

Skalierbares LS-DYNA3D auf der Paragon

Norbert Ibold

Immer mehr kommerzielle Programme mit rechenzeitintensiven Anwendungen machen sich zur Verkürzung der Durchlaufzeit eines Jobs die neue Technologie zunutze, daß mehrere asynchron laufende und mit lokalem Memory ausgestattete Mikroprozessoren simultan an der Lösung einer gestellten Aufgabe arbeiten können. Die Parallelversion von LS-DYNA3D (Firma CAD-FEM GmbH), einem expliziten FEM-Programm für nichtlineare und hochdynamische Anwendungen, wurde auf dem Distributed Memory Supercomputer Intel Paragon XP/S installiert, der seit 1993 im RUS läuft und überwiegend für Softwareentwicklung eingesetzt wird. Anhand eines Datensatzes für eine Crash-Analyse, der freundlicherweise von der AUDI AG auf der VDI-Konferenz 1994 in Würzburg für Benchmark-Zwecke freigegeben wurde, konnten erste Erfahrungen mit dem LS-DYNA3D-Programm auf der Paragon gesammelt werden.

Die drei charakteristischen Zeitpunkte (Submit, Start, End) eines Batch-Jobs bestimmen die beiden zu addierenden Wartezeiten, die bis zum Anlaufen bzw. von da an bis zur Beendigung eines Jobs für den ungedulden Anwender vergehen. Die Supercomputer CRAY C94 und Intel Paragon XP/S des RUS, die jeweils mehrere CPUs verfügbar haben, werden zur Verringerung dieser Wartezeitanteile unterschiedlich eingesetzt.

Da sich auf der CRAY C94 immer viele Jobs, aber nur vier Prozessoren befinden, wird für diesen Rechner eine durchsatzoptimierte Fahrweise bevorzugt, bei der im Normalfall jeder Prozessor an einem anderen Job arbeitet und bei der sich lediglich der Zeitraum bis zum Start des Jobs verringert.

Zur Verkürzung der Rechenzeit dagegen können auf der Paragon bis zu 66 Knoten fest einem Job zugeteilt werden, während man auf ihre Verfügbarkeit eventuell länger warten muß. Das Produkt aus reservierter Prozessorzahl und paralleler Programmlaufzeit, das für die Kosten eines Jobs maßgeblich ist, kann dort von manchen Anwendungen besser und von anderen schlechter ausgenutzt werden.

Interessierte Kunden haben die Möglichkeit, Handhabung und Effizienz eines eigenen LS-DYNA3D-Beispiels (Version 920) auf dem Parallelrechner Paragon zu günstigen Konditionen selbst zu testen.

Durchführung des Crash-Benchmarks auf der Paragon XP/S

Auf dem Titelbild ist die Vergleichsspannung zu sehen, die beim Frontal-Crash eines Autos entsteht, das mit 30 km/h gegen eine starre Wand fährt. Innerhalb der Simulationszeit von 50 ms, in der das Fahrzeug ohne Crash 41.5 cm gefahren wäre, werden nach 91 873 variablen Zeitschritten der mittleren Länge von 0.54 Mikrosekunden die hinteren Fahrzeugbereiche von der Spannung erfaßt und deformiert. Das Modell enthält die räumliche Diskretisierung des realistischen Autos mit etwa 28 000 Schalenelementen, 200 Balkenelementen, 27 000 Knoten, 222 Materialgruppen und 5 200 Kontaktbereichen.

Immer wenn bei dynamischen Simulationen die Ausbreitung der mechanischen Welle ein Aspekt ist, müssen alle von der Last in der Struktur angeregten Frequenzen bei der Zeitintegration genau erfaßt werden. Dies erfordert sehr kurze Simulationszeitschritte, deren Länge mit der Zeit korreliert ist, in der die Welle das kleinste FEM-Element der Struktur mit der Schallgeschwindigkeit des entsprechenden Materials durchquert. Im Gegensatz zu impliziten

Codes kann bei expliziten Codes auch im nichtlinearen Fall an den vielen Zeitschritten die Rechenarbeit einer Gleichgewichtsiteration eingespart werden, da der gesuchte Systemzustand durch algebraisches (explizites) Auflösen der Bewegungsgleichung gefunden wird. Deshalb arbeiten Codes wie LS-DYNA3D im Kurzzeitbereich, z.B. bei Crash-Analysen, besonders effektiv.

