

Grid-Computing in Echtzeit

W. Eppler, T. Müller, V. Hartmann, S. Tchilingarian, A. Augenstein, M. Naji, H. Gemmeke, IPE

Einleitung

Unter Grid-Computing versteht man das Zusammentreffen von hoher Rechenleistung, großen Speicherkapazitäten, breiter Vernetzung und Integration von Diensten. Die Dienste (engl: services) spielen im folgenden Beitrag (wie meist auch in der Industrie) die Hauptrolle. Die Integration von Diensten bedeutet eine Vereinheitlichung von Softwarekomponenten bei teilweise ganz unterschiedlicher Hardware und unterschiedlichen Betriebssystemen, so dass praktisch alles, was über das Grid angefordert wird, nach demselben Schema abläuft. Erreicht wird dies durch Schnittstellen, die dafür sorgen, dass ein Benutzer auf eine Anfrage nicht, wie derzeit im Internet mit der Suchmaschine Google, unsinnige Daten gleichrangig neben sinnvollen Daten zurückerhält, sondern über einen rechnergeführten Dialog gezielt zum gewünschten Ergebnis geführt wird. Die Integration von Datenstrukturen ist daher ein wichtiges Ziel, das von zwei Seiten her angegangen wird: einerseits wird bei neuen Entwicklungen darauf geachtet, dass auf bestehenden Datenstrukturen aufgesetzt wird und anwendungsspezifische Datentypen daraus abgeleitet werden. Andererseits werden Mechanismen geschaffen, die Daten mit unterschiedlichen Formaten ineinander überführen können.

Das Global Grid Forum (GGF) [1] ist eines der wichtigsten Konsortien zur Standardisierung des Grids. Der bisher noch nicht verabschiedete Standard OGSA (Open Grid Services Architecture)

[2] definiert die grundlegenden Dienste im Grid (zu Benutzermanagement, Finden von Diensten, Datenzugriff, ...). Für den einheitlichen Datenzugriff gibt es die Arbeitsgruppe DAIS (Data Access Integration Services), die einen Standard unter gleichem Namen erarbeitet. Wichtige Vorarbeiten werden von einem schottischen Konsortium aus Universitäten und den Firmen IBM und Oracle im Rahmen des Projektes OGSA-DAI [3] geleistet. Resultat ist bisher eine Programmierumgebung für Datenzugriffe im Grid, welche auf der bekanntesten Grid-Middleware Globus aufbaut [4]. Als Middleware wird im Grid die Software-Schicht bezeichnet, die zwischen Betriebssystem und Anwendungsprogramm für Plattformunabhängigkeit und die Bereitstellung von Grid-Diensten sorgt.

Diese Arbeiten reichen noch nicht aus, um Forschungsprojekte wie das Neutrinoexperiment KATRIN [5] auf dem Grid zu realisieren, weil bisher wenig Arbeit aufgewendet wurde, das Grid echtzeitfähig zu machen. Mit Echtzeit ist die strikte Einhaltung von Antwortzeiten gemeint, die ein Computer benötigt, um auf eingehende Daten zu reagieren. Wenn z.B. ein Computer das Signal eines Temperatursensors zu spät an ein Kontrollsystem weiterleitet, weil er durch andere Aufgaben überlastet ist, kann es zu gefährlichen Systemzuständen kommen. Da im Grid die verschiedenen Rechenressourcen üblicherweise über das nicht echtzeitfähige Internetprotokoll TCP/IP vernetzt sind, gibt es hier Handlungsbedarf.

Anwendungen als Motivation

Grid-Computing als momentaner Schrittmacher moderner Informationstechnologien ist eindeutig anwendungsgetrieben. Es erhält seinen Reiz durch die Generalisierung von Komponenten, die einzelnen Anwendungssystemen entspringen und auf einmal auf vielfältige Weise in anderen Systemen verwendet werden oder zumindest mit ihnen interagieren können. Ohne Blick auf die einzelnen Anwendungen könnten die Anforderungen an das Grid nie formuliert werden.

