

UNIVERSITÄT STUTTGART

Institut für Strömungsmechanik und
Hydraulische Strömungsmaschinen
Prof. Dr.-Ing. E. Göde

DELIVERABLE 2.4.1

ÜBER DIE IMPLEMENTIERUNG UND BENUTZUNG
AUSGEWÄHLTER INGENIEURSWERKZEUGE IN EINE
GRID-UMGEBUNG

August/2006

Dipl.-Ing. Ivana Buntić Ogor
Dipl.-Ing. Felix Lippold

Inhaltsverzeichnis

1	Einleitung	1
2	Design und Strömungssimulation auf Höchstleistungsrechnern	2
2.1	Spezielle Preprocessing Werkzeuge in COVISE	2
2.2	Grundlegender Lösungsalgorithmus von FENFLOSS	5
2.3	Parallelisierung mit MPI	5
3	Grid-Middleware	7
3.1	Globus Toolkit 4	7
3.2	UNICORE	7
3.3	gLite	8
3.4	Condor	8
3.5	Auswahl der Middleware	8
4	Konzepte für Entwurf und Berechnung in einer Grid-Umgebung	9
4.1	Anbindung der Entwurfswerkzeuge	9
4.2	Stand der Technik für Simulationen im Grid	10
4.3	Grid-Anbindung von FENFLOSS	11
5	Schlußfolgerungen und Möglichkeiten	13
	Literaturverzeichnis	14

1 Einleitung

Das Rechnen auf Hochleistungsrechnernetzen (Grid) bietet seit einiger Zeit einen vielversprechenden Ansatz, Forschung und Industrie große Rechenleistungen zur Verfügung zu stellen. Dieser Trend begründet sind nicht zuletzt in der Entwicklung der Rechenleistungen und Netzwerkverbindungen sowie der Entwicklung sogenannter Middleware. Außerdem steigt bei den verschiedenen Zielgruppen das Bewußtsein für die Vorteile und Möglichkeiten, die Grid-Computing bietet.

Insbesondere Optimierung und Entwurf im Ingenieurbereich schließen oft rechenintensive Aufgaben ein. Üblicherweise beinhalten diese, die Identifizierung von Entwurfsparameter oder deren Optimierung. Im Fall von Strömungsmaschinen erfordert dies viele Auswertungen der zu optimierenden Funktionen, die jeweils einer numerischen Strömungssimulation (Computational Fluid Dynamics - CFD) mit hohen Anforderungen an Speicherplatz und Prozessorleistung entsprechen.

Die Nachfrage nach den entsprechenden Ressourcen kann durch Zusammenführen verschiedener Hochleistungsrechner zu Rechnernetzen gedeckt werden. Grid-Computing soll also eine bessere Infrastruktur zur schnelleren Lösung von berechnungsintensiven Problemen bieten. Die entstehenden heterogenen Architekturen müssen in einer sicheren aber transparenten und komfortablen Weise für den Benutzer verfügbar sein. Hierzu stehen mittlerweile verschiedene Middlewares wie Globus Toolkit, Condor, gLite oder UNICORE zur Verfügung. Eine schnelle und sichere Datenübertragung in Netzwerken soll dabei gewährleistet werden.

In diesem Bericht wird zunächst näher auf spezielle Werkzeuge zum Entwurf von hydraulischen Maschinen sowie der Strömungssimulation eingegangen. Nach der Auswahl einer Middleware werden die Möglichkeiten zur Verwendung der vorhandenen Software als Grid-Services anhand bisher verfügbarer Ergebnisse diskutiert.

2 Design und Strömungssimulation auf Höchstleistungsrechnern

Geometrieentwurf und Gittergenerierung als Preprocessingschritte sind eng mit der Strömungssimulation gekoppelt. Im Softwarepaket COVISE ist sowohl ein Geometrie- als auch ein Gittergenerator für Anwendungen aus dem Bereich hydraulischer Strömungsmaschinen integriert. Zur Berechnung der Strömung steht das Strömungssimulationsprogramm FENFLOSS als zusätzliches Modul zur Verfügung.

Die Simulation von Strömungen zählt heute zu einem wichtigen Mittel bei der Beurteilung, Entwicklung und Analyse von Maschinen und Anlagen. Durch sie werden zeitaufwändige und kostspielige Laboruntersuchungen vermieden. Innerhalb kurzer Zeit können Parameterstudien mit deutlich reduziertem Aufwand gegenüber Experimenten durchgeführt und Produkte somit optimiert werden.

Die derzeit gängigste Methode ist die iterative Lösung der Reynoldsgemittelten Navier-Stokes Gleichungen (RANS, partielle Differentialgleichungen). Die turbulenten Schwankungen werden ganz oder teilweise mit Hilfe eines der zahlreichen, verfügbaren Turbulenzmodelle modelliert. Zur Diskretisierung des Berechnungsgebietes haben sich Finite-Volumen Methoden und Finite-Elemente Methoden bewährt, siehe FERZIGER ET AL. [2] und ZIENKIEWICZ [16].

