Research Article

Haifeng Ma*

Development of a CPU-GPU heterogeneous platform based on a nonlinear parallel algorithm

Abstract: In order to seek a refined model analysis software platform that can balance both the computational accuracy and computational efficiency, a CPU-GPU heterogeneous platform based on a nonlinear parallel algorithm is developed. The modular design method is adopted to complete the architecture construction of structural nonlinear analysis software, clarify the basic analysis steps of nonlinear finite element problems, so as to determine the structure of the software system, conduct module division, and clarify the function, interface, and call relationship of each module. The results show that when the number of model layers is 10, the GPU is 210.5/s and the CPU is 1073.2/s, and the computational time of the GPU is significantly better, with an acceleration ratio of 5.1. For all the models, the GPU calculation time is much less than that of the CPU, and when the number of model degrees of freedom increases, the acceleration effect of the GPU becomes more obvious. Therefore, the CPU-GPU heterogeneous platform can more accurately describe the nonlinear behavior in the complex stress states of the shear walls, and is computationally efficient.

Keywords: heterogeneous platform, GPU parallel computing, nonlinear finite element analysis program

1 Introduction

With the rapid growth of engineering and scientific computing, even the current multi-core processors can not meet the required computational complexity [1]. Although supercomputers with high-performance servers or even clustered systems can meet certain computing needs, these machines are expensive for maintenance and energy consumption, and only a few large companies and research units have them. In order to meet the needs of the computing capabilities of most small and medium enterprises, research institutes, and individuals, various dedicated computing chips were created, such as Field Programmable Gate Array (FPGA) [2]. Although the computational power of these dedicated computing chips is ten times more than a general CPU, its function is fixed, it is often only possible to accelerate a particular type of algorithm. In addition, its price is more expensive, complicated, and is not suitable for personal and small laboratories. The emergence of parallel computing is the result of social development needs, more accurate weather forecast, analysis of human gene sequences, large-scale engineering system design simulation, massive data, information mining, etc. Relevant science and engineering areas require high strength calculations, earlier the serial calculation method was very hard in the face of such problems. There is a large number of intrinsic parallelities in the calculation of these applications, so these scientific puzzles can be solved by powerful computing performance provided by parallel computing [3]. The essence of parallel computing is to decompose a complex large problem into several simple small problems, which can be solved by multi-threaded computing on multiple processors, so as to solve complex problems more quickly.

Based on this, this article proposes a CPU-GPU heterogeneous platform development based on nonlinear parallel algorithm. According to the nonlinear finite element theory, it is fully understood that the basic numerical solution of the nonlinear finite element of the material and the basic numerical solution of nonlinear equations are determined to determine the structure of the software, and the function interface and the calling relationship between modules are defined.

According to the characteristics of strong CPU logical computing power, sufficient cache space, and superior GPU floating-point computing performance, task is assigned to the heterogeneous platform as shown in Figure 1. The part of calling GPU for calculation is implemented by calling

* Corresponding author: Haifeng Ma, Department of Information Engineering, Qinhuangdao Vocational and Technical College, Qinhuangdao, 066100, China, e-mail: HaifengMa823@126.com

https://doi.org/10.1515/nleng-2022-0027
received December 10, 2021; accepted April 24, 2022

Open Access. © 2022 Haifeng Ma, published by De Gruyter. This work is licensed under the Creative Commons Attribution 4.0 International License.
cuBLAS library and cuSPARSE library, which function to complete the calculation between the vectors and the operation between matrix and vectors, respectively. In order to shorten the development cycle and make full use of the existing open source finite element program code, the reverse engineering analysis of the large-scale general open source finite element program is carried out to master its overall architecture and functional module design, and compared with the nonlinear finite element software designed in this paper. Based on object-oriented design ideas, the details of each module are designed in C++ language, and the structural nonlinear finite element analysis problem and its numerical solution are transformed into computer programs, and based on modular methods, the structural elastoplastic analysis software architecture is completed. This method demonstrates that the CPU-GPU heterogeneous platform can more accurately describe nonlinear behavior in the complex strategy of shear walls, and with higher computational efficiency.

