Алгоритм построения расписаний выполнения параллельных задач на группах кластеров с процессорами различной производительности и его анализ в среднем[1]

[1] *Д.О. Лазарев <dennis810@mail.ru>*
[1,2] *Н.Н. Кузюрин <nnkuz@ispras.ru>*

[1] *Институт системного программирования им. В.П. Иванникова РАН, 109004, Россия, г. Москва, ул. А. Солженицына, д. 25*
[2] *Московский физико-технический институт, 141700, Московская область, г. Долгопрудный, Институтский пер., 9*

Аннотация. В работе рассмотрена задача построения расписаний выполнения параллельных вычислительных задач на группах кластеров с одинаковым числом $\boldsymbol{w}$ одинаковых процессоров, производительность которых для разных кластеров различная. Проведён вероятностный анализ задачи. Получены нижние оценки. Показано, что если число процессоров, необходимых для решения любой задачи имеет равномерное распределение на отрезке $[\boldsymbol{o}, \boldsymbol{w}]$ для любого алгоритма составления расписаний величина математического ожидания свободного объёма вычислений равна $\boldsymbol{\Omega}(\boldsymbol{w}\sqrt{\boldsymbol{N}})$. Получены верхние оценки. Был предложен онлайновый алгоритм построения расписаний с распределением задач в ограниченные области Limited Hash Scheduling для задачи построения расписаний, работающий в режиме closed-end, с математическим ожиданием свободного объёма вычислений, равным $\boldsymbol{O}\left(\boldsymbol{w}\sqrt{\boldsymbol{N}\ \boldsymbol{ln}\ \boldsymbol{N}}\right)$.

Ключевые слова: построение расписаний; онлайновый алгоритм; режим closed-end; вероятностный анализ; процессоры различной производительности; алгоритм размещения задач в ограниченные области; Limited Hash Scheduling.

DOI: 10.15514/ISPRAS-2018-30(6)-6

Для цитирования: Лазарев Д.О., Кузюрин Н.Н. Алгоритм построения расписаний выполнения параллельных задач на группах кластеров с процессорами различной производительности и его анализ в среднем. Труды ИСП РАН, том 30, вып. 6, 2018 г., стр.105-122. DOI: 10.15514/ISPRAS-2018-30(6)-6

[1] Работа выполнена при финансовой поддержке РФФИ, проект 17-07-01006

1. Введение

Задача построения расписаний выполнения параллельных задач на группах кластеров с процессорами различной производительности (далее – задача построения расписаний), объединённых, например, сетями Grid[15] с одинаковым числом процессоров на кластерах ранее изучалась лишь в худшем случае. В случае однопроцессорных машин, в работе [1] был предложен алгоритм A, абсолютная точность R_A которого была не более 8. В работе [2] оценка абсолютной точности была улучшена до $3 + \sqrt{8} \approx 5.828$, а также был предложен рандомизированный алгоритм с абсолютной точностью $R_A \leq 4.311$.

Если представить каждую задачу в виде прямоугольника, ширине которого соответствует число процессоров, необходимых для решения задачи, а высоте прямоугольника без учёта сжатия- нормализованное время исполнения задачи(на кластере единичной производительности), то задача построения расписаний выполнения параллельных задач может рассматриваться, аналогично работе [10], как задача упаковки прямоугольников в несколько полубесконечных полос, в которой, попадая в полосу с номером j, прямоугольник сжимается по высоте в v_j раз, где v_j-скорость j-ой полосы. Только, ввиду того, что задача не обязательно должна занимать последовательные процессоры на кластере, прямоугольник может быть разбит по ширине на несколько меньших прямоугольников прямыми, параллельными боковым сторонам полос.

Таким образом, алгоритмы для задач упаковки в несколько полос могут оказаться эффективными для задачи построения расписаний. Так, например, любой алгоритм для задачи Multiple Strip Packing с полосами одинаковой ширины [6], [11] может использоваться для задачи построения расписаний для процессоров одинаковой производительности с одинаковым числом процессоров на кластере, а любой алгоритм для задачи упаковки прямоугольников в полосы различной ширины [12], [13], [14] может быть использован для задачи построения расписаний для процессоров одинаковой производительности с различными числами процессоров на кластерах. Обратное, однако, неверно.

В [3] С.Н. Жуку удалось получить оценку $R_A \leq 2e$ для задачи построения расписаний на группах кластеров с процессорами различной производительности с различным числом процессоров на кластерах, обобщив алгоритм из работы [4] для задачи упаковки прямоугольников в полосы разной ширины на случай задачи построения расписания на кластере.

В настоящей работе был построен онлайновый алгоритм для задачи построения расписаний Limited Hash Scheduling для N задач, с мат. ожиданием свободного объёма вычислений $E(V_{sp}) = O(w\sqrt{N \ln N})$. При построении алгоритма использовались идеи разбиения на области из работы [5] и упаковки не поместившихся в области прямоугольников из работы [6], а также

методы анализа в среднем алгоритмов из работы [7]. Так как предложенный алгоритм упаковывает каждую задачу на последовательные процессоры, то, положив все ускорения равными, можно свести классическую задачу Strip Packing к задаче построения расписаний, улучшив лучшую из известных оценок для задачи Strip Packing при анализе в среднем [8], [6] до $E(S_{sp}) = O(\sqrt{N \ln N})$.

Также было доказано, что при числе процессоров на каждом кластере, равном w и при числе процессоров, необходимых для решения каждой задачи, имеющем равномерное рапределение на $[o, w]$ любой алгоритм составления расписаний имеет математическое ожидание свободного объёма вычислений $E(V_{sp}) = \Omega(w\sqrt{N})$, где N- число задач, V_{sp} - свободный объём вычислений.

2. Постановка задачи

Имеется группа из k вычислительных кластеров с одинаковым числом процессоров w у каждого кластера и различной производительностью процессоров. Будем называть v_i ускорением кластера с номером i. Так, выполнение любой задачи T_j на кластере с номером i занимает в v_i раз меньше времени, чем нормализованное время выполнения этой задачи на кластере единичной производительности h_i. Требуется составить расписание выполнения задач без прерываний на k вычислительных кластерах, минимизирующее время L завершения выполнения последней задачи.

Будем рассматривать онлайновые алгоритмы, получающие задачи одну за другой и при распределении каждой задачи не знающие ни времени исполнения, ни числа процессоров следующих задач. Рассматриваем алгоритмы, работающие в режиме closed-end, т.е. знающие число задач N, для которых нужно составить расписание до получения первой задачи. Будем проводить вероятностный анализ в среднем случае в предположении, что числа процессоров, необходимых для решения каждой из задач w_i, $i \in \{1,2,\dots,N\}$- независимые в совокупности случайные величины, имеющие равномерное распределение на $[o, w]$, где w- число процессоров на каждом кластере. Предположим также, что нормализированное время выполнения каждой задачи на кластере единичной производительности h_i, $i \in \{1,2,\dots,N\}$- - также независимая случайная величина, имеющая равномерное на $[0,1]$ распределение.

