Lastverteilung in Workstation-Netzen

Wolfgang Becker

Dieser Artikel stellt das HiCon Projekt vor, in dem interessante Konzepte zur dynamischen Lastbalancierung paralleler und konkurrierender Anwendungen in Workstation-Netzen entwickelt wurden. Die Implementierung des Lastbalancierungsansatzes wird anhand realer komplexer Anwendungen validiert.

Einleitung

Netze bestehend aus leistungsfähigen Workstations dienen nicht nur als Rechenkapazität für einzelne Anwender, sondern können darüber hinaus als ein großes, paralleles Rechnersystem betrachtet werden. Workstations in Clustern werden üblicherweise nicht alle über die volle Zeit genutzt, sondern im Durchschnitt sind 90% der Maschinen unbeschäftigt. Dieses hohe Potential an ungenutzter Rechenleistung kann sowohl genutzt werden um andere, momentan überlastete Workstations zu entlasten, indem sie Aufträge abgeben, oder um rechenintensive Anwendungen parallel auf dem Rechnernetz ablaufen zu lassen. Automatische Lastbalancierung ist notwendig, um die Aufträge in Workstation-Netzen so zu verteilen, daß die Ressourcen optimal genutzt werden, ohne die Anwendungsprogrammierer oder Benutzer mit der zusätzlichen Komplexität der effizienten Nutzung paralleler Systeme zu belasten.

Um die Last verschiedener konkurrierender Anwendungen im System optimal zu balancieren, müssen die Anwendungen so verteilt bzw. verlagert werden, daß jeder Rechner entsprechend seiner Leistung und sonstigen Ressourcen (Hauptspeicher etc.) belastet wird. Bei datenintensiven Anwendungen müssen zusätzlich die Kosten für Zugriffe auf entfernte globale Daten, wie z.B. Relationen in Datenbanken, berücksichtigt werden. Beim Lastausgleich für parallele Anwendungen sind die Abhängigkeiten zwischen den Aufträgen sowie der Aufwand für Datenkommunikation zwischen den Rechnern einzubeziehen.

Statische Ressourcenplanung ist aufgrund der dezentralen Verwaltung, wegen der Vielzahl kurzfristig entstehender Lasten und der in vielen Anwendungen frühestens während der Laufzeit absehbaren Lastprofile nicht ausreichend. Dynamische Lastbalancierung ist in der Lage, zur Laufzeit die aktuelle Systemlast zu berücksichtigen. Da sie jedoch zur Laufzeit für die Balancierung selbst Ressourcen benötigt und Verzögerungen verursacht, muß sich dynamische Lastbalancierung auf einfache Heuristiken beschränken.

Im HiCon Projekt wurde ein dynamischer Ansatz zur automatischen Lastbalancierung entwickelt, der auf Workstation-Netze und Gemische großer, paralleler Anwendungen nach dem Client - Server Ablaufmodell zugeschnitten ist. Da Workstation-Netze auf grund ihrer hohen Rechenleistung bei geringen Kosten in Entwicklungs-, Verwaltungs- und Produktionsumgebungen zunehmend an Bedeutung gewinnen, nimmt der Anteil großer, wiederholt laufender Anwendungen zu, und die Lastprofile im System variieren weniger stark.

Das HiCon-Modell gehört zur Klasse der dynamischen Verfahren, da die Entscheidungen, auf aktuellen Lastinformationen basierend, zur Laufzeit getroffen werden. Da heterogene Workstation-Netze betrachtet werden, ist keine Migration laufender Prozesse, sondern nur Zuweisung und Migration von Aufträgen vor Beginn ihrer Bearbeitung möglich. Neben Rechenleistungen, Auftragslasten und CPU-Nutzung werden auch Größen wie Kommunikation und Datenverteilung berücksichtigt. Globale persistente Daten sowie gemeinsame Daten innerhalb paralleler Anwendungen können durch ein Laufzeitsystem im verteilten System migriert und repliziert werden. Das HiCon-Modell stellt einen anwendungsunabhängigen Betriebssystemdienst dar, der nicht nur einzelne, sondern mehrere verschiedene heterogene Applikationen mit dem Ziel eines maximalen Gesamtdurchsatzes balancieren kann. Die Lastinformationsverwaltung und Entscheidungskomponente kann zentral organisiert werden, für große Systeme ist eine zentrale Verwaltung pro Cluster mit dezentraler Kooperation zwischen den Clustern möglich.

Abb.1 zeigt die Gesamtstruktur eines Clusters mit Hardware, Anwendungen und Lastbalancierungskomponenten. Die Anwendungen bestehen aus je einem Client, der den Ablauf koordiniert und Aufträge generiert, sowie aus Servern, die bestimmte Teilfunktionen von Anwendungen übernehmen können. Es können beliebig viele Server jeder Klasse auf das System verteilt werden, wodurch Parallelität genutzt werden kann.

