

GWGD-Bericht Nr. 74

Ulrich Schwardmann (Hrsg.)

## **Grid-Technologie in Göttingen**

**Beiträge zum  
Grid-Ressourcen-Zentrum GoeGrid**

Ulrich Schwardmann (Hrsg.)  
Grid-Technologie in Göttingen  
Beiträge zum  
Grid-Ressourcen-Zentrum GoeGrid



Ulrich Schwardmann (Hrsg.)

# Grid-Technologie in Göttingen

**Beiträge zum  
Grid-Ressourcen-Zentrum GoeGrid**

GWDG-Bericht Nr. 74

Gesellschaft für wissenschaftliche Datenverarbeitung mbH Göttingen

© 2009

*Gesellschaft für wissenschaftliche Datenverarbeitung mbH Göttingen*

*Am Faßberg 11*

*D-37077 Göttingen*

*Telefon: 0551 201-1510*

*Telefax: 0551 201-2150*

*E-Mail: [gwdg@gwdg.de](mailto:gwdg@gwdg.de)*

*Satz: Gesellschaft für wissenschaftliche Datenverarbeitung mbH Göttingen*

*Druck: Goltze Druck, Göttingen*

*ISSN 0176-2516*

---

---

## Inhalt

Vorwort	1
Das Göttinger Grid-Ressourcen-Zentrum GoeGrid <i>Cano Ay, Jörg Meyer, Arnulf Quadt, Christian Boehme, Oswald Haan, Ulrich Schwardmann</i>	5
Ein interdisziplinäres Grid-Anwenderpraktikum basierend auf Instant-Grid <i>Thomas Rings, Fred Viezens, Jörg Meyer, Andreas Aschenbrenner</i>	19
Bericht aus MediGRID – Implementierung eines Testbeds mit Instant-Grid für die perspektivische Nutzung einer Pseudonymisierungskomponente <i>Fred Viezens, Ulrich Sax, Kai Lorberg</i>	29

A two-tier approach to efficient workflow scheduling in  
MediGRID

*Dietmar Sommerfeld, Harald Richter*

39

Instant-Grid – eine Live-CD für die Demonstration, die  
Entwicklung und den Test von Grid-Anwendungen

*Ulrich Schwardmann*

53

---

---

## Vorwort

In verschiedenen wissenschaftlichen Einrichtungen in Göttingen hat sich im Laufe der letzten Jahre ein hohes Maß an Kenntnis im Betrieb und in der Nutzung von Grid-Technologie herausgebildet. Mit MediGRID, einem Community-Projekt, das den Schwerpunkt des Grid-Einsatzes auf die Unterstützung von bioinformatischen und biomedizinischen Forschungsaufgaben legt, und TextGrid, einem geisteswissenschaftlichen Grid-Projekt, das am Aufbau der e-Humanities-Infrastruktur im nationalen und europäischen Rahmen mitwirkt, sind schon zu Beginn der D-Grid-Initiative zwei der Community-Projekte von Göttinger Forschungseinrichtungen geleitet worden. Die GWDG, als weiterer Partner der Göttinger Grid-Projekte, hat sich seit 2004 intensiv mit der Grid-Technologie befasst und im Rahmen der D-Grid-Initiative an verschiedenen Grid-Projekten teilgenommen. Insbesondere durch das ebenfalls durch das BMBF geförderte Instant-Grid-Projekt wurden bei der GWDG umfassende Kenntnisse der Grid-Technologie aufgebaut. Mit der Einrichtung des Forschungsschwerpunktes Hochenergiephysik an der Fakultät für Physik der Universität Göttingen ist das Knowhow zu Grid-Technologien in Göttingen um einen weiteren Bereich erweitert worden.

Dieses Wissen ist nicht in den einzelnen Bereichen isoliert geblieben, sondern es wurde schon sehr früh Wert darauf gelegt, durch einen regen Austausch der Erfahrungen, Informationen und Kenntnisse einen größtmöglichen Synergieeffekt für alle Beteiligten zu erzielen. Bereits in der Phase des D-Grid-Vorprojektes hatten die verschiedenen Göttinger Grid-

Projekte umfangreich zusammengearbeitet. Kurz nach Start des D-Grid wurde die gemeinsame Kooperation gefestigt und verstetigt: In regelmäßigen Treffen der Partner fand der Austausch über relevante Entwicklungen im Grid-Umfeld statt. Gemeinsame Veranstaltungen wurden organisiert und Infrastruktur wurde gemeinsam beantragt und ausgebaut. Dieses Bündnis, das sich alsbald den Namen GoeGrid gab, bekam einen zusätzlichen Schub, als im Jahre 2007 mit der Einrichtung des Forschungsschwerpunktes Hochenergiephysik an der Fakultät für Physik der Universität Göttingen ein starker Partner einer weiteren Grid-Community in Göttingen hinzukam. Der vorliegende GWDG-Bericht soll einen Einblick in die verschiedenen Aktivitäten dieser Projekte und Gruppen geben.

Die steigenden Anforderungen an Rechen- und Speicherkapazität erforderten zum Einen einen starken Ausbau der Grid-Ressourcen in Göttingen, zum Anderen wurde hiermit eine stärkere Formalisierung des Bündnisses unerlässlich. Dies findet seinen Ausdruck in der Einrichtung und im Betrieb des Grid-Ressourcen-Zentrums GoeGrid in Göttingen, in dem sowohl die personelle wie auch die technologische Verzahnung der verschiedenen Göttinger Grid-Projekte zusammengefasst sind. Die Entwicklung von GoeGrid vom informellen Treffen bis zum Grid-Ressourcenzentrum mit einer beträchtlichen Ressourcenbasis wird im Beitrag „Das Göttinger Grid-Ressourcen-Zentrum GoeGrid“ beschrieben.

Die Heranführung von Wissenschaftlern an die Grid-Technologie wurde im Rahmen von GoeGrid immer betont. Zunächst waren Block-Veranstaltungen der geeignete Weg, die Gridtechnologie den Göttinger Wissenschaftlern nahe zu bringen. Später wendeten sich auch interdisziplinäre Lehrveranstaltungen speziell an den Wissenschaftler-Nachwuchs, der so bereits in der Ausbildung an die Arbeitsweise im Grid herangeführt wurden. Der Beitrag „Ein interdisziplinäres Grid-Anwenderpraktikum basierend auf Instant-Grid“ beschreibt die Erfahrungen mit der Durchführung eines Grid-Praktikums, das im Wechsel mit einem Grid-Seminar stattfindet, auf den Ergebnissen dieses Seminars aufbaut, und dessen erklärtes Ziel es ist, die Interdisziplinarität des Grids und seiner Nutzung auch in der Ausbildung zu betonen. Dieses Praktikum greift dabei auf die Instant-Grid-Technologie zurück, was wiederum die Vernetzung der Göttinger Grid-Gruppen dokumentiert.

Im Beitrag „Bericht aus MediGRID - Implementierung eines Testbeds mit Instant-Grid für die perspektivische Nutzung einer Pseudonymisierungskomponente“ geht es um eine Problematik, die speziell bei der Verarbeitung von Daten auftritt, die in ihrem Ursprung personenbezogen sind, aber deren Verwendung im Grid diesen Personenbezug verlieren soll oder muss. Um

sich dieser Fragestellung zu nähern, wurde dabei als Testbed ein verteiltes Grid-System mittels Instant-Grid eingerichtet und in diesem Testbed die Realisierbarkeit eines entsprechenden Pseudonymisierungsverfahrens nachgewiesen. In diesem Beitrag wird unter anderem auch durch die Verwendung von Instant-Grid die Vernetzung der verschiedenen Göttinger Grid-Gruppen sichtbar.

In dem Beitrag „A two-tier approach to efficient workflow scheduling in MediGRID“ geht es um die Frage, wie die Teilaufgaben eines wissenschaftlichen Anwendungs-Workflows in einem Grid auf die vielfältigen, verteilten Ressourcen abgebildet und so gestartet werden, dass der gesamte Workflow mit seinen internen Abhängigkeiten optimal und flexibel orchestriert wird. Diese Fragestellung ist für die Optimierung aller komplexen Abläufe im Grid relevant, da nur mit einer hierin befriedigenden Lösung die voneinander abhängigen Teilaufgaben schnellstmöglich berechnet werden können, ohne dabei Ressourcen unnötig im voraus zu blockieren und damit unter Umständen leerlaufen zu lassen.

„Instant-Grid – eine Live-CD für die Demonstration, die Entwicklung und den Test von Grid-Anwendungen“ ist ein Bericht über die Entwicklung und die Eigenschaften der Instant-Grid-CD. Wie schon in den vorangegangenen Beiträgen zu erkennen gewesen ist, kann diese Technologie auf einfache Weise für verschiedene Lehr- und Forschungszwecke angewandt werden. Der Bericht beschreibt die Technologie der Konfiguration der verteilten Systeme sowie die Software-Komponenten und Grid-Middleware, die die Grundlage des Demonstrations-Grid bilden. Die reiche Auswahl an Anwendungsszenarien und Grid-Applikationen wie auch die erfolgreichen Demonstrationen bei verschiedenen Gelegenheiten sind ebenfalls Gegenstand dieses Beitrags. Abgerundet wird dies durch einen Anhang voll mit zum Teil begeisterten Rückmeldungen über den Nutzen dieses Grid-Demo-Toolkits.

Die Arbeiten wie auch die Ressourcen im Bereich GoeGrid, die in diesem Bericht zum Thema gemacht werden, wurden zum großen Teil durch das Bundesministerium für Bildung und Forschung (BMBF) gefördert, dem an dieser Stelle dafür gedankt sein soll.

Göttingen, Februar 2009

Ulrich Schwardmann



---

---

# **Das Göttinger Grid-Ressourcen-Zentrum GoeGrid**

**Cano Ay, Jörg Meyer, Arnulf Quadt**

*II. Physikalisches Institut, Georg-August-Universität Göttingen*

**Christian Boehme, Oswald Haan, Ulrich Schwarzmann**

*Gesellschaft für wissenschaftliche Datenverarbeitung mbH Göttingen*

## **Zusammenfassung**

GoeGrid ist der Name des gemeinsamen Grid-Ressourcen-Zentrums der in Entwicklung und Nutzung von Grid-Technologien eingebundenen Göttinger Forschungsgruppen, dessen Ressourcen bei der GWDG betrieben werden.

Beschaffung, Administration und Nutzung der Ressourcen erfolgen gemeinsam durch die beteiligten Partner. Diese fächerübergreifende Nutzung von Grid-Ressourcen und deren Einbindung in die nationale D-Grid-Infrastruktur und in das WLCG-Projekt bringen große Vorteile im Hinblick auf Beschaffungskosten, Betreuungsaufwand und Auslastung.

## **Abstract**

GoeGrid is a joint grid resource centre of research groups in Göttingen with grid related activities, operated within the GWDG. All partners of GoeGrid are jointly involved in funding, administrating and using the resources. The

interdisciplinary use of grid resources and the integration of these resources in Germany's national D-Grid infrastructure and the WLCG project result in great advantages regarding acquisition costs, operating expenses and resource utilization.

## **1. Einleitung**

Der Einsatz der Grid-Technologie verspricht eine weitgehende Vereinfachung für den Zugang zu Rechenressourcen, Forschungsdaten und Anwendungen. Diese Vereinfachung beruht in erster Linie darauf, dass die für die Datenverarbeitung benötigten Ressourcen in Form von Diensten bereitgestellt werden, die unabhängig vom Standort des Nutzers mit identischen Schnittstellen aus dem Internet abgerufen werden können. Das Vorbild für diese Art von Dienstangebot ist das Stromnetz, aus dem von überall her mit genormter Dienstleistung in Form von 220-Volt-Steckdosen die Ressource Strom abgenommen werden kann. Dieses Vorbild hat auch den Begriff Grid, der im Englischen für das Stromnetz steht, geprägt.

Der Vergleich mit dem Stromnetz wird allerdings ungenau, wenn die inhaltliche Bestimmung der über das Datenverarbeitungs-Grid bereitgestellten Dienste betrachtet wird. Anders als beim Stromnetz, über das die einheitliche, von allen Endanwendungen nutzbare Leistung Strom verteilt wird, erwartet der Endanwender vom Grid nicht nur reine Rechenleistung und Speicherkapazität, sondern auch die Bereitstellung von für seine Arbeiten wichtigen Anwendungen und Datenbeständen. Die Ausgestaltung der Dienste wird somit spezifisch für unterschiedliche Anwendergruppen ausdifferenziert werden müssen, und bei der Spezifikation und Erstellung der Dienste ist die Mitwirkung der verschiedenen Anwendergruppen von entscheidender Bedeutung.

Die Etablierung eines Grids für die Datenverarbeitung erfordert deshalb zwei Schritte. Zum einen muss eine Infrastruktur von Rechenleistung und Speicherkapazität geschaffen werden, auf die Nutzer über geeignete Software-Komponenten zugreifen können. Diese sog. Grid-Middleware stellt die Mechanismen für die Einhaltung vertraglicher Regelungen zur Grid-Nutzung bereit, wie sichere Zugangsmechanismen, Überprüfung von Nutzungsberechtigungen, Buchführung über abgenommene Leistungen und gegebenenfalls Abrechnung für geleistete Dienste. In einem weiteren Schritt erfolgt der Zusammenschluss von Anwendern mit gemeinsamen Anforderungen zu so genannten „virtuellen Organisationen“, die die jeweils benötigten Dienste spezifizieren und i. d. R. auch für deren Realisierung auf Basis der vorhandenen Grid-Middleware sorgen.

In Deutschland fördert als gemeinsame Initiative mit der deutschen Wissenschaft und Wirtschaft das Bundesministerium für Bildung und Forschung (BMBF) den Aufbau eines solchen Grids, des D-Grid (siehe auch [1]). In einer ersten Förderstufe wurde seit 2005 der Aufbau einer D-Grid-Infrastruktur realisiert. Diese stellt die für die Grid-Nutzung notwendigen Basisdienste bereit. Die dahinterliegenden Rechner- und Speicherressourcen wurden zunächst von den beteiligten Institutionen zur Verfügung gestellt. In den Jahren 2006, 2007 und 2008 konnten die Grid-Ressourcen in erheblichem Maße durch Sonderfördermaßnahmen des BMBF erweitert werden. Zeitgleich mit dem Aufbau der D-Grid-Infrastruktur förderte das BMBF in bisher drei Förderrunden die Bildung sog. D-Grid-Communities, die als virtuelle Organisationen die community-spezifischen Dienste definieren und aufbauend auf die D-Grid-Infrastruktur zur Einsatzreife bringen sollen. Nachdem die Initiative im Jahr 2005 mit den zunächst sieben Communities AstroGrid (Astronomie), C3-Grid (Meteorologie), HEP-Grid (Hochenergiephysik), InGrid (Ingenieurwissenschaften), MediGRID (Medizin, Biologie), TextGrid (Geisteswissenschaften) und WISENT (Energiemeteorologie) gestartet ist, gibt es inzwischen ca. 20 D-Grid-Communities aus unterschiedlichen Bereichen der naturwissenschaftlichen und auch geisteswissenschaftlichen Forschung sowie aus Bereichen industrieller Entwicklung und Produktion.

Auch in Göttingen gab es bereits in der Vorab-Phase von D-Grid ein starkes Engagement zum Aufbau einer Grid-Infrastruktur in Deutschland. Die Konsortialleitungen zweier der Startup-Communities von D-Grid, MediGRID und TextGrid sind in Göttingen angesiedelt.

Die GWDG hatte sich seit 2004 intensiv mit der Grid-Technologie befasst und die D-Grid Initiative aktiv unterstützt. Später kam mit dem Ausbau der Hochenergie-/Teilchenphysik an der Georg-August-Universität Göttingen die starke Beteiligung an einer dritten Grid-Community hinzu. Aus dem gemeinsamen Interesse an einer Bündelung dieser Aktivitäten entstand das Göttinger Grid-Ressourcen-Zentrum GoeGrid.

## **2. Die Gründung von GoeGrid**

Schon kurz nach dem Start der D-Grid-Initiative gab es erste gemeinsame Treffen der GWDG mit Vertretern der Communities MediGRID und TextGrid, um sich gegenseitig über die geplanten Aktivitäten zu informieren, um gemeinsam die Wahrnehmung der neuen Grid-Technologie bei den Göttinger Wissenschaftlern zu verstärken, und um den Betrieb der im Grid bereitzustellenden Ressourcen zu organisieren. Mit dem Ausbau der Hochenergie-/Teilchenphysik an der Universität Göttingen kam später als

dritte Community die Teilchenphysik zu den Beteiligten hinzu. Für diese Aufgaben wurde zunächst ein informeller Rahmen geschaffen, der sich den Namen GoeGrid gab. Anders als beim Strom, für dessen Bereitstellung und Verteilung über das Netz sich ein Geschäftsmodell etabliert hat, von dem viele große Stromkonzerne mit guter Rendite leben, gibt es für die Bereitstellung von Grid-Ressourcen noch kein kommerziell tragfähiges Geschäftsmodell und folglich noch keine kommerziellen Anbieter von Grid-Ressourcen. Vielmehr hat das BMBF die Nutzer des D-Grids, also die D-Grid-Communities, in den Jahren 2006, 2007 und 2008 mit besonderen Fördergeldern unterstützt, um die von diesen für ihre Grid-Nutzung benötigten Rechner und Speichermedien beschaffen zu können. Dabei wurde die Frage nach der Beschaffung, dem Betrieb und der Integration der neuen Ressourcen in das D-Grid nur insoweit berührt, dass eine Lösung dieser Fragen den D-Grid-Communities obliegt. Praktisch hatte das zur Folge, dass nur solche Communities die Ressourcenförderung im Anspruch nehmen konnten, die für diese Aufgaben der Beschaffung, des Betriebs und der D-Grid-Integration auf die Unterstützung durch Rechenzentren oder vergleichbare Institutionen zurückgreifen konnten. Für die Bewältigung dieser Anforderung war die in Göttingen sich etablierende Struktur des GoeGrid besonders gut geeignet, da hier die Vertreter der Communities mit den Vertretern eines Ressourcenproviders, der GWDG, an einem Tisch saßen.

In Zusammenarbeit mit der GWDG haben die Göttinger Gruppen dieser drei Communities jeweils gemeinsame Anträge auf Sonderfördermaßnahmen des BMBF für die Finanzierung von Grid-Ressourcen gestellt. Zudem wurden Berufungsmittel der Theorie des Fachbereichs Physik verwendet (Prof. Müller, Prof. Pruschke). Die GWDG übernahm die Beschaffung der Ressourcen und integrierte sie soweit wie möglich in den Betrieb der sonstigen für die Universität Göttingen und die Max-Planck-Gesellschaft bereitgestellten Ressourcen. Die gemeinsame Beschaffung von Ressourcen für unterschiedliche Communities und deren Integration in die bei der GWDG vorhandene Infrastruktur brachten erhebliche Kostenvorteile bei Beschaffung und Betrieb mit sich.

Mit diesen Beschaffungen aus den Sonderinvestitionen und Berufungsmitteln wurde im Lauf der Jahre Infrastruktur bereitgestellt, die GoeGrid zu einem Zentrum für Grid-Ressourcen für verschiedene Grid-Communities gemacht hat. GoeGrid wurde durch Berufungsmittel der Teilchenphysik erweitert (Prof. Klute, Prof. Quadt). Weitere Mittel der Teilchenphysik kamen aus der Helmholtz-Allianz „Physics at the Terascale“ (siehe auch [2]).



Die Zusammenarbeit von GWDG mit den Göttinger Nutzergruppen des Grids wurde Anfang 2008 durch die Gründung des Göttinger Grid-Ressourcen-Zentrums GoeGrid und dessen offizieller Einweihung im Mai 2008 einer breiten Öffentlichkeit präsentiert. Durch die Einrichtung eines monatlich tagenden Technical Board und eines bei Bedarf einzuberufenen Executive Boards haben die laufenden administrativen und allgemeineren organisatorischen Aufgaben einen formalen Rahmen bekommen. Bestandteil von GoeGrid sind Vereinbarungen zur Nutzungsverteilung der Ressourcen sowie zur Beteiligung an Bereitstellung und Wartung von nutzerspezifischen Softwareumgebungen.

Um die Vorzüge des Grids den Göttinger Wissenschaftlern nahe zu bringen, sind aus den GoeGrid-Treffen heraus die GoeGrid-Seminare organisiert worden, auf denen internationale Apologeten der Grid-Technologie über ihre Erfahrungen berichtet haben und auf denen die Göttinger Gruppen ihre Technologien vorstellen konnten. Später wurden diese Veranstaltungen durch interdisziplinäre universitäre Lehrveranstaltungen ersetzt, die sich speziell an die Studierenden wenden, und die die Einbeziehung von Grid-Technologien in die Arbeitsweise der Wissenschaftler bereits in einem frühen Stadium der Ausbildung etablieren sollen. So findet nun jeweils im Win-

tersemester das GoeGrid-Seminar statt, im Sommersemester werden die Kenntnisse in einem Praktikum vertieft (siehe [3]).

### **3. Die GoeGrid-Partner**

Göttinger Institutionen sind an den D-Grid-Communities MediGRID und TextGrid beteiligt sowie an WLCG(Worldwide LHC Computing Grid)-Grid-Communities der Teilchenphysik. Zudem sind im GoeGrid-Verbund als weitere Partner die Theorie des Fachbereichs Physik sowie die GWDG beteiligt.

#### **3.1 MediGRID**

MediGRID steht für den Einsatz des Grid für bioinformatische und biomedizinische Forschungsaufgaben, für bildgebende Verfahren in der medizinischen Diagnose und bei der Durchführung klinischer Studien. Für das Projekt wird ein modularer Aufbau gewählt: Drei in ihrer Anforderungslage repräsentative Fachbereiche in der biomedizinischen Forschung (Bildverarbeitung, Biomedizinische Informatik, Klinische Forschung) werden unterstützt von vier methodischen Modulen (Middleware, Ontologie-Werkzeuge, Ressourcenusion und eScience). Diese erarbeiten schrittweise eine Grid-Infrastruktur, während parallel dazu exemplarische Anwendungen aus den drei Forschungsbereichen in die prototypischen Grid-Strukturen übergeben werden. Die Verknüpfung der in Bezug auf Herkunft, Datentyp bzw. Datenmodell sowie Verfügbarkeit heterogenen Daten aus Grundlagenforschung, klinischer Forschung und Patientenversorgung soll bewältigt werden, indem die technischen und logischen Insellösungen der bestehenden Verbünde in eine übergreifende Lösung mit einer einheitlichen Arbeits-, Kommunikations-, Ressourcen- und Middleware-Struktur überführt werden. Zentrales Ziel der geplanten Maßnahmen ist eine Grid-unterstützte Verbesserung der interdisziplinären medizinischen Forschung. Zugleich soll durch die geplante Struktur der bilaterale Austausch mit dem Bereich der Patientenversorgung und der Versorgungsforschung unterstützt werden. Die Konsortialleitung des Projektes ist in Göttingen angesiedelt (siehe auch [4], [5]).

#### **3.2 TextGrid**

TextGrid ist das erste geisteswissenschaftliche Projekt, das am Aufbau der e-Humanities-Infrastruktur im nationalen und europäischen Rahmen mitwirkt. Indem TextGrid eine Datenhaltungsinfrastruktur, ein Forschungsnetzwerk und ein umfassendes und erweiterbares Repertoire an Werkzeugen für Geisteswissenschaftler zur Verfügung stellt, schafft es eine Grundlage für die sich entwickelnden e-Humanities. In TextGrid bauen Geisteswissenschaftler und Informatiker in enger Zusammenarbeit eine interdisziplinäre,

virtuelle Arbeits- und Forschungsplattform auf, wobei sie auf vorhandene Expertise im Bereich der e-Sciences zurückgreifen und einen wichtigen Schritt hin zum Semantic Grid vollziehen. Die TextGrid-Community stellt Grid-Dienste zur Bereitstellung, Bearbeitung, Analyse und Herausgabe großer Text-Korpora zur Verfügung. Die Niedersächsische Staats- und Universitätsbibliothek Göttingen (SUB) ist führend an diesem Grid-Projekt aus den Geisteswissenschaften in Deutschland beteiligt (siehe auch [6], [7]).

### **3.3 WLCG und Teilchenphysik**

Die WLCG-Grid-Communities der Elementarteilchen-Physik sind insbesondere darauf ausgerichtet, die riesigen, bei den Experimenten an Teilchenbeschleunigern entstehenden Datenmengen zu speichern und zu analysieren. Deswegen entwickelt die Teilchenphysik in internationalem Rahmen mit hoher Intensität ein Computing Grid. Die bisher entwickelten Werkzeuge werden schon jetzt für einige Aufgaben im Rahmen laufender und zukünftiger Experimente eingesetzt. Mit dem Anlaufen des LHC im Jahre 2008 am CERN in Genf (Schweiz) erreichen die Anforderungen an die Grid-Technologie eine neue Qualität.

Die nächste Generation von Teilchenphysikexperimenten (Alice, Atlas, CMS, LHCb) geht dabei am LHC-Beschleuniger des Europäischen Zentrums für Teilchenphysik CERN in Betrieb (siehe auch [8]). Sie werden Datenmengen von etwa 15 Petabyte pro Jahr produzieren und benötigen für ihre Analyse eine Rechenleistung von weltweit mehreren Zehntausend CPUs. Neben den gemessenen Daten werden simulierte Daten, so genannte Monte-Carlo-Ereignisse, generiert. Die Daten werden redundant und regional verteilt gespeichert und für die Analyse zur Verfügung gestellt. Auf diese Daten greifen dann Tausende von Wissenschaftlern aus allen Kontinenten zu. Die Entwicklung eines Grid für die Teilchenphysik geschieht innerhalb internationaler Initiativen und wird durch nationale Projekte unterstützt.