Обозначим за L время завершения выполнения последней задачи, также L будем называть временем работы алгоритма (на данном наборе задач).

Для задачи i с числом процессоров w_i и нормализованным временем h_i, будем называть величину $w_i h_i$ объёмом вычислений.

За V_i обозначим суммарный объём вычислений по задачам, выполненным на вычислительном кластере с номером i.

За время не большее, чем L с ускорением w_i, на -ом кластере сумеем выполнить объём вычислений $V_i \leq v_i w L$. Свободным объём вычислений на кластере с номером i назовём $U_i = v_i w L - V_i \geq 0$.

За свободный объём вычислений примем сумму свободных объёмов вычислений по всем кластерам: $V_{sp} = \sum_{i=1}^{k} U_i$.

За W обозначим сумму объёмов вычислений для всех задач, которые надо распределить: $W = \sum_{i=1}^{N} h_i w_i$.

$$E(W) = \sum_{i=1}^{N} E(h_i)E(w_i) = \frac{Nw}{4}$$

За L' обозначим минимальное время, за которые задачи вычислительным объёмом W могут быть посчитаны на группе данных вычислительных устройств: $L' = \frac{W}{w\sum_{i=1}^{k} v_i}$.

За $\bar{L}$ обозначим минимальное время, за которые задачи вычислительным объёмом EW могут быть посчитаны на группе данных вычислительных устройств: $\bar{L} = \frac{E(W)}{w\sum_{i=1}^{k} v_i} = \frac{N}{4\sum_{i=1}^{k} v_i}$.

Будем оценивать качество работы алгоритма путём анализа величины $E(V_{sp})$.

Так как $L \geq L'$, то $L \geq \frac{W}{w\sum_{i=1}^{k} v_i}$, следовательно, $E(L) \geq \frac{E(W)}{w\sum_{i=1}^{k} v_i} = \frac{N}{4\sum_{i=1}^{k} v_i} = L$, поскольку $E(W) = \frac{Nw}{4}$.

Пусть время выполнения задач есть L. Тогда $E(V_{sp}) = E(L)w\sum_{i=1}^{k} v_i - E(W) = E(L - \bar{L})w\sum_{i=1}^{k} v_i$. Так как $EW = \bar{L}w\sum_{i=1}^{k} v_i$, то

$$\frac{E(V_{sp})}{E(W)} = \frac{E(L-\bar{L})}{\bar{L}} \quad (1)$$

Отсюда, доказательство того, что $E(V_{sp}) = O(w\sqrt{N \ln N})$, равносильно доказательству того, что $E(L) = \bar{L}(1 + O(\sqrt{\frac{\ln N}{N}}))$:

$$E(V_{sp}) = O(w\sqrt{N \ln N}) \Leftrightarrow E(L) = \bar{L}(1 + O(\sqrt{\frac{\ln N}{N}}))$$

3. Нижние оценки $E(V_{sp})$

Для получения нижнних оценок докажем следующую лемму:

Лемма 1. Пусть 2 случайные величины X и Y имеют непрерывные плотности распределений f_X и f_Y , причём X и Y - симметричны относительно своих мат. ожиданий и принимают лишь неотрицательные значения. Тогда $X + Y$ - тоже имеет непрерывную функцию плотности распределения и симметрична относительно своего мат. ожидания.

Доказательство. $f_{X+Y}(t) = \int_o^t f_X(\tau) f_Y(t - \tau) d\tau \Rightarrow f_{X+Y}(t)$ – непрерывна.
Без ограничения общности рассуждений предположим, что $E(X) \leq E(Y), t \leq \leq E(X) + E(Y)$.
Покажем, что $f_{X+Y}(t) = \int_o^t f_X(\tau) f_Y(t - \tau) d\tau = \int_o^t f_{X+Y}(2E(X) + 2E(Y) - t)(\tau) d\tau = \int_o^{2E(X)+2E(Y)-t} f_X(\tau) f_Y(2E(X) + 2E(Y) - t - \tau) d\tau$.

Посмотрим, при каких τ $f_X(\tau) f_Y(2E(X) + 2E(Y) - t - \tau)$ не равно нулю тождественно.

$$0 \leq \tau \leq 2E(X) \cap 0 \leq 2E(X) + 2E(Y) - t - \tau \leq 2E(Y) \Leftrightarrow 2E(X) - t \leq \leq \tau \leq 2E(X), \text{значит } f_{X+Y}(2E(X) + 2E(Y) - t) =$$

$$= \int_{2EX-t}^{2EX} f_X(\tau) f_Y(2E(X) + 2E(Y) - t - \tau) d\tau =$$

$$= (X \text{ и } Y - \text{симметричны относительно своих мат. ожиданий}) =$$

$$= \int_{2EX-t}^{2EX} f_X(2EX - \tau) f_Y(-2E(X) + t + \tau) d\tau = (\alpha = 2EX - \tau) =$$

$$= \int_o^t f_X(\alpha) f_Y(t - \alpha) d\alpha = f_{X+Y}(t) \blacksquare$$

Теорема 1.(О нижних оценках для задачи о составлении расписаний). При любом способе распределения N задач на группе кластеров с w процессорами различных скоростей, нормализованные времена исполнения которых - независимые в совокупности случайные величины, равномерно распределённые на $[0,1]$ и числа процессоров, занимаемые задачами- независимые в совокупности случайные величины, равномерно распределённые на $[o, w]$,

$$E(V_{sp}) = \Omega(\sqrt{N})$$

Обозначения.
За $W_{>w/2}$ обозначим суммарный объём вычислений для всех задач, занимающих больше, чем $\frac{w}{2}$ процессоров, а за $W_{\leq w/2}$ обозначим суммарный объём вычислений для всех задач, занимающих не более, чем $\frac{w}{2}$ процессоров.
За $h_{>w/2}$ обозначим суммарное нормализованное время исполнения для всех задач, занимающих больше, чем $\frac{w}{2}$ процессоров, а за $h_{\leq w/2}$ обозначим суммарное нормализованное время исполнения для всех задач, занимающих не более, чем $\frac{w}{2}$ процессоров.
За M обозначим число задач, число процессоров, необходимых для решения которых $w_i \geq \frac{w}{2}$.

Пусть $\bar{M} = \frac{N}{2} + \frac{\sqrt{N}}{2}$, где N- число задач.

Доказательство.

Если удастся показать, что

$$\exists \epsilon_0 : \mathbb{P}\left\{ wh_{>\frac{w}{2}} - W \geq \frac{w\sqrt{N}}{8} \right\} \geq \epsilon_0, \qquad (2)$$

то с вероятностью ϵ_0, $V_{sp} \geq \frac{w\sqrt{N}}{8}$, значит, $E(W) \geq \epsilon_0 w\sqrt{N}/8 = \Omega(w\sqrt{N})$.

