

## Parallel Programming for Multi-core Architectures (ParMA)

Im Rahmen des europäischen Forschungsprojektes *Parallel Programming for Multi-core Architectures (ParMA)* hat die *GNS mbH* das Laufzeitverhalten der impliziten Finite-Element-Software *INDEED* analysiert und optimiert.

*ParMA* [1] wird unter der Schirmherrschaft der Organisation *ITEA* und der Beteiligung französischer, spanischer und deutscher Partner aus Forschung und Industrie durchgeführt. Ziel ist es zum einen, neue Werkzeuge zu entwickeln, mit denen der Prozess der Programmparallelisierung vereinfacht wird. Zum anderen sollen bestehende Werkzeuge durch Anwendung von Industrieapplikationen getestet werden. Die gewonnenen Erkenntnisse ermöglichen dann, sowohl die Werkzeuge als auch die Industrieapplikationen zu optimieren.

*GNS* hat für das FE-Umformsimulationsprogramm *INDEED* im Rahmen des *ParMA*-Projektes unter anderem die

Aufbauphase der Elementsteifigkeitsmatrix analysiert. Diese erfolgt in jeder Gleichgewichtiteration und ist verhältnismäßig rechenzeitintensiv. Daher wurde sie mithilfe von Direktiven des *OpenMP Standards* bereits parallelisiert und kann somit von mehreren Prozessen gleichzeitig bearbeitet werden. Das Assemblieren der Gesamtsteifigkeitsmatrix kann innerhalb der Aufbauphase jedoch nur bedingt parallel erfolgen, da hierbei unter Umständen mehrere Prozesse gleichzeitig auf einen gemeinsamen Speicherbereich zugreifen. Abhängig davon, welcher Prozess dabei zuerst von einer gemeinsamen Adresse liest oder diese nach einer Weiterverarbeitung wieder beschreibt, verfälscht sich das Gesamtergebnis; man spricht dann von einer Wettlaufsituation. Der *OpenMP Standard* bietet zur Umgehung dieses Problems die Verwendung spezieller Anweisungen in einer parallelen Schleife. Diese gewährleisten, dass innerhalb ihrer Definition niemals mehrere Prozesse gleichzeitig den betreffenden Programmabschnitt ausführen.

Mithilfe des Programms *Scalasca* [2] wurde der Einfluss dieser Abschnitte auf

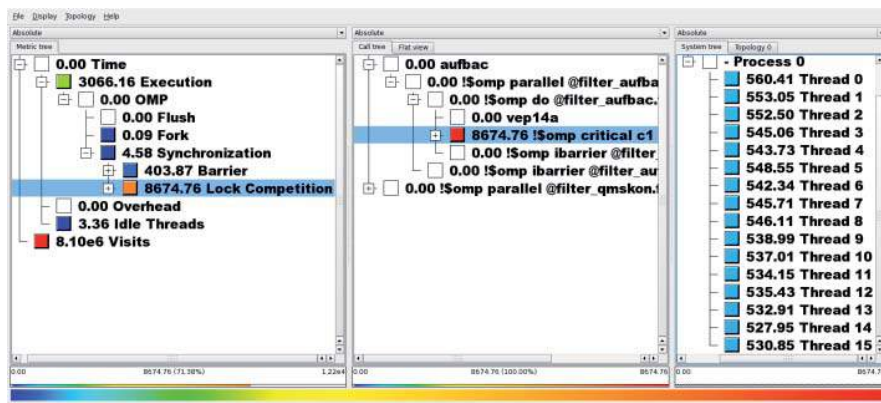


Abb. 1: Laufzeitergebnisse der mit Scalasca analysierten Aufbauphase

## News

- ➔ *GNS Systems* erhält Auftrag von der **Wilhelm Karmann GmbH** zur Unterstützung beim Einsatz von *CAE Bench*.
- ➔ **Hyundai Motor Technical Center Europe** beauftragt *GNS Systems* erneut mit *CAE Support*.

## Beiträge

- ➔ *Parallel Programming for Multi-core Architectures (ParMA)*
- ➔ *Microsoft Windows HPC Server 2008 (Teil 2/2)*
- ➔ *PartnerGrid - Kooperative Grid-Lösungen für industrielle Anwendungen*
- ➔ *Das Speichern von Einstellungen in Animator4*



**GNS** Systems

die Laufzeit untersucht. Ziel der Entwicklung des Programms durch das Forschungszentrum Jülich war es, Informationen zur Laufzeit bzw. zu ineffizientem Programmverhalten für den Anwender übersichtlich zusammenzufassen. *Scalasca* kann hierbei Informationen sowohl bezüglich einer ineffizienten Nutzung paralleler Programmiermodelle aufzeichnen als auch Probleme beim Zugriff auf Speicher dokumentieren.

Abbildung 1 zeigt die Ausgabe einer Analyse von Teilen der Aufbauphase für einen kompletten Rechenlauf. Auf der linken Seite des Ausgabefensters werden verschiedene Metriken dargestellt, die das Laufzeitverhalten des analysierten Programms beeinflussen. In diesem Fall wird durch „Lock Competition“ darauf hingewiesen, dass einzelne Prozesse darauf warten, einen bestimmten Programmabschnitt ausführen zu können. Dieser wird von anderen Prozessoren blockiert. Im mittleren Teil des Ausgabefensters ist zu sehen, wo genau dieses Problem auftritt. Die rechte Seite gibt die Verteilung des Problems über die einzelnen Prozessoren an.