Das II. Physikalische Institut der Universität Göttingen ist Mitglied beim WLCG-Projekt (siehe auch [9]), das den Daten-Speicher und die Analyse-Infrastruktur für die LHC-Experimente bereitstellt. Es ist zudem Mitglied der internationalen ATLAS-Kollaboration. Die Universität Göttingen ist eine von 12 Universitäten und Max-Planck-Instituten, die dem Forschungsnetzwerk des BMBF-Forschungsschwerpunkts „Physics on the TeV Scale with ATLAS at the LHC“ (fsp-101) angehören (siehe auch [10]). ATLAS ist das größte Hochenergiephysik-Experiment am LHC. Das ATLAS-Experiment allein wird Daten in der Größenordnung von 300 MB/sec produzieren. GoeGrid nimmt an der weltweiten Bearbeitung der ATLAS-Daten teil und stellt ein so genanntes Tier-2- und Tier-3-Zentrum zur Verfügung. Für die

primäre Datenrekonstruktion und -verteilung zeichnet das Tier-0-Zentrum am CERN selbst verantwortlich. Die nationalen Tier-1-Zentren organisieren die Verteilung der Daten auf nationaler Ebene, in Deutschland übernimmt dies das Forschungszentrum Karlsruhe. Das Göttinger Tier-2-Zentrum arbeitet in einer Föderation mit dem DESY in Hamburg zusammen. Die wichtigste Anwendung dabei ist die Monte-Carlo-Simulation von Proton-Proton-Kollisionen. Die Ergebnisse dieser Simulationen werden lokal gespeichert und zum zuständigen Tier-1-Zentrum zurückgespielt. GoeGrid wurde im April 2008 vom WLCG bezüglich der Stabilität der Clusterimplementation und der installierten Middleware zertifiziert und konnte damit die Zulassung und den Status als Tier-2-Zentrum erlangen. Die Nutzung der LHC-Daten durch die lokalen Nutzer, i. A. die PhysikerInnen der Universität Göttingen, geschieht auf der Ebene des Tier-3-Zentrums.

### **3.4 Fachbereich Physik**

Der Theorie des Fachbereichs Physik der Universität Göttingen ist über die Aktivitäten in der Teilchenphysik hinaus ebenfalls an den Ressourcen von GoeGrid beteiligt und einer der Hauptnutzer mit Anwendungen, die Quanten-Monte-Carlo-Methoden oder numerische Renormalisierungsgruppen-Methoden nutzen. Eine andere Arbeitsgruppe untersucht die Physik weicher Materie anhand von Computersimulationen (siehe auch [11]).

### **3.5 GWDG**

Die GWDG (Gesellschaft für wissenschaftliche Datenverarbeitung mbH Göttingen) (siehe auch [12]) ist Kompetenz- und Rechenzentrum der Max-Planck-Gesellschaft und der Universität Göttingen. Die GWDG tritt innerhalb des D-Grid als Grid-Ressourcen-Provider auf und betreibt gemeinsam mit den GoeGrid-Partnern die Grid-Infrastruktur und unterschiedliche, von Seiten des D-Grid geforderte Middleware-Komponenten für verschiedene Grid-Communities. Zudem sind die GoeGrid-Ressourcen in den Räumlichkeiten der GWDG untergebracht. Die Wartung der Hardware- und Netzwerkinfrastruktur sowie die Administration dieser Ressourcen erfolgt in enger Kooperation mit den Partnern aus der Theoretischen Physik und Teilchenphysik der Universität Göttingen.

Eines der Grid-Projekte der GWDG ist die Entwicklung und Bereitstellung von Instant-Grid (siehe auch [13], [14], [15]), einer flexiblen und auch ohne Vorkenntnisse einsetzbaren Demonstrations-, Test- und Entwicklungsumgebung für Grid-Technologien. Dabei werden im Wesentlichen dieselben Technologien und Software-Pakete eingesetzt werden wie im D-Grid. Darüberhinaus gibt es bei der GWDG derzeit Forschungs- und Entwick-

lungsprojekte im Bereich Grid zu Monitoring und Scheduling (siehe auch [16]) sowie im e-Science-Bereich.

#### 4. Die Hardware-Ausstattung des GoeGrid

Die Gesamtausstattung von GoeGrid mit Rechenleistung und Speicherkapazität einschließlich der durch die BMBF-Sondermaßnahmen 2008 hinzukommenden Ressourcen ist in der folgenden Tabelle aufgeführt. Pro CPU-Kern gibt es 2 GB Arbeitsspeicher. 30 Knoten wurden mit erhöhtem lokalen Plattenspeicher ausgestattet, um den Anforderungen einiger Anwendungen der Theoretischen Physik gerecht zu werden.

Cluster-Knoten	CPU-Takt [GHz]	Lokaler Plattenspeicher [GigaByte]	Anzahl Knoten	Anzahl CPU-Kerne	Peak-Leistung [Tera-Flop/s]
2 Intel Xeon Dual Core	3,00	400	13	26	0,63
2 Intel Xeon Quad Core	2,66	1.500	30	240	2,55
2 Intel Xeon Quad Core	2,66	292	78	634	6,64
2 x 2 Intel Xeon Quad Core	2,83	240	48	768	8,69
Summen				1.668	18,51

Die Speicherressourcen für die MediGRID- und TextGrid-Communities sowie der Theoretischen Physik im GoeGrid sind in die Massenspeicher-Infrastruktur der GWDG integriert. Die Speicherressourcen für die Teilchenphysik und den Tier-2-Knoten des LHC-Grids wird über die dCache-Speicherverwaltung bereitgestellt. dCache ist das für die HEP-Grid-Community

und das LHC-Grid meistverwendete Speichersystem (siehe auch [17]). Die aktuell verfügbaren Kapazitäten sind der folgenden Tabelle zu entnehmen:

<b>Speicher-Typ</b>	<b>Kapazität [TeraByte]</b>
SAN	300
dCache	570
Archiv	230

## **5. Der Betrieb von GoeGrid**

Die gemeinsame Nutzung und der Betrieb der integrierten GoeGrid-Ressourcen durch verteilte Gruppen mit unterschiedlichen Anforderungen stellt eine besondere Herausforderung dar, deren Bewältigung im Folgenden kurz beschreiben wird.

Die aus den D-Grid Sondermitteln des BMBF beschafften Ressourcen müssen nach den Vorgaben des BMBF von allen D-Grid-Communities nutzbar sein. Um das zu gewährleisten, müssen die GoeGrid-Ressourcen über alle drei von der D-Grid-Infrastruktur unterstützten Grid-Middleware-Systeme Globus Toolkit (siehe auch [18]), Unicore (siehe auch [19]) und gLite (siehe auch [20]) erreichbar sein. Dies wurde durch die Bereitstellung von Portalen für die verschiedenen Middleware-Systeme auf speziellen Service-Knoten gewährleistet. Damit ist die volle D-Grid-Integration des Zugangs zu GoeGrid sichergestellt. Die eigentlichen GoeGrid-Ressourcen können dann über die jeweiligen Grid-Dienste der verschiedenen Middleware-Systeme angefordert werden.

Die verschiedenen an GoeGrid beteiligten Communities haben zum Teil spezielle Anforderungen an die Betriebssystem-Ausstattungen der Ressourcen. So sind die WLCG-Grid-Community wegen der weltweit einheitlichen, für die Auswertung der Beschleuniger-Experimente verwendeten Programmsysteme auf Rechenressourcen mit der Betriebssystem-Software Scientific Linux (siehe auch [21]) angewiesen. Da diese Umgebung auch für die anderen GoeGrid-Communities akzeptabel ist, werden alle Rechenressourcen in GoeGrid unter CentOS, einer Scientific Linux kompatiblen Variante, betrieben. Die Integration der Administration der GoeGrid-Ressourcen mit den übrigen Rechenzentrums-Ressourcen der GWDG, die im Wesentlichen mit

dem Betriebssystem SuSE Linux ausgestattet sind, ist u. a. daher an dieser Stelle nicht möglich.

Die Betreuung der GoeGrid-Rechenressourcen ist deshalb in Betriebssystempflege und in der Bereitstellung der Middleware arbeitsteilig geregelt: Die GWDG ist für die Hardware-Wartung zuständig, die Betriebssysteminstallation und -pflege wird von der Theoretischen Physik und der Teilchenphysik-Community übernommen. Die von WLCG verwendete Middleware gLite wird von den Hochenergie-Physikern betreut, die Pflege der Middleware-Komponenten Globus und Unicore ist Aufgabe der GWDG.

Auch im Speicherbereich werden auf Grund der speziellen Anforderungen einzelner Communities unterschiedliche Grid-fähige Speicher-Verwaltungssysteme in GoeGrid bereitgestellt. Die Teilchenphysik-Grid-Community verwendet das dCache-System, die übrigen Communities die verteilte Speicherverwaltung mit Datenbankfunktionen des Projektes Open Grid Services Architecture Data Access and Integration (OGSA-DAI) (siehe auch [22]) sowie das System für verteilte Speicherverwaltung Storage Resource Broker (SRB) (siehe auch [23]). Auch hier wird eine Arbeitsteilung gemäß dem vorhandenen Erfahrungswissen vorgenommen: Die GWDG betreut die OGSA-DAI- und SRB-, die Teilchenphysik-Community die dCache-Systeme.

Nachdem alle Voraussetzungen für die Nutzbarkeit der GoeGrid-Ressourcen durch die GridCommunities gegeben sind, ist die Frage der Aufteilung der Ressourcen-Nutzung von entscheidender Bedeutung für den GoeGrid-Betrieb. Grundsätzlich haben die an GoeGrid beteiligten Communities Anrecht auf die Nutzung eines Anteils der GoeGrid-Ressourcen, der ihrem finanziellen und personellen Beitrag bei der Beschaffung und dem Betrieb dieser Ressourcen entspricht. Dieser Anspruch wird durch eine prioritäten-gesteuerte Vergabe von Rechenzeiten erfüllt, wobei die Priorität die in der Vergangenheit genutzte Rechenzeit in einer solchen Weise berücksichtigt, dass im Mittel der angestrebte Nutzungsgrad erreicht wird (fairshare). Dies ermöglicht die Auslastung der Ressourcen auch dann, wenn eine Community zeitweise nicht den ihr zustehenden Anteil an Rechenzeit abnimmt. Das setzt natürlich voraus, dass eine andere Community in dieser Zeit einen höheren Bedarf, als ihr im Mittel zusteht, abnehmen kann. Falls zeitweise die Gesamtheit der an GoeGrid beteiligten Communities die Ressourcen nicht vollständig auslastet, werden diese freien Ressourcen für die Nutzung durch andere D-Grid-Communities freigegeben. Die Einbindung des GoeGrid in das D-Grid ermöglicht so auch in diesem Falle eine vollständige Auslastung der Ressourcen.

## 6. Die Zukunft von GoeGrid

Das Göttinger Grid-Ressourcen-Zentrum GoeGrid ist ein Beispiel dafür, wie unter den gegebenen Finanzierungs- und Förderbedingungen Grid-Ressourcen im D-Grid und in WLCG finanziert, betrieben und genutzt werden können. Entscheidend für den Erfolg von GoeGrid ist die Möglichkeit, die vorhandenen Strukturen und Erfahrungen eines Rechenzentrums mitzunutzen. Für eine dauerhafte Lebensfähigkeit solcher Grid-Ressourcen-Zentren ist allerdings unabdingbar, die Frage einer langfristigen Finanzierung der Betriebskosten und der Ersatzbeschaffung für veraltete Komponenten auf Basis eines kostendeckenden Betriebsmodells zu lösen.

Für die GWDG sind die Erfahrungen mit GoeGrid Anlass, über eine weitergehende Integration der eigenen Rechenzentrums-Ressourcen in eine Grid-Struktur nachzudenken. Voraussetzung hierfür wäre die Etablierung von Nutzer-Communities, in die alle von der GWDG betreuten Nutzergruppen der Universität Göttingen und der Max-Planck-Gesellschaft integriert werden können. Dann könnten die Rechenzentrums-Ressourcen der GWDG mit den GoeGrid-Ressourcen in einheitlicher Form als Grid-Dienste bereitgestellt werden, mit den sich daraus ergebenden Vorteilen der einheitlichen Verwaltung und des Lastausgleichs in einer erweiterten Nutzer-Gemeinschaft. In Blick auf dieses Ziel hat die GWDG mit Partnern aus Industrie und Forschungseinrichtungen im Rahmen des jüngsten D-Grid-Förderprogramms des BMBF Mittel für ein Projekt zur Etablierung einer neuen D-Grid-Community beantragt. Diese neue Community mit dem Name OptiNum-Grid will die Optimierung technischer Systeme und naturwissenschaftlicher Modelle mit Hilfe numerischer Simulationen im Grid durchführen. Unter diese Aufgabenstellung lässt sich der überwiegende Teil aller Forschungsarbeiten auf den Rechenzentrums-Clustern der GWDG einordnen. Die in dem Projekt OptiNum-Grid zu entwickelnden Grid-Dienste für numerische Simulationen können deshalb eine Basis für die Ausweitung der Grid-Nutzung auf alle Nutzergruppen der GWDG bilden und damit die Vorbedingung für die angestrebte Integration von Rechenzentrums- und GoeGrid-Ressourcen schaffen.

### Literatur

[1] D-Grid Initiative, <http://www.d-grid.de>.

[2] Helmholtz-Allianz „Physics at the Terascale“, <http://www.terascale.de/>.

- [3] T. Rings, F. Viezens, J. Meyer, A. Aschenbrenner, Ein interdisziplinäres Grid-Anwenderpraktikum basierend auf Instant-Grid, in diesem GWDG-Bericht.
- [4] MediGrid, <http://www.medigrid.de>.
- [5] F. Viezens, U. Sax, K. Lorberg, Bericht aus MediGRID – Implementierung eines Testbeds mit Instant-Grid für die perspektivische Nutzung einer Pseudonymisierungskomponente, in diesem GWDG-Bericht.
- [6] TextGrid, <http://www.textgrid.de/>
- [7] A. Aschenbrenner, P. Gietz, M. W. Küster, C. Ludwig, and H. Neuroth. Textgrid – a modular platform for collaborative textual editing. In: Proceedings of the International Workshop on Digital Library Goes e-Science (DLSci06), pages 27–36, September 2006.
- [8] CERN, <http://www.cern.ch>.
- [9] Worldwide LHC Computing Grid, <http://lcg.web.cern.ch/LCG/>.
- [10] BMBF-Forschungsschwerpunkt FSP-101, <http://www.fsp101-atlas.de/>.
- [11] Department of Physics, University of Göttingen, <http://physik.uni-goettingen.de/>.
- [12] GWDG, <http://www.gwdg.de/>.
- [13] Instant-Grid, <http://www.instant-grid.org/>.
- [14] C. Boehme, T. Ehlers, J. Engelhardt, A. Félix, O. Haan, T. Kálmán, U. Schwarzmann, D. Sommerfeld, A. Willner, Instant-Grid – A Toolkit for Demonstration, Test and Development of Grid-Infrastructure, Proceedings of the German E-Science Conference 2007 (GES2007), Paper 47.
- [15] U. Schwarzmann, Instant-Grid – eine Live-CD für die Demonstration, die Entwicklung und den Test von Grid-Anwendungen, in diesem GWDG-Bericht.
- [16] D. Sommerfeld, H. Richter, A two-tier approach to efficient workflow scheduling in MediGRID, in diesem GWDG-Bericht.
- [17] dCache, <http://www.dcache.org>.
- [18] The Globus Toolkit, <http://www.globus.org>.
- [19] UNICORE, <http://www.unicore.org>.
- [20] gLite, <http://glite.web.cern.ch/glite/>.

[21] Scientific Linux, <http://www.scienticlinux.org/>, <http://linux.web.cern.ch/linux/scientic4/>.

[22] OGSA-DAI, <http://www.ogsadai.org.uk>.

[23] Storage Resource Broker (SRB), [http://www.sdsc.edu/srb/index.php/Main\\_Page](http://www.sdsc.edu/srb/index.php/Main_Page).

---

---

# **Ein interdisziplinäres Grid-Anwenderpraktikum basierend auf Instant-Grid**

**Thomas Rings**

*Institut für Informatik, Georg-August-Universität Göttingen*

**Fred Viezens**

*Institut für Biometrie und Medizinische Informatik, Otto-von-Guericke-Universität Magdeburg*

**Jörg Meyer**

*II. Physikalisches Institut, Georg-August-Universität Göttingen*

**Andreas Aschenbrenner**

*Staats- und Universitätsbibliothek, Georg-August-Universität Göttingen*

## **1. Einleitung**

Die Idee eines interdisziplinären Anwenderpraktikums für Grid-Computing [2] entstand durch eine Überschneidung von Interessen in den Disziplinen Physik, Medizin und Geisteswissenschaften. Hierbei spielte die Informatik

die Verbindungsrolle, da die drei Disziplinen ein wichtiges Forschungsgebiet der Informatik, das Grid-Computing, benutzen, um deren jeweilige Forschung voranzutreiben.

Das Praktikum fand im Sommersemester 2008 an der Universität Göttingen statt. 16 teilnehmende Studierende aus den o. g. Disziplinen wurden in vier Gruppen unterteilt. Jede Disziplin stellte Tutoren zur Betreuung der Studierenden. Zudem wurden jeder Gruppe vier Arbeitsrechner aus dem Rechnernetz zum Aufbau eines lokalen Grids zu Verfügung gestellt. Mit diesem Grid sollten die Aufgaben der verschiedenen D-Grid Projekte praktisch gelöst werden.

Die Funktionsweise von Grid-Technologien wurde den Studierenden im Praktikum durch deren Anwendung erklärt. Mit Hilfe der lokal gestarteten Grid-Software Instant-Grid [8, 14] konnte eine eigenständige Grid-Experimentierumgebung zur Bearbeitung der Aufgaben aus den Disziplinen aufgebaut und benutzt werden.

Das nachfolgende Kapitel ist folgendermaßen gegliedert: In Abschnitt 2 wird die technische Grundlage, die Konfiguration des Instant-Grid, für das Praktikum beschrieben. In Abschnitt 3 werden die Aufgaben der verschiedenen Disziplinen vorgestellt. Abschließend in Abschnitt 4 werden kurz die Ergebnisse diskutiert und ein Ausblick gegeben.

## **2. Technischer Aufbau**

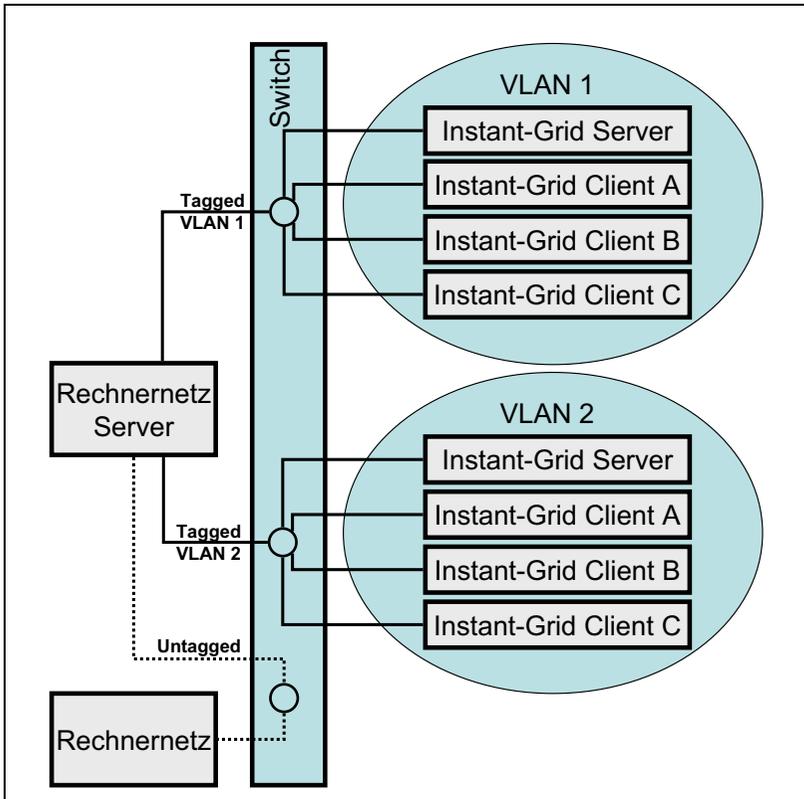
Die praktische Realisierung der Aufgaben des Praktikums benötigt eine Grid-Software. Als Grid-Software wurde Instant-Grid [8,14] ausgewählt, da es vereinfachte Möglichkeiten zum Starten von lokalen Computer-Grids [2] bietet. Instant-Grid basiert auf Linux-Knoppix [9] und beinhaltet die Grid-Middleware *Globus Toolkit 4 (GT4)* [3]. Somit kann mit Instant-Grid ohne aufwändige Installation ein Computer-Grid von CD gestartet werden. Trotzdem benötigt Instant-Grid, wie im folgenden Abschnitt beschrieben, Anpassungen um in einem lokalen Rechnernetz zu starten.

### **2.1 Konfiguration in einem lokalen Rechnernetz**

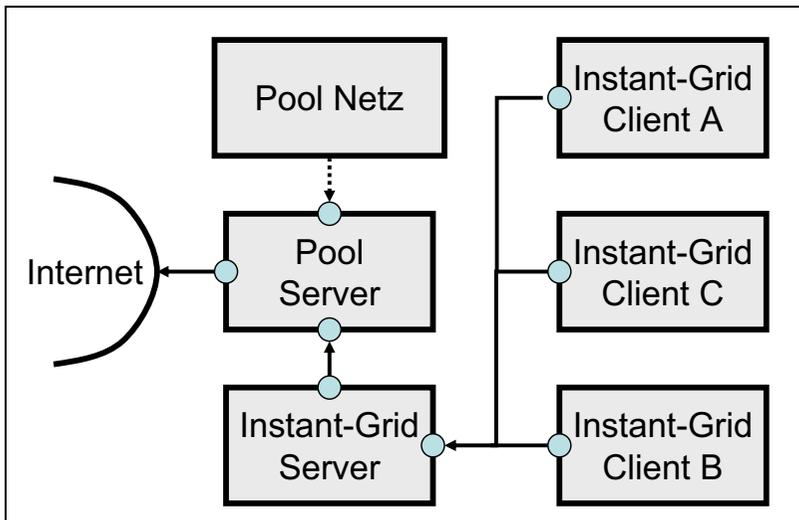
Instant-Grid benötigt ein abgeschlossenes lokales Netzwerk, um Konflikte mit Diensten in existierenden Rechnernetzwerken zu vermeiden. Deswegen müssen Rechner, welche das Instant-Grid starten, von ihrem Rechnernetz getrennt werden. Dazu werden die Rechner eines Instant-Grid auf Netzwerkeschicht-Ebene in einem *Virtual Local Area Network (VLAN)* zusammengefasst und somit vom Rechnernetz getrennt. Wie in Abbildung 1 beispielhaft dargestellt, gibt es zwei Projektgruppen, VLAN 1 und VLAN 2, welche

jeweils unabhängig voneinander ein Instant-Grid mit eigenem Server und drei Clients starten können. Um ein Routing zwischen den Netzen und ggf. Firewallregeln auf dem Rechnernetzserver zu ermöglichen, verlässt das eigentliche Rechnernetz den Switch direkt (untagged) und die Instant-Grid-Netze mit ihrer jeweiligen VLAN-Kennung (tagged).

Die Benutzung des Instant-Grid erfolgt nur zu vorgegebenen Zeiten, da in dem Switch die jeweiligen VLANs durch einen Tutor manuell eingerichtet oder entfernt werden müssen. Dies ist nötig, damit die Rechner zusätzlich ihr Standardbetriebssystem starten können. In unserer Konfiguration wurden jeder Gruppe vier Rechner zugeteilt. Auf diesen konnten die Studierenden ihr eigenes Grid aufbauen, starten und konfigurieren.



**Abb 1: VLAN-Konfiguration**



**Abb. 2: NAT-Konfiguration**

Damit von Instant-Grid-Rechnern auf externes Netz (Zugang zum Internet) zugegriffen werden kann, sollten auf dem Rechnernetzserver sowie auf dem Instant-Grid-Server *Network Address Translation* (NAT)-Gateways eingerichtet werden, wie in Abbildung 2 dargestellt ist.

## 2.2 Technische Herausforderungen des Instant-Grids

Durch die Benutzung der Linux-Knoppix-Live-CD fehlt es Instant-Grid an einer Komponente zum Management von Benutzer-Accounts. Konfigurationsdaten können nicht permanent geschrieben werden. Somit können Benutzer nicht permanent erstellt werden. Deswegen können die Studierenden nur vor Ort arbeiten und werden zusätzlich von einem Tutor betreut, da die Studierenden als Administrator agieren können und das Rechnernetz schädigen können.

Die Instant-Grid-Netzwerkumgebung erfordert ein lokales Netzwerk. Deshalb ist, wie in Abschnitt 2.1 beschrieben, die Konfiguration des Instant-Grid in einem Rechnernetz sehr umständlich. Studierende sollten die Konfiguration des VLANs nicht selbst vornehmen können. Bei unserer Konfiguration wurde ein Script von einem Tutor gestartet, welches das VLAN aktiviert bzw. deaktiviert. Wird Instant-Grid ohne die VLAN-Aktivierung gestartet, kann es zu erheblichen Konflikten mit dem Rechnernetz kommen.

### **3. Aufgaben aus den Disziplinen**

Das Ziel des Praktikums war es, den Studierenden unabhängig von deren Studienfach Konzepte des Grid-Computings durch Anwendung von Grid-Technologie zu vermitteln. Der Praxisbezug konnte durch die drei in Göttingen ansässigen Projekte des D-Grids [7] der Disziplinen Physik, Medizin und Geisteswissenschaften hergestellt werden. Somit befassen sich die Aufgaben mit HEP Grid [5] aus der Physik, MediGRID [11] aus der Medizin und TextGrid [13] aus den Geisteswissenschaften.

Jede Disziplin erstellte einen Aufgabenblock, welcher sich auf das jeweilige D-Grid Projekt bezog und von jeder Gruppe rotierend bearbeitet wurde. Die drei Aufgabenblöcke wurden folgendermaßen unterteilt:

1. Einführung / Grundlagen
2. Übung / Praxis
3. Reflexion / weiterführende Fragen

Teil 1 gab eine Einführung in die jeweilige Disziplin und behandelte die Voraussetzungen und den Stand des D-Grid Projektes. Die Studierenden recherchierten die Besonderheiten des jeweiligen Themas und arbeiteten Beziehungen der Disziplin zur Grid-Technologie heraus. Zudem sollten Optimierungsvorschläge zu den Grid-Projekten bezüglich Grid-Technologie gegeben werden.