Рассмотрим случайное событие Z, получающееся в результате двух испытаний:

1. Одно испытание Z_0 в схеме Бернулли, возвращающееся 0 или 1 с одинаковой вероятностью.
2. Если после первого испытания $Z_0 = 0$, то $Z = U(0, w/2]$, иначе, $Z = U(w/2, w]$, где за $U(a,b]$ обозначена случайная величина, имеющая равномерное распределение на полуинтервале $(a,b]$.

Плотность вероятности случайной величины Z поточечно равна плотности распределения равномерно распределённой на $(0,w]$ случайной величины $U(0,w]$, следовательно, $Z = U(0,w]$.

Поэтому будем рассматривать следующую последовательность испытаний I, в результате которой нам будут известны все параметры задач:

1. Серия испытаний Бернулли из N испытаний с вероятностью $\frac{1}{2}$ того, что число процессоров $w_i \leq \frac{w}{2}$ и с вероятностью $\frac{1}{2}$ того, что число процессоров $w_i > \frac{w}{2}$, $i \in \{1, \dots, N\}$.
2. Выбираем нормализованное время исполнения каждой из задач h_i, имеющее равномерное распределение на $[0,1], h_1, \dots, h_N$ - независимы в совокупности.
3. Для всех $1 \leq i \leq N$: из первого испытания $w_i \leq \frac{w}{2}$, выбираем w_i, как случайную велиину, равномерно распределённую на полуинтервале $(o, w/2]$, а для всех осталных i выбираем w_i, как случайную велиину, равномерно распределённую на полуинтервале $(w/2, w]$.

По интегральной теореме Лапласа-Муавра,

$$\mathbb{P}\left\{ M \geq \frac{N}{2} + \frac{\sqrt{N}}{2} \right\} \geq \mathbb{P}\left\{ 1 \leq \frac{M - \frac{N}{2}}{\frac{\sqrt{N}}{2}} \leq 2 \right\} \geq (N > N_0) \geq$$

$$\geq \frac{1}{\sqrt{2\pi}} \int_1^2 e^{-\frac{x^2}{2}} dx \geq 1/30$$

Пусть $M \geq \bar{M} = \frac{N}{2} + \frac{\sqrt{N}}{2}$. Рассмотрим $h_{w/2}$. Эта случайная величина равна сумме по всем M задачам, занимающим более $\frac{w}{2}$ процессоров, нормализованных времён исполнения этих задач,- случайных симметричных

относительно мат. ожидания случайных величин $U(0,1]$, имеющих равномерное распределение на $(0,1]$. Значит, по Лемме 1, $h_{>w/2}$ - имеет симметричное относительно своего математического ожидания, равного $\frac{M}{2}$, распределение, следовательно,

$$\mathbb{P}\left\{h_{>\frac{w}{2}} \geq \frac{M}{2}\right\} = \frac{1}{2}$$

Аналогично,

$$\mathbb{P}\left\{h_{\leq\frac{w}{2}} \leq \frac{N-M}{2}\right\} = \frac{1}{2}$$

Предположим, что после первых 2 шагов в последовательности испытаний I верно событие A: $\left(M \geq \frac{N}{2} + \frac{\sqrt{N}}{2}\right) \cap \left(h_{>\frac{w}{2}} \geq \frac{M}{2}\right) \cap \left(h_{\leq\frac{w}{2}} \leq \frac{N-M}{2}\right)$. $\mathbb{P}(A) \geq \frac{1}{120}$.

Тогда $W_{\frac{w}{2}} = \sum_{i:w_i>\frac{w}{2}} h_i w_i$. При выбранном h_i и равномерно распределённом на $(w/2, w]w_i$, случайная величина, равная $h_i w_i$- симметрична относительно своего мат. ожидания, равного $\frac{3wh_i}{4}$. Тогда, по Лемме 1, $W_{>w/2}$- симметрична относительно своего мат. ожидания, более или равного $\frac{3wh_i}{4}$ и $\mathbb{P}\left(W_{>\frac{w}{2}} \leq \frac{3wh_{\geq w/2}}{4}\right) \geq \frac{1}{2}$. Аналогично, $\mathbb{P}\left(W_{\leq\frac{w}{2}} \leq \frac{w(N-M)}{8}\right) \geq \frac{1}{2}$. Тогда обозначим за B следующее случайное событие:

$$B = \left(M \geq \frac{N}{2} + \frac{\sqrt{N}}{2}\right) \cap \left(h_{>\frac{w}{2}} \geq \frac{M}{2}\right) \cap \left(h_{\leq\frac{w}{2}} \leq \frac{N-M}{2}\right) \cap \left(W_{>\frac{w}{2}} \leq \frac{3wh_{>w/2}}{4}\right) \cap \left(W_{\leq\frac{w}{2}} \leq \frac{w(N-M)}{8}\right)$$

$$\mathbb{P}(B) \geq \frac{1}{480} = \epsilon_0$$

Если B - верно, то

$$\left(W = W_{\leq\frac{w}{2}} + W_{>\frac{w}{2}} \leq \frac{w\left(N + 6h_{>w/2} - M\right)}{8}\right) \cap \left(h_{>\frac{w}{2}} \geq \frac{M}{2}\right) \Rightarrow wh_{>\frac{w}{2}} - W \geq \frac{w\left((8-6)h_{>w/2} - N + M\right)}{8} \geq \frac{w\sqrt{N}}{8}$$

Значит, верна формула 2, и, стало быть, верно, что

$$E(V_{sp}) \geq \frac{1}{3840} w\sqrt{N} = \Omega\left(w\sqrt{N}\right) \blacksquare$$

4. Алгоритм Limited Hash Scheduling для задачи построения расписаний

Обозначения.

$$v_i = \frac{v_i}{2\sum_{i=1}^{k} v_i}\ , i \in \{1, \dots, k\}$$

$$\bar{L} = \frac{EW}{w\ \sum_{i=1}^{k} v_i} = \frac{N}{4\sum_{i=1}^{k} v_i}$$

$$s = \left\lfloor \frac{\sqrt{N}}{k} \right\rfloor$$

4.1 Разбиение на области

Определение 1. Назовём вычислительной областью, или просто областью некоторое подмножество пространства $\prod$ = время × процессоры. Будем говорить, что задача кладётся на верх текущей упаковки в вычислительной области, если задача начинает исполняться на процессорах, принадлежащих области во всё время исполнения задачи, в самый ранний момент времени из возможных после завершения всех предыдущих задач, исполняемых внутри данной области.