Das IPE ist für die Messwerterfassung, die Datenakquisition, die Systemkontrolle, die Datensichtung und die Datenanalyse einer Reihe von Experimenten im Forschungszentrum zuständig. Dazu zählen Projekte aus der Teilchenphysik, Fusion, Krebsforschung und Proteomik. Bei diesen Projekten sind meist die gleichen Aufgaben zu lösen, allerdings unter verschiedenen Umgebungen und unterschiedlichen Anforderungen. Bisher ist es nicht gelungen, ein einheitliches System zu finden, das alle diese Aufgaben löst. Es ist aber äußerst unökonomisch, bei jeder neuen Systementwicklung wieder von vorne beginnen zu müssen. Unabhängig von den übrigen Vorteilen des Grid mit seinen fast unbegrenzten Rechen-, Speicher- und Netzwerkressourcen ist sein Bestreben zur Vereinheitlichung von Software-Diensten genau das Richtige, um modulare und wiederverwendbare Software-Komponenten für die zuvor erwähnten Anwendungen zu entwerfen. Zum ersten Mal ist es mit

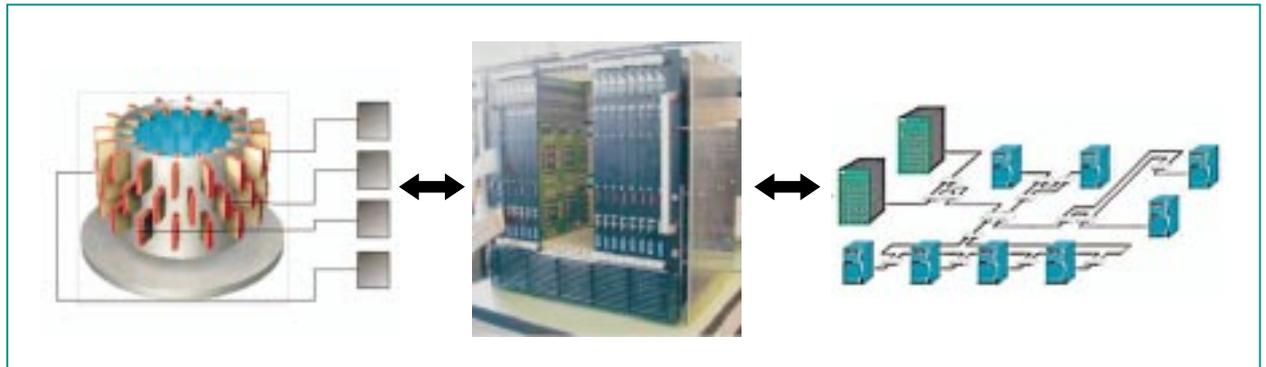


Abb. 1: Ultraschallcomputertomograph mit den Komponenten Sensortopf, Auswerteelektronik und Grid. Die Auswerteelektronik wurde ursprünglich für das Teilchenphysikprojekt AUGER entwickelt.

Hilfe von Grid-Technologien möglich, Systemteile, die beispielsweise für das Projekt TOSKA [6] entworfen wurden, für die Projekte KATRIN [5], AUGER North [7] und für einen Ultraschallcomputertomographen (USCT) [8] zur Brustkrebsuntersuchung zu benutzen (Abb. 1). In allen diesen Fällen verwenden wir eine Datenbank, einen Web-Applikationsserver und

ein Kontrollsystem, die über einheitliche Schnittstellen und einen Standard-Datentransfer miteinander kommunizieren (Abb. 2). Der Übergang zu den neuen Grid-Technologien ist für diese Aufgaben mit vernünftigem Aufwand machbar, da diese sich stark auf die von uns bereits bisher verwendeten Internet-Technologien beziehen. Weitere Anwendungen

des IPE, die mit denselben Technologien behandelt werden, kommen aus der Proteomik (Vorhersage der dreidimensionalen Proteinstruktur durch Optimierungsverfahren – zusammen mit INT und IAI [9]; Protein-Protein-Interaktionen, insbesondere bei der Hefe – zusammen mit ITG [10]; und Proteinanalytik im Chip-Format – zusammen mit IFIA [11]).