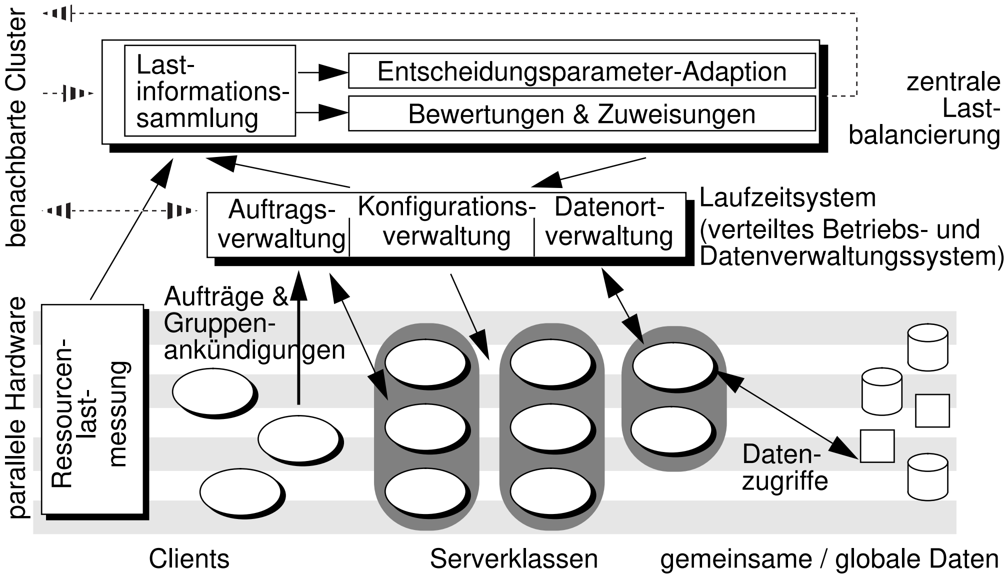

Abb.1: Skizze des Gesamtsystems im HiCon-Modell zur dynamischen Lastbalancierung.

Die folgenden Kapitel stellen einzelne Konzepte genauer vor, für umfassende Beschreibungen muß jedoch auf [1, 2, 5] verwiesen. Das vorletzte Kapitel stellt einige realisierte Anwendungen vor und präsentiert Messungen zur Bewertung des Lastbalancierungsansatzes.

Unterstützung der Lastbalancierung durch Vorabschätzungen

Für große, relevante Anwendungen lohnt sich der Aufwand einer effizienten Parallelisierung. Dabei oder zusätzlich dazu können Abschätzungen über das Zusammenspiel der Teilaufträge sowie die Lastprofile der Einzelaufträge untersucht werden. Sie sind zwar gewöhnlich nicht statisch vorhersehbar, aber im Verlaufe der Anwendung können beim Absenden eines Auftrags oft eine realistische Abschätzung über den Rechenaufwand (zumindest relativ zu anderen Aufträgen der Anwendung) und Abschätzungen über die vermutlich benötigten Datenbereiche gemacht werden. Etwas aufwendiger ist es, bei dynamisch strukturierten parallelen Anwendungen Abschätzungen über die

Abhängigkeiten innerhalb kleiner Gruppen in naher Zukunft anstehender Aufträge zur Laufzeit zu erstellen.

Das HiCon-Modell bietet den Clients zur Laufzeit die Möglichkeit, beim Absenden von Aufträgen deren Rechenaufwand (Instruktionen, Anteil an Fließkomma-Operationen) und Datenreferenzen (wenige Bereiche globaler Datensätze, Wahrscheinlichkeiten für exklusiven Zugriff) abzuschätzen. Die Lastbalancierung kann diese Angaben für Zuweisungsentscheidungen verwenden. Da solche Angaben in der Praxis durch Abhängigkeit von den Eingabedaten, von der Problemgröße und durch Fehler in der Abschätzungsheuristik um Faktoren bis zu 1000 von den realen Profilen abweichen, verfügt die HiCon-Lastbalancierung über eine Adaptionskomponente, die durch längerfristige Rückkopplung aufgrund real beobachtetem Aufwand Korrekturfaktoren für die Auftragsgrößenabschätzungen nachregelt (je Serverklasse und je Auftragstyp). Auch Datenreferenzabschätzungen können adaptiv korrigiert werden, was allerdings etwas diffiziler ist (siehe unten). Bei Vorankündigungen kleiner Auftragsgruppen ermittelt die Lastbalancierung Prioritäten für die Aufträge aufgrund kritischer Pfade im Abhängigkeitsgraphen, die später bei Zuweisungen berücksichtigt werden [4].

Globale Lastkontrolle

In nahezu allen realisierten Lastbalancierungsansätzen werden entstehende Prozesse sofort irgendwo im System gestartet. Die Lastbalancierung kann dann zwar beim Auftragsstart einen Prozessor auswählen und in manchen Ansätzen sogar laufende Prozesse auf minderbelastete Knoten migrieren, aber Überlastungssituationen des gesamten Systems, in denen wegen Speicherüberlastung, Prozeßwechseln und Netzüberlastung der Durchsatz stark sinkt, können nicht behoben werden.

Im HiCon-Modell verfügt die Lastbalancierungskomponente eines Clusters über eine Auftragswarteschlange, in die ankommende Aufträge eingereiht werden. Dort werden Aufträge bewertet und zu beliebigen Zeitpunkten an Server zugewiesen. Die Zuweisung geschieht nach dem first-in-first-out Prinzip, wobei Aufträge mit höheren Prioritäten (wegen angekündigten Folgeaufträgen) vorrangig zugewiesen werden. Die Lastbalancierungsstrategie ist bestrebt, sämtliche Prozessoren auf einen kurzen Zeitraum im voraus zu einem geeigneten Maß mit Aufträgen auszulasten. Dazu können Aufträge auch in lokalen Warteschlangen der Server gepuffert werden. Die zentrale Warteschlange verwirklicht auch die Idee der möglichst späten Zuweisung, was in dynamischen Ansätzen und bei Verzicht auf nachträglicher Migration aufgrund schnell wechselnder Lastsituationen wichtig ist [3].

Für eine sinnvolle Abschätzung der momentanen und erwarteten Knotenauslastungen ist die üblicherweise gemessene *run queue length* (mittlere Zahl laufbereiter Proesse in den letzten Sekunden) problematisch, da sie stark schwankt, sie keinen direkten Zusammenhang zu Start und Ende von Aufträgen erfaßt, und sie keine Aussage für die Zukunft erlaubt. Im HiCon-Ansatz wird diese Größe nur zur längerfristigen Adaption des relativen CPU-Bedarfs von Auftragstypen verwendet. Die momentane und in Kürze erwartete Last kann aufgrund der Summe der auf den Prozessoren (durch Server) laufenden bzw. wartenden Aufträgen, gewichtet nach dem mittleren CPU-Bedarf der Auftragstypen, akkurater ermittelt werden. Auftragslaufzeiten können aufgrund von Größen-Vorabschätzungen und vermuteten Datenkommunikationszeiten eingeschätzt werden.