Turbulente, instationäre Berechnungen in komplexen Geometrien gehören mittlerweile zu den Standardmethoden im Ingenieurbereich. Allerdings erfordern die zu lösenden Probleme in der Regel viel Speicherplatz und Rechenleistung. Durch Parallelisierung können diese Anforderungen erfüllt werden.

2.1 Spezielle Preprocessing Werkzeuge in COVISE

Durch Parametrisierung und Verwendung spezieller Parameter beim Design von Strömungsmaschinen kann auf bewährte Methoden zurückgegriffen werden. Daher bietet sich die Verwendung eines speziellen Entwurfsprogramms an, das auch zur Generierung des Berechnungsgitters eingesetzt werden kann.

Die von COVISE bereitgestellten Werkzeuge zum Geometrieentwurf von hydraulischen Maschinen bestehen im wesentlichen aus einer grafischen Oberfläche, zur Eingabe der Parameter. Zur Geometrieberechnung und Gittergenerierung werden externe Funktionen aufgerufen, die ein Geometrie- bzw. Gitterobjekt liefern.

Die Abbildungen 1 und 2 zeigen den Mapeditor, in dem der Datenfluss festgelegt wird, und die grafische Oberfläche des Fenfloss-Moduls. Das Modul RadialRunner liefert die Geometrie, die im Renderer dargestellt wird sowie das Berechnungsgitter mit automatisch erzeugten Randbedingungen für die Simulation im Fenfloss-Modul.

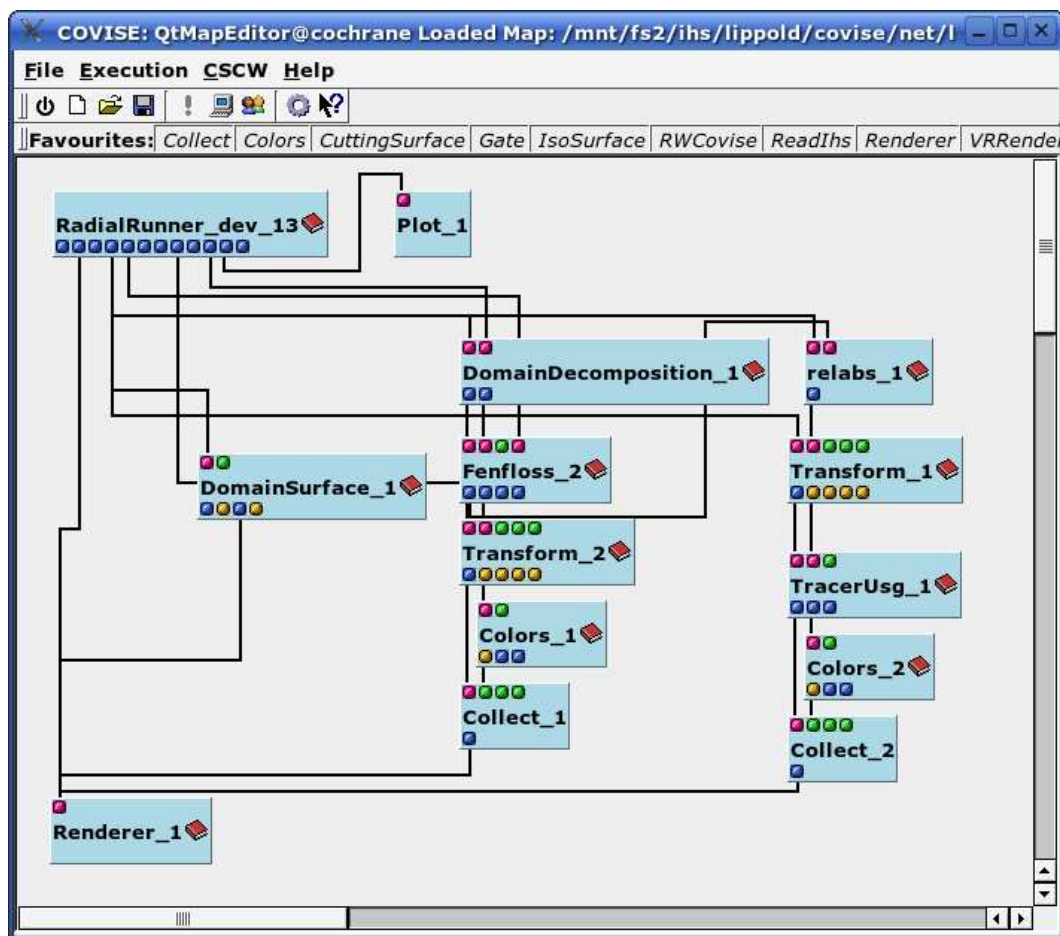


Abbildung 1: COVISE MapEditor mit Modulen zum Entwurf, Simulation und Visualisierung

