

КОМПЬЮТЕРНОЕ МОДЕЛИРОВАНИЕ РЕКОНСТРУИРОВАННОГО СРЕДНЕГО УША С ПОМОЩЬЮ РАСПРЕДЕЛЕННЫХ ВЫЧИСЛЕНИЙ

С.А. Ермоченко, В.В. Новый

Учреждение образования «Витебский государственный университет
имени П.М. Машерова»

В статье рассматривается математическая модель колебательной системы реконструированного среднего уха, которая построена для расчета ее напряженно-деформированного состояния. Проблемой компьютерного расчета является большой объем необходимых вычислений.

Цель работы – построение на основе математической модели компьютерной модели и ее расчет с применением технологии распределенных вычислений.

Материал и методы. *В статье рассматривается математическая модель реконструированного среднего уха, построенная для расчета напряженно-деформированного состояния его колебательной системы. Основными методами исследования являются описательно-аналитический, математического моделирования деформируемых твердых тел, компьютерного моделирования и проектирования распределенных вычислительных систем.*

Результаты и их обсуждение. *Для расчета напряженно-деформированного состояния колебательной системы среднего уха построена упрощенная математическая модель. Затем математическая модель переведена в компьютерную модель с оптимизацией ее расчета на распределенной вычислительной сети. Рассмотрены вопросы проектирования, разработки и тестирования вычислительной системы. Отдельное внимание уделено управлению системой и подсистеме хранения данных.*

Заключение. *Построенная компьютерная модель может быть рассчитана на кластере из обычных маломощных персональных компьютеров, объединенных в локальную сеть. Использование построенной системы дало существенный прирост производительности по сравнению с применением одного персонального компьютера.*

Ключевые слова: *реконструированное среднее ухо, напряженно-деформированное состояние, компьютерная модель, распределенные вычисления, подсистема хранения данных.*

COMPUTER MODELING OF RECONSTRUCTED MIDDLE EAR USING DISTRIBUTED COMPUTING

S.A. Yermochenko, V.V. Novyi

Educational Establishment «Vitebsk State P.M. Masherov University»

The mathematical model of the vibratory system of the reconstructed middle ear, which is built to calculate its strain-stress state, is considered in the article. The problem of computing is a large amount of the required calculations.

The purpose is to build a computer model on the basis of the mathematical model and to calculate it applying the technology of distributed computing.

Material and methods. *The material is a mathematical model of the reconstructed middle ear, which is built to calculate the strain-stress state of its vibratory system. The basic research methods are the descriptive and analytical, the method of mathematical modeling of strain solid, the method of computer modeling and the method of designing distributed computing systems.*

Findings and their discussion. *To calculate the strain-stress state of the vibration system of the middle ear a simplified mathematical model was built. Then the mathematical model was converted into the computer model with the optimization of its calculation on the distributed computer network. Issues of designing, development and testing of the computing system were considered. Special attention was paid to the system management and to data warehouse subsystem.*

Conclusion. *The built computer model can be calculated on the cluster of usual low power PCs joined into a local network. The use of the elaborated system resulted in a considerable performance gain compared to using one PC.*

Key words: *reconstructed middle ear, strain-stress state, computer model, distributed computing, data warehouse subsystem.*

Среднее ухо (СУ) человека представляет собой сложнейшую колебательную систему, передающую энергию звуковой волны, воспринимаемой внешним ухом, во внутреннее ухо, где она преобразуется в нервные импульсы. Компоненты среднего уха могут подвергаться деструктивным изменениям и патологиям, вызванным различными травмами или заболеваниями. В результате этого порог восприимчивости слухового анализатора к звуковым колебаниям может существенно снижаться, что приводит к частичной или полной потере слуха. В ряде случаев с целью улучшения или восстановления слуха в оториноларингологии прибегают к хирургической реконструкции СУ, заключающейся во введении имплантатов и протезов, заменяющих поврежденные естественные компоненты СУ [1].

Замену компонент СУ (барабанной перепонки, слуховых косточек), подвергнувшихся различным патологиям, на искусственные имплантаты важно изучить с точки зрения механических свойств реконструированной колебательной системы (их близости к характеристикам среднего уха в норме) [2; 3].

Основным направлением исследований в большинстве известных работ являлось изучение динамических характеристик реконструированного среднего уха (PCY), позволяющих оценить успешность проведенной операции. Однако в подавляющем числе научных публикаций авторы пренебрегают начальными напряжениями, возникающими в тимпанальной мембране (ТМ) и связках после ввода протеза. Вместе с тем излишние начальные напряжения в ТМ и связке овального окна могут приводить к заметному искажению спектра собственных частот колебательной системы PCY [4], а также быть причиной ограниченной подвижности стремени или даже его люкации (вывиха).

С учетом сказанного являются актуальными исследования напряженно-деформированного состояния (НДС) реконструированного среднего уха в зависимости от различных входных параметров, характеризующих физические свойства, геометрические размеры и положение используемых искусственных имплантатов. В данной работе для расчета напряженно-деформированного состояния PCY строится математическая модель, основанная на уравнениях статического равновесия всех компонент PCY. Из этих уравнений выводится система нелинейных трансцендентных уравнений относительно неизвестных параметров, характеризующих НДС PCY.

Но при этом сложностью данной задачи является достаточно большой объем вычислений, необходимых для подробного анализа всех возможных вариантов значений входных параметров, поскольку запись соотношений между входными и выходными параметрами в виде аналитически заданных функций представляется практически невозможной. Между тем исследование зависимости между этими параметрами – важная практическая задача для специалистов в области оториноларингологии, так как позволяет экспертам подбирать оптимальные значения входных параметров с учетом индивидуальных особенностей среднего уха пациента. Таким образом, для анализа зависимости между входными и выходными параметрами модели необходимо рассчитывать НДС PCY для большого количества вариантов значений входных параметров, а затем уже анализировать их изменения и влияние на выходные параметры. Для ускорения вычисления можно воспользоваться технологиями распределенных вычислений.

Цель работы – построение математической модели PCY, позволяющей рассчитать напряженно-деформированное состояние колебательной системы, а также перевод модели в компьютерную, расчет последней с применением технологии распределенных вычислений.