Als weiteres Programm zur Analyse des Laufzeitverhaltens wurde *Vampir* [3] eingesetzt. *Vampir* wird von der Tech-



Abb. 2: Laufzeitanalyse einer Aufbauphase mit *Vampir*

nischen Universität Dresden entwickelt; der Vertrieb erfolgt durch die *GWT GmbH*. Im Gegensatz zu *Scalasca* stellt dieses Werkzeug eine Analyse des untersuchten Codes über der Zeit dar. Durch das Vergrößern relevanter Abschnitte des Zeitachsenfensters kann der Anwender hierbei einen sehr detaillierten Ein-

blick in das Verhalten des untersuchten Programms erhalten. So zeigt *Abbildung 2* den Ablauf einer einzelnen Aufbauhäse auf 16 Prozessoren. *Vampir* unterteilt hierbei das Zeitintervall in Anteile, in der die Applikation Rechenzeit beansprucht oder aber den Zeitanteil, der

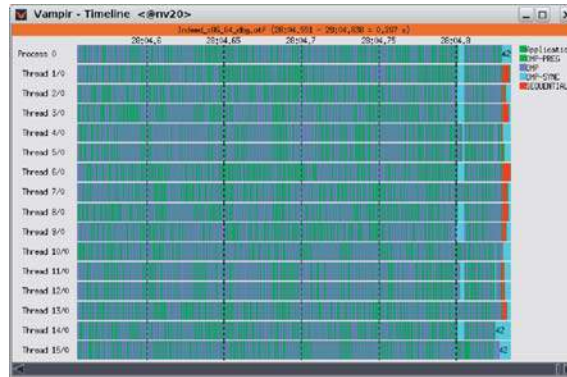


Abb. 3: Laufzeitanalyse der optimierten Aufbauphase

z.B. für eine Synchronisation benötigt wird. In der Abbildung rot dargestellt sind die Anteile, die innerhalb der parallelen Schleife sequentiell abgearbeitet werden und somit zu Laufzeitverlängerungen führen. Wie zu erkennen ist, dominiert dieser Anteil die Aufbauphase stark. Im Rahmen des *ParMA*-Projektes wurde für die Aufbauphase die bestehende Parallelisierung von der *GNS mbH*

erweitert und dahingehend optimiert, dass die sequentiellen Programmanteile minimiert wurden. Gleichzeitig musste aber gewährleistet werden, dass es nicht zu den eingangs beschriebenen Wettlaufsituationen kommt. Dieses konnte durch eine interne Vorsortierung der Elemente mithilfe von Algorithmen der *METIS*-Bibliothek [4] erreicht werden. Hierbei werden die Elemente in Gruppen eingeteilt, für

die gewährleistet ist, dass es zu keinen Überschneidungen im Speicherbereich kommt. In einem ersten Schritt wird dann in den Assemblierungsroutinen der größte Anteil der Elemente komplett parallel bearbeitet.

Am Ende erfolgt eine Synchronisation

aller Prozesse, bevor in einem zweiten Schritt eine zweite Gruppe von Elementen ebenfalls parallel bearbeitet wird. In der Assemblierung von Restelementen sind sequentielle Anteile enthalten, wie der *Abbildung 3* zu entnehmen ist.

Die Dauer der parallelen Aufbauhäse konnte mithilfe der durchgeführten Optimierungen deutlich verkürzt werden. Als Maß hierfür kann der so genannte Speed-up herangezogen werden. Dieser gibt an, um welchen Faktor sich die Laufzeit bei der Verwendung mehrerer Prozessoren reduziert. Im Idealfall steigt dieser linear an. In der Realität ist dieses, z.B. aufgrund von Synchronisierungen, jedoch selten der Fall.

*Abbildung 4* zeigt die Skalierung der Aufbauhäse vor und nach der beschriebenen bzw. weiteren Optimierungen.

Insgesamt konnte durch die Arbeiten im *ParMA*-Projekt die Laufzeit der Software *INDEED* um ca. 30-40% reduziert werden. Anwender können ab Version 8.0.4 hiervon profitieren.

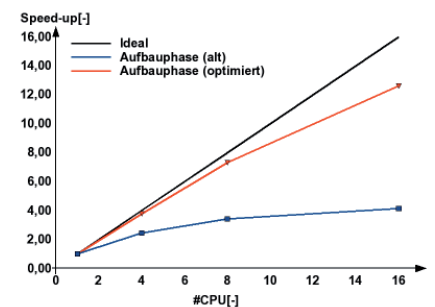


Abb. 4: Speed-up der alten und der optimierten Aufbauhäse

- [1] <http://www.parma-itea2.org/>
- [2] <http://www.fz-juelich.de/jsc/scalasca/>
- [3] <http://www.vampir.eu/>
- [4] "A Fast and Highly Quality Multilevel Scheme for Partitioning Irregular Graphs". George Karypis and Vipin Kumar. *SIAM Journal on Scientific Computing*, Vol. 20, No. 1, pp. 359-392, 1999

Rene Menzel, *GNS mbH*  
menzel@gns-mbh.com

## Microsoft Windows HPC Server 2008 (Teil 2/2)

Wie in der letzten Ausgabe des Newsletters (1/2009) schon angekündigt, stellen wir Ihnen hier im zweiten Teil dieses Artikels die Themenbereiche Ressourcen-Management, Administration des laufenden Cluster-Systems und Nutzung des Cluster-Systems durch den Anwender vor.

### Einfache Komplettlösung

Mit dem *Windows HPC Server 2008* bietet Microsoft eine Softwareinfrastruktur-Lösung für den Einsatz in sehr rechenintensiven Anwendungsbereichen. Zu den möglichen Anwendungsgebieten gehören z.B.:

- Simulation in der Automobil- und der Luft- und Raumfahrtindustrie
- Konvertierung von Konstruktionsdaten
- Berechnung von photorealistischen Bildern

*Windows HPC Server 2008* ist für AMD x86\_64 oder Intel EM64T Prozessortechnologie verfügbar und beinhaltet einen kompletten Satz an Funktionsmodulen für den Einsatz in Berechnungsumgebungen jeglicher Größe.