Globale Durchsatzoptimierung

Zentrale Balancierungskomponenten haben den Vorzug, daß sie zur Plazierung von Aufträgen die gesamte Systeminformation und sämtliche anstehenden Aufträge einbeziehen können, und erreichen damit deutliche Leistungssteigerungen auch bei der Balancierung komplexer Anwendungen und Lastsituationen.

Im HiCon-Modell wird im Prinzip jeweils der bereits am längsten wartende bzw. höchstpriorisierte Auftrag dem Server zugewiesen, bei dem die kürzeste Antwortzeit erwartet wird. Die Antwortzeitabschätzung setzt sich jeweils aus drei Komponenten zusammen: Erster Summand ist die vermutliche Restarbeitszeit des Servers, da Server ihre Aufträge sequentiell in der Ankunftsreihenfolge abarbeiten. Die Restarbeitszeit wird aus den Antwortzeitabschätzungen früher zugewiesener Aufträge und der bisherigen Arbeitszeit des Servers an den Aufträgen gewonnen. Zweiter Summand ist die vermutliche Rechenzeit, wobei der Rechenaufwand des Auftrags durch die zum vermutlichem Startzeitpunkt des Auftrags auf dem Server verfügbare Rechenleistung dividiert wird. Der dritte Summand ist die vermutete Kommunikations- und Wartezeit für Zugriffe auf gemeinsame Daten, über die der Server momentan lokal nicht verfügt.

Die Auftragszuweisung optimiert also die Antwortzeit einzelner Aufträge und den Gesamtdurchsatz des Systems im Mehrbenutzerbetrieb, weil sie eine geeignete Auslastung der Prozessoren bewirkt, und berücksichtigt die Kommunikationsbeziehungen bzw. Datenaffinitäten in parallelen Programmen und in Anwendungen mit globalen Datenzugriffen. Die Durchsatzoptimierung beruht nicht auf einer *steady state* Annahme, sondern plant die Auslastung für die nahe Zukunft. Weiterhin enthält der Zuweisungsalgorithmus eine Konfliktauflösung im Sinne einer sozialen Lastbalancierung: obwohl im Prinzip der erste Auftrag an den Server zugewiesen wird, auf dem er alleine am schnellsten abgearbeitet wird, prüft er, ob nachfolgende Aufträge denselben Server als besten wünschen. Im Falle eines solchen Konflikts weicht der Auftrag auf den nächstbesten Server aus, wenn die Summe der Antwortzeiten beider Aufträge geringer eingeschätzt wird als umgekehrt.

Berücksichtigung des Kommunikationsaufwands

In lose gekoppelten Systemen, die keinen gemeinsamen Speicher oder Sekundärspeicher besitzen, sind der Kommunikationsaufwand innerhalb von parallelen Anwendungen und der Aufwand für entfernten Zugriff auf globale Daten relevante Größen. Das Laufzeitsystem im HiCon-Modell bietet den Anwendungs-Servern die Möglichkeit, auf konsistent auf gemeinsamen und auf globalen Daten zu operieren, wobei die Daten tatsächlich zwischen den Knoten ausgetauscht bzw. repliziert werden.

Für dynamische Lastbalancierung ist es hier wichtig, das richtige Verhältnis zwischen maximaler Ausnutzung der parallelen Rechenkapazität und den dabei entstehenden Datenkommunikationskosten zu erhalten. Die Lastbalancierung im HiCon-Modell weist daher Aufträge nicht den Servern zu, wo sie am schnellsten berechnet werden können, sondern den Servern, auf denen die Summe aus Rechenzeit und Datenkommunikationszeit minimal ist. Dadurch werden Server favorisiert, die einen großen Teil der im Auftrag benötigten Daten lokal verfügbar haben.

Clients können bei Absendung eines Auftrags zur Laufzeit Abschätzungen geben, welche Bereiche gemeinsamer oder globaler Daten gelesen oder modifiziert werden (siehe oben). Die Lastbalancierung kann aufgrund der Zugriffsprofile und der aktuellen Datenverteilung im System die Datenkommunikationskosten für die verschiedenen Server einschätzen. Die Kommunikationskosten pro Datentyp werden dabei adaptiv durch exponentielle Glättung aus den tatsächlich gemessenen Kosten der letzten Zeit bestimmt und berücksichtigen dadurch die aktuelle Netzbelastung sowie die aktuellen Datensatzgrößen. Da die Abschätzungen der Clients in der Praxis weder vollständig noch korrekt sind, versucht die Lastbalancierung pro Auftragstyp, durch Vergleich mit kürzlich beobachteten tatsächlichen Kosten, einen Korrekturfaktor nachzuregeln, der die Fehleinschätzungen auf Auftragsebene kompensiert.

