

Проблемы двумерной упаковки и задачи оптимизации в распределенных вычислительных системах¹

*Н.Н. Кузюрин {nknuz@ispras.ru}, Д.А. Грешин {grushin@ispras.ru},
С.А. Фомин {fomin@ispras.ru}*

Аннотация. В статье рассмотрена задача управления потоками параллельных программ на группе вычислительных кластеров и формализации этого процесса в виде оптимизационной задачи упаковки набора прямоугольников в группу полубесконечных полос различной ширины (Multiple Strip Packing). Приводятся современные результаты по решению этой задачи и ряд открытых проблем. Рассмотрены практические аспекты оптимизации управления потоками параллельных задач с различными критериями оценки их качества, дается описание созданной в ИСП РАН системы моделирования предназначенной для экспериментального исследования алгоритмов управления, описаны ее возможности.

Ключевые слова: параллельные задачи, группы кластеров, оптимизация, расписания, моделирование

1. Введение

В последние годы во всем мире происходит значительный рост потребности в вычислительных ресурсах. Если раньше суперкомпьютеры были крайне дороги и доступны единицам, то с появлением вычислительных кластеров, собранных из общедоступных компонентов, наука и промышленность получили в своё распоряжение простой и недорогой способ использования высокопроизводительных вычислений.

Типичный вычислительный кластер (Beowulf кластер) состоит из широко распространённого аппаратного обеспечения и работает под управлением операционной системы GNU/Linux или FreeBSD. Если кластер предназначен для использования многими пользователями, то управление кластером осуществляет менеджер ресурсов. Пользователи отправляют свои задания

менеджеру, который ставит их в очередь и, по мере высвобождения вычислительных узлов, осуществляет запуск заданий.

От количества поступающих задач зависит, сколько узлов кластера будет занято выполнением задач, а сколько простаивать. Согласно статистике большинство кластеров испытывает периодическую нагрузку – когда интенсивность потока задач различается в несколько раз в разное время суток. Это означает, что даже при относительно плотной загрузке заданиями в среднем, существуют периоды, когда большая часть узлов кластера не выполняет заданий и простаивает.

Исторически проблемы управления потоками задач с целью их оптимального распределения по процессорам рассматривались в теории расписаний [1]. Классический алгоритм Грэхема размещения очередной задачи на наименее загруженную из m машин является первым примером приближенного алгоритма построения расписания с гарантированной оценкой точности. Задачи и машины в этой постановке были однопроцессорными, а алгоритм обладал важным свойством – работал в режиме он-лайн, т.е. очередная задача размещалась без знания всех последующих, т.е. по мере поступления.

С широким распространением кластеров актуальной стала задача распределения по процессорам параллельных задач. При этом каждая задача занимает некоторое число процессоров на время вычисления. С проблемой оптимизации выполнения заданного набора задач на кластере тесно связана классическая задача двумерной упаковки, так называемая strip packing problem. Она заключается в размещении множества прямоугольников внутри полубесконечной вертикальной полосы единичной ширины, при этом стороны прямоугольников должны быть параллельны сторонам полосы (вращения запрещены) и прямоугольники не должны пересекаться. Нужно минимизировать «высоту упаковки» – расстояние от основания полосы до верхней грани верхнего прямоугольника в упаковке.

Эта задача рассматривалась с 1980 г. [2]. Она допускает естественную интерпретацию, в которой каждый прямоугольник – вычислительная задача, ширина прямоугольника соответствует количеству процессоров необходимых для вычисления задачи, высота – времени. Единственное отличие задачи упаковки от задачи составления расписания для кластера состоит в том, что в задаче упаковки процессоры выделяемые для задачи имеют последовательные номера. Поскольку эта задача NP-трудна интерес представляют приближенные полиномиальные алгоритмы. Особый интерес представляют «онлайновые» алгоритмы, размещающие прямоугольники в полосе по мере их поступления, без знания параметров всех последующих прямоугольников.

Алгоритмы упаковки в полосу обычно оцениваются по худшему случаю (worst-case analysis) или в среднем (average-case analysis). При анализе по худшему случаю минимизируют «высоту упаковки». При анализе в среднем

¹ Работа поддержана грантом РФФИ 11-07-00610а.

минимизируемой функцией является математическое ожидание незаполненной прямоугольниками площади полосы от основания полосы до верхней грани самого верхнего прямоугольника в упаковке. При этом ширины и высоты всех прямоугольников являются независимыми в совокупности равномерно распределенными на отрезке $[0,1]$ случайными величинами.

2. Некоторые модели и теоретические результаты

2.1. Упаковка прямоугольников в полосу

Даже для случая одной полосы, рассматриваемая задача является NP-трудной, поскольку ее частными случаями являются задача об упаковке в контейнеры и задача о расписании для m машин [1]. Поэтому изучают приближенные алгоритмы для этой задачи и для оценки их точности используют следующие характеристики. Пусть $H_O(T)$ и $H_A(T)$ – высота оптимального размещения прямоугольников и высота размещения, полученная с помощью алгоритма A . Всюду далее мы предполагаем, что высота каждого прямоугольника не превосходит 1. Тогда мультипликативная ошибка приближенного алгоритма A определяется как

$$R_A = \sup_T \{H_A(T)/H_O(T)\},$$

а асимптотическая мультипликативная ошибка как

$$R_A^\infty = \lim_{k \rightarrow \infty} \sup_T \{H_A(T)/H_O(T) \mid H_O(T) \geq k\}.$$

Особый интерес представляют так называемые онлайн-алгоритмы. Это означает, что множество T рассматривается как последовательность, члены которой подаются на вход алгоритма один за другим, причем размещение очередного прямоугольника из T в полосу производится без какой-либо информации о последующих членах последовательности.

Еще один важный частный случай задачи упаковки в полосу – это задача о составлении расписания для m машин, которая получается, если имеется полоса ширины m и все прямоугольники имеют ширину 1. В [1] для нее был предложен простой онлайн-алгоритм, который размещает каждую новую задачу на наименее загруженную машину. Мультипликативная ошибка этого алгоритма не превосходит $2 - 1/m$, для m машин. В [3] был предложен онлайн-алгоритм с мультипликативной ошибкой не более 1.986 для всех $m > 70$. Это значение было улучшено до 1.945, а затем и до 1.923 в работах [4] и [5]. В [6] было показано, что ни один детерминированный онлайн-алгоритм

не может иметь мультипликативной ошибку меньше, чем 1.837 для достаточно больших m . Этот результат был улучшен в [5] до 1.852.

В [7] был рассмотрен класс шельфовых алгоритмов и в этом классе предложен онлайн-алгоритм, имеющий асимптотическую мультипликативную ошибку 1.7. В [8] было доказано, что никакой шельфовый алгоритм не может иметь асимптотическую мультипликативную ошибку меньше, чем 1.69103, и построен шельфовый алгоритм, позволяющий сколь угодно близко подойти к данному значению. В [9] было показано, что никакой онлайн-алгоритм упаковки прямоугольников в одну полосу не позволяет добиться асимптотической мультипликативной ошибки меньше, чем 1.5401. В [10] получена верхняя оценка в классе онлайн-алгоритмов 1.5889.

