

ПРИМЕНЕНИЕ МЕТОДА БАЛАНСИРОВКИ НАГРУЗКИ НА ВЫСОКОПАРАЛЛЕЛЬНЫХ ВЫЧИСЛИТЕЛЬНЫХ КЛАСТЕРНЫХ СИСТЕМАХ

ИСАХОВ А.А.^{1,2}, АБЫЛКАСЫМОВА А.Б.², МАНСУРОВА М.Е.¹

¹Казахский Национальный университет им. аль-Фараби
²АО «Казахстанско-Британский технический университет»

Аннотация: Высокая параллельная эффективность для различных крупномасштабных задач требует, чтобы вычислительная нагрузка была равномерно распределена между всеми вычислительными процессорами. В данном исследовании представлена схема динамического балансирования нагрузки, которая позволяет повысить эффективность сложных связанных симуляций с нетривиальными расширениями в области. Кроме того, результаты эксперимента с сильным масштабированием показывают, что выгода от выравнивания нагрузки пропорциональна степени параллелизма.

Ключевые слова: динамическое выравнивание нагрузки, параллельная эффективность, теплообмен в носовой полости, схема расщепления по физическим параметрам, метод конечных объемов

КЛАСТЕРЛІК ЖОҒАРЫ ПАРАЛЛЕЛЬДІ ЕСЕПТЕУ ЖҮЙЕЛЕРІНДЕ ЖҮКТЕМЕНІ ТЕҢДЕСТІРУ ӘДІСІН ҚОЛДАНУ

Аңдатпа: Әртүрлі масштабты тапсырмалар үшін жоғары параллельді тиімділік есептеу жүктемесін барлық есептеу процессорлары арасында біркелкі бөлуді талап етеді. Бұл зерттеу аймақтағы тривиалды емес ыдырауымен күрделі байланысты модельдеудің тиімділігін арттыруға мүмкіндік беретін динамикалық жүктемені теңдестіру схемасын ұсынады. Сонымен қатар жоғары масштабтау экспериментінің нәтижелері жүктемені теңестірудің пайдасы параллелизм деңгейіне пропорционалды екенін көрсетеді.

Түйінді сөздер: жүктемені динамикалық теңестіру, параллельді тиімділік, мұрын қуысындағы жылу алмасу, физикалық параметрлер бойынша ыдырату схемасы, ақырлы көлем әдісі

APPLICATION OF THE LOAD BALANCING METHOD ON HIGH PARALLEL COMPUTING CLUSTER SYSTEMS

Abstract: High parallel efficiency for various large-scale tasks requires that the computational load is evenly distributed among all computational processors. This study presents a dynamic load balancing scheme that improves the performance of complex coupled simulations with non-trivial domain expansions. In addition, the high scaling experiment results show that the benefit of load balancing is proportional to the degree of parallelism.

Key words: dynamic load balancing, parallel efficiency, air flow in the human respiratory system, alveolar state, heat exchange in the nasal cavity, projection method, finite volume method

Введение

Носовая полость человека выступает в качестве важного компонента дыхательной системы с множеством жизненно важных функций, включая нагревание, фильтрацию, увлажнение воздушного потока и обоняние. Эти функции основаны на явлениях переноса, которые зависят от характера воздушного потока в носовой структуре. Обычная скорость дыхания в носовой полости рассматривается как ламинарный поток, который ускоряется в определенных местах носа и превращается в переходный поток, постепенно превращаясь в турбулентный поток из-за уменьшения площади поперечного сечения [1]. В работе [2] провели измерения с помощью лазерной анемометрии на модели человеческого носа и обнаружили, что поле потока в основном ламинарное, а профили скорости почти параболические во всех поперечных сечениях. С развитием высокопроизводительных вычислений в последние годы многие исследователи рассматривали моделирование с помощью вычислительной гидродинамики (CFD) в качестве альтернативного подхода для изучения воздушного потока в дыхательных системах человека. В работах [3-6] для моделирования воздушного потока в носовой полости используют ламинарную модель при относительно низких скоростях. Так для получения более точного численного результата для моделирования воздушного потока в носовой полости используется высокопроизводительные вычисления. Однако основным недостатком использования решателей является необходимость обмена членами источника через дисковый ввод-вывод. Помимо огромных требований к дисковому пространству для крупномасштабных задач, общая параллельная масштабируемость ограничена доступной пропускной способностью ввода/вывода. Таким образом, в работе был разработан новый связанный метод. Это позволяет одновременно выполнять оба метода с обменом данными в памяти. Кроме того, используя разделение вычислительной области в комбинации с динамической баланси-

