Acta Technica *62* No. 4B/2017, 1–8

A data-centered materials computing platform with distributed task dispatching[1]

Qinggong Ding[2], Guannan Hu[2], Wenhao Zhu[2,5], Jiong Yang[3], Zhiguo Lu[4], Wu Zhang[2,3]

Abstract. Due to the urgent requirement of novel materials, materials computing, which can be used to predict properties without investment of real experiment, is becoming a hot research topic. Available materials platforms are usually implemented in a fully integrated way. That is, computing tasks submitted in these platforms are dispatched to a private cluster to ensure the integration of result data. Such kind of integration could be essential for materials discovery, however, the use of a private cluster also introduces the limitation of computing capability, which will make the platform inefficient. In this paper, a Materials Computing Platform is introduced. Computation efficiency could be largely improved by dispatching computing tasks to user side clusters instead of server side ones. Meanwhile, an integrated data storage is implemented along with the platform to ensure the integration of data. Additionally, computing templates are used to enhance platform usability and users can carry out high-throughput computing tasks with simple setup steps.

Key words. Materials platform, materials computing platform, materials discovery, task dispatch..

1. Introduction

Research methods of traditional materials science are to develop research contents, work steps and experimental means according to specific goals, and then, use mathematical methods to analyze and process the resulting data from experiments. Scientific report and mathematical models may be established via such a research

[1]The work of this paper is supported by National Natural Science Foundation of China (No. 91630206).

[2]School of Computer Engineering and Science, Shanghai University, Shanghai 200444, China
[3]Materials Genome Institute, Shanghai University, Shanghai 200444, China
[4]Library, Shanghai University, Shanghai 200444, China
[5]Corresponding author; e-mail: whzhu@shu.edu.cn

work flow.

Clearly, this kind of flow may involve of expensive and time consuming experiments along with huge investments. With the rapid development of computer technology, materials computing is becoming an alternative way of material research. On basis of that, it has become an important supplementary ways of material design that adopting supercomputers (usually computing clusters) to simulate the composition, structure, performance and service performance of materials with certain compositions and structures and then using simulation results to assist material discovery. As a result of integration, a lot of material computing and data platforms are proposed.

These platforms, e.g. Materials Project (MP) [1], Automatic-FLOW (AFLOW) [2], etc., which not only provide searching and downloading of computational data, but also provide data analysis and services of high-throughput material computing [3]–[5], are usually based on Browser/Server architecture. With the help of material computing platforms, high-throughput material computing can be carried out. The iteration of the trial-and-error material development process can be repeated recursively and automatically according to user request. During the iterations, characteristics with certain physical property and microstructure can be 'designed' based on data analysis of calculation result. Such kind of high-throughput process will significantly speed up material discovery and reduce the costs.

Obviously, data plays a vital role in the high-throughput computing process. Thus, in order to achieve standardized storage and efficient search, existing platforms are generally designed as a centralized structure. Most of these platforms regard a certain supercomputer as its calculation support to ensure the centralized usage and the retrieval of data. Although this architecture design can ensure the centralized use of data to certain extent, it may bring about the problem of insufficient computing resources. In other words, computing resources integrated inside of the platform may not be able to meet the needs of large-scale computing.

Regarding to this problem, this paper proposes a Data-Centered Materials Computing Platform (DCMCP) with distributed task dispatching. The platform is based on ADDS model (Automation, Data, Dispatch and Sharing), which implements a distributed dispatch of material calculation tasks around a data center. Based on the data center, the platform allocates calculation tasks for each computing cluster with automatic concurrency module of distributed process, and then imports the result data from each cluster to data center through user configuration interface.

2. The ADDS model for material simulation

The aim of this chapter is to introduce the ADDS model, as shown in Fig. 1, to analyse the model rationality and the impetus for material simulation from both theoretical and technical aspects.

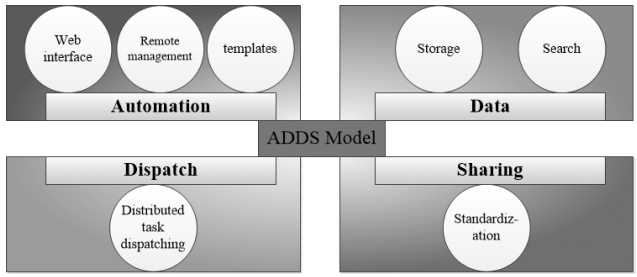

Fig. 1. ADDS model

2.1. Automated process

According to the workflow of material calculation task, including design automated submission, implementation, error detection and correction, the functional requirements are illustrated as follows:

The first section is Web Interface. In DCMCP, users can perform material calculations through the web services front-end page, including data query, calculation task submission, cluster registration, calculation result download and calculation task query, etc. The web daemon receives the user request, and then generates the task after the specified web service transforms requests as instructed. Web interface provides a simple and intuitive material simulation interface, thus blurring the differences in computing environment and simplifying the operation of material simulation.