2.2. Задача упаковки в несколько полос

В случае одной полосы фактически мы имеем дело с упаковкой параллельных вычислительных задач на одной многопроцессорной машине типа вычислительного кластера. С развитием Грид-вычислений появилась возможность управлять потоком задач распределяя их на различные группы кластеров. При этом кластеры могут иметь различное число процессоров. Далее мы рассмотрим интерпретацию распределения параллельных задач на группах кластеров в виде задачи упаковки прямоугольников (задач) в несколько полос (вообще говоря, различной ширины).

Задача упаковки прямоугольников в несколько полос заключается в следующем.

Входом является конечная последовательность открытых прямоугольников $T = \{T_1, \dots, T_n\}$, где $h(T_j)$ и $w(T_j)$ – соответственно высота и ширина прямоугольника T_j , и $C = \{C_1, \dots, C_m\}$ – множество полубесконечных вертикальных полос, где w_i – ширина i -й полосы.

Требуется найти ортогональное размещение последовательности прямоугольников T по этим полосам (без вращений и пересечений, стороны прямоугольников параллельны сторонам полос), минимизирующее полную высоту этого размещения, то есть максимум по всем прямоугольникам и по всем полосам расстояния от дна полосы до верхней грани прямоугольника.

Пусть $H_O(T, C)$ и $H_A(T, C)$ – высота оптимального размещения прямоугольников и высота размещения, полученная с помощью алгоритма A . Тогда асимптотическая мультипликативная ошибка приближенного алгоритма A определяется как

$$R_A^\infty = \limsup_{k \rightarrow \infty} \{H_A(T, C) / H_O(T, C) \mid H_O(T, C) \geq k\}.$$

Задача об упаковке прямоугольников в несколько полос до сих пор оставалась в значительной степени не исследованной. Мы начали ее исследование в 2003 году. В [11] был предложен приближенный алгоритм для нее с мультипликативной точностью 3. Однако, алгоритм не был онлайн-овым. Вскоре выяснилось, что многие естественные эвристики, работающие в режиме он-лайн упаковки в случае нескольких полос могут приводить к большой разбалансировке и не обеспечивать аппроксимацию с константной точностью [12,13]. Там же построены первые алгоритмы с гарантированными оценками с константной точности. Следует отметить, что в [14] также рассмотрена модель распределения параллельных задач в системе Грид на основе алгоритмов упаковки прямоугольников в полосу. Однако, алгоритмов распределения прямоугольников по полосам рассмотрено не было и не было получено результатов о точности алгоритмов упаковки.

В [15] был предложен алгоритм, размещающий прямоугольники по полосам в онлайн-овом режиме (но для размещения в полосе использовались не онлайн-овые алгоритмы), и имеющий мультипликативную ошибку не превышающую 10.

Далее e обозначает основание натурального логарифма. Следующие две теоремы доказаны в [16].

Теорема 1. Для любого онлайн-ового алгоритма A выполнено неравенство

$$R_A^\infty \geq e.$$

Пусть r – произвольная константа, такая что $0 < r < 1$. Построен онлайн-овый алгоритм A_r и доказана следующая теорема.

Теорема 2. Для алгоритма A_r справедливо неравенство

$$R_{A_r}^\infty \leq \frac{2e}{r}.$$

Ряд результатов, использующих специфику входных, данных был получен в [17,18,19].

В 2009 году задача упаковки прямоугольников в несколько полос привлекла внимание зарубежных ученых. Однако был рассмотрен и исследован только частный случай, в котором все полосы имеют одинаковую ширину. Этот случай в зарубежной литературе получил название Multiple Strip Packing (MSP).

Для алгоритмов, работающих в режиме off-line получены следующие результаты [20,21]: предложена асимптотически полностью полиномиальная

аппроксимационная схема (AFPTAS) и предложен приближенный алгоритм с оценкой точности $R_A \leq 2$.

Предложен также детерминированный алгоритм, работающий в режиме on-line с оценкой точности:

$$R_A \leq 3 + \delta_m,$$

и вероятностный алгоритм с оценкой

$$R_A \leq 2.7 + \delta_m,$$

$$\text{где } \delta_m \rightarrow 0, \text{ при } m \rightarrow \infty.$$

Здесь представляется перспективным анализ других частных случаев задачи упаковки в несколько полос, когда не все полосы имеют одинаковую ширину, но и не произвольную (например степени двойки и т.п.).

2.3. Вероятностный анализ алгоритмов упаковки

Отметим, что вероятностному анализу различных эвристик одно и двумерной упаковки посвящено много работ [22-25]. Целевой функцией в таком анализе для задачи упаковки в полосу обычно является величина незаполненной площади от дна полосы до до верхней грани прямоугольника.

Так называемые шельфовые алгоритмы были рассмотрены в [3] для случая одной полосы. Пусть задана некоторая константа $r \in (0,1)$. Шельф – это прямоугольная часть полосы, с шириной равной ширине полосы. Как только приходит новый прямоугольник R , шельфовый алгоритм определяет целое число k , такое что $r^{k+1} < h(R) \leq r^k$ и после этого он размещает данный прямоугольник в шельфе высотой r^k . Прямоугольник может быть размещен в один из существующих шельфов, или сверху может быть создан новый шельф высоты r^k . Прямоугольники внутри каждого шельфа размещаются по горизонтали в ряд. Таким образом, упаковка прямоугольников, попавших в один класс по высоте (которым соответствует одно значение k), производится с помощью некоторой эвристики, осуществляющей одномерную упаковку в контейнеры.

Мы будем называть алгоритмом $A(E)$ шельфовый алгоритм упаковки, который на втором этапе использует некоторую произвольную эвристику E . В [26-27] описан общий метод вероятностного анализа шельфовых алгоритмов $A(E)$, позволяющий для многих шельфовых алгоритмов оценивать ожидаемую незаполненную площадь, используя соответствующие результаты для одномерной эвристики E .

Мы будем рассматривать вероятностное распределение $U([0,1])$ – равномерное распределение на отрезке $[0,1]$. Всюду в дальнейшем будем считать, что для каждого прямоугольника высота h_i и ширина w_i имеют равномерное распределение на отрезке $[0,1]$. Будем предполагать, что все случайные величины w_i, h_i – независимы в совокупности. В дальнейшем будем обозначать через Σ математическое ожидание площади, не заполненной прямоугольниками, между основанием полосы и верхней границей самого верхнего шельфа. Будем предполагать, что число прямоугольников N – бесконечно большая величина ($N \rightarrow \infty$).

Пусть для одномерной эвристики **E** выполнено соотношение

$$E(L - \sum_{i=1}^N w_i) = O(f(N)),$$

где L – число шельфов, в которые эвристика **E** упаковывает набор отрезков $\{w_i\}$.

Через $A(E)$ будем обозначать шельфовый алгоритм, использующий эвристику **E**.