### Resource Management

Nachdem das Netzwerk und die Knoten eingerichtet sind, kann eine weitere Konfiguration des Clusters erfolgen. Es gibt eine Reihe von verschiedenen Mechanismen, mit denen insbesondere die Verwendung von Ressourcen, z.B. CPUs, Plattenplatz, Anwendungen und Lizenzen, gesteuert werden kann. Diese Mechanismen sind in *Abbildung 1* aufgeführt.

### Administration

Für die Administration und Überwachung eines existierenden Clusters werden von dem *Windows HPC Server 2008* eine Reihe von Werkzeugen angeboten. Die unterschiedlichen Aufgabengruppen können über das in *Abbildung 2* aufge-

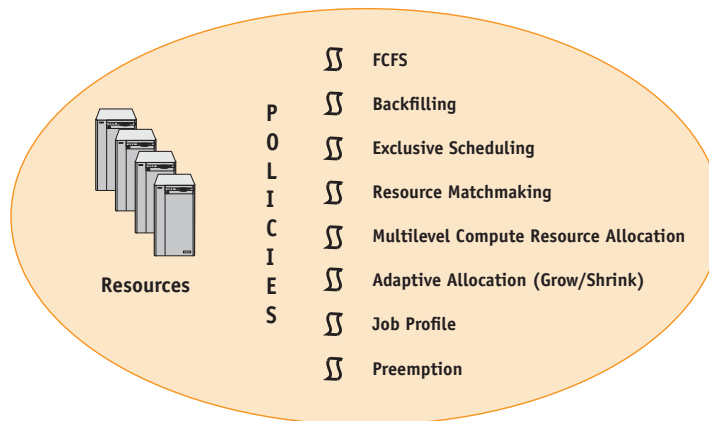


Abb. 1: Mechanismen für das Ressourcen-Management

führte Menü ausgewählt werden. Eine Aufgabengruppe ist z.B. das Node Management. Das hierfür vorhandene Menü ist in *Abbildung 3* dargestellt. In diesem Menü sind folgende Aktionen möglich:

- Darstellung und Änderung der Knoten-Eigenschaften
- Darstellung und Änderung des Knoten-Zustandes
- Darstellung des Zustandes von Knoten-Ressourcen, zu CPU Auslastung oder Memory Nutzung
- Ausführung von Befehlen auf Knoten oder Knotengruppen
- Ausführung von Diagnostik-Routinen
- Darstellung des Netzwerkverkehrs

- Darstellung von Event-Historien
- Neuinstallation des Betriebssystem-Images

### Nutzung

*Windows HPC Server 2008* bietet für die Nutzung durch den Anwender Oberflächenelemente an, über die Batch Jobs erstellt und eingereicht werden können. Die Jobs selber bestehen wiederum aus einer oder mehreren Aufgaben (Tasks), die in Abhängigkeit zueinander stehen können. Sämtliche Tasks können unterschiedliche Systemanforderungen wie z.B. die benötigte Anzahl CPUs stellen. Um die Erstellung von Jobs für den Anwender zu vereinfachen, können Jobs