2 Literature review

In order to successfully complete the parallel calculation tasks, first three essential conditions are to be satisfied: first, there must be equipment with parallel computing power. Second, the task that needs to be calculated can be solved by breaking it into multiple relatively independent sub-tasks, then these independent sub-tasks can be performed in parallel to achieve parallel solutions for the target [4]. Third, in the parallel equipment, the corresponding programming environment is configured, and the corresponding algorithm is designed according to the specific problem design, and the program executed in parallel is implemented by a valid programming language, and ultimately solved in parallel. Based on GPU-based general calculations, the problem is solved, it is an important branch of parallel computing. Under the double stimulation of the user’s more visual experience and the demand for military field simulation, GPU’s computational power increased, far exceeding the current CPU’s development speed. The main responsibility of the GPU is the calculation and rendering of complex graphics. The graphical rendering process has a high degree of parallelism. The generation of the vertex, the generation of the element generation, and the generation process of the unit are almost independent, which makes it perform their calculations in parallel [5]. Given the GPU in parallelism and powerful computing power, it is desirable to handle data calculations other than graphical calculations by means of its powerful computing power, which is GPGPU (based on GPU-based general calculation). Modern computers basically consist of CPU and GPU, which bring gospel to most parallel program developers, but the relatively low price GPU can not provide powerful computing power. Hou et al. introduced the parallel tabu search algorithm (GPT) based on GPU-based PATPUL for HW/SW partitions. A single GPU core compacted in the neighborhood is proposed, which theoretically reduces the amount of GPU global memory. A kernel fusion strategy is further proposed to reduce GPT GPU global memory visits. In order to further minimize the transfer overhead of the GPT between the CPU and GPUs, an optimized transfer strategy based on the GPU-based TABU assessment is proposed, which considers that all candidates do not satisfy a given constraint. Experiments have shown that GPTS is better than tabu search, and it is competitive with other HW/SW partitioning methods. When considering a normal GPU platform, the proposed parallelization is significant [6]. A framework, supporting perceptively delayed data initialization on an integrated heterogeneous platform, is proposed by Wang et al. The framework not only includes three data initialization modes, central processor initialization, GPU initialization, and initialization and mixing, but also wisely utilizes an affinity estimation model to determine the best application initialization mode, which can optimize the initialization delay performance of the application. Design was evaluated on the NVIDIA TX2 and AGX platforms. It is shown that the proposed framework is able to accurately select data initialization patterns for a given application, thus significantly reducing the initialization delay. We envision that this delay-aware data initialization framework will be adopted in a future full version of an autonomous solution (such as Autoware) [7]. Currently, GPU achieves
high-throughput computation by running large numbers of threads. Li et al. introduced xflow, which enables streamlined execution by leveraging the hardware mechanism in the new generation of GPU. The xflow significantly reduces the cost of explicit copying and kernel startup in existing ways. As an alternative, xflow introduces a persistent operator that continuously processes data by sharing topics and establishes efficient interprocessor data channels through hardware page failures. To demonstrate its potential, two applications are also evaluated for GPU acceleration, including data encoding and OLAP queries [8]. Hybrid multi-core processors will dominate the next-generation of computing. By integrating several types of cores into a single chip, designers expect continued performance growth, while reducing reliance on raw circuit speed and reducing the power demand per unit of performance. The performance/energy tradeoff problem of a heterogeneous platform consisting of Intel multicore processors and multiple NVIDIA GPU coprocessors was investigated by Gadou et al. Using CMT-bone, an agent application designed by the University of Florida, we explored the load-balancing strategies for various combinations of CPU and GPU architectures to optimize performance, power, and energy measurements. (C) 2017 Elsevier Inc. All rights reserved [9].

3 Research methods

3.1 Modular design of the nonlinear software for a heterogeneous platform

Nonlinear finite element problems can usually be divided into three kinds: (1) material nonlinearity, structural nonlinearity of the material constitutive relationship still satisfies the assumption of infinite small displacement, (2) geometric nonlinearity, mainly shows the relationship between displacement and strain is nonlinear including large strain and geometric small deformation, and (3) boundary nonlinearity, the structural nonlinear effects of the boundary conditions, in which the contact problem is most typical. In the field of civil engineering, structural nonlinearity is generally caused by material nonlinearity. The nonlinearity mentioned in this article considers only the nonlinearity of the material. The material nonlinearity problem is a small deformation elastoplastic problem, considering only the nonlinearity of the material constitutive relation, satisfying the assumption of infinite small node displacement, and thus not considering the change in node coordinates [10]. The solution of the structural nonlinear problem by finite element method is essentially the solution of a system of equilibrium equations nonlinear with the node shift as basically unknown quantity, as shown in formula (1):

\[ K(\delta)\delta = P. \]

where \( \delta \) is the displacement vector; \( K(\delta) \) is the structural total rigid matrix; and \( P \) is the load vector.