Skalierbarkeit

Zentrale Lastbalancierung besitzt aufgrund des globalen Überblicks das größtmögliche Optimierungspotential. In dezentralen Verfahren muß Zustandsinformation überall hin kopiert werden, oder jede Balancierungskomponente verfügt nur über partielle Informationen, was zu insgesamt kontraproduktiven Entscheidungen führt und nur lokale Optimierung bewirkt. In großen Systemen und Situationen mit hohen Auftrags-

Ankunftsraten wird jedoch eine zentrale Balancierungskomponente zum Flaschenhals, weil ständig aktuelle Information von überall gesammelt werden muß und Balancierungsentscheidungen nicht mehr schnell genug gefällt werden können, was die Auftragsausführungen verzögert. Besonders in weitverteilten Systemen mit langsamen Netzverbindungen ist zentrale Lastbalancierung nicht effektiv. Das HiCon-Modell bietet daher die Möglichkeit, zentrale Balancierungsagenten je Cluster einzusetzen, die in beliebiger Topologie untereinander dezentral groben Lastausgleich durchführen. Innerhalb eines Clusters werden Aufträge sehr akkurat verteilt, während zwischen Clustern Aufträge aus den zentralen Warteschlangen ausgetauscht werden, wenn es für die einzelnen Aufträge günstig erscheint. Die Lastbalancierung betrachtet also benachbarte Cluster abstrahiert als Server, denen sie gleichermaßen Aufträge zuweisen kann. Lediglich die Lastinformationen und die Kostenabschätzungen sind unterschiedlich.

Effizienzsteigerung für ein breites Anwendungsspektrum

Der Lastbalancierungsansatz im HiCon-Modell kann aufgrund der Komplexität des Systems und der Balancierungstechniken nicht durch ein analytisches Modell beurteilt werden. Daher wurde er in Form einer prototypischen Realisierung für Workstation-Netze anhand verschiedener realer Anwendungen validiert. Als Anwendungen wurden typische Vertreter aus relevanten Problemklassen der Bildverarbeitung, der numerischen Simulation sowie der kommerziellen Datenverarbeitung gewählt. Im folgenden werden drei der Anwendungen kurz skizziert und das Potential für dynamische Lastbalancierung nach dem HiCon-Konzept durch Vergleichsmessungen ermittelt.

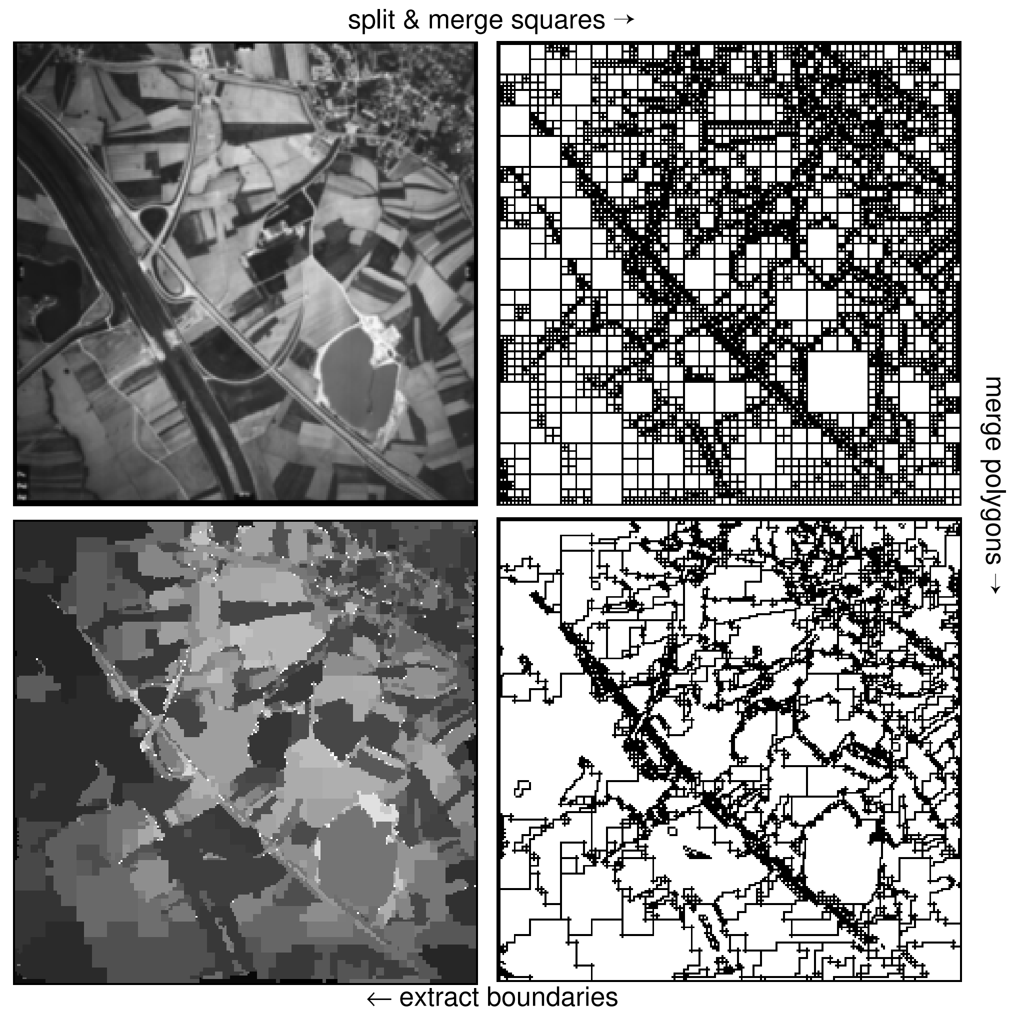

Abb2: Phasen der Flächenerkennung am Beispiel eines Luftbildes.

