Lastausgleich

- Einführung und Problemstellung
- Die grundsätzliche Lösung
- Interessante Fragestellungen
- Die Lastbewertung
- Die Lastverschiebung

Lastausgleich
Die acht wichtigsten Fragen

- Was ist Lastungleichheit
- Was charakterisiert das Problem?
- Wie sieht die grundsätzliche Lösung aus?
- Welche interessanten Fragestellungen gibt es?
- Wie kann man Lastausgleich integrieren?
- Wie werden Lasten bewertet?
- Wie werden Lasten verschoben?
- Wie arbeiten Systeme mit Lastausgleich?

Was ist Lastausgleich?

Besser sollte man zuerst fragen:

Was ist Lastungleichheit?

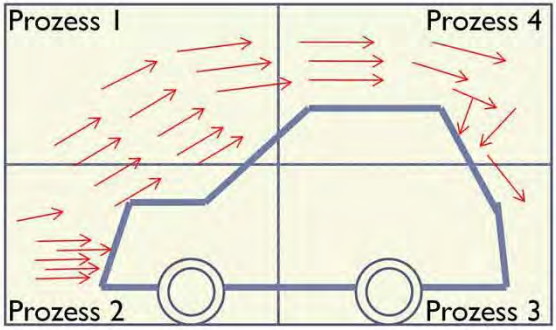
- Kritische Situation: die parallele Anwendung nutzt nicht alle Rechen-Ressourcen zu jeder Zeit

  Folge: Speedup-Kurve bekommt Knick
- Grund: Rechenlast ist unausgeglichen

  Möglicherweise zu Programmstart ausgeglichen, dynamischer Effekt zur Laufzeit
- Folgen: längere Programmlaufzeit, geringere Effizienz der Parallelisierung

Beispiel für Lastungleichheit

- Gleichverteilung der Volumen zu Beginn:
  - alles ausgeglichen
- Danach Verlagerung der Teilchen:
  - Lastungleichheit zwischen den Volumenteilen



Warum ist das interessant?

Praktische Gründe

- Reduziere Programmlaufzeit
- Erhöhe Effizienz der Rechnernutzung

Forschungsgegenstände

- NP-harte Optimierungsprobleme
  Globales Scheduling (was wird wann wo berechnet?)
- Anspruchsvolle Fragestellungen aus dem Bereich der Betriebssystemtechnik

Das Problem

Lastungleichheit variiert dynamisch

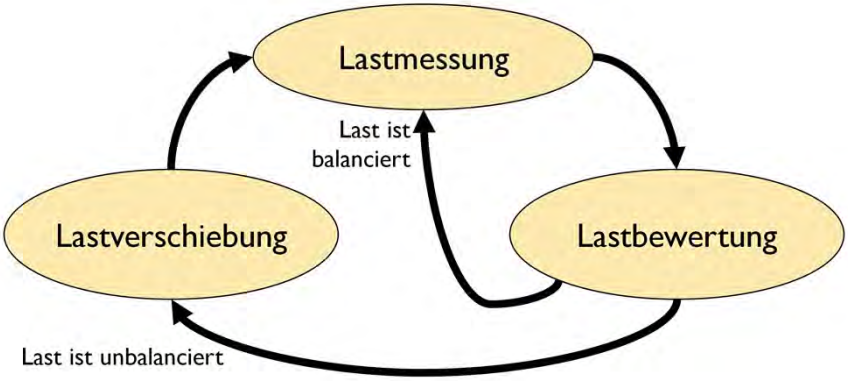
- Typisches Verhalten vieler paralleler und verteilter Anwendungen
- Statischer Ausgleich (bei Programmstart) nicht möglich, da Lastungleichheit daten- und zeitabhängig
- Lastungleichheit verringert den Systemdurchsatz

Ressourcen von Interesse

- Hauptsächlich der Prozessor
- Seltener: Speicher, Netzwerk, Platten

Die grundsätzliche Lösung

Regelkreis der Lastverwaltung



Die grundsätzliche Lösung...

Es gelten die üblichen Regelkreischarakteristika

- Schnelle Reaktion
- Genaues Nachregeln
- Keine Überreaktion
- Kein Oszillation
- Keine Belastung des geregelten Systems

Technische Umsetzung der Anforderungen sehr komplex

Teilweise sich widersprechende Anforderungen

Interessensbereiche

- Integrationstyp
  - Anwendungs- oder Systemintegration
- Lastmessung
  - Schnell, genau, umfassend, beeinflussungsarm
- Lastbewertung
  - Schnell, korrekt
  - Die Zukunft aus der Vergangenheit vorhersagen
- Lastverschiebung
  - Muss mehr bringen als kosten

Integrationstyp

Anwendungsintegriert

- In den Code der Anwendung integriert
- Überschaubare Implementierungskomplexität
- Wiederholter Implementierungsaufwand
- Meist nicht für andere Programme direkt übernehmbar
- Optimal an eine Anwendung angepasst
- Arbeitet mit allen Betriebssystemen zusammen (auch mit lastbalancierten)

Integrationstyp...