ровкой нагрузки (DLB) является возможной. Таким образом данный метод сочетает в себе преимущества подходов прямого и гибридного решения, то есть минимизирует время для ввода-вывода и использует различные длины шкалы. Разделение вычислительной сетки широко используется для распараллеливания и балансировки нагрузки в приложениях для научных вычислений [7]. Данное разделение обеспечивает линейаризацию объектов, расположенных в многомерном пространстве, сохраняя пространственную локальность. Таким образом, определение разложения домена сводится к решению проблемы одномерного разбиения, которая известна как проблема разбиения цепочек на цепочки (ССР). Проблема ССР широко изучалась в литературе, давая точные решения [8], параллельные эвристики [9], масштабируемые иерархические алгоритмы [10], а также расширение для гетерогенных систем [11]. Все эти методы предполагают, что имеется достаточно точная информация о распределении рабочей нагрузки. Это может быть достигнуто, например, с помощью инструментария кода для измерения времени вычислений на объектной основе [12]. Однако в случае, если такой подход неосуществим, требуется альтернатива для оценки весов для различных типов объектов с разными вычислительными затратами. Это предполагает, что каждый объект определенного типа демонстрирует одинаковую постоянную нагрузку. Тем не менее, для сложных приложений это часто приводит к неоптимальной декомпозиции домена со сбалансированной нагрузкой, поскольку локальные изменения рабочей нагрузки не фиксируются. Следовательно, требуется метод DLB для секционирования на основе разбиения вычислительной области, который более непосредственно нацелен на дисбаланс нагрузки. В данной работе представлена схема динамического выравнивания нагрузки для связанных методов, которая позволяет повысить эффективность комплексного моделирования с нетривиальными разложениями доменов.

1. Распараллеливания с помощью динамической балансировки нагрузки

Для моделирования потока воздуха в носовой полости, основные уравнения для сохранения массы, импульса, температуры и относительной влажности. Для численного решения данной системы уравнений используется схема расщепления по физическим параметрам [13]. Уравнения дискретизируются с помощью метода конечных объемов [13]. Конечная цель динамической балансировки нагрузки – максимизировать общую производительность параллельных вычислений путем перераспределения рабочей нагрузки между процессами так, чтобы вся доступная вычислительная мощность использовалась на полную мощность [7]. Балансировка нагрузки в высокопараллельных приложениях является ключом для эффективного использования ресурсов [14], поскольку даже незначительные дисбалансы могут серьезно повлиять на производительность и масштабируемость вычислений. Численное моделирование обычно включает частую связь между подобластями MPI, так что усугубление дисбаланса будет происходить на каждой итерации или стадии временного шага из-за синхронизации [15]. Кроме того, потеря производительности чувствительна к конкретной модели дисбаланса. Один перегруженный процесс значительно снизит общую производительность. Причина этого заключается в том, что все остальные процессы ждут и бездействуют, пока не будет продолжена связь между соседними областями. С другой стороны, некоторые недостаточно загруженные процессы будут иметь незначительное влияние на общую производительность. Наконец, дисбалансы можно классифицировать как динамические или статические [15]. Динамические дисбалансы возникают, если распределение рабочей нагрузки изменяется в зависимости от времени, что происходит, например, при использовании адаптивных сеток решения. Постоянное распределение рабочей нагрузки приведет к статическому дисбалансу.