The second section is Remote management. Remote management is divided into two parts, one is the server program on central server, and the other part is the client program on computing cluster. The server program distributes the data and instructions to the specified client program and retrieved the calculation results. The client program mainly contains the following functions: (1) Configuring environment of material simulation. (2) Accepting and executing instructions. (3) Performing submission, error detection and auto error correction of calculation task. (4) Monitoring status of computing cluster.

The last section is Task Template. Most of the material calculation tasks have the following main features: (1) Invoke multiple calls to the calculation software. (2) Consistent internal logic in simulating the same property. (3) Require manual calculation data and estimate through calculation process.

2.2. Data processing

The calculated and verified simulation data provides calibrations and inspirations for new material calculation tasks. This part has two portions as follows:

A material calculation task produces a large number of data, including input data, calculated result, and calculation task information. The number can reach to hundreds of Mbyte. When the frequency of material calculation tasks reaches to tens of thousands of magnitude, the total amount of data will be very large. At the same

time, due to the diversity of software, calculation tasks and computing environments, the isomerization of the data produced by the same kind of task is quite serious, which indirectly raises the amount of data. Therefore, the key point of the storage strategy is to address the massive and heterogeneous data properly. Distributed storage system which is scalable can solve the above problems and improve the speed of data searching [7].

2.3. Dispatching system

Due to fact that user's computing clusters are treated as computing resources, whether the calculation tasks and computing resources can be effectively dispatched is the key to complete the calculation task quickly.

Distributed task dispatch module, consists of dispatching calculation tasks, optimization and processing of error calculation tasks. Dispatching calculation tasks need to ensure high utilization of calculation resources and short run cycles of calculation tasks. By using the general configuration of the same calculation task, the calculation task optimization strategy is to modify the input parameters of the current calculation task such as the number of CPU core, the memory size, precision, convergence steps, etc. to speed up the execution of calculation tasks, and reduce calculation errors. For processing of error calculation tasks, we encode them as an error correction program by a large number of correct input parameters of calculation tasks and the consensus experience of the material researchers, and the error signals returned by the template objects are inputs of the program.

2.4. Sharing

In addition to the basic data sharing, the distributed material calculation has calculation resources sharing. Whether data and calculation resources in the same format can be used by more people, standardization is the key point.

Standardization is beneficial t o d ata s haring a nd t he g eneration o f n ew data, and related work has been launched. For example, MP uses the Python Materials Genomics package to store material data in the same format [6]. Another example is NoMaD [8], who stores its data in xml, JSON format. Because most of material calculation software have become commercial, therefore, some users' computers are unable to implement material simulations. Thus, sharing resources is very meaningful for most people. Standardizing data and computing resources can dramatically improve the efficiency of material calculation and data search.

3. DCMCP framework

DCMCP is built based on the ADDS model and its task is to provide distributed material calculations, data search and cluster registration services, as shown in Fig. 2. DCMCP consist of four independent subprojects. The first project is the web interface. The second project is data standardization and storage. The third project is the remote management program based on C/S model. The last one is the function

template.

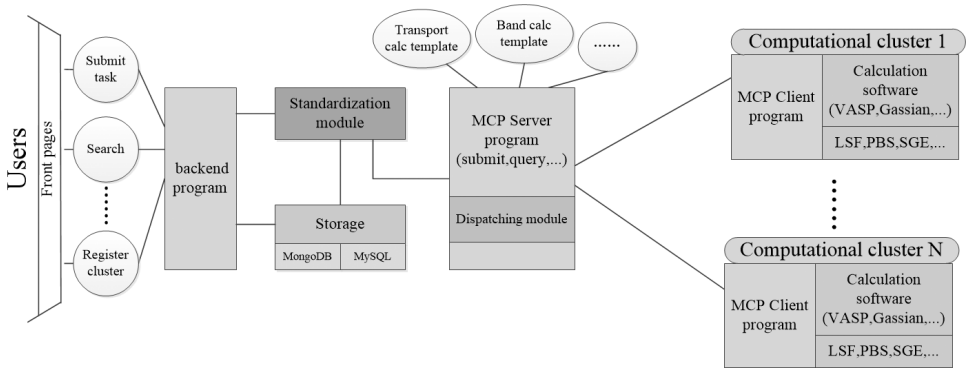

Fig. 2. DCMCP framework

3.1. Web interface

The web interface provides users with an operator GUI so that users can perform data interaction. And it offers cluster registration, searching and task submission services. Cluster registration requires information about SSH login and the environment information of computing clusters such as queue systems, computing software, etc. Search includes the task query and the data search. The task query is visualizing the calculation progress of user's tasks based on the calculation information fed back by the function template. The data search returns the data under user's request, then uses Jsmol to visualize the structure data. Task submission consists of function template selection and initial data input.