The iterative method and incremental method are commonly used in nonlinear finite element analysis. The basic process of the iterative method is first determined to determine the initial displacement \( u^0 \), the next step \( u^{n+1} \) can be determined using the formula \( u^{n+1} = (K)^{-1}F \), where \( K^n \) needs to be determined and updated according to the unit state of each step. The basic idea of the incremental method is to specifically divide the load into several increments when performing structural non-linear finite element analysis. Each load incremental step is formed by linear processing, solving the displacement increment of each node, the cumulative load of the current step is transferred, and the state of the unit is active, and the structural rigidity matrix is reprocessed according to the state, and the new level load increment is applied to the structure or member. Nonlinear finite element analysis basic flow is shown in Figure 2.

3.2 Nonlinear software FEM model is modular based on CPU platform architecture

The finite element domain class describes the finite element model, manages and stores the instance objects of node class, unit class, material class, section class, and load class, providing a unified interface for the information exchange between node class, unit class, material class, and analysis class. In the finite element analysis process, the information exchange includes unit class and load class to transfer the corresponding stiffness matrix and node load information to the analysis class, and return the updated node displacement information to the node class in the analysis class [11]. Also, the finite element domain class creates and initializes its own objects based on the information provided by the modeling class. When using elastoplastic analysis, the relationship between stress and strain is nonlinear, with the difficulty of the tangent modulus. In order to solve this problem, the procedure used to solve the stress increment to determine the current
stress, according to the ratio of strain increment, which is an effective algorithm to solve the material constitutive matrix [12,13]. The main calculation steps are as follows:

1. Material elastic modulus calculation stress $\sigma_{n+1}^\text{trial}$: as shown in formula (2):
   $$\sigma_{n+1}^\text{trial} = \sigma^n + D_{el} \Delta \varepsilon.$$

2. Calculate the value of the production function $f_{n+1}^\text{trial}$ under the test force.

3. Test yield conditions

4. If in the yield surface, $\sigma_{n+1}^\text{trial} = \sigma_{n+1}^\text{trial}$, return the elastic modulus, the $\sigma_{n+1}$ and tangent modulus should be recalculated according to the constitutive relationship according to the plastic stage.

5. Update the material stress and tangential modulus, and the status variable of the material.

6. In order to achieve the application of software in the field of civil engineering, the process of concrete one-dimensional primary model under monotonous load is prepared, derived from material, inheriting the basic properties and methods of materials, and used for materials the method of material stress and tangent rigidity is overloaded [14,15]. As shown in the formulas (3)–(6):
   $$y = \frac{n x}{\alpha_c (x - 1)^2 + x^n}, \quad x > 1,$$

   $$x = \frac{\varepsilon}{\varepsilon_{ek}},$$

   $$\varepsilon_{ek} = f_{ek}^{0.34} \times 65 \times 10^{-6},$$

   $$a_t = 0.312 f_t^2.$$

For the compression stage: as shown in formulas (7)–(12):

$$y = \frac{n x}{n - 1 + x^n}, \quad x \leq 1,$$

$$y = \frac{x}{\alpha_c (x - 1)^2 + x^n}, \quad x > 1,$$

$$x = \frac{\varepsilon}{\varepsilon_{ek}},$$

$$y = \frac{\sigma}{f_{ek}},$$

$$\alpha_c = 0.715 f_c^{0.285} - 0.905,$$

$$n = \frac{E \varepsilon_{ek}}{E \varepsilon_{ek} - f_c},$$

In the design of the nonlinear analysis program, when the strain reaches the ultimate concrete pressure strain, consider that the structure concrete is damaged, the structure reaches the ultimate bearing capacity, the unit related to the material quits, without considering the concrete softening effect [16,17].

### 3.3 Design and implementation of analytical modules based on heterogeneous platform

The key to the development of finite element computing software based on heterogeneous platform is to coordinate the CPU platform and the GPU platform work together and reasonably call the implementation of the finite element calculation process. This article designed software which is a high-efficiency structure nonlinear analysis software based on heterogeneous platforms by completing the use of CUDA’s use of the GPU. The use of CUDA first requires an anti-Weida driver for the current operating system [18,19]. Second, configure the environment you need to run in the program development tool Visual Studio. Finally, the CUDA library is added to the GPU-based mathematical computational library cuSPARSE (sparse matrix) and cuBLAS (linear algebraic libraries), so that the CPU and GPU collaborative completion analysis calculations are achieved when the program is running [20,21]. According to the CPU logic capacity, the cached space is sufficient and the GPU floating
point computational performance is superior, and the heterogeneous platform is assigned as shown in Figure 3. The part in which the cuBLAS library and the cuSPARSE library implementation are called by calling the cuBLAS library and the cuSPARSE library, and the functions are completed between the vectors and the operation between the matrices and the vector, respectively [22,23].