Дисбаланс нагрузки и результирующее влияние на производительность могут быть количественно определены с помощью различных показателей. Здесь процент дисбаланса [44], который выражает серьезность дисбаланса, и определяется в таком виде

де $I_{\%} = \frac{t_{\max} - t_{\text{avg}}}{t_{\max}} \frac{N}{N-1}$. Величины t_{\max} и t_{avg} представляют максимальное и среднее время, необходимое для обработки данного фрагмента программного кода, а N - количество параллельных процессов. Значение $I_{\%}=0\%$ соответствует идеально сбалансированному распределению нагрузки, а $I_{\%}=100\%$ связано с разделом кода, выполняемым только для одного процесса. Таким образом, данная метрика показывает количество потраченных впустую ресурсов. Это предполагает, что все, кроме самого медленного процесса, простаивают в конце соответствующего раздела кода, что может быть связано с завершением одного временного шага или этапа схемы интегрирования по времени в численном моделировании. Потенциальная экономия времени работы выражается через время дисбаланса $I_t = t_{\max} - t_{\text{avg}}$ в предположении, что достигается идеальный баланс нагрузки [16]. Кроме того, влияние времени распределения оценивается как $I_T = N I_p$, что является верхней границей общего количества потраченных ресурсов [15].

Ключом к разделению и балансировке нагрузки для связанных задач является оценка вычислительных затрат для различных объектов при моделировании [12]. Стандартный подход для сред с единственным методом направлен на равномерное распределение объектов по всем параллельным поддоменам, предполагая однородную вычислительную среду и то, что каждый объект имеет одинаковые постоянные вычислительные затраты. Однако это создает трудности, если существует несколько типов объектов с различными вычислительными затратами. Эта проблема может быть уменьшена путем использования заранее определенных вычислительных весов для каждого типа объекта, который для рассматриваемого случая включает в себя вычислительные ячейки. Однако для слож-

ных приложений это часто приводит к неоптимальной декомпозиции с балансировкой нагрузки из-за неточных вычислительных весов, которые не учитывают вычислительные нагрузки всех задействованных алгоритмических элементов или неоднородного вычислительного оборудования. Инструментальные средства кода для измерения фактического времени вычислений каждого объекта представляют собой возможные контрмеры [17]. Однако этот подход часто оказывается невозможным для научных приложений из-за значительных усилий по внедрению и дополнительных накладных расходов, вызванных различными инструментами. Таким образом, для оценки вычислительных весов требуется минимально навязчивый и надежный метод, который, в свою очередь, может использоваться для балансировки нагрузки.

Алгоритм DLB, предложенный в этом исследовании, оценивает вычислительные веса на основе измерений вычислительного времени в каждом параллельном поддомене. Кроме того, в текущее распределение объектов включены различные методы среди всех поддоменов. Эти типы объектов, которые можно рассматривать как различные типы нагрузки, приводят к изменению состава нагрузки. Во время моделирования общее время вычислений всех методов измеряется локально для каждого временного шага. В каждом поддомене i среднее значение r_i определяется как усеченное среднее значение на 25% для фильтрации внешних программных воздействий [15]. С глобальным средним временем вычислений среди всех N параллельных процессов задаются как

$$\bar{r} = \frac{1}{N} \sum_{i=0}^{N-1} r_i \quad (1)$$

локальная вычислительная нагрузка l_i определяется в таком виде

$$l_i = \frac{r_i}{\bar{r}} \quad (2)$$

В последствии вычислительные веса c для различных типов нагрузки могут быть оценены путем решения задачи $Ac = l$ с помощью метода наименьших квадратов, причем правая часть определяется вектором нагруз-

ки l , а левая матрица A представляет текущее распределение рабочей нагрузки среди всех поддоменов. Это предполагает, что в среднем нагрузка может быть выражена как линейная комбинация отдельных вкладов рабочей нагрузки. Линейная задача наименьших квадратов с единственным решением минимальной нормы может быть записана как

$$\|Ac - l\|_2 = \min_v \|Av - l\|_2 \text{ и } \|v\|_2 \quad (3)$$

В настоящей работе для решения проблемы методом наименьших квадратов используется процедура DGELSD в LAPACK [17], приведенной в формуле (3). Пример для этой процедуры с двумя типами нагрузки, распределенными среди $N=4$ параллельных поддоменов, приведен в формуле (4). Решение переопределенной системы линейных уравнений методом наименьших квадратов дает расчетное весовое соотношение 2.61 между двумя типами нагрузки. Таким образом, среднее время вычислений для второго типа объекта оценивается в 2.6 раза выше, чем у первого.