3.2. Data standardization and storage

In DCMCP, using standardization module to standardize the input files, process information, and calculation results, then store the data in the mongod of the MongoDB cluster in json format. Then, providing users with download services and visualized data through the web interface.

DCMCP uses the MongoDB sharded cluster to store massive and heterogeneous data files. For the massive and heterogeneous data generated by the calculations, MongoDB sharded cluster enables fast data retrieval and large-scale data requests. MongoDB sharded cluster, as shown in Fig. 3. Mass data is stored in sharding, and multiple shards can be retrieved simultaneously to speed up data retrieval. Separating data storage and retrieval services can increase the number of simultaneous retrieval requests and enhance data security and cluster scalability.

3.3. Remote management program

DCMCP Server and DCMCP Client are core components of the automatic processing and dispatching system. DCMCP Server combines standardized data of

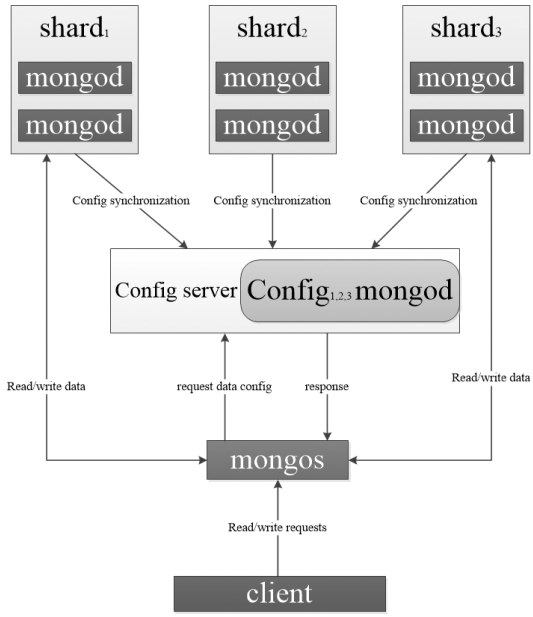

Fig. 3. Framework of MongoDB sharded cluster

calculation tasks and the function template into a completed calculation task, and calls the dispatching module to dispatch the calculation task dispersedly. Adopting the paramiko module of python to establish the network connection between the DCMCP server and the DCMCP client. Paramiko module follows the SSH2 protocol and supports remote connection to the cluster in an encrypted and authenti-cated manner. The DCMCP Client is responsible for submitting tasks, monitoring tasks and resources on the computing cluster. The DCMCP Client submits the calculation tasks and obtains the computing resources information through the queue system (e.g. LSF, PBS, etc.). At the same time, it uses the Custodian (an error correction module of MP) to correct the errors caused by the calculation inputs.

3.4. Function template

The function template contains researchers' material calculation experience, which is Python template objects based on the implementation process of specific material calculation tasks. Based on several VASP implementation processes of material calculation tasks, several function templates have been generated, including the structure relaxation template, as shown in Fig. 4, the static calculation template, the band calculation template, etc. The standardized data of calculation tasks is the input parameter of the object template so that constituting a completed calculation task.

