Формальная модель задания в распределенных вычислительных средах

А.В. ШАМАКИНА, Л.Б. СОКОЛИНСКИЙ

ЮЖНО-УРАЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Цель

Разработать алгоритм планирования ресурсов в проблемно-ориентированных распределенных вычислительных средах.

Алгоритмы планирования ресурсов

1. Алгоритм доминирующей последовательности (DSC)
2. Алгоритм Саркара
3. Алгоритм Кима и Брауна (KB/L)

Требования к модели

- Представление потока работ в виде размеченного взвешенного ориентированного ациклического графа
- Возможность кластеризации вершин графа
- Возможность масштабирования отдельных задач в задании
- Указание числа процессорных ядер для вычислительных узлов
- Описание уже известных и новых алгоритмов кластеризации

Ориентированный граф

Ориентированным графом называется четверка

$$G = \langle V, E, init, fin \rangle,$$

где

V – множество вершин;

E – множество дуг;

$init: E \to V$ – функция, определяющая *начальную вершину* дуги;

$fin: E \to V$ – функция, определяющая *конечную вершину* дуги.

Функции взвешивания и разметки графа

Пусть задан ориентированный граф $G = \langle V, E, init, fin \rangle$.

- *Функция взвешивания* $\delta(e)$ дуги e определяет объем данных, который необходимо передать по дуге e от задачи, ассоциированной с вершиной $init(e)$ к задаче, ассоциированной с вершиной $fin(e)$.
- *Разметкой графа G* будем называть функцию $\gamma : V \to \mathrm{N}^2$

Граф задания

Графом задания называется размеченный взвешенный ориентированный ациклический граф

$$G = \langle V, E, init, fin, \delta, \gamma \rangle,$$

где

V – множество вершин, соответствующих задачам,

E – множество дуг, соответствующих потокам данных.