Systemintegriert

- In das Betriebssystem integriert oder mit ihm eng verbunden
- Hohe Implementierungskomplexität
- Einmaliger Implementierungsaufwand
- Optimal an das System angepasst, aber an keine spezifische Anwendung
- Arbeitet mit allen Anwendungen zusammen (auch mit lastbalancierten)

Lastmessung

Dynamisches Messen verschiedener Kenndaten

- Zunächst meist nur knotenbezogen
- Zur Verfeinerung auch prozessbezogen
- Meist sehr beeinflussungsarm

Mechanismen

- Online-Werkzeuge wie z.B. Dyninst/Paradyn

Lastbewertung

Einfach

- Einzelwerte: Prozessorleerlaufzeit, Speichernutzung

Komplex

- Systemkenndaten, Anwendungskenndaten
- Auswertung mit neuronalen Netzen
- Global konsistenter Blick
- Metawissen (z.B. vergangene Entscheidungen)

Problem: alle Daten beschreiben Vergangenheit

Lastbewertung...

Konkretes System

- Zentrale Bewertungskomponente
- Erfasse alle knotenbezogenen Daten
- Bestimme einen überlasteten und einen unterlasteten Knoten
- Messe prozessbezogene Daten auf dem überlasteten Knoten
- Identifiziere geeigneten Kandidaten zur Verschiebung

Lastbewertung...

Probleme

- Zentrale Komponente in größeren Systemen nicht mehr praktikabel
- Daten müssen über ein Intervall erfasst werden, das nicht zu kurz sein darf
- Gleichzeitig aber schnelle Reaktion erwünscht
- Situationsabhängige Verfeinerung der Messungen
- Stabil gegen Oszillationen

Lastverschiebung

Verschiebeobjekte

- Einfacher: Anwendungsebene (feingranular)
  Datenpakete, Anfrage
- Komplex: Systemebene (grobgranular)
  Prozesse, Objekte, Dateien

Allgemeine Probleme

- Bezug zu anderen Objekten muss gesichert sein
- Verschiebung muss Gewinn bringen

Lastverschiebung...

Beispiel: Prozessverschiebung

- Wie stoppt man einen laufenden Prozess?
- Wie verschiebt man ihn?
- Was verschiebt man konkret?
- Wie behandelt man offene Dateien?
- Wie behandelt man Signale?
- Was geschieht während der Verschiebung?
- Was passiert mit Nachrichten nach der Verschiebung?
- Wie finden sich die Kommunikationspartner nach der Verschiebung wieder?

Lastverschiebung...

Beispiel: Intel Hypercube Parallelrechner

- Verschiebung mittels Paging-Mechanismus über das Netzwerk
- Nur aktive Code-Seite verlagern; restliche Seiten werden durch auftretende Seitenfehler geholt
- Nachrichten an den Prozess bei den Sendern zwischengespeichert
- Quellknoten der Verschiebung gibt neuen Ort an Sender bekannt; diese korrigieren ihre Tabellen

Lastverschiebung...

Beispiel: Verschiebung im Cluster mit CoCheck

- Alle Kommunikationen stoppen
- Prozessabbild durch Dump-Funktion erstellen
- Prozess zum Ziel kopieren
- Andere Prozesse weiterlaufen lassen; ggf. über neuen Ort informieren

Lastverschiebung...

Beispielproblem: Offene Dateien

- Mitprotokollieren ***aller*** Systemaufrufe durch zwischengeschaltete Bibliothek (wie MPI-Profiling-Bibliothek)
- Damit weiß man immer, wo der Prozess in einer Datei gerade liest
- Nach der Verschiebung Dateien wieder öffnen und die vermerkte Stelle wieder anfahren

Zur Lage der Werkzeuge

Benötigte Zusatzwerkzeuge

- Visualisierung der Messungen und Entscheidungen des Lastausgleichers

Situation der aktuellen Werkzeuge

- Keines kann mit Prozessmigration umgehen
- Beispielproblem: Verschiebung eines Prozesses, für den ein Haltepunkt definiert ist
- Beispielproblem: Verschiebung eines Prozesses, dessen Code dynamisch instrumentiert ist

Und weil das ganze so schwer ist...

... gibt es kaum Systeme mit dynamischem Lastausgleich

- Aber es gibt Tonnen von Literatur, bei der Simulationen des Verhaltens eines solchen Systems analysiert werden
- Und es gibt die initiale Lastplazierung, die meist banal ist, aber auch als Lastausgleich verkauft wird

Aber es gibt viele Anwendungen, bei denen die Programmierer mühevoll einen Lastausgleich eingebaut haben

Lastausgleich
Zusammenfassung

- Lastungleichheit ist die ungenügende Nutzung wichtiger Ressourcen
- Lastungleichheit variiert dynamisch und lässt sich deshalb nicht durch einen statischen Ansatz kontrollieren
- Die Lastverwaltung folgt dem Prinzip des Regelkreises
- Die Lastverwaltung kann in die Anwendung oder in das System integriert werden
- Eine effiziente Lastbewertung unterliegt vielen Einzelfragestellungen
- Die Lastverschiebung ist die technisch anspruchsvollste Komponente
- Aufgrund der Komplexität gibt es kaum Realisierungen auf Betriebssystemebene
- Häufig wird aber Lastausgleich in die Anwendung eingebaut