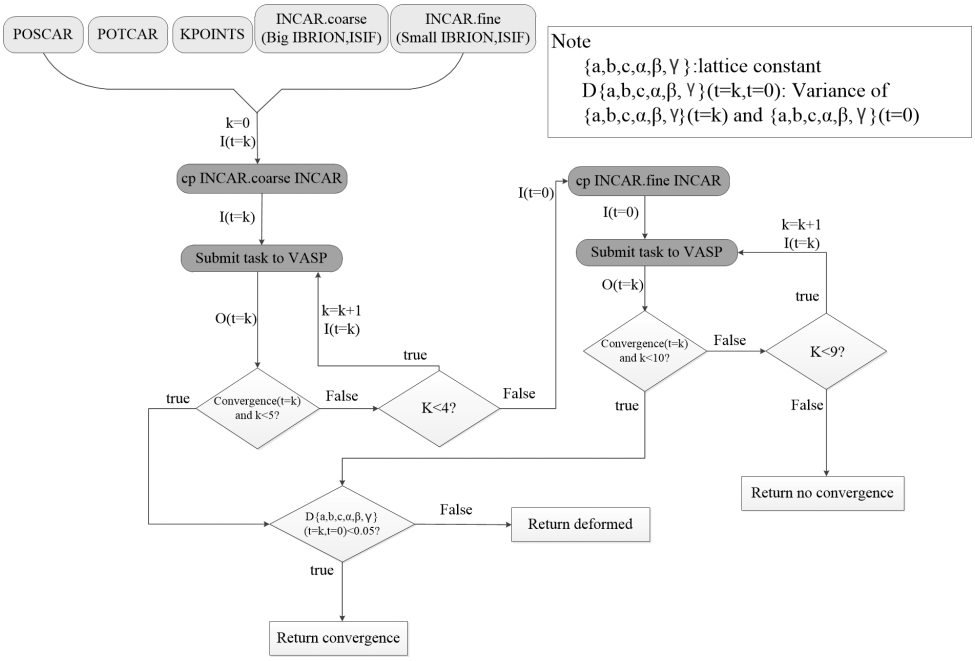

Fig. 4. Structure relaxation template

4. Conclusion

Existing material computing and data platforms, such as AFLOW and MP, both take single supercomputer as a calculation support to execute high-throughput material computing. In this design pattern, these platforms do not share computing resources and high-throughput computing code, resulting in the external users can not directly use these platforms to implement high-throughput computing, and then affecting the computing efficiency and resource utilization of platforms.

Concerning this question, in this paper we proposed a DCMCP platform which supported by the users' computing clusters. Based on function templates and web GUI, it has reduced material computing demands of computer hardware and user's computer operational difficulty, which make the material calculation process and the environment more open to users. The combination of distributed material computing and MongoDB sharded clusters not only improves computing speed, but also increases material data capacity and speeds up data retrieval. DCMCP has set up a MongoDB cluster with seven storage nodes to store 130,000 VASP calculations.

In the future work, machine learning and data mining methods would be applied to train classification models [9]–[11] so that discover the rules that may exist in the material data. Designing a calculation program of dopant material, and the program can find the most suitable structure data under the calculated results of all doping structure. Furthermore, DCMCP will be compatible with more calculation software to broaden the research area.

References

[1] A. Jain, S. Ong, G. Hautier, W. Chen, W. Richards, S. Dacek, S. Cholia, D. Gunter, D. Skinner, G. Ceder, K. Persson: Commentary: The Materials Project: A materials genome approach to accelerating materials innovation. Apl Materials 1 (2013), No. 1, paper 011002.

[2] S. Curtarolo, W. Setyawan, G. Hart, M. Jahnatek, R. Chepulskii, R. Taylor, S. Wang, J. Xue, K. Yang, O. Levy, M. Mehl: AFLOW: an automatic framework for high-throughput materials discovery. Computational Materials Science. 58 (2012), 218-226.

[3] W. Setyawan, S. Curtarolo: High-throughput electronic band structure calculations: Challenges and tools. Computational Materials Science. 49 (2010), No.2, 299-312.

[4] S. Curtarolo, G. Hart, M. Nardelli, N. Mingo, S. Sanvito, O. Levy: The high-throughput highway to computational materials design. Nature materials. 12 (2013), No.3, 191-201.

[5] J. Saal, S. Kirklin, M. Aykol, B. Meredig, C. Wolverton: Materials design and discovery with high-throughput density functional theory: the open quantum materials database (OQMD). Jom. 65(2013), No.11, 1501-1509.

[6] A. Dimakis, P. Godfrey, Y. Wu, M. Wainwright, K. Ramchandran: Network coding for distributed storage systems. IEEE Transactions on Information Theory. 56 (2010), No.9, 4539-4551.

[7] S. Ong, W. Richards, A. Jain, G. Hautier, M. Kocher, S. Cholia, D. Gunter, V. Chevrier, K. Persson, G. Ceder: Python Materials Genomics (pymatgen): A robust, open-source python library for materials analysis. Computational Materials Science. 68 (2013), No.2, 314-319.

[8] L. Ghiringhelli, C. Carbogno, S. Levchenko, F. Mohamed, G. Huhs, M. Lüders, M. Oliveira, M. Scheffler: Towards a Common Format for Computational Material Science Data. arXiv preprint arXiv: 1607.04738 (2016).

[9] P. Raccuglia, K. Elbert, P. Adler, C. Falk, M. Wenny, A. Mollo, M. Zeller, S. Friedler, J. Schrier, A. Norquist: Machine-learning-assisted materials discovery using failed experiments. Nature. 533 (2016), No.7601, 73-76.

[10] H. Bhadeshia, R. Dimitriu, S. Forsik, J. Pak, J. Ryu: Performance of neural networks in materials science. Materials Science and Technology. 25 (2009), No.4, 504-510.

[11] S. Curtarolo, D. Morgan, K. Persson, J. Rodgers, G. Ceder: Predicting crystal structures with data mining of quantum calculations. Physical review letters 91 (2003), No.13, paper 135503.

Received October 31, 2017