Пример графа задания

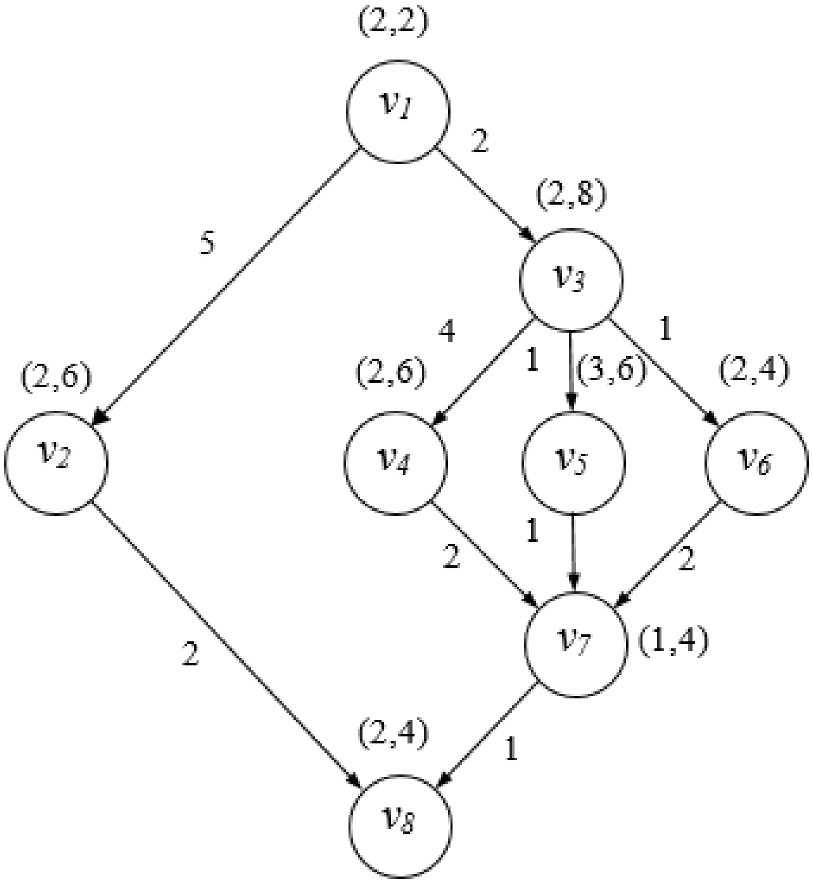
Функция разметки

$$\gamma(v) = (m_v, t_v),$$

где

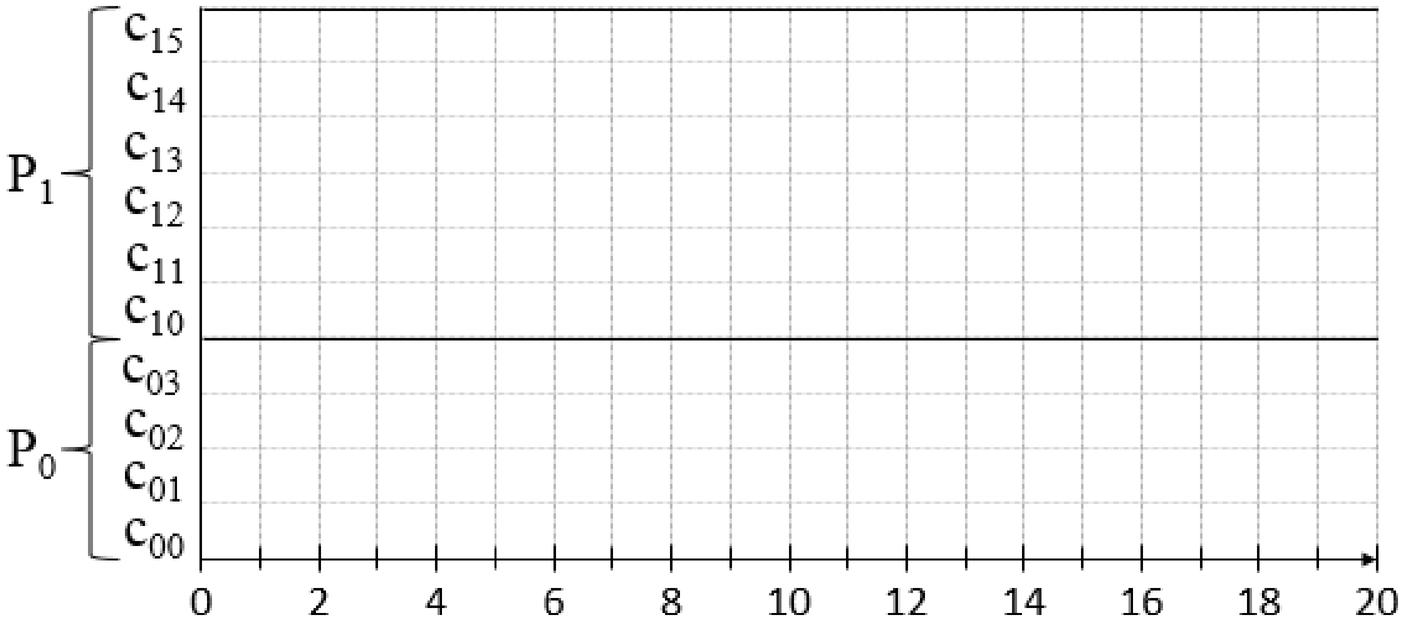
m_v - максимальное количество процессорных ядер, на которых задача v имеет ускорение, близкое к *линейному*,

t_v - время выполнения задачи v на одном ядре



Вычислительная система

- Вычислительный узел P – это упорядоченное множество процессорных ядер $\{c_0, \dots, c_{q-1}\}$.
- Вычислительная система – это упорядоченное множество вычислительных узлов $\mathfrak{P} = \{P_0, \dots, P_{k-1}\}$.



Функция кластеризации

- *Кластеризацией* называется однозначное отображение

$$\omega: V \to \mathfrak{P}$$

множества вершин V графа задания G на множество вычислительных узлов $\mathfrak{P}$.

- *Кластер* W_i – это подмножество всех вершин, отображаемых на вычислительный узел $P_i \in \mathfrak{P}$.

- Пример:
$W_0 = \{v_1, v_2, v_8\}$ и
$W_1 = \{v_3, v_4, v_5, v_6, v_7\}$.

Вершина v	$\omega(v)$
v_1	P_0
v_2	P_0
v_3	P_1
v_4	P_1
v_5	P_1
v_6	P_1
v_7	P_1
v_8	P_0

