesammelt und an die MPICH-G-Entwickler weitergegeben. Die Handhabung solcher großen verteilten Simulationen stellte eine der hauptsächlichen Motivationen für das TIKSL-Projekt dar, hat aber auf Grund von Performanceproblemen und Verfügbarkeit von Supercomputern und Netzwerken (transatlantisch) keinen dominanten Stellenwert im Projekt erreicht.

ASC/KDI

Eine weitere Gruppe, die sich mit dem Zugriff auf entfernte und verteilte Simulationen beschaftigt, ist die des ASC-KDI-Projekts (ASC: Astrophysics Simulation Collaboratory, gefördert über das NSF-KDI-Programm: Knowledge and Distributed Intelligence). Dieses Projekt, was auf Anwenderseite ebenfalls Cactus betrachtet, entwickelt sogenannte *"Portals"* zu Simulationen. Diese stellen ein Web-Interface für den Nutzer dar, und ermöglichen Konfiguration, Kompilation und Programmkontrolle von entfernten Rechnern aus. Die Portale nutzen dabei wiederum Grid-Middleware, um diese Funktionalität zu implementieren (Globus, Streaming-HDF5), und stellen außerdem Schnittstellen zu anderen Diensten wie Visualisierung/Steuerung zur Verfügung. Daring liegen eben auch die Synergien zum TIKSL-Projekt.

Visualisierung in der Allgemeinen Relativitätstheorie

Die im Projektzeitraum implementierten Visualisierungsalgorithmen in Cactus wurden im Wesentlichen von der ART-Projektgruppe am AEI/ZIB geleistet. Diese Gruppe beschäftigt sich gerade mit Problemstellungen zu Visualisierungsmethoden in der Allgemeinen Relativitätstheorie, wie z.B. von Tensorfeldern. Hier ergab sich eine sehr fruchtbare Kooperation zwischen Visualisierung und entferntem Dateizugriff. Diese erstreckt sich insbesondere auch auf Untersuchungen zu Strukturierung wissenschaftlicher Datensätze.

Visualisierung von Daten Adaptiver Hierarchischer Netze

Die in Cactus-Simulationen anfallenden großen Datenmengen sind nicht nur ein Problem der darauf folgenden Visualisierung, sondern behindern die Simulation unter Umständen selber sehr stark (Speicherbedarf!). Durch die Nutzung von adaptiv-verfeinerten Gittern (AMR: Adaptive Mesh Refinement) wird dieses Problem angegangen. Diese interne Datenstruktur wiederum erfordert auch Arbeiten auf der I/O-Ebene, im Datenlayout, und in den Visualisierungsalgorithmen. Dieser Themenkomplex wurde von vom AMR-Visualisierungsprojekt am ZIB, der Cactus-Gruppe und der TIKSL-Projektgruppe gemeinsam bearbeitet.

Synergien durch andere Projekte

Die Zusammenarbeit mit verschiedensten internationalen Projektgruppen stellt unserer Meinung nach einen sehr wesentlichen Aspekt unserer Arbeit dar. Zum einen hat dies die Entwicklungen unseres Projekts entscheidend geprägt. Zum anderen führten diese Kooperationen auch zu weitreichender Akzeptanz der geschaffenen Lösungen, und zur nahtlosen Integration dieser in andere, große Grid-Projekte (HDF5, Globus). Wir möchten uns an dieser Stelle bei allen beteiligten Projekten sehr herzlich bedanken!

5 Auswirkungen des TIKSL-Projekts auf GF, GGF und EGrid

Erst spät in den 90er Jahren wurde der Name *Grid-Computing* als neuer Oberbegriff für verteilte Systeme in Weitverkehrsnetzen geprägt. Dieser beinhaltet eigentlich keine maßgeblich neuen Ideen und Ansätze für solche verteilten Systeme, stellt jedoch diese auf eine solide theoretische

und konzeptionelle Grundlage. Dies wiederum ermöglicht es der internationalen Gemeinschaft von Forschern und Entwicklern, ihre Anstrengungen auf diesem Gebiet zu systematisieren, und gemeinsam zu koordinieren. Deutlichste Anzeichen für diese Bemühungen sind die Gründungen verschiedener Arbeits- und Standardisierungsgremien, die sich derzeit zum Global Grid Forum (GGF) vereinigen[8]

Der ursprüngliche US-amerikanische Zweig des Grid Forums wurde im Sommer 1999 mit Unterstützung der NASA und des NSF gegründet, und wurde wesentlich von den "Vätern" des Grid-Konzepts, Ian Foster (ANL), Carl Kesselmann (ISI) und der Globus-Gruppe beeinflusst. Der europäische Zweig wurde unabhängig davon im Herbst 1999 ins Leben gerufen. Während der IGrid-Konferenz im Sommer 2000 in Yokahama äußerten auch Gruppen aus dem Asiatisch-Pazifischen Raum Interesse an der globalen Organisation der Grid-Aktivitäten — die eigenständigen Gremien verschmolzen daraufhin zum GGF. Das TIKSL-Projekt spielt insofern in diesen Bemühungen eine nicht unwesentliche Rolle, indem sie in 2 der Arbeitsgruppen des EGrid die Chairs stellt, und an deren Arbeit aktiv beteiligt ist.