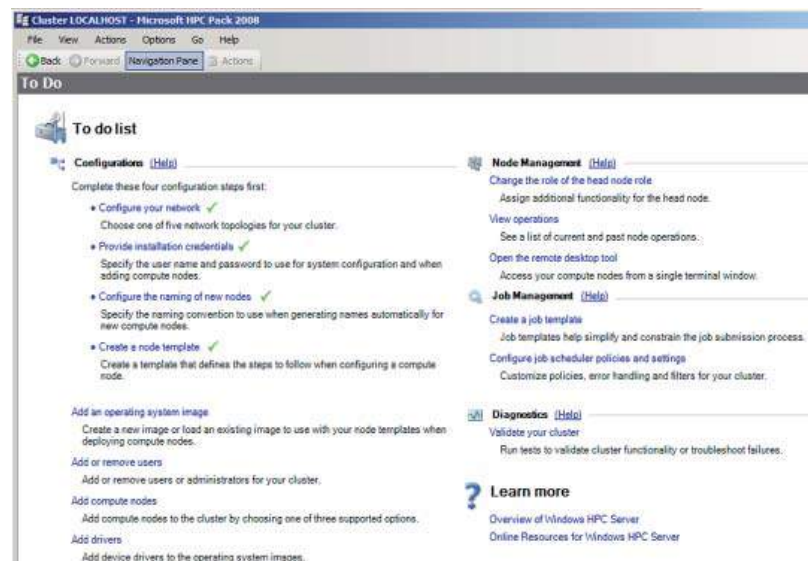


Abb. 2: Menü zur Auswahl von Aufgabengruppen für die Cluster Administration



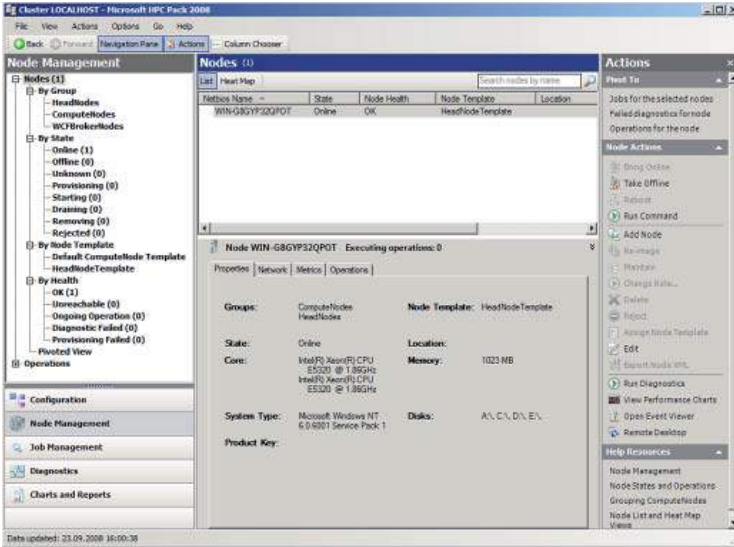


Abb. 3: Menü für die Aufgabengruppe Node Management

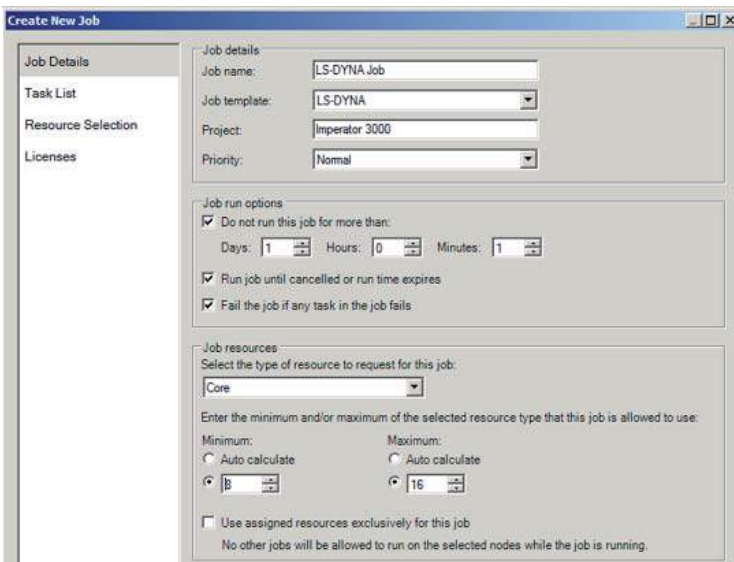


Abb. 4: Grafische Oberfläche für die Definition von Jobs

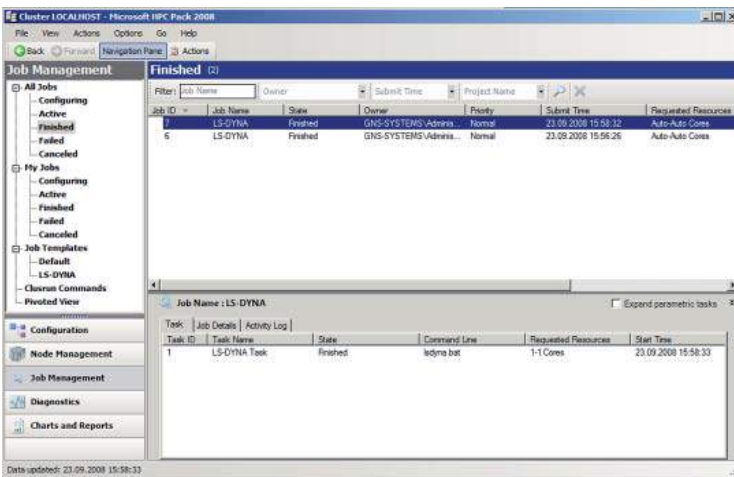


Abb. 5: Menü für das Job- und Task Management

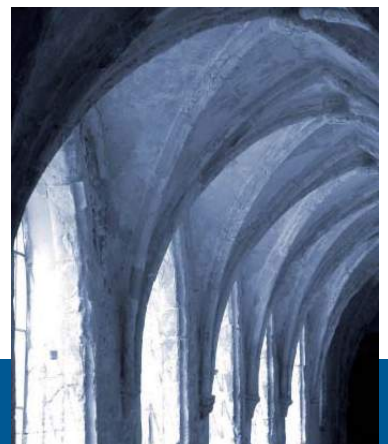
auf Basis von vorab erstellten Mustern, den Job Templates, erzeugt werden. *Abbildung 4* zeigt das Menü für die Definition eines Jobs. In diesem Menü kann das gewünschte Job Template, auf dem der Job beruhen soll, ausgewählt werden. In weiterführenden Menüs können außerdem notwendige Angaben zu einzelnen Tasks und dem Ressourcen- und Lizenzbedarf des Jobs gemacht werden.

Die Job Queue und die darin enthaltenen Jobs bzw. Tasks können ebenfalls über eine dedizierte grafische Oberfläche verwaltet werden. Diese ist in *Abbildung 5* dargestellt. Die Automatisierung von Prozessabläufen innerhalb der einzelnen Tasks sollte über die *Windows HPC PowerShell* erfolgen.

Die von den Oberflächenelementen bereitgestellte Funktionalität kann auch über folgende Kommandozeilenbefehle genutzt werden:

- job Jobs verwalten
- task Tasks verwalten
- node Knoten aufnehmen und verwalten
- cluscfg Queue überwachen und verwalten
- clusrun Befehl auf Cluster-Knoten ausführen

Job Informationen werden im XML Format gespeichert und so an den Job Scheduler übergeben. Über diesen Mechanismus können auch andere Softwarewerkzeuge für die Erstellung von Jobs verwendet werden. Diese müssen lediglich die relevanten Informationen im vorgegebenen XML-Format erzeugen. Ein Beispiel für ein solches Werkzeug zur Job-Erstellung ist in *Abbildung 6* dargestellt. Vorteil eines solchen Werk-