4 Result analysis

4.1 Based on static elastic analysis of beam and column structure of heterogeneous platform

Example 1 is a high-rise reinforced concrete frame, whose plane layout and elevation layout are shown in Figure 4. The column section is square with concrete with C40 strength grade, and the beam section is rectangular with C30 concrete. The floor constant load is 3.0 kN/m² and the live load is calculated as 2.0 kN/m². The load condition adopts 1.0 times constant load + 0.5 times live load combination. The structure is analyzed under gravity load [24,25].

The framework model is analyzed by ABAQUS and the development software, the results are shown in Table 1. The calculation results of the software are highly consistent with the ABAQUS analysis results and the error control is within 1%, verifying the correctness of the program developed in this article [26,27]. It also shows that the software is verified in the software. By analyzing the framework model under the CPU solver of the software and the heterogeneous platform-based solver, the CPU computing time is 22.7 s and the heterogeneous platform computing time is only 4.496 s. The computing efficiency of the software is significantly improved, and the efficiency of the nonlinear software based on the heterogeneous platform structure is initially verified. To further verify the computing efficiency of the software, more example analysis is needed in the later stage [28,29].

Example 2 is a multi-storey reinforced concrete frame with a height of 3.6 m with 6 layers, and the structure layout is shown in Figure 5. Column section is square, area 0.25 m², beam section width is 250 mm, and height is 600 mm. The floor constant load is 5.0 kN/m² and the live load is calculated as 2.0 kN/m². The characteristic period is 0.35 s, structure is 0.912 s. C30 concrete and grade HRB33 reinforcement were used. Statistic elastic-plastic analysis of the frame structure under gravity load and horizontal earthquake [30,31] were performed.

![Figure 3: A heterogeneous platform design.](image)

![Figure 4: Example framework model. (a) Elevation and (b) plan.](image)

| Floor number | SOR-PCG (mm) | ABAQUS (mm) | Error (%) |
|--------------|--------------|-------------|-----------|
| 1            | -0.80774     | -0.80142    | 0.788     |
| 2            | -1.53207     | -1.51839    | 0.901     |
| 3            | -2.17417     | -2.15399    | 0.937     |
| 4            | -2.73468     | -2.71007    | 0.908     |
| 5            | -3.21418     | -3.18456    | 0.930     |
| 6            | -3.61317     | -3.57718    | 1.006     |
| 7            | -3.93204     | -3.89241    | 0.972     |
| 8            | -4.17113     | -4.12959    | 1.006     |
| 9            | -4.33062     | -4.28749    | 1.006     |
| 10           | -4.44112     | -4.39915    | 0.954     |
Established model and the generated data file, and the ABAQUS and this software are shown in Table 2 and Figure 6. From the error between the figure and ABAQUS analysis, we prove the accuracy of the software in structural static nonlinear calculation [32,33]. In order to test the effectiveness of high-performance computing software, this article analyzes the high-level framework model with the same structure layout using the CPU platform and the heterogeneous platform based software, and calculates the calculation time required for the same model based on the two platforms [18,33]. The results are shown in Table 3. It is seen from Table 3 that when the number of model layers is 10, GPU is 210.5/s, CPU is 1073.2/s, and GPU's computational time is significantly superior, with an acceleration ratio of 5.1. For all models, GPU computation time is much less than CPU, especially when the acceleration effect of GPU becomes more obvious, the number of model degrees of freedom increases [34,35].

5 Conclusion

Based on the combination of CPU serial computing and GPU high-performance parallel computing, we established a heterogeneous platform of CPU-GPU hybrid programming, proposed the CPU-GPU heterogeneous platform development based on nonlinear parallel algorithm, conducted program implementation, and successfully added to the solution library, accelerating the solution speed of the software. Using multiple parallel optimization strategies suitable for GPU computing, we improved the computational efficiency of large sparse linear equations through the elastic analysis of the solver based on CPU, and verified the reliability and effectiveness of the heterogeneous platform. The results show that the spatial shell element established by CPU-GPU heterogeneous platform can accurately describe the nonlinear behavior in the complex stress state of shear wall and have high computational efficiency. Although this article is based on CPU-GPU heterogeneous platform for building structure elastically, plastic dynamic time range reaction refined numerical model and numerical algorithm made more detailed research and made some research results, but based on CPU-GPU heterogeneous platform parallel software development, need to combine the latest computer technology, finite element theory, algorithm research, software tools, is a long-term and persistent work.
Funding information: The author states no funding involved.