Zur Simulation mit der Paragon sind vor dem Start von LS-DYNA3D folgende vorbereitende Schritte durchzuführen:

- Zerschneiden (Domain Decomposition) des Gesamtmodells in eine der gewünschten Prozessorzahl entsprechende Zahl von Teilmodellen (s. Bild 1)
- Belegung der Prozessoren in der Compute Partition.

Die vollautomatische Zerlegung des Autos wird mit einem seriellen Hilfsprogramm am besten vor dem Start von LS-DYNA3D vorgenommen. Das Zerlegungsprogramm kann auch nach dem Start als Unterprogramm aufgerufen werden, mit dem Nachteil, daß dann alle bis auf einen der reservierten Prozessoren auf das Ergebnis der Domain Decomposition unnötig warten müssen. Die Modelldaten der Teilmodelle, die aus einem Inputfile für jeden Prozessor bestehen, und das komplette LS-DYNA3D-Programm werden dann in das lokale Memory eines jeden Prozessors kopiert. Die Verformung der in Bild 1 für acht Prozessoren zu sehenden Teile wird asynchron von den zugeordneten Prozessoren im **Single Programm Multiple Data Modus** der Paragon berechnet.

Da jede CPU direkt nur auf ihr lokales Memory zugreifen kann und das Berechnungsergebnis für das Gesamtmodell richtig sein soll, müssen die Daten der jeweils an den Grenzen der Teilmodelle wirkenden Knotenkräfte an jedem Zeitschritt unter den CPUs benachbarter Domänen ausgetauscht werden. Zur Durchführung des Message Passing ist jeder Prozessor via Coprozessoren mit vier seiner Nachbarn vernetzt und die LS-DYNA3D Software mit den Subroutinen des Message Passing Standards MPI gelinkt. Die Zeit, in der die Coprozessoren Daten zu einem wartenden Prozessor befördern, geht bei der Berechnung verloren. Deshalb verschlechtert sich bei ungünstigem Verlauf der Domaingrenze, wenn diese zu viele FEM-Knoten und damit Kommunikationsarbeit enthält, das Verhältnis von Kommunikations- zu Rechenzeit und der zu erzielende Speed-up. Um bei gleicher Prozessorzahl alternative Teilmodellgrenzen erhalten zu können, werden von LS-DYNA3D mehrere Algorithmen zur Verfügung gestellt.

Zur Gewährleistung eines hohen Speed-ups sollten die den Prozessoren statisch zugeordneten Domänen gleich viel Rechenarbeit enthalten. Um ein gutes Load Balancing zu erreichen, werden den FEM-Elementen vor der Zerlegung Wichtungsfaktoren zugeordnet, die vom Element- und Materialtyp und von der Kontaktwahrscheinlichkeit des Elementes abhängen. Mit dem vom Benutzer veränderbaren Parameter COSTINC wird so auch der Aufwand für die Berechnung der Kontaktkräfte zwischen den Komponenten vor der Crash-Analyse erfaßt und durch entsprechende Anpassung der Elementezahl in einer Domäne das statische Load Balancing positiv beeinflusst.