$$N \times \begin{bmatrix} 10 & 7 \\ 13 & 4 \\ 12 & 2 \\ 5 & 8 \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \end{bmatrix} = \begin{bmatrix} 1.2 \\ 0.9 \\ 0.8 \\ 1.1 \end{bmatrix} \Leftrightarrow Ac = l \xrightarrow{\text{least squares solution}}$$

$$c = \begin{bmatrix} 0.0420 \\ 0.1097 \end{bmatrix} \quad \frac{c_1}{c_0} = 2.61 \quad (4)$$

В общем, вычисление с помощью декомпозиции области на основе моделирования сводится к решению возникающей проблемы ССР. Простое расширение для динамической балансировки нагрузки дается путем оценки вычислительных весов для различных типов нагрузки во время моделирования, чтобы определить новые рабочие нагрузки в качестве входных данных для проблемы разделения с помощью 1D декомпозиции области. Однако, поскольку локальные изменения рабочей нагрузки не фиксируются, разделение может быть неоптимальным с точки зрения производительности, поскольку любой выбранный алгоритм ССР стремится к оптимальному качеству раздела. Поэтому для более строгого устранения дисбаланса нагруз-

ки предлагается другая процедура для определения новых смещений домена для разделения на основе моделирования. Во-первых, начальное разбиение улучшается за счет использования вышеупомянутого способа для получения достаточно хорошей начальной точки для последующих этапов. На следующих этапах перераспределения отдельные смещения области уточняются итеративно на основе измеренного дисбаланса нагрузки и вычисленных весов вычислений. В качестве отправной точки, кумулятивные дисбалансы нагрузки, представленные в виде

$$s_j = \sum_{i=0}^{j-1} l_i - 1, \forall j \in \{1, \dots, N\} \text{ с } s_0 = 0 \quad (5)$$

Каждое значение количественно определяет для соответствующего смещения домена o_j общий дисбаланс нагрузки области слева и справа от позиции разделения. Таким образом, оптимальное локальное положение требует смещения o_j во время проведения моделирования, чтобы минимизировать совокупный дисбаланс. Общее предположение этого подхода состоит в том, что путем оптимизации каждого смещения в отдельности может быть получен глобальный баланс нагрузки. Эта процедура может быть понята как инкрементный, диффузионный метод DLB [17], основанный на разделении CFD, который может лучше подходить для крупномасштабного моделирования [18]. С вычислительной нагрузкой l_i каждого параллельного поддомена, совокупные дисбалансы s_j вычисляются в соответствии с формулой (5). Таким образом, расхождение нагрузки между всеми процессами слева и справа от каждой позиции разделения определяется количественно. Это подразумевает, что, учитывая значение $s_2=0.45$ в примере, вычислительная нагрузка первых двух процессов значительно выше, чем у двух оставшихся. Поэтому, сдвигая смещение влево, рабочая нагрузка перемещается из перегруженных в недостаточно загруженные процессы. Соответственно, глобальные дисбалансы нагрузки могут быть уменьшены путем индивидуального сдвига каждого смещения домена в направлении, заданном

$$w_k = l_i w_k \quad (6)$$

Распределение нагрузки w_k ячейки раздела k в домене i вычисляется как $\tilde{w}_k = l_i \bar{w}_k$

с, $\bar{w}_k = \frac{w_k}{W_i}$, что можно интерпретировать как распределение нагрузки l_i на ячейки локального раздела. Путем уравнивания совокупного дисбаланса s_j с пересекаемыми долями нагрузки на ячейки раздела

$$s_j^k = s_j^{k-1} + d_j \cdot f_{\text{penalty}} \cdot \tilde{w}_m(k) \quad \text{для } k \geq 1 \quad \text{с}$$

$$m(k) = o_j + d_j k - \frac{1}{2}(d_j + 1) \text{ и } s_j^0 = s_j \quad (7)$$

необходимое смещение каждого смещения o_j можно оценить по индексу последовательности k , для которого выполняется $s_j^k \approx 0$. Дополнительный коэффициент штрафования $f_{\text{penalty}} \geq 1$ позволяет ограничить смещения и предотвратить выбросы. Таким образом, алгоритм DLB уточняет разбиение во время моделирования. Этот подход иллюстрируется с помощью уравнения (1) для второго смещения домена o_2 , используя штрафной коэффициент $f_{\text{penalty}} = 1.25$. Согласно формуле (7), начальный совокупный дисбаланс $s_2 = 0.45$ минимизируется путем сдвига смещения на две ячейки разделения, что дает смещение нового домена.

