Приближенные алгоритмы составления расписаний в многопроцессорных компьютерных системах с учетом расхода энергии

Захарова Юлия Викторовна

Институт математики им. С.Л. Соболева СО РАН (Омский филиал)

Исследование выполнено за счет гранта Российского научного фонда № 22-71-10015.

Мотивация (типы работ)

Распараллеливаемые работы

- **Rigid jobs** ($size_j$): фиксированное число требуемых процессоров.
- **Moldable jobs** (δ_j): число используемых процессоров выбирается при составлении расписания и не меняется.
- **Malleable jobs** (δ_j): число используемых процессоров может меняться в процессе выполнения.

Многопроцессорные работы

- **Single mode jobs** (fix_j): фиксированный набор требуемых процессоров.
- **Multimode jobs** (set_j): альтернативные наборы используемых процессоров.

Входные данные

Процессоры и работы

m – общее число процессоров.
$\mathcal{J} = \{1, \ldots, n\}$ – множество работ:
$size_j(\delta_j)$ – число (верхняя граница) требуемых процессоров для работы j;
W_j – объем работы j.

Типы систем

Прерывания и миграция: системы с общей памятью и системы с распределенной памятью.

Скорости работ

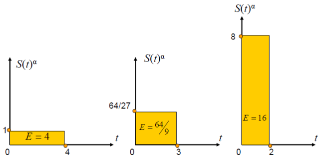
Если процессор работает со скоростью s, то энергопотребление составляет s^α в единицу времени, где $1 < \alpha \leq 3$ – константа. Предполагается, что процессор может работать с любой скоростью из заданного диапазона.

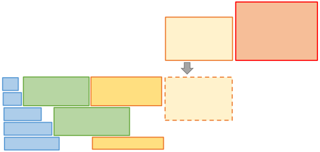
Скорость и энергия

Процессоры ($\alpha = 1.11$ для Intel PXA 270, $\alpha = 1.62$ для Pentium M770, $\alpha = 1.66$ для TCP offload engine, $\alpha = 3$ для CMOS устройств) могут менять скорость при выполнении операций.

Потребление энергии (статическая и динамическая части)

$$\sum_{j \in J} p_j \left(\frac{W_j}{p_j} \right)^{\alpha} + m \cdot P_{stat} \cdot C_{\max}$$





Предыдущие исследования

$P|pmtn, energy|C_{\max}$

Shabtay, Kaspi (2006): $O(n^2)$

$P|(prec), energy|C_{\max}$

Pruhs, van Stee (2007): PTAS; $O(\log^{1+2/\alpha} m)$
Bampis, Letsios, Lucarelli (2014): $(2 - \frac{1}{m})$
Bunde (2009): poly, r_j, $W_j = 1$

Распараллеливаемые работы

Drozdowski (2008): Scheduling for Parallel Processing
Kononov, Zakharova (2022): Approximation Algorithms for Speed Scaling Scheduling
Benoit, Canon, Elghazi, Heam (2023): Schedules for Independent Parallel Tasks to Minimize Energy Consumption

Двухэтапный алгоритм (Rigid jobs)

Этап 1 (нижняя граница, условия Куна-Таккера)

$$LB \to \min,$$
$$\max_{j\in\mathcal{J}} \frac{p_j}{size_j} \le LB,$$
$$\frac{1}{m}\sum_{j\in\mathcal{J}} size_j p_j \le LB,$$
$$\sum_{j\in\mathcal{J}} W_j^{\alpha} p_j^{1-\alpha} \le E.$$

Этап 2 (алгоритм списочного типа)

На каждом шаге алгоритм назначает очередную работу, для которой имеется достаточное количество процессоров.

Теорема

$(2-\frac{1}{m})$-приближенное решение может быть найдено за время $O(n^2)$ для задачи $P|size_j, energy|C_{\max}$.

Модель с конфигурациями работ

Конфигурации

c есть допустимое подмножество работ $\mathcal{J}_c$ и назначенные скорости $S_{j,c}$ для них.

$E_c = \sum_{j \in J_c} (S_{j,c})^{\alpha} + mP_{stat}$

Прямая и двойственная задачи

$$\begin{gathered} \sum_{c \in C} x_c \to \min, \\ \sum_{c \in C:\ j \in J_c} \frac{S_{j,c} x_c}{W_j} \geq 1,\ j \in \mathcal{J}, \\ -\sum_{c \in C} E_c x_c \geq -E, \\ x_c \geq 0,\ c \in C. \end{gathered}$$

$$\begin{gathered} \sum_{j \in \mathcal{J}} \lambda_j - E\mu \to \max, \\ \lambda_j \geq 0,\ j \in \mathcal{J}, \\ \mu \geq 0, \\ E_c\mu - \sum_{j \in J_c} \frac{S_{j,c}\lambda_j}{W_j} \geq (-1),\ c \in C. \end{gathered}$$

Теорема

Для $P|size_j, pmtn, energy|C_{\max}$ расписание с длиной не более OPT и потреблением энергии $E + \varepsilon$ может быть найдено за время, полиномиальное от m, $1/\varepsilon$ и длины входа.

Результаты для различных типов работ

Тип работ	Прерывания и миграция	Без прерываний	Частичный порядок
Rigid	OPT+ε	$2 - \frac{1}{m}$	$\frac{2-q}{1-q}$
Moldable	?	$2 - \frac{2}{m+1}$	$\frac{3+\sqrt{5}}{2}$
Malleable	poly	–	?

Относительное отклонение от нижней границы LB