Теорема 3. [27] Пусть $E f(N) = N^\alpha \log^\beta N$, где $0 < \alpha < 1, \beta \geq 0$. Тогда для алгоритма $A(E)$ справедлива следующая оценка:

$$\Sigma = O\left(N^\alpha \left(\frac{\log^\beta N}{\delta^{1-\alpha}} + \delta N^{1-\alpha}\right)\right).$$

Выбирая $\delta = N^{(\alpha-1)/(2-\alpha)} \log^{(\beta/(2-\alpha))} N$, получаем оценку

$$\Sigma = O(N^{1/(2-\alpha)} \log^{(\beta/(2-\alpha))} N).$$

Отметим, в частности, что для трех важных эвристик упаковки в контейнеры известно: для **FF** $f(N) = N^{2/3}$, для **BF** $f(N) = N^{1/2} (\log N)^{3/4}$, и для наилучшей онлайн-эвристики (**best on-line**) **BO** $f(N) = (N \log N)^{1/2}$. Поэтому из теоремы сразу вытекает, что для шельфового алгоритма $A(\mathbf{FF})$ $\Sigma = O(N^{3/4})$, для $A(\mathbf{BF})$ $\Sigma = O(N^{2/3} (\log N)^{1/2})$, для $A(\mathbf{BO})$ $\Sigma = O(N^{2/3} (\log N)^{1/3})$.

Известно, что математическое ожидание незаполненной площади полосы у оптимальной упаковки есть $O(\sqrt{N})$, где N – число прямоугольников. В 1993 году Коффман и Шор предложили «офлайн-» (информация о всех прямоугольниках известна заранее) алгоритм [22], для которого по порядку

достигается оценка $O(\sqrt{N})$. В той же работе они предложили онлайн-алгоритм, требующий знания числа прямоугольников заранее, с оценкой $O(N^{2/3})$ для математического ожидания незаполненной площади полосы. Этот алгоритм относится к классу так называемых шельфовых алгоритмов, активно изучавшихся ранее.

В [28] предложен онлайн-алгоритм с той же оценкой, но не требующий знания числа прямоугольников заранее. Тем не менее вопрос о возможности улучшения оценки $O(N^{2/3})$ в классе онлайн-алгоритмов оставался открытым с 1993 г. Ранее были получены результаты о том, что в классе шельфовых алгоритмов (в котором была получена оценка Шором и Коффманом) существенно улучшить оценку качества упаковки невозможно.

Эта проблема была решена недавно М.А. Трушниковым [29], который предложил принципиально новый онлайн-алгоритм упаковки прямоугольников. Проведенные вычислительные эксперименты показали, что предложенный алгоритм обеспечивает верхнюю оценку $C\sqrt{N}$ для математического ожидания незаполненной площади полосы, что существенно лучше, чем у алгоритма Коффмана-Шора. При этом константа C равна примерно 1.5–1.6, в вычислительных экспериментах число прямоугольников N принимало значения от 80000 до 2000000000. К сожалению получить такую же оценку математически пока не удалось, однако удалось получить близкую к ней, а именно, $O(\sqrt{N} (\log N)^c)$ [30].

Сформулируем вспомогательную задачу, которая возникла в ходе анализа описанного алгоритма и положительное решение которой может позволить получить желаемую оценку $O(\sqrt{N})$ для нашего алгоритма.

Задача. Имеется n урн (они пронумерованы), каждая из которых может вместить ровно n шаров и есть n^2 шаров.

В начале процесса все урны пусты. На каждом шаге один из шаров попадает в любую урну с вероятностью n^{-1} .

- если урна не полна (в ней меньше n шаров), шар остается в ней;
- в противном случае он перемещается в урну с номером на единицу меньше. Если она не полна шар остается в ней, иначе перемещается в урну с меньшим номером;
- если шар попадает в урну с номером 1 и она полна, шар выпадает (не попадает ни в одну из урн).

Вопрос: Верно ли, что математическое ожидание числа выпавших шаров есть величина $O(n)$?

Недавно нам удалось получить положительный ответ на этот вопрос. Доказательство и возможное применение этого утверждения для улучшенного анализа алгоритма из [29] мы оставляем для отдельной публикации.

Кроме того, представляется интересным предложить аналоги и обобщения алгоритма из [29] для случая нескольких полос при анализе в среднем. Исследования в этом направлении начаты в [31].

В настоящее время нам удалось доказать следующую теорему

Теорема 4. Для любого фиксированного m имеется полиномиальный онлайн-алгоритм упаковки в несколько полос гарантирующий при анализе в среднем дефект упаковки $O(N^{1/2}(\log N)^c)$.

Экспериментальные исследования показывают, что аналогичный результат справедлив и для более общего случая когда $m = O(N^{1/3})$.

2.4. Расписания для машин с различной производительностью

Поскольку в реальности вычислительные кластеры имеют не только различное число процессоров, но и процессоры в различных кластерах могут иметь различную производительность, возникает вопрос о распределении задач по кластерам и процессорам с учетом этого фактора. В теории расписаний этот случай описывается как случай *related machines*, когда все процессоры имеют различную производительность. Однако, там речь идет только об однопроцессорных задачах и даже в этом случае задача построения наилучших приближенных алгоритмов далека от полного решения. В заключение мы приведем интерпретацию распределения параллельных задач на группах кластеров с различными производительностями процессоров в виде задачи упаковки прямоугольников в несколько полос. К предыдущей нашей модели для каждой полосы добавим еще число (не менее единицы каждое), которое характеризует относительную производительность процессоров в кластере, соответствующем данной полосе. Таким образом, если прямоугольник помещается в полосу с приписанным числом d , то его высота уменьшается в d раз.

Для этой модели недавно удалось обобщить результаты из [16]. Доказано, что результаты о существовании онлайн-алгоритма гарантирующего асимптотическую точность сколь угодно близкую к $2e$ справедливы и для данной модели [32].

Этот результат весьма важен и является достаточно общим, поскольку даже для случая однопроцессорных задач проблема весьма нетривиальна и интенсивно исследовалась ранее. В теории расписаний этот случай называется *scheduling for related machines*. В данной постановке имеется m однопроцессорных машин, каждая машина имеет свою производительность и список однопроцессорных задач.

Для этой задачи в 1993 г. был предложен детерминированный онлайн-алгоритм с $R_A \leq 8$ и вероятностный алгоритм с $R_A \leq 2e$, а затем позднее предложен детерминированный онлайн-алгоритм с $R_A \leq 3 + \sqrt{8}$ и вероятностный алгоритм с $R_A \leq 4.31$ [33].

3. Система моделирования и экспериментальные исследования

3.1. Система моделирования

Поскольку получение аналитических результатов, касающихся анализа качества различных алгоритмов очень часто оказывается трудной задачей, то актуальным является создание и использование средств моделирования групп распределенных вычислительных систем. Такая система моделирования была создана в ИСП РАН [34].

Система моделирования позволяет оценивать поведение распределенных вычислительных систем в различных ситуациях, например:

- при изменении нагрузки: количества поступающих задач, их размерности, приоритета, периода поступления и т.д.;
- при отключении части вычислительных ресурсов или добавлении новых ресурсов;
- при увеличении количества передаваемых данных;
- при выходе из строя части коммуникационных каналов

При этом, оценка эффективности управления может проводиться по следующим наиболее популярным критериям [35]:

- минимизация среднего времени ожидания задачи в очереди;
- минимизация максимального времени выполнения группы задач (*makespan*);
- максимизация пропускной способности – числа завершенных задач в единицу времени;
- минимизация простоев процессоров
- и т.д.