Пример кластеризованного графа

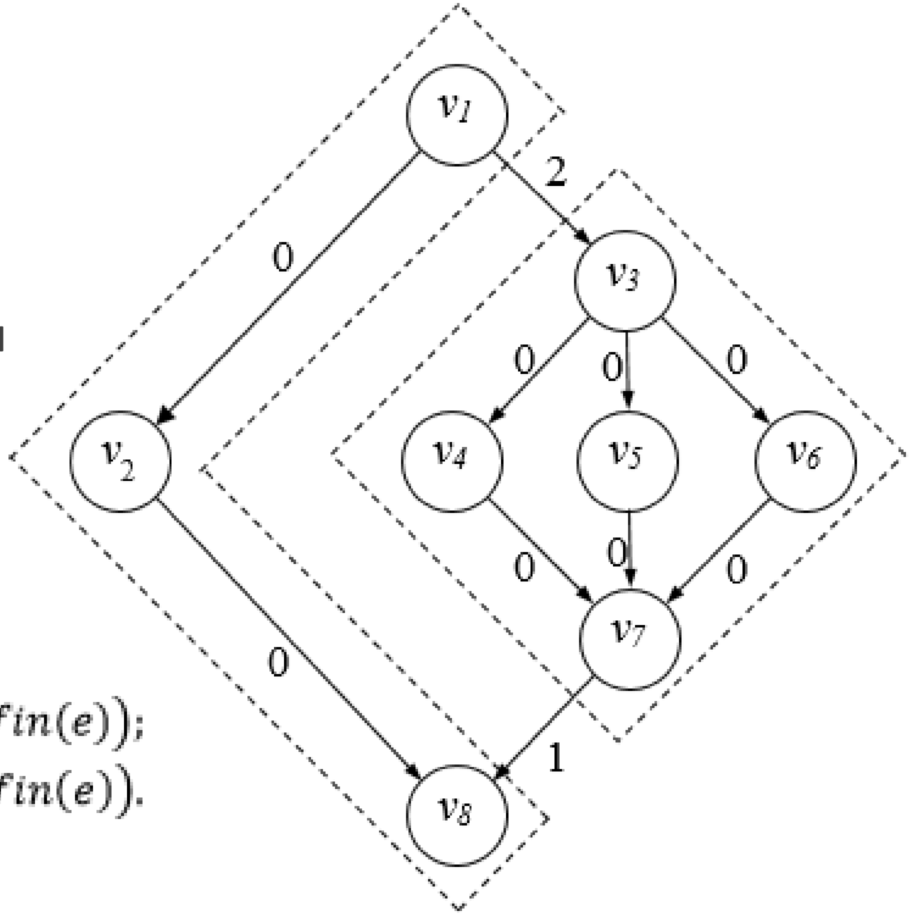
- Кластеры:

$W_0 = \{v_1, v_2, v_8\}$ и
$W_1 = \{v_3, v_4, v_5, v_6, v_7\}$.

- *Коммуникационная стоимость* – время передачи данных по дуге $e \in E$.

- Функция, вычисляющая коммуникационную стоимость:

$$\sigma(e) = \begin{cases} 0, & \text{если } \omega(init(e)) = \omega(fin(e)); \\ \delta(e), & \text{если } \omega(init(e)) \neq \omega(fin(e)). \end{cases}$$



Расписание

Расписанием называется отображение

$$\xi : V \to \mathbb{Z}_{\geq 0} \times \mathbb{N},$$

которое произвольной вершине $v \in V$ сопоставляет двойку чисел

$$\xi\xi(v) = (\tau_v, j_v),$$

где τ_v определяет время запуска задачи v, j_v – количество процессорных ядер, выделяемых задаче v.

Вершина $\boldsymbol{v}$	$\xi(v) = (\tau_v, j_v)$	
	Время запуска $\boldsymbol{\tau_v}$	Количество ядер $\boldsymbol{j_v}$
v_1	0	2
v_2	1	2
v_3	3	2
v_4	7	2
v_5	7	2
v_6	7	2
v_7	10	1
v_8	15	2

Вычислительная стоимость

Вычислительная стоимость $\chi(v, j_v)$ задачи v на j_v процессорных ядрах определяется следующей формулой:

$$\chi(v, j_v) = \begin{cases} t_v / j_v\,, & \text{если } 1 \le j \le m_v; \\ t_v / m_v\,, & \text{если } m_v < j_v. \end{cases}$$

Вершина v	$\chi(v, j_v)$
v_1	1
v_2	3
v_3	4
v_4	3
v_5	3
v_6	2
v_7	4
v_8	2

План $\alpha = \{\alpha_0, \alpha_1\}$ выполнения расписания ξ

Отображение $\alpha_0 \colon W_0 \to P_0$

Задача	Процессорные ядра
v_1	$\{c_{00}, c_{01}\}$
v_2	$\{c_{00}, c_{01}\}$
v_8	$\{c_{00}, c_{01}\}$

Отображение $\alpha_1 \colon W_1 \to P_1$

Задача	Процессорные ядра
v_3	$\{c_{10}, c_{11}\}$
v_4	$\{c_{10}, c_{11}\}$
v_5	$\{c_{12}, c_{13}\}$
v_6	$\{c_{14}, c_{15}\}$
v_7	$\{c_{10}\}$

Распланированный граф

- Кластеризованный граф G с заданными расписанием ξ и соответствующим планом α будем называть *распланированным*.