Parallele Flächensegmentierung
Aufgabe der Flächensegmentierung ist es, ein gegebenes Punktrasterbild in eine Menge homogener Flächen (Polygone) zu konvertieren. Dieser Vorgang ist gewöhnlich die erste Phase einer Bilderkennung. Eine Fläche soll farblich beieinander liegende Punkte mit einem kleinen Anteil von Ausnahmen enthalten. Der verwendete Algorithmus besteht aus vier Schritten (Abb.2): Ausgehend von einer initialen Rasterung wird versucht, benachbarte Quadrate zusammenzufassen (Square Merge), sofern das neue Quadrat eine homogene Fläche ergibt. Parallel dazu werden die Quadrate solange verfeinert (Square Split), bis jedes Quadrat eine homogene Fläche enthält. Danach werden soviel als möglich benachbarte Quadrate zu beliebigen Polygonen zusammengefaßt (Polygon Merge). In der letzten Phase werden die Kantenzüge berechnet, welche die Polygone umgeben (Boundary Trace). Abb.3 veranschaulicht das parallele Ablaufprinzip einer Flächensegmentierung.

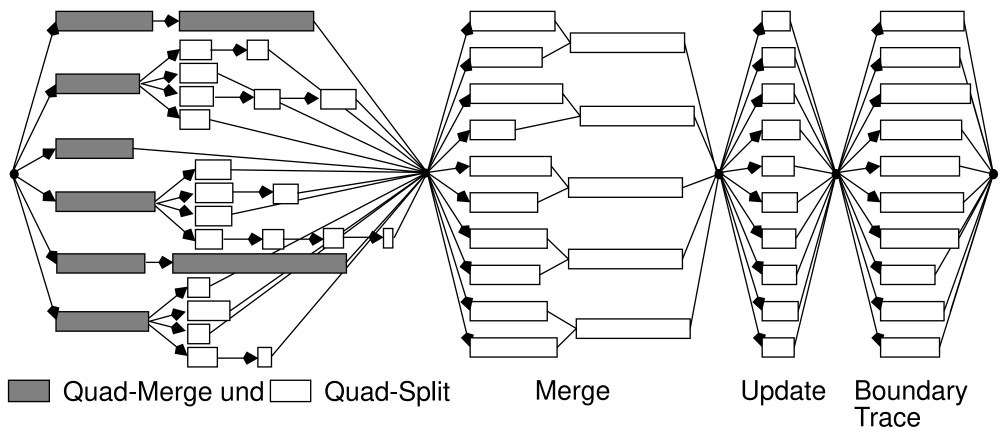

Abb.3: Prinzip des parallelen Ablaufs einer Flächenerkennung (Auftragstruktur).

Quad-Split-Operationen sind dabei meist sehr feingranular (einige Millisekunden). Es handelt sich hier um eine unstrukturiert parallele Anwendung und die Anzahl sowie das Granulat der Aufträge sind stark von der jeweiligen Bildstruktur abhängig. Die Anwendung besteht insgesamt aus mehreren aufwendigen Phasen mit sehr unterschiedlichen Profilen. Insgesamt und innerhalb mancher Phasen treten sehr unterschiedliche Auftragsgrößen auf. Die Aufträge verursachen häufige Datenzugriffe auf große Mengen gemeinsamer Daten.

Parallele Spannungsberechnung nach der Methode der finiten Elemente
Die Methode der finiten Elemente wird vor allem in der Mechanik und Thermodynamik angewandt, um das Verhalten von Konstruktionen und Körpern unter Einwirkung von Kräften und Temperatureinflüssen durch Simulation zu untersuchen. Die zu untersuchenden Gebilde werden geeignet in eine Vielzahl kleiner Elemente zerlegt. Innerhalb der Elemente und zwischen benachbarten Elementen werden die physikalischen Gesetze, die gewöhnlich durch Differentialgleichungen spezifiziert sind, mithilfe numerischer Näherungsverfahren berechnet. Dies führt insgesamt zu einem großen Gleichungssystem, das die Einflußgrößen der Einzelelemente aufeinander enthält. Die Lösung des Gleichungssystems ergibt einen Ergebnisvektor, der die Verschiebungen, Spannungen, Drücke oder Temperaturen der einzelnen Elemente des Gebildes wiedergibt.

Die Berechnungsformeln und der Berechnungsaufwand für die gegenseitigen Einflüsse der einzelnen Elemente im Gesamtsystem hängt sehr stark vom betrachteten physikalischen Problem, von der Struktur der Elemente, der Art der Randbedingungen und dem Grad der Ansatzfunktionen zur näherungsweisen Lösung der durch Differentialgleichungen beschriebenen Gleichgewichtsbedingungen ab. Hier wurde lediglich ein einfaches Problem mit einem einzigen Elementtyp und linearen Ansatzfunktionen betrachtet (Abb.4).

Die Lösung des großen Gleichungssystems erfolgt in parallelen Implementierungen üblicherweise durch das iterative Näherungsverfahren der konjugierten Gradienten. Die Gesamtberechnung besteht im wesentlichen aus einer Sequenz von 5 Phasen (Abb.5), von denen die rechenaufwendigen Phasen parallelisiert wurden.

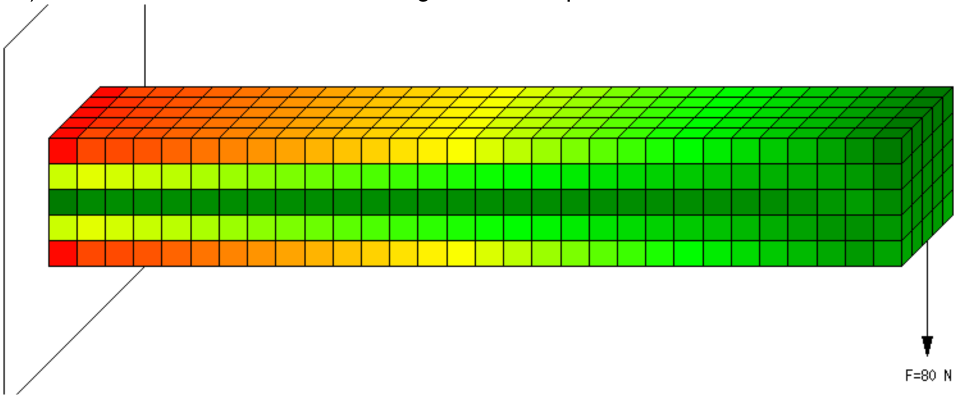

Abb.4: Ergebnis einer Spannungsberechnung eines einseitig fixierten, einseitig radial belasteten Stabes.