Im zweiten Teil sollten diese Vorschläge und weitere Aufgabenstellungen in einem Grid praktisch umgesetzt werden. Hierzu zählte unter Anderem der Versuchsaufbau des Instant-Grid und die technische Umsetzung der Fragestellungen.

Die Vor- und Nachteile dieser Lösung sowie weitere Optimierungsvorschläge wurden im dritten Teil diskutiert und wiederum umgesetzt. Hierbei wurde die entwickelte Grid-Software vorgestellt und ausgeführt sowie Herausforderungen bei der Entwicklung erörtert und diskutiert. Im Folgenden werden die Aufgaben bezüglich der D-Grid Projekte näher erläutert.

#### **3.1 MediGRID [11]**

Innerhalb des MediGRID Aufgabenblocks erfolgte eine Einweisung in verschiedene biomedizinische Anwendungen. Hierzu konnten die Studierenden Anwendungen zur Bildverarbeitung, Bioinformatik und klinischen Forschung über das MediGRID-Portal [12] als Gast benutzen. Um sensible Bereiche dieser Anwendungen zu benutzen, mussten die Studierenden die Beantragungsprozedur von D-Grid-Zertifikaten durchlaufen. Diese Zertifi-

kate ermöglichten durch Eintrag in der Virtual Organisation Education-VO, sensible Grid-Ressourcen und Grid-Anwendungen des MediGRID zu benutzen.

Zudem wurden Programme gridifiziert und in das Instant-Grid eingebunden. Die Verteilung im Grid erfolgte durch Kommandozeilenaufrufe. Diese Aufrufe wurden in ein Script übertragen und ausgeführt. Der Ablauf der Kommandos in diesen Script entsprach einem Grid-Arbeitsablauf (Workflow). Dieses Script wurde exemplarisch in der Applikation für Genvorhersagen angewendet. Die Ausführungen der von den Studierenden entwickelten Workflows im Instant-Grid spiegelten die realen Abläufe auf den Ressourcen des D-Grid wieder. Die Ergebnisse durch Anpassungen und Lösungen aus den thematischen Praktikumsaufgaben fanden Berücksichtigung in den laufenden MediGRID-Entwicklungen.

### 3.2 HEP Grid [5,6]

Der Praktikumsteil der *Hochenergie-Physik* (HEP) hatte als Ziel, einen Einblick in die typischen Anforderungen moderner Teilchenphysik-Experimente an das Computing zu geben und die verschiedenen Herausforderungen zu untersuchen. Als Beispiel wurde das DØ Experiment am Proton-Antiproton-Beschleuniger Tevatron am Fermilab [1] in den USA betrachtet. Am Tevatron finden Kollisionen von Protonen- und Antiprotonen-Paketen bei den derzeit höchstmöglichen Energien mit einer Rate von etwa 2,5 MHz statt. Die Signale der bei einer Kollision gestreuten und neu produzierten Teilchen wurden nach einer Ereignis-Vorselektion mit einer Rate von 50 Hz vom DØ Detektor aufgezeichnet. Die Rohdaten-Menge pro Ereignis beträgt etwa 250 KB. Das Experiment läuft seit 2001, so dass bereits eine enorme Menge an Daten gemessen und gespeichert wurde. Aus den Rohdaten wurden Physik-Objekte rekonstruiert und ebenfalls gespeichert, um diese in verschiedenen Analysen weiter zu verarbeiten. Neben den gemessenen Daten werden für die Physik-Analysen Simulationen von Kollisionen, so genannte Monte-Carlo-Ereignisse, benötigt. Dazu wurde etwa die gleiche Menge an simulierten und gemessenen Daten gebraucht.

Im Praktikum wurde die Simulation von DØ Ereignissen schrittweise durchgeführt. Der Workflow der Simulationsschritte sowie das Workflow-Skriptsystem wurden im Einzelnen betrachtet. Aus der Dauer der Simulation einzelner Ereignisse und aus den Dateigrößen der verschiedenen Ausgabedaten wurde extrapoliert, wie viel CPU- und Speicherbedarf das Experiment in etwa hat. Im Jahr 2005 hatte das DØ Experiment beispielsweise 250 TB aus gemessenen Rohdaten neu rekonstruiert. Diese gewaltige Anforderung an Computer-Ressourcen lässt sich nur mit weltweit verteilten Grid-Clustern

bewältigen. In einem weiteren Praktikumsteil wurde durch die Studierenden die interaktive Generation von Monte-Carlo-Ereignissen auf ein solches Grid-System übertragen. Hierzu wurde wie in anderen Versuchsteilen Instant-Grid verwendet. Die Studierenden schickten über Kommandozeilen-Befehle des Globus-Toolkits Simulationsergebnisse ins Instant-Grid. Die Verwendung von Kommandozeilen und Skripten ist typisch für die Anwendung von Grid-Computing in der Hochenergie-Physik.

Im Praktikum wurden Aspekte wie Daten-Archivierung, Datensicherheit, Aufbau eines Grid-Systems, Anforderungen an die verwendete Middleware sowie mögliche künftige Weiterentwicklungen diskutiert sowie praktisch betrachtet und umgesetzt. Die Computing-Anforderungen der weiteren Experimente am *Large Hadron Collider* (LHC) [10] am CERN in Genf erhöhen den Bedarf an Grid-Computing in der Hochenergie-Physik. Da die Hochenergie-Physik Grid-Computing zur häufigen und ständigen Erstellung von Simulationen und Analysen verwendet, ist eine stabile, standardisierte und getestete Umgebung wichtiger als die Verwendung neuester Entwicklungen in der Grid-Technologie.

### **3.3 TextGrid [13]**

Der Aufgabenblock zu TextGrid vermittelte den Studierenden die Chancen einer Grid-Infrastruktur für die Geisteswissenschaften. Die Diskussionen mit den Studierenden bauten dabei auf den konkreten Arbeiten des Projektes TextGrid auf, gingen aber weit darüber hinaus.

Der erste Teil vermittelte ein grundlegendes Verständnis für digitale Kulturgüter (z. B. Digitalisate) und darauf basierende geisteswissenschaftliche Forschung. Anforderungen in Bezug auf den wissenschaftlichen Workflow (speziell wie er in TextGrid unterstützt wird), sowie auch in der Verwaltung von digitalen Objekten (z. B. Metadatenverwaltung, Langzeitarchivierung) wurden daraus abgeleitet.

Mechanismen zur Datenreplikation und verteilten Speicherung von Digitalisaten in einer Grid-Umgebung sind wichtige Grundlagentechnologien für die Geisteswissenschaften. Der zweite Teil demonstrierte den Studierenden daher verteilte Datenvorhaltung sowie die Ausführung von Grid-Jobs wie z. B. zur verteilten Konvertierung von Digitalisaten von 100 Megabyte TIFF-Bildern in 1,4 Megabyte große JPEG-Bilder.

In der abschließenden Diskussion im dritten Teil arbeiteten die Studierenden die Chancen einer verteilten digitalen Infrastruktur für die Geisteswissenschaften heraus. Umgekehrt zeigten sie auch einige mögliche Beiträge geisteswissenschaftlicher Technologien in eine disziplinübergreifende Grid-

Infrastruktur auf – von interaktiven Tools bis zur kollaborativen Forschungs-  
umgebung.

Die Studierenden lernten in diesem Aufgabenblock die durchaus kniffligen Anforderungen der Geisteswissenschaften und dahingehend schon entwickelte Technologien und Konzepte kennen. Interaktion und Vergleich mit den Physikern, Medizinerinnen und Informatikern in GoeGrid [4] haben den Studierenden in besonderem Maße die unterschiedlichen Anwendungsgebiete näher gebracht und die zukünftigen e-Humanities greifbar werden lassen.

#### **4. Ergebnisse und Ausblick**

Das Praktikum wurde von den Studierenden gut evaluiert und war somit ein Erfolg. Aus technischer Sicht hat sich das Instant-Grid als ein ideales Werkzeug für die Durchführung des Praktikums erwiesen. Instant-Grid ist ein Instrument für ad-hoc-Grid-Anwendungen, welches erweiterbar und gut skalierbar ist. Es sind auch Anpassungen im Sinne kurzfristiger Entwicklungen wie Portlets bis hin zum Aufbau vollständiger Entwicklungsumgebungen möglich. Dem Charakter des Praktikums Rechnung tragend, sind Lösungswege eruiert worden, um Problemstellungen der Disziplinen zu lösen.

Auch zur Vermittlung von tiefer gehenden Grid-Programmierkenntnissen, speziell für die Studierenden der Informatik, ist Instant-Grid das Werkzeug der Wahl. Der Bedarf nach Grid-Entwicklern aus den Reihen der Informatik sowie der anderen Fachdisziplinen ist klar erkennbar. Die Vermittlung von Grid-Programmiertechniken ist in einem interdisziplinären Anwenderpraktikum nicht möglich. Deshalb ist ein Programmierpraktikum für Grid-Technologien in Vorbereitung.

In der abschließenden Diskussion und Nachbereitung dieses interdisziplinären Praktikums wurde sich mit den Interaktionen und der Zusammenarbeit zwischen den Studierenden der verschiedenen Fachrichtungen näher auseinander gesetzt. Es ist nicht Ziel dieses interdisziplinären Praktikums, dass Geisteswissenschaftler programmieren oder Informatiker die chemischen Prozesse der Genvorhersage lernen, sondern dass sie gemeinsam und sich gegenseitig ergänzend die Aufgaben lösen. Idealerweise würde z. B. der Physiker die richtigen Fragen stellen, der Geisteswissenschaftler die Methodik zur Beantwortung der Frage einbringen, der Mediziner das Hintergrundwissen und der Informatiker zur Umsetzung beitragen. In jedem Aufgabenblock wurde die jeweils zur Disziplin passende Rolle durch die Studierenden wahrgenommen und dieser somit einander ergänzend gelöst.

In diesem Sinne gehen die Fertigkeiten, die dieses interdisziplinäre Praktikum vermittelt, weit über das „Grid“ hinaus. Es geht um Kooperation mit

Kollegen, die ganz anderes Wissen einbringen und vielleicht ganz anders an Fragestellungen herangehen. Ebenso geht es um Kommunikation, in der Anforderungen und Vorschläge nicht alleine mit Fachbegriffen und Konzepten gespickt sind, die nur Kollegen der eigenen Disziplin verstehen können. Vor allem aber geht es darum, die unterschiedlichen Disziplinen auch in der Universität stärker zusammen zubringen und die Studierenden auf eine spätere Berufswelt vorzubereiten, in der die Disziplinen meist verschmelzen. Gerade die Vermittlung solcher Eigenschaften ist daher für die Studierenden eine Erfahrung für das Leben, wie sie nur eine interdisziplinäre Lehrveranstaltung vermitteln kann.

## **Literatur**

- [1] Fermilab. [Online; <http://www.fnal.gov> überprüft am 14.10.2008].
- [2] Ian Foster. What is the Grid? A Three Point Checklist. Grid Today, 1(6):22, 2002.
- [3] Ian Foster. A globus primer, An Early and Incomplete Draft. Technical report, Globus Alliance, 2005.
- [4] Göttinger Grid-Ressourcen-Zentrums (GoeGrid). [Online; <http://www.d-grid.de/index.php?id=438> überprüft am 10.10.2008].
- [5] HEP-Grid. [Online; <http://www.d-grid.de/index.php?id=44> überprüft am 29.09.2008].
- [6] Worldwide LHC Computing Grid. [Online; <http://lcg.web.cern.ch/LCG> überprüft am 29.09.2008].
- [7] D-Grid Initiative. [Online; <http://www.d-grid.de/> überprüft am 29.09.2008].
- [8] Instant-Grid. [Online; <http://instant-grid.de> überprüft am 29.09.2008].
- [9] KNOPPER.NET. Knoppix. [Online; <http://www.knoppix.org> überprüft am 29.09.2008].
- [10] The Large Hadron Collider (LHC). [Online; <http://lhc.web.cern.ch/lhc/> überprüft am 14.10.2008].
- [11] MediGRID. [Online; <http://www.d-grid.de/index.php?id=42> überprüft am 29.09.2008].
- [12] MediGRID-Portal. [Online; <https://portal.medigrid.de> überprüft am 09.10.2008].

[13] TextGrid. [Online; <http://www.textgrid.de/>, <http://www.d-grid.de/index.php?id=167> überprüft am 29.09.2008].

[14] Alexander Willner. Entwurf und Implementierung einer Ressourcen-Datenbank für das Instant-Grid-Projekt der GWDG. Master's thesis, Georg-August-Universität Göttingen, 2006.

---

---

# **Bericht aus MediGRID – Implementierung eines Testbeds mit Instant-Grid für die perspektivische Nutzung einer Pseudonymisierungskomponente**

**Fred Viezens**

*Institut für Biometrie und Medizinische Informatik, Otto-von-Guericke-Universität Magdeburg*

**Ulrich Sax, Kai Lorberg**

*Abteilung Medizinische Informatik, Universitätsmedizin Göttingen, Georg-August-Universität*

Motivation für die Einbindung einer Komponente, die eine Pseudonymisierung personenbezogener Daten vor der Verarbeitung in einem Grid vornehmen kann, war die Nutzung verteilter IT-Systeme für die Medizin zur Verarbeitung genetischer und phänotypischer Daten in entsprechend großen Mengen und in rechenintensiven Algorithmen und Methoden. Der Mediziner soll in die Lage versetzt werden, in einer sicheren verteilten Umgebung umfangreiche genetische und klinische Informationen zu verarbeiten, um neue Erkenntnisse zu erzielen .

## 1. Methodik

In der virtualisierten Teststellung wurde ein Instant-Grid [1] CD-Image benutzt, das innerhalb einer VMware [2] auf drei virtuellen Maschinen gebootet wurde. In dieser Testumgebung wurde zuerst die Erreichbarkeit eines Pseudonymisierungsdienstes von außen getestet. In einer zweiten Phase wurde ein solcher Dienst versuchsweise geschrieben, im Grid-Testbed deployed (verteilt) und getestet.

## 2. Instant-Grid

Instant-Grid basiert auf dem bekannten Konzept der Live-CDs, hier eine Knoppix-Distribution [3]. Mit der Knoppix-Live-CD wird der Frontend gebootet. Dieser agiert im LAN als Server. Weitere Rechner des gleichen Subnetzes können per Preboot eXecution Environment (PXE) booten, das Betriebssystem vom Server laden und sich im Grid als Ressource anmelden. Dies läuft automatisch ab. In der genutzten Version 0.5 (Stand November 2006) war es nötig, in der Phase des Bootens die Netzwerkschnittstelle selbst zu wählen. Ist der Systemstart abgeschlossen, sind sofort alle integrierten Web-Schnittstellen für Grid-Infrastruktur und Komponenten des Globus Toolkit [4] verfügbar. Als Benutzerschnittstelle zur Grid-Middleware steht ein browserbasiertes Portal bereit (Projekt Gridsphere [5]). Das User-Interface ist ein Web-Portal. Integriert sind auch Beispiel-Applikationen. Sie demonstrieren die Vorteile des Grid-Computing, unter anderem durch Komponenten zur Ressourcenüberwachung und Veranschaulichung der Lastverteilungen. Instant-Grid ist nicht nur Präsentationswerkzeug, sondern eine kostengünstige Testumgebung für Entwicklungen im Bereich der Grid-Applikationen. In besonders datenschutzsensiblen Forschungsbereichen muss dieser Bereich isoliert und vor Zugriffen von außen geschützt sein. Wie bei einer Live-CD üblich, verfügt auch Instant-Grid über ein feststehendes und bekanntes Login für die Anmeldung. Aus Sicherheitsgründen sind externe Anmeldeversuche mit diesen Zugangsdaten generell geblockt. Darüber hinaus laufen alle Web-Services mit einem eigenen gesperrten Systemaccount. Das Instant-Grid ist vor direkten Zugriffen von außen durch eine Firewall geschützt, die Clients können im Bedarfsfall jedoch per NAT (Network Address Translation) über den Frontend hinaus kommunizieren. Auf Basis von SimpleCA (Certificate Authority) werden neuen Entities, wie etwa Nutzern, Diensten oder Ressourcen, X.509-Zertifikate [6] zugeordnet.

### **3. Pseudonymisierung**

Der Gesetzgeber führt zur Pseudonymisierung in § 3 Abs. 6a Bundesdatenschutzgesetz (BDSG) folgendes dazu aus:

*„Pseudonymisieren ist das Ersetzen des Namens und anderer Identifikationsmerkmale durch ein Kennzeichen zu dem Zweck, die Bestimmung des Betroffenen auszuschließen oder wesentlich zu erschweren.“*

Was nichts anderes bedeutet, als dass eine direkte Identifikation des Betroffenen bzw. Patienten nicht möglich sein darf bzw. massiv erschwert wird. Die Pseudonymisierung ist jedoch nur bedingt ein irreversibler Vorgang. Bei zwingender Notwendigkeit und unter Einhaltung vorher festgelegter rechtlicher Voraussetzungen kann der Personenbezug von einer hierzu berechtigten Institution wieder hergestellt werden. Das ist dann der Fall, wenn der Betroffene direkt von neusten Forschungsergebnissen partizipieren kann. Indirekt und juristisch abgesichert ist es also möglich, eine Verbindung zurück herzustellen.

### **4. Pseudonymisierungsdienst**

Die Pseudonymisierung medizinischer Daten geschieht in erster Linie in der Verschlüsselung des Patienten-Identifikators (PID). Das Pseudonym (PSN) wird von einer juristisch legitimierten Einrichtung während der Pseudonymisierung erzeugt und in Verbindung mit den medizinischen Daten an eine Dateninstanz weitergereicht. Das PSN agiert als Zuordnungskriterium für Speicherung, Zusammenführung und Verarbeitung. Es werden zu keiner Zeit identifizierende Daten (IDAT) oder die PID zusammen mit dem PSN hinterlegt. Die Re-Identifikation von Daten aus der Forschung auf die entsprechende Person ist nur über die De-Pseudonymisierung des PSN in den PID durch die berechnete Institution möglich. Die Telematikplattform medizinischer Forschungsnetze e.V. (TMF) [7] stellt den validierten Pseudonymisierungsdienst (PSD) zur Verfügung.

### **5. Einbindung des PSD ins Grid**

Im Instant-Grid wurde der Versuch einer Implementierung eines PSD als Servicenutzung einer schon existierenden Lösung und einer proprietären Grid-Anwendung unternommen, um eine allgemeine Aussage für den Einsatz in verteilten Systemen wie dem Grid zu machen. So wurden zunächst einmal die nötige Testumgebung geschaffen und dann im weiteren Verlauf eine Beispiel-Implementierung eines PSD generiert und eine Legacy-Anwendung integriert.

## 6. Testbed und Konfiguration

Der Testrechner war ausgestattet mit einem Linux als Betriebssystem. Zur Virtualisierung wurde VMware Server verwendet, dabei war ein Server mit zwei Clients für eine experimentelle Teststellung ausreichend. Dem Frontendserver wurde ein CD-Laufwerk konfiguriert, um das Instant-Grid-Image zu booten. Zuerst wurde die Instant-Grid Version 0.4.0.2, später Version 0.5 verwendet. Die zweite Anpassung bezog sich auf die Ethernet-Netzwerkkarte, damit im Host-Only-Modus Dienste wie PXE und TFTP (Trivial File Transfer Protocol) unterstützt wurden, um das Booten der Clients zu ermöglichen. Gleiches galt für die Netzwerkkarten der Clients. Da der Frontend-Rechner den DHCP-Server (Dynamic Host Configuration Protocol) des Netzwerks stellt, war zu vermeiden, dass zeitgleich auch die NAT-Option aktiviert war. Der DHCP-Server der VMware würde sonst den des Frontend behindern, denn er antwortet schneller und somit wäre keine Kommunikation untereinander möglich. Für Nachinstallationen etc. war eine Verbindung nach außen nötig. Daher wurde dem Frontend eine weitere Netz Karte konfiguriert, die NAT-fähig war. Diese blieb beim Start deaktiviert, um jene Interferenzen zu vermeiden. Die Startsequenz hatte folgende Teilschritte:

1. Frontend booten und DHCP-Server *eth0* als Schnittstelle zuweisen
2. Clients booten, die Informationen über TFTP vom Frontend beziehen
3. zweite Netzwerkkarte (*eth1*) des Frontend über Konfigurationsmenü aktivieren
4. Eingabe *pump -i eth1* (Beziehen der IP für Internet-Access) in der Servershell

## 7. Probleme bei der Teststellung und deren Lösungen

Zunächst konnten die Clients nach dem Start des Frontends nicht über das Netzwerk booten. Die Lösung war das Setzen des PXE-Modus in den Netzwerkkarten. Die Clients benutzen diesen Modus automatisch, wenn weder Festplatte noch andere Laufwerke zugeordnet sind. Das nächste Hindernis war die automatische IP-Vergabe des DHCP-Servers während des Bootvorganges der Clients. Mit dem Editor vi [8] wurde die Host-Konfigurationsdatei unter */etc/hosts* angepasst und per *ifconfig* die richtige IP eingetragen, damit auf die entsprechenden Maschinen gezeigt wurde. Um eine GridSphere-Anwendung zu implementieren, wurde auf Ant [9] zurückgegriffen. Mit Ant lassen sich bei der Kompilierung schnell Grundstrukturen für neue Portlet-Projekte erzeugen. Alle nötigen Verzeichnisse können zeitnah angelegt und kompilierte, ausführbare Dateien automatisch darin abge-

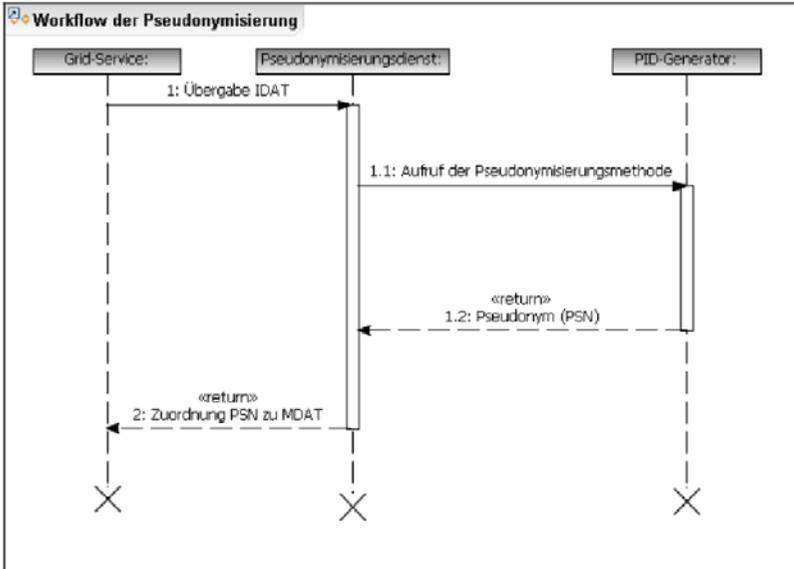
speichert werden. In der Version 0.4 von Instant-Grid war Ant noch nicht einbezogen, daher erfolgte ein Update auf das Release der Version 0.5. Beim Ausrollen der Projektstruktur fehlte das File *build.xml*. Die erste Lösungsmöglichkeit bestand darin, die build-Datei selbst zu schreiben. Man nutzt Ant, um die reine Kompilierung in eine ausführbare Datei vorzunehmen. Die ausführbare Datei würde dann als portal-unabhängige Applikation zur Verfügung stehen. Die andere Option sah das Ausrollen der Anwendung als Portlet vor. Dafür fehlte jedoch die entsprechende build-Datei. Instant-Grid verwendet GridSphere als Portal-Framework. Das ist nötig, um eine Laufzeitumgebung für die Portlets des Grids zu bilden. Im Release 0.5 liegt GridSphere kompiliert und konfiguriert in der Version 2.2.5 vor. Da die GridSphere-Quelldateien nicht beiliegen, konnte die benötigte Datei nicht selbst generiert werden. Aber diese Dateien sind im Internet verfügbar. Nach dem Download konnte durch Kompilierung der Quelldateien die build-Datei erzeugt werden. Das Portlet konnte damit im Tomcat-Verzeichnis [10] ausgerollt werden. In der verwendeten Instant-Grid-Version lief Tomcat in der Version 5.0.28 mit der Option *-security*. Das Verzeichnis war in der Pfadvariablen *CATALINA\_BASE* gesetzt und bildete den Pfad */usr/local/jakarta-tomcat-5.0.28*.

## 8. Integration

Die Integration der Beispielimplementierung erfolgt in zwei Formen. Einmal als eigenständiges Programm und in der anderen Version als Java-Portlet, welches in das GridSphere-Portal integriert wurde.

## 9. Implementierung

Für die Implementierung wurden Instant-Grid 0.5 und das dort eingebundene GridSphere 2.5.5 verwendet, wichtig für die Reproduzierbarkeit der Resultate. Weiterhin sind aus dem Internet für das Deployment wichtige Dateien nachgeladen worden. Bei diesen Dateien handelte es sich um Konfigurationsdateien in XML (Extensible Markup Language), die für das Kompilieren und Veröffentlichen in GridSphere notwendig waren. Anhand des UML (Unified Modeling Language)-Sequenzdiagramms (siehe Abb.1) erfolgte die Programmierung von zwei Java-Klassen. Beiden Java-Klassen ist die PID-Generator-Methode [11] in ihrer Grundfunktion gemein. In der ersten Klasse wird über die Kommandozeile eine Zeichenkette als IDAT eines Patienten pseudonymisiert. Die zweite Klasse wurde nach JSR 168 [12] erstellt und so mit dem Portal-Framework von GridSphere kompatibel. Dafür wurde eine zusätzliche JSP-Datei (Java Server Pages) einbezogen, die die Userinteraktion abbildet.