Author contributions: The author has accepted responsibility for the entire content of this manuscript and approved its submission.

Conflict of interest: The author states no conflict of interest.

References

[1] Song P, Zhang Z, Liang L, Zhang Q, Zhou Y. Study on optimization of parallel efficiency of CPU-GPU heterogeneous parallelization for MOC neutron transport calculation. Yuanzining Kexue Jishu/Atomic Energy Sci Technol. 2019;53(11):2209–17.

[2] Song P, Zhang Z, Zhang Q, Liang L, Zhao Q. Implementation of the CPU/GPU hybrid parallel method of characteristics neutron transport calculation using the heterogeneous cluster with dynamic workload assignment. Ann Nucl Energy. 2020;135:106957.1–12.

[3] Lee C, Lee HJ. Effective parallelization of a high-order graph matching algorithm for GPU execution. IEEE Trans Circuits Syst Video Technol. 2019;29(2):560–71.

[4] Allc SI, Sun Y, Sun J, Chang C, Wong B. Heterogeneous CPU + GPU-enabled simulations for large chemical and biological systems. J Chem Theory Computation. 2019;15(5):2807–15.

[5] Li D, Zhang Z, Yu K, Huang K, Tan T. IEEE: an intelligent scene exploration and evaluation platform for large-scale visual surveillance. IEEE Trans Parallel Distrib Syst. 2019;30(12):2743–58.

[6] Hou N, He F, Zhou Y, Chen Y. An efficient GPU-based parallel tabu search algorithm for hardware/software co-design. Front Computer Sci. 2020;14(5):1–18.

[7] Wang Z, Jiang Z, Wang Z, Tang X, Hu Y. Enabling latency-aware data initialization for integrated CPU/GPU heterogeneous platform. IEEE Trans Comput Des Integr Circuits Syst. 2020;39(11):1.

[8] Li Z, Peng B, Weng C. XFlow: Streamlining inter-processor pipeline execution for the discrete CPU-GPU platform. IEEE Trans Computers. 2020;69(6):819–31.

[9] Gadoiu M, Banerjee T, Arunachalam M, Ranka S. Multiobjective evaluation and optimization of CMT-bone on multiple CPU/GPU systems. Sustain Comput. 2020;22(JUN):259–71.

[10] Gárate KZ. GPU parallel visibility algorithm for a set of segments using merge path – scientiedirect. Electron Notes Theor Computer Sci. 2019;342(5):57–69.

[11] Munk DJ, Kipouros T, Vio GA. Multi-physics bi-directional evolutionary topology optimization on GPU architecture. Eng Computers. 2019;35(3):1059–79.

[12] Escobar JJ, Ortega J, Diaz AF, Gonzalez J, Damas M. Energy-aware load balancing of parallel evolutionary algorithms with heavy fitness functions in heterogeneous CPU-GPU architectures.Concurrency Comput Pract Exp. 2019;31(6):e4688.1–15.

[13] Chrysogelos P, Karpathiotakis M, Appuswamy R, Ailamaki A. HetExchange: encapsulating heterogeneous CPU-GPU parallelism in JIT compiled engines. Proc of the VLDB Endowment. 2019;12(5):544–56.

[14] Zhao T, Yang C, Li Y, Gan Q, Wang Z, Liang F, et al. Space4gnn: a novel, modularized and reproducible platform to evaluate heterogeneous graph neural network. Cornell University. 2022.

[15] Kurth A, Forsberg B, Benini L. Herov2: full-stack open-source research platform for heterogeneous computing. Cornell University. 2022.

[16] Salami B, Noori H, Naghibzadeh M. Online energy-efficient fair scheduling for heterogeneous multi-cores considering shared resource contention. J Supercomput, 2022;78:729–7748.

[17] Bai L, Gong C, Chen X, Zheng J, Yang J, Li K, et al. Heterogeneous compressive responses of additively manufactured Ti-6Al-4V lattice structures by varying geometric parameters of cells. Int J Mech Sci. 2022;214:106922.