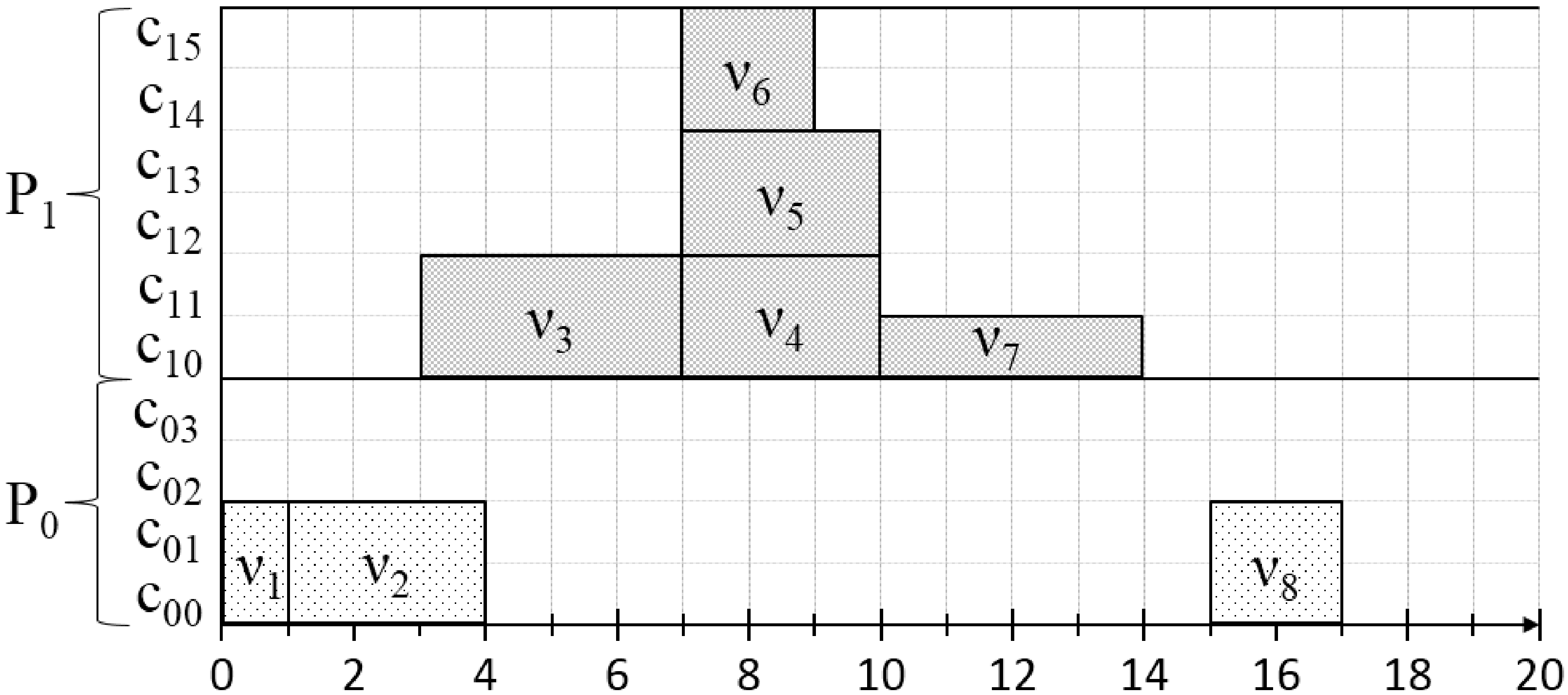
Критический путь

- Пусть в распланированном графе G задан простой путь $y = (e_1, e_2, \dots, e_n)$. *Стоимостью пути* $u(y)$ называется величина

$$u(y) = \chi\left(fin(e_n), j_{fin(e_n)}\right) + \sum_{i=1}^{n}\left(\chi\left(init(e_i), j_{init(e_i)}\right) + \max\left(\sigma(e_i), \tau_{fin(e_i)} - s_{init(e_i)}\right)\right)$$

- Пусть Y – множество всех простых путей в распланированном графе G. Простой путь $\bar{y} \in \mathrm{Y}$ называется *критическим*, если он обладает максимальной стоимостью.

