Wissenschaftliches Rechnen in der Lehre am Beispiel des Studienprojekts „Computational Steering - der virtuelle Windkanal“

Martin Bernreuther
Martin.Bernreuther@ipvs.uni-stuttgart.de
Institut für Parallele und Verteilte Systeme, Abteilung Simulation großer Systeme
Universität Stuttgart, Universitätsstr. 38, 70569 Stuttgart

Hans-Joachim Bungartz
bungartz@in.tum.de
Institut für Informatik
Technische Universität München, Boltzmannstr. 3, 85748 Garching

Abstract

The recently finished project „Computational Steering - the virtual wind tunnel“ is a team work of nine software engineering students, who worked one year to develop a software system for the simulation of wind tunnel tests in a virtual environment. A special feature of the package is the simulation steering capability, where not only an online visualization of the CFD simulation results is provided, but also the possiblity to interact with the simulation during a run in a Virtual Reality environment. To achieve this goal an intense use of special HPC and VR hardware is indispensable. The software runs on a distributed system of parallel architectures and was realized on the department’s Linux CoW „Mozart“, an SGI Onyx multiprocessor visualization system driving a Powerwall, and a tracking system for user input. The development process itself is based on software engineering methods while the student project imitates all phases of a commercial software production process.

1 Einleitung

Die Abteilung „Simulation großer Systeme“ (SgS) des Instituts für Parallele und Verteilte Systeme (IPVS) der Universität Stuttgart ist in der Lehre hauptsächlich im Themenbereich des wissenschaftlichen Rechnens vertreten. Das Studienprojekt „Computational Steering - der virtuelle Windkanal“ war eine einjährige Gruppenarbeit [4] von neun Studenten der Softwaretechnik, welche hier viele Aspekte in sich vereint.

2 Studienprojekte

Der Studienplan des Diplomstudiengangs „Softwaretechnik“ an der Universität Stuttgart sieht zwei Studienprojekte vor, wobei das erste (SP A) innerhalb der Fakultät Informatik, das zweite (SP B) im Anwendungsfach abgeleistet wird [1]. Die dem Studienprojekt zugeordneten und über ein Jahr verteilten 16 SWS zerfallen in 4 bis 5 SWS Vorlesungen, 2 SWS Seminar und 9 bis 10 SWS für den praktischen Teil und werden entsprechend mit den Notenwichtungen $3 : 2 : 5$ bewertet. Ein SP beinhaltet ein vollständiges Softwareprojekt mit Konzeptions-, Realisierungs- und Montagephase. Neben der Entwicklungsarbeit kommen auf die Projektbearbeiter auch Management und Qualitätssicherungsaufgaben zu. Das betreuende Institut stellt neben Betreuer und Prüfer auch einen unabhängigen Kunden, welcher die Einhaltung von Terminen und eine hohe Qualität der Arbeit fordert. Neben dem Softwaresystem wird auch eine Dokumentation verlangt. Ein Projektbericht schildert die Leistung der einzelnen Bearbeiter. Die Ziele und Anforderungen an das Endprodukt werden in einer vom Kunden erstellten Ausschreibung zusammengefasst. Das Team erstellt hierfür im Vorprojekt ein Angebot und einen Prototypen. Zu Beginn des Hauptprojekts legt eine Spezifikation exakt fest, was realisiert wird. Eine Präzisierung der Anforderungen erfolgt auf Basis von Kundengesprächen und wird nach einem Review vom Kunden geprüft. Erst danach schließt sich die Entwurfs- und Implementierungsphase an. Die Abnahme wird nach einer Testphase fällig.

Ziel des SP „Computational Steering - der virtuelle Windkanal“ war es, einen virtuellen Windkanal nach Vorbild von [2] zu schaffen. Neben den Vorlesungen der Vertiefungslinie „Modellbildung und Simulation“ wurde hierfür speziell eine Vorlesung „Lattice Boltzmann Verfahren“ angeboten, um die Projektteilnehmer auf die Implementierung des Strömungssimulationsprogramms vorzubereiten. Auch das zum SP angebotene Seminar war speziell auf die Aufgabenstellung zugeschnitten. In den Vorträgen sollte ein direkter Bezug zum Projekt hergestellt werden. Die Einarbeitung in eines der Spezialgebiete bedeutete eine simultane Vorbereitung auf Aufgabenstellungen im Konzeptions- und Entwicklungsprozess. Das Seminar wurde deshalb zu Beginn des Projekts abgehalten. Ein Angebot mit Zeitplan und Meilensteinen wurde Ende Mai 2004 abgegeben. Nach eigenen Angaben stecken nach Fertigstellung jetzt etwa 600 Personentage im Projekt.