[18] Gan S, Zeng Y, Liu J, Nie J, Lu C, Ma C, et al. Click-based conjugated microporous polymers as efficient heterogeneous photocatalysts for organic transformations. Catal Sci Technol. 2022;12:1202–10.

[19] Sharma K, Chaurasia BK. Trust Based Location Finding Mechanism in VANET Using DST. Fifth International Conference on Communication Systems & Network Technologies; 2015 Apr 4–6; Gwalior, India. IEEE: 2015;763–6.

[20] Li Z, Di M, Zhang Y, Zhang B, Zhang Z, Zhang Z, et al. Covalent triazine frameworks with palladium nanoclusters as highly efficient heterogeneous catalysts for styrene oxidation. ACS Appl Polym Mater. 2022;4:1047–54.

[21] Kokiz R, Pawlicki M, Kula S, Chora M. Fake news detection platform—conceptual architecture and prototype. Log J IGPL. 2022. Special issue CISIS 2020-IGPL.

[22] Litany O, Maron H, Acuna D, Kautz J, Chechik G, Fidler S. Federated learning with heterogeneous architectures using graph hypernetworks. arXiv e-prints; 2022.

[23] Varvello M, Katevas K, Plesa M, Haddadi H, Bustamante F, Livihsits B. BatteryLab: A collaborative platform for power monitoring. International Conference on Passive and Active Network Measurement. 2022;13210:97–121.

[24] Liu Z, Huang Y, Huang Y, Song YA, Kumar A. How does one-sided versus two-sided customer orientation affect b2b platform’s innovation: differential effects with top management team status. J Bus Res. 2022;141:631–632.

[25] Bruschi N, Haugou G, Tagliavini G, Conti F, Benini L, Rossi D. GVSoC: a highly configurable, fast and accurate full-platform simulator for RISC-V based IoT processors. Cornell University. 2022.

[26] Nabi S, Sofi FA, Rashid N, Ingle P, Bhat MA. Metal–organic framework functionalized sulphur doped graphene: a promising platform for selective and sensitive electrochemical sensing of acetaldehyde, dopamine and H2O2. N J Chem. 2022;46:1588–600.

[27] Li J, Das A, Ma Q, Bedzyk MJ, Kratis Y, Marks TJ. Diverse mechanistic pathways in single-site heterogeneous catalysis: alcohol conversions mediated by a high-valent carbon-supported molybdenum-dioxo catalyst. ACS Catal. 2022;12:1247–57.

[28] Liang Q, Cheng H, Li C, Ning L, Shao L. A covalent modification strategy for di-alkyne tagged metal–organic frameworks to access efficient heterogeneous catalysts toward C–C bond formation. N J Chem. 2022;46:1221.
[29] Mirhosseini M, Fazlali M, Tabatabaeemalazi H, Izadi SK, Nezamabadi-Pour H. Parallel quadri-valent quantum-inspired gravitational search algorithm on a heterogeneous platform for wireless sensor networks. Computers Electr Eng. 2021;92(4):107085.

[30] Rd B, Egds A, Pp A. High performance computing of stiff bubble collapse on CPU-GPU heterogeneous platform. Computers Math Appl. 2021;99:246–56.

[31] Ke J, Li X, Yang H, Yin Y. Pareto-efficient solutions and regulations of congested ride-sourcing markets with heterogeneous demand and supply. Transport Res Part E Logist Transport Rev. 2021;154:102483.

[32] Liu Y, Chen M, Xu X. Forest rainfall characteristics based on heterogeneous computing and influencing factors of athletes' physical supplement. Arab J Geosci. 2021;14(15):1516.

[33] Senapati D, Sarkar A, Karfa C. PRESTO: A penalty-aware real-time scheduler for task graphs on heterogeneous platforms. IEEE Trans Computers. 2021;71(2):421–35.

[34] Yu L, Liu M, Zhang Y, Ni Y, Wu S, Liu R. Magnetically induced self-assembly DNAzyme electrochemical biosensor based on gold-modified α-Fe$_2$O$_3$/Fe$_3$O$_4$ heterogeneous nanoparticles for sensitive detection of Ni$^{2+}$. Nanotechnology. 2022;33(9):095601.

[35] Ma J, Tran G, Wan AMD, Young E, Zandstra PW. Microdroplet-based one-step RT-PCR for ultrahigh throughput single-cell multiplex gene expression analysis and rare cell detection. Sci Rep. 2021;11(1):6777.