Система расширяема и рассчитана на гибкое использование. Возможно моделировать различные Grid архитектуры: одно и двух-уровневые системы с одним или несколькими брокерами, добавлять хранилища данных, определять топологию сетевых соединений и т.д. Система включает в себя множество реализованных компонент, таких как брокер, кластер, поток задач и т.д. Кроме того, пользователи могут расширять систему, реализовывая свои собственные компоненты.

Поведение отдельных элементов моделируется с помощью конечных автоматов, что позволяет работать с моделями больших систем – порядка тысяч процессоров и более миллиона задач.

В системе поддерживается возможность проведения серии экспериментов, состоящей из последовательных запусков выполняемой модели с изменением некоторых параметров при каждом следующем запуске. Например, может изменяться поток задач, конфигурация кластеров, сетевых соединений и т.п. Это позволяет в рамках одного эксперимента посмотреть динамику изменения эффективности системы и определить узкие места.

В системе реализован удобный механизм обработки результатов моделирования. Результат выполнения модели хранится в отдельном профиле и может обрабатываться независимо. Пользователь может использовать свой шаблон для выбора и визуализации только необходимой в данный момент информации. Это позволяет нескольким исследователям провести моделирование один раз, а затем независимо анализировать полученную информацию.

Система также включает в себя редактор и анализатор записей потоков задач (workload) [36,37]. Запись потока представляет собой текстовый файл, каждая строка которого содержит характеристики отдельной задачи: время порождения, время запуска, общее время выполнения, количество занимаемых процессоров и т.д. Анализатор позволяет отобразить различные характеристики потока – количество задач, соотношение однопроцессорных и параллельных задач, график порождения задач во времени и т.п. С помощью редактора можно изменять поток – копировать и перемещать части потока, соединять несколько потоков в один, изменять характеристики группы задач и т.п. Также, редактор позволяет создавать синтетический поток по заданным параметрам.

Разработанная система применялась для моделирования поведения реально существующей Grid системы Sharcnet [38] при различных условиях.

Распределенная вычислительная система Sharcnet (Shared Hierarchical Academic Research Computing Network) – это консорциум из 16 колледжей и университетов в юго-западной части провинции Канады Онтарио, вычислительные ресурсы которых объединены высокоскоростной оптической сетью.

В качестве входных данных была использована запись реального потока задач (“workload“ поток), выполнявшихся на кластерах с декабря 2005 по январь 2007 года [37].

Результаты моделирования показали, что средние длины очередей в некоторых случаях отличаются почти в 100 раз. Это приводит к тому, что среднее время ожидания в очереди для различных кластеров отличается в несколько десятков раз.

Для проведения экспериментов, на основе данного потока мы создали несколько синтетических потоков. Необходимость использования синтетических потоков была обусловлена различными факторами:

- запись оригинального потока может содержать различные несистематические особенности и всплески, связанные с конкретными событиями, что делает его неподходящим для анализа и прогнозирования поведения данной системы, но эти особенности могут быть сглажены при создании синтетических потоков;
- оригинальные потоки часто имеют большой размер и поэтому не очень удобны для выявления локальных свойств алгоритмов;
- синтетические потоки могут позволить создавать новые ситуации, которых не было в исходных данных.

Для создания синтетических потоков могут быть использованы различные подходы. Полностью синтетические потоки иногда бывают удобны для отладки, однако для анализа поведения моделируемых систем более подходящим является использование синтетических потоков, основывающихся на записях оригинальных потоков [39].

Для создания таких синтетических потоков мы использовали следующий подход. Для того, чтобы определить поток задач

$$\{P_j\}_{j=1}^M$$

необходимо определить следующие параметры:

- R_j – промежуток времени между поступлением j и $j + 1$ задачами ($j = 1, 2, \dots, M - 1$);
- H_j – запрашиваемое время исполнения ($j = 1, 2, \dots, M$);
- W_j – запрашиваемое число процессоров ($j = 1, 2, \dots, M$).

На основе оригинального потока задач для этих параметров оцениваются кумулятивные функции распределения и первые моменты. Далее, для каждого из параметров подбирается функция распределения в виде свертки нескольких

распространенных функций распределения. Подбор осуществляется с помощью минимизации отклонения моментов и графиков функций по параметрам распределений в свертке и коэффициентам свертки.

Полученные таким образом распределения были использованы для генерирования нескольких синтетических потоков.

Мы применили разработанную систему моделирования и сравнили эффективность распределения задач в сети Sharcnet в оригинальном случае (без брокера) с распределением, получаемым с помощью брокера. Без брокера задачи поступали на кластеры в оригинальной последовательности, указанной в файле загрузки. В обоих случаях на каждом кластере использовалась реализация алгоритма Backfill.

Результаты показывают, что для данной вычислительной системы распределение потока однопроцессорных заданий через брокер дает значительный эффект: снижается среднее время ожидания заданий в очереди, а также происходит более равномерная загрузка вычислительных ресурсов. Мы исследовали алгоритмы распределения только однопроцессорных задач, поскольку, для проведения данных экспериментов нам была доступна информация только в виде записи потока задач. Из записи потока невозможно определить почему задача отправляется пользователем на тот или иной вычислительный ресурс. Причиной может быть архитектура системы, наличие специального программного обеспечения, личные пристрастия и т.п. Мы сделали предположение, что однопроцессорные задачи менее привязаны к конкретному кластеру, так как не зависят от среды передачи данных, которая установлена на кластере.

3.2. Оптимизация энергосбережения

В последние годы весьма популярным и важным критерием стал критерий энергоэффективности вычислительных систем. Одной из первых систем, проектировавшихся с эффективными параметрами энергопотребления, была система Green Destiny, разработанная в конце 2001 года в рамках проекта "Высокопроизводительные вычисления в небольших пространствах". Проведенные в рамках проекта Green Destiny работы получили продолжение в виде двух основных направлений:

- архитектура с низким энергопотреблением (статический подход);
- снижение энергопотребления программными средствами (динамический подход).

Статический подход предполагает построение вычислительной системы из компонент, обладающих низким энергопотреблением. Так, в Green Destiny и в проекте MegaScale был использован процессор Transmeta Crusoe. Наиболее заметной в данном классе систем с пониженным энергопотреблением можно

назвать систему IBM BlueGene/L, которая дебютировала в ноябре 2004 года как самый производительный и как самый энергоэффективный суперкомпьютер в списке Top500. Однако, архитектурный подход крайне дорог, так как в его основе лежит применение нестандартных компонент и инженерных решений.