Предположим без ограничения общности рассуждений, что $v_1 \geq v_2 \geq \cdots \geq v_k$. Тогда рассмотрим вычислительную область $\prod_1$, равную произведению временного отрезка $[0, \bar{L}]$ на все процессоры всех кластеров. Разделим $\prod_1$ по времени на s меньших непересекающихся вычислительных областей с числом процессоров wk и временем исполнения $\frac{\bar{L}}{s}$. В итоге получим sk вычислительных областей шириной w и временем $[(i-1)\frac{\bar{L}}{s}, i\frac{\bar{L}}{s}]$, $\forall i \in \{1, \dots, k\}$, по s областей для каждого вычислительного кластера.

Области кластера с ускорением v_1, в свою очередь, разделим по ширине на 2 подобласти, состоящие из двух областей шириной

$$\frac{w\ i\ v_1}{s} \text{ и } w - \frac{w\ i\ v_1}{s}, \qquad \forall i \in \{1, \dots, s\}$$

Области кластера с ускорением $v_l, \forall l = \{1, \dots, k\}$, разделим по ширине на 2 подобласти, состоящие из двух областей шириной

$$\frac{w\ i\ v_l}{s} + w\sum_{i=1}^{l-1} v_i \text{ и } w - \left(\frac{w\ i\ v_l}{s} + w\sum_{i=1}^{l} v_i\right), \qquad \forall i \in \{1, \dots, s\}$$

Всего получим $2sk$ областей.

Ниже разбиение на области проиллюстрировано на рисунке 1, на котором полосам соответствуют кластеры с соответствующим ускорением(speeding), координате x- число процессоров, координате y- время(работы вычислительных кластеров).

4.2 Алгоритм упаковки в области Limited Hash Scheduling

Определение 2. Если после завеhшения выполнения предыдущих задач исполняемых в области, задачу можно исполнить на процессорах, в каждый момент времени исполнения задачи принадлежащих области, то говорим, что задача исполнима в области.

Определение 3. Скажем, что задача исполняется на верху текущей упаковки некоторой вычислительного кластера, если она начинает исполняться в самый ранний момент времени, но не ранее завершения выполнения всех областей(см. разбиение из предыдущего подраздела, т.е. не ранее, чем $\bar{L}$) и не ранее завершения выполнения всех задач, исполняемых на данном вычислительном кластере.

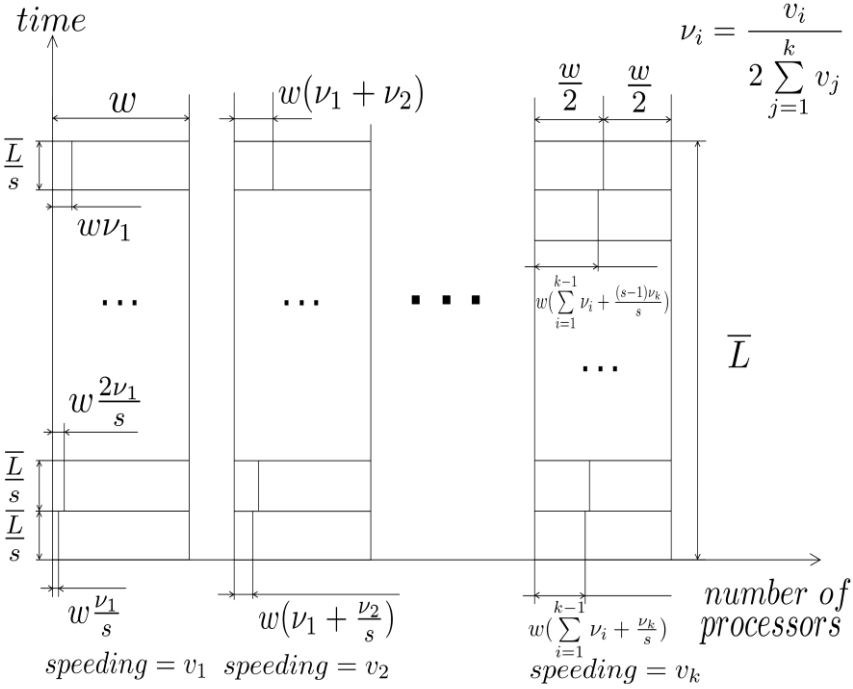

Рис. 1. Разбиение вычислительной области $\prod_1$ на $2sk$ меньших областей
Fig. 1. Hashing $\prod_1$ into $2sk$ areas with smaller volume of computations.

Алгоритм составления расписаний с использованием ограниченных областей Limited Hash Scheduling:

1. Если очередная задача исполнима в одной из $2sk$ областей, полученных в предыдущем подразделе(при условии исполнения предыдщих задач согласно алгоритму), то кладём задачу на верх текущей упаковки в области с минимальным числом процессором, в которой данная задача исполнима.

2. Иначе задача называется **выпавшей**, либо **попавшей в переполнение** и исполняется на верху текущей упаковки одного из тех вычислительных кластеров, при исполнении на которых время завершения задачи - самое меньшее из возможных.

5. Верхние оценки $E(V_{sp})$ для алгоритма Limited Hash Scheduling

В работе [7] была показана следующая Лемма, являющаяся усилением неравенства Азумы в случае определённым образом распределённых случайных величин.

Лемма. (Трушников, 2013.) Пусть случайная величина $X = X_1 + X_2 + \cdots + X_N$, где $X_i = \xi_i \eta_i$, ξ_i принимает значение 1 с вероятностью p и 0 с вероятностью $1 - p$, η_i — равномерно распределенная на отрезке (0,1] случайная величина, причем все случайные величины $\xi_i, \eta_i, i \in \{1, \dots, N\}$ независимы в совокупности. Тогда для любого α из интервала (0,1] выполняется неравенство

$$\mathbb{P}\{X > (1+\alpha)EX\} \le e^{-\frac{5}{9}\alpha^2 E(X)}$$

Докажем следующую лемму, покрывающую случаи $\alpha > 1$:

Лемма 2. В обозначениях предыдущей леммы, для любого α из интервала (0, 1] выполняется неравенство

$$\mathbb{P}\{X > (1+\alpha)EX\} \le e^{-\frac{5}{9}\alpha^2 E(X)}$$

,а при α > 1 верно, что

$$\mathbb{P}\{X > (1+\alpha)EX\} \le e^{-\frac{1}{3}\alpha E(X)}$$

Доказательство. Нужно доказать утверждение в случае $\alpha > 1$. В работе [7] было доказано, что

$$\mathbb{P}\{X > (1+\alpha)EX\} \le \exp\left(pN\left(\frac{(e^t - 1)}{t} - 1 - \frac{t}{2}(1+\alpha)\right)\right) \quad \forall t > 0$$

Выберем $t = 1$. Тогда

$$\mathbb{P}\{X > (1+\alpha)EX\} \le \exp\left(pN\left(e - 1 - 1 - \frac{1}{2}(1+\alpha)\right)\right)$$

$$\le \exp\left(pN\left(e - \frac{5}{2} - \frac{1}{2}\alpha\right)\right) \le \exp\left(pN\left(-\frac{1}{6}\alpha\right)\right) = e^{-\frac{1}{3}\alpha EX} \quad \blacksquare$$

Отсюда выведем следующую лемму:

Лемма 3. Пусть случайная величина $X = X_1 + X_2 + \cdots + X_N$, где $X_i = \xi_i \eta_i$, ξ_i принимает значение 1 с вероятностью p и 0 с вероятностью $1 - p$, η_i — равномерно распределенная на отрезке (0,1] случайная величина, причем все случайные величины $\xi_i, \eta_i, i \in \{1, \dots, N\}$ независимы в совокупности. Тогда при всех $N \ge N_0$

$$\mathbb{P}\{X > EX + 2\sqrt{N \ln N}\} \le N^{-3}$$

Доказательство.