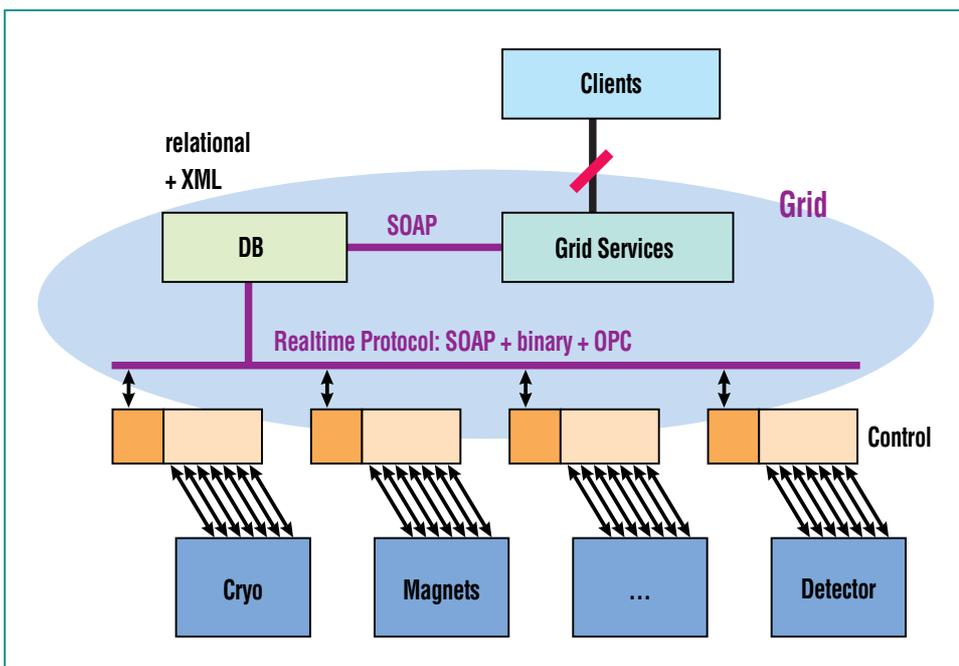


Abb. 2: Grid-Komponenten mit Subsystemen (hier für KATRIN), Kontrollsystem, Datenbank und Web-Applikationsserver.

Herausforderungen für die Forschung

Die Forschungsaktivitäten des IPE konzentrieren sich auf Erweiterungen der derzeitigen Grid-Middleware um Echtzeit-Komponenten. Ziel ist dabei, auf entstehende Spezifikationen einzuwirken bzw. entsprechende Ergänzungsspezifikationen für Echtzeit-Anwendungen in die entsprechenden Gremien einzubringen. Eine gute Basis dafür ist die Spezifikation OGSA-DAIS, welche sich zur Zeit im Diskussionsstadium befindet (siehe Abb. 3). IPE schlägt dafür eine Erweiterung vor (Spezifikation für Echtzeit-Datenintegration: OGSA-R(ealtime)DAIS), die sich bereits in Arbeit befindet und folgende Punkte umfasst:

1. Schnelles Standard-Protokoll für den Datentransfer

– basierend auf den Grid-Standards OGSA [2] und WSRF [12]

– basierend auf dem OPC-XML-Standard [13]

– basierend auf den binären Transportprotokollen XDR [14] und P BIO [15]

2. Datenschema für Experimentdaten (Primärdaten, relational) und Verwaltungsdaten (Metadaten, XML [16]) getrennt, aber durch einheitlichen Zugriff

3. Einbinden von Legacy-Systemen (Altsysteme) in das Grid durch standardisierte Datentransformationen (sog. Stylesheet-Transformationen XSLT [17] und offene Frameworks wie Struts [18] und Chiba [19])

4. Einbinden von Arbeitsplatzrechnern in Virtuelle Organisationen (VO) [2] durch den Benutzer

Im Folgenden werden diese Aktivitäten erläutert.

1. Datentransferprotokoll

Das derzeit einzige Datenprotokoll GridFTP [20] muss ergänzt werden, da es nicht echtzeitfähig und nicht ereignisgesteuert ist und somit für asynchron gesendete kleine bis mittelgroße Datenpakete nicht in Frage kommt. Interessant ist die Kombination von Standards, die aus der Automatisierungsindustrie kommen (OPC XML) [13], mit dem Web Service Standard SOAP [21] und der Erweiterung auf binäre Daten, um so ein standardisiertes, flexibles