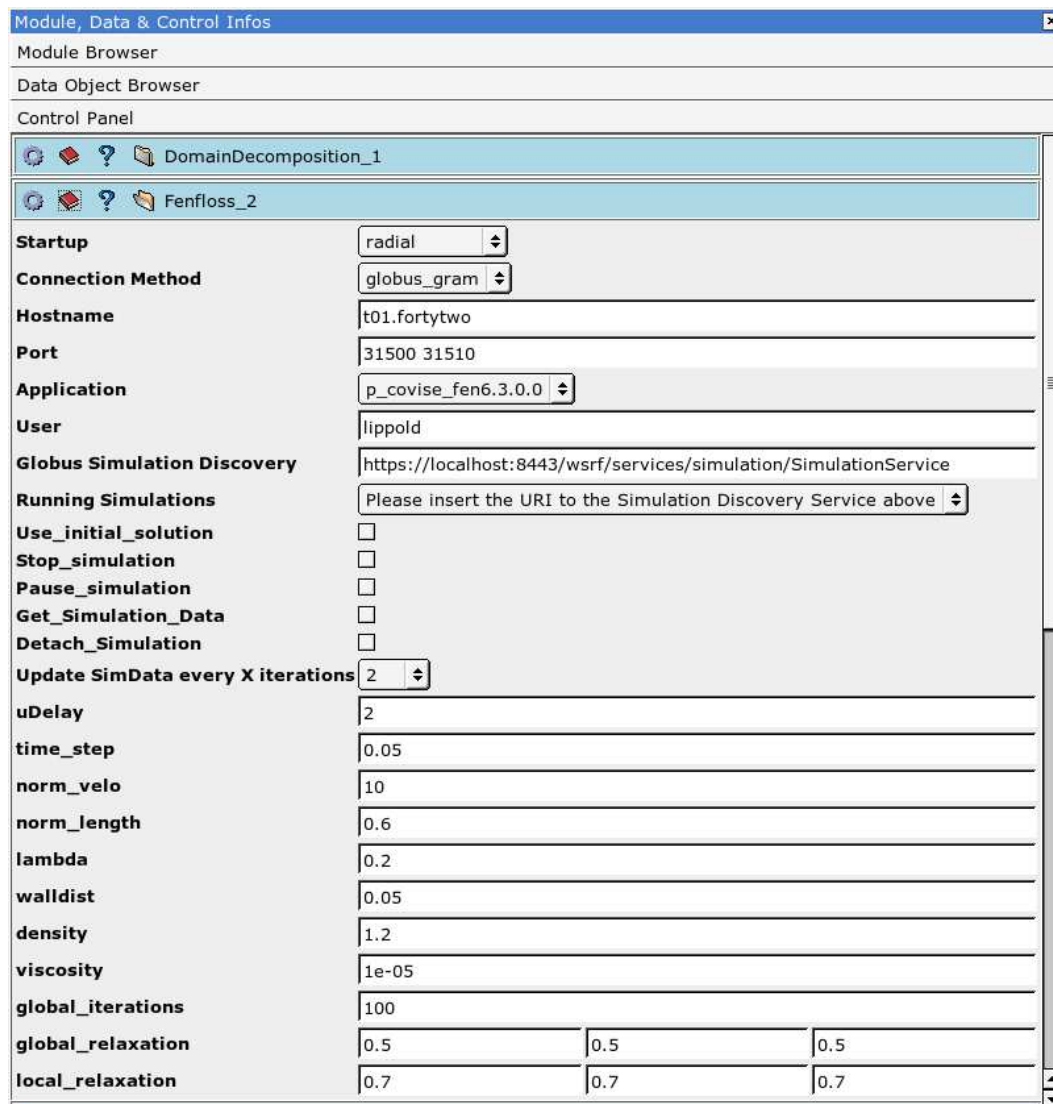


Abbildung 2: Grafische COVISE Oberfläche des Fenfloss-Moduls

2.2 Grundlegender Lösungsalgorithmus von FENFLOSS

Am IHS wird der CFD-Löser FENFLOSS zur Berechnung der stationären und instationären Strömung reibungsbehafteter Fluide eingesetzt und entwickelt. FENFLOSS verwendet zur Approximation der nichtlinearen, partiellen Differentialgleichungen einen Finite-Elemente Ansatz auf unstrukturierten Hexaedernetzen, siehe RUPRECHT [11]. Die diskretisierten Gleichungen werden in einer globalen Iteration gelöst, vgl. Abbildung 3, die für instationäre Simulationen die globale Iterationsschleife der impliziten Zeitintegration darstellt. Zur Glättung des Residuums des linearisierten Problems innerhalb der Iterationsschleife kommt ein iterativer Gleichungslöser auf konjugierte Gradientenbasis zum Einsatz, BICGStab(2), siehe VAN DER VORST [13].