Диаграмма Ганта



Критический путь равен 17.

Алгоритм планирования POS (Problem-Oriented Scheduling)

1. Построение начальной конфигураций графа $G_0 = \langle V, E, init, fin, \delta, \gamma, \omega_0, \xi_0 \rangle$.

2. Найти критический путь $\bar{y}_0$ в G_0.

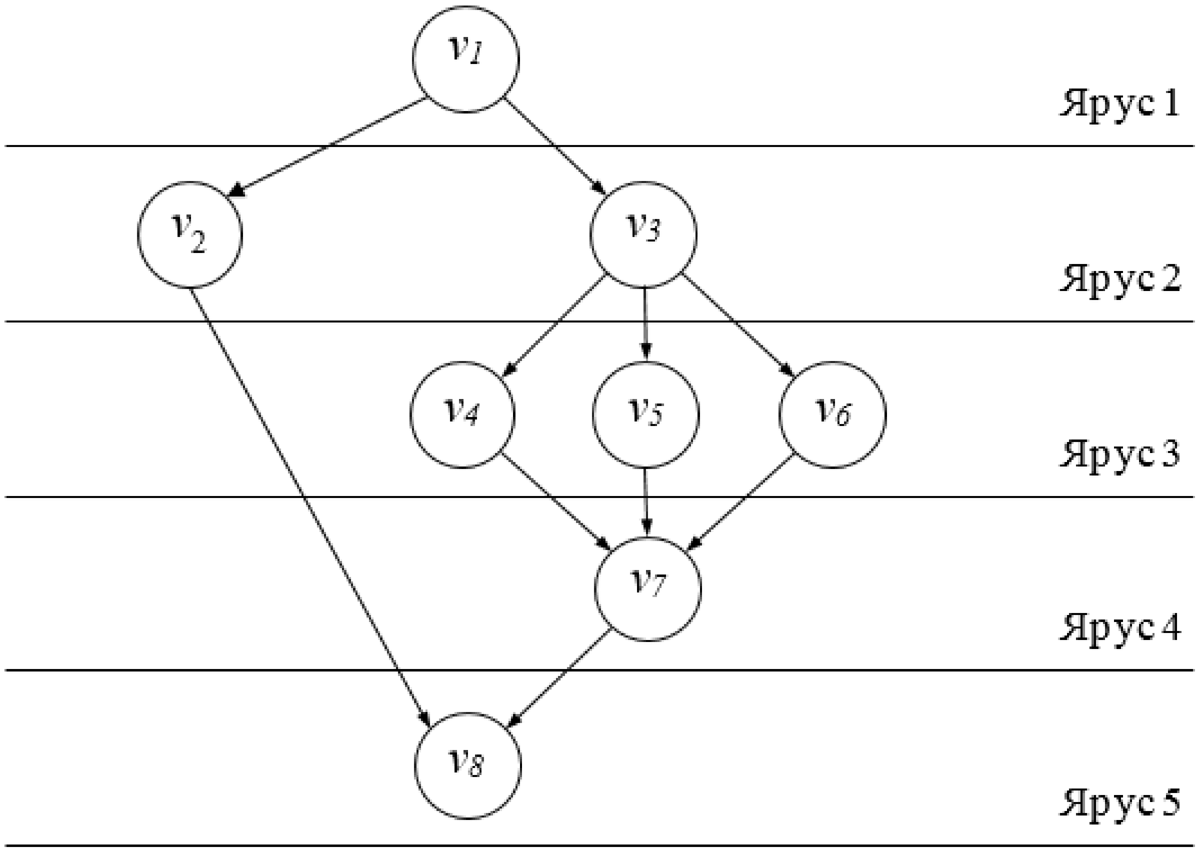
3. $i := 0$.

4. Перейти от конфигурации $G_i = \langle V, E, init, fin, \delta, \gamma, \omega_i, \xi_i \rangle$ к конфигурации $G_{i+1} = \langle V, E, init, fin, \delta, \gamma, \omega_{i+1}, \xi_{i+1} \rangle$, такой что (попутно помечая рассмотренные дуги).

5. Если остались нерассмотренные дуги, $i := i + 1$ и перейти на шаг 4.

6. Стоп.

Каноническая ярусно-параллельная форма графа



Заключение

Разработана алгоритм планирования ресурсов в проблемно-ориентированных распределенных вычислительных средах.