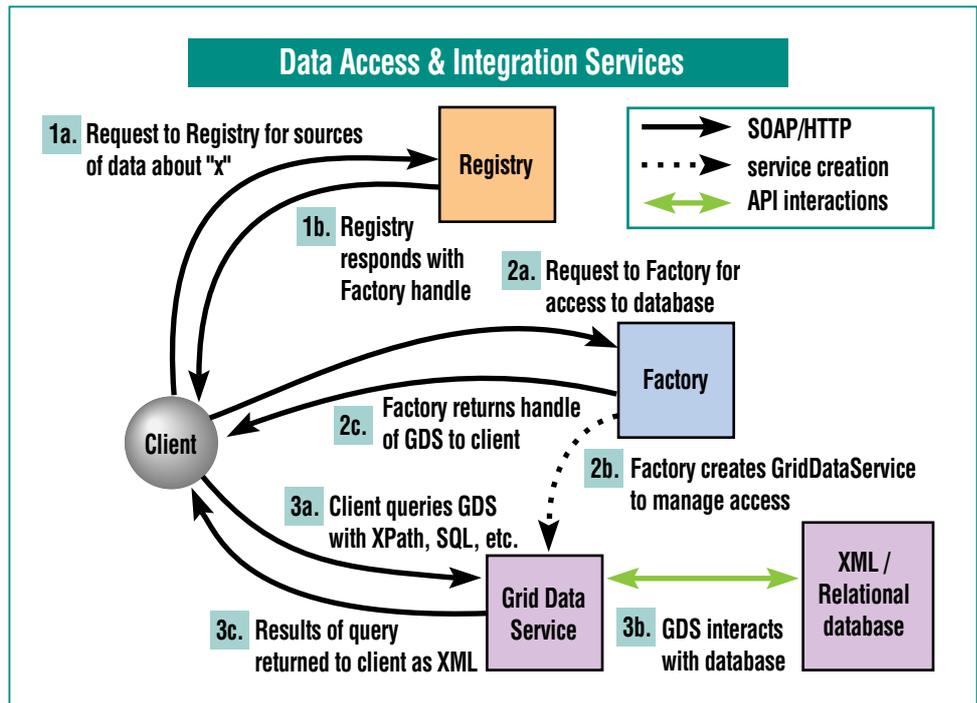


Abb. 3: DAIS-Ablauf für eine Datenbank-Abfrage.

und dennoch äußerst effektives Datentransferprotokoll zu schaffen. Weitere Details hierzu finden sich in [22].

2. Einheitlicher Datenzugriff

In praktisch allen relevanten Anwendungen, die mit Datenbeständen zu tun haben, gibt es die Unterscheidung in Primär- und Metadaten (bzw. Nutz- und Verwaltungsdaten). Die Metadaten beschreiben dabei die Struktur der Primärdaten und geben ihnen einen gewissen Sinn. Bei technischen Experimenten entsprechen die Primärdaten den Messdaten und die Metadaten sowohl den Parameterdatensätzen von Gerätekonfigurationen als auch Verwaltungsinformationen zu Benutzern oder Sicherheitseinstellungen. Die Primärdaten sind in der Regel homogen strukturiert, die Metadaten sehr heterogen. Man

spricht von strukturierten bzw. unstrukturierten Daten [23], was aber irreführend ist, da z.B. Texte sehr wohl eine gewisse Struktur aufweisen. Mit (homogen) strukturierten Daten sind in Tabellen organisierte Datenreihen gemeint, die in relationalen Datenbanksystemen mit SQL (Structured Query Language) als Zugriffssprache sehr effizient organisiert werden können. Heterogen strukturierte Daten sind z.B. Beschreibungen, denen auch Strukturelemente fehlen können. Sie werden am besten mit baumartigen Strukturen dargestellt, was direkt auf die moderne Datenbeschreibungssprache XML (eXtensible Markup Language) und die Zugriffssprache XQuery [24] deutet. Vorteil ist die flexible Struktur, die gegenüber der Organisation in Tabellen sehr leicht Änderungen zulässt. Nachteil ist der große Overhead, da die Struktur

beschreibenden Elemente direkt in die Daten eingebettet sind. Für die Darstellung von Metadaten ist diese Art also wie geschaffen.