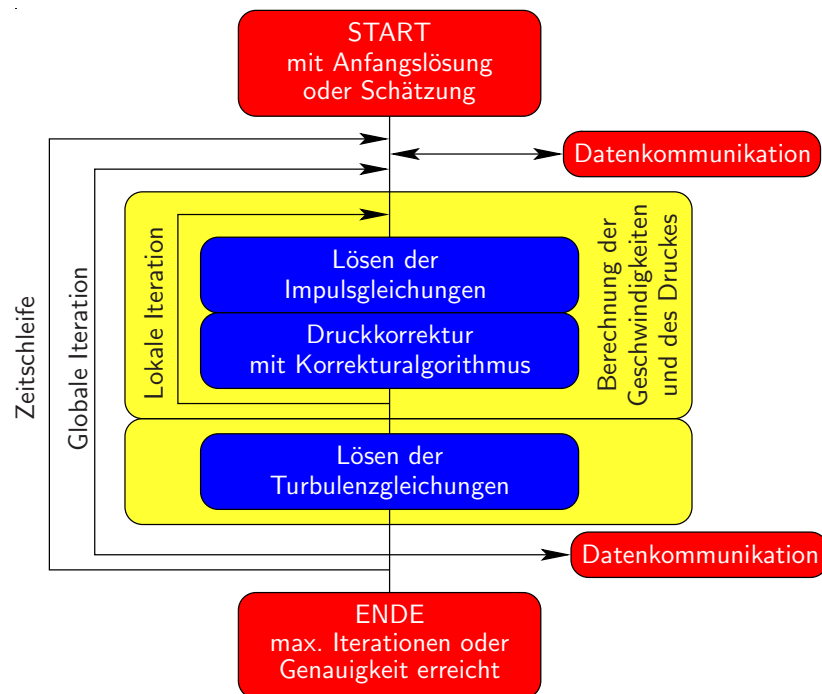


Abbildung 3: FENFLOSS - Basisalgorithmus

FENFLOSS bietet außerdem die Möglichkeit während der Berechnung Daten zur Kopplung mit anderen Berechnungsprogrammen (bspw. über MpCCI, siehe MpCCI [4]) oder zur Visualisierung (Anbindung an COVISE) auszutauschen.

2.3 Parallelisierung mit MPI

Zur Lösung großer Probleme steht eine parallelisierte Version von FENFLOSS zur Verfügung. Das Berechnungsgebiet wird in annähernd gleich große Teilgebiete zerlegt und die Anzahl der Knoten an deren Ränder, an denen Daten ausgetauscht werden, minimiert, siehe METIS [7] und Abbildung 4. Eine Zer-

legung in Gebiete unterschiedlicher Größe zur Nutzung von heterogenen Rechnernetzen ist derzeit nicht vorgesehen. Der Datenaustausch erfolgt über MPI, siehe MPI [9], im wesentlichen in der Iterationsschleife des Gleichungslösers. Der Vorteil ist dabei, dass der Algorithmus des seriellen Programmes beibehalten wird, und ein von der Anzahl der Prozessoren nahezu unabhängiges Konvergenzverhalten erreicht werden kann. Der Austausch im Matrix-Vektor und den Skalarprodukten des Lösers bedingt jedoch eine hohe Anzahl an Kommunikationen, was eine schnelle Verbindung zwischen den einzelnen Prozessoren erfordert.

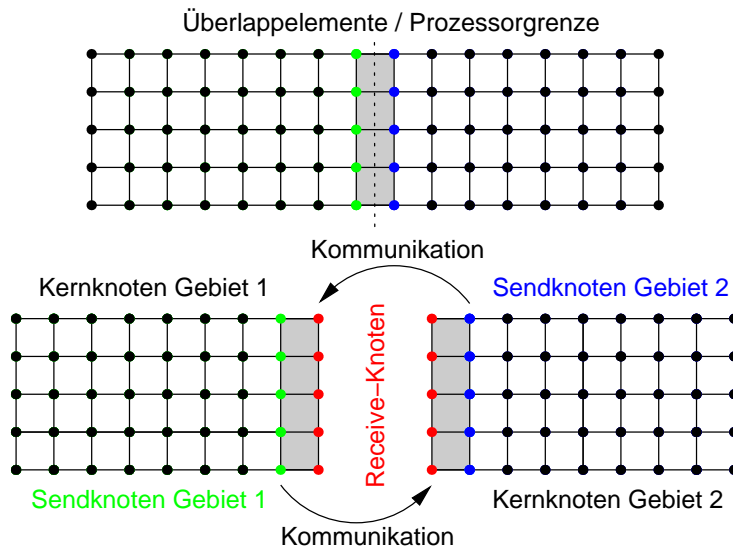


Abbildung 4: Datenaustausch zwischen zwei Partitionen des Berechnungsgebietes, aus MAIHÖFER [8]

3 Grid-Middleware

Die Idee von Grid-Computing ist es, komplexe wissenschaftliche Problemstellungen unabhängig von den vor Ort vorhandenen Rechner- und Softwareausstattung sowie Datenbanken zu lösen. Der Benutzer soll möglichst einfach auf diese Ressourcen zugreifen können. Dadurch wird die verwendete Middleware zu einer wichtigen Komponente.