Для уменьшения времени вычислений было принято решение построить кластер из нескольких персональных компьютеров, объединенных в одну локальную компьютерную сеть и представляющих собой в итоге специализированную распределенную вычислительную систему.

Материал и методы. Нами рассматривается математическая модель реконструированного среднего уха, построенная для расчета напряженно-деформированного состояния его колебательной системы.

Основными методами исследования являются описательно-аналитический, математического моделирования деформируемых твердых тел, компьютерного моделирования и проектирования распределенных вычислительных систем.

Результаты и их обсуждение. Функциональное назначение колебательной системы среднего уха заключается в преобразовании звуковых колебаний, улавливаемых наружным ухом, в механические колебания цепи косточек «молоточек–наковальня–стремя» [5]. Далее эти колебания передаются внутриулитковой жидкости и преобразуются внутренним ухом в нервные импульсы. Роль СУ, таким

образом, состоит в передаче звукового давления и защите внутреннего уха от различных механических повреждений.

Здесь и далее приводится упрощенная математическая модель РСУ из [6].

Для определения различных величин, характеризующих свойства компонент РСУ, вводится глобальная система координат $Oxuz$, связанная с реконструированной ТМ. При реконструкции ТМ в полость СУ вводится искусственный протез.

Обозначим длину стержня протеза l_p , минимальное значение длины стержня протеза l_{min} .

Разница длин $\Delta l = l_p - l_{min}$ обуславливает деформацию реконструированной ТМ и связок овального окна, и, как следствие, возникновение усилия в соединении «протез–стремя» (точка K).

Рассмотрим силы и моменты, действующие на протез и стремя.

F_{TM} , M_{TM} – сила и момент, действующие на основание протеза и обусловленные деформацией реконструированной ТМ. F_{AL} , M_{AL} – сила и момент, действующая на основание стремени со стороны деформированных связок овального окна. F_{CL} – сила, действующая на основание стремени со стороны внутриулитковой жидкости. F_{TR} , M_{TR} – распределенные по всему внешнему контуру силы и моменты, действующие со стороны тимпанального кольца на ТМ. P_s – разница атмосферного давления и давления, возникающего в полости среднего уха в послеоперационный период.

К входным параметрам, описывающим РСУ, относятся $D = Eh^3/[12(1 - \nu^2)]$, $K = Eh/(1 - \nu^2)$ – жесткости пластины, E – модуль упругости Юнга, ν – коэффициент Пуассона, h – толщина пластины.

Внутренний контур представляет собой линию сопряжения пластины и круглого основания протеза радиуса b . Соотношения упругости для пластины, моделирующей ТМ, записываются в цилиндрической системе координат $Cr\phi z$, связанной с центром основания протеза. Так как основание протеза прогибает пластину, совершая перемещения u_p , ϕ_p , w_p и повороты θ_x , θ_z , то точки внутреннего контура получают некоторое приращение.

Внешний контур представляет собой окружность радиуса a , а центры ТМ и основания протеза совпадают. На внешнем контуре предполагается упругая заделка. Коэффициенты k_l и k_t – коэффициенты жесткости распределенных линейных и торсионных пружин, моделирующих тимпанальное кольцо. Данные коэффициенты были вычислены экспериментально в [3]. Там же показано, что эти жесткости не являются постоянными по всему контуру. В данной работе приняты усредненные значения этих величин: $k_l = 0,075 \text{ Н / мм}^2$, $k_t = 6,5 \times 10^{-5} \text{ Н}$.

Условием равновесия протеза и стремени является равенство нулю главных векторов, действующих на них сил и моментов.

На конец ствола протеза со стороны стремени действует сила F_{SP} , под влиянием которой центр основания протеза перемещается в точку G . Координаты этой точки в системе координат $Cr\phi z$ обозначим $G(u_p; \phi_p; w_p)$. Здесь u_p – тангенциальное смещение, w_p – нормальное смещение, ϕ_p – направление смещения центра основания протеза.

Протез совершает повороты вокруг некоторой оси, лежащей в плоскости пластины (ТМ), а также вокруг оси, перпендикулярной плоскости пластины. Угол ψ между осью Ox и осью Cx' выберем таким образом, чтобы ось поворота протеза, лежащая в плоскости основания пластины, совпадала с осью Cx' . Тогда угол поворота протеза вокруг данной оси обозначим θ_x , а неизвестный угол ψ будет определять положение этой оси поворота. Ось поворота протеза, перпендикулярная плоскости пластины, совпадает с осью Cz , а угол поворота протеза вокруг этой оси обозначим θ_z . Определим силу, действующую на конец ствола протеза, в системе координат $Cx'y'z$ как $F_{SP}(f_{SP}^{x'}; f_{SP}^{y'}; f_{SP}^z)$, а момент этой силы обозначим $M_{SP} = r_K \times F_{SP}$, где r_K – радиус-вектор точки K в системе координат $Cx'y'z$:

$$\begin{aligned} r_K (u_p \cos \phi_p + w_p \sin \theta_x \sin \theta_z + u_p \cos(\psi - \phi_p) \cos \theta_z + \\ + u_p \sin(\psi - \phi_p) \cos \theta_x \sin \theta_z; \\ u_p \sin \phi_p - w_p \sin \theta_x \cos \theta_z + u_p \cos(\psi - \phi_p) \sin \theta_z - \\ - u_p \sin(\psi - \phi_p) \cos \theta_x \cos \theta_z; \\ w_p (1 + \cos \theta_x) - u_p \sin(\psi - \phi_p) \sin \theta_x). \end{aligned} \quad (1)$$