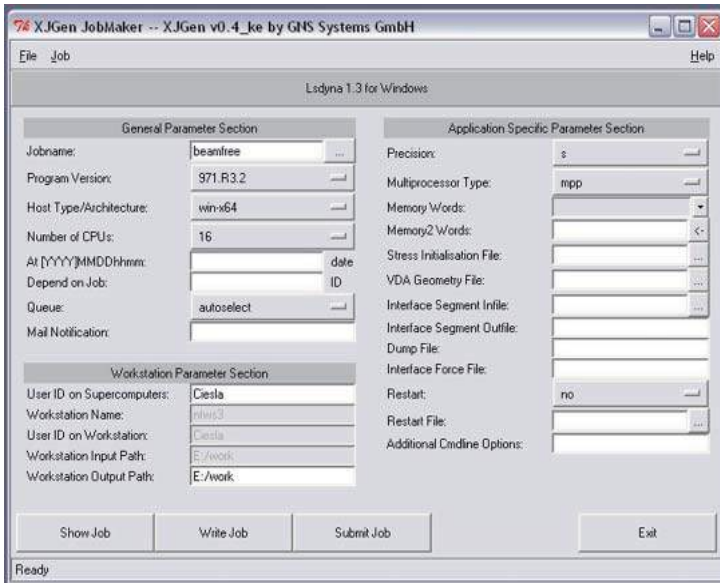


Abb. 6: Softwarewerkzeug für die Erstellung von Batch Jobs

zeugs gegenüber den vom *Windows HPC Server* angebotenen Methoden ist, dass es an eine spezifische Anwendung und eine spezifische Cluster-Konfiguration angepasst ist.

Natürlich kann der *Windows HPC Server 2008* in umfangreiche Gesamtprozesse, wie z.B. die in *Abbildung 7* dargestellte CAE-Arbeitsumgebung, integriert werden.

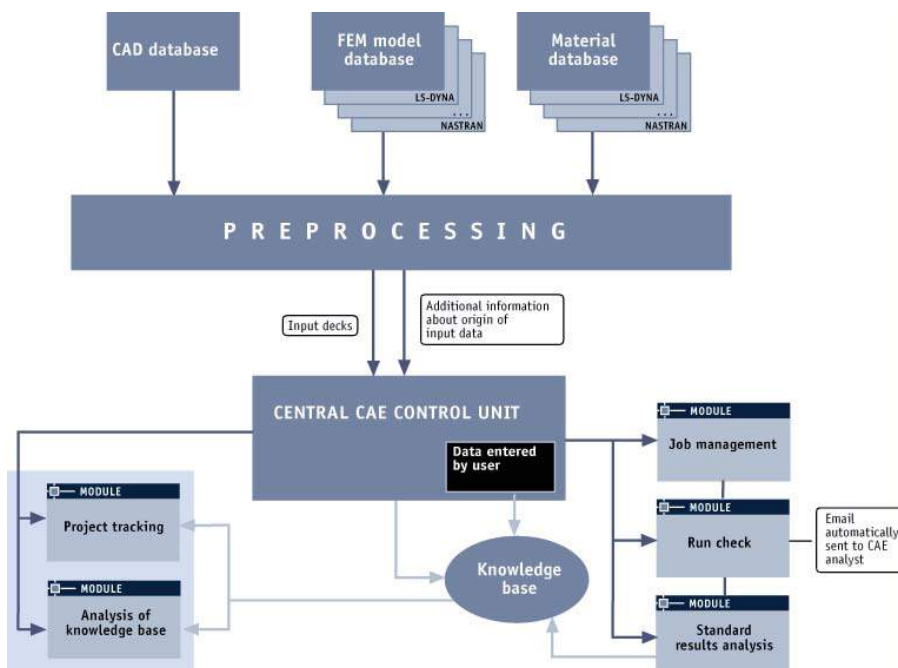


Abb. 7: Beispiel für eine CAE-Arbeitsumgebung

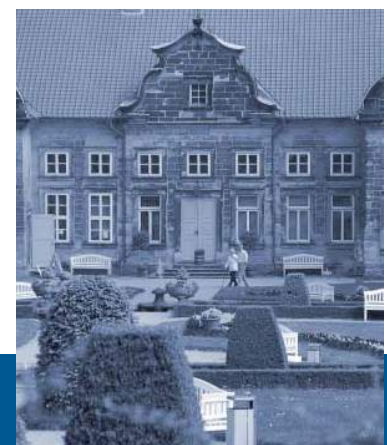
## Zusammenfassung

*Windows HPC Server 2008* ist eine umfangreiche Softwareinfrastrukturlösung von *Microsoft* für den Einsatz in sehr rechenintensiven Anwendungsbereichen. Die Lösung ermöglicht die Konfiguration und Administration der Nutzung von *Windows*-basierten Compute Server Systemen. Sie wird ergänzt durch eine umfangreiche Sammlung weiterer *Microsoft*-Komponenten und -Dienste mit denen beliebige HPC-Infrastrukturen aufgebaut werden können.

## Dienstleistungen

Als IT-Dienstleister kann *GNS Systems* sämtliche Aufgaben im Zusammenhang mit einer Infrastruktur auf Basis von *Windows HPC Server 2008* übernehmen. Diese reichen von der Planung und der Implementierung bis zum kompletten Betrieb.

Jan Martini, GNS Systems GmbH  
jan.martini@gns-systems.de



## PartnerGrid - Kooperative Grid-Lösungen für industrielle Anwendungen

Am 1. Juni 2007 startete das vom *Bundesministerium für Bildung und Forschung* geförderte Projekt *PartnerGrid*. Ziel des Projektes ist die kooperative Nutzung von Grid-Ressourcen zur Lösung industrienahe Problemstellungen in den Bereichen Umformsimulation und Gießereitechnik. Bei dem Forschungsprojekt steht die Nutzung der nationalen Grid-Rechenressourcen im Vordergrund. Diese Ressourcen werden von diversen deutschen Forschungs- und Rechenzentren betrieben und bereitgestellt.