Grid-Middleware bezieht sich auf verschiedene Leistungen (Services) wie Sicherheit, Ressourcenverwaltung, Datenzugriff, Accounting, Zugriffsrechte usw., die für Anwendungen, Benutzer und Ressourcenanbieter in einer Grid-Umgebung notwendig sind. Sie bietet dem Anwender Schnittstellen, die ihm die Bedienung und Nutzung aller im Grid beteiligten Systeme unabhängig von Plattform und Hersteller erleichtern. Entsprchende Software wird von verschiedenen Gruppen entwickelt. Häufig benutzte Middleware oder Grid-Computing Managementsysteme sind

- GLOBUS Toolkit (aktuell Version 4)
- UNICORE
- gLite
- Condor

3.1 Globus Toolkit 4

Globus Toolkit 4, vgl. FOSTER [3], ist eine weit verbreitete Open-Source Middleware, um auf Grid-Ressourcen zuzugreifen und stellt somit einen Standard im Bereich Grid-Computing dar. Es repräsentiert ein breites Spektrum an Funktionalitäten zur Erzeugung von Leistungen und Anwendungen auf verteilten Systemen. So bietet Globus Softwarefunktionen und Bibliotheken zur Erfassung und Verwaltung von Ressourcen an. Eine Stärke von Globus ist die große Flexibilität und Vielfältigkeit.

3.2 UNICORE

Die Software UNICORE (Uniform Interface to Computing Resources, UNICORE-WEBSITE [12]) bietet nahtlosen und sicheren Zugang zu verfügbaren Ressourcen an unterschiedlichen Standorten. Der Client verfügt über eine intuitiv und einfach zu bedienende grafische Oberfläche. Plattformunterschiede sind für den normalen Benutzer nicht sichtbar, wodurch ein nahtloses Portal für den Zugriff und die Verwendung von Supercomputern entsteht. Wegen seiner hohen Sicherheitsstandards und guten Bedienbarkeit wird UNICORE häufig zur Produktion im Bereich des wissenschaftlichen Rechnens eingesetzt. Leider bietet die aktuelle Produktionsversion im Gegensatz zu Globus keine Unterstützung für Web-Services.

3.3 gLite

gLite (www.glite.org) ist eine Middleware-Lösung, die im Rahmen des von der EU geförderten EGEE-Projektes entwickelt wird und ebenfalls als Open-Source verfügbar ist. Es integriert Komponenten anderer Middleware wie Condor und Globus. Es kombiniert eine Reihe von Services auf höherer Ebene mit Kernelementen auf niedriger Ebene. Durch eine serviceorientierte Architektur ist es möglich, eine Verbindung zur Software anderer Grid-Services herzustellen, wie z.B. dem Web Service Resource Framework (WSRF). Einsatzgebiete sind zur Zeit High Energy Physics, Biomedizin und Geowissenschaften.

3.4 Condor

Condor ist ein spezielles Managementsystem (eine Art Batchsystem), das besonders für Aufgaben mit hohem Berechnungsaufwand wie z.B. Optimierungen entwickelt wird. Es bietet einen Mechanismus für Queingsysteme und Ressourcenmanagement. Der Benutzer kann eine Job an Condor übermitteln, das die entsprechende Ressource entsprechend verschiedener Kriterien auswählt. Es unterstützt außerdem verschiedene Methoden und Protokolle. So ist Condor-G in der Lage Ressourcen, die von Globus verwaltet werden, zu verwenden. Verwendet wird Condor beispielsweise im GEODISE-Projekt, vgl. XUE ET AL. [15]

3.5 Auswahl der Middleware

Globus bietet von den oben besprochenen Middlewares die größte Flexibilität und Vielfaltigkeit. UNICORE ist zwar gut entwickelt, unterstützt aber in der aktuellen Version noch keine Web/Grid-Services. gLite und Condor bieten zwar alternative Ansätze sind aber noch relativ wenig untersucht und nicht so weit entwickelt. Der Vorteil von Globus ist, dass viele der neuen Middlewares Verbindungen zu Globusressourcen zulassen. Darum wird Globus für die weitere Entwicklung verwendet.

4 Konzepte für Entwurf und Berechnung in einer Grid-Umgebung

Geometrieentwurf und Gittergenerierung als Preprocessingschritte sind generell auf einzelnen PCs gut umsetzbar, was die Rechenleistung angeht. Durch die Vielfältigkeit der Plattformen, auf denen die Programme betrieben werden sollen, muss der Code jedoch immer für diese verfügbar sein. Diese Funktionen als Grid-Service anzubieten ist daher eine attraktive Möglichkeit. So können sie außerdem einzeln lizenziert und ihre Benutzung abgerechnet werden anstatt eine Lizenz zu kaufen.