Die Ablaufstruktur ist vergleichsweise regulär. Dynamischer Lastausgleich ist vor allem wegen unterschiedlichem Aufwand in den Elementberechnungen und zur Berücksichtigung unterschiedlicher Rechenkapazitäten sowie Lastunterschiede durch Mehrbenutzerbetrieb erforderlich.

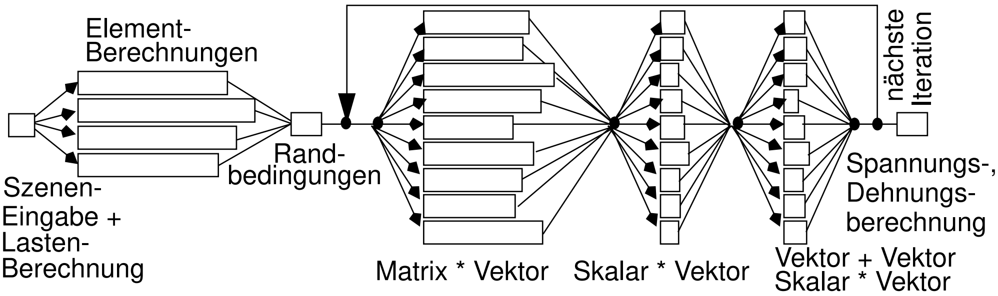

Abb.5: Prinzip des parallelen Ablaufs einer Berechnung nach der Methode der finiten Elemente (Auftragstruktur).

Parallele relationale Datenbankoperationen

In der kommerziellen Datenverarbeitung besteht ein großer Anteil des Rechenaufwands und des Ein-/Ausgabe-Aufwands im Auffinden, in der Verknüpfung und Filterung sowie Modifikation von Daten in einer Menge großer, inhaltlich korrelierter Datenbestände. In relationalen Datenbankverwaltungssystemen werden aufwendige Operationen auf großen Datenmengen in einer deskriptiven, mengenorientierten Sprache interaktiv oder in Anwendungsprogrammen formuliert. Diese Operationen werden automatisch in einen Satz sehr einfacher Grundoperationen konvertiert, die vom Datenbanksystem zur Laufzeit ausgeführt werden können.

Im HiCon-Projekt können durch einfache Beschreibungsdateien komplexe strukturierte Datenbankoperationen ausgeführt werden. Die Grundoperationen bestehen hier im wesentlichen aus folgenden vier Typen: die Scan-Operation durchsucht große Datenmengen (üblicherweise Dateien oder Relationen) nach Datensätzen, die bestimmte Bedingungen erfüllen. Die Projektion filtert aus großen Datenmengen die relevanten Eigenschaften (Attribute) der Datensätze heraus. Die Verbundoperation verknüpft verschiedene Datenmengen aufgrund eines bestimmten Kriteriums, d.h. bildet inhaltlich zusammengehörige Paare von Datensätzen. Die Ladeoperation fügt große Mengen neuer Datensätze in die Datenbank ein.

Die Grundoperationen innerhalb einer Anfrage hängen teilweise voneinander ab und tauschen große Mengen von Daten (Zwischenresultate) aus. Anfrageübersetzer und Optimierer zerlegen deskriptiv vorgegebene, komplexe Operationen funktional, und erzeugen eine Struktur von Aufträgen, die funktionale Parallelität und auch Datenparallelität innerhalb der Grundoperationen entfaltet. Zur Ausnutzung der Datenparallelität werden die Basisrelationen sowie die Zwischenergebnisse nach Attributwertebereichen auf verschiedene Dateien partitioniert. Reihenfolge-Abhängigkeitsgraphen sind in diesem Bereich meist baumstrukturiert.

Die parallele Ausführung solcher Operationen ergibt charakteristische Lastprofile für kommerzielle Datenverarbeitung, da sie als Basisdienste für verschiedenartige Anwendungen dienen. Für die Lastbalancierung solcher Profile können die Auftragsgraphen, meist mit Abschätzungen über die Größen und die Datenreferenzmuster der einzelnen Aufträge bereits zu Beginn der Bearbeitung der komplexen Operation angegeben werden, da sie von Anfrageübersetzern und Anfrageoptimieren generiert und abgeschätzt wurden. Statische Lastbalancierung zur Übersetzungszeit ist jedoch unzureichend, wenn sich die Größen der Basisrelationen und in Folge der Rechenaufwand der Aufträge im laufenden Betrieb ändern, wenn die Systemauslastung schwankt oder wenn sich die Lageorte der Daten im System dynamisch ändern.

Abb.6 zeigt links ein für Messungen verwendetes Szenario, das die relationale Anfrage $R_8 = ((\Pi_{a0}\ (\sigma_{a1>1300}R_2))\ x_{a0=a0}\ (\sigma_{a4>1200}\ ((\sigma_{a1>1250}\ R_0)\ x_{a0=a0}\ R_1)))$ berechnet, und rechts einen möglichen parallelen Ausführungsplan dazu.