3 Modelle

Das Studienprojekt deckt nicht nur alle Phasen eines Softwareprojektes ab, sondern auch alle wesentlichen Aufgabenbereiche einer numerischen Simulation. Vor der Berechnung steht hier die Modellierung, wobei für das im Windkanal zu untersuchende Objekt unterschiedliche Repräsentationsformen generiert werden. Zunächst einmal wird von einem geometrischen bzw. Volumenmodell ausgegangen (vgl. Abschn. 4.2), wie es z. B. mit Hilfe eines CAD-Systems erstellt wird und als Datei in einem standardisierten Format vorliegt. Um das Objekt effizient visualisieren zu können (vgl. Abschn. 4.3), wird aus dieser Darstellungsform eine Oberflächentriangulation generiert. Für die Strömungsberechnung muss das Ob-

jekt bzw. die Domäne in einer diskretisierten Form vorliegen (vgl. Abschn. 4.4). Auch hier kann ein geeignetes Modell automatisch aus dem vorliegenden CAD-Volumenmodell erzeugt werden. Änderungen sollen ausschließlich am Volumenmodell vorgenommen und die betroffenen Visualisierungs- und Simulationsteilmodelle automatisch synchronisiert werden.

4 Programmstruktur

4.1 Übersicht

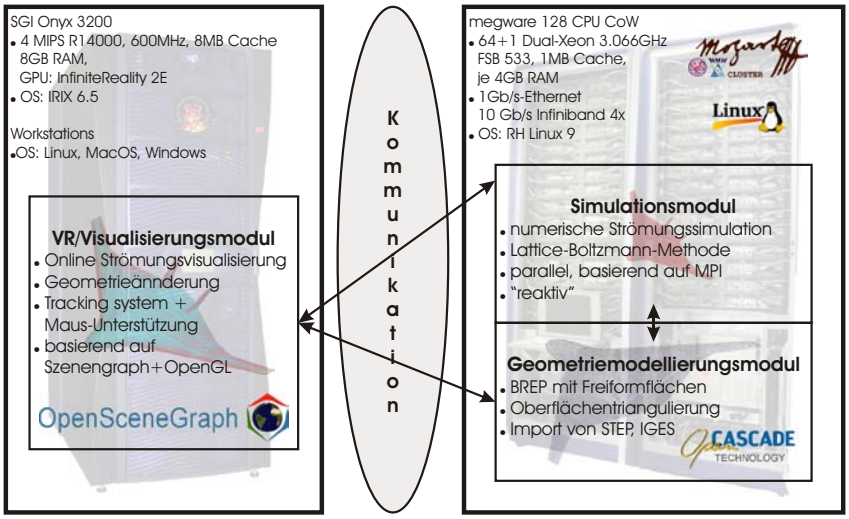


Abbildung 1: Programmmodule

Das im Studienprojekt „Computational Steering – der virtuelle Windkanal“ geschaffene verteilte Softwaresystem mit Namen „SToRM“ enthält nach Angaben der Bearbeiter aus 25225 Programmzeilen Quellcode (LOC) und basiert ausschließlich auf frei erhältlichen Bibliotheken. Das Einbinden bereits existierender Programmcodes ist wichtiger Bestandteil für eine effiziente Softwareentwicklung. Das Softwaresystem besteht aus 3+1 Komponenten (s. Abb. 1), welche im Folgenden vorgestellt werden. Es wird als Open-source Projekt unter

`http://ipvs.informatik.uni-stuttgart.de/SGS/projekte/storm/`

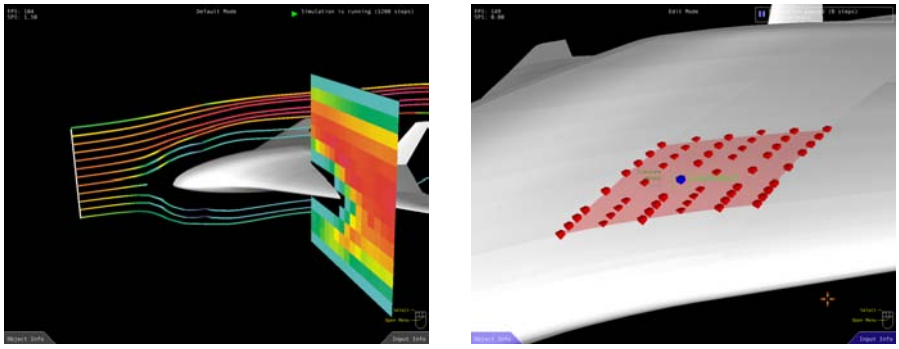
weitergeführt.