Die Simulation von Strömungen (Computational Fluid Dynamics - CFD) gehört derzeit zu einer der Anwendungen, die hohe Anforderungen an die verwendeten Rechnerressourcen stellen; sowohl im Hinblick auf Speicherplatz als auch Rechenleistung. Moderne Großrechner bieten dies zwar aber auch nur in gewissen Grenzen. Folglich ergibt sich das Ziel, die Simulationen auf mehrere auch räumlich verteilte Ressourcen zu verteilen. Der Ansatz, CFD und Grid-Computing zu verbinden, liegt also nahe.

In der Regel ist die verfügbare Software z.B. für CFD nicht für die Verwendung auf verteilten und heterogenen Architekturen ausgelegt und muss daher angepasst werden. Ein weiterer, häufig vorausgesetzter, aber in der Praxis nicht immer selbstverständlicher Punkt, ist die Verfügbarkeit der verwendeten Software auf den ausgewählten Plattformen. Im Rahmen von Forschungsprojekten kann diese zwar erreicht werden, ist jedoch ansonsten nicht immer gegeben. Für industrielle Anwendungen spielt außerdem die Sicherheit eine große Rolle. Ein weiterer wichtiger Punkt zur Beurteilung der Qualität einer parallelen Rechnung ist die Zeit, die zur Kommunikation benötigt wird.

4.1 Anbindung der Entwurfswerkzeuge

Einzelne Funktionen der COVISE-Entwurfswerkzeuge können als Grid/Web-Service bereitgestellt werden. Hierzu ist im Fall der Verwendung von GLOBUS die Einbindung in ein Java-Programm notwendig, das den eigentlichen Grid-Service darstellt. Verschiedene Methoden können dann vom COVISE-Entwurfsmodule aufgerufen und auf dem Server bearbeitet werden. Als Ergebnis liefern sie die zur Visualisierung benötigten Daten.

Das Gitter und auch die Berechnungsergebnisse sind große Datenmengen, die von GLOBUS verschickt werden. Dadurch vergrößert sich die Größe der zu verschickenden Datenpakete zusätzlich und erhöht die Kommunikationszeiten.

4.2 Stand der Technik für Simulationen im Grid

Die Studie einiger Ansätze im Bereich wissenschaftliches Rechnen auf Grid-Architekturen, im Besonderen bezüglich CFD, liefert hier erste Zahlen und Anhaltspunkte. Auch hier werden vor allem die oben angesprochenen Aufgaben hinsichtlich Kommunikationszeiten und Heterogenität behandelt.

Bei KARONIS ET AL. [6] wird eine gridfähige MPICH-Implementierung (MPICH-G2) vorgestellt, die sich einiger Leistungen aus der Middleware GLOBUS bedient. Das vorgestellte Konzept, siehe Abbildung 5, soll vor allem dem Benutzer Transparenz bei gleichzeitiger Benutzerfreundlichkeit bieten. So können Berechnungen mit dem üblichen MPI-Befehl gestartet werden. Mit dem Unterschied, dass der GLOBUS-Teil der Implementierung die Verteilung auf die verfügbaren Ressourcen übernimmt. Messungen auf verschiedenen Maschinen zeigen, dass das vorgestellte Konzept der Verbindung von MPICH und GLOBUS gute Performance liefern kann. Zur besseren Lastverteilung muss die Heterogenität der Architektur jedoch vom Berechnungsprogramm bzw. dem Gebietszerleger selbst berücksichtigt werden.

