SIN-CHN
SCIENTIFIC PRESS PTE.LTD.

Molecular & Cellular Biomechanics 2024, 21(3), 658.
https://doi.org/10.62617/mcb658

Article

Full-process supported Simulation Platform Framework based on cloud computing and HPC integration

Hao Wang[1], Jinghua Feng[2,*], Lin Wang[3]

[1] Tianjin Maritime College, Tianjin 300350, China
[2] National Supercomputer Center in Tianjin, Tianjin 300457, China
[3] Yanshan university in Qinhuangdao, Hebei 066004, China
*** Corresponding author:** Jinghua Feng, fengjh21@nscc-tj.cn

CITATION

Wang H, Feng J, Wang L. Full-process supported Simulation Platform Framework based on cloud computing and HPC integration. Molecular & Cellular Biomechanics. 2024; 21(3): 658. https://doi.org/10.62617/mcb658

ARTICLE INFO

Received: 29 October 2024
Accepted: 7 November 2024
Available online: 13 November 2024

Abstract: Simulation technology is widely used in many fields, it usually involves three processes, (i) pre-process, (ii) analysis solver, and (iii) post-process. Simulation calculations require a large amount of computing resources, and users usually need to use cloud and High-Performance Computing (HPC) systems to complete works. Simulation works are increasingly depending on the capacity of HPC or cloud, for cost reasons, people are more willing to use the services than self-built an HPC or Cloud computing cluster. However, that leads to the isolation of calculations and pre- and post-processing work, adding additional time for data transfer. Moreover, simulation engineers also want to use cloud servers in pre-processing and post-processing, since compared with local workstations, cloud servers have significant advantages in saving hardware investment, remote office collaboration, and data integration management. Therefore, we provide a platform framework based on cloud computing and HPC integration that supports the full process of simulation. Then, we implemented it in Tianhe-1A and Tianhe exascale supercomputers and THCloud environments. Through a city area-level explosion simulation experiment, it was verified that the framework can fully support the whole process of simulation, and effectively reduce the time of simulation work, improving the simulation engineer's work efficiency. The study shows that the platform provides a feasible solution for full-process simulation. Compared with other platforms, it has the characteristics of full-process, high performance and high security.

Keywords: simulation; HPC; cloud computing; full-process; integration

1. Introduction

At present, simulation technology has been widely used in the design and production of industrial products [1]. In general, simulation technology includes Finite Element Analysis (FEA), Computational Fluid Dynamics (CFD), Multibody Dynamics (MBD), and Durability for engineering or product performance analysis [2–5]. Generally, simulation analysis involves pre-processing, Analysis solver and post-processing. Pre-processing mainly includes the establishment of simulation physical model, such as geometric model, grid division, and the addition of physical properties and boundary conditions. Numerical solution is the core of Analysis solver, which includes structure analysis, fluid dynamics analysis, electromagnetic field analysis, sound field analysis, piezoelectric analysis, and coupling analysis of multiple physical fields. Post-processing mainly realizes the interpretation and evaluation of the analysis results, such as displaying the computation results in the form of color cloud maps, vector maps, particle flow traces, sections and other graphics, or outputting them in the form of text and charts. With the development of computer technology and simulation software, the analysis solver of simulation has been developed from using a single workstation to supporting large-scale High-Performance Computing (HPC) cluster [6,7]. The application of HPC technology has greatly shortened the time requirement of the analysis solver and achieved more working conditions [8].

More and more simulation experts are abandoning the tradition of only paying attention to parts-level simulation in the past, and including the whole system of the whole objects (such as the whole machine and the whole car) into the computation scope at one time. Single physical field analysis (such as structure mechanics and fluid dynamics) is often unable to truly simulate some physical phenomena. Considering the influence of each physical factor on the analysis object has become the most popular technology for the development of simulation. Nowadays, the optimization of a single discipline has long been unable to meet the needs of product development. For one example, the overall design of the aircraft involves aerodynamics, propulsion system, flight dynamics, electromagnetism, weight center of gravity, stealth, expense, and so on [9,10]. The simulation technology is also more and more applied to the simulation and analysis of ultra-large scenarios, such as the damaging effect of the explosion on the surrounding environment at the urban area level [11]. The surge of simulation computation brought by this series of development changes can only be solved by HPC.