Таким образом, уравнения равновесия протеза относительно шести неизвестных параметров $u_p, \varphi_p, w_p, \psi, \theta_x, \theta_z$ запишем в виде:

$$f_{TM}^{x'} + f_{SP}^{x'} = 0; \quad f_{TM}^{y'} + f_{SP}^{y'} = 0; \quad f_{TM}^z + f_{SP}^z = 0; \quad (2)$$

$$m_{TM}^{x'} + f_{SP}^z(u_p \sin \varphi_p - w_p \sin \theta_x \cos \theta_z) - f_{SP}^{y'} w_p (1 + \cos \theta_x) + f_{SP}^z u_p \sin \theta_z \cos(\psi - \varphi_p) + \quad (3)$$

$$+ u_p (f_{SP}^{y'} \sin \theta_x - f_{SP}^z \cos \theta_x \cos \theta_z) \sin(\psi - \varphi_p) = 0;$$

$$m_{TM}^{y'} + f_{SP}^{x'} w_p (1 + \cos \theta_x) - f_{SP}^z (u_p \cos \varphi_p + w_p \sin \theta_x \sin \theta_z) - f_{SP}^z u_p \cos \theta_z \cos(\psi - \varphi_p) - \quad (4)$$

$$- u_p (f_{SP}^{x'} \sin \theta_x + f_{SP}^z \cos \theta_x \sin \theta_z) \sin(\psi - \varphi_p) = 0;$$

$$m_{TM}^z + f_{SP}^{y'} (u_p \cos \varphi_p + w_p \sin \theta_x \sin \theta_z) - f_{SP}^{x'} (u_p \sin \varphi_p - w_p \sin \theta_x \cos \theta_z) + u_p (f_{SP}^{y'} \cos \theta_z - f_{SP}^{x'} \sin \theta_z) \cos(\psi - \varphi_p) + \quad (5)$$

$$+ u_p (f_{SP}^{y'} \cos \theta_x \sin \theta_z + f_{SP}^{x'} \cos \theta_x \cos \theta_z) \sin(\psi - \varphi_p) = 0.$$

Рассмотрим здесь условие равновесия стремени с учетом всех сил, действующих со стороны протеза и связки овального окна.

Для определения силы \mathbf{F}_{AL} и момента \mathbf{M}_{AL} , действующих на основание стремени со стороны связок овального окна, рассмотрим модель, предложенную в работе [5], в соответствии с которой данная связка рассматривается как упругая лента переменной ширины.

Жесткостные свойства связки при этом описываются симметричной матрицей 6×6 :

$$C_{RB} = c_{ref} \begin{pmatrix} 51,4 & -0,24 & -1,37 & 0,04 & 9,66 & 0,35 \\ -0,24 & 49,2 & -0,6 & -7,87 & -1,01 & -8,4 \\ -1,37 & -0,6 & 27,8 & 0,37 & 17,1 & 0,96 \\ 0,04 & -7,87 & 0,37 & 8,29 & 0,58 & 2,6 \\ 9,66 & -1,01 & 17,1 & 0,58 & 29,7 & 1,6 \\ 0,35 & -8,4 & 0,96 & 2,6 & 1,6 & 12,9 \end{pmatrix}, \quad (6)$$

где коэффициент $0,035 \leq c_{ref} \leq 0,05$ определяет общую жесткость связки овального окна и имеет размерность $[Н \text{ мм}^{-1}]$.

Тогда сила \mathbf{F}_{AL} и момент \mathbf{M}_{AL} выражаются следующим образом [5]:

$$\left(f_{AL}^{\xi}; f_{AL}^{\eta}; f_{AL}^{\zeta}; m_{AL}^{\xi}; m_{AL}^{\eta}; m_{AL}^{\zeta} \right)^T = C_{RB} \left(\xi_r; \eta_r; \zeta_r; \alpha_{\xi}; \alpha_{\eta}; \alpha_{\zeta} \right)^T, \quad (7)$$

где символ T обозначает транспонирование строки.

Теперь аналогично уравнениям равновесия протеза запишем уравнения равновесия стремени:

$$\mathbf{F}_{AL} + \mathbf{F}_{CL} + \mathbf{F}_{PS} = 0, \quad (8)$$

$$\mathbf{M}_{AL} + \mathbf{r}_K \times \mathbf{F}_{PS} = 0, \quad (9)$$

где $\mathbf{r}_K(\xi_k; \eta_k; \zeta_k)$ – радиус-вектор точки K в локальной системе координат $S\xi\eta\zeta$.

Уравнения (2–5, 8, 9) задают двенадцать уравнений относительно пятнадцати неизвестных. Еще одно уравнение можно выписать из условия недеформируемости стремени. Так, расстояния между центром основания стремени и головкой стремени до введения протеза (точки S и H) и после (точки R и K) принимаются одинаковыми. Отсюда

$$\sqrt{(\xi_s - \xi_h)^2 + (\eta_s - \eta_h)^2 + (\zeta_s - \zeta_h)^2} = \sqrt{(\xi_r - \xi_k)^2 + (\eta_r - \eta_k)^2 + (\zeta_r - \zeta_k)^2}. \quad (10)$$

Еще два уравнения связывают координаты точек до введения протеза (S, H) и после (R, K) с углами поворота $\alpha_{\xi}, \alpha_{\eta}$ стремени вокруг соответствующих осей координат $S\xi$ и $S\eta$:

$$\sin \alpha_{\xi} = \frac{(\eta_s - \eta_h)(\zeta_r - \zeta_k) - (\eta_r - \eta_k)(\zeta_s - \zeta_h)}{(\eta_s - \eta_h)^2 + (\zeta_s - \zeta_h)^2}, \quad (11)$$

$$\sin \alpha_{\eta} = \frac{(\zeta_s - \zeta_h)(\xi_r - \xi_k) - (\zeta_r - \zeta_k)(\xi_s - \xi_h)}{(\zeta_s - \zeta_h)^2 + (\xi_s - \xi_h)^2}. \quad (12)$$

Уравнения (2–5, 8–12) являются алгебраическими уравнениями относительно 15 неизвестных параметров, характеризующих напряженно-деформированное состояние реконструированного среднего уха. Данная система представляет математическую модель реконструированного среднего уха, позволяющую рассчитать его напряженно-деформированное состояние.

К известным входным параметрам данной модели необходимо отнести следующие характеристики:

- геометрические параметры протеза b , l_p и ψ_p ;
- место установки протеза x_p , y_p ;
- геометрические параметры хрящевого имплантата (a – радиус пластины, h – толщина пластины);
- физические параметры хрящевой ткани (ν – коэффициент Пуассона, E – модуль упругости Юнга, k_l и k_t – коэффициенты жесткости линейных и торсионных пружин, распределенных по внешнему контуру хрящевого имплантата и моделирующих тимпанальное кольцо, фиксирующее имплантат);
- геометрические параметры стремени (координаты центра основания стремени и головки стремени, определяющие его размеры и ориентацию в пространстве);
- физические параметры связки овального окна (матрица жесткости связки C_{RB});
- геометрические и физические параметры мембраны круглого окна.