**Abb. 1: Workflow-Diagramm zur Pseudonymisierung**

Die doView-Methode wird aufgerufen, wenn in GridSphere das Portlet geöffnet wird. Erst wird die URL generiert und gespeichert, die in der JSP-Datei notwendig ist. Dann wird das Attribut *name* vom *request* geladen. Zum Schluss wird die definierte JSP-Datei aufgerufen, um das Ergebnis auszugeben. Mit HTML (Hypertext Markup Language) erfolgt die strukturierte Ausgabe. Außer den JSR-168-typischen Methoden beinhaltet diese Klasse die generatePID-Methode. Die Ausgabe des Ergebnisses erfolgt dann entweder auf der Konsole oder durch JSP-erzeugte HTML-Fragmente, die vom Portlet-Container gerendert werden.

## 10. Deployment

In diesem Abschnitt geht es um die Strategie und Methodik des Ausrollens und Kompilierens des Pseudonymisierungsprojektes. Dabei ist wieder die Unterscheidung bei den oben angemerkten Möglichkeiten zu beachten. Für das erste Szenario wurde ein Verzeichnis angelegt, in dem das Projekt gespeichert wurde. Dazu wurde weiter ein Unterverzeichnis angelegt, wo alle Quelldateien gesichert wurden. Diese Grundstruktur genügte der entwickelten build-Datei, die darauf arbeitete. Folgender Programmiercode veranschaulicht dies:

```

<?xml version="1.0" encoding="UTF-8"?>
<project name="PSD" default="compile" basedir=". ">
  <description>
    Build file for the PSD application.
  </description>
  <property name="classes.dir" location="classes" />
  <property name="dist.dir" location="dist" />
  <property name="dist.jarfile" value="PSD.jar" />
  <target name="compile" depends="-init"
    description="Compile all sources.">
    <mkdir dir="${classes.dir}" />
    <javac srcdir="src" destdir="${classes.dir}" />
  </target>
  <target name="clean" description="Clean output dir.">
    <delete dir="${classes.dir}" />
  </target>
  <target name="rebuild" depends="clean,compile"
    description="Cleanly compiles all sources." />
  <target name="-init">
    <!-- Create the time stamp. -->
    <tstamp>
      <format property="TODAY_UK"
        pattern="dd MMM yyyy HH.mm" locale="en_GB" />
    </tstamp>
  </target>
  <target name="dist" depends="rebuild"
    description="Creates the binary distribution.">
    <mkdir dir="${dist.dir}/${TODAY_UK}" />
    <jar basedir="${classes.dir}"
      destfile="${dist.dir}/${TODAY_UK}/${dist.jarfile}" />
  </target>
</project>

```

### Abb. 2: Programmiercode 1 build.xml

Der Code initiiert, das Ant sich die Quelldaten aus dem angelegten Ordner holt. Vor dem Kompilieren wird ein Verzeichnis erstellt, das als Ablage von kompilierten Klassen dient. Danach generiert Ant ein JAR-File zur Archivierung, mit eigenem Verzeichnis inklusive Timestamp. Dies ist eine Art der Versionierung, da bei jedem Kompilierungsvorgang der Klassenordner geleert wird. Diese Prozedur erfolgte über den commandline-Befehl *ant dist*. Die zweite Variante funktionierte mit ähnlichem Vorgehen. Mit Ant und dem nachträglich installierten build-File wurde mit *ant new-project* die Ordner-

struktur erzeugt. Danach wurden die Quelldateien (Pseudonymisierungsdienst.java und psd.jsp) in die Verzeichnisse gespielt. Mit der erzeugten build-Datei im Hauptverzeichnis erfolgte dann die Kompilierung. Zum Schluss sei darauf hingewiesen, dass alle Variablen der build-Datei auf entsprechenden Pfadvariablen zu den Verzeichnissen von Ant, Tomcat und GridSphere beruhen.

## **11. Ergebnisse**

Die Teststellung hat bewiesen, dass alle Szenarien funktionierten. Das unflexible Konstrukt eines Systems auf Basis von Live-CDs ist bei der gewählten Experimentierumgebung der Nachteil dieses Integrationsvorhaben. Damit die Beispielapplikationen auf nahezu jeder Maschine liefen, wurden Programme wie GridSphere teilweise sehr stark angepasst. Der Einsatz einer Legacy-Anwendung, wie die des eigentlichen PID-Generators, in einem Grid-Umfeld ist nachweislich realisierbar. Somit kann sie von allen Diensten innerhalb des Grids angesteuert werden, wenn dies für Verarbeitung im Sinne datenschutzrechtlicher Belange notwendig ist.

## **12. Schlussfolgerungen**

Instant-Grid ist ein Instrument für ad hoc-Grid-Anwendungen. Die Anpassungen in diesem konkreten Anwendungsfall mussten zwar manuell konfiguriert werden, könnten bzw. sind in der Weiterentwicklung Teil des Instant-Grid geworden. Das zusätzliche Einbinden von Ressourcen konnte auch in einer virtualisierten Umgebung vorgenommen werden. Die zu verarbeitenden Daten, durch einen stets erreichbaren Dienst pseudonymisiert, beweisen ein Höchstmaß an verfügbarem Datenschutz, was bis dato immer als Hindernis bei der Verarbeitung medizinischer Daten in Grids gesehen wurde. Eine weiterentwickelte Version von Instant-Grid wurde erfolgreich im interdisziplinären Praktikum am Institut für Informatik Göttingen des Sommersemesters 2008 eingesetzt.

### **Danksagung**

Diese Arbeit wurde durch die D-Grid Projekte Instant-Grid und MediGRID unterstützt, gefördert durch das Bundesministerium für Bildung und Forschung (BMBF), Förderkennzeichen 01AK807 und 01AK803H.

### **Referenzen**

[1] <http://instant-grid.de/>

[2] <http://www.vmware.com/de/>

[3] <http://www.knoppix.org/>

- [4] <http://www.globus.org/>
- [5] <http://www.gridisphere.org/>
- [6] <http://tools.ietf.org/html/rfc3280>
- [7] <http://www.tmf-ev.de/>
- [8] <http://de.wikipedia.org/wiki/Vi>
- [9] <http://ant.apache.org/>
- [10] <http://tomcat.apache.org/>
- [11] <http://www.tmf-ev.de/Produkte/Uebersicht.aspx>
- [12] <http://jcp.org/aboutJava/communityprocess/final/jsr168/>



---

---

# **A two-tier approach to efficient workflow scheduling in MediGRID**

**Dietmar Sommerfeld**

*Gesellschaft für wissenschaftliche Datenverarbeitung mbH Göttingen*

**Harald Richter**

*Department of Informatics, Clausthal University of Technology*

## **Abstract**

In this report we describe a methodology to schedule application workflows on the Grid. The algorithm combines existing scheduling strategies for the Grid and for workflows to meet the dynamic characteristics of heterogeneous Grid resources. It employs a list scheduling heuristic and performance predictions to create a full-ahead schedule of Grid jobs, which are then distributed according to resource performance measurements.

## **1. Introduction**

The MediGRID project which is part of the German e-Science initiative D-Grid, has created a community Grid for researchers in the fields of medicine, biomedical informatics, and life sciences. Four types of pilot applications were selected for the first phase of MediGRID: bioinformatics, image pro-

cessing, biomedical ontology, and clinical research applications. MediGRID has set up a service Grid that uses the Globus Toolkit 4 (GT4) as its Grid middleware. GT4 is based on stateful web services and provides basic services for security infrastructure, job execution, data management, monitoring and resource discovery.

On top of GT4, MediGRID employs an advanced workflow system for orchestrating the distributed execution of workflow applications on Grid resources. The important characteristic of these applications is that they are not monolithic and single-executable applications, but incorporate multiple dependent computational modules, and entail transfer and storage of a large amount of data. The execution of these applications involves the concurrent and sequential execution of multiple programs, and the automatic and timely data transfer between programs which are also called tasks in workflow terms. A very important issue in executing a scientific workflow application in computational Grids is how to map and schedule workflow tasks onto multiple distributed resources, and how to handle task dependencies in a timely manner to deliver the requested performance.

In this report, we describe a methodology to schedule application workflows that combines existing scheduling strategies for workflow tasks and Grid jobs in a performance effective approach with manageable complexity. This approach consists of two tiers. In the first tier a full-ahead schedule of the workflow tasks is created. This means the whole workflow is scheduled before its execution. Therefore, we use a list scheduling heuristic similar to the Heterogeneous Earliest-Finish-Time algorithm (HEFT) [1]. In the resulting schedule, all tasks are assigned ranks which determine static prioritizations. The second tier operates at runtime and processes the workflow tasks according to their priorities. It performs a just-in-time mapping of tasks to Grid resources and is equivalent to common meta-scheduling on the Grid. To improve the scheduling decision, it uses dynamic resource information and short-time predictions.

## **2. Related Work**

Scheduling of scientific workflow applications on the Grid is a challenge, and the number of approaches is as numerous as the projects dealing with this topic. Related work can be found in the domains of workflow scheduling and meta-scheduling. A comprehensive overview of meta-scheduling on the Grid is given in [2].

Important aspects of workflow scheduling are trade-offs between dynamism and long term look-ahead. In [3] Deelman proposes a scheduling strategy

based on initial partitioning of the workflow into sequential sub-workflows that are scheduled one after another. Prodan [5] applies genetic algorithms to schedule the whole workflow at once. Then it is rescheduled many times during the execution.

In [4] Wieczorek compares three scheduling algorithms for scheduling scientific workflows in Grid environments. The scheduling algorithms comprise a genetic algorithm, the HEFT algorithm, and a just-in-time algorithm, similar to the Condor DAGMan resource broker [6], which schedules the next task onto the best machine available without any long-term optimization strategy. Additionally, different scheduling strategies are compared including full-graph scheduling and an incremental workflow-partitioning strategy. In a series of experiments, it is shown in practice that the HEFT algorithm applied with the full-ahead scheduling strategy performs best compared to other approaches.

A general analysis of multi-criteria grid workflow scheduling is performed in [7]. It proposes novel taxonomies of the problem, based on the aspects (e.g. models, criteria) which influence the decision which scheduling strategy is most appropriate in a given case. Many of the existing workflow scheduling approaches for the Grid are then analyzed and classified according to the proposed taxonomies.

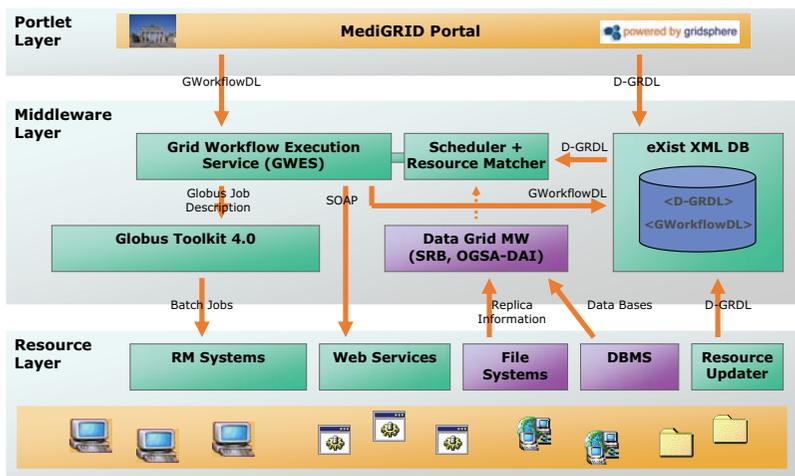
### **3. Current Scheduling in MediGRID**

In MediGRID, the execution of complex workflow applications consisting of several program executions is handled by the Grid Workflow Execution Service (GWES) [8]. This workflow management system is used in many European Grid projects, e.g. MediGRID, Services@MediGRID, BauVO-Grid, Instant-Grid, K-Wf Grid and CoreGRID [9], [10], [11], [12], [13], [14].

The workflow model for the GWES environment is based on Petri nets. Petri nets consist of places (symbolized by circles) and transitions (symbolized by squares). Places and transitions are connected by directed edges and always alternate with each other. In the workflow, transitions stand for conditional activities while places represent input and output data. Places contain so-called tokens if the input data are available. Transitions can only be activated when all input places of the transition have a token. When the transition is started, one token from every input place of the transition is taken, and when the transition is completed one token is given to every output place. This way, Petri nets model control and data flow between application compon-

ents. MediGRID workflows are specified with the XML-based Grid Workflow Description Language (GWorkflowDL).

The workflow management infrastructure consists of several components depicted in Fig. 1. The central service is the GWES which processes the workflow documents. The first action after initialization is the resource matching. GWES includes a Resource Matcher that searches for available instances of the program classes specified in the workflow. For this, GWES communicates with the eXist resource database that contains information about software instances available on the machines in the Grid. The resource database also contains machine descriptions with information about the utilization of the machines. Both types of information are specified by means of the XML-based D-Grid Resource Description Language (D-GRDL). Additionally, the eXist database stores the GWorkflowDL documents of active and completed workflows.



**Fig. 1: Architecture of the MediGRID middleware**

The scheduling in MediGRID is done by the scheduler included in GWES. When a transition is ready to execute, i.e. all input data are available, GWES automatically selects one of the suitable machines based on their current utilization. This information is periodically collected by the Resource Updater and stored in the resource database. The scheduling algorithm calculates a quality value, i.e. metric, between 0 (busy) and 1 (idle) for all resources. The calculation differs between workstation and cluster resources. Workstations are single machines that are run in time sharing. In this case, the scheduler

uses the „one minute load“ for the calculation [15]. Clusters consist of many machines that are used in space sharing and that are controlled by a local resource management system (LRMS). For clusters, the metric is calculated by means of the number of running and waiting jobs. In the end, the scheduler chooses randomly one resource from the sub list of resources whose metric is greater than a given threshold value (e.g. 0.6). The selection is randomized to prevent an overloading of machines in case of outdated or insufficient monitoring data. The scheduler provides a rough load balancing between resources and tries to improve the job throughput of the Grid.

After the scheduling, GWES performs automatic file stage-in, i.e. it organizes all necessary data transfers for the chosen resource. For file transfers, GWES uses the RFT [16] service from the underlying GT4 middleware. Then, GWES either invokes a web service on the selected resource or it creates a job description and submits a job there which executes a command-line program. The job submission is done via WS-GRAM [17], the job manager of GT4. If a file transfer fails or the job has not finished successfully, GWES performs a fault management and reschedules it on another machine.

#### **4. Workflow-level Scheduling**

The current scheduler implementation in MediGRID belongs to the class of just-in-time algorithms, which make the planning based on locally reasonable decisions. Research shows [4] that just-in-time algorithms can produce good results for independent Grid jobs and rather simple workflows provided that accurate resource performance information is available. However, they do not provide any full Petri-net-graph analysis for task dependencies, i.e. they do not consider the order of task execution for scheduling. Therefore, just-in-time schedules lack performance in complex application workflows with many concurrent tasks. This affects especially the class of strongly unbalanced (asymmetric) workflows with parallel threads that differ significantly in expected thread execution times. In this case, preference has to be given to the longer threads to allow all threads to finish within similar time. The execution of such workflows can be improved by employing a full-graph scheduling algorithm as the HEFT algorithm.

The HEFT algorithm is an extension to allow for heterogeneous environments for the classical list scheduling algorithm based on directed acyclic graphs (DAG). HEFT is a simple but computationally inexpensive algorithm, which schedules a workflow by “backward” traversing the directed graph from output to input, constructing an ordered list of tasks, and mapping the tasks to resources.

The HEFT algorithm consists of 3 phases [4]:

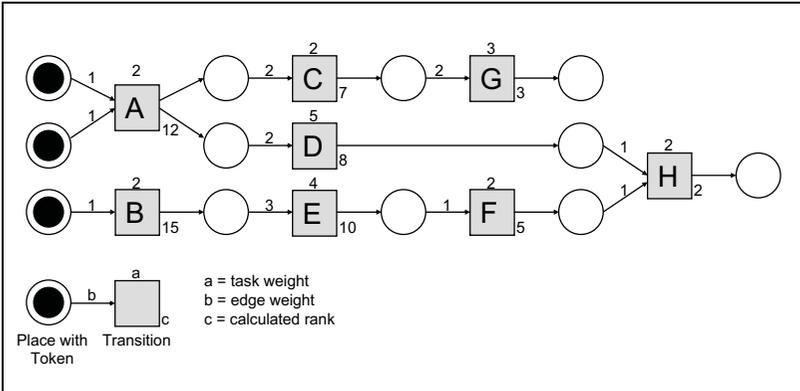
1. **Weighting:** it assigns weights to the nodes and edges in the graph;
2. **Ranking:** it creates a sorted list of tasks, ordered by their priorities;
3. **Mapping:** it assigns tasks to resources.

In phase 1, the weights assigned to nodes correspond to predicted execution times of the tasks while the edge weights correspond to predicted data transfer times between the resources. HEFT assumes these times to be known. In environments with homogeneous resources, the weights directly reflect the predicted times. In heterogeneous environments, the weights must be adjusted considering variances in execution times on resources, and different data transfer times on data links. Several adjustment methods were proposed and compared. Each of them provides another accuracy with respect to the considered scenario. The common method is to take the arithmetic average over all resources.

In the ranking phase 2, the workflow graph is traversed backward, and a rank value is assigned to each of the tasks. The value denotes the task's priority, thus a higher rank means a greater priority. The rank of a task is equal to the tasks' weight plus the maximum successive weight. This means for every edge leaving the task, the edge weight is added to the previously calculated rank of the adjacent node, and the maximum of the summations is chosen. In the end, the tasks are sorted by decreasing rank order. This results in an ordered ranking list.

In the mapping phase, tasks from the ranking list are mapped to the resources one after the other, and each task is assigned to that resource which minimizes the task's earliest expected finish time.

Figure 2 shows the ranks of HEFT for a simple example workflow graph.  $a$  and  $b$  are weights and  $c$  is the calculated rank.  $a$  denotes the average predicted execution time, and  $b$  denotes the average predicted data transfer time. In the example, the weights are arbitrary. The ranking of tasks is: B, A, E, D, C, F, G and H.



**Fig. 2: Example workflow graph with weights and ranks calculated by HEFT**

For the application on Petri net graphs, some modifications of the HEFT algorithm are necessary. Transitions are not connected directly with each other, but separated by places. Grid data transfer is performed when a token moves from the place to the next transition. That is why the weight of the transfer is written atop the input edges of the transition. Output edges do not have weight values.

In contrast to DAGs, Petri nets can contain control flow transitions which contain conditions for firing and loops. Both can only be evaluated at runtime. When the full-ahead workflow-scheduling is applied, control flow constructs are ignored, because evaluation cannot be made beforehand. This way, the branch with the higher weight will be considered for the ranking. If the branch leads to a loop, the loop has to be detected to prevent the traversal of the graph from looping forever. For the ranking we assume loops to be taken exactly once. This is achieved by virtually unrolling one iteration of the loop.

The value of the predicted execution time  $a$  of a transition is derived from previous executions of the same task. To this end, the runtime of each job is measured during execution. As the job runtime can vary quite much, we use exponential smoothing to estimate the next value based on previous values. Exponential smoothing is equivalent to a first order infinite impulse response low pass filter. The prediction is calculated as  $a_t = m' + (1 - m') * a_{t-1}$ , where  $m' = m * p$ .  $m$  denotes the latest measured execution time, and  $a_{t-1}$  is the previous predicted value of  $a$ .  $p$  is the performance factor of the resource and used to normalize  $m$  in comparison to a reference system with

2,66 GHz Intel Xeon processors.  $p$  has to be calculated by means of SPEC benchmarks for all Grid resources.  $\alpha \in [0,1]$  is called smoothing factor. We use  $\alpha = 0,3$  which gives more weight to the past for a stronger smoothing. Using exponential smoothing also facilitates the integration into the GWES environment, because only one value of  $m$  and the current value of  $a$  have to be stored in the D-GRDL description of the software instance in the resource database.

The data transfer time depends on the data size and the bandwidth of the network link between the resources. Predicting these values ahead of the workflow execution is difficult because neither data size nor the sites between which data transfer will take place are known. Additionally, data transfer rates vary during workflow execution time. Because of this, prediction of  $b$  would require a large set of measurements covering multiple previous values for all combinations of sites. This would imply major changes in the existing resource database. Therefore, for the workflow-level scheduling we assume instead that tasks produce output data in the amount of 10 MB, and that bandwidth behaves as a normally distributed random variable with a mean of 10 Mbit/s and variance 4. The transfer setup time is assumed to be 2 seconds. These assumptions are supported by practical experience and result in an average transfer time of 10 seconds.

Another aspect that has to be considered is potential queue wait times that are imposed on tasks at resources and that lead to a prolongation of execution time. Many Grid-scheduling research-groups assume a Grid model with high availability and good control over the resources by the scheduler, which is often the case for scientific workflows executed in research institutions. However, in the D-Grid environment availability of resources is limited and no control exists over the site level resource management systems. From the perspective of the site scheduler, the Grid scheduler only has the role of a “power user” that competes with other users, e.g. other Grid schedulers.

At present, MediGRID resources are highly utilized and waiting times at clusters can last up to hours. For small jobs, queue waiting time considerably exceeds the actual execution time. On the other hand, determining earliest available times of processors for task execution is very difficult for Grid schedulers because site level schedulers do not provide sufficient information. Every site uses a custom configuration of queues, and processors can be shared among queues or dedicated exclusively. Usually the information from the LRMS does neither allow to reliably determine the number of free processors nor the total number of processors. The only statistics commonly available are the number of running and waiting jobs. However, few D-Grid sites do not even provide this information because of nondisclosure. Additio-

nal to the lack of information, Grid schedulers have no control over the site scheduler's policy, and thus over the prioritization of the jobs waiting in the queues.

These circumstances lead to the conclusion that static full-ahead workflow scheduling is inappropriate in the D-Grid environment. Therefore, we only employ the first two phases of the HEFT algorithm to calculate static priorities for the Petri net transitions before the actual processing of the Petri net starts. For the mapping phase that assigns tasks to resources, we employ an improved version of the existing dynamic just-in-time scheduling in MediGRID.

## 5. Grid-level Scheduling

The second tier in the proposed approach performs a just-in-time Grid scheduling based on dynamic resource data, short-time predictions and additional information available at runtime.

During the workflow processing, transitions that are ready to execute are placed in a queue within GWES. While GWES processes many workflows at the same time, the internal queue holds tasks from several workflows. The operation principle of the new approach is to schedule tasks belonging to the same workflow in the order of their ranks calculated by phase 1 and 2 of the HEFT algorithm. Among different workflows, tasks are prioritized equally to provide fairness between workflows. This means, tasks from different workflows are processed with the first in first out principle.

While the only optimization goal of the workflow-level scheduling was to decrease the execution time of a single workflow, joint Grid-level scheduling of tasks of all workflows allows for overall improvements. The most important goal is the reduction of execution time but it is extended by Grid-wide optimizations. Like the current implementation, the scheduler will try to balance load across the resources and improve job throughput.

As described in section 3, the current MediGRID scheduling randomly selects a resource whose quality value is greater than the threshold. We propose to improve resource selection with regard to data locality. Data transfer times can be avoided if tasks are scheduled onto compute resources that already have the required data available. This requires the introduction of an interface between GWES and the Data Grid middleware. If a data transfer is performed, preference will be given to resources with a fast network connection to the data source. To this end, the data transfer rates between the MediGRID sites are continuously measured using the Grid benchmarking service Jawari

[18]. During just-in-time scheduling, source and destination of the transfer as well as data size are known, therefore it is possible to predict transfer time,  $t_{\text{transfer}} = t_{\text{setup}} + \text{size} / \text{bandwidth}_{(\text{source}, \text{dest})}$ .

Test measurements in D-Grid have shown that variance in bandwidth is acceptable for short-time predictions. However, in contrast to common opinion the bandwidth is usually not equal in both transfer directions between D-Grid sites. After data transfer, the replica has to be registered in the Data Grid middleware so that it is available for future job executions.

Due to differences in queue wait times it will often be advantageous to transfer data and to perform the computation on a resource with less queuing delay. However, to correlate transfer and queuing delay, the current algorithm that calculates a quality metric in  $[0,1]$  has to be replaced by a new method that allows to estimate queue wait time. The estimation, works similar to the prediction of job execution time in the previous section. It uses exponential smoothing to estimate queue wait time based on previous measurements. The calculation is  $s_t = \alpha n + (1 - \alpha) s_{t-1}$ , where  $n$  denotes the latest measured queue wait time, and  $s_{t-1}$  is the previous estimated value of  $s$ .  $\alpha$  is the smoothing factor and set to  $\alpha = 0,3$  to compensate fluctuations. Several values should be calculated for different job durations because some sites have exclusive nodes for short and long running jobs or employ backfilling. The values are stored for each resource in the D-GRDL machine description in the resource database.

Another factor that adds to pre-execution delay is the middleware overhead. It is introduced by the middleware components involved in job submission. However, it is almost equal on all sites because they use the same middleware and much smaller than the queuing delay because in D-Grid queues are normally full. Therefore, middleware delay is neglected. In the result, our new scheduling method assigns tasks to resources based on expected delay until the jobs starts to execute. Execution delay is the sum of data transfer time  $t_{\text{transfer}}$  and queuing delay  $s$ . This resembles the mapping phase of the HEFT algorithm.

Workflow scheduling strategies that employ full-ahead scheduling on the complete workflow graph can use sophisticated task scheduling heuristics such as list scheduling or genetic algorithms but have problems to cope with the dynamism of the Grid. Therefore, they require subsequent rescheduling during the execution of the workflow. The two-tier approach does not suffer from this problem. Nevertheless, repetition of the workflow-level scheduling with phase 1 and 2 of the HEFT algorithm can be useful, for example in case

of a failure during job execution before the fault management of GWES tries to resubmit the task to another machine.

To apply the HEFT algorithm to Petri nets, some modifications of the algorithm were necessary to handle non-DAG characteristics of Petri nets as control flow transitions and loops. With the just-in-time Grid scheduling in the second part of the new approach, GWES keeps the capability to handle all Petri net control flow constructs as in the current implementation. Branching into a loop is another condition to trigger a rescheduling that corrects the ranking of the tasks.