Im *PartnerGrid* wurde eine erweiterte Infrastruktur erarbeitet, welche das bestehende nationale *D-Grid* um *PartnerGrid*-spezifische Funktionen erweitert und eine einfache, kooperativ nutzbare Webschnittstelle, das Portal, hinzufügt.

Im Folgenden wird ein Anwendungsfall präsentiert, anschließend werden einige erforschte Technologien näher vorgestellt.

### Projektentwicklung: Kollaborationsplattform und Simulation im Grid

In *Abbildung 1* ist exemplarisch die durch die *PartnerGrid*-Infrastruktur unterstützte Abwicklung eines Projektes dargestellt.

Am Anfang eines Projektes steht die Kontaktaufnahme des Kunden mit dem Berechnungsdienstleister. Diese Kontaktaufnahme kann durch das Portal „geführt“ und beschleunigt werden. Möglicherweise handelt es sich um einen Neukunden, in diesem Fall hilft das Portal bereits bei der Aufnahme der Kundendaten. Der nächste Schritt beinhaltet die Initialisierung eines Projektes, auch hier hilft das Portal bei der Datenaufnahme.

Nach dieser initialen Phase folgt die gesicherte Übertragung des Eingabedatensatzes zum Portal. Dieser wird erst von einem Berater des Dienstleisters begutachtet, der die Simulationsrech-

nungen für den Kunden plant und mit Grid-Ressourcen durchführt. Die Dienstleistung selbst kann ein iterativer Prozess sein. Generell besteht eine Berechnungsdienstleistung aus dem Pre-processing, der eigentlichen Simulation (z.B. mit *INDEED*) und dem Postprocessing (z.B. mit *Animator4* und *Evaluator*). Automatisierbare, rechenintensive Teile, vor allem die eigentliche Simulation, können ins Grid ausgelagert werden. Bei der Durchführung der Beratungsdienstleistung können Rücksprachen mit dem Kunden erforderlich sein. Auch dieses wird vom Portal unterstützt. Es stellt hierzu ein Projektlogbuch, projektspezifische Diskussionsforen und weitere Kollaborationselemente zur Verfügung. Der letzte Schritt eines Projektes endet mit der Erstellung des Projektberichtes, der Rechnungsstellung und gegebenenfalls der kurzzeitigen Archivierung der Projektdaten. Von einer Langzeitarchivierung der Simulationsdaten im Grid oder im Portal wird aus Sicherheitsgründen abgesehen.

### Schlüsseltechnologien

Im Folgenden werden einige im *PartnerGrid* erforschte Schlüsseltechnologien vorgestellt. Dazu gehören:

- Virtualisierung
- Lizenzvirtualisierung
- Komponentenbasierte Software-Architekturen
- Internetportale

### Virtualisierung

Grid-Ressourcen sind stark heterogen, jeder Anbieter von Rechenressourcen im Grid nutzt andere Hardware und installiert andere Betriebssysteme. Systembibliotheken haben unterschiedliche Versionsstände. Dieser heterogenen Umgebung kann man mit zwei Methoden begegnen: zeit- und kostenintensiver Pflege eines Applikationsverzeichnis für verschiedene Umgebungen oder Virtualisierung.

Im *PartnerGrid* wurde die Virtualisierung als Lösung gewählt. Diese kostet zwar Performance, ermöglicht aber die Erschaffung einer homogenen Umge-

### PartnerGrid im Überblick

*PartnerGrid - Kooperative Grid-Lösungen für industrielle Anwendungen*

Projektstart: 1.6.2007

Projektende: 31.5.2010

Laufzeit: 3 Jahre

Fördervolumen: 1,6 Millionen Euro

Forschungspartner:

- Fraunhofer-Institut für Arbeitswirtschaft und Organisation IA0, Stuttgart
- Deutsches Zentrum für Luft- und Raumfahrt e.V., Einrichtung Simulations- und Softwaretechnik, Köln
- Fraunhofer-Institut für Algorithmen und Wissenschaftliches Rechnen SCAI, Sankt Augustin
- Fraunhofer-Institut für Techno- und Wirtschaftsmathematik ITWM, Kaiserslautern

Industriepartner:

- GNS mbH, Braunschweig
- GNS Systems GmbH, Braunschweig
- MAGMA Gießereitechnologie GmbH, Aachen

WWW: <http://www.PartnerGrid.de/>

Das Projekt *PartnerGrid* wird vom Bundesministerium für Bildung und Forschung unter dem Förderkennzeichen 01G07009A-D gefördert.

bung. Die einzige Voraussetzung ist eine auf den Rechenressourcen installierte Virtualisierungslösung. Im Falle von *PartnerGrid* wurde *Xen* als Virtualisierungstechnik gewählt.

### Lizenzvirtualisierung

Eine wichtige Fragestellung in virtualisierten Umgebungen ergibt sich im Umgang mit lizenzpflichtiger Software. Die Nutzung von Lizenzservern in großen, nationalen Grid-Umgebungen verbietet sich nicht nur aus Kostengründen, sondern auch aus rein praktischen Überlegungen. Da die Grid-Ressourcen dezentral administriert werden und von verschiedenen Organisationen kontrolliert werden, sind die Abstimmungen und Koordination, zum Beispiel zu Firewall-Regelungen, entsprechend



schwierig. Daher können so genannte Floating Lizenzen nicht genutzt werden. Node-Locked, also an Hardware gebundene, Lizenzen können ebenfalls nicht genutzt werden, da die virtualisierten Rechner immer andere Netzwerkadressen und Hostnamen besitzen.

