

Минимизация времени решения задач



Майданов

Юрий Сергеевич

к.т.н., доцент Кафедры ВС

Минимизация времени решения прикладных задач

Рассмотрим планирование работы вычислительной системы при одинаковой трудоемкости фрагментов задач по заданным графу информационных связей $D = (W, P, \Gamma)$ и диагностическому графу $G = (U, T)$

При этом будем стремиться минимизировать время цикла решения при ограничении на длительность цикла диагностирования $T_{\text{цк}} \leq T_{\text{цкмак}}$

Будем получать целевую диаграмму загрузки элементарных машин в несколько этапов

Сначала исходную $P1$ без учета диагностических процедур, потом на основе $P1$ – целевую $P2$, обеспечивающую как решение прикладных задач, так и проведение самодиагностики

Использование парных диагностических взаимопроверок

При таком методе проведения взаимных тестов между элементарными машинами можно заранее определить количество дополнительных ресурсов, необходимых для проведения одного цикла диагностирования

Так как трудоемкость фрагментов задач одинакова, то будем считать, что общее количество ресурсов, необходимых для проведения диагностических операций, численно равно количеству решенных фрагментов задач

Таким образом, общее количество ресурсов составляет $m * 2$ решений фрагментов задач, где $m = |T|$ - количество связей диагностического графа.

Половина этих ресурсов m используется для решения прикладных задач, а другая половина выделяется дополнительно

Использование парных диагностических взаимопроверок

Исходная диаграмма загрузки Р1 должна позволять за период $T_{цк\max}$ выделить дополнительные элементарные машины для решения m фрагментов задач

Иначе говоря, должно выполняться условие:

$$\sum_{\tau=1}^{T_{цк\max}} (n - F(\tau)) \geq m$$

Кроме того, необходимым условием проведения каждой элементарной проверки является загрузка одной из участвующих в ней элементарных машин фрагментом прикладных задач wq в качестве основной

Поэтому даже если на некотором t -м такте имеем $F(t) < n / 2$, то все равно максимальное количество элементарных проверок, которые могут быть выполнены на этом такте, не превышает $n / 2$

Использование парных диагностических взаимопроверок

Таким образом, исходная диаграмма загрузки должна удовлетворять следующему условию:

$$\sum_{\tau=1}^{T_{\text{цкmax}}} m_{\tau} \geq m,$$

$$\text{где } m_{\tau} = \min\{F(\tau), n - F(\tau)\}$$

Если условие не выполняется, то диаграмму загрузки Р1 следует модифицировать с целью сокращения загрузки фрагментами прикладных задач за счет увеличения времени ее решения

При этом может быть использован алгоритм заполнения свободных временных тактов

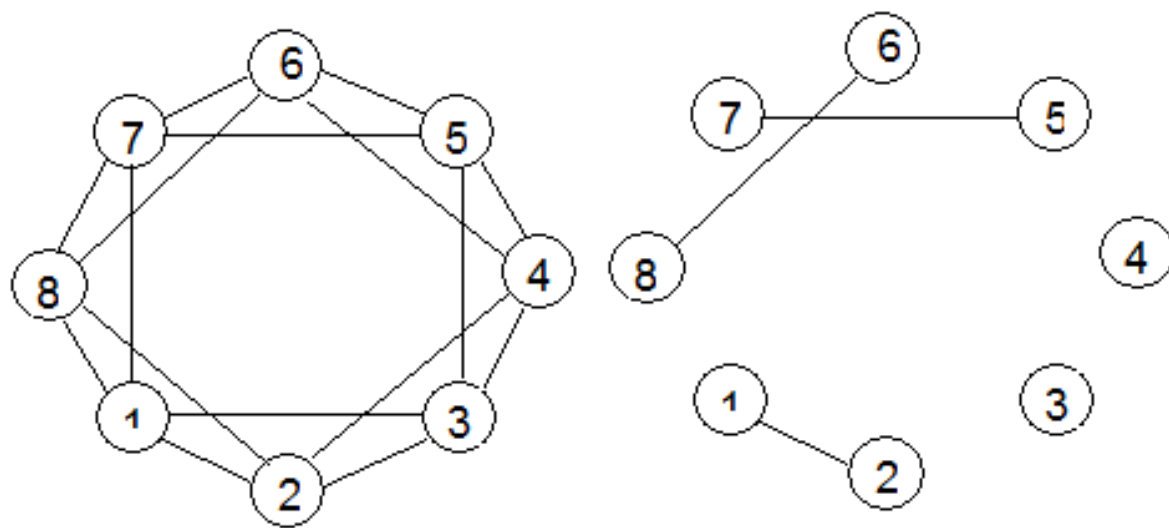
Использование парных диагностических взаимопроверок

После этого производится попытка продублировать решение некоторых фрагментов прикладных задач с целью выполнения элементарных проверок, описанных диагностическим графом $G = (U, T)$

Выполнение условия слайда 4 является необходимым, но недостаточным для обеспечения $T_{цк} \leq T_{цк\max}$. На каждом t -м такте работы системы выполняется множество элементарных проверок, описываемых подграфом $G_t = (U, T_t)$ диагностического графа G . Диаграмма загрузки P_1 накладывает на G_t следующие ограничения:

$ T_\tau \leq F(\tau);$	Кроме того, для любой вершины $u_j \in U$ любого из графов G_τ выполняется
$ T_\tau \leq n - F(\tau)$	требование $ \Gamma^{-1}(u_j) \leq 1$

Пример такого диагностического
графа G и его подграфа G_t



Использование парных диагностических взаимопроверок

Так как в процессе диагностирования необходимо выполнить все элементарные проверки, описанные в графе G , то должно выполняться следующее условие:

$$G_1 \cup G_2 \cup \dots \cup G_{T_{\text{ЦК}}} = G$$

Таким образом, для расширения диаграммы загрузки диагностическими процедурами необходимо решить следующие две задачи:

1. Определить множества элементарных проверок, выполняемых на каждом шаге работы системы
2. Распределить исходные данные по элементарным машинам в соответствии с полученными результатами

Распределение исходных данных между элементарными машинами

1. Присвоить $\tau = 0$.
2. Выбрать G_τ .
3. На основе диаграммы загрузки определить множество фрагментов задач W_τ , которые должны быть выполнены в момент времени τ .
4. Все вершины диагностического графа G пометить как свободные.
5. Выбрать $u_i \in U$, помеченную в качестве свободной.
6. Выбрать $w_k \in W_\tau$.
7. Назначить w_k на решение элементарной машине, соответствующей u_i .
8. Копии w_k назначить на решение всем машинам, соответствующим $u_j \in \Gamma^{-1}(u_i)$.
9. Вершины u_i и $u_j \in \Gamma^{-1}(u_i)$ пометить как занятые.
10. Назначить $W_\tau = W_\tau \setminus w_k$.
11. Если $|W_\tau| > 0$, то выполнить переход к п.5.
12. Присвоить $\tau = \tau + 1$.
13. Если $\tau < T_{\text{цкф}}$ то выполнить переход к п.2.
14. Конец алгоритма.

Спасибо за внимание!