Dies betrifft zum einen die "Testbed and Applications Working Group", die das Ziel hat, ein europaweites Grid-Testbed aufzubauen, und dieses zur Entwicklung und Evaluierung von Grid-Technologien zu nutzen und anderen Gruppen zur Verfügung zu stellen. Cactus wurde dabei zu einer der wesentlichen Testapplikationen gewählt. Diese Gruppe konnte bereits zur SC2000 ein Testbed mit 11 europäischen Institutionen präsentieren. Diese WG verschmolz mit der "Application and Testbeds Working Group" des US-Grid-Forums.

Zum anderen stellt die TIKSL-Gruppe auch den Chair der "Data Storage and Management Working Group". Das Ziel dieser Gruppe ist es, Schnittstellen und Protokolle für Grid-fähige Data Management Systeme zu definieren und zu implementieren. Diese WG verschmolz mit der "Remote Data Access Working Group" des US-Grid-Forums.

Durch die zentrale Stellung von Cactus als typische Grid-Testanwendung, als auch durch die Standardisierungsbemühungen bezüglich des Zugriffs auf entfernte und verteilte Daten im Grid hat die Arbeit des TIKSL-Projekts einen sehr umfangreichen Rahmen erhalten, und ordnet diese auch in noch folgende Aktivitäten ein.

6 Publikationen, Konferenzbeiträge, Technologiedemonstrationen und weitere Aktivitäten der TIKSL-Projektgruppe

Publikationen

In Zusammenhang mit den Arbeiten des TIKSL-Projekts wurden eine Anzahl von wissenschaftlichen Arbeiten veröffentlicht, die im Folgenden aufgelistet werden.

— Werner Benger, Ian Foster, Jason Novotny, Edward Seidel, John Shalf, Warren Smith, Paul Walker: *"Numerical Relativity in a Distributed Environment."* **Proceedings of the Ninth SIAM Conference on Parallel Processing for Scientific Computing, March, 1999**

[8] Vom 4. bis 7. März 2001 findet in Amsterdam die erste Konferenz des GGF statt.

— Hubert Busch, André Merzky: *"Kollision von Neutronensternen über dem Atlantischen Ozean – Transatlantisches Metacomputing auf dem Globus - mit Globus?"* **Tagung des IGC-Arbeitskreisses, Düsseldorf, 16./17. September 1999**

— André Merzky: *"Fernsteuerung und Fernüberwachung von verteilten Anwendungen auf Höchstleistungsrechnern."* **Tagungsbroschüre zum Workshop "Wissenschaftliche Anwendungen auf Höchstleistungsrechnern", Sep. 1999, RRZN/Universität Hannover**

— Gabrielle Allen, Werner Benger, Tom Goodale, Hans-Christian Hege, Gerd Lanfermann, Joan Masso, André Merzky, Thomas Radke, Edward Seidel, John Shalf: *"Solving Einstein's Equations on Supercomputers."* **IEEE Computer, Dec. 1999, pp. 52-59**

— Werner Benger, Hans-Christian Hege, André Merzky, Thomas Radke, Edward Seidel: *"Efficient Distributed File I/O for Visualization in Grid Environments."* **PDC Proceedings, "Simulation and Visualization on the Grid", Lecture Notes in Computational Science and Engineering, PDC'99, Springer, Jan. 2000**

— Werner Benger, Hans-Christian Hege, André Merzky, Thomas Radke, Edward Seidel: *"Efficient Distributed File I/O for Visualization in Grid Environments."* **ZIB Technical Report SC-99-43, Januar 2000**

— Werner Benger, Hans-Christian Hege, André Merzky, Thomas Radke, Edward Seidel: *"Schwarze Löcher sehen."* **DFN-Mitteilungen, Bd. 52, 2000**

— Gabrielle Allen, Tom Goodale, Gerd Lanfermann, Thomas Radke, Edward Seidel: *"The Cactus Code: A Problem Solving Environment for the Grid."* **1st EGrid Workshop at ISTHMUS 2000, April 2000, Poznan**