Кроме того, путем введения факторов

$$\begin{aligned} s_2^0 = 0.45 &\rightarrow s_2^1 = 0.45 - 1.25 \cdot 1.2 \cdot 0.1 = 0.3 \\ &\rightarrow s_2^2 = 0.3 - 1.25 \cdot 1.2 \cdot 0.15 = 0.075. \end{aligned}$$

производительности, оценивающих скорость обработки на разных вычислительных узлах, этот подход может быть расширен и нацелен также на вычисления на разнородных вычислительных аппаратных средствах. Относительная производительность обработки может быть определена путем соотнесения локальной нагрузки с обратной величиной относительной рабочей нагрузки домена. Вес мощности для количественной оценки относительных скоростей вычислений [19], нахо-

диться в таком виде: $p_j = l_i \bar{W} / W_i$. Используя соответствующие вычислительные мощности в качестве дополнительного фактора в процедуре итерации, приведенной в формуле (7), большая рабочая нагрузка может быть перенесена из более медленных процессов в более быстрые и наоборот. Например, учитывая высокую и низкую вычислительную мощность $p_{i-1}=0.8$ и $p_i=1.2$, рабочая нагрузка, смещенная с более быстрого на более медленные процессы, взвешивается с коэффициентом $p_i/p_{i-1}=1.5$. В целом, этот подход DLB подходит для приложений со статическим или динамическим дисбалансом нагрузки. В последнем случае, однако, стоимость частого перераспределения должна быть сбалансирована с потенциальным увеличением производительности. В этом исследовании используется последовательная версия алгоритма DLB, которая требует сбора рабочих нагрузок ячейки раздела в одном домене. Чтобы снизить накладные расходы на связь и требования к памяти, возможно расширение иерархической параллельной версии. Тем не менее, при наличии только одного значения на вычислительную ячейку раздела объема связи довольно низок по сравнению, например, с большим количеством информации о нагрузке [20].

2. Результаты численного моделирования и параллельный анализ эффективности

В результате численного моделирования аэродинамики носовой полости человека получены следующие данные. Для проведения анализа параллельного численного алгоритма использовались данные из статьи [15], где описаны профили продольной компоненты скорости и температуры в трех поперечных сечениях. Среднее время потраченное на шаг времени моделирования разделено на вычисления и связь между всеми параллельными поддоменами. При использовании DLB общая вычислительная нагрузка распределяется между параллельными поддоменами более равномерно, в то время как одновременно в вычислении CFD участвует больше рангов. Процент дисбаланса $I_{\%}$ уменьшен с 31% до

9%, что соответствует значительному улучшению производительности при экономии времени выполнения на 22%. Как показывает этот показатель дисбаланса, потенциальная экономия еще выше при условии идеального баланса нагрузки. Однако, что касается сложности полностью связанного моделирования с резко изменяющимися составами нагрузки, и без того достаточно хорошее начальное разбиение значительно улучшается. Автоматический алгоритм DLB работает с минимальной информацией о типе задач. Суммарные вычислительные ресурсы, требуемые для одного шага по времени моделирования с сильным масштабированием от 28 до 80 вычислительных ядер без DLB и с DLB, изображены на рисунке 1. Кроме того, среднее время простоя показано для каждой конфигурации. Для меньшего числа вычислительных ядер параллельные вычисления демонстрируют лишь небольшой дисбаланс. Из-за низкой степени параллелизма каждому домену назначается огромная вычислительная нагрузка, которая сохраняет дисбалансы, вызванные, например, неточными вычислительными весами. Тем не менее, для DLB наблюдаются небольшие улучшения в отношении общей параллельной производительности. Во время масштабирования дисбалансы нагрузки в эталонном моделировании начинаются с процента дисбаланса $I_{\%}=9.6\%$ для 28 ядер и увеличиваются до $I_{\%}=31\%$ для 84 ядер. Одновременно общий объем вычислительных ресурсов, необходимых для выполнения вычислений в моделировании, увеличивается почти на 20% из-за уменьшения размеров локальных задач. С балансировкой нагрузки дисбалансы уменьшаются до примерно $I_{\%}=4.6\%$ и $I_{\%}=22\%$ для моделирования на 28 и 84 вычислительных ядрах. Таким образом, несмотря на то, что все еще есть потенциал для достижения дальнейших улучшений с помощью DLB, выгода увеличивается со степенью параллелизма, экономя 22% вычислительных ресурсов при использовании 84 ядер. Тенденция расчетных весовых соотношений между вычислительными ячейками дает дополнительное представ-