1. $EX > 2\sqrt{N \ln N}$. Тогда по первой части Леммы 2,

$$\mathbb{P}\{X > EX + 2\sqrt{N \ln N}\} = \mathbb{P}\left\{X > EX\left(1 + \frac{2\sqrt{N \ln N}}{EX}\right)\right\}$$
$$\le \exp\left(-4\frac{5}{9}\frac{N \ln N}{E^2 X}EX\right) \le \left(\text{т. к. } EX < \frac{N}{2}\right) \le \exp\left(\frac{-40 \ln N}{9}\right) \le N^{-3}$$

2. $EX \le 2\sqrt{N \ln N}$. Тогда по второй части Леммы 2,

$$\mathbb{P}\{X > EX + 2\sqrt{N \ln N}\} = \mathbb{P}\left\{X > EX\left(1 + \frac{2\sqrt{N \ln N}}{EX}\right)\right\}$$
$$\le \exp\left(-\frac{2\sqrt{N\, ln\, N}}{3}\right) \le (\forall N \ge N_0) \le N^{-3} \quad \blacksquare$$

Упорядочим области по числу процессоров в порядке убывания: пусть первая область содержит $u_1 = w\left(1 - \frac{v_1}{s}\right)$ процессоров, …, область с номером $2sk$ содержит $u_{2sk} = w\frac{v_1}{s}$ процессоров. Скажем, что область с номером i имеет ускорение v_{n_i}.

Определние 4. За $H_i, i = \{1, \dots, 2sk\}$ обозначим сумму времён исполнения всех задач, число процессоров, необходимых для решения которых не превосходит ширину самой широкой области u_1, но строго больше, чем ширина $i-1$-ой области u_{i-1}: $H_i = \sum_{(1 \le j \le N) \cap (u_{i-1} < w_j \le u_1)} h_j$, где w_j и h_j- число процессоров и нормализованное время, необходимые для решения задачи номер j;

За H_0 обозначим сумму времён исполнения всех задач, число процессоров, необходимых для решения которых строго больше, чем ширина 1-ой области u_1: $H_i = \sum_{(1 \le j \le N) \cap (w_j > u_1)} h_j$.

Определние 5. За l_i обозначим сумму времён i самых широких областей, умноженную на ускорение того кластера, в вычислительном пространстве которой данная область лежит: $l_i = \sum_{j=1}^{i} v_{n_j}$.

Напомним, что задача, не исполнимая ни в одной из областей при условии составления расписания для предыдущих задач согласно с алгоритмом Limited Hash Scheduling, называется выпавшей задачей.

Определние 6. За H обозначим суммарное время исполнения всех выпавших задач.

Лемма 4.

$$H \le H_0 + \max\{0, \max_i\{H_i - l_i + i\}\}$$

Доказательство. Назовём область числом процессоров u_i не переполненной, если все задачи, для решения которых нужно не больше, чем u_i процессоров, упакованы в области числом процессоров $\leq u_i$.
Остальные области назовём переполненными.
Если область i ускорением v_{n_i}- переполнена, то некая задача, требующая не более, чем u_i процессоров, не исполнима ни в одной из областей шириной $\leq u_i$. Значит, суммарная нормализованная высота задач, исполняемых в -ой области не менее, чем $v_i \frac{\bar{L}}{s} - 1$.
Пусть i_0- номер не переполненной области с самым большим числом процессоров. Если все области - переполнены, то определим $i_0 = 2sk + 1$ и $u_i = 0$. Заметим, что все задачи с $w_i < u_{i_0}$ - не выпавшие. Если $i_0 = 1$, то $H = H_0$. Если $i_0 > 1$, в каждую из областей с номером $i < i_0$, попали задачи суммарным временем выполнения не менее, чем $v_i \frac{\bar{L}}{s} - 1$. Значит, $H \leq H_0 + (H_{i_0-1} - l_{i_0-1} + i_0 - 1) \leq H_0 + \max\{0, \max_i\{H_i - l_i + i\}\}$ ■

Предложение 1. После завершения работы алгоритма Limited Hash Scheduling при $H > 0$,

$$L \leq \bar{L} + \frac{H}{\sum_{i=1}^{k} v_i} + \frac{k}{\sum_{i=1}^{k} v_i},$$

где L- самое позднее время завершения задачи на одном из вычислительных кластеров, $\bar{L} = \frac{N}{4\sum_{i=1}^{k} v_i}$- время начала выполнения выпавших задач, H- суммарное время исполнения всех выпавших задач.

Обозначение.
За α_i, $i \in \{1, \dots, k\}$ обозначим $\max\{\bar{L}$, время завершения последней задачи на вычислителном кластере с номером $i\}$.

Доказательство. $H = \sum_{i=1}^{k} (\alpha_i - \bar{L}) v_i$. Заметим, что

$$L \leq \bar{L} + \frac{H}{\sum_{i=1}^{k} v_i} + \frac{k}{\sum_{i=1}^{k} v_i} \Leftrightarrow \sum_{i=1}^{k} (L - \alpha_i) v_i \leq k$$

Докажем последнее неравенство индукцией по числу попавших в переполнение задач, что $\sum_{i=1}^{k} (L^j - \alpha_i^j) v_i \leq k$, где L^j и α_i^j- промежуточные значения L и α_i после попадания в переполнение j задач.
База индукции при $j = 0$ очевидна: $0 \leq k$.
Переход. Пусть $\sum_{i=1}^{k} (L^{j-1} - \alpha_i^{j-1}) v_i \leq k$. Рассмотрим 2 случая:
1. $L_j = L_{j-1}$. Тогда $\sum_{i=1}^{k} (L^j - \alpha_i^j) v_i \leq \sum_{i=1}^{k} \left(L^{j-1} - \alpha_i^{j-1}\right) v_i \leq k$
2. $L_j > L_{j-1}$. Тогда из того, что, в соответствии с алгоритмом Limited Hash Scheduling, при вычислении задачи с номером j на верху текущей упаковки в том кластере, в который алгоритм размещает задачу, время завершения задачи минимальное и из того, $h_j \leq 1$, следует, что

$$L^j - \alpha_i^j \le \frac{1}{v_i} \ \forall 1 \le i \le k \Rightarrow \sum_{i=1}^{k} \left(L^{j-1} - \alpha_i^{j-1}\right) v_i \le k \quad \blacksquare$$

Предложение 2. $E(H_i) \le l_i \ \forall i \in \{1, \dots, k\}$ (см. определения 4 и 5.)