Die in Bild 1 zu sehende Zerlegung des Autos für acht Prozessoren erreicht man mit dem GREEDY-Algorithmus und dem Default-Wert des Parameters COSTINC. Sie lieferte für alle Prozessoren etwa gleich viele FEM-Elemente:

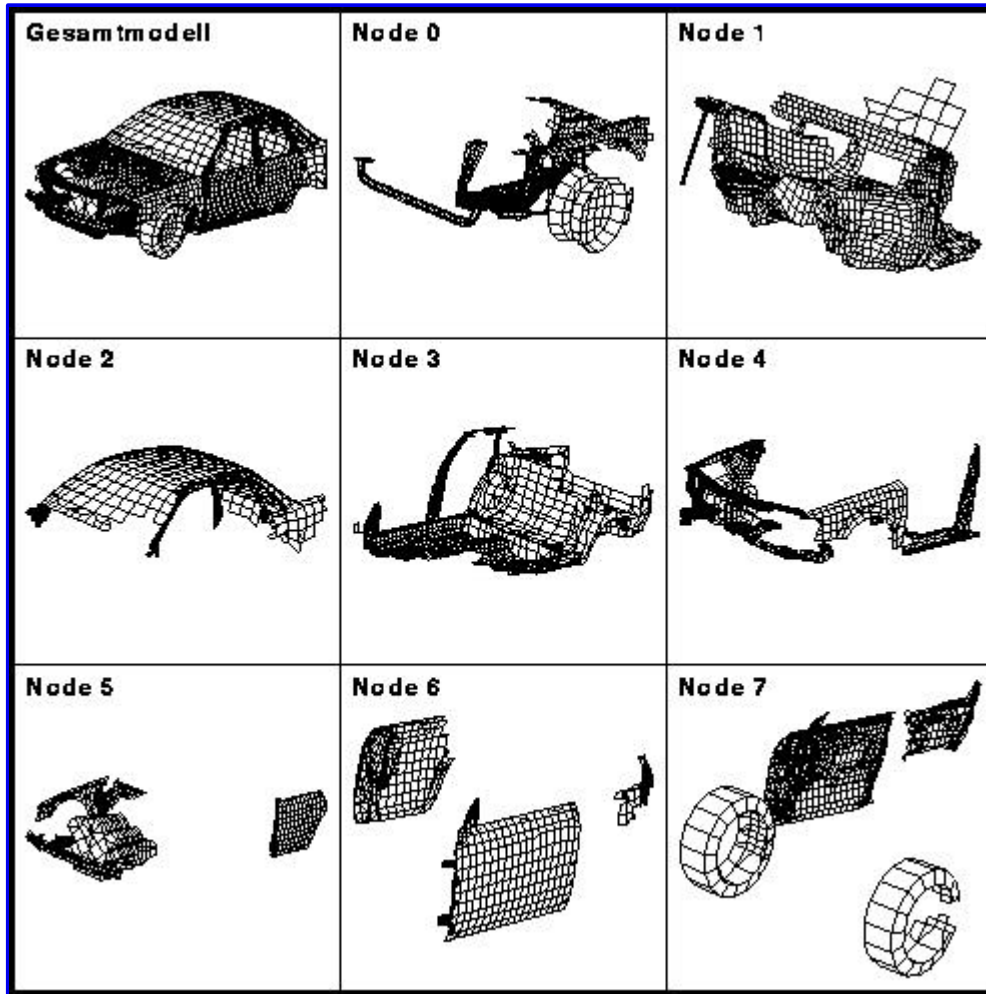


Bild 1: Domain Decomposition für acht Prozessoren

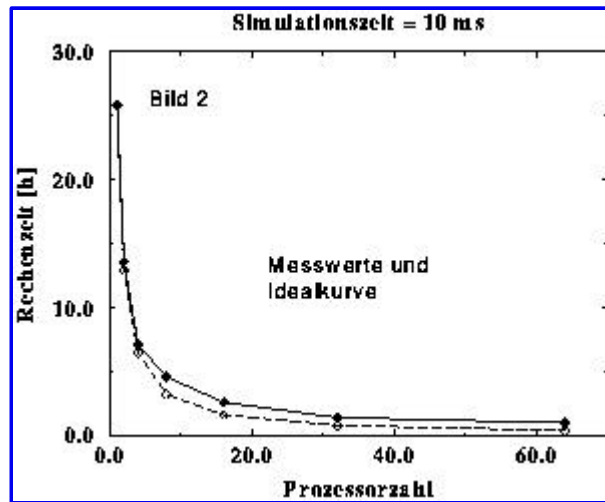
Beim Experimentieren mit dem alternativen RCB-Algorithmus und mit einem veränderten Wert des COSTINC-Parameters ergaben sich Unterschiede in der Gesamtrechenzeit von bis zu (+/-)15 % für das zugrundegelegte Automodell. Es ist wahrscheinlich, daß sich mit einer problemangepaßten Zerlegung von homogeneren und unkomplizierteren Modellen, wie sie beispielsweise in der Umformtechnik auftreten, noch signifikantere Unterschiede erzielen lassen.

