Многопроцессорное расписание

Габдрахманов Азат

27 ноября 2024 г.

Введение

- Многопроцессорное расписание — важная задача в области теории расписаний и оптимизации.
- Применяется в компьютерных системах, производственных процессах, логистике и других областях.
- Цель: оптимальное распределение заданий между процессорами для достижения наилучшего результата.

Формальное определение задачи

Условие:

- Задано множество заданий T
- Число процессоров m
- Длительность каждого задания $l(t)$ для $t \in T$
- Общий директивный срок D

Вопрос: Существует ли m-процессорное расписание для заданий из T, которое удовлетворяет общему директивному сроку D?

Формальное условие:

$$\exists\, o : T \to \mathbb{Z}, \quad \forall u \geq 0, \quad \#\{t \in T \mid o(t) \leq u < o(t) + l(t)\} \leq m$$

и

$$\forall t \in T, \quad o(t) + l(t) \leq D$$

Общие виды заданий

- **Процессоры с разной производительностью:**
  - Некоторые процессоры могут выполнять задания быстрее других.
  - Время выполнения задания на процессоре P_j: $l(t)/s_j$, где s_j — скорость процессора.
- **Зависимости между заданиями:**
  - Некоторые задания не могут начинаться до завершения других.
- **Приоритеты заданий:**
  - Задания имеют разные приоритеты, влияющие на порядок их выполнения.

Частные случаи задачи

- **Одинаковое время выполнения всех заданий:**
  - $l(t) = l$ для всех $t \in T$
  - Проблема становится аналогом задачи разбиения множества (Partition)
  - В этом случае задача может быть решена за полиномиальное время.
- **Ограниченное количество процессоров:**
  - Для фиксированного m задача может быть проще решаемой.
- **Отсутствие дедлайнов:**
  - Задача упрощается, если нет ограничений по времени завершения.

Доказательство NP-полноты: Принадлежность к классу NP

Класс NP:

- Включает задачи, решения которых можно **проверить** за полиномиальное время.

Сертификат для задачи многопроцессорного расписания:

- Функция назначения $o : T \to \mathbb{Z}$, определяющая время начала выполнения каждого задания.

Процесс проверки корректности расписания:

1. **Проверка выполнения всех заданий до директивного срока** D**:**

$$\forall t \in T, \quad o(t) + l(t) \leq D$$

2. **Проверка ограничения на количество одновременно выполняющихся заданий:**

$$\forall u \geq 0, \quad \#\{t \in T \mid o(t) \leq u < o(t) + l(t)\} \leq m$$

- Используем алгоритм сортировки событий (начало и конец

Доказательство NP-полноты: NP-Трудность

Класс NP-трудных задач:

- Задачи, к которым **любые** задачи из NP могут быть сведены за полиномиальное время.

Сведение из задачи Partition:

- **Задача Partition:**
  - Дано множество чисел $S = \{s_1, s_2, \ldots, s_{2m}\}$ с суммой $2K$.
  - Вопрос: Можно ли разбить S на два подмножества S_1 и S_2 такие, что $\sum_{s \in S_1} s = \sum_{s \in S_2} s = K$?
- Partition является NP-полной задачей.

Сведение Partition к многопроцессорному расписанию:

1. **Построение экземпляра задачи расписания:**
   - **Задания** T**:** Для каждого числа $s_i \in S$ создаём задание t_i с длительностью $l(t_i) = s_i$.
   - **Количество процессоров** $m' = 2$.
   - **Директивный срок** $D = K$.
2. **Эквивалентность решений:**
   - Если существует расписание, завершающее все задания до K на 2 процессорах, то это соответствует разбиению S на S_1 и S_2 с

Применение задачи многопроцессорного расписания

- **Компьютерные системы:**
  - Распределение процессов и потоков между ядрами процессора.
- **Производственные системы:**
  - Планирование производственных процессов на различных машинах или линиях.
- **Логистика:**
  - Организация доставки и распределения ресурсов.
- **Облачные вычисления:**
  - Оптимизация использования виртуальных машин и серверов.

Заключение

- Многопроцессорное расписание — фундаментальная задача в оптимизации и теории сложности.
- Задача NP-полная, что делает её решение для больших экземпляров вычислительно сложным.
- Практическое применение в различных областях требует использования эвристических и приближенных методов.
- Понимание сложности задачи важно для разработки эффективных алгоритмов распределения ресурсов.

Список литературы

- M. R. Garey, D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W.H. Freeman, 1979.
- R. Graham. *Scheduling: Theory, Algorithms, and Systems*. Prentice Hall, 1966.
- K. Pruhs, J. R. Rice. *Scheduling on Multiprocessors: Algorithms and Complexity*. Springer, 1997.