Die Lösung des Problems innerhalb von *PartnerGrid* besteht darin, für einen Eingabedatensatz einen kryptographisch gesicherten Fingerabdruck zu erstellen. Dieser kann von einer zentralen Instanz erstellt und abgerechnet werden. Es wird letztlich für die Simulation eines Eingabedatensatzes bezahlt. Hierfür ist jedoch die Integration der Lösung in die verwendeten Solver notwendig. Dies ist eine Aufgabe der Softwarehersteller.

## Komponentenbasierte Software-Architekturen

Als Schnittstelle zur nationalen *D-Grid* Infrastruktur dient die Integrationsplattform *RCE*. *RCE* (Remote Component Environment) ist eine komponentenbasierte, verteilte Integrationsplattform. *RCE* wurde auf Basis von *OSGi* entwickelt, dem De-facto-Standard für komponentenbasierte Software-Architekturen in Java. *OSGi* wurde ursprünglich für mobile Geräte entwickelt und ist dementsprechend leichtgewichtig. Eine solche komponentenbasierte Applikation beinhaltet, neben den eigentlichen Komponenten, eine Grundplattform, die *OSGi*-Implementierung. Im Falle von *RCE* handelt es sich um das aus dem Eclipse-Projekt bekannte Equinox.

In diese Plattform können zur Laufzeit der Applikation einzelne Komponenten geladen, gestartet, beendet und ausgetauscht werden. Die Komponenten an sich können wiederum noch einmal durch Plug-Ins erweitert werden. Mit dieser Technik ist eine äußerst flexible Software-Architektur möglich geworden. Zudem bietet sie den Vorteil von Java: Plattformunabhängigkeit. Die Softwareplattform kann sowohl unter *UNIX* als auch unter *Microsoft Windows* eingesetzt werden.

*RCE* wird vom *Fraunhofer-Institut für Algorithmen und Wissenschaftliches Rechnen* und dem *Deutschen Zentrum für*

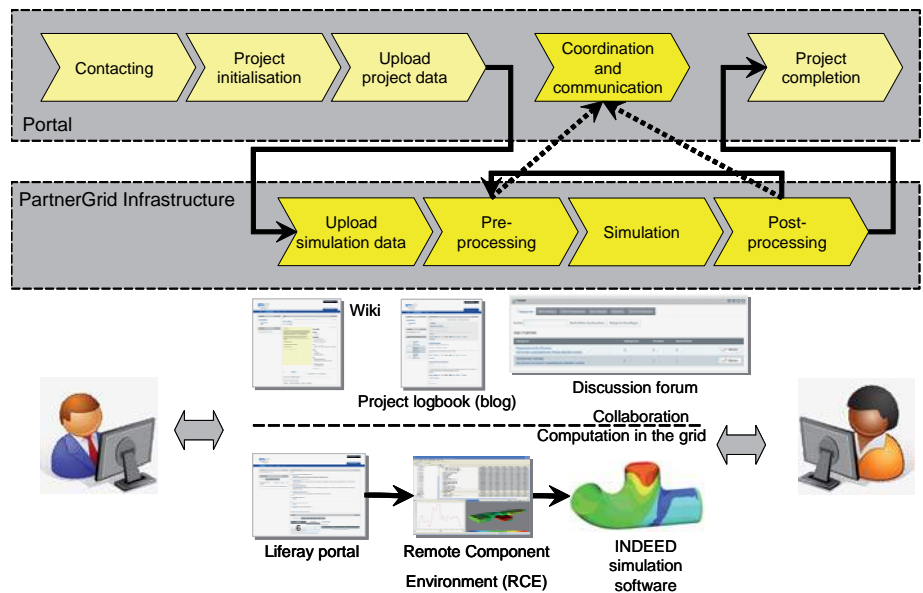


Abb. 1: PartnerGrid Infrastruktur: Kollaborationsportal und Berechnung im Grid. (Quelle: Fraunhofer IA0)

*Luft- und Raumfahrt e.V.* entwickelt.

## Internetportale

Der einfache Zugriff auf die Grid-Ressourcen geschieht im *PartnerGrid* über eine Portalwebseite. Ein Portal ist nicht einfach nur eine Webseite. Es besteht aus einem Portalcontainer und aus mehreren, frei definierbaren Portlets. Alle Teile sind in Java implementiert und ebenfalls portabel.

Portlets sind Software-Module die einzelne Funktionen oder Gruppen von Funktionen implementieren. Sie nutzen zur Darstellung HTML-Fragmente und werden vom Portalcontainer in Fenstern dargestellt, die sich frei auf der Webseite positionieren lassen. Eine Portalwebseite besteht demnach aus den HTML-Fragmenten vom Portalcontainer selbst, sowie von den verwalteten, dargestellten Portlets. Eine Anfrage an das Portal, z.B. ein Click auf einen Knopf in einem Portletfenster, wird zuerst vom Portalcontainer entgegengenommen und dann an das Portlet weitergeleitet.

Bei der im Projekt verwendeten Portalsoftware handelt es sich um *Liferay*. *Liferay* bringt bereits eine umfangreiche

Sammlung von Portlets mit, die bei der Gestaltung des *PartnerGrid*-Portals genutzt werden konnten und umfangreiche Funktionen zur Kooperation bereitstellen. Durch die Portaltechnologie lassen sich elegante, dynamische Webseiten gestalten.

## Ausblick

*GNS mbH* wird ein Portal auf Basis der im *PartnerGrid* erarbeiteten Architektur aufbauen. *GNS Systems GmbH* wird die erstellte Portallösung rund um die Integrationsplattform *RCE* und das *Liferay*-Portal im Förderprojekt *PT-Grid* weiterentwickeln.

Ich möchte mich an dieser Stelle für die bisherige, hervorragende Zusammenarbeit mit den Forschungspartnern bedanken.