4.2 Geometriemodellierungsmodul

Das Erstellen eines leistungsfähigen Modellierkerns mit Unterstützung von Starrkörpern mit NURBS-Freiformoberflächen würde den Umfang des Arbeitsaufwands für das SP sprengen. Zudem ist die Möglichkeit des Imports standardisierter Formate wie STEP und IGES

gefordert. Statt einer Eigenentwicklung wird der frei verfügbare Modellierkern „OpenCASCADE“[1] (OCC) eingebunden, der die gestellten Anforderungen bereits erfüllt. OCC basiert auf einer BREP-Darstellung, und die Bibliothek beinhaltet eine Oberflächentriangulierung, welche zur Konvertierung des Modells in das Visualisierungsmodell nach Abschn. 3 verwendet wird. Zur Konvertierung in ein Modell zur Strömungssimulation nach Abschn. 4.4 wird ein Oktalbaum-basiertes Verfahren nach [5] eingesetzt. Das Teilmodul basiert auf dem in dieser Arbeit erstellten Programm[2] und liefert eine Voxel-Normzellenzerlegung. Das gesamte Modul schlägt mit insgesamt 6541 LOC zu Buche.

4.3 VR-Modul



(a) Strömungsvisualisierung (b) Editieren der Geometrie mit Hilfe von Kontrollpunkten

Abbildung 2: Benutzerschnittstelle

Das Virtual-Reality-Modul dient zur Visualisierung und stellt die Benutzerschnittstelle dar. Das umströmte Objekt liegt im Geometriemodellierungsmodul als trianguliertes Modell vor und wird über das Netzwerk übertragen. Die Daten der Strömungssimulation hingegen sind gitterbasiert. Der Visualisierung liegt die Bibliothek „OpenSceneGraph“[3] zugrunde. Einzelne Visualisierungsobjekte können so im Szenengraph strukturiert gespeichert werden. Neben dem umströmten Objekt dienen Schnittebenen zur Darstellung skalarer Werte wie Dichte, Druck oder Geschwindigkeitsbetrag. Zudem können Stromlinien angezeigt werden (s. Abb. 2(a)). Die Steuerung der Simulation erfolgt durch Geometrieänderungen am Objekt während der Simulation. Hierzu können NURBS-Kontrollpunkte verschoben werden (s. Abb. 2(b)). Das Programm wird mit Hilfe eines Flysticks bedient, dessen Position über ein Trackingsystem aufgenommen, berechnet und von einem Server bereitgestellt wird. Die Quadbuffer-Stereovisualisierung lässt einen räumlichen Eindruck und insgesamt einen immersiven VR-Eindruck entstehen. Alternativ kann das Programm auch mit Hilfe

[1] `http://www.opencascade.org/`
[2] `http://cad2octree.sourceforge.net/`
[3] `http://www.openscenegraph.org/`

der Maus bedient und ohne Stereovisualisierung betrieben werden. Die Visualisierung ist mit 9748 LOC das größte Modul innerhalb des Projekts.

4.4 Simulationsmodul

Eine Vorlage für das zu erstellende parallele Simulationsprogramm auf Basis einer Lattice-Boltzmann-Methode (LBM) war durch [6] vorhanden. Das Programm muss reaktiv im Hinblick auf interaktive Änderungen und Ergebnisanforderungen sein und durch eine effiziente Nutzung der Hardwareresourcen die Ausführungsgeschwindigkeit optimieren. Die LBM eignet sich besonders für eine parallele Implementierung, und das uniforme Gitter macht die automatische Generierung der Hinderniszellen einfach. Es wird der Kollisionsoperator nach Bhatnagar, Gross und Krook (BGK) verwendet. Die Neuentwicklung sollte die in der Vorlesung vorgestellten Algorithmen und Parallelisierungstechniken umsetzen. Die angestrebte Effizienz wurde von der Implementierung aber nicht vollständig erreicht. Für das Projekt war das Vorhandensein eines abteilungseigenen CoW im Interaktivbetrieb unerlässlich. Der klassische Batch-Betrieb ist für diese Art Anwendung nicht geeignet. Der Simulationscode ist mit 3539 LOC vergleichsweise schlank. Eine Einbindung eines weiteren (kommerziellen) Simulationsprogramms musste aus Zeitgründen entfallen.