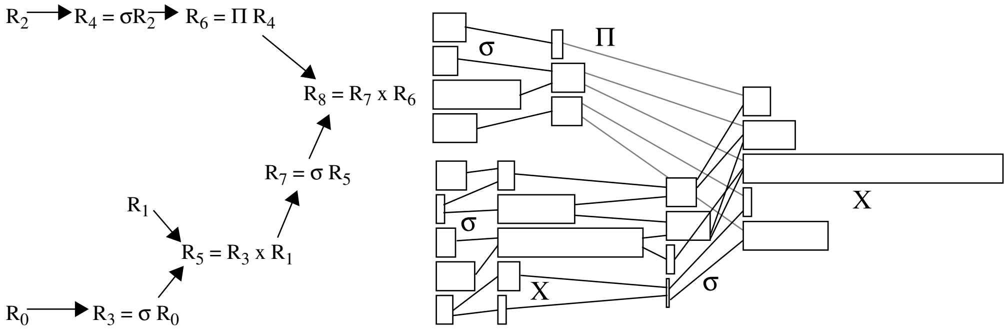

Abb.6: Operatorgraph und paralleler Ausführungsgraph einer relationalen Datenbankoperation.

Leistungsbewertung

Um die Fähigkeiten der Lastbalancierung zu evaluieren, wurden obige drei Anwendungen jeweils sowohl alleine als auch im Mehrbenutzerbetrieb unter HiCon-Lastbalancierung, mit simpler Lastbalancierung (zufällige Verteilung der Aufträge) und ohne Lastbalancierung beobachtet. Ohne Lastbalancierung läuft jede Anwendung sequentiell auf dem Knoten des Clients ab. Im Mehrbenutzerbetrieb wurde im Falle der Flächenerkennung und der Datenbankoperationen bereits eine recht günstige, d.h. balancierte Situation vorgegeben, da jeder Client, d.h. jede der Anwendungen, auf einem einzelnen Prozessor ablief. Die Prozessoren sind jedoch unterschiedlich leistungsfähig. Bei der finite Elemente Berechnung wurde im Mehrbenutzerbetrieb ein Lastungleichgewicht vorgegeben, indem nur auf dreien der fünf Rechner Anwendungen gestartet wurden. Als Rechnersystem wurde das in Abb.7 gezeigte heterogene Workstation-Cluster gewählt. Im Bild sind auch die real gemessenen Laufzeiten der Anwendungen gegenüber gestellt. Im Mehrbenutzerbetrieb ergibt sich die Zeit für den unbalancierten Ablauf aus der Zeit für die Anwendung auf dem langsamsten Prozessor; die unbalancierten Einzelanwendungen wurden hingegen auf einem Rechner mittlerer Leistung gemessen.

Eine Zufallsverteilung erbringt durch die Parallelität bereits eine starke Beschleunigung einer einzelnen Flächenerkennung bzw. finite Elemente Rechnung bzw. Datenbankoperation. Dennoch kann die Lastbalancierung durch Berücksichtigung der unterschiedlichen Rechnerleistungen, der tatsächlichen aktuellen Systemlast, der verschiedenen Auftragsgrößen und der Datenkommunikation zwischen den Servern eine weitere Verbesserung um 24% (Bilderkennung), 16% (FE-Berechnung) bzw. 44% (Datenbank) erreichen.

Im Mehrbenutzerbetrieb ohne Balancierung hängt die Gesamtlaufzeit von den Anwendungen auf den langsamen Rechnern ab, da jeweils Anwendungen gleichen Aufwands gewählt wurden. Eine einfache Lastbalancierung, die alle Aufträge wahllos verteilt, verschlechtert bei der Flächenerkennung sogar den Durchsatz, weil hohe Datenkommuniaktionskosten entstehen, obwohl die Rechenkapazitäten im Mittel besser genutzt werden könnten als ohne Lastbalancierung. Die Lastbalancierung nach dem HiCon-Modell bewirkt bei den Flächenerkennungs-Rechnungen trotz der im Prinzip mit Anwendungen gleich beladenen Rechner eine Durchsatzsteigerung von 5.4%, indem sie unter Beachtung der entstehenden Kommunikationskosten, d.h. geeigneter Partitionierung der Daten, die Anwendungen parallel laufen läßt und stets die Phasen geringerer Parallelität in einer Anwendung für Berechnung anderer Anwendungen nutzt. Eine Zufallsverteilung der finite Elemente Berechnung im Mehrbenutzerbetrieb verbessert den Ablauf bereits um deutliche 34%, da sie die beiden freien Prozessoren nutzt. Die HiCon-Lastbalancierung kann den Durchsatz jedoch um weitere 17% steigern. Die Datenbankoperationen sind aufgrund der Abhängigkeiten innerhalb der einzelnen Anfragen und wegen der sehr unterschiedlichen Auftragsgrößen schwer effizient zu balancieren. Eine wahllose Verteilung des Mehrbenutzerbetriebs erbringt 18% Steigerung, was durch geeignete Lastbalancierung um weitere 28% gesteigert werden kann.