In recent years, the number and computing speed of supercomputers in the world are increasing rapidly. More and more supercomputing centers are not only providing services to scientific research but also providing services to industrial enterprises [12]. The supercomputing centers have launched the pay-as-you-use mode HPC services, under which enterprise users only need to pay for the cost of CPU hours, file storage and network according to their usage [13]. This kind of service can avoid the need to build and maintain the HPC cluster by the users themselves and greatly save the cost of using HPC and also obtain larger scale computing power from the supercomputing centers [14].

Cloud computing brings many benefits to the HPC environment for simulation [15]. With the development of cloud computing, more and more commercial simulation software begins to support Licenses for Cloud Computing or use the Bring Your Own License mode. Users do not need to buy the licenses by themselves, and they only need to pay the rental fees of cloud licenses instead. In this way, enterprise users can reduce the input cost while achieving the ability of License scaling.

Despite the advancements in simulation technology, several challenges persist, particularly in the integration and utilization of HPC resources. Traditional approaches often require simulation experts to work locally with pre- and post-processing software while relying on remote HPC for the solver stage. This disconnects results in significant inefficiencies due to the need for repeated file uploads and downloads, bandwidth limitations, and the additional workload associated with software maintenance and licensing [6]. Furthermore, the integration of HPC with cloud computing presents both opportunities and complexities, as it can offer scalability and on-demand resources but also introduces latency and security concerns.The research gap addressed in this study lies in the lack of a comprehensive, full-process supported simulation platform that seamlessly integrates cloud computing and HPC resources. Current solutions, such as CloudSME, LincoSim, CloudPass, and MCX Cloud, offer varying degrees of web-based functionality and computing resource architectures but do not provide a unified, end-to-end solution for simulation workflows [7]. This limitation hinders the efficiency and productivity of simulation engineers, particularly in large-scale and complex simulations.

Therefore, in order to solve the above problems, we proposed a platform framework named Full-process supported Simulation Platform Framework (FSPF) based on cloud computing and HPC integration. The contribution of this study is to provide a comprehensive solution that integrates the strengths of both cloud computing and high-performance computing (HPC) to enhance the efficiency and productivity of simulation workflows.

This paper is organized as follows. First, the literature review section reviews the current status of the application of cloud computing and high-performance computing (HPC) in simulation workflow, as well as their respective advantages and limitations. Then, the paper elaborates the process and methodology of simulation, which lays the foundation for the FSPF platform framework proposed in this bit. Then, the article will detail the concept and architecture of FSPF, including the design and functions of user interface, application layer, middle layer and resource layer. Finally, the researcher demonstrates the practical effects and advantages of the FSPF platform framework by comparing it with other platforms and analyzing the experiments.

2. Literature review

Simulation systems can be generally divided into two types: discrete systems and continuous systems [16]. Discrete system simulations are typically either discrete-event or agent-based. Discrete-event is mainly used to study complex queuing systems in equipment manufacturing, automobile engineering, and supply chains while agent-based is used to social media, traffic, infection spread, etc. Continuous system simulations are used to analyze flow-related problems such as naval architecture and ocean engineering, oil prospecting, aircraft design and manufacturing, etc.

There are numerous different approaches to developing simulation systems ranging from “local computing cluster” developments, and cloud-based developments to comprehensive workflow systems. Ieshkin et al. [17] described a flow visualization system using glow discharge with annular or plane electrodes and performed numerical simulation as well. Jorge et al. [18] proposed an architecture layout for a real-time power system simulation based on a distributed cluster of IBM PC clusters. Urban Borštnik et al. [19] presented the design and implementation of the Force Decomposition Machine, a cluster of personal computers that are tailored to running molecular dynamics simulations using the distributed force decomposition parallelization method.The cases above are the development approaches based on local computing clusters. Although advanced and convenient, significant barriers exist to the widespread adoption of these tools. In particular, “local computing cluster” development: 1. is considered complex to share work; 2. needs in-house expertise; 3. requires high capital costs; and 4. is difficult to carry out large-scale simulation operations with the increasing amount of raw data [20].

Guzzetti et al. [21] investigated the impact of different types of numerous simulation platforms. They compared in-house computing clusters, a large-scale university-based high-performance computing (HPC), and a regional supercomputer with clouds (Penguin's On-Demand HPC Cloud Service and Amazon's EC2). They showed that clouds may be easy to use and utilized for scientific simulation possibly at lower cost and better performance than using a more expensive local computing cluster.