Доказательство. Пусть $\left[\frac{i}{s}\right] = m, i - ms = p.$

$$l_i = s\frac{\bar{L}}{s}\sum_{i=1}^{m} v_i + \frac{\bar{L}}{s}\sum_{i=1}^{p} v_{m+1} = \frac{\frac{N}{4}\left(\sum_{i=1}^{m} v_i + \frac{p}{s} v_{m+1}\right)}{\sum_{i=1}^{k} v_i}, \qquad i \le sk$$

$E(H_i) = \frac{N}{2}\,\mathbb{P}(u_{i-1} \le w_1 \le u_1)$, где w_1 - имеет равномерное распределение на $[0, w]$.

$$E(H_i) = \frac{N}{2}\frac{v_1\,\frac{s-1}{s} + \sum_{i=1}^{m} v_i + \frac{p+1}{s} v_{m+1}}{2\sum_{i=1}^{k} v_i} = l_i - \frac{N}{4\sum_{i=1}^{k} v_i}\frac{v_1 - v_{m+1}}{s}$$

$$\le l_i, \text{т. к. } v_1 \ge v_{m+1}, \text{при всех } i \le sk$$

Аналогично доказывается при $i > sk$, что $E(H_i) \le l_i \quad \blacksquare$

Теорема 2 (Верхние оценки $E(V_{sp})$ для алгоритма Limited Hash Scheduling). При составлении расписаний согласно алгоритму Limited Hash Scheduling, для числа вычислительных кластеров $k \le \sqrt{N}$ ускорением процессоров самого быстрого кластера $v_1 \le \sqrt{\ln N}\frac{\sum_{i=1}^{k} v_i}{k}$, верно, что для всех $N > N_0 \in \mathbb{N}$ математическое ожидание свободного объёма вычислений

$$E(V_{sp}) \le 4w\sqrt{N \ln N} = O(w\sqrt{N \ln N}),$$

где w- число процессоров на каждом из вычислительных кластеров.

Доказательство.

$$E\left(V_{sp}\right) = E\left(Lw\sum_{i=1}^{k} v_i - W\right) = E\left(Lw\sum_{i=1}^{k} v_i\right) - E(W) = E(L - \bar{L})w\sum_{i=1}^{k} v_i \quad (3),$$

где V_{sp}- свободный объём вычислений, L- время завершения исполнения последней задачи, W- суммарный объём вычислений по всем задачам, $\bar{L} = \frac{N}{4\sum_{i=1}^{k} v_i}$.

По предложению 1,

$$(L - \bar{L})w\sum_{i=1}^{k} v_i \le wH + wk \Rightarrow E(L - \bar{L})w\sum_{i=1}^{k} v_i \le E(wH) + E(kH), \qquad (4)$$

где H - суммарное время выполнения выпавших задач.

По Лемме 4, так как в силу выбора $s, \frac{\sqrt{N}}{k} \le s < 2\frac{\sqrt{N}}{k}$

$H \le H_0 + \max\{0, \max_i\{H_i - l_i + i\}\} \Rightarrow EH \le EH_0 +$

$$+E(\max\{0, \max_i\{H_i - l_i\}) + i \le (i \le 2sk)$$
$$\le E(H_0) + E(\max\{0, \max_i\{H_i - l_i\}\}) + 4\sqrt{N} \quad (5)$$

(см. определения 4 и 5)

В последней формуле заметим, что $E(H_0) = \frac{N}{2}\mathbb{P}\left(w_i \ge w\left(1 - \frac{v_1}{2s\sum_{i=1}^{k} v_i}\right)\right) = \frac{N}{2}\frac{v_1}{2s\sum_{i=1}^{k} v_i}$. По условию теоремы, $v_1 \le \sqrt{\ln N}\frac{\sum_{i=1}^{k} v_i}{k} \Rightarrow E(H_0) \le \frac{N}{2}\frac{\sqrt{\ln N}}{2sk} < \frac{\sqrt{N \ln N}}{2}$.

Также по предложению 2, выполнено, что $E(H_i) \le l_i$, следовательно, по лемме 3, $\forall i$, $\mathbb{P}(H_i \ge l_i + 2\sqrt{N \ln N}) \le \mathbb{P}(H_i \ge E(H_i) + 2\sqrt{N \ln N}) \le N^{-3}$. Значит, $\mathbb{P}\{\max_i\{H_i - l_i\} \ge 2\sqrt{N \ln N}\} \le N^{-2}$

Стало быть, $E(max\{0, \max_i\{H_i - l_i\}\}) \le 2\sqrt{N \ln N} + O(N^{-1})$. Подставим полученные результаты в формулу 5:

$$E(H) = E(H_0) + E(\max\{0, \max_i\{H_i - l_i\}\}) + 4\sqrt{N}$$
$$\le \sqrt{N \ln N}/2 + 2\sqrt{N \ln N} + O(N^{-1}) + 4\sqrt{N}$$

Подставляя полученный результат в формулы 3 и 4, имеем:

$$E(V_{sp}) \le E(wH) + kw \le w\left(\frac{\sqrt{N \ln N}}{2} + 2\sqrt{N \ln N} + O(N^{-1}) + 4\sqrt{N}\right) + kw \le$$
$$w\left(\frac{\sqrt{N \ln N}}{2} + 2\sqrt{N \ln N} + O(N^{-1}) + 5\sqrt{N}\right) \le (N > N_0) \le 4w\sqrt{N \ln N}$$
$$= O(w\sqrt{N \ln N}) \quad \blacksquare$$

6. Заключение и направление дальнейших исследований

Была рассмотрена задача построения расписаний выполнения вычислительных задач на группах вычислительных кластеров с одинаковым числом w одинаковых процессоров, производительность которых для разных кластеров различная. Число процессоров, необходимых для решения задач является случайной величиной, имеющих равномерное на $[0, w]$ распределение. Был проведён вероятностный анализ. Для любого алгоритма построения расписаний была получена оценка снизу на математическое ожидание свободного объёма вычислений $E(V_{sp}) = \Omega(w\sqrt{N})$.