Наиболее доступным для большинства существующих систем подходом к снижению энергопотребления является использование программных средств. Данный подход объединяет ряд методов, позволяющих достигнуть энергоэффективного выполнения потока задач за счет выборочного выключения или снижения производительности компонент вычислительной системы, когда они простаивают или загружены частично. Перечислим основные способы, основанные на использовании программных средств, позволяющие повысить энергоэффективность:

- выключение простаивающих компонент вычислительной системы;
- перераспределение вычислительных заданий по времени при условии наличия многотарифной схемы оплаты электроэнергии (например день-ночь). Тогда за счет повышения загрузки системы ночью, днем вычислительная нагрузка будет снижена и простаивающие компоненты вычислительной системы отключены. Общее количество потребленной электроэнергии не снижается, однако уменьшается ее стоимость, что также рассматривается как повышение энергоэффективности;
- программное управление производительностью компонент вычислительной системы. Современные процессоры и оперативная память имеют возможность динамически изменять свою частоту и рабочее напряжение. Такой механизм носит название DVS (dynamic voltage and frequency scaling). Основным принцип данного механизма заключается в том, что при понижении напряжения процессора время вычислений увеличивается, однако общее количество энергии потраченной на вычисления уменьшается.

Важный аспект динамического управления энергопотреблением состоит в том, что смена состояния системы (включение, изменение производительности и т.п.) имеет стоимость W , выраженную в дополнительном количестве потребленной энергии, задержки или потери производительности, что, вообще говоря, не гарантирует снижения энергозатрат при переводе системы в спящий режим в отсутствие работы и обратно по мере надобности. Например, если стоимость включения W и задачи поступают в систему последовательно, каждая следующая сразу после

завершения предыдущей, то выключение системы после каждой задачи приведет к дополнительному расходу энергии. Это делает задачу динамического управления энергопотреблением нетривиальной.

Задачи снижения энергопотребления вычислительных систем программными средствами в последние годы привлекают внимание многих исследователей. Предложены разные теоретические модели и постановки задач построения энергоэффективных алгоритмов [40–47]. Разрабатываются первые коммерческие системы, позволяющие кластерам более эффективно расходовать электроэнергию. Задачи, связанные с увеличением энергоэффективности компьютеров, составляют важную часть популярного направления, называемого за рубежом "Green Computing". Мы показали, что временный перевод простаивающих узлов кластера в спящий режим приводит к существенной экономии электроэнергии [40]. Также мы рассмотрели возможность снижения расхода электроэнергии для группы кластеров, находящихся под управлением одного менеджера ресурсов – брокера, который, получая поток заданий от пользователей, распределяет их между кластерами [41,42]. В такой системе существует несколько возможностей для экономии электроэнергии (как в количественном смысле, так и в денежном – снижая стоимость энергии):

- различная энергоэффективность (отношение производительности к энергопотреблению) кластеров;
- географическое положение кластеров. Стоимость энергии в разных регионах может существенно различаться.

Отправляя задачи на кластер, с более низкой ценой на электроэнергию, можно уменьшить общую стоимость энергии. Стоимость энергии различается также в разное время суток, что даёт дополнительные возможности выбора если кластеры находятся в разных часовых поясах.

Мы провели большое число экспериментов моделирования с различными входными данными для оценки возможного снижения общего количества энергии и её стоимости в подобной вычислительной системе. Предложенные в предыдущей работе алгоритмы управления потоком задач были пересмотрены и улучшены.

Результаты проведенных экспериментов показали, что стоимость энергии возможно снизить, однако при этом увеличивается среднее время ожидания в очереди. Конкретные значения зависят от плотности потока задач – чем больше плотность, тем меньше возможностей для выбора кластера, и тем меньше величина экономии [42].

Таким образом, мы считаем, что использование информации о стоимости энергии каждого кластера при распределении задач способно существенно снизить расходы на электроэнергию для владельца распределенной вычислительной системы. Однако, стоит отметить, что в реальной жизни

такое перераспределение задач не всегда возможно. Обычно дата-центры строятся для обслуживания пользователей в определенном регионе, чтобы уменьшить "время отклика". Также задачи могут работать с большими объемами локальных данных, которые не так просто переместить с одного кластера на другой. Несмотря на это многие компании уже используют как альтернативные источники энергии для питания своих дата-центров, так и особенности географического расположения для снижения расходов на электроэнергию [44,45].

4. Заключение

В статье описан предложенный в ИСП РАН новый подход к построению алгоритмов управления ресурсами распределенных вычислительных систем на основе геометрической модели упаковки в несколько полос. Эта модель не имела аналогов за рубежом, однако ее частный случай упаковки в одну полосу интенсивно исследовался. В 2009 году эту новую модель в частной постановке, когда все полосы имеют одинаковую ширину, начали исследовать за рубежом [20,21].

В ходе наших исследований удалось разработать алгоритмы упаковки с наилучшими среди известных оценками точности (на сегодняшний день), в частности, впервые получены гарантированные константные оценки балансировки нагрузки в режиме он-лайн. При этом разработанные алгоритмы распределения ресурсов – двухуровневые: распределения задач по кластерам, затем распределение по процессорам.

С 2010 года ИСП РАН ведет работы в области облачных вычислений на базе инфраструктуры "Университетский кластер". Была разработана система Unihub, которая представляет собой WEB-лабораторию – систему, позволяющую пользователям работать с HPC-приложениями через WEB-браузер. В 2013 году в основе системы стало применяться ПО для развертывания и управления облачными вычислительными системами OpenStack.

Актуальным представляется исследование модели распределения задач в облачной среде, обобщающей нашу модель геометрической упаковки прямоугольников в несколько полос. Эта тематика весьма популярна, однако признанных моделей и постановок задач еще очень мало. Одна из моделей предложена в [48] и для нее рассмотрена задача составления расписаний.

Литература

- [1] Graham R. L., Bounds for certain multiprocessor anomalies, Bell System Technical Journal, 1966, vol. 45, p. 1563–1581.
- [2] Baker B.S., Coffman E.J., Rivest R.L. Orthogonal packings in two-dimensions, SIAM J. Computing, 1980, V. 9, P. 846-855.