При расчете построенной модели важным является характер зависимости выходных параметров модели от изменения входных параметров, заданных на определенном промежутке. При этом численные расчеты могут осуществляться при изменении таких основных параметров, как геометрические размеры протеза, хрящевого имплантата (в первую очередь толщины) и физических характеристик хряща (в первую очередь модуль упругости Юнга).

И хотя изменяемыми параметрами являются лишь некоторые из входных величин, но их изменение может изучаться с достаточно мелким шагом. Для каждой из этих величин могут потребоваться порядка 10^2 – 10^3 значений, что в целом может давать до 10^8 вариантов конфигурации математической модели. При этом расчет математической модели для одного набора входных данных на персональном компьютере с маломощной конфигурацией (двухъядерный процессор AMD E1-1200 и 2 Gb оперативной памяти) с помощью математического пакета Maple занимает порядка 0,05 секунды. Таким образом, для расчета математической модели для всех возможных конфигураций входных параметров на данном компьютере может потребоваться около 50 суток непрерывной работы.

Построенная модель имеет несколько особенностей:

- во-первых, это нелинейность уравнений, входящих в систему уравнений, что делает затруднительным распараллеливание расчетов модели для одного набора входных данных на несколько вычислительных платформ;
- во-вторых, большое количество различных вариантов входных данных этой модели;
- в-третьих, относительно несложный расчет модели для одного набора входных данных.

Последние две особенности позволяют организовать вычисления на нескольких компьютерах, выполняя расчет одного набора данных на одном компьютере в кластере, а распределение вычислений производить за счет распределения самих таких наборов входных данных между узлами кластера. При этом несложность расчета модели для одного набора данных помогает повысить эффективность работы одного узла за счет передачи ему не единичного набора входных данных, а блока таких входных данных.

Для организации взаимодействия управляющего кластером программного обеспечения и вычислительных модулей, работающих на узлах кластера, можно использовать событийную модель управления, что позволит не только динамически добавлять и удалять узлы в кластере, но и распределять задания между узлами, исходя из вычислительных возможностей каждого узла.

Заметим, что в системе уравнений (2–5, 8–12) можно выразить силу F_k , возникающую в шарнирном соединении, и момент этой силы $r_k \times F_k$. Это позволит разбить решение системы уравнений на два этапа. Однако эти этапы не допускают независимого решения, что не позволит вычислять их параллельно на разных узлах кластера. Но такое разбиение на этапы сможет уменьшить количество

элементарных операций сложения и умножения вещественных чисел, что в целом даст некоторый прирост производительности каждого узла кластера.

Заметим также, что вычисление силы F_{AL} и момента M_{AL} , действующих на основании стремени со стороны связки овального окна, согласно выражениям (6, 7), сопряжено с перемножением матрицы на вектор. Малая размерность этой матрицы делает нецелесообразным распараллеливание данной операции на несколько узлов в компьютерной сети. Но может оказаться эффективным распараллеливание таких вычислений на многоядерных процессорах, что увеличит производительность узлов кластеров, имеющих подобные процессоры. Следует также учесть, что операция перемножения матрицы на вектор будет выполняться на каждой итерации численного метода решения системы нелинейных алгебраических уравнений.

Кроме использования операции перемножения матриц для вычисления компонентов векторов F_{AL} и M_{AL} , такая операция будет применяться и для вычисления компонентов векторов F_{TM} и M_{TM} . И хоть размерность матриц-операндов и смысл их элементов будут различными, но операции перемножения матриц будут одинаковыми. Это позволяет вынести в отдельный поток исполнение таких операций.

Реализацию компьютерной модели для расчета математической модели будем осуществлять с использованием языка программирования C. Применение этого языка позволяет напрямую работать с памятью, используя указатели. В будущем можно продолжить более тонкую оптимизацию программного кода, переписывая критичные блоки кода с применением ассемблерных вставок. Кроме того, язык C является высокоуровневым с достаточно удобным синтаксисом, кроссплатформенным на уровне исходного кода, что позволит компилировать вычислительный модуль под различные платформы.

В настоящей работе модуль разрабатывался только под операционные системы семейства Windows, однако все части кода, имеющие привязку к операционной системе, были локализованы. Так, например, язык C не имеет встроенной высокоуровневой библиотеки для работы с TCP-протоколом, а использование низкоуровневой библиотеки winsock под Windows стало бы достаточно трудоемкой задачей. Одновременно при портировании приложения на другие платформы мог бы встать вопрос о переписывании данной части программного кода. Поэтому вычислительный модуль, несущий основную вычислительную нагрузку, был вынесен в отдельную системную библиотеку dll, которая подключается к Java-приложению, а уже Java-приложение осуществляет взаимодействие по сети с управляющими узлами и организует запуск функций из dll-библиотеки в отдельных потоках.

Такой подход позволяет, не теряя эффективности вычислений, локализовать используемые особенности операционной системы только в части компиляции системной библиотеки, которая далее применяется в приложении, являющемся кроссплатформенным на уровне скомпилированного кода (Java-приложение).

Ранее было показано, что целесообразно решение системы уравнений (2–5, 8–12) для одного набора входных параметров находить на одном вычислительном узле. Распределенные вычисления будут основываться на так называемом параллелизме данных, то есть каждому вычислительному узлу нужно предоставить один набор данных из множества еще не обработанных наборов данных. Подобное множество наборов формируется исходя из тех входных параметров, значение которых может варьироваться на некотором отрезке с заданным шагом.

Для задания одного входного набора данных необходимо передать вычислительному узлу некий набор вещественных чисел и по окончании вычисления получить от него результирующий набор вещественных чисел. Так как требуемая точность должна быть достаточно высокой, каждое число будет представляться вещественным числом с двойной точностью (стандарт IEEE 754), для хранения которого понадобится 8 байт.