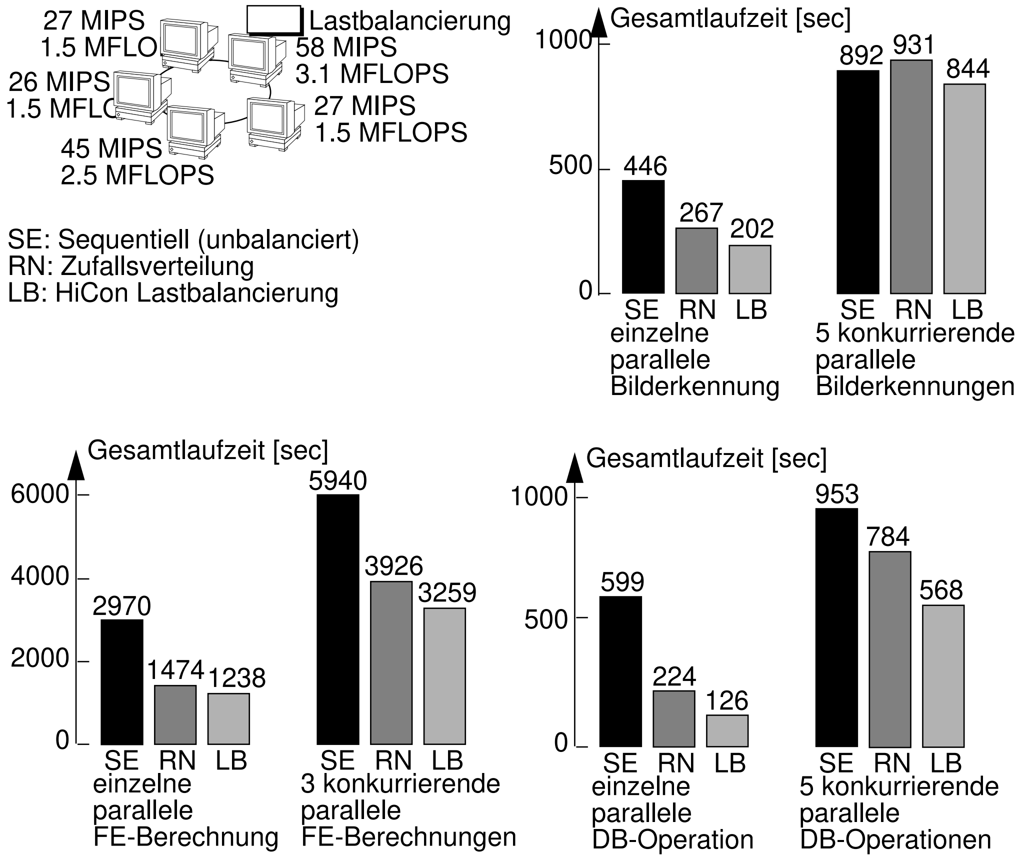

Abb.7: Systemkonfiguration und Laufzeiten der verschiedenen Anwendungen im Ein- und Mehrbenutzerbetrieb bei unterschiedlicher Lastbalancierungsunterstützung.

Für Evaluierungen der dezentralen Ansätze sei auf [6, 7] verwiesen. Abschließend sollte erwähnt werden, daß die vorgestellten Meßergebnisse trotz der realistischen Anwendungen und des Mehrbenutzerbetriebs lediglich Einzelbetrachtungen sind und nicht generell auf beliebig heterogen gemischte Lastszenarien oder Situationen hoher Last bei simplen Lastprofilen übertragen werden können. Man kann von guter dynamischer Lastbalancierung Durchsatzsteigerungen im Bereich 5 - 10% erwarten, sollte aber bedenken, daß Lastbalancierung die Ziele hat, katastrophale Zustände zu vermeiden, die Portabilität und Flexibilität von Anwendungen zu erhöhen, und die Komplexität, um von parallelen Systemen zufriedenstellende Leistung zu erhalten, zu reduzieren.

Zusammenfassung

In diesem Artikel wurde ein Ansatz zur anwendungsunabhängigen dynamischen Lastbalancierung in Workstation-Netzen vorgestellt und durch Messungen für ein breites Anwendungsspektrum bewertet. Die wesentlichen Merkmale im Vergleich zu existierenden Ansätzen sind zum einen der zentrale Ansatz zur Balancierung einzelner Cluster, dessen vielfältige Vorteile erläutert wurden, zum anderen die Berücksichtigung von Datenkommunikationsaufwand durch Zugriff auf globale Daten und zur Kooperation innerhalb paralleler Anwendungen. Der Ansatz bietet hohes Optimierungspotential, verlangt jedoch, daß existierende Anwendungen geeignet auf eine Client - Server Ablaufstruktur umgesetzt werden. Das HiCon-Modell ist daher nicht unmittelbar auf breiter Ebene einsatzfähig, liefert aber richtungsweisende Konzepte für die Weiterentwicklung dynamischer Lastausgleichsdienste in parallelen und verteilten Systemen.

Literatur

[1] W. Becker, Globale dynamische Lastbalancierung in datenintensiven Anwendungen, Fakultätsbericht 1993/1, Universität Stuttgart, Institut für parallele und verteilte Höchstleistungsrechner (IPVR), 1993.

[2] W. Becker, Das HiCon-Modell: Dynamische Lastverteilung für datenintensive Anwendungen auf Rechnernetzen, Fakultätsbericht 1994/4, Universität Stuttgart, Institut für parallele und verteilte Höchstleistungsrechner (IPVR), 1994.

[3] W. Becker, R. Pollak, Efficiency of Server Task Queueing for Dynamic Load Balancing, Fakultätsbericht 1994/9, Universität Stuttgart, Institut für parallele und verteilte Höchstleistungsrechner (IPVR), 1994.

[4] W. Becker, G. Waldmann, Exploiting Inter Task Dependencies for Dynamic Load Balancing, Proc. IEEE 3rd Int. Symp. on High-Performance Distributed Computing (HPDC), San Francisco, 1994.

[5] W. Becker, Das HiCon-Modell: Dynamische Lastverteilung für datenintensive Anwendungen auf Rechnernetzen, Informatik Forschung und Entwicklung 1994-3, Springer Verlag, 1994.

[6] W. Becker, J. Zedelmayr, Scalability and Potential for Optimization in Dynamic Load Balancing - Centralized and Distributed Structures, GI/ITG Workshop Parallel-Algorithmen, -Rechnerstrukturen und -Systemsoftware (PARS), Potsdam, 1994.

[7] P. Huish (Ed.), Interim Report, Deliverable CEC Projekt B2010 TEN-IBC E=MC^2, 1994.

Dipl. Inform. Wolfgang Becker, Telefon 0711 7816 433
E-Mail: `Wolfgang.Becker@informatik.uni-stuttgart.de`
Institut für parallele und verteilte Höchstleistungsrechner (IPVR), Universität Stuttgart