- [3] Bartal Y., Fiat A., Karloff H., Vohra R., New algorithms for an ancient scheduling problem, *J. Comput. Syst. Sci.*, 1995, vol. 51, No 3, p. 359-366.
- [4] Karger D. R., Phillips S. J., Torng E., A better lower bound for on-line scheduling, *Proc. of the 5th Annu. ACM-SIAM Symp. on Discrete Algorithms*, 1994, p. 132-140.
- [5] Albers S., Better bounds for online scheduling, *Proc. of the 29th ACM Symp. on Theory of Computing*, 1997, p. 130-139.
- [6] Bartal Y., Karloff H., Rabani Y., A better algorithm for an ancient scheduling problem, *Inf. Process. Letters*, 1994, vol. 50, p. 113-116.
- [7] Baker B.S., Schwarz J.S., Shelf algorithms for two dimensional packing problems, *SIAM J. Computing*, 1983, V. 12, P. 508-525.
- [8] Csirik J., Woeginger G.J. Shelf Algorithms for On-Line Strip Packing, *Inf. Process. Lett.*, 1997, V. 63, N 4, P. 171-175.
- [9] Van Vliet A. An improved lower bound for on-line bin packing algorithms, *Inf. Process. Lett.*, 1992, V. 43, P. 277-284.
- [10] X. Han, K. Iwama, D. Ye, G. Zhang, Strip Packing vs. Bin Packing, *Algorithmic Aspects in Information and Management (AAIM)*, 2007, LNCS, 2009, V. 4508, P. 358-367.
- [11] Поспелов А.И., Анализ одного алгоритма упаковки прямоугольников, связанного с построением расписаний для кластеров, *Труды ИСП РАН, М.*, 2004, т. 6, с. 7-12.
- [12] S. Zhuk, A. Chernykh, A. Avetisyan, S. Gaissaryan, D. Grushin, N. Kuzjurin, A. Pospelov, A. Shokurov, Comparison of scheduling heuristics for Grid resource broker, *Proc. Fifth Mexican Int. Conf. in Computer Science, ENC'04*, IEEE Computer Society, 2004, p. 388-392.
- [13] Жук С.Н., Анализ некоторых эвристик в задаче упаковки прямоугольников в несколько полос, *Труды ИСП РАН, М.*, 2004, т. 6, с. 7-12.
- [14] С.Н. Жук. Приближенные алгоритмы упаковки прямоугольников в несколько полос, *Дискретная математика*, т.18, 2006, N , С. 91-105.
- [15] M. Caramia, S. Giordani, A. Iovanella, Grid scheduling by on-line rectangle packing, *Networks*, 2004, p. v. 44, N 2, p. 106-119.
- [16] С.Н. Жук. Об-онлайн алгоритмах упаковки прямоугольников в несколько полос, *Дискретная математика*, т.19, 2007, N 4, С. 117-131.
- [17] A. Tcherykh, J. Ramirez, A. Avetisyan, N. Kuzjurin, D. Grushin, S. Zhuk, Two level job-scheduling for a computational grid, *PPAM*, 2005, P. 774-781.
- [18] A. Tcherykh, U. Schwiegelshohn, R. Yahyapour, N. Kuzjurin. On-line Hierarchical Job Scheduling on Grids with Admissible Allocation, *Proc. Euro-Par*, 2008, LNCS, v. .
- [19] A. Tcherykh, U. Schwiegelshohn, R. Yahyapour, N. Kuzjurin. On-line Hierarchical Job Scheduling on Grids with Admissible Allocation, *Journal of Scheduling*, 2010, v. 13, N 5, pp. 545-552.
- [20] D. Ye, X. Han, G. Zhang, Online multiple strip packing, *Theor. Comp. Sci.*, 2011, V. 412, N 3, P. 233-239.
- [21] M. Bougert, P.F. Dutot, K. Jansen, C. Otte, D. Trystam, Approximation Algorithms for Multiple Strip Packing, *Proc. WAOA-2009*, LNCS, V. 5893, 2010, P. 37-48.
- [22] E. G. Coffman, Jr., P. W. Shor. Packings in two dimensions: Asymptotic average-case analysis of algorithms, *Algorithmica*, 1993, V. 9, P. 253-277.
- [23] E. G. Coffman, Jr., D. S. Johnson, P. W. Shor and G. S. Lueker. Probabilistic analysis of packing and related partitioning problems, *Statistical Science*, 1993, V. 8, P. 40-47.
- [24] P. W. Shor, The average-case analysis of some on-line algorithms for bin packing. *Combinatorica*, 1986, v. 6, 179-200.
- [25] P. W. Shor, How to pack better than Best Fit: Tight bounds for average-case on-line bin packing, *Proc. 32nd Annual Symposium on Foundations of Computer Science*, 1991, pp. 752-759.
- [26] Кузюрин Н.Н., Поспелов А.И., Вероятностный анализ шельфовых алгоритмов упаковки прямоугольников в полосу, *Дискретная математика*, 2006, т. 18, N 1, с. 76-90.
- [27] Н.Н. Кузюрин, А.И. Поспелов. Вероятностный анализ одного алгоритма упаковки прямоугольников в полосу. *Труды Института системного программирования РАН*, том 19, 2010 г. С. 157-164.
- [28] Н. Н. Кузюрин, А.И. Поспелов, Вероятностный анализ одного класса алгоритмов упаковки прямоугольников в полосу, *Журнал вычислительной математики и математической физики*, (ISSN 0044-4669), 2011, т. 51, N 10, С. 1931-1936.
- [29] М.А. Трушников. Об одной задаче Коффмана-Шора, связанной с упаковкой прямоугольников в полосу, *Труды Института системного программирования РАН*, том 22, 2012 г. ISSN 2220-6426 (Online), ISSN 2079-8156 (Print), DOI: 10.15514/ISPRAS-2012-22-24, С. 456-462.
- [30] М.А. Трушников. Вероятностный анализ нового алгоритма упаковки прямоугольников в полосу. *Труды Института системного программирования РАН*, том 24, 2013 г. ISSN 2220-6426 (Online), ISSN 2079-8156 (Print), DOI: 10.15514/ISPRAS-2013-24-21, С. 457-468.
- [31] Kuzjurin N.N., Trushnikov M.A., Average case analysis of a new class of on-line algorithms for Multiple Strip Packing, *Proceedings of the 24th ECCO Conference of the European Chapter on Combinatorial Optimization*, University of Amsterdam, May 30-June 1, p. 68-69.
- [32] С.Н. Жук. О построении расписаний выполнения параллельных задач на группах кластеров с различной производительностью, *Труды Института системного программирования РАН*, том 23, 2012 г. ISSN 2220-6426 (Online), ISSN 2079-8156 (Print), DOI: 10.15514/ISPRAS-2012-23-27, С. 447-454.
- [33] P. Berman, M. Charikar, M. Karpinski, On-line load balancing for related machines, *J. of Algorithms*, 2000, v. 35, N 1, p. 108-121.
- [34] Д.А. Грушин, А.И. Поспелов. Система моделирования Grid: реализация и возможности применения. *Труды Института системного программирования РАН*, том 18, 2010 г. С. 243-260.
- [35] D.G. Feitelson, L. Rudolph, Metrics and Benchmarking for Parallel Job Scheduling, *Lecture Notes in Computer Science*, 1998, v. 1459, p. 1+
- [36] warchive, Parallel Workloads Archive, <http://www.cs.huji.ac.il/labs/parallel/workload/>, dim, 2007.10.26
- [37] garchive, The Grid Workloads Archive, <http://gwa.ewi.tudelft.nl/pmwiki/>, dim, 2007.10.26
- [38] sharcnet, The Shared Hierarchical Academic Research Computing Network, www.sharcnet.ca
- [39] D. G. Feitelson, *Workload Modeling for Computer Systems Performance Evaluation Book draft*, 2005

- [40] В. П. Иванников, Д. А. Грушин, Н. Н. Кузюрин, А. И. Поспелов, А. В. Шокуров. Программная система увеличения энергоэффективности вычислительного кластера. Программирование, 2010, № 6, С. 28–40
- [41] Грушин Д.А., Кузюрин Н.Н., Энерго-эффективные вычисления на группе кластеров, Труды Института системного программирования РАН, том 23, 2012. DOI: 10.15514/ISPRAS-2012-23-26.
- [42] A. Grushin and N. N. Kuzyurin, Energy-Efficient Computing for a Group of Clusters, Programming and Computer Software, vol. 40, #6, 2013
- [43] М.А. Трушников. Об одной задаче построения энергосберегающих расписаний. Программирование, 2010, № 6.
- [44] Y. Sverdlik. Microsoft gets wind power for Dublin data center, <http://www.datacenterdynamics.com>. — 2011.
- [45] Ward Van Heddeghem, W. Vereeckena, D. Colle et al. Distributed computing for carbon footprint reduction by exploiting low-footprint energy availability, Future Generation Computer Systems. — 2012. — Vol. 28. — P. 405–414.
- [46] Xu X., Jiaying Wu G. Y., Wang R. Low-power task scheduling algorithm for large-scale cloud data centers, Journal of Systems Engineering and Electronics. – 2013. – T. 24. – №. 5. – С. 870-878.
- [47] D.A. Grushin, N.N. Kuzyurin. Energy-efficient computations on a group of clusters. Proceedings of the Institute for System Programming of RAS, volume 23, 2012, pp. 433-446.
- [48] Y. Azar, N. Ben-Aroya, N.R. Devanur, N. Jain, Cloud Scheduling with Setup Cost, SPAA'13, June 23-25, 2013, Montreal, Canada.