## **6. Conclusions and Future Work**

In MediGRID, applications are described as Petri net workflows which are executed by the GWES workflow management system. In this report, we have described a new approach for the scheduling module of GWES which replaces the current just-in-time Grid scheduling by an algorithm that makes scheduling decision based on estimations of execution delay. Additionally, a preceding workflow-level scheduling phase is introduced that employs HEFT, as well as performance predictions for full-graph analysis.

The two-tier approach retains the features of the current implementation to effectively distribute workflow tasks on the resources. In the same time, it can improve the performance on complex application workflows, especially for unbalanced workflows. If the predictions of execution times are incorrect, the performance is expected to be similar to the current scheduling.

Table 1 compares the current algorithm with the proposed new two-tier approach using the taxonomies from [7].

	<b>current implementation</b>	<b>two-tier approach</b>
multiple criteria	x	x
multiple workflows	x	x
full-ahead planning	-	x
just-in-time	x	x
workflow optimization	-	x
grid-wide optimization	x	x

**Tab. 1: Comparison of scheduling methods**

Future work on the two-tier approach is the extension of prediction techniques to incorporate additional parameters and a practical implementation in MediGRID to allow for an evaluation of scheduling results.

## References

- [1] Haluk Topcuoglu , Salim Hariri , Min-you Wu, Performance-Effective and Low-Complexity Task Scheduling for Heterogeneous Computing, IEEE Transactions on Parallel and Distributed Systems, v.13 n.3, pages 260-274, March 2002.
- [2] J. Nabrzyski, J.M. Schopf, J.Weglarz. (Eds.), Grid Resource Management - State of the Art and Future Trends, Kluwer Academic Publisher, 2004.
- [3] Ewa Deelman, James Blythe, Yolanda Gil, Carl Kesselman, Gaurang Mehta, Sonal Patil, Mei-Hui Su, Karan Vahi, and Miron Livny. Pegasus: Mapping scientific workflows onto the grid. In European Across Grids Conference, pages 11-20, 2004.
- [4] Marek Wieczorek, Radu Prodan, and Thomas Fahringer. Comparison of Workflow Scheduling Strategies on the Grid. PPAM 2005, LNCS 3911, pp. 792-800. Springer-Verlag Berlin Heidelberg 2006.
- [5] Radu Prodan and Thomas Fahringer. Dynamic Scheduling of Scientific Workflow Applications on the Grid using a Modular Optimisation Tool: A

Case Study. In 20th Symposium of Applied Computing (SAC 2005), Santa Fe, New Mexico, USA, March 2005.

[6] DAGMan (Directed Acyclic Graph Manager). <http://www.cs.wisc.edu/condor/dagman/>

[7] Marek Wiczorek, Andreas Hoheisel, Radu Prodan. Taxonomies of the Multi-criteria Grid Workflow Scheduling Problem. In Proceedings of the CoreGRID Workshop on Grid Middleware, Dresden, Germany, Springer-Verlag, June 2007.

[8] A. Hoheisel. Grid Workflow Execution Service - Dynamic and interactive execution and visualization of distributed workflows. In Proceedings of the Cracow Grid Workshop 2006, Cracow, 2006.

[9] MediGRID. <http://www.medigrd.de>

[10] Services@MediGRID. <http://services.medigrd.de>

[11] BauVOGrid. <http://www.bauvogrid.de>

[12] Instant-Grid. <http://www.instant-grid.de>

[13] K-Wf Grid. <http://www.kwfgrid.eu>

[14] CoreGRID - Network of Excellence. <http://www.coregrid.net>

[15] A. Hoheisel and H. Rose. Konzept für das Scheduling von Workflow-Aktivitäten in Instant Grid. Technical report, Fraunhofer Institut für Rechnerarchitektur und Softwaretechnik, June 2006.

[16] Reliable File Transfer (RFT) Service, <http://www.globus.org/toolkit/docs/4.0/data/rft/>

[17] Web Service Grid Resource Allocation and Management, <http://www.globus.org/toolkit/docs/4.0/execution/wsgram/>

[18] Ely de Oliveira. <http://www.jawari.net>



---

---

# **Instant-Grid – eine Live-CD für die Demonstration, die Entwicklung und den Test von Grid-Anwendungen**

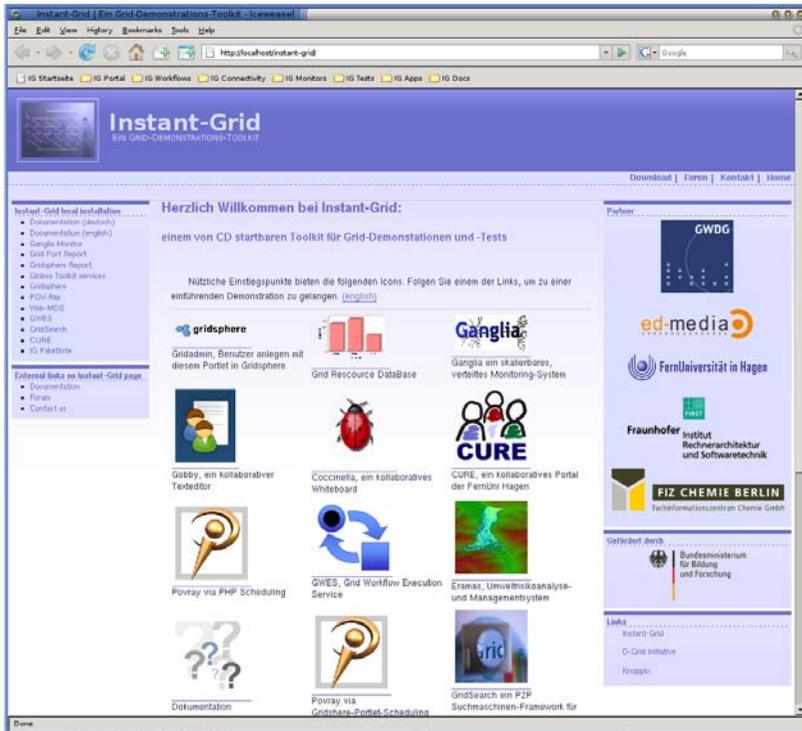
**Ulrich Schwardmann**

*Gesellschaft für wissenschaftliche Datenverarbeitung mbH Göttingen*

*Instant-Grid wurde finanziell gefördert durch das Bundesministerium für Bildung und Forschung unter dem Förderkennzeichen 01AK807.*

## **1. Aufgabenstellung**

Instant-Grid stellt ein Toolkit zur Demonstration der Grid-Technologie und ihres Einsatzes in ausgewählten Beispielanwendungen bereit und ermöglicht die Entwicklung eigener Grid-Anwendungen. Es trägt damit zur Etablierung von e-Science bei, indem es die Kenntnis über die zugrunde liegende Grid-Technologien verbreitet und diese Möglichkeiten in einer wachsenden Zahl von Anwendungen realisiert und damit nutzbar gemacht werden.



Die im Instant-Grid bereitgestellten Anwendungen demonstrieren die für das Grid typische Arbeitsweise mit transparentem Zugang zu unterschiedlichen und verteilten Ressourcen, zeigen den durch den Einsatz von Grid-Strukturen erzielbaren Leistungsgewinn auf und geben einen Einblick auf die durch e-Science neu sich bietenden Arbeitsmöglichkeiten. Die Anwendungen erlauben auch Nicht-Fachleuten, die Aufgabenstellungen interaktiv zu modifizieren, und liefern intuitiv verstehbare Resultate.

Das Instant-Grid gibt Wissenschaftlern, Studenten und anderen potentiellen Nutzern von e-Science die Möglichkeit, ihnen zur Verfügung stehende vernetzte Rechnerressourcen ohne weitere Kenntnisse und ohne Konfigurationsaufwand zu einem lokalen Grid zu konfigurieren. In diesem „privaten“ Grid können dann mit den im Instant-Grid bereitgestellten Grid-Anwendungen Erfahrungen gesammelt werden und eigene Anwendungen für die Nutzung in einem Grid vorbereitet werden.

Im Ergebnis bietet das Projekt eine mit minimalen Aufwand zu installierende, die vorhandene Umgebung zunächst nicht verändernde, weitestge-

hend vorkonfigurierte Grid-Umgebung in Form eines von einer CD-ROM startbaren PC-Clusters. Die Software besitzt eine hohe Stabilität und demonstriert dem Nutzer die Vorteile des Grid auf einfache Weise.

Über die Bereitstellung einer solchen CD hinaus stellt Instant-Grid mit dem Web-Portal <http://instant-grid.de> eine Schnittstelle zwischen Interessierten, Anwendern und Entwicklern zur Verfügung. Instant-Grid versteht sich als Initiative innerhalb D-Grid, ist dort Partnerprojekt und verwendet wesentliche Teile des D-Grid-Software-Stacks.

## 2. Partner im Instant-Grid-Projekt

### 2.1 Verantwortlicher Partner

	Gesellschaft für wissenschaftliche Datenverarbeitung mbH Göttingen (GWDC) Am Faßberg 11, 37077 Göttingen Ansprechpartner: Dr. Ulrich Schwarzmann, Prof. Dr. Oswald Haan
---	--

### 2.2 Assoziierte Partner

	Fraunhofer-Institut für Rechnerarchitektur und Softwaretechnik (FIRST) Kekuléstr. 7, 12489 Berlin Ansprechpartner: Dipl.-Phys. Andreas Hoheisel
	ed-media education & media GmbH Hüttenweg 43, 67659 Kaiserslautern Ansprechpartner: Dipl.-Ing. Steffen Franke
	Fernuniversität Hagen, Fakultät für Mathematik und Informatik, Lehrgebiet Rechnerarchitektur 58097 Hagen Ansprechpartner: Prof. Dr. Wolfram Schiffmann, Dipl.-Ing. Ludger Menzel
	FIZ Chemie Berlin Franklinstraße 11, D-10587 Berlin Ansprechpartner: Dr. Michael Langner

## 3. Komponenten der Instant-Grid-CD

Die Instant-Grid-CD stellt einen Mechanismus bereit, um Rechner in einem Netz durch Neustart zu einem Cluster zu konfigurieren und auf diesem durch Installation einer Grid-Middleware lokale Grid-Dienste bereitzustellen, die für die mitgelieferten Grid-Anwendungen eingesetzt werden können.

Damit das Instant-Grid einen breiten Kreis wissenschaftlicher Anwender anspricht, sind für seine Handhabung keine besonderen Vorkenntnisse notwendig. Dies wird durch einen automatisierten Konfigurationsmechanismus erreicht, der das Instant-Grid auf handelsüblichen vernetzten Rechnern ohne Benutzereingriff installiert.

Dieser Mechanismus zur Konfiguration und Installation des lokalen Grid und der Grid-Anwendungen kann im Regelfall ohne besondere Kenntnisse von Rechner- und Netzkonfiguration oder Grid-Strukturen gestartet werden, da er automatisch und ohne äußeren Eingriff abläuft. Dabei wird die Knoppix-Technologie eingesetzt, die eine zuverlässige Hardwareerkennung für alle gängigen Komponenten bereitstellt.

Die Robustheit und Stabilität der Instant-Grid-Konfiguration wurde im Projektverlauf in Zusammenarbeit mit externen Industrie-Partnern kontrolliert.

### **3.1 Cluster-Installation**

#### **3.1.1 Betriebssystem und Softwarebasis**

Instant-Grid basiert auf Knoppix (<http://www.knopper.net/knoppix/>). Knoppix ist eine komplett von CD lauffähige Zusammenstellung von GNU/Linux-Software mit automatischer Hardware-Erkennung und Unterstützung für eine Vielzahl möglicher Konfigurationen. Für die Verwendung im Instant-Grid musste in verschiedenen Versionen der Umfang von Knoppix stark reduziert werden. Die derzeitige Version 5 wurde als Grundlage für einen kompletten Neuaufbau der Instant-Grid-CD mit allen bis dahin vorhandenen Cluster- und Grid-Diensten verwendet.

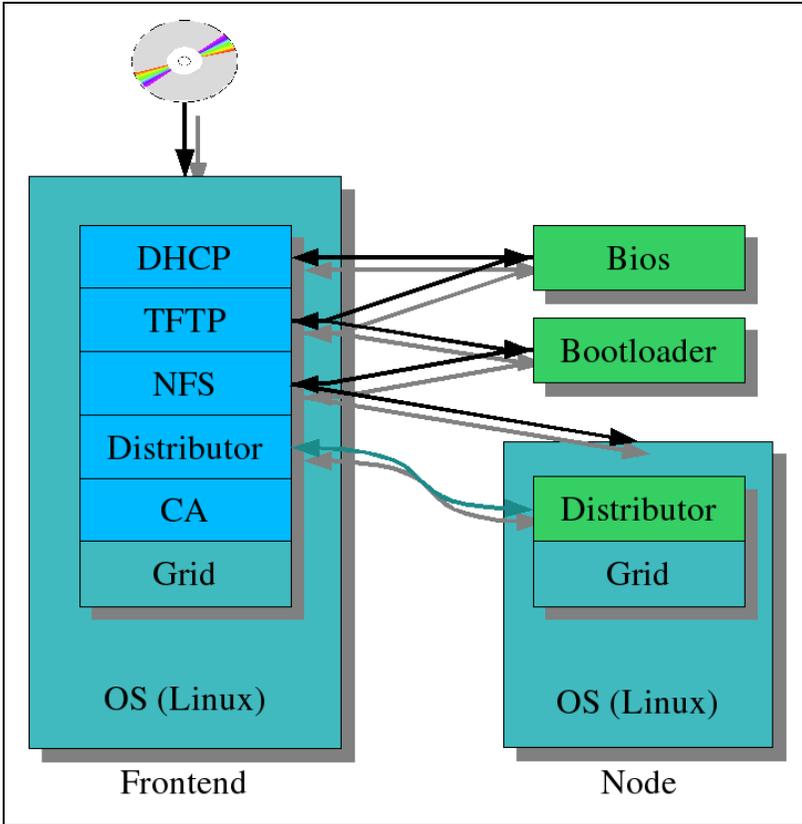
#### **3.1.2 Netzwerk-Konfiguration**

Grundsätzlich wird für den Start des Instant-Grid von einem lokalen Subnetz ausgegangen. Um den vielfältigen möglichen Gegebenheiten in den lokalen Netzwerken der Nutzer gerecht zu werden, wurden zwei Strategien verfolgt: Der überwiegende Teil wird automatisch durch ein intelligentes Skript konfiguriert. Für den Wunsch nach individueller Anpassung ist im Bootvorgang eine einfache Konfigurations-Möglichkeit integriert.

#### **3.1.3 Terminal-Server**

Unter diesem Begriff sind verschiedene Server zusammengefasst, die auf dem Cluster-Frontend vorkonfiguriert von der CD-ROM gestartet werden, um im Folgenden weiteren Rechnern (Clients) den Start und die Integration in den Cluster per PXE-Boot zu erlauben. Dazu gehören vor allem ein

DHCP-Server für die Zuteilung der IP-Nummern und die Bekanntgabe der PXE-Boot-Informationen, ein TFTP-Server, auf dem ein Bootloader, Kernelimage und Miniroot für die Clients bereitgehalten werden, und ein NFS-Server, von dem die Clients das eigentliche Root-Verzeichnis (auf der CD-ROM enthalten) sowie eventuell vorbereitete Verzeichnisse auf der lokalen Festplatte des Frontend einhängen können.



### 3.1.4 Terminal-Client

Die Klienten starten nach etablierten Mechanismen Linux über das Netzwerk per PXE-Boot. Zu den Aufgaben des Terminalclients gehören das Einhängen der vom Frontend exportierten Verzeichnisse und die Anmeldung im Cluster und bei der Grid-Middleware. Außerdem gibt es einen Client-Daemon als Gegenseite für den Server-Daemon zum Kopieren von Konfigurationsdateien.

### 3.1.5 Dynamische Ressourcen

In Instant-Grid können Klienten während dessen gesamter Lebensdauer, also solange der Frontend läuft, hinzukommen oder unerreichbar werden, denn die Ressourcen in einem Grid müssen, im Unterschied zu vielen Cluster-Lösungen, als dynamisch angesehen werden. Da die Topologie des Netzwerkes sowohl auf dem Frontend als auch auf den Klienten bekannt sein muss, wird ein Mechanismus benötigt, mit dessen Hilfe die Netzwerktopologie ermittelt und verteilt werden kann.

In Instant-Grid wird dazu vom Frontend aus das lokale Instant-Grid-Netzwerk nach verfügbaren Klienten durchsucht (durch die IP-Collector-Komponente), und die Konfigurationsdateien werden aktualisiert und auf ein mittels NFS exportiertes Dateisystem kopiert (durch die server-seitige Distributor-Komponente). Dieses Dateisystem wird auf den Klienten periodisch untersucht, und aktualisierte Konfigurationsdateien werden gegebenenfalls übernommen (durch die klientenseitige Distributor-Komponente). Auf diese Weise werden Änderungen im Instant-Grid innerhalb einer Minute auf allen Klienten bekannt.

In Instant-Grid sind Modifikationen an den lokalen Massenspeichern normalerweise unnötig und nur nach expliziter Freigabe möglich. Für eigene Applikationen und Daten kann optional ein Verzeichnis auf einer lokalen Speicherressource des Frontends eingerichtet werden, welches dann automatisch per NFS an die Klienten exportiert und von diesen genutzt wird.

## 3.2 Grid-Installation

Das eigentliche Toolkit zur Demonstration von Grid-Technologie und ihres Einsatzes besteht aus vier Komponenten: *Grid-Middleware*, *Monitorfunktionen*, *Grid-Anwendungen* und den *Zugangs-Schnittstellen*. Als *Grid-Middleware* kommt Globus Toolkit 4.0 in Instant-Grid zum Einsatz, da dies mit der D-Grid-Integrationsplattform kompatibel ist und die geplanten Anwendungen unterstützt. Zusätzlich wurden in Absprache mit dem DGI-Projekt weitere Komponenten der D-Grid-Integrationsplattform (Gridsphere, Garglia, GWES) aufgenommen.

### 3.2.1 Grid-Middleware: Basissystem

Zu den Funktionen dieser Komponente gehören Ressourcen-Management, Daten-Management, Informationsdienste und Sicherheit (Authentifizierung/Autorisierung) im Grid. Dabei wird die Middleware so vorkonfiguriert, dass Instant-Grid als ein lokales Modellgrid fungiert, bei dem sowohl Rechner wie Nutzer automatisch mit einem Zertifikat versehen werden. Basis dieser

Funktionalität ist Globus Toolkit in Version 4, was seit Anfang 2006 im Instant-Grid eingesetzt wird.

In Instant-Grid stehen auch die Daten-Management-Dienste von Globus Toolkit zur Verfügung. Der Basisdienst ist GridFTP [6], mit dessen Hilfe Dateien zwischen den Rechnern kopiert werden können. Auf GridFTP setzt der Dienst RFT [7] (Reliable File Transfer) auf, der Daten-Transfers zusätzlich in einer SQL-Datenbank speichert. Auf diese Weise kann der korrekte Abschluss von Transfers gewährleistet werden.

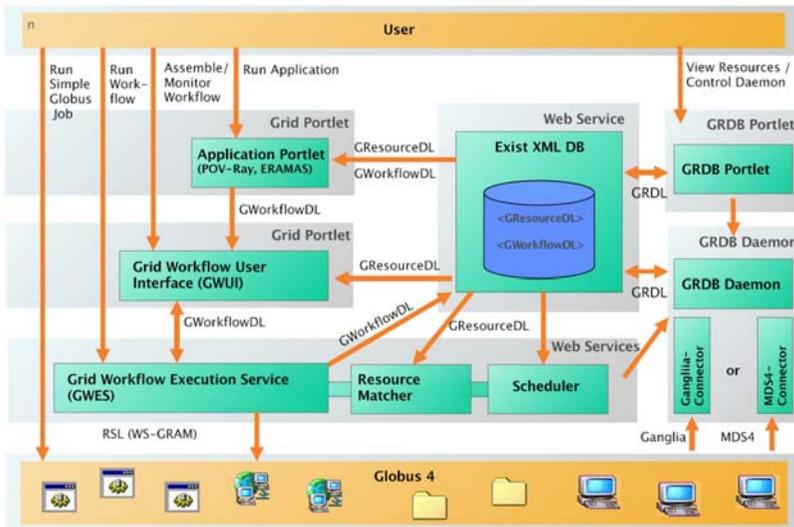
### 3.2.2 Grid-Middleware: Workflowsystem

Im Rahmen des Projektes „I-Lab“ wurde von Fraunhofer-FIRST für das Fraunhofer Resource Grid (FhRG) ein Workflow-Management-System für Grid-Anwendungen entwickelt, welches auf dem Globus Toolkit 2 aufsetzt und die Ausführung von komplexen, lose gekoppelten Anwendungen mit räumlich verteilten Grid-Ressourcen auf einfache Art und Weise ermöglicht.

Das Grid-Workflow-Management-System des Fraunhofer FIRST steht als wesentliche Komponente im Bereich der Middleware in Instant-Grid zur Verfügung.

Die zentrale Systemkomponente hierbei ist der *GWES (Grid-Workflow-Execution-Service)*, der abstrakte Grid Workflows – die mit Hilfe von intuitiv verständlichen Petrinetzen modelliert werden – automatisch auf die konkrete Grid-Infrastruktur abbildet und dort ausführt. Dieser, wie auch der ResMatcher zur Suche passender Ressourcen und der Scheduler (Prorator), der die Auswahl der Ressourcen optimiert, sind in wesentlich verbesserter und stabiler Form Bestandteil von Instant-Grid. Implementiert sind ferner weitgehend unverändert Exist als XML-Datenbank, die als Container und Austauschmedium für die Beschreibungen der im Grid vorhandenen Ressourcen und der formulierten Prozesse dient, und die Linux-Toolbox, die Beispiel-Webservices und Graph-Layouts bereitstellt.

Neu im Bereich GWES ist die Möglichkeit des Rechnens mit EdgeExpressions, was den Einbau von Zählern in einen Workflow wesentlich vereinfacht. Außerdem gab es im Rahmen einer Diplomarbeit Entwicklungen hin zu fehlertoleranten Workflows, sodass bei Fehlschlag die Aktivität auf einer anderen Ressource ausgeführt werden kann, und, falls die Ausführung bei allen Kandidaten fehlschlägt, das Resource-Matching vollständig wiederholt wird. Bei den GWES GridSphere Portlets, die als Benutzerschnittstelle für den Grid Workflow Execution Service dienen, und als GridSphere Portlet implementiert sind, können nun Uploads von Workflow-Beschreibungen getätigt werden, das Auflisten von aktuellen Workflows ist jetzt ebenfalls möglich.



Die Schnittstellen der Grid-Ressource-DB (GRDB) und das Portlet zur Darstellung ihres Inhalts, sind im Rahmen von Instant-Grid auf Basis der GResourceDL entwickelt worden. (Siehe auch *Entwurf und Implementierung einer Ressourcen-Datenbank für das Instant-Grid-Projekt der GWDG*, Willner, Masterarbeit im Studiengang „Angewandte Informatik“.) Diese Komponenten sind in der Folge im Projekt weiter verbessert und stabilisiert worden. Insbesondere wurde die Ressourcen-Beschreibungssprache auf GLUE-Syntax umgestellt und im Zuge dessen in D-GRDL umbenannt. Die Ressourcen-Datenbank GRDB ist mittlerweile Kandidat des Software-Stack des DGI der D-Grid-Initiative und wurde in die Middleware-Komponenten von MediGrid als zentrale Ressourcen-Datenbank eingefügt.

### 3.3 Monitoring

*Monitoring* ist ein wichtiger Bestandteil gerade des Instant-Grid-Konzeptes, da dieses Feature es erlaubt, Grid-Funktionalität sichtbar zu machen, was auf einem Demonstrations- und Schulungs-Toolkit von besonderer Bedeutung ist.



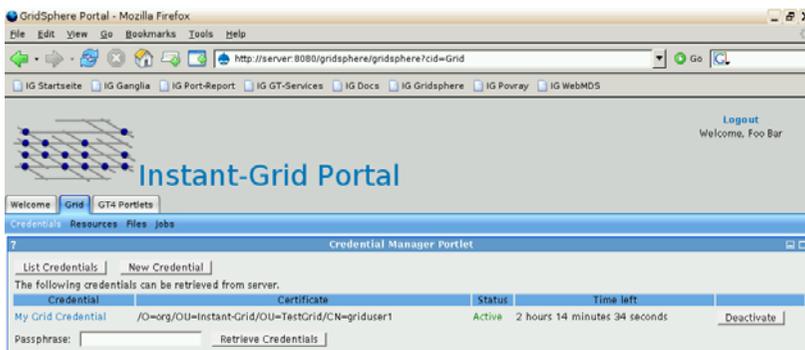
Neben dem Informationsgehalt ist hier auch eine ansprechende Darstellung erforderlich. Die Monitoring-Funktionen der Basis-Middleware erlauben die Abfrage statischer Eigenschaften von Rechnern im Grid sowie des Bearbeitungszustandes von Nutzeraufgaben. Durch die Erweiterung der Basis-Middleware um das Monitoring-System Ganglia (<http://ganglia.info/>) werden auch dynamische Eigenschaften der Rechner, wie beispielsweise die Systemlast, erfasst. Dadurch wird die Verteilung von Aufgaben auf die Ressourcen im Grid anschaulich gemacht. Ganglia eignet sich für diese Aufgabe nicht nur wegen seiner bewährten Stabilität in Hunderten Clustern weltweit, sondern auch wegen der bereits erfolgten Integration mit dem Globus Toolkit. Ganglia spielt mittlerweile auch im D-Grid für das Monitoring eine wichtige Rolle.

Für die automatische Nutzung der Ressourcen durch Grid-Workflows waren die Umsetzung der Monitoring-Parameter in eine entsprechende Beschreibungssprache (hier die GResourceDL) und deren Speicherung in einer für

die Workflows erreichbaren Datenbank, der Grid-Ressource-DB (GRDB), notwendig. Diese Arbeiten sowie ein Portlet zur Darstellung des Inhalts der GRDB wurden im Rahmen von Instant-Grid durchgeführt. Diese Software wird in den Software-Stack des DGI der D-Grid-Initiative integriert und wurde in die Middleware-Komponenten von MediGRID als zentrale Ressourcen-Datenbank eingefügt. In Instant-Grid sind diese Monitoring-Komponenten über das Gridsphere-Portal dargestellt.