Für den Entwurf modularer Systeme wäre es sehr hinderlich, für die verschiedenen Datensorten jeweils unterschiedliche Zugriffsmechanismen zu verwenden. Für die Darstellung von Ergebnissen auf einer Web-Seite sind meist beiderlei Datentypen vonnöten. Eine unterschiedliche Behandlung verkompliziert das System drastisch. OGSA-DAI legte für einen integrierten Zugriff den ersten exzellenten Vorschlag vor. Er definiert allerdings nur die *Möglichkeit* des Zugriffs auf unterschiedliche Datenbanktypen. Wir wollen einen Schritt weitergehen und – ausgehend von der Datenakquisition – die grundsätzliche Datenorganisation durch Basisfor-

mate festlegen. Grundlegend ist das XML Schema [25], dessen Basisdatentypen für diese Aufgaben erweitert werden. Für spezifische Anwendungen werden weitere Datentypen benötigt, die aus den Basisdatentypen abgeleitet werden. Dies wird für die Datenakquisition in einer zweiten Stufe weiter betrieben, so dass für ein spezielles Experiment wie KATRIN in einer dritten, applikationsspezifischen Schicht auf beiden vorangegangenen Stufen aufgebaut werden kann. In Abb. 4 ist der Basisdatentyp für die Administrationsdaten von KATRIN zu sehen. Ziel der Erforschung einheitlicher Datentypen ist die Erzeugung von Standardschnittstellen bis weit in die Anwendungen hinein, um ein dynamisches Plug-and-Play von Diensten zu erreichen.

3. Einbinden von Legacy-Systemen

Obwohl die Integration von Diensten das Ziel von Grid-Computing ist, kann bei den wenigsten Entwürfen neuer Systeme davon ausgegangen werden, dass alle Systemkomponenten tatsächlich einer einheitlichen Schnittstellenbeschreibung gehorchen. Die Zahl der Aufgaben und Hersteller ist einfach zu groß und die Komplexität mancher Systeme zu hoch, um sie an die neuen Standards anzupassen. Mit den neuen Web- und Grid-Technologien können solche Alt- oder Legacy-Systeme so verpackt werden, dass sie sich nach außen wie eine Standardkomponente verhalten. Üblicherweise werden auch Legacy-Systeme laufend verbessert und verändert, was ohne intelligente Um-