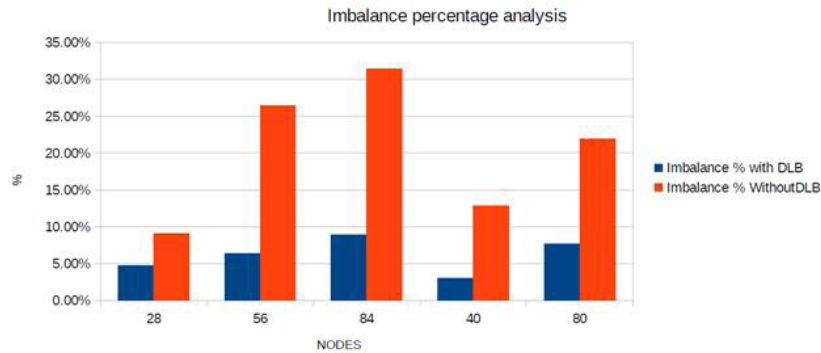


Рис. 5 – Распределение нагрузки для моделирования с использованием различных вычислительных ядер без (слева) и с балансировкой нагрузки (справа)

ление о худших характеристиках эталонного моделирования при использовании большого числа ядер. Для низких степеней параллелизма весовое соотношение вычислений близко к расчетному априорному значению, которое используется для начального разбиения. Однако для большего числа вычислительных ядер это соотношение значительно уменьшается. Аналогично наблюдениям, основанным на времени вычисления на ячейку и временном шаге это поведение может быть объяснено снижением эффективности решателя при моделировании при уменьшении размера задачи в каждом параллельном поддомене. Из-за растущего расхождения в отношении предписанного весового соотношения качество разложения домена постепенно ухудшается и вызывает значительный дисбаланс нагрузки.

Заключение

Моделирование движения воздуха в носовой полости демонстрирует эффективность схемы DLB для различных крупномасштабных связанных задач. Детальный анализ производительности показал необходимость метода DLB для непосредственного определения дисбалансов нагрузки, которые, например, вызваны индивидуальной эффективностью вычисления в зависимости от состава локальной рабочей нагрузки и масштабируе-

мости отдельных программных кодов. Кроме того, эксперимент с сильным масштабированием показал улучшение производительности при растущей степени параллелизма, когда используются априорные оценочные вычислительные веса для начального разбиения. А также в ходе исследования носовой полости можно сделать следующие выводы, что стенки носовой полости способствуют нагреву воздуха и появлению вихрей, которые имеют немаловажное значение для перехода входного воздуха в альвеолярное состояние, перед тем как попасть в носоглотку. А также немаловажную роль играет, относительная влажность в носовой полости, так как при низких температурах окружающей среды за счет влажности идет нагревание входного воздуха. Исследования движения воздуха в носовой полости являются актуально значимыми, так как в настоящее время, по разным причинам увеличивается количество людей с проблемами дыхания через нос. Так данная проблема разрешается хирургическим путем, где важно оперировать структуру носа оптимально, так чтобы носовая полость правильно функционировала, поскольку нормальное дыхание должно осуществляться с помощью носовой полости.