Тогда для вычисления системы уравнений (2–5, 8–12) для одного набора данных необходимо передать около 1 Кбайт и получить около 150 байт информации.

Но такой объем данных минимален, если протокол прикладного уровня взаимодействия управляющего сервера и вычислительных узлов в компьютерной сети будет являться бинарным. Подобные протоколы, как правило, имеют небольшой объем служебной информации по сравнению с полезной информацией. Также бинарный протокол выигрывает у текстовых протоколов в скорости извлечения полезных данных. Негативной стороной бинарных протоколов является затруднительный анализ данных для человека. Чаще всего такой анализ необходим при отладке и тестировании приложения, что

является важным этапом в процессе разработки, особенно распределенных систем. Еще один недостаток бинарных протоколов – сложность обеспечения кроссплатформенности. Последний фактор также является достаточно важным в организации распределенных вычислительных сетей, в которых вычислительные узлы различны по аппаратному и программному обеспечению.

Альтернативой бинарным протоколам выступают текстовые протоколы. В данной работе мы рассмотрим текстовый протокол на основе формата XML. К слабым сторонам текстовых протоколов относится чрезмерная избыточность информации. Рассмотрим запрос от управляющего сервера вычислительному узлу:

```
<?xml version="1.0" encoding="utf-8">
<request>
  <param name="prosthesis-base-radius"
    value="1.45"/>
  <param name="prosthesis-length"
    value="5.12"/>
  <!-- и другие параметры -->
</request>
```

В данном примере выписаны только два параметра, но даже на примере этих параметров видно, что количество символов для каждого параметра может отличаться. В среднем на один параметр необходимо около 60 символов. При передаче запроса в формате XML в однобайтовой кодировке это соответствует 60 байтам на один параметр. На весь запрос, содержащий 50 параметров, понадобится около 3000 байт, что в 3 раза превышает размер запроса в байтовом формате.

Тем не менее даже при передаче данного запроса в локальной сети с достаточно низкой скоростью (10 Мбит/с) время передачи будет меньше 0,01 секунды.

Следует также учесть тот факт, что время, необходимое на вычисление решения системы для одного входного набора данных, даже на персональных компьютерах с низким быстродействием (компьютер на основе процессора Intel Pentium 4 с одним ядром и тактовой частотой 1,9 ГГц и оперативной памятью объемом 1 Гб) не превышает 4–5 секунд. По сравнению со временем решения системы скорость передачи входных параметров и результатов даже в текстовом формате с большой избыточностью информации не оказывает существенного влияния на скорость вычислений.

Таким образом, при решении системы уравнений (2–5, 8–12) объем передаваемых по сети данных является сравнительно небольшим, даже при значительной избыточности информации при использовании текстового протокола передачи, основанного на формате XML. Время передачи данного объема информации значительно меньше времени, необходимого для решения этой системы на маломощном персональном компьютере.

Но при увеличении быстродействия вычислительного узла время расчета задачи уменьшается. В подобном случае время передачи данных по сети может оказывать решающее влияние на скорость вычислений в целом. Кроме того, управляющий сервер, получающий ответ от вычислительного узла, должен будет сохранить результаты и найти из множества наборов входных данных еще не обработанный набор и отправить его вычислительному узлу. Все это потребует некоторого времени работы сервера, в течение которого вычислительный узел будет простаивать. И чем большей будет производительность вычислительного узла, тем большим будет процент времени простоя от общего времени работы узла.

Если рассмотреть вычислительную сеть, состоящую из вычислительных узлов с различным быстродействием (при этом различие будет достаточно значительным), то в данном случае рациональнее будет передавать узлам наборы данных не по одному, а пакетами. При этом в зависимости от быстродействия вычислительного узла управляющий сервер может варьировать размер пакета данных таким образом, чтобы поддерживать в оптимальном соотношении время передачи данных по сети и время полезной работы вычислительных узлов.

Для передачи вычислительному узлу пакета входных данных неэффективно будет передавать каждый набор по отдельности, пусть и объединенный в единый запрос. В таком случае экономии объема передаваемых данных не будет. Удастся лишь сократить количество передач, что при использовании постоянных подключений между управляющим сервером и вычислительным узлом не даст никакого прироста производительности.

Если учесть, что большое количество входных наборов формируется из-за варьирования большого количества входных параметров на различных отрезках значений с определенным шагом изменения, то по такому же принципу можно формировать и пакет данных для одного узла. Рассмотрим пример для XML-запроса, рассмотренного выше, но при варьировании второго параметра на отрезке [4,95; 5,25] с шагом 0,025:

```
<?xml version="1.0" encoding="utf-8">
<request>
  <param name="prosthesis-base-radius"
    type="fixed"
    value="1.45"/>
  <param name="prosthesis-length"
    type="interval"
    start="4.95"
    end="5.25"
    step="0.025"/>
  <!-- и другие параметры -->
</request>
```

При таком подходе размер запроса практически остается неизменным (то есть меняется незначительно, в зависимости от соотношения количества варьируемых параметров к фиксированным и в зависимости от нужного количества знаков после десятичной запятой при указании значений параметров). Но при слабой изменчивости размера запроса количество наборов данных может быть произвольным. Фактически в таком формате можно одним запросом описать все интересующие нас наборы параметров, что позволит не только передавать запросы вычислительным узлам, но и конфигурировать через XML-файлы настройки управляющего сервера для организации всего процесса вычисления в целом.

Кроме функциональных особенностей вычислительных и логических узлов, необходимо рассмотреть также способ взаимодействия всех этих узлов в единой вычислительной системе. Поскольку в вычислительной системе будет присутствовать всего два типа узлов (вычислительные и управляющие), то между ними возможны три типа связей:

- вычислительный узел с вычислительным узлом,
- управляющий узел с вычислительным узлом,
- управляющий узел с управляющим узлом.

Так как вычислительные задачи каждого узла независимы друг от друга, то связи между вычислительными узлами в проектируемой вычислительной системе не имеют никакого смысла.

Связь вычислительного узла и управляющего узла была подробно рассмотрена выше при рассмотрении функциональных особенностей этих узлов.