Prozessorzahl	1	2	4	8	16	32	64
GREEDY [h]	25.8	13.6	7.1	4.0	3.0	1.3	1.0
	COSTINC=default				COSTINC=default		
RCB [h]	25.8	13.8	7.0	4.4	2.7	1.5	1.3
	COSTINC=default				COSTINC=90		

Tabelle 1: Rechenzeit für den 10 ms dauernden Crash bei verschiedenen Zerlegungen

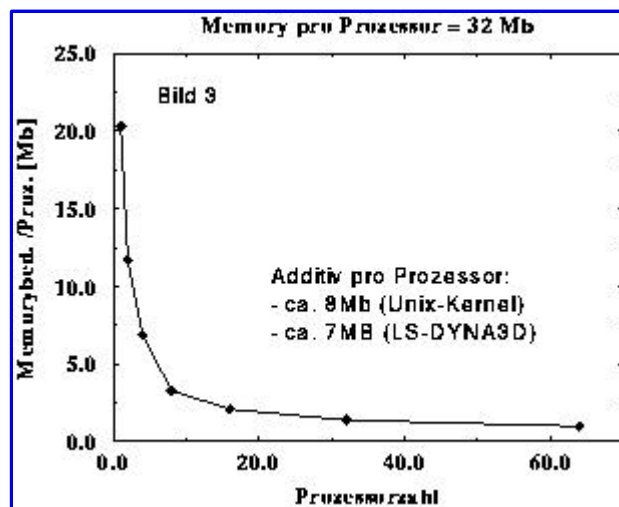
Diskussion der Benchmark-Ergebnisse

In Bild 2 ist die Rechenzeit, gemessen vom Programmstart bis zur Freigabe aller Prozessoren, inklusive der Zeit für die Domain Decomposition, als Funktion der Prozessorzahl für einen 10 ms dauernden Crash zu sehen:



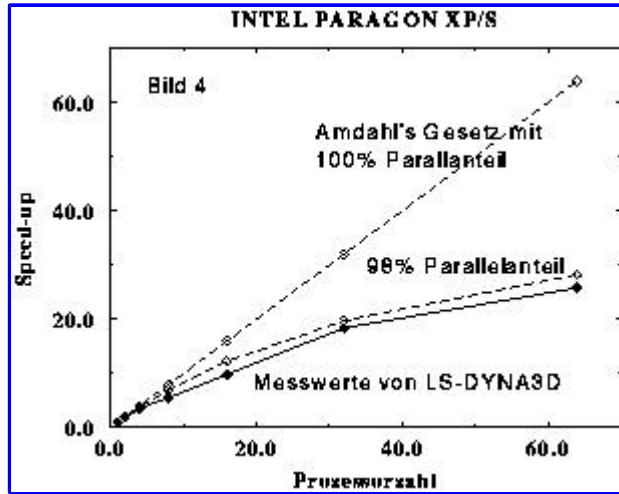
Da innerhalb der Rechenzeit nicht immer alle Prozessoren aktiv sind, wird die Idealkurve einer 100%-igen Auslastung nicht erreicht. Bei mehrmaliger Messung der Rechenzeit eines identischen Jobs wurden zum Teil erhebliche Schwankungen von bis zu 30 Prozent beobachtet. Es ist zu vermuten, daß diese Streuungen von der Nutzung derjenigen Prozessorpartitionen auf der Paragon kommen (z.B. bei der Datenein-/ausgabe), die während der Analyse nicht nur dem eigenen Job, sondern auch den anderen Jobs im Time Sharing-Betrieb verfügbar sind.