### 3.4 Zugang

Der *Zugang* zum Instant-Grid erfolgt über ein Web-Portal oder alternativ über Kommandozeilen. Als Portal-Lösung ist GridSphere in Verbindung mit den Grid Portlets (<http://www.gridisphere.org/>) gewählt worden. Für die Nutzung des Portals ist eine Anmeldung erforderlich. Bei der Erstanmeldung wird automatisch eine neue Nutzererkennung einschließlich der für Globus notwendigen Zertifikate eingerichtet. Das Portal erlaubt den Nutzern den Zugriff auf sämtliche Hardware-Ressourcen des Instant-Grid, auf die Monitoring-Funktionen, die Dokumentation einschließlich der Tutorien und auf die meisten Anwendungen.



Nach dem Start von Instant-Grid erscheint automatisch die Hauptseite des Portals auf dem Frontend. Basis und Rahmen für das Instant-Grid-Portal bildet das Open-Source-Projekt Gridsphere [8]. Gridsphere ist eine Portallösung, die auf Portlets aufsetzt, die dem JSR-168-Standard entsprechen. Das Projekt besitzt eine große und rege Benutzergemeinde und erlaubt Entwicklern, als eines der Schlüsselemente des Projektes, die Nutzung von Portlet-Web-Applikationen, die aus dem Third-Party-Bereich stammen. Diese können dann innerhalb des Gridsphere-Portlet-Container administriert und gestartet werden.

Eine Portlet-Anwendung zur Grid-Benutzerverwaltung ist in Instant-Grid implementiert (s. vorige Abbildung). Normalerweise basiert Gridsphere auf einer eigenen Benutzerverwaltung und besitzt keine Schnittstelle zur Benutzerdatenbank des Betriebssystems. Daher müssen Grid-Nutzer für gewöhnlich sowohl auf Gridsphere-Ebene als auch auf der Ebene des Betriebssystems verwaltet werden. Der Grid-Administrator hat in dem Fall die notwendige Zuordnung für die Nutzer auf beiden Ebenen vorzunehmen. Dann muss der CA-Administrator Zertifikate für neu hinzukommende Benutzer ausstellen. Danach wird der Benutzer im Normalfall sein eigenes Grid-Credential erzeugen und dieses online in ein entsprechendes Repository für derartige Credentials einfügen (MyProxy). Das Benutzerverwaltungs-Portlet von Instant-Grid ermöglicht es, all diese Schritte gemeinsam und transparent durchzuführen: Die Portal-Besucher können neue Betriebssystem- und Portal-Benutzerkonten in wenigen Schritten anlegen – ohne jedes Hintergrundwissen über Benutzerverwaltung oder die Sicherheits-Infrastruktur im Grid. Darüber hinaus gibt es weitere portlet-basierte Web-Anwendungen: Grid-Portlets können für das Submittieren von Jobs verwendet werden. Es gibt die Möglichkeit für Datei-Transfer und das Verwalten von Credentials im Instant-Grid, womit die wesentlichsten Portlets zum Betrieb eines Grid-Portals in einer Produktionsumgebung bereitgestellt sind.

### **3.5 Anwendungen**

Die folgenden Grid-Anwendungen sind auf der Instant-Grid-CD verfügbar:

#### **3.5.1 ERAMAS**

*ERAMAS (Environmental Risk Analysis and Management System)* (<http://www.eras.de/>) ist ein rechnergestütztes Analyse- und Managementsystem schadstoffbedingter Umweltrisiken, welches von Fraunhofer FIRST zusammen mit der Dresdner Grundwasser Consulting GmbH und dem Ingenieurbüro Beger für Umweltanalyse und Forschung entwickelt wurde. Für Instant-Grid wurde der Teilbereich *Schadstofftransport in der Atmosphäre* als Beispielanwendung angepasst und öffentlich verfügbar gemacht. Diese Anwendung ist in Instant-Grid als Workflow implementiert.

#### **3.5.2 POV-Ray**

Das Raytracing-Programm POV-Ray wurde in Instant-Grid als verteilte Grid-Applikation überarbeitet. Das Scheduling geschieht über PHP-Skripte, die auf die benötigten Services GRAM (Globus Resource Allocation Manager) und Grid-FTP des Globus Toolkit zugreifen. Es gibt Beispiele zur Berechnung einzelner Szenen, die über eine Aufteilung der Szene in

Abschnitte parallelisiert sind, sowie Beispiele zur Erstellung von Animationen. Die Parallelisierung geschieht hier über eine Aufteilung der zu berechnenden Frames. Diese Anwendungen sind um eine workflow-basierte Version, als ein Beispiel eines komplexen Workflows, ergänzt worden. Der Workflow ist über ein GridSphere Portlet im Portal erreichbar. Beide Versionen dienen gemeinsam der Anschauung der verschiedenen Zugänge zum Grid und der Verwendung von Ressourcen im Grid. Diese Anwendung spielt für die Verbreitung von Instant-Grid und von Grid-Technologie eine besondere Rolle, da ein bedeutender Teil der an Instant-Grid Interessierten aus der Community des Ray-Tracing stammt. Für Ray-Tracing können erhebliche Ressourcen notwendig sein, und Instant-Grid bietet dieser Community eine einfache Plattform zur Nutzung gebündelter Ressourcen über Grid-Technologie.

### 3.5.3 Kollaboration

The screenshot displays a collaborative environment. On the left, a window titled 'Coccinella' shows a contact list with 'server' selected. The main workspace is titled 'Chat mit server@server/coccinella' and contains a 3D ball-and-stick model of a molecule and its corresponding 2D chemical structure. The chemical structure is that of Ascorbic acid (Vitamin C), with the SMILES string C(C(O)C1=CC(=O)O1)O. A chat window titled 'Chat: server (server@server/coccinella)' is open, showing a conversation about Vitamin C. The chat history includes:

- [12.01] <server> Vitamin C oder Ascorbinsäure
- [12.01] <bombay> Aha, du meinst: (R)-5-[(S)-1,2-Dihydroxyethyl]-3,4-dihydroxy-5H-furan-2-on
- [12.02] <server> ja genau, die sieht so aus:
- [12.03] <bombay> Die Strukturformel hab ich gefunden:
- [12.04] <server> Wo sind die Kohlenstoffatome?
- [12.04] <bombay> Hier:

The interface also features a toolbar with icons for file operations (Speichern, Öffnen, Import, Senden, Drucken, Stopp) and a status bar at the bottom showing the Jabber ID 'server@server/coccinella'.

Als Beispiel für ein kollaboratives Werkzeug war zunächst die Software AccessGrid geplant. Bei dieser Software gab es aber, wider Erwarten, keine Entwicklung für GT4, die Grid-Middleware von Instant-Grid. Daher wurde entschieden, die für Instant-Grid wesentlichen Komponenten dieser Soft-

ware durch den kollaborativen Editor Gobby und für Whiteboard- und Chat-Funktionen die Kollaborationssoftware Coccinella in Instant-Grid zu integrieren. In diesem Zusammenhang war zusätzlich die Implementation eines Jabber-Servers notwendig.

### **3.5.4 Gridlabs**

In Zusammenarbeit mit dem Lehrgebiet Rechnerarchitektur der Fernuniversität Hagen wurde die Kollaborationssoftware GridLabs (auch CURE genannt) auf der Instant-Grid-CD implementiert. Die Instant-Grid-Version ist eine Simulation der auf dem Server der Fernuniversität Hagen verfügbaren Version. Sie enthält drei Demo-Räume für die lokale Nutzung von Instant-Grid und einen Raum, der auf dem Webserver in Hagen liegt, und die ebenfalls nutzbar ist, sofern eine Internetanbindung existiert. In den drei lokal nutzbaren Demo-Räumen wird ein programmierbarer Mikrorechner dargestellt, mit dem in einer Arbeitsgruppe Experimente durchgeführt werden können, sowie eine Webcam simuliert, die den Mikrorechner in seinen Aktivitäten beobachtet. Der Zugang zu den Laborräumen wird über ein Schlüsselvergabe-System geregelt.

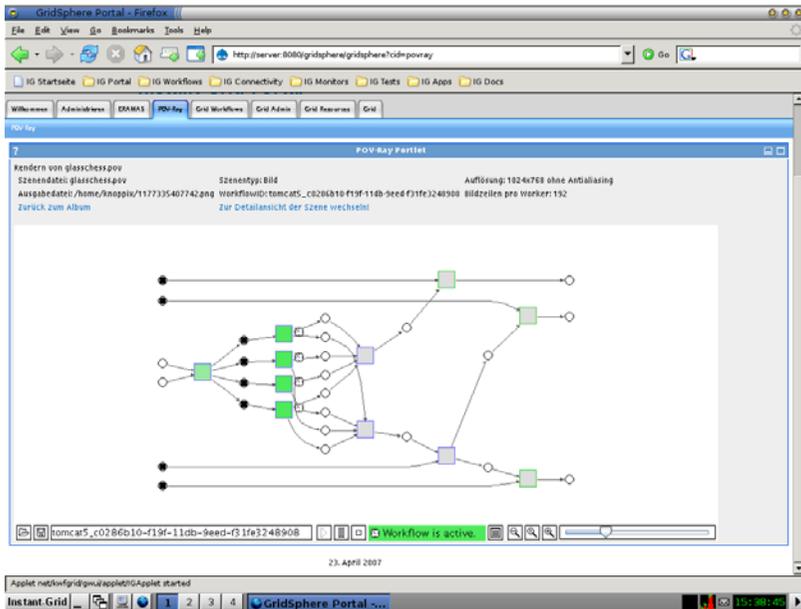
### **3.5.5 GridSearch**

Die Integration der Software GridSearch in Instant-Grid wurde in Zusammenarbeit mit dem FIZ-Chemie durchgeführt. GridSearch ist eine verteilte Suche im Grid. Im Instant-Grid werden Festplatten und Hotplug Devices (z. B. CD-ROMs und USB-Sticks) auf den Clients und Server indexiert. Als Formate für indexierbare Textkorpora sind z. Zt. HTML, PDF, .Doc und Ascii-Text möglich. Der Benutzer wird beim Hinzufügen einer durchsuchbaren Ressource sowohl auf den Clients als auch auf dem Server in einem Dialog gefragt, ob die Dateien auf dieser Ressource indexiert werden sollen. Bei Bestätigung dieser Frage wird ein Index erstellt. So entsteht ein Index über alle im Instant-Grid verteilt verfügbar gemachten Textkorpora, der dann wiederum für alle im Grid Beteiligten nutzbar ist.

### **3.5.6 Workflow-Anwendungen**

Im Rahmen des Workflow-Systems GWES steht dem Benutzer ein Baukasten weiterer Beispiele für Anwendungen zur Verfügung, die die Arbeitsweise des Workflow-Systems verdeutlichen und auch im Sinne einer Toolbox für die eigene Erstellung komplexerer Workflows verwendet werden können. Hierbei handelt es sich um einfachste Workflows wie String-Sorts, Token-Duplizierung, Benutzer-Interaktion, Schleifen, Workflow-Start im Workflow, Datei-Aufspaltung und -Zusammenführung, Datei-Transfer.

Aber es stehen auch komplexe Workflows wie die oben genannten für POV-Ray und ERAMAS sowie für DDFEM, einer Finite-Elemente-Simulation, oder PAUP, einem Algorithmus aus der Biologie, zur Verfügung.



### 3.6 Überlauf-Kapazitäten

Es wurde ein Konzept für die Nutzung von Überlauf-Kapazitäten im realen Grid entwickelt und umgesetzt. Dabei wurde für die Instant-Grid-Nutzer mit einer D-Grid-Nutzungsberechtigung die Möglichkeit geschaffen, Aufgaben vom lokalen Portal ins D-Grid zu delegieren. Damit aber auch diejenigen, die dazu (noch) nicht autorisiert sind, einen Weg zu realen Grid-Ressourcen vorfinden, wurden spezielle für Instant-Grid permanent verfügbare Ressourcen eingerichtet, die von Instant-Grid aus verwendet werden können.

Auf den lokalen Grid-Funktionalitäten von Instant-Grid aufbauend ist ein Konzept für die Nutzung von Überlaufkapazitäten im realen Grid entwickelt worden, das zum einen für diejenigen Instant-Grid-Nutzer, die ein D-Grid-Zertifikat für die Nutzung von D-Grid-Ressourcen besitzen, die Möglichkeit schafft, Aufgaben, die die lokalen Kapazitäten überfordern, ins D-Grid zu delegieren. Zum anderen soll aber auch denjenigen, die dazu (noch) nicht autorisiert sind, ein Weg bereitet werden, speziell für Instant-Grid eingerichtete, permanent verfügbare Ressourcen im Netz über die Grid-Schnittstellen des lokalen Instant-Grid zu verwenden.

Eine grundsätzliche Einschränkung für derartige Realisierungen stellen dabei die Netzwerk-Policies auf Seiten der jeweiligen Institutionen dar, innerhalb derer die Instant-Grid-CD genutzt wird. Um die Funktionen der Grid-Middleware (hier Globus-Toolkit) nutzen zu können, ist es notwendig, dass der Rechner über bestimmte Ports aus dem Internet heraus erreichbar ist und dass die IP-Adresse des Rechners im Internet bekannt ist. Diese Anforderungen können nur auf der Seite des Nutzers von Instant-Grid erfüllt werden und sind nicht durch Konfigurationen im Instant-Grid zu lösen.

### **3.6.1 Nutzung von speziellen Ressourcen für Instant-Grid**

Um Nutzern, die über keine Berechtigung zur Nutzung von D-Grid-Ressourcen verfügen, trotzdem Grid-Ressourcen bereitzustellen, werden von der GWDG Rechner zur Verfügung gestellt, die als Überlaufkapazität genutzt werden können. Aufgrund der Anonymität dieser Instant-Grid-Nutzer ist hier ein besonderes Konzept zur Sicherheit der GWDG-Ressourcen entwickelt worden. Diese speziellen Rechner sind mit der notwendigen Grid-Middleware ausgestattet worden.

Um den Job-Zustand verfolgen und die Ergebnisse von den GWDG-Ressourcen abholen zu können, müssen die oben genannten Voraussetzungen an die Netzwerkanbindung der Instant-Grid-Instanz gestellt werden.

### **3.6.2 Nutzung von Ressourcen des D-Grid**

Falls Instant-Grid-Nutzer gleichzeitig mit einem gültigen D-Grid-Zertifikat über die Berechtigung zur Nutzung von D-Grid-Ressourcen verfügen, sind diese von Instant-Grid aus als Überlaufkapazität ansprechbar. Die Verfolgung des Job-Zustandes und die Beobachtung des Ergebnisses sind über die Portale möglich. Sollen diese Aktivitäten darüberhinaus auf dem Instant-Grid-Portal durchführbar sein, müssen zur Kommunikation mit den D-Grid-Ressourcen ebenfalls die oben genannten Voraussetzungen an die Netzwerkanbindung der Instant-Grid-Instanz gestellt werden.

### **3.6.3 Notwendige Änderungen in der Instant-Grid-Konfiguration**

Die Standard-Konfiguration von Instant-Grid beruht auf dem fest vergebenen Namen des Servers im lokalen Netzwerk. Um den Grid-Zugang zu externen Ressourcen zu erlauben, wurde in die Konfiguration von Instant-Grid zusätzlich der von der lokalen Institution des Nutzers vergebene, dynamische und im Internet bekannte Name des externen Interfaces aufgenommen.

Die Anbindung externer Ressourcen, insbesondere die des D-Grid sowie die allgemein verfügbaren bei GWDG, ist in den Komponenten der Grid-Middleware verankert worden. Die von Instant-Grid selbst eingesetzte Firewall wurde entsprechend den Vorgaben von D-Grid angepasst.

### **3.6.4 Benutzerführung**

Eine anschauliche Benutzerführung ist für diese beiden Grid-Szenarien implementiert worden, indem diese Art der Nutzung in einer Dokumentationsseite unter dem entsprechenden Button im Browser ausführlich beschrieben wird. Zum anderen ist die Nutzung selbst durch den Button im Browser klar und einfach strukturiert.

## **4. Weitere Aktivitäten von Instant-Grid**

### **4.1 Zusammenarbeit D-Grid**

Instant-Grid hat als wesentliche Grid-Middleware und Software-Komponenten einen Ausschnitt dessen implementiert, was als der D-Grid-Software-Stack bezeichnet wird. Der Ausschnitt bezieht sich auf die Auswahl einer der drei zugrundeliegenden Grid-Middleware-Systeme des D-Grid, nämlich auf Globus. Aufbauend auf dieser Basis hat Instant-Grid alle wesentlichen Komponenten der darüberliegenden und flankierenden Software des D-Grid implementiert, und dies schon zu einer recht frühen Phase der D-Grid-Förderung. So konnte Instant-Grid bei Neu-Implementationen im D-Grid oft als zusätzliche Referenz verwendet werden.

Mitarbeiter des Instant-Grid-Teams haben an vielen der D-Grid-Workshops und D-Grid-Meetings teilgenommen, wo ihre Teilnahme begrüßt wurde und wo oft ein fruchtbarer wechselseitiger Austausch der Erfahrungen die jeweilige Arbeit positiv beeinflusst hat.

Die starke Zusammenarbeit des Instant-Grid-Teams mit dem Team im Medi-GRID-Modul Ressourcenfusion hat sowohl innerhalb der Göttinger Grid-Gruppe und bei den beteiligten Mitarbeitern von Fraunhofer-FIRST als auch in der gemeinsamen Arbeit beider Gruppen zu einem hohen Maß an Synergie geführt. Hier konnten die Möglichkeiten, die in den einfachen Testbedingungen durch Instant-Grid begründet sind, besonders effizient genutzt werden.

Speziell beim Aufbau des Workflow-Systems, das zunächst für das Medi-GRID-Projekt im D-Grid aufgesetzt werden sollte, haben die Erfahrungen der Implementation im Instant-Grid, die zuvor erfolgt waren, wesentlich die Stabilität und Nutzbarkeit verbessert. Darüberhinaus konnten notwendige

Entwicklungen im Bereich der Ressourcenbeschreibung im Instant-Grid-Projekt rechtzeitig vorangetrieben werden, um so im MediGRID genutzt werden zu können. Mittlerweile wird dieses Workflow-System in verschiedenen anderen Communities des D-Grid als Plattform diskutiert und könnte zu einer wichtigen Grundlage der D-Grid-Software werden.

Eine Kooperation mit dem C3-Grid bezüglich der Implementation von GridShib im Instant-Grid war während der GES 2007 ins Auge gefasst worden. Zu dieser Kooperation gab es schon erste Vorarbeiten und Aufwandsabschätzungen. Letztlich hat sich die Herausgabe einer brauchbaren Version von GridShib aber so lange herausgezögert und auch die Implementation so sehr von der Globus-Version 4.0.3 des Instant-Grid entfernt, dass das Projekt nicht weiter verfolgt werden konnte.

Die beiden in Göttingen angesiedelten D-Grid-Projekte, MediGRID und TextGrid, konnten die durch die Arbeit mit Instant-Grid gewonnenen Erfahrungen im Grid-Middleware-Bereich bereits zu Beginn ihrer Projektlaufzeit in vollem Umfang nutzen. Diese Zusammenarbeit mündete in eine Partnerschaft, genannt GoeGrid, die sich die Entwicklung und Unterstützung lokaler Grid-Aktivitäten zur Aufgabe gemacht hat. Eine erste gemeinsame Veranstaltung im Jahr 2006, das Göttinger Grid-Seminar, fand soviel Zuspruch, dass sie im letzten Jahr ebenso erfolgreich wiederholt wurde.

Die Partnerschaft wurde im letzten Jahr durch eine große Arbeitsgruppe im HEP-Grid verbreitert. Seit letztem Jahr veranstalten diese drei D-Grid-Gruppen gemeinsam mit Instant-Grid Lehrveranstaltungen zu Grid-Technologien in der Universität. Einem Seminar im Wintersemester 2007/08 folgte im Sommersemester 2008 ein Grid-Praktikum, das auf Instant-Grid als Plattform für die Arbeitsgruppen aufsetzt.

Gemeinsam mit der GWDG haben die drei D-Grid-Gruppen aufgrund der Erfahrungen der Zusammenarbeit beschlossen, den zusätzlichen Bedarf an Grid-Ressourcen zu bündeln und die bestehende Grid-Hardware mit den D-Grid-Sonderinvestitionen 2007 zum Grid-Ressourcenzentrum Göttingen im D-Grid ausgebaut.

#### **4.2 Die Instant-Grid Web-Präsenz**

Die wesentliche Komponente zur Kommunikation des Projektes nach außen wie auch für die internen Informationsflüsse war der Aufbau eines Web-Auftritts für das Instant-Grid (<http://instant-grid.de>). Dieser Web-Auftritt dient der Darstellung des Projektes in der Öffentlichkeit, als Frontend für den Download der Instant-Grid-CD-Versionen und als wesentliche Online-Dokumentationsplattform. Ein integriertes Forum dient dem Erfahrungsaus-

tausch unter den Nutzern des Instant-Grid, der Beantwortung häufig gestellter Fragen zur Nutzung des Instant-Grid und es können frühzeitig nicht vorausgesehene Schwachstellen des Toolkits oder des Installationsmechanismus verbreitet werden. Darüber hinaus hat sich das Forum als das wichtigste Instrument für die tägliche Arbeit der Entwickler innerhalb des Projektes, die ja über verschiedene Standorte innerhalb Deutschlands verteilt waren, herausgebildet. Die GWDG hat sich bereiterklärt, diese wichtige Kommunikationsplattform auch über die Projektlaufzeit hinaus aufrecht zu erhalten und damit zusammen mit dem Versionierungssystem die technischen Basis-komponenten für den Erhalt des Instant-Grid-Projektes zu sichern.

### **4.3 Versionierung**

Während der Anfangsphase des Projektes hat sich recht schnell herausgestellt, dass die technologische Herausforderung den Einsatz eines Versionskontrollsystem zwingend erforderlich macht. Der Einsatz eines solchen Systems war zwar nicht in der Beantragung vorausgesehen worden, der durchaus aufwendige Umstieg des Projektes auf das Versionierungssystem Subversion (SVN) Anfang Mai 2006 hat der Entwicklung von Instant-Grid aber wesentliche neue Möglichkeiten gegeben. Insbesondere konnte damit der Zyklus von Entwicklung, Test, Fehlerbehebung und Weiterentwicklung deutlich verkürzt werden.

### **4.4 Image-Reduktion**

Eine wesentliche Aufgabe innerhalb des Projektes war immer die Beschränkung der Software dahingehend, dass das gesamte System in den Grenzen einer CD blieb. Auch wenn in der Planung des Projektes bereits die Möglichkeit der Verwendung von DVDs in Betracht gezogen wurde, war es immer ein Anliegen des Projektes gewesen, die CD als Volumengrenze zu belassen. Dies hatte vor allem Gründe darin, dass die Einschätzung vorherrschte, dass ein größeres Volumen die Probleme beim Download und auch die Zurückhaltung der Nutzer erhöhen würde. Die hohe Zahl der heruntergeladenen CDs spricht dafür, dass sich diese Sichtweise ausgezahlt hat.

Um die Beschränkung auf die Volumengrenze einer CD aufrecht erhalten zu können, waren erhebliche Anstrengungen notwendig. Vor allem musste die Softwarebasis der jeweiligen zugrunde liegenden Knoppix-Version, die ab 2007 eine reine DVD-Version war, komplett unterbrochen werden auf die Bedürfnisse für die gewünschte Grid-Technologie. Dies war insofern schwierig, da die vorhandenen Pakete in vielerlei Abhängigkeiten zueinander stehen.

Darüberhinaus musste aber auch die zum Teil sehr redundante Grid-Software genau gesichtet werden. Zum einen konnte durch eine genaue Analyse des Globus-Paketes eine deutliche Reduktion des Umfangs erreicht werden. Zum anderen gab es Abhängigkeiten der verschiedenen Software-Komponenten und Anwendungen zu verschiedenen Java-Versionen. Durch eine genaue Analyse der Java-Abhängigkeiten der Instant-Grid-Software konnte aber die Voraussetzung geschaffen werden, die bislang zusätzlich verwendete ältere Java-Version zu entfernen. Damit konnte weiterer Platz für andere Komponenten auf der CD geschaffen werden.

#### **4.5 Qualitätskontrolle**

Von zentraler Bedeutung für den erfolgreichen Einsatz des Instant-Grid sind der einfache und störungsfreie Ablauf des Installationsvorgangs, die Stabilität des Grid-Middleware und die verständliche Nutzungsumgebung für die Anwendungen und Monitorfunktionen. Die Qualitätssicherung dieser Eigenschaften ist deshalb ein wesentlicher Bestandteil des Projektes. Während der Laufzeit des Projektes war für den Einsatz der Knoppix-Technologie, der Grid-Middleware und einzelner Anwendungen die Qualitätskontrolle in Zusammenarbeit mit kompetenten assoziierten Partnern außerordentlich erfolgreich.

#### **4.6 Die Instant-Grid Version 1.1**

Die Instant-Grid-Version, die für die GES 2007 erstellt worden war, war aufgrund der Fokussierung des Projektes auf die Förderung der deutschen Grid-Gtruktur von der Präsentation, wie auch von der Dokumentation, stark an das deutsche Publikum gerichtet gewesen. Es war ein Anliegen der Instant-Grid-Entwickler und natürlich auch der internationalen Nutzerschaft, diese Beschränkung aufzuheben und eine deutlich bessere Präsentation von Instant-Grid in englischer Sprache herauszugeben. Mit der im Februar 2008 herausgegebenen Version 1.1 von Instant-Grid, die damit lange nach Ende der Förderphase durch die Bereitschaft der GWDG ermöglicht wurde, den entsprechenden Personalaufwand dafür bereitzustellen, konnte dieses Anliegen, mit einigen wesentlichen weiteren Verbesserungen, umgesetzt werden.

