

Ereignisgesteuerte Ablaufplanung von Mehrkern-Systemen

Frederik Völkel

Friedrich-Alexander-Universität
Erlangen-Nürnberg

02.12.2015

Motivation

- Prozessoren werden nicht mehr signifikant schneller.
- Anzahl an Prozessoren auf einem Chip steigt jährlich.
- Neue Probleme für Echtzeit-Ablaufplanung
 - Wo sollen Arbeitsaufträge ausgeführt werden?
 - Dürfen Arbeitsaufträge migriert werden?

Inhalt

- 1 Klassen von Ablaufplanern
- 2 Definitionen
- 3 Partitionierte Ablaufplanung
- 4 Globale Ablaufplanung
- 5 Hybride Ablaufplanung
- 6 Effizienz
- 7 Aktuelle und Offene Fragen

Klassen von Ablaufplanern

- Allokations Problem
 - ① Keine Migration
 - ② Migration auf Task Ebene
 - ③ Migration auf Job Ebene
- Prioritäten Vergabe
 - ① Feste Vergabe auf Task Ebene
 - ② Feste Vergabe auf Job Ebene
 - ③ Dynamische Prioritäten Vergabe

Auslastung(Utilization)

Die Auslastung u einer Aufgabe ist die WCET geteilt durch die Periode der Aufgabe.

Planbarkeit

Eine Aufgabe ist mit einem bestimmten Ablaufplaner planbar, wenn ihre WCET kleiner oder gleich ihres Termins ist.

Eine Menge an Aufgaben ist planbar, wenn alle ihre Aufgaben planbar sind.

Machbarkeit(feasibility)

Eine Menge von Aufgaben ist machbar, wenn ein Ablaufplaner existiert, der jede mögliche Reihenfolge von Aufgaben planen kann.

Bedingung für Machbarkeit von Aufgaben Mengen mit impliziten Terminen:

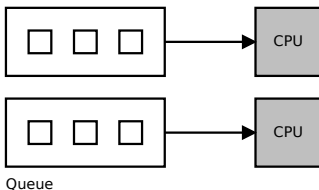
$$u_{sum} \leq m. \quad (1)$$

Optimalität(Optimality)

Ein Ablaufplaner ist Optimal bezüglich eines Aufgaben Modells, wenn alle seine Mengen von Aufgaben machbar sind.

Partitionierte Ablaufplanung

- Keine Task oder Job Migration.
- Probleme:
 - Auf welchem Prozessor wird ein Task ausgeführt(Allokations Problem)?
 - Wie werden Prioritäten vergeben?



Vorteile/Nachteile

- Überschreiten eines Termins kann nur Auswirkungen auf Aufgaben auf den selben Prozessor haben.
- Separate Run-Queus für jeden Prozessor.
- Keine Migrationskosten.
- Auf einzelnen Prozessoren können einzel Prozessor Techniken angewandt werden.
- Prozessoren sind oft im Leerlauf.
- Für jede neue Menge an Aufgaben muss einen neue Verteilung auf die Prozessoren gefunden werden.

Allokations Problem

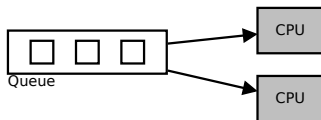
- Next Fit ($O(n)$)
- First Fit ($O(n * \log(n))$)
- Best Fit ($O(n * \log(n))$)
- Small Tasks ($O(n)$)
- General Task ($O(n)$)

Prioritäts Vergabe

- Rate Monotonic (RM)
 - Geringste Periode \rightarrow höchste Priorität
- Deadline Monotonic (DM)
 - kürzester relativer Termin \rightarrow höchste Priorität
- Earliest Deadline First (EDF)
 - frühester absoluter Termin \rightarrow höchste Priorität

Globale Ablaufplanung

- Migration von Aufgaben ist erlaubt.
- In der Regel vielfältiger als Partitionierte Algorithmen.
- Beispiel: Ein Globaler Algorithmus mit dynamischen Prioritäten dominiert alle anderen Klassen von Algorithmen



Vorteile/Nachteile

- Weniger Kontext Wechsel.
- Unerwartete Ausführungszeiten wirken sich auf das gesamte System aus.
- Es werden keine Allokations Algorithmen benötigt.
- Nur eine Run-Queue für alle Prozessoren.
- Migrations Kosten.
- Analyse is komplex.

Auslastungs basierte Ablaufplanung

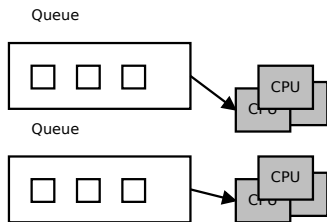
- Aufgaben werden in zwei Gruppen aufgeteilt.
- Gruppe mit Auslastung über einer Schelle \Rightarrow höchste Priorität.
- Gruppe mit Auslastung unter der Schwelle \Rightarrow Priorität von EDF oder ähnlichem.
- $EDF(k)$. k Aufgaben mit der größten Auslastung bekommen die höchste Priorität.

Proportional Faire Algorithmen(Pfair)

- Zeit wird in gleich lange Stücke aufgeteilt.
- Planungs Entscheidungen nur am Anfang eines solchen Stücks.
- Ausführungszeit proportional zur Auslastung einer Aufgabe.
- Optimal für periodische Aufgaben Mengen mit impliziten Terminen.
- Spezialfall: Boundary Fair.
- Problem: Am Anfang eines Abschnitts müssen sich alle Prozessoren synchronisieren.

Hybride Ablaufplanung

- Kombination aus Partitioniertem und Globaler Ablaufplanung um Nachteile zu Eliminieren.
- Im besonderen um Migrations Kosten zu minimieren und um die Auslastung von Prozessoren zu Verbessern.
- Beispiel: Prozessoren zu Clustern zusammenfassen.



Auslastungs Grenzen

- Für Aufgaben Mengen mit Impliziten Terminen.
- Die schlimmst mögliche Auslastungs Grenze eines Ablaufplaners ist die minimale Auslastung einer Menge von Aufgaben die gerade noch Planbar ist.

Approximations Rate

- Ermöglicht den Vergleich eines Algorithmus mit einem optimalen Algorithmus.
- Minimale Anzahl an Prozessoren um eine Menge an Aufgaben mit dem optimalen Algorithmus zu Planen, wird ins Verhältnis zur Anzahl der Prozessoren die der andere Algorithmus benötigt um die selbe Menge zu Planen, gesetzt.

Empirische Verfahren

- Generiere eine Anzahl an beliebigen Aufgaben Mengen und teste wie viele davon verschiedene Algorithmen Planen können.
- Simulationen: Können keine Planbarkeit zeigen aber sind hinreichend für nicht Planbarkeit.

- Aktuell:
 - Energie effiziente Ablaufplanung.
 - Fehler Tolerante Ablaufplanung.
- Offene Fragen:
 - Prozessor Auslastung bei Partitionierter Ablaufplanung
 - Berücksichtigung von moderner Hardware Architektur.
 - Grundsätzlich neue Ansätze die nicht auf einzel Prozessor Algorithmen basieren.

Danke für eure Aufmerksamkeit!

Fragen?!?