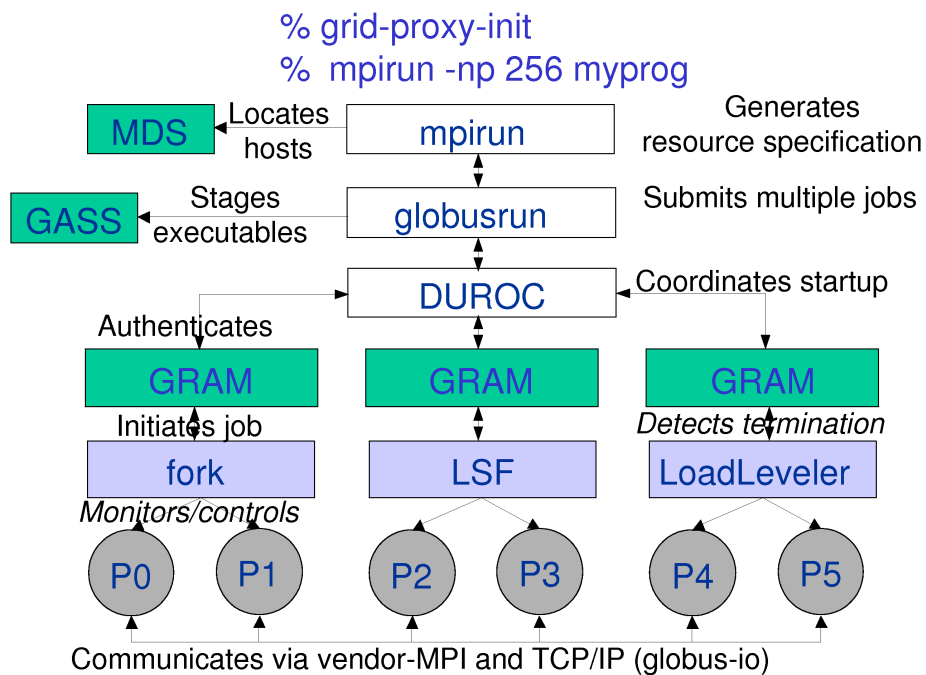


Abbildung 5: Ablaufschema einer Anwendung mit MPICH-G2, aus KARONIS ET AL. [6]

Als nächster Schritt kommt MPICH-G2 in ALLEN ET AL. [1] zur Grid-Anbindung des Berechnungsprogramms CACTUS zum Einsatz. CACTUS wurde speziell entwickelt, um mit heterogenen Architekturen und hohen Latenz-

zeiten umzugehen. So beinhaltet es nicht nur eine dynamische Lastverteilung sondern auch adaptive Überlappung der Berechnungsgebiete, die zwar zusätzlichen Berechnungsaufwand aber dafür weniger Kommunikationen bedeuten. So liefert dieser Ansatz auch bei Latenzzeiten von mehr als 100 ms immernoch einen Anteil der Rechenzeit von 80% an der Gesamtzeit.

In den Berichten von NIVET [10] zum MecaGRID-Projekt, und WORNOM [14] geht es um die Anbindung eines bereits existierenden, parallelen CFD-Codes. Auch hier kommt MPI zur Parallelisierung zum Einsatz. Desweiteren ist das Programm mit einem adaptiven Gebietszerleger ausgestattet, der die Heterogenität des Rechnernetzes berücksichtigt.

Da die IPs der einzelnen Clusterknoten nur innerhalb eines Clusters bekannt sind, kann von MPI keine direkte Verbindung zwischen Knoten unterschiedlicher Cluster aufgebaut werden. Zur Lösung dieses Problems wird ein VPN (Virtual private Network) installiert auf dem GLOBUS als Middleware installiert wird.

Rechenzeitvergleiche zwischen Simulationen auf homogenen Clustern und dem Netz zeigen deutliche Unterschiede in den absoluten Rechenzeiten. Der Großteil der Zeit auf dem Rechnernetz wird dabei zur Kommunikation benötigt. Auch bei Berücksichtigung der unterschiedlichen Prozessorgeschwindigkeiten ist die Netzlösung noch deutlich im Nachteil. Ein Punkt zur Erklärung dieses Unterschiedes ist die langsame Verbindung zwischen den einzelnen Clustern, die eine 5 Mb/s Leitung verwendet.

Ein aktuelles Projekt der selben Forschungsgruppe befasst sich mit der Strömungssimulation auf einem Rechnernetz mit 5000 Prozessoren. Erste Messungen liefern hier sogar trotz schneller Verbindungen (1 Gb/s) noch schlechtere Zeiten als im oben besprochenen MecaGRID. Der genaue Grund wird zwar nicht genannt aber als wesentlicher Faktor wird die hohe Latenzzeit genannt, die auf Grund der größeren Entfernungen entsprechend ansteigt.

4.3 Grid-Anbindung von FENFLOSS

Die oben besprochenen Ergebnisse liefern eine wichtige Basis, um die Möglichkeiten und Bedürfnisse einer Grid-Anbindung von FENFLOSS festzulegen. Im vorhandenen Zerleger wird eine homogene Rechnerarchitektur sowie eine bekannte Anzahl an Prozessoren vorausgesetzt. Rechnungen auf heterogenen Netzen führen also zu ungleicher Lastverteilung und damit geringer Effizienz. Auch die sehr feingranulare Kommunikationsstruktur von FENFLOSS bedeutet eine besondere Sensibilität hinsichtlich der Latenzzeiten. Die zu übertragenden Datenmengen liegen im Bereich unter 1 Mb pro Paket, was bei den verfügbaren schnellen Verbindungen kein Problem darstellt. Durch hohe Latenzzeiten wird der Zeitaufwand für die Kommunikation also enorm ansteigen.

Eine Möglichkeit zur Parallelisierung, die auch bei MAIHÖFER [8] untersucht wird, ist der Einsatz eines Teilgebietslösers. Dies bedeutet, den Code größtenteils neu zu schreiben und anzupassen und stellt somit einen enormen Aufwand

dar. Es wird außerdem gezeigt, dass auf Grund der zusätzlich notwendigen Iterationen, die Konvergenz und der Speed-up von Teilgebietslösern deutlich schlechter ist als bei der implementierten Methode.