4.5 Kommunikation der verteilten Anwendung

Nach einer Prüfung von Middleware-Plattformen wie z. B. CORBA fiel die Entscheidung auf eine einfache direkte Kommunikation mit Hilfe von Sockets. Neben Kommandos auf ASCII-Basis sind auch binäre Ströme implementiert, um eine effiziente Übertragung größerer Datenmengen zu gewährleisten. Die eingesetzten Plattformen machen es unerlässlich, auch Konvertierungen zwischen verschiedenen Repräsentationen (z. B. „little endian“ für den PC-CoW und „big endian“ für die SGI Onyx) zu berücksichtigen. Das Framework zur Kommunikation zwischen verteilten Systemen erforderte 5397 LOC.

5 Bewertung

Das vorgestellte Studienprojekt war das Erste im Bereich des wissenschaftlichen Rechnens an der Universität Stuttgart. Die Verwendung von Methoden der Softwaretechnik ist für die Erstellung von Softwaresystemen für die numerische Simulation immer noch nicht selbstverständlich, und entsprechend werden viele Projekte für Softwaresysteme im wissenschaftlichen Rechnen gänzlich ohne Softwaretechnik-Methodik durchgeführt. Nicht zuletzt deswegen wird bereits eine Software-Krise im Bereich der rechnergestützten Ingenieurwissenschaften prognostiziert [3]. Das Verbinden dieser Bereiche kann als Herausforderung angesehen werden. Die Studenten waren zu Beginn des Projekts in der Mehrzahl weder numerisch besonders vorgebildet noch interessiert und wählten die Themenstellung aufgrund der Verfügbarkeit teilweise nur als Notlösung. Der Schwerpunkt der Anforderungen war weniger auf optimale Lösungen für einzelne Komponenten als auf die Gesamtarchitektur und das Systemdesign gelegt, und diese Aufgabe wurde hervorragend gelöst. Außerdem

war ja auch der Weg selbst ein Ziel, sodass das Gesamtergebnis des Projekts sehr positiv zu bewerten ist. Das Projekt hat Modellcharakter für weitere Projekte wie z. B. ein aktuell in der „Bavarian Graduate School of Computational Engineering“[4] durchgeführtes Projekt im Bereich Molekulardynamik.

6 Zusammenfassung

Es wurde ein Studienprojekt vorgestellt, das in besonderer Weise diverse Aspekte des Wissenschaftlichen Rechnens wie z. B. Modellierung, numerische Algorithmen und Visualisierung verbindet. Die eingesetzte Hardware sind verteilte parallele Architekturen. Die Simulation wird hierbei auf einem Hochleistungsrechner ausgeführt und erfordert aufgrund des hohen Rechenaufwands in besonderer Weise eine effiziente Numerik und Implementierung. Die resultierende Software beinhaltet eine Reihe praktischer Umsetzungen von Themen, welche in Vorlesungen sonst meist nur theoretisch erläutert werden. Die Gruppengröße und Dauer des Projekts lässt einen großen Arbeitsumfang zu, erfordert aber ein Projektmanagement, das als zusätzliche Aufgabe gemeistert werden muss.

Literatur

[1] *Ludewig, J. (Hrsg.):* Praktische Lehrveranstaltungen im Studiengang Softwaretechnik: Programmierkurs, Software-Praktikum, Studienprojekte, Fachstudie. Bericht der Fakultät Informatik, Universität Stuttgart, 2. Auflage, 2001.

[2] *Bryson, S., Levit, C.:* The Virtual Wind Tunnel. IEEE Computer Graphics and Applications 12(4), 1992.

[3] *President's Information Technology Advisory Committee:* Computational Science: Ensuring America's Competitiveness. Report to the president, Executive Office of the President of the United States, June 2005.[5]

[4] *Hohloch, R., Jung, B., Markovic, O., Nausedat, J., Prokharau, M., Rathgeber, T., Rössler, D., Spenrath, C., Walter, T.:* Studienprojekt „Computational Steering - der virtuelle Windkanal“. Universität Stuttgart, 2005.

[5] *Mahler, S.:* Erzeugung und Evaluierung von Oktalbaumstrukturen als Schnittstelle zu CAD-Programmen. Diplomarbeit Nr. 2035, Fakultät Informatik, Universität Stuttgart, 2003.

[6] *Kaiser, F.:* Entwicklung eines modularen Softwarepaketes für reaktive massiv parallele dreidimensionale Lattice-Boltzmann Strömungsberechnungen. Diplomarbeit Nr. 2169, Fakultät Informatik, Universität Stuttgart, 2004.

[4] `http://www.bgce.de/`

[5] `http://www.nitrd.gov/pitac/reports/20050609_computational/computational.pdf`