##### **4.6.1 Knoppix 5.1.1**

Instant-Grid Version 1.1 basiert auf Knoppix 5.1.1 (Live-Linux-Umgebung), wodurch Instant-Grid nun auch mit neuerer Hardware gestartet werden kann. Als Grid Middleware wird weiterhin Globus Toolkit 4.0.3 genutzt. Die wichtigsten sonstigen Änderungen betreffen eine automatisierte Test-Umgebung, zusätzliche Unterstützung für das Cluster-Setup und größere Stabilität bei

der Verwendung von Workflows. Die Stabilität des Grid Workflow Managers GWES wurde verbessert.

#### **4.6.2 Automatische Tests**

Um einen einfachen Gesamttest von Instant-Grid durchführen zu können und um das Feedback über Probleme mit Instant-Grid zu vereinfachen, wurde ein automatisierter Test erstellt (ig-test.sh, aufzurufen im Home-Verzeichnis des Knoppix-Benutzers), mit dem die meisten grundlegenden Cluster- und Grid-Middleware-Komponenten geprüft werden können.

#### **4.6.3 Portreport**

Der Portreport ist vollständig umgeschrieben worden. Es gibt nun unterschiedliche Test-Zusammenstellungen für die Ports vom Server und von den Klienten. Der Portreport ist zudem besser in die Web-Oberfläche integriert.

#### **4.6.4 Externe Netzkonfiguration**

Ein externes Netz kann jetzt nach dem Start von Instant-Grid konfiguriert werden, z. B. falls dies beim Start nicht möglich war. Ein entsprechendes Tool ist in den Menü-Balken von Instant-Grid integriert. Um auch auf den Klienten das externe Netzwerk durchzureichen, muss hier noch pump neu aufgerufen werden (sudo pump).

#### **4.6.5 Internationalisierung**

Um die Benutzbarkeit von Instant-Grid für nicht-deutschsprachige Benutzer zu erhöhen, wurden die Default-Sprache des Boot-Menüs und das Boot-Tastaturlayout in Englisch geändert. Dies war ein Problem, das neue internationale Benutzer schon vor dem Start von Instant-Grid in Schwierigkeiten gebracht hat und daher besonders störend gewesen war. Die Hilfe-Bildschirme am Bootprompt werden nun beim Boot auch auf den Klienten angezeigt.

### **5. Verwertung**

#### **5.1 GES 2007**

Auf der Konferenz GES 2007 in Baden-Baden, Mai 2007, hat sich Instant-Grid in vielfältiger Weise präsentieren können.

Vor Beginn der Konferenz wurde ein Workshop zum Thema „Grid-Workflows with Instant-Grid“ angeboten. Dieser Workshop hatte etwa 15 Teil-

nehmer, die mit ihren Laptops in einem lokalen Instant-Grid an der Erstellung eigener Workflows gearbeitet haben.

Das Vortragspaper mit dem Titel „Instant-Grid – A Toolkit for Demonstration, Test and Development of Grid-Infrastructure“ für die Konferenz war angenommen worden. Damit konnte Instant-Grid auch im Konferenzprogramm selbst in Erscheinung treten.

Auf einem Stand war ständig eine Instant-Grid-Installation in Betrieb, Benutzer konnten sich mit eigenen Rechnern in das laufende Instant-Grid einhängen. Der Betrieb des Grids wurde mit einem Beamer auf einer Projektionsfläche dargestellt.

Es wurde ein themenzentriertes Pausenprogramm angeboten, mit dem in allen Pausen der Konferenz zu einer speziellen Komponente von Instant-Grid informiert wurde. Die Themen waren: Connectivity D-Grid / GWDC, Collaboration, GridSearch, POV-Ray und GridLabs.



Der Auftritt von Instant-Grid auf der Konferenz hat zu einer Fülle von Gesprächen und Kontakten geführt. Auf dem Bild oben informiert sich Ian Foster auf der GES 2007 bei den Instant-Grid-Entwicklern über den Stand des Projektes.

## 5.2 Veröffentlichungen und Präsentationen über Instant-Grid

Eine Liste der Publikationen ist in Kapitel 7 zu finden.

Das Paper [1] wurde auf der Konferenz ICNS2006 am 18. Juli 2006 vorgelesen und wurde im Tagungsband veröffentlicht.

[2] ist im Rahmen einer Sondernummer zu Grid-Aktivitäten in Deutschland der Zeitschrift „Praxis der Informationsverarbeitung und Kommunikation“ (PIK) veröffentlicht worden.

Im Rahmen eines bei der GWDC durchgeführten Masterabschlusses zu Ressourcen-Monitoring im Studiengang „Angewandte Informatik“ ist die Arbeit [3] entstanden.

Ein Paper zu *Grid-Demonstration und Schulung mit Instant-Grid* [4] zur 21. DFN-Arbeitstagung über Kommunikationsnetze wurde angenommen und am 31. Mai 2007 auf der Tagung vorgestellt.

Das Paper für die GES 2007 Konferenz [5] ist auf der Konferenz vorgetragen und unter der Nummer 47 ID: 316534.0 im Tagungsband veröffentlicht worden.

Zur Konferenz IEEE IM2007 in München vom 21.-25. Mai 2007 wurde Instant-Grid unter dem Titel *Building a Grid Infrastructure with Instant-Grid* [6] im Rahmen eines Vortrags vorgestellt und als Paper unter der Nummer 23964 im Tagungsband veröffentlicht. Dieses Paper wurde als bestes Paper in der Rubrik „Application Session“ prämiert.

Der Abschlussbericht zum Instant-Grid-Projekt [7] ist im März 2008 erstellt und an die Technische Informationsbibliothek in Hannover geschickt worden und mit der Kennung <http://edok01.tib.uni-hannover.de/edoks/e01fb08/577029800.pdf> versehen worden.

Die erhoffte Publikation von Instant-Grid in der Zeitschrift c't konnte allerdings nicht erreicht werden, da das Thema Grid insgesamt vorerst aus dem publizistischen Fokus dieser Zeitschrift geraten ist. Der erste Artikel über Globus-Installation und Nutzung sowie eine Skizze eines zweiten Artikels über Instant-Grid selbst ist der Redaktion im November 2006 vom Instant-Grid-Projekt übergeben worden.

Die Artikel sollten im Rahmen einer Reihe zur Programmierung paralleler Systeme erscheinen, die bereits im Spätsommer 2006 gestartet wurde. Zuletzt sind in dieser Reihe Artikel zu MPI 2 und zu Debugging erschienen. Die Artikel zu Cluster-Software-Produkten und zur Kernel-Thread-Programmierung wurden allerdings von der Redaktion bereits auf unbestimmte Zeit vertagt und damit die Reihe zur Programmierung paralleler Systeme vorerst abgesetzt, wie uns Herr Lau, der zuständige Redakteur bei der c't, Mitte Oktober 2007 mitteilte. Damit entfällt vorerst auch das Thema Grid-Computing, was die Serie abschließen sollte, und unter dem der allgemeine Artikel zu Globus-Techniken und der spezielle zu Instant-Grid erscheinen sollte. Der Versuch, den Artikel anderen Redaktionen, wie der iX und dem Linux-Magazin anzubieten, hat bislang ebenfalls keinen Erfolg gezeigt.

### **5.2.1 Präsentationen**

Neben den oben bereits genannten Präsentationen auf Tagungen, die jeweils mit einer Veröffentlichung verknüpft waren, gab es verschiedene Gelegenheiten für weitere Darstellungen des Instant-Grid-Projektes.

Auf der ISC2006 in Dresden wurden Instant-Grid-CDs der vorliegenden Version verteilt und Instant-Grid mit einem Flyer auf dem Stand der D-Grid-Initiative präsentiert.

Instant-Grid wurde auf dem Linux-Treffen der Max-Planck-Gesellschaft am 23.-24.3.2006 in Leipzig in Form einer Live-Demo präsentiert.

Eine ähnliche Präsentation wurde auf einem Treffen der Göttinger Grid-Initiativen (Gö-Grid) am 21.4.2006 wiederholt.

Im Vortrag „MOSILAB-on-Instant-Grid-Demonstrator“ auf dem Workshop „Grid-Technologie für den Entwurf technischer Systeme“ am 7.4.2006 wurde die Instant-Grid-Technologie als Grundlage zur Demonstration einer Anwendung im Grid verwendet.

Auf dem Linux-Tag 2006 in Wiesbaden, 3.-6-5.2006, wurde ein Vortrag mit dem Thema „Instant-Grid, ein Grid in 15 Minuten“ gehalten.

### **5.3 Eine prominente Stimme zu Instant-Grid**

Im Oktober vermerkt Ian Foster in seinem Web-Blog:

Instant Grid from Germany: I've been watching the Instant Grid web develop for a while. Unfortunately, my German isn't too good; fortunately, there is more and more content in English, and from what I can see, it seems really nice.

Instant Grid combines Knoppix Linux, Globus 4.0.2 as Grid middleware, GridSphere (for portals), and some other technology to build a Grid-enabled cluster system that you can boot from CD-ROM onto a collection of PCs. If anyone has experience using this, let me know.

October 30, 2006, Ian Foster (in [http://ianfoster.typepad.com/blog/2006/10/instant\\_grid\\_fr.html](http://ianfoster.typepad.com/blog/2006/10/instant_grid_fr.html))

### **5.4 Weiternutzung von Entwicklungen im Instant-Grid-Projekt**

Für die automatische Nutzung durch Grid-Workflows war die Umsetzung der Monitoring-Parameter in eine entsprechende Beschreibungssprache (hier die GResourceDL) und deren Speicherung in einer für die Workflows erreichbaren Datenbank, der Grid-Ressource-DB (GRDB), notwendig. Diese Arbeiten sowie ein Portlet zur Darstellung des Inhalts der GRDB wurden im Rahmen von Instant-Grid durchgeführt. Diese Software wird vermutlich in den Software-Stack des DGI der D-Grid-Initiative integriert und wurde in die Middleware-Komponenten von MediGRID als zentrale Res-

sourcen-Datenbank eingefügt. In Instant-Grid sind diese Monitoring-Komponenten über das Gridsphere-Portal dargestellt.

## **5.5 Neue Versionen**

Seitdem das Projekt im Herbst 2007 zu Ende gegangen ist, hat es eine weitere Version 1.1 von Instant-Grid gegeben. Es war ein Anliegen der Instant-Grid-Entwickler und natürlich auch der internationalen Nutzerschaft, eine bessere Internationalisierung von Instant-Grid zu implementieren. Mit der im Februar 2008 herausgegebenen Version 1.1 von Instant-Grid konnten dieses Anliegen, und einige weitere wesentliche Verbesserungen umgesetzt werden.

### **5.5.1 Neuerungen der Version 1.1**

Instant-Grid Version 1.1 basiert auf dem derzeitigen Knoppix 5.1.1 (Linux live CD) und unterstützt damit neuere Hardware. Als Middleware ist noch Globus Toolkit 4.0.3 in Benutzung.

Die wichtigsten Neuerungen sind eine bessere Internationalisierung von Instant-Grid, eine verbesserte Test-Umgebung von Instant-Grid, zusätzliche Unterstützung beim Cluster-Setup und eine höhere Stabilität bei der Workflow-Engine.

Im Einzelnen gibt es die folgenden Neuerungen:

- Der Grid-Workflow-Manager GWES wurde auf eine neuere, stabilere Version gebracht.
- Als einfachen übergreifenden Test von Instant-Grid und um besseres Feedback bei Fehlern im Instant-Grid zu bekommen, wurde ein Test implementiert, der alle wesentlichen Cluster und Grid-Komponenten testet und eine standardisierte Ausgabe liefert.
- Der Port-Report wurde vollständig überarbeitet. Es gibt jetzt eine eigene Port-Test-Konfiguration für die Klienten und den Server, und er ist besser in das Web-Interface integriert.
- Um ein externes Netzwerk auch nach dem Start noch aufsetzen zu können, zum Beispiel weil der Setup beim Bootvorgang gescheitert war, ist ein neues Tool in das Menu aufgenommen worden. Auf den Klienten bedarf es dafür allerdings im Nachhinein eines „sudo pump“.
- Um die Benutzbarkeit für nichtdeutsche Benutzer zu erhöhen, wurde die Default-Sprache im Bootmenu auf Englisch gesetzt.

- Die überwiegende Zahl der Dokumentationsartikel ist nun in Englisch verfügbar.
- Der Hilfe-Bildschirm beim Booten ist nun auch auf den Klienten zu sehen.

## 6. Nutzung der Instant-Grid-Software

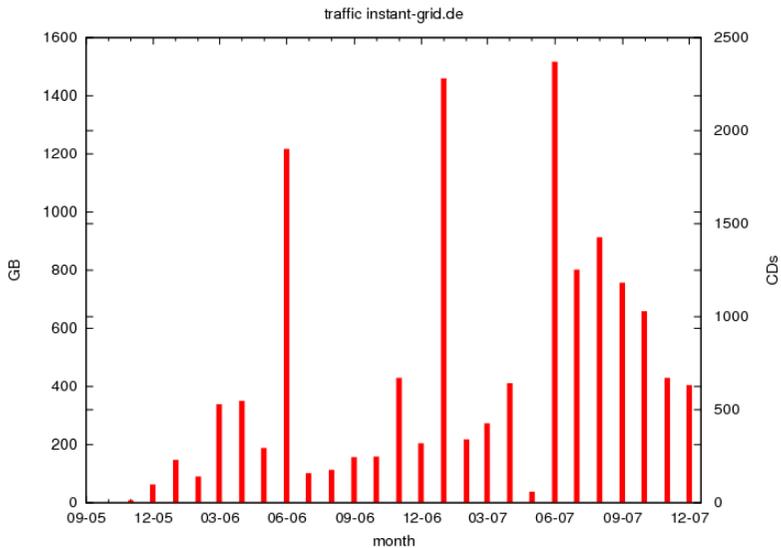
### 6.1 Datentransfer-Abschätzungen

Instant-Grid hat mit einem geschätzten Gesamtvolumen von fast **20.000 heruntergeladenen CDs** einer Vielzahl Wissenschaftlern, Studenten und anderen potentiellen Nutzern von e-Science die Möglichkeit gegeben, die ihm zur Verfügung stehenden vernetzten Rechnerressourcen ohne weitere Kenntnisse und ohne Konfigurationsaufwand zu einem lokalen Grid zu konfigurieren. In solchen „privaten“ Grids konnten dann mit den im Instant-Grid bereitgestellten Grid-Anwendungen Erfahrungen gesammelt werden und eigene Anwendungen für die Nutzung in einem Grid vorbereitet werden.



Die Art der jeweiligen Anwendungen wurden beim Download von den Nutzern auf freiwilliger Basis abgefragt, auf der Web-Seite des Projektes dokumentiert und finden sich nun im Anhang dieses Berichtes wieder. Eine Übersicht über die weltweite Verbreitung von Instant-Grid-CDs, die durch die Rückverfolgung der Download-Adressen erstellt werden konnte, befindet sich in der obigen Abbildung und wird aktualisiert auf der Homepage von Instant-Grid gezeigt.

Die Anzahl der übertragenen Bytes bzw. CDs seit Beginn des Projektes bis Ende 2007 stellt sich anhand der Log-Daten des Webservers in einer monatlichen Statistik wie folgt dar:



Diese Verkehrszahlen akkumulieren sich über den genannten Zeitraum zu 12,6 TB. Dabei sind allerdings nicht alle Datenvolumen auf Downloads zurückzuführen. Der Anteil des Transfers am Download betrug zum Beispiel im Dezember 2006 nach der folgenden Tabelle mehr als 99%, wobei sich knapp 1/10 der Anfragen auf das jeweilige „nightly-build“ (Verzeichnis /testing) bezog. Der Rest fiel auf die reguläre Version (Verzeichnis /files). In diesem Monat war eine neue Instant-Grid-Version 0.6 erschienen. Aber auch in anderen Monaten seit Herausgabe der ersten Instant-Grid-CD-Version hat das Gesamttransferaufkommen niemals 1 % überschritten.

Anzahl Anf.	Bytes (%)	Verzeichnis
771	91,11 %	/files/
114	8,34 %	/testing/
27564	0,18 %	[Root-Verzeichnis]
328	0,14 %	/flash_demos/
540	0,13 %	/vortraege/
15660	0,06 %	/site-pics/

Das Verhältnis in anderen Monaten ist manchmal anstelle der letzten Instant-Grid-Version eher zu Gunsten des aktuellen Build aus dem Versionskontrollsystem gegangen, wobei dies lediglich Ende 2005, Anfang 2006 mehr als 30 % betrug. Sonst war der Anteil unter 20 %. Die Verteilung der Transferolumina auf Netze und der Anfragen auf Organisationen ist beispielhaft für den Dezember 2006 aufgeführt.



Damit fallen insgesamt etwa 12,5 TB Transfervolumen für Instant-Grid-CDs an, was etwa 20.000 Instant-Grid-CDs entspricht, die hauptsächlich innerhalb Deutschlands und Europas, aber letztlich auch in alle Welt in diesem Zeitraum heruntergeladen wurden.

## 6.2 Nutzergruppen

Eine weitere Quelle zur Einschätzung der Nutzung von Instant-Grid sind die freiwilligen Mitteilungen, die uns von den Nutzern beim Download über eine spezielle Eingabemaske zugeschickt werden. Ein Ausschnitt aus diesen Rückmeldungen ist im Anhang dokumentiert. Hieraus geht hervor, dass etwa die Hälfte der Nutzungen im Bereich Grid- und HPC-Computing, oder allgemeiner im Bereich des wissenschaftlichen Rechnens, angesiedelt werden kann, während die andere Hälfte aus dem Umfeld des Raytracing, also im weitesten Sinne aus dem Medienbereich, stammt.

Diese letztere Nutzergruppe ist insofern besonders interessant, da sie primär keinerlei Interesse an Grid-Technologie anmeldet, aber die Instant-Grid-Technologie als eine besonders klar strukturierte Möglichkeit der Verwaltung von verteilten Aufgaben in einem Rechnernetz erkennt. Das heisst, diese Gruppe lernt auf diese Weise über die Anwendung Grid-Technologie nutzen und schätzen. Diese Gruppe kann damit als eine echte Stärkung der Wahrnehmung von Grid-Technologien in der Gesellschaft angesehen werden.

Die Interessen dieser Gruppe sind durch hohe Anforderungen an CPU-Leistung, einer leicht zu parallelisierenden und zu verteilenden Aufgabenstellung und einer gewissen Notwendigkeit zur Verbesserung der Arbeitsorganisation bei der Massenverarbeitung geprägt. Insofern decken sich hier diese Interessen, die weitgehend aus dem Bereich der privaten Nutzung kommen, mit denen der Nutzer vieler wissenschaftlicher und kommerzieller Grid-Communities.

Weitere derartige Gruppen im Medien- und Konsumerbereich sind vermutlich beim Image-, Video- und Audio-Processing zu erwarten. Hier wäre in Zukunft durch Bereitstellung einfacher Schnittstellen und Workflows ebenfalls Grid-Technologie über Instant-Grid sehr gut vermittelbar.

Eine interessante Untergruppe aus dem Bereich des wissenschaftlichen Rechnens ist die derjenigen, die Grid-Technologie in der Ausbildung vermitteln möchten. Etliche Beschreibungen der gewünschten Nutzung gehen in diese Richtung und äußern sich sehr positiv über die mit Instant-Grid gegebenen Möglichkeiten. Leider gibt es bislang keine Rückmeldungen über die Erfahrungen mit solchen Veranstaltungen. In Göttingen wurde im Sommersemester 2008 ein Grid-Praktikum auf Basis von Instant-Grid veranstaltet, bei dem die Studenten spezielle Aufgabenstellungen aus den Bereichen HEP-Grid, MediGRID und TextGrid mit Mitteln der Grid-Technologie lösen (siehe auch T. Rings, F. Viezens, J. Meyer, A. Aschenbrenner, Ein interdisziplinäres Grid-Anwenderpraktikum basierend auf Instant-Grid, in diesem GWDG-Bericht).

## **7. Publikationen**

[1] *Instant-Grid. Fully automated Middleware Deployment using a Live-CD*, Christian Boehme, Tim Ehlers, Jan Engelhardt, Andreas Félix, Oswald Haan, Tibor Kálmán, Bernhard Neumair, Ulrich Schwardmann, Dietmar Sommerfeld, Networking and Services, 2006. ICNS '06, ISBN: 0-7695-2622-5.

[2] *Instant-Grid: Demonstration, Entwicklung und Test von Grid-Anwendungen*, Christian Boehme, Andreas Félix, Bernhard Neumair, Ulrich Schwardmann, Praxis der Informationsverarbeitung und Kommunikation 3/06, ISSN 0930-5157, Juli - September 2006, pp 171-177.

[3] *Entwurf und Implementierung einer Ressourcen-Datenbank für das Instant-Grid-Projekt der GWDG*, Alexander Willner, Masterarbeit im Studiengang „Angewandte Informatik“ am Lehrstuhl für Praktische Informatik des Zentrums für Informatik an der Georg-August-Universität Göttingen, 27. Oktober 2006, pp. iv+92, ISSN 1612-6793, Nummer ZFI-BM-2006-36.

[4] *Grid-Demonstration und Schulung mit Instant-Grid*, Christian Boehme, 21. DFN-Arbeitstagung über Kommunikationsnetze, 30. Mai -1. April 2007.

[5] *Instant-Grid – A Toolkit for Demonstration, Test and Development of Grid-Infrastructure*, Boehme, Christian; Ehlers, Tim; Engelhardt, Jan; Félix, Andreas; Haan, Oswald; Kálmán, Tibor; Schwardmann, Ulrich; Sommerfeld, Dietmar; Willner, Alexander, GES 2007 conference, 2007, number 47, ID: 316534.02007.

[6] *Building a Grid Infrastructure with Instant-Grid*, Christian Boehme, Ulrich Schwardmann, (2007), Paper Number 23964 of the Application Session of the IEEE IM2007.

[7] *Instant-Grid: Abschlussbericht*, Ulrich Schwardmann, Göttingen: GWDG, 2008, Online-Ressource (19 S., 3,71 MB). : Ill., graph. Darst., Förderkennzeichen BMBF 01AK807, <http://edok01.tib.uni-hannover.de/edoks/e01fb08/577029800.pdf>

## Anhang zu Instant-Grid: Nutzer-Feedback

Die folgenden Rückmeldungen, die uns auf die Bitte einer kurzen Beschreibung der Aktivitäten, in denen Instant-Grid verwendet werden soll, zugesandt wurden, beschreiben eine Auswahl aus den vielfältigen Einsatzformen von Instant-Grid.

Ich habe von IG gehoert durch: grid website

I am a masters student doing research in grid project. I am interested in deploying instant-grid for the grid project in my school campus.

---

Ich habe von IG gehoert durch: google

I'm a beowulf cluster administrator and have been tasked with grid enabling our cluster. Instant-Grid sounded like an easy to use system to familiarise me with grid systems.

---

Ich habe von IG gehoert durch: Google

I am interested in home grid computing, in order to harness the computing power of my old PCs. It's a waste of power to leave old machines in the cupboard !

---

I have been interested in Beowulf cluster computers using linux, and several years ago, built a rudimentary cluster that used message passing between the nodes. I also studied a book called, building beowulf clusters with linux, by O'reilly.

I am interested in scalable computing to model biological, molecular, and chemical processes, among manifold other applications too numerous to name in the limited space offered by this message. I look forward to evaluating the "grid" (your terminology).

---

Ich habe von IG gehoert durch: I have no idea what the title of this field means

I'm building a render farm, and this looks like it might me the simplest solution. Any likelihood of automatic distributed rendering with "Blender" soon? :-)

Ich habe von IG gehoert durch: Arts and Humanities e-Science Support Centre ([www.ahessc.ac.uk](http://www.ahessc.ac.uk))

I'm running a research theme at the e-Science Institute in Edinburgh on Adoption and Sustainability of e-Research and Instant Grid seems like an excellent opportunity to increase uptake through providing demonstrators that people can take home with them.

---

Ich habe von IG gehoert durch: Internet

Wir moechten in der Fachhochschule Luzern verschiedene Grid systeme aufsetzen und diese miteinander vergleichen, welches wir im sinne einer Projektarbeiten dienen soll.

---

Ich habe von IG gehoert durch: Knoppix-Links auf knopper.net

Ich interessiere mich fuer Open-Source Software zum verteilten Rechnen auf einem lokalen Cluster, insbes. fuer moegliche Alternativen zu Open-Mosix, das leider momentan nur wenig weiterentwickelt wird.

---

Ich habe von IG gehoert durch: The Knoppix Wiki

I'd like to donwload Instant-Grid for testing purposes only, as I'm currently experimenting with ClusterKnoppix, Parallel Knoppix and the Bootable CD Cluster.

---

Ich habe von IG gehoert durch: A thesis advisor

I am perfoming my final work (grade thesis) in Computer Science. I am implementing a Grid environment at the Institute of Electrical Energy at the UNSJ (Argentina). Do you know similar software resources that could be implemented over Windows Clients, besides GNU/Linux ones?

---

Ich habe von IG gehoert durch: GridWorkflow searches on Google

I am quite excited with pulling down and starting to get familiarized with InstantGrid. I did pull down a previous release. However, I previously ran into a road block whereby the Knoppix instance came up with German. I am

hoping that this is easily rectified. Assuming I will run into the same thing, how could I change the config to start up with English?

Thanks....

---

Ich habe von IG gehoert durch: Freshmeat.net

I've always had an interest. I have multiple PC's at home. Thought Mosix was interesting. Stated looking into Globus a few years back. Played a little with pcm(sp?) quite a few years ago. Seems I don't have the time to really get into things and this looked like a really quick ramp up project.

---

Ich habe von IG gehoert durch: Google Treffer bei einer eurer Präsentationen

Ich wollte mir das Instant-Grid mal anschauen, da ich auf Arbeit nicht mehr auf den leistungsstarken PCs rechnen darf (Datenmengen zu groß und die Kisten werden anders gebraucht) und die Cluster-Warteschlangen sind mir zu lang um sinnvoll Daten auszuwerten. Im Moment verteile ich das ganze manuell auf mehrere P3 800 und vielleicht lässt sich da Abhilfe schaffen, wenn ich daraus ein kleines "lokales" Grid mache. Freundlichst Oliver Lischtschenko

---

Ich habe von IG gehoert durch: povray website

I am the primary mover behind the veggie oil powered junk cluster at the 2006 make faire. I've been building cluster of various types for about the last 10 years and would like to try your stuff. I can provide a "bottom of the barrel" (slow machines in huge numbers) test enviroment as I have access to hundreds if not thousands of pc's at any given time, mostly p3's and p4's To check it out google "james burgett" or "accrc"

---

Ich habe von IG gehoert durch: Lecturer at uni

I did a subject about Grid computing and they introduced us to your site. I've used the older version of IG to learn more about the Globus Toolkit and I'd like to see what the new version of IG will do.