An example of a cloud-based simulation is the distributed agent-based traffic simulator called Megaffic, that allows adaptive resource provisioning to speed up the execution of a single simulation run. Hanai et al. [22] used Bulk Synchronous Processing to process and synchronize between elements of a simulation assigned to distributed processors. At each checkpoint, a decision was made to add or remove processors and to re-balance the processing load. The simulator was implemented using the Google Compute Engine cloud. Another example of a cloud-based simulation is GridSpice, a scalable open-source simulation framework for modeling, designing and planning of the smart grid. Kyle Anderson et al. [23] seamlessly integrated existing electric power simulation tools with a Representational State Transfer (REST) application programming interface, which allowed the cloud-based architecture simulation platform easy to use.

Besides, there are many advantages to using clouds for workflows, including on-demand resource provisioning; the ability to allocate resources that match the workflow needs; and the ability to use custom software configurations that support applications with complex dependencies. What's more, users can easily share their computing environments with colleagues and reviewers, thus increasing collaboration of scientific results [24]. CloudSME [16] simulation Platform is an architecture that uses a cloud broker platform to enable the capabilities of a multi cloud environment and combines with workflow development. LincoSim [25] is a web based HPC-cloud platform for automatic virtual towing tank analysis. CloudPSS [26] is a cloud-computing based power system simulator. Based on an open service integrating framework, a self-developed electromagnetic transients (EMT) simulator with an automatic code generator is provided to accelerate EMT simulations using heterogeneous computing devices in the cloud, such as CPU and GPU. MCX Cloud [27] is a configuration-free, in-browser 3D MC simulation platform. Monte Carlo eXtreme (MCX) Cloud—built upon an array of robust and modern technologies, including a Docker Swarm-based cloud-computing backend and a web-based graphical user interface (GUI) that supports in-browser 3D visualization, asynchronous data communication, and automatic data validation via JavaScript Object Notation (JSON) schemas.

Wolstencroft et al. [28] designed the Taverna workflow tool suite to combine distributed Web Services and/or local tools into complex analysis pipelines. These pipelines can be executed on local desktop machines or through larger infrastructures such as supercomputers or cloud environments. There are also some researches on running simulation computing tasks based on cloud computing platforms [28,29], and using resources integrated with HPC and cloud computing to build simulation platforms [30,31].

In the next section, we will propose the theoretical basis to create a full-process supported simulation platform framework, that integrates cloud platform and high-performance computing (HPC), which means it has the advantages of cloud computing and super-computing. Tianhe-1A, the fastest computer in the world from October 2010 to June 2011, is one of the few peta-scale supercomputers in the world. Tianhe exascale supercomputer has a theoretical peak performance of 3.15 Pflops. By utilizing Tianhe-1A and Tianhe exascale supercomputers, this platform is generic as it supports a wide range of simulation applications while keeping high-level efficiency, stability, security and good user experience. Users can reduce expense and time spent significantly and reduce error level at the same time [32].

3. Theoretical foundation

Taking structural calculation as an example, it usually includes three processes: pre-processing, Analysis solver, and post-processing. In the pre-processing, geometric modeling and detail processing are first carried out to establish the geometric structure involved in the analysis purpose and minimize the structural intervention according to the analysis focus. Then, use the grid generation tool to generate the grid. Then, simplify the geometric structure into elements such as mass point, shell, membrane, beam, truss or entity according to the analysis purpose. Next, give the parameters such as density, elastic modulus, Poisson ratio, yield strength, fracture strength, and fracture strain. Finally, set calculation conditions including analysis step, interaction, load and constraint conditions.

When finishing the work above, enterprise users can submit the jobs. After completing the structure calculation, they can extract the computation results they need such as energy curve, displacement curve, velocity curve, acceleration curve, etc. The whole structure calculation progress is shown in **Figure 1**.

The earliest simulation computing software was deployed on one single server, and its performance could not be extended, making the simulation computing time very long or even impossible to calculate. The application of HPC technology has greatly shortened the time required for simulation solutions or completed more working conditions and more accurate solutions within the specified time. By using HPC technology, enterprise users can significantly shorten the research and development cycle, improve the product performance index, ensure product quality, reduce the number of experiments, and reduce the research and development cost. In addition, the surge of simulation computation caused by the bigger simulation scale and higher degree of refinement can only be solved by HPC.