Связи между управляющими узлами в этом случае представляют особый интерес. Прежде всего, встает вопрос: какое количество управляющих узлов должно быть в проектируемой вычислительной системе? При наличии всего одного управляющего узла третий тип связей становится невозможным. Кроме того, упрощается задача связи системы с внешними подсистемами.

Для ответа на поставленный вопрос нужно рассмотреть специфику решаемой задачи с точки зрения требований к распределенным вычислительным системам.

Для распределенной системы важным является возможность гибко добавлять и исключать узлы. При этом описанный выше способ подключения к управляющему узлу новых вычислительных узлов и их отключение обеспечивают гибкое масштабирование вычислительной сети. Но если использовать вычислительную систему с несколькими управляющими узлами, то операции добавления и исключения таких узлов уже не будут являться такими тривиальными.

Тем не менее можно предложить решение вопроса коммуникации управляющих узлов между собой. Управляющие узлы можно организовать в иерархическую структуру так, что каждый управляющий узел может одновременно принимать задания, состоящие из некоторого набора входных данных (фактически являясь при этом вычислительным узлом для другого управляющего узла), далее дробить

данное задание на более мелкие задания и распределять их между своими дочерними вычислительными узлами.

Следовательно, если интерфейс взаимодействия внешней системы (которая может, например, предоставлять графический интерфейс конечному пользователю) и управляющего узла будет таким же, что и управляющего узла с вычислительными узлами, то будет достигнута достаточная гибкость при масштабировании не только при изменении количества вычислительных узлов, но и при изменении количества управляющих узлов.

Иерархический способ организации управляющих узлов предоставляет простой интерфейс взаимодействия с распределенной вычислительной системой, а также позволяет гибко настраивать конфигурацию вычислительной сети.

Рассмотрим теперь структуру программного обеспечения для вычислительных узлов. Каждый вычислительный узел должен решать систему нелинейных алгебраических уравнений. Для этого ему необходимо передать несколько входных параметров, а результатом решения будут являться несколько выходных параметров. Для решения нелинейной системы алгебраических уравнений было решено реализовать метод Ньютона [7]. При реализации выбранного метода решения важно учесть ряд требований:

1) поскольку процесс решения системы численным методом требует значительных вычислительных ресурсов, то программа, реализующая метод Ньютона решения систем нелинейных алгебраических уравнений, должна компилироваться в машинный код процессора, на базе которого будет функционировать вычислительный узел;

2) поскольку аппаратное и программное обеспечение вычислительных узлов может быть разнообразным по своему составу, то программа, реализующая метод Ньютона решения систем нелинейных алгебраических уравнений, должна компилироваться под различные программно-аппаратные платформы (быть кроссплатформенной на уровне исходного кода).

Исходя из предъявленных требований в качестве языка программирования для реализации метода Ньютона решения систем нелинейных алгебраических уравнений выбран язык С. Данный язык является широко распространенным инструментом написания программного обеспечения. Он поддерживает возможность оптимизации программного кода при компиляции, увеличивая эффективность программы. При этом у программиста остается возможность самостоятельно управлять памятью, применяемой программой, а также использовать возможности операционной системы для распараллеливания вычислений.

Тестирование вычислительных узлов показало также большое накопление погрешности вычислений.

При анализе аналитических соотношений математической модели были выделены блоки исходного кода вычислительного узла, в которых происходит накопление погрешности. Так, вычисление координат радиус-вектора r_k шарнирного соединения стержня протеза и головки стременной косточки осуществляется по формулам (1). Здесь величины, входящие в формулу, представляют собой координаты в декартовых или полярных координатах различных точек (центры основания протеза и стремени). При этом для удобства моделирования каждого компонента реконструированного среднего уха (упругая пластина, искусственный протез, стремянная косточка, упругие связки овального окна) вводились локальные системы координат, связанные с центрами пластины и основаниями протеза и стремени. Для анализа формулы с целью оптимизации вычислений и минимизации накопления вычислительной погрешности смысл каждой переменной, входящей в формулу, не так важен.

Программное обеспечение вычислительного узла выполняет расчет данной формулы непосредственно в той последовательности, в какой действия записаны в математической форме записи. При этом, как видно из формул, после вычисления значения тригонометрических функций полученные значения перемножаются между собой, что и приводит к накоплению погрешности. Для минимизации погрешности в формуле (1) после раскрытия скобок произведения тригонометрических функций заменены на соответствующие суммы.

Нагрузочное тестирование вычислительной системы показало ряд недостатков разработанного программного обеспечения:

— программное обеспечение позволяет гибко расширять вычислительную систему, однако при удалении вычислительного или управляющего узла в момент, когда он осуществляет некоторые

вычисления, приводит к тому, что вышестоящий управляющий узел некорректно обрабатывает такую ситуацию и необработанные входные данные на обработку другим узлам подаются с задержкой, что приводит в некоторых ситуациях к простоям узлов;

— при проведении тестирования оказалось неудобным добавлять узлы только перед началом работы системы. И хотя все блоки входных данных передаются вычислительным узлам не сразу, а по мере выполнения расчетов каждым узлом, что позволяет балансировать нагрузку между узлами, добавить новый вычислительный узел после начала процесса вычисления уже нельзя;

— при большом количестве вычислительных узлов, в случае, когда использовалась простая двухуровневая организация вычислительной системы (только один управляющий узел и восемь вычислительных, непосредственно ему подчиняющихся), в которой каждый вычислительный узел выполнял расчет переданного ему блока входных данных в несколько потоков (по 15 потоков на каждом узле), общее количество подключений к централизованной базе данных составило 120 одновременных подключений, что привело к замедлению работы сервера баз данных, так как система управления базами данных MySQL также была запущена на обычном персональном компьютере. Таким образом, снизилась эффективность использования единого централизованного хранилища результатов вычислений при обработке большого входного набора данных большим количеством узлов.

Для решения возникшей проблемы естественным решением кажется применение распределенной подсистемы хранения, строящейся на базе нескольких серверов баз данных, объединенных в общий кластер, с настроенной репликацией данных между узлами кластера. Но такое решение требует времени на репликацию. Правда, необходимости в постоянном обмене данными между узлами кластера подсистемы хранения на самом деле нет. Можно развернуть несколько серверов баз данных, которые будут играть роль кэширующих серверов, каждый из которых будет писать данные в центральный сервер (это можно реализовать триггером в кэширующем сервере баз данных).