— André Merzky: *"Data Description."* **1st EGrid Workshop at ISTHMUS 2000, April 2000, Poznan**

— Gabrielle Allen, Werner Benger, Tom Goodale, Hans-Christian Hege, Gerd Lanfermann, André Merzky, Thomas Radke, Edward Seidel, John Shalf: *"The Cactus Code: A Problem Solving Environment for the Grid (??)."* **Proceedings of the Ninth IEEE International Symposium on High-Performance Distributed Computing (HPDC'00)**

— Gabrielle Allen, Thomas Dramlitsch, Tom Goodale, Gerd Lanfermann, Thomas Radke, Ed Seidel, Thilo Kielmann, Kees Verstoep, Zoltan Balaton, Peter Kacsuk, Ferenc Szalai, Joern Gehring, Axel Keller, Achim Streit, Luděk Matyska, Miroslav Ruda, Aleš Křenek, Harald Knipp, André Merzky, Alexander Reinefeld, Florian Schintke, Bogdan Ludwiczak, Jarek Nabrzyski, Juliusz Pukacki, Hans-Peter Kersken, Giovanni Aloisio, Massimo Cafaro, Wolfgang Ziegler, Michael Russell: *"Early Experiences with the EGrid TestBed."* **Submitted zur CCGrid 2001 (angenommen)**

— André Merzky, Reagan Moore, Omer F. Rana, Heinz Stockinger: *"Data Management for Grid Environments."* **Submitted for HPCN 2001**

— Gabrielle Allen, Werner Benger, Thomas Dramlitsch, Tom Goodale, Hans-Christian Hege, Gerd Lanfermann, André Merzky, Thomas Radke, Edward Seidel, John Shalf: *"Cactus Tools for Grid Applications."* **Submitted for Cluster Computing Journal (eingeladen)**

Vorträge

— André Merzky: *"Fernsteuerung und Fernüberwachung von verteilten Anwendungen."* **IGC-Meeting, Düsseldorf, Sep. 16/17 1999**

— André Merzky: *"Visuelle Kontrolle und Steuerung von verteilten Anwendungen auf Höchstleistungsrechnern."* **Workshop "Wissenschaftliche Anwendungen auf Höchstleistungsrechnern", 28./29.9.99 im RRZN/Universität Hannover**

— André Merzky: *"Efficient Distributed File I/O for Visualization in Grid Environments."* **ParallellDatorCentrum Annual Conference on "Simulation and Visualization on the Grid", December 15-17, 1999, Center for Parallel Computers (PDC), Royal Institute of Technology (KTH), Stockholm, Sweden**

— Edward Seidel: *"EGrid Panel Session."* **1st EGrid Workshop at ISTHMUS 2000, April 2000, Poznan**

— Edward Seidel: *"EGrid Testbeds."* **1st EGrid Workshop at ISTHMUS 2000, April 2000, Poznan**

— André Merzky, Alexander Reinefeld: *"High-Performance Computing on German Gigabit WANs."* **ISTHMUS 2000 Conference, April 2000, Poznan**

— André Merzky: *"Data Description."* **1st EGrid Workshop at ISTHMUS 2000, April 2000, Poznan**

Demonstrationen

Zu verschiedenen Anlässen wurden von der TIKSL-Gruppe Technologie-Demonstrationen durchgeführt. Diese sind im folgenden kurz aufgelistst.

— HPDC8 (Aug 3-6, 1999, Redondo Beach):
`http://www.ece.arizona.edu/ hpdc/hpdc8/`

— SC'99 (Novi 13-19, 1999, Portland/OR):
`http://www.sc99.org/`

— CeBit 2000 (24.02.2000, Hannover):
`http://www.dfn.de/cebit2000/`

— HPDC9 (August 1-4, 2000, Pittsburgh): `http://www.cs.cmu.edu/ hpdc/demos.html`

— iGrid2000 (8-21 July 2000, Yokahama):
`http://www.startap.net/igrid2000/ger-usa_blackhole.html`

— SC 2000 (Nov 4-10, 2000, Dallas/TX):
`http://www.sc2000.org/`

Weitere Aktivitäten

— Einweihung des Niederländischen Höchstleistungsrechners “Unite” (24.03.1999, Sara, Amsterdam):
`http://www.unite.nl/nieuws/algemeen/index.html`

— Offizieller Start des Gigabit-Wissenschaftsnetzes: Live Demonstration “Schwarze Löcher kollidieren lassen” (30.06.2000, DFN, ZIB, Berlin):
`http://www.dfn.de/presse/dfn-presse/pm00-06-30b.html`