Был предложен алгоритм Limited Hash Scheduling размещения задач в ограниченные области с $E(V_{sp}) = \Omega(w\sqrt{Nln\,N})$, работающий в режиме closed-end, то есть знающий число задач N до начала построения расписания. Алгоритм с меньшей шириной областей также может быть применён для числа процессоров , необходимых для решения каждой задачи, равномерно распределённом на $[o, u]$, при $w \vdots u$. В общем случае при $u < v$, даже для

задачи Bin Packing не известно онлайнового алгоритма с математическим ожиданием незаполненного пространства контейнерои $E(L_{sp}) = o(N)$ [9].

Большой интерес представляет исследование задачи в случае алгоритмов, работающих в режиме open-end, т.е. не имеющих никаких данных о числе задач, для которых требуется построить расписание. В работе [9] для задачи Bin Packing упаковки объектов размером, имеющим равномерное распределение на $[0, u]$, $u \leq 1$ в контейнеры единичного размера при анализе в среднем в режиме open-end $E(L_{sp}) = \Omega(\sqrt{N})$. Интересен вопрос о возможности получении оценки $E(V_{sp}) = \Omega(w\sqrt{N})$ для любого онлайнового алгоритма, работающего в режиме open-end для задачи построения расписаний.

Список литературы

[1]. Aspens J., Azar Y., Fiat A., Plotkin S., Waarts O. On - line load balancing with applications to machine scheduling and virtual circuit routing. In Proc. of the 25th ACM STOC. 1993. pp. 623 - 631, DOI: 10.1145/167088.167248

[2]. Berman P., Charikar M., Karpinski M. On-line load balancing for related machines. LNCS, v. 1272, 1997, pp. 116-125, DOI: https://doi.org/10.1007/3-540-63307-3_52

[3]. С.Н. Жук. О построении расписаний выполнения параллельных задач на гуппах кластеров с различной производительностью. Труды ИСП РАН, том 23, 2012, стр. 447-454, DOI: 10.15514/ISPRAS-2012-23-27

[4]. S.N. Zhuk. Approximate algorithms for packing rectangles into several strips. Discrete Mathematics and Applications, vol 18, issue 1, 2006, pp. 91-105, DOI: https://doi.org/10.1515/156939206776241264

[5]. М.А. Трушников. Об одной задаче Коффмана-Шора, связанной с упаковкой прямоугольников в полосу. Труды ИСП РАН, том 22, 2012, стр. 456-462, DOI: 10.15514/ISPRAS-2012-22-24

[6]. Лазарев Д. О., Кузюрин Н. Н. Алгоритм упаковки прямоугольников в несколько полос и анализ его точности в среднем. Труды ИСП РАН, том 29, вып. 6, 2017 г., стр. 221-228, DOI: 10.15514/ ISPRAS-2017-29(6)-13

[7]. Трушников М. А. Вероятностный анализ нового алгоритма упаковки прямоугольников в полосу., Труды ИСП РАН, том 24, 2013, стр. 457-468, DOI: 10.15514/ISPRAS-2013-24-21

[8]. Лазарев Д.О., Кузюрин Н.Н. Об онлайновых алгоритмах для задач упаковки в контейнеры и полосы, их анализе в худшем случае и в среднем. Труды ИСП РАН, том 30, вып. 4, 2018 г., стр. 209-230. DOI:10.15514/ISPRAS-2018-30(4)-14

[9]. E. G. Coffman, C. Courcoubetis, M. R. Garey, D. S. Johnson, L. A. McGeoch, P. W. Shor, R. R. Weber, M. Yannakakis. Fundamental discrepancies between average-case analyses under discrete and continuous distributions: a bin packing case study. In Proc. of the twenty-third annual ACM Symposium on Theory of computing, 1991, pp. 230-240, doi:10.1145/103418.103446

[10]. Кузюрин Н., Грушин Д., Фомин С. Проблемы двумерной упаковки и задачи оптимизации в распределенных вычислительных системах. Труды ИСП РАН, том 26, вып. 1, 2014 г., стр. 483–502, DOI: 10.15514/ISPRAS-2014-26(1)-21

[11]. Ye D., Han X., Zhang G. Online multiple-strip packing. Theoretical Computer Science, Volume 412, Issue 3, 2011, pp. 233-239. DOI: https://doi.org/10.1016/j.tcs.2009.09.29

[12]. Жук С.Н. Онлайновый алгоритм упаковки прямоугольников в несколько полос с гарантированными оценками точности. Труды ИСП РАН, том 12, 2007 г., стр. 7-16, ISSN 2079-8056

[13]. Zhuk S.N. On-line algorithms for packing rectangles into several strips. Discrete Mathematics and Applications, vol. 17, issue 5, 2007, pp. 517-531. DOI: https://doi.org/10.1515/dma.2007.040

[14]. Жук С.Н. О построении расписаний выполнения параллельных задач на группах кластеров с различной производительностью. Труды ИСП РАН, том 23, 2012 г., стр. 447-454. DOI: https://doi.org/10.15514/ISPRAS-2012-23-27

[15]. Foster I., Kesselman C. The grid: Blueprint for a new computing infrastructure. Morgan Kaufmann Publishers Inc., 1999, 748 p.

On-line algorithm for scheduling parallel tasks on related computational clusters with processors of different capacities and its average-case analysis.

[1] *D.O. Lazarev <dennis810@mail.ru>*
[1,2] *N.N. Kuzyurin <nnkuz@ispras.ru>*

[1] *Ivannikov Institute for System Programming of the Russian Academy of Sciences, 25, Alexander Solzhenitsyn st., Moscow, 109004, Russia*
[2] *Moscow Institute of Physics and Technology (State University), 9 Institutskiy per., Dolgoprudny, Moscow Region, 141701, Russia*