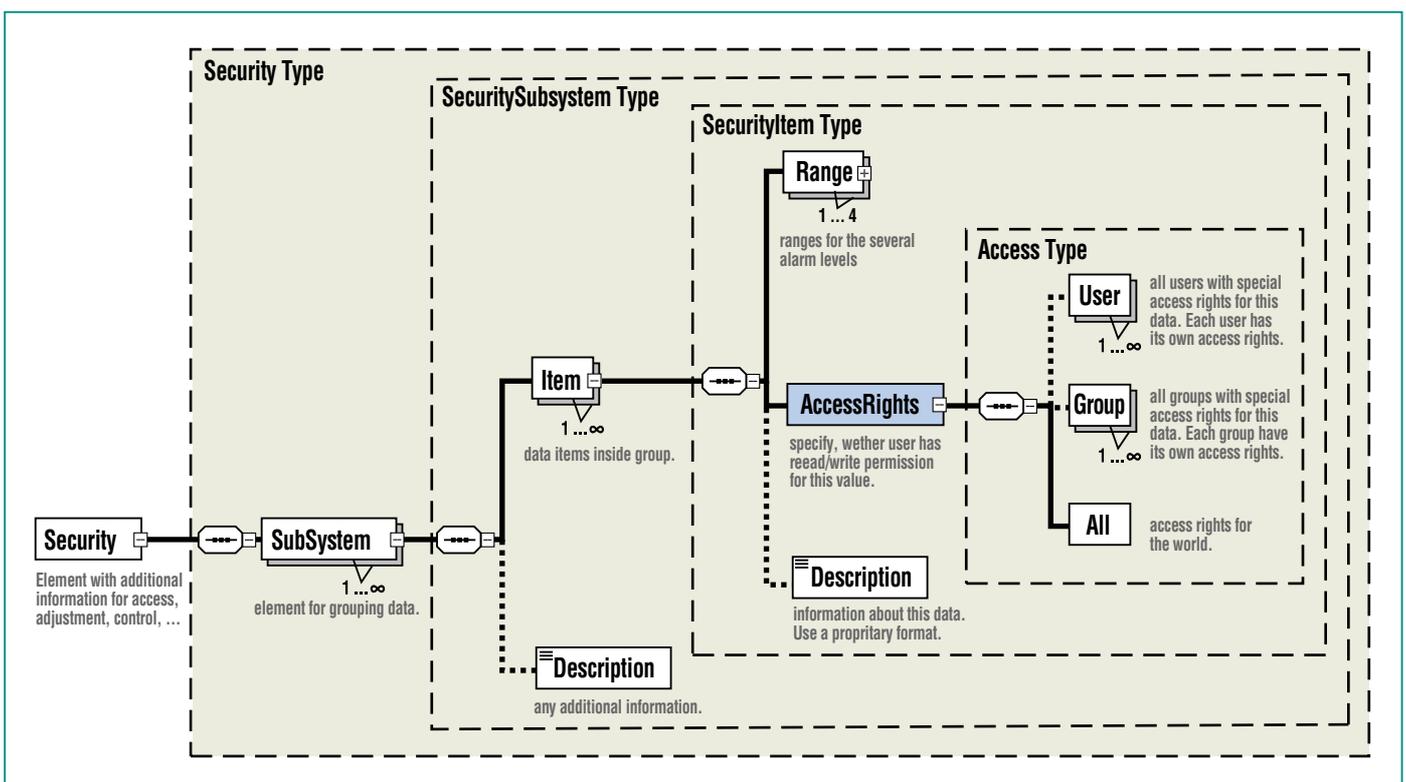


Abb. 4: XML-Schema für die Verwaltungsdaten von KATRIN.

setzungstechniken fehleranfällige Änderungen der Verpackung nach sich ziehen kann. Dieses Problem wird umgangen, wenn eine grundsätzliche Transformation des einen abstrakten Datenformats in das andere beschrieben wird. Für das Projekt KATRIN wurde diese Methode bereits umgesetzt und implementiert. Sie lässt sich verallgemeinern immer dann, wenn eine XML-Beschreibung des Legacy-Datenformats vorliegt. Bei KATRIN ist das Kontrollsystem LabView [26] die Legacy-Komponente, welche nach insgesamt vier Transformationen dem übrigen System dynamisch angepasst wird [27].

4. Einbinden von Arbeitsplatzrechnern in Virtuelle Organisationen (VO)

Es gibt zur Zeit zwei divergierende Richtungen des Grid-Computings. Die eine, hier vorwiegend dargestellte Herangehensweise, konzentriert sich auf die Standardisierung von Grid-Komponenten und deren Zusammenschluss zu einem großen Grid. Dies muss mehrere Aufgaben und Benutzer gleichzeitig bedienen können, die sich nicht gegenseitig stören dürfen. Dafür wurden so genannte Virtuelle Organisationen (VO) [2] geschaffen, die eine logische Auf-

teilung der gemeinsamen physikalischen Ressourcen vornehmen. Die andere Herangehensweise wird durch Projekte wie seti@home [28] oder zetaGrid [29] repräsentiert, die – vorwiegend als Bildschirmschoner – versuchen, möglichst viele brachliegende Arbeitsplatz-PCs zu nutzen. Wir wollen eine Synthese beider Richtungen vorantreiben, um die begrenzten Grid-Ressourcen bei Einhaltung von Standards und Sicherheitsvorkehrungen zu vergrößern. Dafür muss das Konzept der VO erweitert werden, so dass neue Ressourcen dynamisch eingebunden werden können. Damit ist möglich, für bestimmte Zeiten ungeheure Rechen- und Speicherkapazitäten preisgünstig zu nutzen. Dies wird von uns (in Zusammenarbeit mit INT und IAI) exemplarisch für die Vorhersage dreidimensionaler Proteinstrukturen angegangen [9].

Ausblick

Grid-Computing stellt eine gewaltige Herausforderung dar. Für die reibungslose Integration unterschiedlichster Daten und Dienste ist die Durchformalisierung (bzw. „Ontologisierung“) nahezu aller Wissensbereiche nötig, die stark untereinander vernetzt sind. Dies wird zunächst nur für spezielle,

von einander entkoppelte Anwendungsbereiche funktionieren. Die Standardisierung schreitet zügig voran, kann aber nicht verhindern, dass sich manche Standards nicht oder nur zögerlich durchsetzen und immer wieder korrigiert werden. Wir müssen uns daher immer wieder mit veränderten oder halbfertigen Softwarewerkzeugen beschäftigen. Diese zählen zur Infrastruktur, welche immer komplexer wird und immer mehr Zeit und Personal zu ihrer Verwaltung erforderlich macht.