Two-dimensional packing problems and optimization in distributed computing systems

N.N. Kuzyurin, D.A. Grushin, S.A. Fomin
nnkuz@ispras.ru, dgrushin@gmail.com, fomin@ispras.ru
ISP RAS, Moscow, Russia

Abstract. A classical problem of scheduling the set of tasks optimizing load balancing for a set of given processors was considered in 1966 by Graham who proposed algorithm sending each task to a least loaded machine (processor). This was the first example of approximate algorithm with guaranteed constant approximation quality. But all tasks and processors (machines) in Graham's example were simple (only one processor was sufficient for each task). On the other hand Graham's algorithm was an on-line one that is each task was sent to processor without knowing future tasks. With wide use of computational clusters very actual became the problem of scheduling parallel tasks on the set of parallel clusters in the class of on-line algorithms. This problem has very close relation to a two dimensional packing problem, so-called multiple strip packing problem which we consider in this paper. So, in this paper the problem of scheduling parallel tasks on a group of clusters is addressed. We propose new formal model for this process as an optimization of packing rectangles in a set of strips of different widths (Multiple Strip Packing problem). Some modern results concerning this problem and some open problems are presented. Practical aspects of optimization of scheduling parallel tasks process with different criteria of its quality are considered. The description of modeling system developed in the ISP RAS for experimental investigation of scheduling algorithms is presented and its properties are described.

Keywords: parallel tasks, groups of clusters, optimization, scheduling, modeling

References

- [1]. Graham R. L., Bounds for certain multiprocessor anomalies, Bell System Technical Journal, 1966, vol. 45, p. 1563–1581.
- [2]. Baker B.S., Coffman E.J., Rivest R.L. Orthogonal packings in two-dimensions, SIAM J. Computing, 1980, V. 9, P. 846-855.
- [3]. Bartal Y., Fiat A., Karloff H., Vohra R., New algorithms for an ancient scheduling problem, J. Comput. Syst. Sci., 1995, vol. 51, No 3, p. 359-366.
- [4]. Karger D. R., Phillips S. J., Torng E., A better lower bound for on-line scheduling, Proc. of the 5th Annu. ACM-SIAM Symp. on Discrete Algorithms, 1994, p. 132–140.

- [5]. Albers S., Better bounds for online scheduling, Proc. of the 29th ACM Symp. on Theory of Computing, 1997, p. 130-139.
- [6]. Bartal Y., Karloff H., Rabani Y., A better algorithm for an ancient scheduling problem, Inf. Proces. Letters, 1994, vol. 50, p. 113-116.
- [7]. Baker B.S., Schwarz J.S., Shelf algorithms for two dimensional packing problems, SIAM J. Computing, 1983, V. 12, P. 508-525.
- [8]. Csirik J., Woeginger G.J. Shelf Algorithms for On-Line Strip Packing, Inf. Process. Lett., 1997, V. 63, N 4, P. 171-175.
- [9]. Van Vliet A. An improved lower bound for on-line bin packing algorithms, Inf. Process. Lett., 1992, V. 43, P. 277-284.
- [10]. X. Han, K. Iwama, D. Ye, G. Zhang, Strip Packing vs. Bin Packing, Algorithmic Aspects in Information and Management (AAIM), 2007, LNCS, 2009, V. 4508, P. 358-367.
- [11]. Pospelov A.I., Analiz odnogo algoritma upakovki pryamougol'nikov, svyazannogo s postroeniem raspisanij dlya klasterov [Analysis of cluster scheduling packing algorithm], Trudy ISP RAN [The Proceedings of ISP RAS], M., 2004, vol. 6, p. 7-12. (in Russian)
- [12]. S. Zhuk, A. Chernykh, A. Avetisyan, S. Gaissaryan, D. Grushin, N. Kuzjurin, A. Pospelov, A. Shokurov, Comparison of scheduling heuristics for Grid resource broker, Proc. Fifth Mexican Int. Conf. in Computer Science, ENC'04, IEEE Computer Society, 2004, p. 388-392.
- [13]. Zhuk S.N., Analiz nekotorykh ehvristik v zadache upakovki pryamougol'nikov v neskol'ko polos [Analysis of some heuristics in multiple rectangle packing problem], Trudy ISP RAN [The Proceedings of ISP RAS], M., 2004, vol. 6, p. 7-12. (in Russian)
- [14]. S.N. Zhuk. Priblizhennyye algoritmy upakovki pryamougol'nikov v neskol'ko polos [Approximate algorithms of packing rectangles into several stripes], Diskretnaya matematika [Discrete mathematics], vol.18, 2006, p. 91-105 (in Russian).
- [15]. M. Caramia, S. Giordani, A. Iovanella, Grid scheduling by on-line rectangle packing, Networks, 2004, p. v. 44, N 2, p. 106-119.
- [16]. S.N. Zhuk. Ob onlajn algoritmakh upakovki pryamougol'nikov v neskol'ko polos [Online algorithms for packing rectangles into several stripes], Diskretnaya matematika [Discrete mathematics], vol.19, 2007, N 4, p. 117-131 (in Russian).
- [17]. A. Tchernykh, J. Ramirez, A. Avetisyan, N. Kuzjurin, D. Grushin, S. Zhuk, Two level job-scheduling for a computational grid, PPAM, 2005, P. 774-781.
- [18]. A. Tchernykh, U. Schwiegelshohn, R. Yahyapour, N. Kuzjurin. On -line Hierarchical Job Scheduling on Grids with Admissible Allocation, Proc. Euro-Par, 2008, LNCS, v.
- [19]. A. Tchernykh, U. Schwiegelshohn, R. Yahyapour, N. Kuzjurin. On-line Hierarchical Job Scheduling on Grids with Admissible Allocation, Journal of Scheduling, 2010, v. 13, N 5, pp. 545-552.
- [20]. D. Ye, X. Han, G. Zhang, Online multiple strip packing, Theor. Comp. Sci., 2011, V. 412, N 3, P. 233-239.
- [21]. M. Bougert, P.F. Dutot, K. Jansen, C. Otte, D. Trystam, Approximation Algorithms for Multiple Strip Packing, Proc. WAOA-2009, LNCS, V. 5893, 2010, P. 37-48.
- [22]. E. G. Coffman, Jr., P. W. Shor. Packings in two dimensions: Asymptotic average-case analysis of algorithms, Algorithmica, 1993, V. 9, P. 253-277.
- [23]. E. G. Coffman, Jr., D. S. Johnson, P. W. Shor and G. S. Lueker. Probabilistic analysis of packing and related partitioning problems, Statistical Science, 1993, V. 8, P. 40-47.