Die von LS-DYNA3D ausgewiesene Kurve des mittleren Memory-Bedarfs pro Knoten, die sich mit dem GREEDY-Zerlegungsalgorithmus und der Default-Einstellung der Wichtungsfunktion ergab, ist in Bild 3 zu sehen:



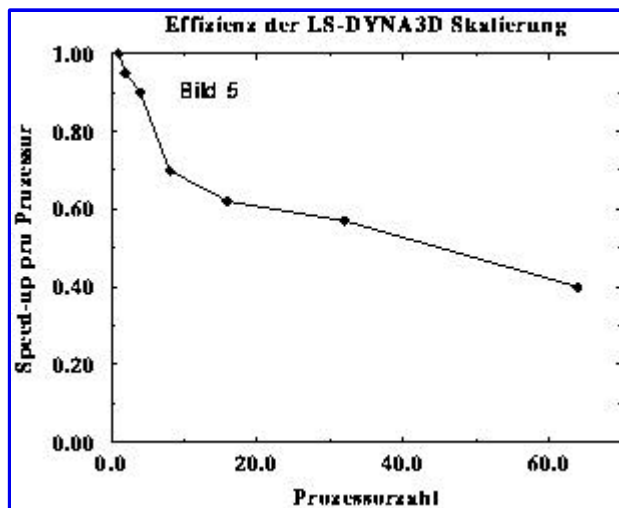
Die Kurve zeigt, daß mit dieser Einstellung der Parameter die zunächst gewünschte gleichmäßige Memory-Auslastung erreicht wird. Da zu den Kurvenwerten der MemoryBedarf für den Code und das Betriebssystem addiert werden muß, ist darauf zu achten, daß die Gesamtspeichergröße eines Knotens von 32 Mb nicht überschritten wird. Um ein Paging des Knotens zu vermeiden, wurde der serielle Referenzlauf auf einem Systemprozessor der Paragon ausgeführt, der mit 40 MB Memory ausgestattet war. Eine zu kleine Prozessorzahl führt bei vorgegebener Memory-Größe zu einer drastischen und unsinnigen Verlängerung der Rechenzeit und sollte daher vermieden werden.

Die mit verschiedenen Prozessorzahlen auf der Paragon gemessenen Speed-ups sind in Bild 4 zu sehen:



Aus der Gegenüberstellung mit den Werten, die sich durch die Anwendung des Amdahl'schen Gesetzes für Parallelrechner ergeben, kann man abschätzen, daß LS-DYNA3D in diesem Benchmark-Beispiel einem Programm mit ca. 95 % Parallelisierung äquivalent ist.

Die Skalierung von LS-DYNA3D ist nur so effektiv, wie sie die Hardware der angeforderten Prozessoren zu nutzen in der Lage ist. Zur Veranschaulichung wurde in Bild 5 der Speed-up pro Prozessor aufgetragen:



Die Effizienz sinkt mit steigender Prozessorzahl auf 40 Prozent des maximal möglichen Wertes, da dann die Kommunikation eines Prozessors zunimmt, die Granularität des simulierten Problems steigt und das Load Balancing schlechter wird.

Ausblick

Um die Belegung von höheren Prozessorzahlen auf der Paragon zu erleichtern und zur Verkürzung der Anlaufzeit von Jobs, wenn mehrere Benutzer auf ihr aktiv sind, sollten sehr viel mehr Prozessoren verfügbar sein. Ab Juli diesen Jahres werden RUS-Kunden auf eine NEC SX-4 mit 32 CPUs (Shared Memory) und auf eine CRAY T3E mit 512 CPUs zugreifen können. Man darf auf die Leistung der Hard- und Software gespannt sein.