Основным недостатком центрального сервера баз данных является его медленная работа при обслуживании большого количества вычислительных узлов. Но при этом нельзя забывать, что визуализация результатов в виде различных графиков должна происходить на базе всех обработанных данных. Следовательно, децентрализация базы данных за счет использования так называемого механизма горизонтального шардинга [8] приносит ускорение при записи результатов расчетов математической модели вычислительными узлами в базу данных, но существенно осложняет работу с этими данными в последующем.

Как было объяснено выше, мы не ставим своей целью распределять данные на несколько серверов из-за ограниченности объема жестких дисков для хранения базы данных. В таком случае можно сделать вывод, что централизованная база данных в конечном итоге будет удобным вариантом для чтения данных приложением, визуализирующим результаты вычислений. Но для более гибкого масштабирования распределенной вычислительной системы необходимо обеспечить более быструю запись в эту централизованную базу данных. Сделать это можно, используя несколько кэширующих серверов, собирающих данные от вычислительных узлов. При этом количество кэширующих серверов может варьироваться в зависимости от количества вычислительных узлов. Такая схема представляет собой идею, обратную идее репликации данных по методу Master-Slave [8]. При репликации мы записываем данные на один Master-сервер, дублируем их на несколько Slave-серверов и читаем информацию с этих Slave-серверов. Но подобный подход эффективен для систем, в которых чтение данных занимает подавляющее время работы системы. В нашем случае как раз Slave-серверы будут сохранять информацию от вычислительных узлов, отдельным фоновым потоком копируя ее на Master-сервер, а приложение уже будет читать данные с этого централизованного Master-сервера. Но проблема в том, что такой механизм не реализуем стандартными возможностями MySQL-сервера реляционных баз данных.

Для реализации этого подхода необходимо разработать подобное решение неким альтернативным способом. При необходимости использования кэширующих серверов баз данных зачастую принято применять некие решения на основе NoSQL баз данных. Но такие решения отличаются, как правило, требованием быстрого поиска информации на кэширующем сервере, пока этот сервер сохраняет в фоновом режиме информацию в едином хранилище. Обозначенный подход ориентирован, в основном, на распределение нагрузки между сервером, обеспечивающим надежное и достаточно

объемное хранилище, и серверами, обеспечивающими быстрый доступ (и чтение, и запись) к информации. Для нашего случая такое решение будет не совсем оправданным, так как затраты на разворачивание и настройку отдельного кластера NoSQL-серверов без необходимости использования их основных преимуществ (хранения индексов, обеспечивающих быстрый поиск) будут превышать выигрыш от оптимизации подсистемы хранения (так как вычислительные узлы не читают результаты вычислений из базы данных, а только пишут их в базу).

В данной работе в качестве альтернативы применена своя собственная простая реализация сервера на языке программирования Java. Серверное приложение для хранения данных может использовать очередь с блокировками для добавления информации от каждого вычислительного узла в конец очереди. Отдельный же низкоприоритетный поток из начала очереди может извлекать данные и сохранять их в централизованную реляционную базу данных под управлением MySQL-сервера.

Использование кэширующих серверов позволило уменьшить количество обращений к серверу баз данных. И даже если количество обращений все равно будет достаточно большим (при очень большом количестве вычислительных узлов), это не скажется на работе вычислительного узла, так как он по-прежнему сможет быстро сохранять данные на кэширующий сервер, после чего скорость их копирования на основной сервер баз данных будет уже не так критична. Возникает лишь проблема с отказоустойчивостью такой вычислительной системы. Если один из кэширующих серверов отказывается, не успев сохранить данные в централизованное хранилище, то эти данные придется пересчитывать снова.

Количество кэширующих серверов можно гибко подбирать в зависимости от количества и производительности вычислительных узлов. Влияние количества вычислительных узлов на загрузку подсистемы хранения мы рассмотрели, но важным является и влияние производительности вычислительных узлов. Так как чем производительнее узел, тем чаще он будет производить сохранение результатов в базу данных.

Теоретически, если масштабировать распределенную вычислительную систему на все большее и большее количество узлов, то даже использование нескольких кэширующих серверов может не обеспечивать необходимую производительность и стать узким местом такой системы.

Для решения подобной проблемы можно построить иерархию кэширующих серверов. Однако для организации сохранения кэширующим сервером данных на другой кэширующий сервер последний должен иметь тот же интерфейс, что и сервер MySQL. Это можно реализовать в общем случае, реализовав некий стандартный интерфейс доступа к данным, такой как ODBC, JDBC, ADO или другой. Но, во-первых, обозначенный подход будет чрезмерно трудоемким; а во-вторых, необходимо помнить, что, в конечном счете, все данные будут сохраняться в единую базу данных под управлением сервера MySQL. В последнем случае узким местом системы, вероятнее всего, станет именно сервер MySQL, а не подсистема кэширующих серверов.

В данной работе мы остановили свой выбор на кэширующих серверах, использующих протокол взаимодействия с вычислительным узлом, основанный на формате XML и сохраняющий информацию в базу данных с помощью SQL-запросов.

Следовательно, мы рассмотрели все аспекты проектирования, разработки и тестирования вычислительной системы и решили задачу расчета построенной математической модели реконструированного среднего уха на вычислительном кластере.

Заключение. В результате проведенного исследования была уточнена и оптимизирована для компьютерного расчета математическая модель реконструированного среднего уха, предназначенная для анализа напряженно-деформированного состояния колебательной системы реконструированного среднего уха. На базе данной математической модели спроектировано, разработано и протестировано программное обеспечение, позволяющее организовать распределенную вычислительную систему для численных расчетов построенной модели.

Вычислительная система строится на базе кластера из обычных персональных компьютеров, объединенных в локальную компьютерную сеть. Такое решение позволяет гибко масштабировать вычислительную систему и экономить время на расчеты математической модели без значительных материальных затрат на аппаратное обеспечение.

Программное обеспечение строилось на базе ранее предложенной авторами концепции применения объектно-ориентированной парадигмы программирования [9].