- [24]. P. W. Shor, The average-case analysis of some on-line algorithms for bin packing, Combinatorica, 1986, v. 6, 179-200.
- [25]. P. W. Shor, How to pack better than Best Fit: Tight bounds for average-case on-line bin packing, Proc. 32nd Annual Symposium on Foundations of Computer Science, 1991, pp. 752-759.
- [26]. Kuzyurin N.N., Pospelov A.I., Veroyatnostnyj analiz shel'fovykh algoritmov upakovki pryamougol'nikov v polosu [Probability analysis of shelf algorithms packing rectangles into several stripes], Diskretnaya matematika [Discrete mathematics], 2006, vol. 18, N 1, p. 76-90. (in Russian)
- [27]. N.N. Kuzyurin, A.I. Pospelov. Veroyatnostnyj analiz odnogo algoritma upakovki pryamougol'nikov v polosu [Probability analysis of multiple rectangle packing problem]. Trudy Instituta sistemnogo programmirovaniya RAN [The Proceedings of ISP RAS], vol 19, 2010 g. p. 157-164. (in Russian)
- [28]. N. N. Kuzyurin, A.I. Pospelov, Veroyatnostnyj analiz odnogo klassa algoritmov upakovki pryamougol'nikov v polosu [Probability analysis of one class of algorithms for packing rectangles into several stripes], Zhurnal vychislitel'noj matematiki i matematicheskoy fiziki [Journal of computational mathematics], (ISSN 0044-4669), 2011, vol. 51, N 10, p. 1931-1936. (in Russian)
- [29]. M.A. Trushnikov. Ob odnoj zadache Koffmana-SHora, svyazannoj s upakovkoj pryamougol'nikov v polosu [About one Koffman-Shore problem of packing rectangle into stripe], Trudy Instituta sistemnogo programmirovaniya RAN [The Proceedings of ISP RAS], vol 22, 2012 g. ISSN 2220-6426 (Online), ISSN 2079-8156 (Print), p. 456-462. DOI: 10.15514/ISPRAS-2012-22-24. (in Russian)
- [30]. M.A. Trushnikov. Veroyatnostnyj analiz novogo algoritma upakovki pryamougol'nikov v polosu [Probability analysis of new algorithm for packing rectangles into stripe]. Trudy Instituta sistemnogo programmirovaniya RAN [The Proceedings of ISP RAS], vol 24, 2013 g. ISSN 2220-6426 (Online), ISSN 2079-8156 (Print), DOI: 10.15514/ISPRAS-2013-24-21, p. 457-468. (in Russian)
- [31]. Kuzjurin N.N., Trushnikov M.A., Average case analysis of a new class of on-line algorithms for Multiple Strip Packing, Proceedings of the 24th ECCO Conference of the European Chapter on Combinatorial Optimization, University of Amsterdam, May 30-June 1, p. 68-69.
- [32]. S.N. Zhuk. O postroenii raspisanij vypolneniya parallel'nykh zadach na gruppakh klasterov s razlichnoj proizvoditel'nost'yu [About building schedule for parallel tasks on a group of clusters with different performance], Trudy Instituta sistemnogo programmirovaniya RAN [The Proceedings of ISP RAS], vol 23, 2012 g. ISSN 2220-6426 (Online), ISSN 2079-8156 (Print), DOI: 10.15514/ISPRAS-2012-23-27, p. 447-454. (in Russian)
- [33]. P. Berman, M. Charikar, M. Karpinski, On-line load balancing for related machines, J. of Algorithms, 2000, v. 35, N 1, p. 108-121.
- [34]. D.A. Grushin, A.I. Pospelov. Sistema modelirovaniya Grid: realizatsiya i vozmozhnosti primeneniya [Grid modelling system: implementation and application]. Trudy Instituta sistemnogo programmirovaniya RAN [The Proceedings of ISP RAS], vol 18, 2010 g. p. 243-260. (in Russian)
- [35]. D.G. Feitelson, L. Rudolph, Metrics and Benchmarking for Parallel Job Scheduling, Lecture Notes in Computer Science, 1998, v. 1459

- [36]. warchive, Parallel Workloads Archive, <http://www.cs.huji.ac.il/labs/parallel/workload/>, dim, 2007.10.26
- [37]. garchive, The Grid Workloads Archive, <http://gwa.ewi.tudelft.nl/pmwiki/>, dim, 2007.10.26
- [38]. sharcnet, The Shared Hierarchical Academic Research Computing Network, www.sharcnet.ca
- [39]. D. G. Feitelson, Workload Modeling for Computer Systems Performance Evaluation Book draft, 2005
- [40]. V. P. Ivannikov, D. A. Grushin, N. N. Kuzyurin, A. I. Pospelov, A. V. SHokurov. Programmnyaya sistema uvelicheniya ehnergoehffektivnosti vychislitel'nogo klastera [Software system for enhancing energy efficiency of a cluster]. Programmirovaniye [Programming and Computer Software], 2010, No 6, p. 28–40
- [41]. Grushin D.A., Kuzjurin N.N., Jenergo-jeffektivnye vychisleniya na gruppe klasterov [Energy-efficient computations on a group of clusters], Trudy ISP RAN [The Proceedings of ISP RAS], tom 23, 2012. DOI: 10.15514/ISPRAS-2012-23-26.
- [42]. A. Grushin and N. N. Kuzyurin, Energy-Efficient Computing for a Group of Clusters, Programming and Computer Software, vol. 40, #6, 2013
- [43]. M. A. Trushnikov. On one problem of construction of energy-saving schedules. Programming and Computer Software, November 2010, Volume 36, Issue 6, pp 337-342.
- [44]. Y. Sverdlik. Microsoft gets wind power for Dublin data center, <http://www.datacenterdynamics.com>. — 2011.
- [45]. Ward Van Heddeghem, W. Vereeckena, D. Colle et al. Distributed computing for carbon footprint reduction by exploiting low-footprint energy availability, Future Generation Computer Systems. — 2012. — Vol. 28. — P. 405–414.
- [46]. Xu X., JiayingWu G. Y., Wang R. Low-power task scheduling algorithm for large-scale cloud data centers, Journal of Systems Engineering and Electronics. – 2013. – T. 24. – №. 5. – C. 870-878.
- [47]. D.A. Grushin, N.N. Kuzyurin. Energy-efficient computations on a group of clusters. The Proceedings of ISP RAS, volume 23, 2012, pp. 433-446.
- [48]. Y. Azar, N. Ben-Aroya, N.R. Devanur, N. Jain, Cloud Scheduling with Setup Cost, SPAA'13, June 23-25, 2013, Montreal, Canada.