Jan Niemann, *GNS Systems GmbH*,  
jan.niemann@gns-systems.de

## Termine

### ➔ Automotive Grand Challenge

Datum: 30.-31.03.2010  
Veranstaltungsort: Hanau  
Teilnehmer: GNS mbH

### ➔ International Supercomputing Conference 2010 (ISC)

Datum: 31.05.-03.06.2010  
Veranstaltungsort: Hamburg  
Teilnehmer: GNS Systems GmbH als Mitaussteller von Microsoft GmbH

### ➔ SIMVEC 2010

Berechnung und Simulation im Fahrzeugbau

Datum: 16.-17.11.2010  
Veranstaltungsort: Baden-Baden  
Teilnehmer: GNS Systems GmbH & GNS mbH



### GNS mbH

Am Gaußberg 2  
38114 Braunschweig  
Telefon: 05 31-8 01 12 0  
Fax: 05 31-8 01 12 79  
www.gns-mbh.com



### GNS Systems GmbH

Am Gaußberg 2  
38114 Braunschweig  
Telefon: 05 31-1 23 87 0  
Fax: 05 31-1 23 87 11  
www.gns-systems.de

### Impressum

Ausgabe 1/2010  
Erscheinungstermin: März 2010  
Herausgeber: GNS Systems GmbH

Verantwortlich: Jan Martini  
Redaktion: Anette Tröger  
Layout: Anette Tröger

Alle Rechte vorbehalten.  
Vervielfältigung, auch auszugsweise, nur mit schriftlicher Genehmigung des Herausgebers.  
Alle Markennamen sind in der Regel eingetragene Warenzeichen der entsprechenden Hersteller oder Organisationen.

## Das Speichern von Einstellungen in Animator4

Es gibt in der aktuellen Version des *Animator4* eine große Anzahl von Einstellungen, die gesetzt werden können, um die tägliche Arbeit zu vereinfachen. Angefangen bei Einstellungen, welche die einzelnen Views betreffen, über die Schriftauswahl und die diversen Farbeinstellungen bis hin zu Einstellungen, die das Verhalten der Oberfläche beeinflussen. All diese Einstellungen sind im Options-Dialog (Settings->Global Options) zu finden (Abb. 1). Dieser Dialog ermöglicht es, die zahlreichen Einstellungen komfortabel zu modifizieren.

Hinter allen Einstellungen, die in diesem Dialog vorgenommen werden können,

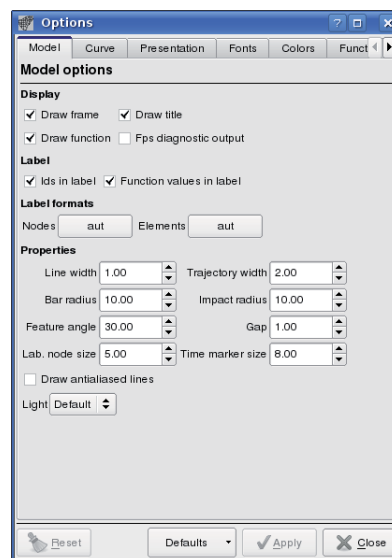


Abb. 1: Options-Dialog

nen, gibt es Kommandos, die ausgeführt werden, sobald man eine Einstellung modifiziert.

Es besteht die Möglichkeit, die vorgenommenen Einstellungen als Default abzuspeichern. So lässt sich *Animator4* schnell und unkompliziert den eigenen Bedürfnissen anpassen. Beim Abspeichern der Defaults wird das Kommando `wri ses opt` `~/ .a4dir/Sessions/options.ses` abgesetzt. Die Datei `options.ses`

wird von *Animator4* beim Programmstart automatisch gelesen. Es ist aber auch möglich, die Einstellungen in einer beliebigen Datei zu speichern, um ggf. auch mehrere Sätze projektspezifischer Einstellungen verwenden zu können. Ebenso können Einstellungen, die in dem Sessionfile stehen, auch in ein `a4_default.ses` übernommen werden.

Zu beachten ist dabei, dass seit der Version 1.2.0 eine zunehmende Anzahl von Einstellungen speziell für einzelne Views gesetzt werden können. Ein Beispiel ist die Hintergrundfarbe, die mit dem Kommando `col bac` gesetzt wird. Um die einzelnen Views anzusprechen, muss das Kommando-Ziel angegeben werden. Möchte man z.B. die Hintergrundfarbe des aktuellen Views auf Weiß setzen, so nutzt man das Kommando `col bac white`, welches implizit zu `v[act]:col bac white` übersetzt wird. Einzelne Views lassen sich über ihre Namen und Nummer ansprechen: `v[1]` bzw. `v[Model]`. Über diese Kommando-Ziele ist es möglich, bestehende Views anzupassen. Einstellungen für die später erst noch zu erstellenden Views werden mit dem Kommandoziel `std` gesetzt. Am Beispiel der Hintergrundfarbe: `v[std]:col bac white` bewirkt, dass alle nach diesem Kommando geöffneten Fenster die Hintergrundfarbe Weiß haben. Führt man die Änderungen über den Dialog durch, so geht man davon aus, dass die eingestellte Farbe für alle existierenden und neu erstellten Views gelten soll. Somit lautet das Kommando von dort: `v[std]v[all]:col bac white`. An diesem Beispiel erkennt man, dass die Kommando-Ziele kombiniert werden können.

Jedes Kommando (auch jenseits der Einstellungen) unterstützt eine gewisse Anzahl von Zielen. Analog zu `v` für Views gibt es noch `p` für Präsentationen und `s` für Slots. Neben den erwähnten Zielen `act`, `all` und `std` existieren auch noch die Ziele `first` und `last`, mit denen man den zuerst bzw. zuletzt erstellten View (Presentation oder Slot) adressieren kann.

Haiko Etzel, GNS mbH  
etzel@gns-mbh.com