ЛИТЕРАТУРА

1. Ситников, В.П. Оссикулопластика: варианты и результаты / В.П. Ситников, И.И. Чернушевич, Б.А. Заварзин // Российская оториноларингология. – 2004. – № 4(11). – С. 148–150.
2. Eiber, A. On the coupling of prosthesis to the middle ear structure and its influence on sound transfer / A. Eiber [et al.] // The Function and Mechanics of Normal, Diseased and Reconstructed Middle Ear. Proceedings of the Second International Symposium on Middle-Ear Mechanics in Research and Otorrhinology, held in Boston, MA, USA, October 21st–24th, 1999 / edited by J.J. Rosowski, S.N. Merchant. – The Hague, The Netherlands: Kugler Publication, 2000. – P. 297–308.
3. Koike, T. Analysis of the finite-element method of transfer function of reconstructed middle ear and their postoperative changes / T. Koike, H. Wada, T. Kobayashi // The Function and Mechanics of Normal, Diseased and Reconstructed Middle Ear. Proceedings of the Second International Symposium on Middle-Ear Mechanics in Research and Otorrhinology, held in Boston, MA, USA, October 21st–24th, 1999 / edited by J.J. Rosowski, S.N. Merchant. – The Hague, The Netherlands: Kugler Publication, 2000. – P. 309–320.
4. Любина, Е.А. Влияние сил натяжения восстановленной барабанной перепонки, вызванных введением протеза, на собственные частоты колебательной системы реконструированного среднего уха / Е.А. Любина, Г.И. Михасев // Механика – 2007: сб. науч. тр. III Белорус. конгресса по теоретической и прикладной механике, Минск, 16–18 окт. 2007 г. / НАН Беларуси, Объединенный институт машиностроения; под общ. ред. М.С. Высоцкого. – Минск, 2007. – С. 399–405.
5. Beer, H.-J. Modeling of Components of the Human Middle Ear and Simulation of Their Dynamic Behavior / H.-J. Beer [et al.] // Audiology & Neuro-Otology. – 1999. – Vol. 4. – P. 156–162.
6. Ермоченко, С.А. Напряженно-деформированное состояние реконструированного среднего уха: дис. ... канд. физ.-мат. наук: 01.02.04 / С.А. Ермоченко; Белорус. нац. техн. ун-т. – Минск, 2011. – 94 л.
7. Самарский, А.А. Численные методы / А.А. Самарский, А.В. Гулин. – М.: Наука, 1989. – 432 с.
8. Schwartz, B. High Performance MySQL. Optimization, Backups, and Replication Third Edition / B. Schwartz, P. Zaitsev, V. Tkachenko. – 3rd Edition. – O'Reilly Media, 2012. – 826 p.
9. Ермоченко, С.А. Применение объектно-ориентированного программирования для реализации операций над матрицами / С.А. Ермоченко, О.Г. Казанцева, В.В. Новый // Весн. Віцеб. дзярж. ун-та. – 2013. – № 6(78). – С. 14–21.

REFERENCES

1. Sitnikov V.P., Chernushevich I.I., Zavarzin B.A. *Rossiiskaya otorinolaringologiya* [Russian Otorhinolaryngology], 2004, 4(11), pp. 148–150.
2. Eiber, A. On the coupling of prosthesis to the middle ear structure and its influence on sound transfer / A. Eiber [et al.] // The Function and Mechanics of Normal, Diseased and Reconstructed Middle Ear. Proceedings of the Second International Symposium on Middle-Ear Mechanics in Research and Otorrhinology, held in Boston, MA, USA, October 21st–24th, 1999 / edited by J.J. Rosowski, S. N. Merchant. – The Hague, The Netherlands: Kugler Publication, 2000. – P. 297–308.
3. Koike, T. Analysis of the finite-element method of transfer function of reconstructed middle ear and their postoperative changes / T. Koike, H. Wada, T. Kobayashi // The Function and Mechanics of Normal, Diseased and Reconstructed Middle Ear. Proceedings of the Second International Symposium on Middle-Ear Mechanics in Research and Otorrhinology, held in Boston, MA, USA, October 21st–24th, 1999 / edited by J.J. Rosowski, S.N. Merchant. – The Hague, The Netherlands: Kugler Publication, 2000. – P. 309–320.
4. Liubina E.A., Mikhasev G.I. *Mekhanika – 2007: Sbornik nauchnykh trudov III belorusskogo kongressa po teoreticheskoi i prikladnoi mekhanike, Minsk, 16–18 oktiabria 2007 g., NAN Belarusi, obyedinennii indtitut mashinistroyeniya* [Mechanics – 2007: Proceedings of the 3rd Belarusian Theoretical and Applied Mechanics Congress, Minsk, October 16–18, 2007, NASc of Belarus, Joint Institute of Machine Building], Minsk, 2007, pp. 399–405.
5. Beer, H.-J. Modeling of Components of the Human Middle Ear and Simulation of Their Dynamic Behavior / H.-J. Beer [et al.] // Audiology & Neuro-Otology. – 1999. – Vol. 4. – P. 156–162.
6. Yermochenko S.A. *Napriazhenno-deformirovannoye sostoyaniye rekonstruirovannogo srednego ukha: dissertatsiya kandidata fiziko-matematicheskikh nauk* [Strain-Stress State of the Reconstructed Middle Ear: PhD (Physics and Mathematics) Dissertation], Minsk, 2011, 94 p.
7. Samarski A.A., Gulin A.V. *Chislennyye metody* [Numerical Methods], Moscow: Nauka, 1989, 432 p.
8. Baron Schwartz, Peter Zaitsev, Vadim Tkachenko. High Performance MySQL, 3rd Edition. Optimization, Backups, and Replication Third Edition. – O'Reilly Media, 2012. – 826 p.
9. Yermochenko S.A., Kazantseva O.G., Novii V.V. *Vesnik Vitsebskaga dziazhaunaga universiteta* [Journal of Vitebsk State University], 2013, 6(78), pp. 14–21.

Поступила в редакцию 26.08.2019

Адрес для корреспонденции: e-mail: yermochenko@gmail.com – Ермоченко С.А.