Abstract. In this article the on-line problem of scheduling parallel tasks on related computational clusters with processors of different capacities was studied in average case. In the problem the objective is to make a schedule on $\boldsymbol{k}$ clusters with $\boldsymbol{w}$ processors each of $\boldsymbol{N}$ tasks, where the task $\boldsymbol{i}, \boldsymbol{i} \leq \boldsymbol{N}$ requires time $\boldsymbol{h_i}$ on cluster with nominal capacity of processors and $\boldsymbol{w_i} \leq \boldsymbol{w}$ processors. We presume for all $\boldsymbol{1} \leq \boldsymbol{i} \leq \boldsymbol{N}$ that $\boldsymbol{w_i}$ has uniform distribution on $(\boldsymbol{0}, \boldsymbol{w}]$ and that $\boldsymbol{h_i}$ has uniform distribution on $(\boldsymbol{0}, \boldsymbol{1}]$. The processors on different clusters have different capacities $\boldsymbol{v_1}, \dots, \boldsymbol{v_k}$. The task with nominal time $\boldsymbol{h_i}$ will require $\boldsymbol{w_i}$ processors be computed in time $\frac{\boldsymbol{h_i}}{\boldsymbol{v_j}}$ on cluster number $\boldsymbol{j}$. Let sum volume of computations $\boldsymbol{W}$ be the sum of volumes of computations for each task: $\boldsymbol{W} = \sum_{\boldsymbol{i=1}}^{\boldsymbol{N}} \boldsymbol{w_i h_i}$. Let $\boldsymbol{L}$ be the minimal time at which all clusters will compute all the tasks, assigned to them, where each task is assigned to one cluster. The expected value of free volume of computations $\boldsymbol{E}(\boldsymbol{V_{sp}})$is used to analyze the quality of an algorithm, where $\boldsymbol{V_{sp}} = \sum_{\boldsymbol{1 \leq i \leq k}} \boldsymbol{v_i}\ \boldsymbol{L} - \boldsymbol{W}$. It was shown that for every algorithm for scheduling parallel tasks on related clusters $\boldsymbol{E}(\boldsymbol{V_{sp}}) = \boldsymbol{\Omega}(\boldsymbol{w}\sqrt{\boldsymbol{N}})$. An on-line algorithm Limited Hash Scheduling was proposed, which has $\boldsymbol{E}(\boldsymbol{V_{sp}}) \leq \boldsymbol{4}\left(\boldsymbol{w}\sqrt{\boldsymbol{N}\ln \boldsymbol{N}}\right) = \boldsymbol{O}\left(\boldsymbol{w}\sqrt{\boldsymbol{N}\ln \boldsymbol{N}}\right), \boldsymbol{for}\ \boldsymbol{N} > \boldsymbol{N_0} \in \mathbb{N}$ if $\boldsymbol{k} \leq \sqrt{\boldsymbol{N}}$ and $\boldsymbol{v_j} \leq \sqrt{\ln \boldsymbol{N}} \frac{\sum_{i=1}^{k} v_i}{\boldsymbol{k}}\ \forall\ \boldsymbol{1} \leq \boldsymbol{j} \leq \boldsymbol{k}$. The idea of the algorithm is to schedule tasks of close required number of required processors into different limited in time and the number of allowed to use processors areas on clusters.

Keywords: on-line algorithm; closed-end; probabilistic analysis; processors of different capacities; scheduling using limited computational areas; Limited Hash Scheduling.

DOI: 10.15514/ISPRAS-2018-30(6)-6

For citation: Lazarev D.O., Kuzyurin N.N. On-line algorithm for scheduling parallel tasks on related computational clusters with processors of different capacities and it's average-case analysis. Trudy ISP RAN/Proc. ISP RAS, vol. 30, issue 6, 2018, pp. 105-122 (in Russian). DOI: 10.15514/ISPRAS-2018-30(6)-6

References

[1]. M. Aspens J., Azar Y., Fiat A., Plotkin S., Waarts O. On - line load balancing with applications to machine scheduling and virtual circuit routing. In Proc. of the 25th ACM STOC. 1993. pp. 623 - 631, DOI: 10.1145/167088.167248

[2]. Berman P., Charikar M., Karpinski M. On-line load balancing for related machines. LNCS, v. 1272, 1997, pp. 116-125, DOI: https://doi.org/10.1007/3-540-63307-3_52

[3]. Zhuk S.N. On-line algorithm for scheduling parallel tasks on a group of related clusters. Trudy ISP RAN/Proc. ISP RAS, vol. 23, 2012, pp. 447-454 (in Russian). DOI: 10.15514/ISPRAS-2012-23-27

[4]. S.N. Zhuk. Approximate algorithms for packing rectangles into several strips. Discrete Mathematics and Applications, 2006, vol 18, issue 1, pp. 91-105, DOI: https://doi.org/10.1515/156939206776241264

[5]. M. A. Trushnikov. On one problem of Koffman-Shor connected to strip packing problem. Trudy ISP RAN/Proc. ISP RAS, vol. 22, 2012, pp. 456-462 (in Russian). DOI: 10.15514/ISPRAS-2012-22-24 p.

[6]. Lazarev D.O., Kuzyrin N.N. An algorithm for Multiple Strip Package and its average case evaluation. Trudy ISP RAN/Proc. ISP RAS, vol. 29, issue 6, 2017. pp. 221-228 (in Russian). DOI: 10.15514/ISPRAS-2017-29(6)-1

[7]. M. A. Trushnikov. Probabilistic analysis of a new strip packing algorithm. Trudy ISP RAN/Proc. ISP RAS, vol. 24, 2013, pp. 457-468 (in Russian). DOI: 10.15514/ISPRAS-2013-24-21

[8]. Lazarev D.O., Kuzjurin N.N. On on-line algorithms for Bin, Strip and Box Packing, and their worst- and average-case analysis. Trudy ISP RAN/Proc. ISP RAS, vol. 30, issue 4, 2018. pp. 209-230 (in Russian). DOI: 10.15514/ISPRAS-2018-30(4)-14

[9]. E. G. Coffman, C. Courcoubetis, M. R. Garey, D. S. Johnson, L. A. McGeoch, P. W. Shor, R. R. Weber, M. Yannakakis. Fundamental discrepancies between average-case analyses under discrete and continuous distributions: a bin packing case study. In Proc. of the twenty-third annual ACM Symposium on Theory of computing, 1991, pp. 230-240, doi:10.1145/103418.103446

[10]. N.N. Kuzyurin, D.A. Grushin, S.A. Fomin. Two-dimensional packing problems and optimization in distributed computing systems. Trudy ISP RAS/Proc. ISP RAS. 2014, vol. 26, issue 1, 2015, pp. 483–502 (in Russian). DOI: 10.15514/ISPRAS-2014-26(1)-21

[11]. Ye D., Han X., Zhang G. Online multiple-strip packing. Theoretical Computer Science, Volume 412, Issue 3, 2011, pp. 233-239. DOI: https://doi.org/10.1016/j.tcs.2009.09.29

[12]. Zhuk S.N. Online algorithm for packing rectangles into several strips with guaranteed accuracy estimates. Trudy ISP RAN/Proc. ISP RAS, vol. 12, 2007, pp. 7-16 (in Russian). ISSN 2079-8056

[13]. Zhuk S.N. On-line algorithms for packing rectangles into several strips. Discrete Mathematics and Applications, vol. 17, issue 5, 2007, pp. 517-531. DOI: https://doi.org/10.1515/dma.2007.040

[14]. Zhuk S.N. On-line algorithm for scheduling parallel tasks on a group of related clusters. Trudy ISP RAN/Proc. ISP RAS, vol. 23, 2012, pp. 447-454 (in Russian). DOI: https://doi.org/10.15514/ISPRAS-2012-23-27

[15]. Foster I., Kesselman C. The grid: Blueprint for a new computing infrastructure. Morgan Kaufmann Publishers Inc., 1999, 748 p.