1) The essence of simulation computation is to solve linear equations and the utilization of HPC generally has two objectives:
2) One goal is to solve a larger scale model, so that the numerical computation model is closer to the real one, and the computation will be more accurate. Different emulator solvers have different requirements for computer hardware resources, so the maximum size of the solution is also different. For instance, the largest model at present reported by CFD software is the American Cup sailboat model computed by FLUENT, with a solution of 1 billion units. FLUENT is generally a CFD software used to model fluid flow, chemical reaction, heat & mass transfer, etc. It is also a user-friendly interface enabling streamlining of the CFD pre- to post-processing workflow within a single window.

Another goal is to shorten the time of solving one case of the same scale, or to compute more times in unit time. For example, FLUENT provides a standard test example to evaluate the performance of HPC system, which is measured by the number of computing times per day.

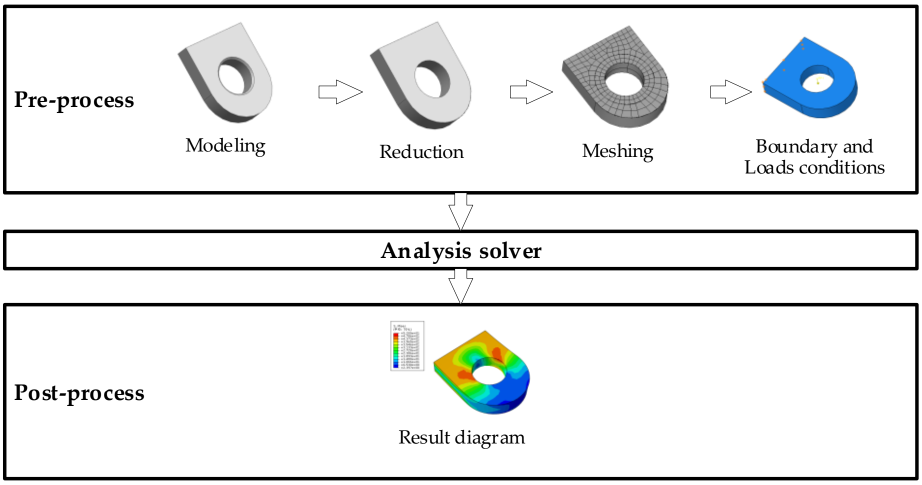


Figure 1. The whole process of a structural simulation.

When simulation engineers need to use HPC for simulation computation, the best choice is to use the supercomputer provided by the supercomputing centers. As mentioned in section 1, more and more supercomputing centers are also focusing on applications in the industrial field. Typically, using a supercomputer requires three steps as shown in **Figure 2**. Firstly, one needs to log in to the VPN server to build the VPN Tunnel. Next, open the SHELL Client that is installed locally. Then, use the CLI provided by the supercomputer in the Terminal. Such a way is not friendly enough because it requires users to have certain computer knowledge and skills, otherwise, they will not be able to use supercomputers smoothly.

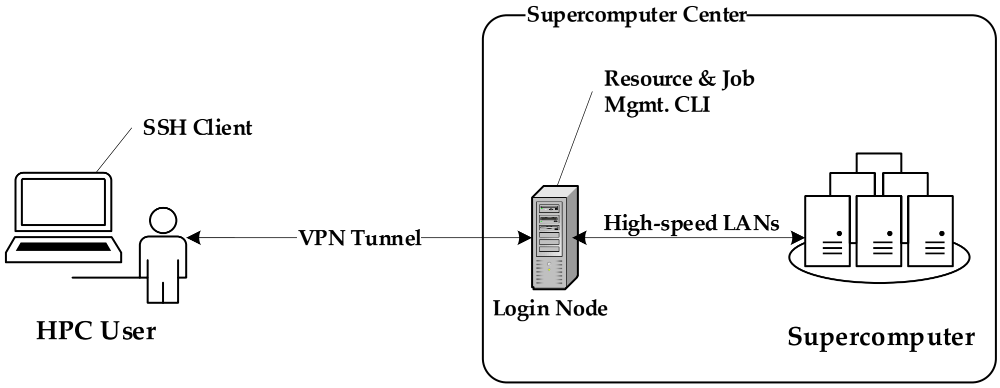


Figure 2. Usual way of using Supercomputer.