Ich habe von IG gehoert durch: povray.org

This type of distribution is essentially what we have been looking for in a long time. We originally used ROCKS Cluster to setup the cluster system and then implement (build from source) the POVRay application, but it looks like you have done this with a much smaller linux footprint. I cannot wait to start testing/using this system. Tanx!

---

Ich habe von IG gehoert durch: tec channel compact

Sehr geehrte Damen und Herren,

ich bin Netzwerkadmin für Win Sun server. Zur Zeit habe ich ein Projekt was die möglist effektive Vernetzung von Servern und Services zur Folge haben soll. Bei den Überlegungen, welche Arte der Vernetzung die Beste ist, bin ich auf die Grid-Technologie gestoßen. Diese Form der Vernetzung scheint mir eine der besten Ideen zu sein, Services zur Verfügung zu stellen. Aus diesem Grund möchte ich mehr über die Abläufe von Grids erfahren und eventuell, falls mir das erlaubt ist, an der Weiterentwicklung dieses Systems teilnehmen und teilhaben.

---

Ich habe von IG gehoert durch: The PovRay website

I like to tinker. Due to my job, building and repairing pcs I have a number of slightly older pcs, I also enjoy using PovRay and other graphics programs as a hobby. I'd like to put the pcs to use as a rendering cluster, to learn a little about clustering and possibly speed the rendering process. I'm not a programmer just a dabbler, so for me it's the curiosity/coolness factor of trying it. Thanks for being there,

---

Ich habe von IG gehoert durch: Knoppix website

This may be just what I'm looking for. I'm keenly interested in this distribution and its possible application for use to create a "proof-of-concept" initial grid application. I am mostly interested in running loosely-coupled grid apps for Monte Carlo simulations, statistical tests using permutations of regression residuals, and stability testing of spatial/temporal models. After that, we want to do some more interesting things.

Ich habe von IG gehoert durch: knoppix website

We are running a research oriented cluster that over time grew in size. I would like to explore how the globus toolkit can improve system administration work.

---

Ich habe von IG gehoert durch: google search

Just looking for various grid/cluster applications, demonstrations and tools to experiment with, not for any official projects.. yet. Nice looking toolkit, I will try it out soon. -TM

---

Ich habe von IG gehoert durch: google

Hört sich sehr interessant an, wollte es mal antesten endlich ein sinnvoller Nutzen der Praktikumsrechner außerhalb der Semester bzw. Praktikumszeiten Besten Dank

---

Ich habe von IG gehoert durch: <http://journal.mycom.co.jp/news/2007/06/07/008/index.html>

I am weak in English, but am interested in a grid. I send a message and do not understand whether licensing is granted, but ask because I want to use it.

---

Ich habe von IG gehoert durch: Google

Instant-Grid is important for me because I'm teaching a distributing system class. So studying actual grids will give my students a better approach to what they really are.

---

Ich habe von IG gehoert durch: POV-Ray Homepage

I've been looking for a simple solution that allows distributed rendering for a long time. Previously I was interested in Dyne::Bolic but this distribution looks like it has more momentum!

---

Ich habe von IG gehoert durch: eigene Recherche im Zusammenhang mit OGSA

Liebes Instant-Grid-Team, ich bin im Rahmen des GDI-Grid-Projektes an der Uni Bonn tätig und mit der Gridifizierung von OGC Diensten (OWS) betraut. Von Instant-Grid erhoffe ich mir einen ersten Einblick in die Arbeitsweise eine Grid. Gruß und Dank.

---

I would like to run distributed computations of multiple neural networks for finding bounded numerical solutions to differential equations of any dimensionality/order...using java and openMosix (was using python and MPI but switched because of the better java libraries)

---

Ich habe von IG gehoert durch: EGEE Grid

There is a need to setup a GRID-based environment that can be used for perhaps a variable amount of time and morphed into another environment or information system. An example is a Technical Conference where the Instant Grid can be functional before, during and after the Conference, ultimately being replaced by some other vehicle but not destroyed.

This also raises the possibility of a resurrection to serve future requirements. An example would be an Instant-Grid to serve Public Health Authorities in times of need.

My impression is that the Instant-Grid would be a good candidate for this development. I would like to make contact with others with similar ideas.

---

Ich interessiere mich dafür, weil ich ein Grid-Computing konstruieren muss um zwei Cluster von Faultät Hochschule Esslingen zu verbinden. Ich forsche gerade nach Globus Toolkit, weil ich eine Vermittlung von Diensten zwischen Globus Toolkit und Povray machen muss, aber ich habe gerade Probleme mit Installation von Zertifikate. Ich möchte wissen, ob es möglich ist diese Aufgabe mit Instand-Grid zu machen. Wenn Sie in meiner Arbeit interessiert ist, könnten Sie mir schreiben damit ich Sie mehr Erklärung davon geben würde.

---

Ich habe von IG gehoert durch: [http://www.knoppix.net/wiki/Cluster\\_Live\\_CD](http://www.knoppix.net/wiki/Cluster_Live_CD)

I'm a graduated student of physics at University of Pisa, Italy, and I'm very interested in numerical parallel programming and grid computing. Many Thanks

---

Ich habe von IG gehoert durch: [news.povray.org](http://news.povray.org)

I'm an instructor at Navajo Technical College (a tribal community college) in Crownpoint, New Mexico, USA. I am interested in integrating your work into my curriculum development at the college. I am new to grid computing and have been looking at ways to start via cgi rendering. Your work will assist in my introduction to the grid and putting together a linux render farm on the hardware I have available. The hardware includes an IBM blade center with 4 blades and a SAN with 4.8 TB of storage. We hope to expand our blade center in the near future.

Thank you for making this work available. I will certainly do my best to put it through its paces as I learn more about grid computing.

---

Ich habe von IG gehoert durch: [google](http://google)

Well I am a student who tend to be a expert in this area (parallel processing, clusters, GRIDs etc.) - soon, I will start to do my final diploma homework so, I don't have to tell you nothing more. I am very interested in this area.

---

Ich habe von IG gehoert durch: [Google](http://Google)

Ich möchte genau das machen, was Instant-Grid bietet: Ein Tool für die Demonstration von Grid-Computing über MPI, am liebsten mit Open-FOAM (einem CFD Programm). Ich bin schon sehr gespannt! Ich würde mich über weitere Entwicklungen in dieser Richtung freuen!

---

Ich habe von IG gehoert durch: [google search](http://google search)

Ich haette interesse an einer einfachen installierungsmoeglichkeit fuer Globus Toolkit bzw. VDT, um mein eigenes Grid aufzustellen (mit eigenem schlusselmanagement) anstatt mich an ein existierendes Grid anzuschleisen, was anscheinend bei den meisten howtos und installation documents angenommen wird. Instant-Grid scheint vielleicht das anzubieten.

Ich habe von IG gehoert durch: google search

Have many older PC's that are not very useful in todays environment. Would like to experiment with them in a cluster or grid situation to see what can be achieved.

---

Ich habe von IG gehoert durch: google, knoppix

I am very interested in cluster / grid serverinf for older computers in 3th world countries or economies. Poorer Americans / Start up companies / churches and NGO's

---

Ich bin sehr an der Thematik Clustering interessiert. Bisher habe ich mich mit Hochverfügbarkeitslösungen beschäftigt (u.a. Novell Netware Cluster 1.6). Da ich derzeit in der glücklichen Lage bin, vier baugleiche Systeme ohne dedizierte Anwendung mein Eigen zu nennen, möchte ich anhand Ihrer "Live-CD" das "andere Clustering" kennenlernen. Es ist aber hauptlich privates Interesse, da ich es vermutlich in meinem Job als Admin in einem Krankenhaus nicht einsetzen kann. Danke!

---

Hallo! Ich schreibe an einer Arbeit über Servervirtualisierung und möchte letztlich auf einem Hardwarepool (Servercluster) mehrere davon logisch abstrahierte Virtuelle Server laufen lassen. Daher interessieren mich einerseits Projekte wie dieses und andererseits VMWare, XEN u. a.

---

I'm trying to build a budget render farm, using old laptops glued into 19" racking shelves. I need to find an OS I can boot from a USB drive, and this is a possibility. I just want to download it to experiment with transferring it from a 650MB CD to a 16MB USB stick at the moment, dumping all the components that aren't absolutely necessary... :-)

---

Ich find's eine tolle Sache, eine so komplexe Middleware wie GTK 4 "zum -hoffentlich ;-)-sofortigen Verzehr" bereit zu stellen, das würde ich gerne selbst 'mal ausprobieren. In den letzten Jahren meiner Berufstätigkeit (jetzt nicht mehr aktiv) hatte ich Gelegenheit, schon mit den früheren Versionen von GTK (2.x) selbst konkrete Erfahrungen in kleineren Grid-Projekten zu sammeln.

Hallo, ich habe hier eine Farm von Sun v40z Rechnern, die ich gerne mal mit der Grid software ausprobieren möchte. Unsere Anwendungen im Clustermfeld sind im Bereich Bioinformatik. Auf dem Linuxtag habe ich den Vortrag von Andreas Felix gehört und ich bin gespannt wie gut das Knoppix Basierte Demo-Grid funktioniert.

---

I'm the IT guy at a library in Westerville Ohio USA. I'm trying to help a student with a large rendering project for school. So far I haven't had much luck and would like to try your cd.

---

Ich habe von IG gehoert durch: [link von Knoppix website](#)

Instant-Grid is interesting for me to be able to demonstrate to those involved in hard/software development at our University, what this technology could do for us.

---

Ich habe von IG gehoert durch: Browsing

Hallo, es gibt eine offizielle Grid-Seite von mir, die noch auf dessen Finanzierung wartet. Da sollte sich dann auch einiges an Überlappungen mit dem InstantGrid ergeben. Die mehr private Seite ist hier <http://wiki.debian.org/pkg-escience>. Gratuliere erstmal! Meine erste Evaluation schlug wegen fehlenden Speichers (hatte nur 256 MB) fehl. Es wäre hilfreich, wenn diesbezüglich mehr Abfragen getätigt würden.

---

Ich habe von IG gehoert durch: <http://news.povray.org/povray.news-submissions/thread/%3Cweb.44214c8f83d9a21c9a963ce10@news.povray.org%3E/>

Need a quick easy render farm. your product looks great. Is it in english too, or just german?

---

Ich habe von IG gehoert durch: [freshmeat](#)

Hi, I manage a couple of beowulf-style compute clusters and some stand-alone SMP systems for faculty researchers here at UNT. We have watched the development of grids, but to this point have not found any compelling reason to deploy one (or join one). I would like to "prototype" something

that would give us more experience with the environment, and perhaps argue the advantages of grid computing in a more convincing manner.

---

I am curious about all types of linux, particularly the 'live' distributions and ones that allow easy creation of clusters... but I download many types and share them with others

---

Ich habe von IG gehoert durch: povray website

I,ve been involved in clusters for years. recently made a vwo powered cluster at makefaire. I would like to check your solution out.

---

Ich habe von IG gehoert durch: Google

Ich bin immer auf der Suche nach einfachen Tools zum Erstellen und Verwalten von Linux-Clustern, da sie einen einfachen Weg zur Ablage und Analyse grosser Mengen an Medizinischen MRT-Daten bieten. So mal sehen . . . Jedenfalls schon mal vielen Dank für die Mühe und dic Bereitschaft, alles frei ins Netz zu stellen

---

Ich habe von IG gehoert durch: web surfing

Just testing cluster software to learn more about them. I used to be a developer for Digital Equipment Corp running VMS Clusters...

---

Ich habe von IG gehoert durch: proactive tinkerer

Hello, I would like to evaluate instant grid. I am doing graph processing systems, and recently I had to make a multithreaded graph writer. I noticed, it was very diffucult to break up the problem for each thread so it is divided. What can instant-grid compute for graph related algorithms? -Best Regards

---

Ich habe von IG gehoert durch: Internet

Ich hatte selber die Idee zu einer GTK4-Live-Cd für Test-Grids und war erfreut zu sehen, dass es bereits ein Projekt gibt.

Ich habe von IG gehoert durch: I am project partner

Nun möchte ich doch endlich die neue GWES-Version mit der aktuellen Instant-Grid dist testen.

Gruß nach Göttingen,

---

Ich habe von IG gehoert durch: google

We are trying to make a 32 node cluster. I heard about your live cd on google. It says that it allows pxe boot on several machines. this would save us so much time. thanks

---

Ich habe von IG gehoert durch: Andreas Hoheisel

Greetings,

I was hoping to download Instant-Grid to gain insight into the possible application of such concepts in my environment. I belong to an organization that provides financial Web Services to the banking industry in Canada. We currently do not utilize much in the way of Grid, but it look promising. The work and Andreas is doing around Grida and Workflow is especially interesting.

Thx....

---

Ich habe von IG gehoert durch: Suche im Internet

Mich interessieren Cluster und automatische Last-/Job-Verteilung sowie virtuelle Maschinen.

Unabhaengig vom derzeitigen Boom wird das die Zukunft sein. Ich will mitmachen!

---

Ich habe von IG gehoert durch: google

Hi, I'm researching a PhD into Grid based distributed digital forensic analysis. I will probably use Instant-Grid to build a prototype/test system.

I'll send you any results as they happen.

Ich habe von IG gehoert durch: Andreas Hoheisel

Looking at some of the work being done by Andreas Hoheisel on grid workflows, it was suggested that I use Instant Grid as a base.

---

Ich habe von IG gehoert durch: Searching solutions for parallel processing

A partner and I are interested in parallel and grid computing. Right now we are developing a scholar project that maybe the foundation for a future career.

---

Ich habe von IG gehoert durch: selbst entdeckt nach Daten-Verifikation für MNFT

Mich interessiert, wie man in so einem Grid arbeiten kann - resp. für Physik- und Meteorologie-Simulationen.

---

Ich habe von IG gehoert durch: Freshmeat

Just the usual keeping up-to-date on distributed computing alternatives. I'm a biology student and might have some use for this kind of environment.

---

Ich habe von IG gehoert durch: In knoppix website

I want to try GRID applications first and step through the installation process after beeing a litle more familiar with the concept and the tools. I expect to take a faster experience with this pre-package solution, and then move on, more confident, to other decisions.

---

Ich habe von IG gehoert durch: Artikel in PIK

Einsatz als Demo in meiner Lehrveranstaltung "Kommunikationsnetze", in der ich u. a. auch Grid-Computing als Anwendung in HS-Netzen behandle.

---

Ich habe von IG gehoert durch: [www.google.com](http://www.google.com)

Hallo, schreibe eine Semesterabreint über GRID - Computing und möchte daher ein Versuchsnetzwerk aufbauen.

Ich habe von IG gehoert durch: Google

Hallo liebes Instant-Grid Team,

ich möchte gerne ihre Software testen, da ich eine Diplomarbeit über Sicherheit im Grid Computing schreibe.

---

Ich habe von IG gehoert durch: Freshmeat.net

I like the idea of grids, but lack the knowhow to build my own. Your IG software offers me a way to play with grid technology, easy, and quickly. Thank You IG Community, and Merry Christmass.

---

Ich habe von IG gehoert durch: Internet Search

Happy New Year. I am very interested in see nodes which can be instantiated quickly in order to participate in a wider grid. Your work looks very interesting and hopefully you will continue to be successful.

---

Ich habe von IG gehoert durch: [www.krugle.com](http://www.krugle.com)

Really excited to test Instant Grid. I am a student working on a grid project and was really hoping for some easy way to install GT.

---

Ich habe von IG gehoert durch: Google search

I would like to use the instant grid as a teaching tool in the tutorials to demo some of the basic features of the grid and grid enabled applications.



## **Autorenverzeichnis**

*Aschenbrenner, Andreas*

Staats- und Universitätsbibliothek, Georg-August-Universität Göttingen

E-Mail: aschenbrenner@sub.uni-goettingen.de

*Ay, Cano*

II. Physikalisches Institut, Georg-August-Universität Göttingen

E-Mail: Cano.Ay@phys.uni-goettingen.de

*Boehme, Christian*

Gesellschaft für wissenschaftliche Datenverarbeitung mbH Göttingen

E-Mail: cboehme1@gwdg.de

*Haan, Oswald*

Gesellschaft für wissenschaftliche Datenverarbeitung mbH Göttingen

E-Mail: ohaan@gwdg.de

*Lohrberg, Kai*

Abteilung Medizinische Informatik, Universitätsmedizin Göttingen, Georg-August-Universität

E-Mail: lorberg@med.uni-goettingen.de

*Meyer, Jörg*

II. Physikalisches Institut, Georg-August-Universität Göttingen

E-Mail: joerg.meyer@phys.uni-goettingen.de

*Quadt, Arnulf*

II. Physikalisches Institut, Georg-August-Universität Göttingen

E-Mail: aquadt@uni-goettingen.de

*Richter, Harald*

Institut für Informatik, Technische Universität Clausthal

E-Mail: hri@tu-clausthal.de

*Rings, Thomas*

Institut für Informatik, Georg-August-Universität Göttingen

E-Mail: rings@cs.uni-goettingen.de

*Sax, Ulrich*

Abteilung Medizinische Informatik, Universitätsmedizin Göttingen, Georg-August-Universität

E-Mail: usax@med.uni-goettingen.de

*Schwardmann, Ulrich*

Gesellschaft für wissenschaftliche Datenverarbeitung mbH Göttingen

E-Mail: uschwar1@gwdg.de

*Sommerfeld, Dietmar*

Gesellschaft für wissenschaftliche Datenverarbeitung mbH Göttingen

E-Mail: dsommer@gwdg.de

*Viezens, Fred*

Institut für Biometrie und Medizinische Informatik, Otto-von-Guericke-Universität Magdeburg

E-Mail: fred.viezens@med.ovgu.de

## **In der Reihe GWDG-Berichte sind zuletzt erschienen:**

Nähere Informationen finden Sie im Internet unter  
[http://www.gwdg.de/forschung/publikationen/  
gwdg-berichte](http://www.gwdg.de/forschung/publikationen/gwdg-berichte)

- Nr. 40** *Plessner, Theo und Peter Wittenburg* (Hrsg.):  
**Forschung und wissenschaftliches Rechnen - Beiträge zum  
Heinz-Billing-Preis 1994**  
1995
- Nr. 41** *Brinkmeier, Fritz* (Hrsg.):  
**Rechner, Netze, Spezialisten. Vom Maschinenzentrum zum  
Kompetenzzentrum - Vorträge des Kolloquiums zum  
25jährigen Bestehen der GWDG**  
1996
- Nr. 42** *Plessner, Theo und Peter Wittenburg* (Hrsg.):  
**Forschung und wissenschaftliches Rechnen - Beiträge zum  
Heinz-Billing-Preis 1995**  
1996
- Nr. 43** *Wall, Dieter* (Hrsg.):  
**Kostenrechnung im wissenschaftlichen Rechenzentrum - Das  
Göttinger Modell**  
1996
- Nr. 44** *Plessner, Theo und Peter Wittenburg* (Hrsg.):  
**Forschung und wissenschaftliches Rechnen - Beiträge zum  
Heinz-Billing-Preis 1996**  
1997
- Nr. 45** *Koke, Hartmut und Engelbert Ziegler* (Hrsg.):  
**13. DV-Treffen der Max-Planck-Institute - 21.-22. November  
1996 in Göttingen**  
1997
- Nr. 46** **Jahresberichte 1994 bis 1996**  
1997
- Nr. 47** *Heuer, Konrad, Eberhard Mönkeberg und Ulrich Schwardmann*:  
**Server-Betrieb mit Standard-PC-Hardware unter freien UNIX-  
Betriebssystemen**  
1998

- Nr. 48** *Haan, Oswald* (Hrsg.):  
**Göttinger Informatik Kolloquium - Vorträge aus den Jahren 1996/97**  
1998
- Nr. 49** *Koke, Hartmut und Engelbert Ziegler* (Hrsg.):  
**IT-Infrastruktur im wissenschaftlichen Umfeld - 14. DV-Treffen der Max-Planck-Institute, 20. - 21. November 1997 in Göttingen**  
1998
- Nr. 50** *Gerling, Rainer W.* (Hrsg.):  
**Datenschutz und neue Medien - Datenschutzbildung am 25./26. Mai 1998**  
1998
- Nr. 51** *Plesser, Theo und Peter Wittenburg* (Hrsg.):  
**Forschung und wissenschaftliches Rechnen - Beiträge zum Heinz-Billing-Preis 1997**  
1998
- Nr. 52** *Heinzel, Stefan und Theo Plesser* (Hrsg.):  
**Forschung und wissenschaftliches Rechnen - Beiträge zum Heinz-Billing-Preis 1998**  
1999
- Nr. 53** *Kaspar, Friedbert und Hans-Ulrich Zimmermann* (Hrsg.):  
**Internet- und Intranet-Technologien in der wissenschaftlichen Datenverarbeitung - 15. DV-Treffen der Max-Planck-Institute, 18. - 20. November 1998 in Göttingen**  
1999
- Nr. 54** *Plesser, Theo und Helmut Hayd* (Hrsg.):  
**Forschung und wissenschaftliches Rechnen - Beiträge zum Heinz-Billing-Preis 1999**  
2000
- Nr. 55** *Kaspar, Friedbert und Hans-Ulrich Zimmermann* (Hrsg.):  
**Neue Technologien zur Nutzung von Netzdiensten - 16. DV-Treffen der Max-Planck-Institute, 17. - 19. November 1999 in Göttingen**  
2000

- Nr. 56** *Plessner, Theo und Helmut Hayd* (Hrsg.):  
**Forschung und wissenschaftliches Rechnen - Beiträge zum Heinz-Billing-Preis 2000**  
2001
- Nr. 57** *Hayd, Helmut und Rainer Kleinrensing* (Hrsg.):  
**17. und 18. DV-Treffen der Max-Planck-Institute**  
**22. - 24. November 2000 in Göttingen**  
**21. - 23. November 2001 in Göttingen**  
2002
- Nr. 58** *Plessner, Theo und Volker Macho* (Hrsg.):  
**Forschung und wissenschaftliches Rechnen - Beiträge zum Heinz-Billing-Preis 2001**  
2003
- Nr. 59** *Suchodoletz, Dirk* von:  
**Effizienter Betrieb großer Rechnerpools - Implementierung am Beispiel des Studierendennetzes an der Universität Göttingen**  
2003
- Nr. 60** *Haan, Oswald* (Hrsg.):  
**Erfahrungen mit den IBM-Parallelrechnersystemen RS/6000 SP und pSeries690**  
2003
- Nr. 61** *Rieger, Sebastian*:  
**Streaming-Media und Multicasting in drahtlosen Netzwerken - Untersuchung von Realisierungs- und Anwendungsmöglichkeiten**  
2003
- Nr. 62** *Kremer, Kurt und Volker Macho* (Hrsg.):  
**Forschung und wissenschaftliches Rechnen - Beiträge zum Heinz-Billing-Preis 2002**  
2003
- Nr. 63** *Kremer, Kurt und Volker Macho* (Hrsg.):  
**Forschung und wissenschaftliches Rechnen - Beiträge zum Heinz-Billing-Preis 2003**  
2004

- Nr. 64 *Koke, Hartmut* (Hrsg.):  
**GÖ\* – Integriertes Informationsmanagement im heterogenen eScience-Umfeld: GÖ\*-Vorantrag für die DFG-Förderinitiative „Leistungszentren für Forschungsinformation“**  
2004
- Nr. 65 *Koke, Hartmut* (Hrsg.):  
**GÖ\* – Integriertes Informationsmanagement im heterogenen eScience-Umfeld: GÖ\*-Hauptantrag für die DFG-Förderinitiative „Leistungszentren für Forschungsinformation“**  
2004
- Nr. 66 *Bussmann, Dietmar und Andreas Oberreuter* (Hrsg.):  
**19. und 20. DV-Treffen der Max-Planck-Institute**  
**20. - 22. November 2002 in Göttingen**  
**19. - 21. November 2003 in Göttingen**  
2004
- Nr. 67 *Gartmann, Christoph und Jochen Jähnke* (Hrsg.):  
**21. DV-Treffen der Max-Planck-Institute**  
**17. - 19. November 2004 in Göttingen**  
2005
- Nr. 68 *Kremer, Kurt und Volker Macho* (Hrsg.):  
**Forschung und wissenschaftliches Rechnen - Beiträge zum Heinz-Billing-Preis 2004**  
2005
- Nr. 69 *Kremer, Kurt und Volker Macho* (Hrsg.):  
**Forschung und wissenschaftliches Rechnen - Beiträge zum Heinz-Billing-Preis 2005**  
2006
- Nr. 70 *Gartmann, Christoph und Jochen Jähnke* (Hrsg.):  
**22. DV-Treffen der Max-Planck-Institute**  
**16. - 18. November 2005 in Göttingen**  
2006
- Nr. 71 *Hermann, Klaus und Jörg Kantel* (Hrsg.):  
**23. DV-Treffen der Max-Planck-Institute**  
**15. - 17. November 2006 in Berlin**  
2007

- Nr. 72** *Kremer, Kurt und Volker Macho* (Hrsg.):  
**Forschung und wissenschaftliches Rechnen - Beiträge zum Heinz-Billing-Preis 2006**  
2007
- Nr. 73** *Baumann, Thomas, Dieter Ruder und Bertram Smolny* (Hrsg.):  
**24. DV-Treffen der Max-Planck-Institute**  
**6. - 8. November 2007 in Jena**  
2008
- Nr. 74** *Schwardmann, Ulrich* (Hrsg.):  
**Grid-Technologie in Göttingen - Beiträge zum Grid-Ressourcen-Zentrum GoeGrid**  
2009