ЛИТЕРАТУРА

1. Doorly, D., Taylor, D., Schroter, R., 2008c. Mechanics of airflow in the human nasal airways. *Respir. Physiol. Neurobiol.* 163, 100–110.

2. Girardin, M., E. Bilgen, and P. Arbour. Experimental study of velocity fields in a human nasal fossa by laser anemometry. *Ann. Otol. Rhinol. Laryngol.* 92:231–236, 1983.
3. Croce, C., Fodil, R., Durand, M., Sbirlea-Apiou, G., Caillibotte, G., Papon, J.-F., Blondeau, J.-R., Coste, A., Isabey, D., Louis, B., 2006. In vitro experiments and numerical simulations of airflow in realistic nasal airway geometry. *Ann. Biomed. Eng.* 34, 997–1007.
4. Keyhani, K., Scherer, P., Mozell, M., 1995. Numerical simulation of airflow in the human nasal cavity. *J. Biomech. Eng.* 117, 429–441.
5. Wang, T., Chen, D., Wang, P., Chen, J., Deng, J., 2016. Investigation on the nasal airflow characteristics of anterior nasal cavity stenosis. *Braz. J. Med. Biol. Res.* 49.
6. Naftali S, Rosenfeld M, Wolf M, Elad D (2005) The air-conditioning capacity of the human nose. *Ann Biomed Eng* 33:545–553.
7. Hendrickson B., Devine K. Dynamic load balancing in computational mechanics, *Comp. Methods in Appl. Mech. Eng.*, vol. 184, no. 2–4, pp. 485–500, Apr. 2000.
8. Pinar A., Aykanat C. Fast optimal load balancing algorithms for 1D partitioning, *J. Parallel Distrib. Comput.*, vol. 64, no. 8, pp. 974–996, Aug. 2004.
9. Pilkington J. R., Baden S. B. Dynamic partitioning of non-uniform structured workloads with spacefilling curves, *IEEE Trans. Parallel Distrib. Syst.*, vol. 7, no. 3, pp. 288–300, Mar. 1996.
10. Lieber M., Nagel W. E. Scalable high-quality 1D partitioning, in *HPCS*, 2014, pp. 112–119.
11. Pinar A., Kartal Tabak E., Aykanat C. One-dimensional partitioning for heterogeneous systems: Theory and practice, *J. Parallel Distrib. Comput.*, vol. 68, pp. 1473–1486, Nov. 2008.
12. Menon H., Jain N., Zheng G., Kale L. Automated load balancing invocation based on application characteristics, in *IEEE Cluster Comput.*, 2012, pp. 373–381.
13. Issakhov A. Mathematical modeling of the discharged heat water effect on the aquatic environment from thermal power plant under various operational capacities // *Applied Mathematical Modelling*, –2016, Volume 40, Issue 2, pp. 1082–1096
14. Ashby S. The Opportunities and Challenges of Exascale Computing. ASCAC subcommittee, US - DOE Report, 2010.
15. Böhme D. Characterizing Load and Communication Imbalance in Parallel Applications, ser. IAS. Forschungszentrum Jülich, 2014, vol. 23.
16. DeRose L., Homer B., Johnson D. Detecting Application Load Imbalance on High End Massively Parallel Systems, *Parallel Processing*, pp. 150–159, Aug. 2007.
17. Anderson E., Bai Z., Bischof C., Blackford S., Demmel J., Dongarra J., Du Croz J., Greenbaum A., Hammarling S., McKenney A., Sorensen D. *LAPACK Users' Guide*, 3rd ed. SIAM, 1999. ISBN 0-89871-447-8
18. Lieber M., Gößner K., Nagel W. E. The potential of diffusive load balancing at large scale, in *EuroMPI*, 2016, pp. 154–157.
19. Zhang X., Yan Y. Modeling and characterizing parallel computing performance on heterogeneous networks of workstations, *Parallel Distrib. Comput.*, pp. 25–34, Oct. 1995.
20. Zheng G., Bhatel   A., Meneses E., Kal   L. V. Periodic hierarchical load balancing for large supercomputers, *Int. J. High Perform. Comput. Appl.*, vol. 25, no. 4, pp. 371–385, Nov. 2011.