5 Schlußfolgerungen und Möglichkeiten

Die Implementierung der zur Geometrie- und Gittergenerierung verwendeten Funktionen als Java-Methoden eines Grid-Services kann umgesetzt werden. Noch nicht geklärt ist der Sicherheitsaspekt bezüglich der Sicherheit der verschickten Daten.

Sowohl die Anpassung des Gebietszerlegers an heterogene Rechnernetze als auch die Verringerung der Granularität der Kommunikation erfordern extrem große Änderungen und teilweise Verschlechterungen des CFD-Codes FENFLOSS. Sie sind darum hier nicht vorgesehen.

Im Hinblick auf die Anbindung von FENFLOSS an die Visualisierungssoftware COVISE und die geplanten Fluid-Struktur-Simulationen wird verstärkt an einer Umsetzung eines CFD-Grid/Web-Services gearbeitet. Auch die Verwendung verschiedener Ressourcen für Parameterstudien und Optimierungen sind umsetzbar. Hier ruft der Optimierer die Auswertung der Zielfunktion, in diesem Fall eine CFD-Analyse, als Grid-Service auf einer homogenen Ressource auf. Bei diesen Szenarien sind zwar die zu übertragenden Datenmengen im dreistelligen MB-Bereich für die Visualisierung, Kommunikation findet aber wesentlich seltener statt. Daher spielt vor allem die verfügbare Bandbreite eine Rolle und nicht die Latenzzeiten. Dies stellt eine lösbare Herausforderung dar.

Literatur

- [1] ALLEN, G., DRAMLITSCH, T., FOSTER, I., KARONIS, N.T., RIPEANU, M., SEIDEL, E. AND TOONEN, B.: *Supporting efficient execution in heterogeneous distributed computing environments with Cactus and Globus.*, Supercomputing '01: Proceedings of the 2001 ACM/IEEE conference on Supercomputing (CDROM), Denver, Co, 2001.
- [2] FERZIGER, J.H., PERIĆ, M.: *Computational Methods for Fluid Dynamics (third Ed.)*. Springer, 2002.
- [3] FOSTER, I.: *Globus Toolkit Version 4: Software for Service-Oriented Systems*. IFIP International Conference on Network and Parallel Computing, Springer LNCS 3779, 2-13, 2005.
- [4] FRAUNHOFER SCAI: *MpCCI 3.0 Manual*, www.scai.fraunhofer.de/mpcci.html, Fraunhofer SCAI, 2005.
- [5] GRESHO, P.M., SANI, R.L.: *Incompressible Flow and the Finite Element Method, Vol. I*. John Wiley & Sons (1999).
- [6] KARONIS, N.T., TOONEN, B., FOSTER, I.: *MPICH-G2: A Grid-Enabled Implementation of the Message Passing Interface.*, <http://www-fp.mcs.anl.gov/toonen/Papers/mpich-g2-jpdc-2003-preprint.ps>, 2002.
- [7] KARYPIS, G., KUMAR, V.: *METIS: A software package for partitioning unstructured graphs, partitioning meshes and computing fill-reduced orderings of sparse matrices.*, User's Manual - Version 4.0, 1998.
- [8] MAIHOEFER, M.: *Effiziente Verfahren zur Berechnung dreidimensionaler Stroemungen mit nichtpassenden Gittern*. Dissertation, Universität Stuttgart, 2002.
- [9] MPI-FORUM: *A Message-Passing Interface.*, Message Passing Interface Forum, 1995.
- [10] NIVET, P.: *A computational grid devoted to fluid mechanics*. INRIA, Rapport de Recherche, 2006
- [11] RUPRECHT, A.: *Finite Elemente zur Berechnung dreidimensionaler turbulenter Stroemungen in komplexen Geometrien*. Dissertation, Universität Stuttgart, 1989.
- [12] UNICORE: www.unicore.org
- [13] VAN DER VORST, H.A.: *BI-CGSTAB: A fast and smoothly converging variant of BI-CG for the solution of nonsymmetric linear systems.*, SIAM J. Sci. Stat. Comp., **13**, 631-644, 1992
- [14] WORNOM, S.: *Dynamic Load balancing and CFD Simulations on the MecaGRID and GRID5000*. INRIA, Rapport de Recherche No. 5884, 2006

- [15] XUE, G., FAIRMAN, M., POUND, G.E., COX, S.J.: *mplementation of a Grid Computation Toolkit for Design Optimisation with Matlab and Condor.* , Euro-Par 2003 Parallel Processing, Lecture Notes in Computer Science, 2790, 357-365, 2003
- [16] ZIENKIEWICZ, O.C., TAYLOR, R.L.: *The Finite Element Method, Vol. I* McGraw-Hill, 1989.