Die Chancen zur Weiterentwicklung des Grid-Computing sind allerdings groß. Die sich entwickelnden Standards werden im Internet offen diskutiert, jeder kann Vorschläge unterbreiten und mit kompetenten Experten diskutieren. Die ungeahnten Möglichkeiten durch vereinheitlichten Zugriff auf Daten und Dienste verursachen eine Aufbruchstimmung, die sehr produktiv ist. Wir nützen sie für wichtige Experimente, die im Forschungszentrum durchgeführt werden.

Literatur

- [1] GGF: <http://www.ggf.org>
- [2] I. Foster, C. Kesselman, J. Nick, S. Tuecke, *Global Grid Forum, June 22, 2002*, Available: <http://www.globus.org/research/papers/ogsa.pdf>
- [3] M. Antonioletti, et al., Available: <http://www.ogsadai.org.uk/>
- [4] *The Globus Alliance*, Available: <http://www-unix.globus.org/toolkit/>
- [5] A. Osipowicz et al., *Letter of Intent*, *Los Alamos e-print archive* <http://xxx.lanl.gov/abs/hep-ex/0109033>
- [6] TOSKA: http://hikwww4.fzk.de/itp/toska/d_index.html
- [7] AUGER North: http://www.hpe.fzk.de/projekt/auger/auger-north/e_index.html
- [8] USCT: http://www.ipe.fzk.de/projekt/med/usct/d_index.html

-
-
- [9] W. Eppler, V. Hartmann, *2nd Workshop on Information Technology and its Disciplines, WITID 2004, Kish, Iran*
- [10] J. Silva, R. Stotzka, N. Ruitter, P. Uetz, *In Bildverarbeitung für die Medizin 2004, Informatik Aktuell, 2004*
- [11] M. Kautt, J. Reichert, W. Hoffmann, M. Strasser, M. Rapp, A. Voigt, K. Länge, K.-F. Weibezahn, E. Gottwald, *Nachrichten, Jahrgang 34 · 2-3/2002*
- [12] K. Czajkowsky et al., *Version 1.0, Available: <http://www.globus.org/wsr/>, 12 Feb 2004*
- [13] *OPC Foundation, June 2002, Available: <http://www.opcfoundation.org>*
- [14] B. Meyers, G. Chastek, *Carnegie Mellon Univ, Software Engineering Institute, Technical Report CMU/SEI-93-TR-10 October 1993*
- [15] F. Bustamante, G. Eisenhauer, K. Schwan, P. Widener, *Proc. ACM/IEEE conference on Supercomputing 2000*
- [16] T. Bray, J. Paoli, C. Sperberg-McQueen, *February 1998. Available: <http://www.w3.org/TR/REC-xml>*
- [17] J. Clark, *November 1999. Available: <http://www.w3.org/TR/xslt>*
- [18] *Struts: <http://jakarta.apache.org/struts>*
- [19] J. Turner, U. N. Lissé, *2003, <http://chiba.sourceforge.net/>*
- [20] W. Allcock, J. Bester, J. Bresnahan, A. Chervenak, L. Liming, S. Tuecke, *<http://www-fp.mcs.anl.gov/dsl/GridFTP-Protocol-RFC-Draft.pdf>, 2001*
- [21] *W3C, September 2002. Available: <http://www.w3.org/TR/soap12-af/>*
- [22] W. Eppler, A. Beglarian, S. Chilingarian, S. Kelly, V. Hartmann, H. Gemmeke, *IEEE TNS, Vol. 51, No. 3, June 2004*
- [23] S Banerjee, *The Twelfth International World Wide Web Conference, 2003, Budapest, HUNGARY, Available: www2003.org*
- [24] *XQuery: <http://www.w3.org/XML/Query>*
- [25] D. Fallside, *May 2001. Available: <http://www.w3.org/TR/xmlschema-0/>*
- [26] *LabView: <http://www.ni.com/labview/>*
- [27] W. Eppler, V. Hartmann, *Multilateral Workshop on Intelligent Information Technology in Control and Data Processing, IITinCDP 2004, Tehran, Iran*
- [28] *seti@home: <http://setiathome.ssl.berkeley.edu/>*
- [29] *zetaGrid: <http://www.zetagrid.net/>*