Under the usage mode of the supercomputer mentioned in **Figure 2**, the whole process of the simulation work completed by the simulation engineer is shown in **Figure 3**. The Pre-processing data files need to be uploaded to remote HPC, and the computed result data files need to be downloaded to the local for post-processing. The file transfer on the network brings a lot of unnecessary waste of time. In addition, local pre- and post-processing needs to invest in high-performance workstations, software licenses and other costs, which many medium-sized enterprises (SMEs) may not be able to afford.

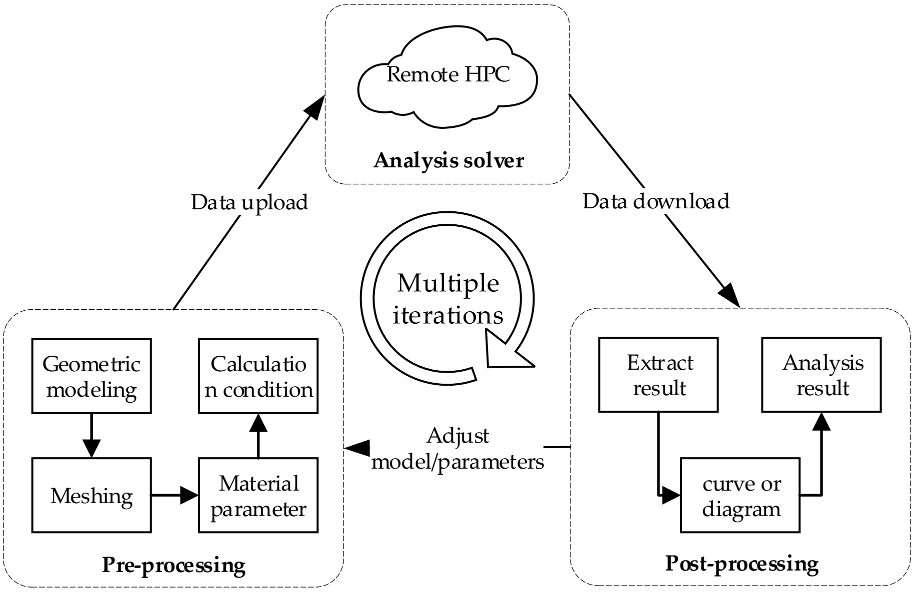


Figure 3. The workflow of using remote HPC only for solver computing.

The aforementioned methodologies and procedures have laid a solid foundation for the development of our FSPF framework.

Hence, the pressing challenge now lies in how to actualize the seamless integration of the simulation process within the cloud environment, and how to harness the immense computing power of supercomputers to offer simulation engineers on-demand simulation services.

Addressing these considerations, we introduce the Full-process supported Simulation Platform Framework (FSPF), which integrates cloud computing and HPC. The specifics of the FSPF are elaborated in the following section.

4. Concept and architecture of the FSPF

The generic layered architecture of the FSPF is shown in **Figure 4**. The platform adapts and encapsulates the super computing and RemoteApp services. It uses message queue technology to isolate the underlying resource management system from the upper application, and provides users with an integrated graphical operation interface through the web portal. Users do not need to install any application programs. They can complete the whole process of Pre-processing, Analysis solver, and post-processing by using a web browser. Meanwhile, the configuration requirements of the platform for local PCs are very low, and the use of thin clients can meet the requirements. Therefore, the platform can provide integrated SaaS services for the cloud simulation process, reduce local hardware and software procurement and management investment, thus significantly shortening the research and development (R&D) cycle, improve product performance indicators, ensure product quality and reduce R&D costs.

The FSPF consists of four layers as **Figure 4** shown:

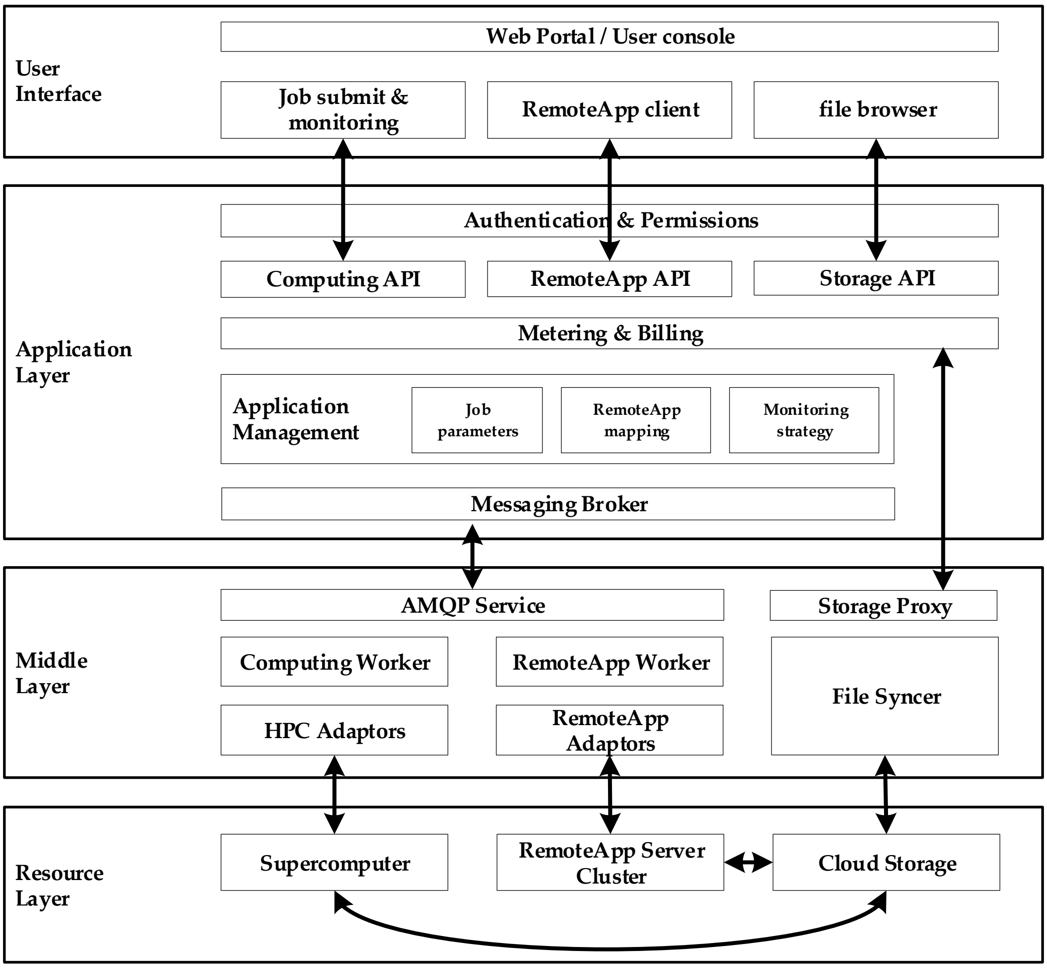


Figure 4. The architecture of FSPF.

User Interface Layer that presents graphical interfaces as Software-as-a-Service (SaaS) in a wide range of scenarios. This layer provides convenience and easy-using to end users.

1) Application Layer that provides encapsulated API services for Authentication, Billing and Application management. This layer can pass the API requests to messaging queue services.
2) Middle layer that realizes the transmission of operation orders and data files. It isolates the Application Layer and Resource Layer logically to ensure the security access of the underlying system.
3) Resource Layer that provides underlying compute, storage and visualization nodes as an Infrastructure-as-a-Service (IaaS). It offers access to the supercomputer, remote application server cluster and cloud storage resources. Among them, cloud storage and supercomputer have encrypted data transmission channel connections. These layers are presented in detail in **Figure 4**.
4) The combine of cloud platform and HPC system mainly includes data integration and resource integration. Data integration is achieved by establishing an encrypted channel between cloud platform and HPC system. Storage Proxy establishes encrypted transmission channel between Cloud Storage and Supercomputer internal storage. The details are described in 4.3. Middle Layer.

Resource integration is based on system resource scheduling system such as slurm, which implements call encapsulation to schedule cloud platform and HPC resources. The details are described in 4.4. Resource Layer.

4.1. User interface

The User Interface provides end users with graphical interfaces. We recommend establishing the User Interface based on the web, because web applications are easy to use and develop. The User Console provided by the User Interface includes:

1) Submission and monitoring of the simulated job. Support different simulation job software, graphical parameter setting interface, one-click submission job function, and real-time monitoring of running job status.
2) RemoteApp client supports starting and using simulation Pre-processing and Post-processing applications.
3) File browser supports visiting file list, file uploading, file downloading and file management.

4.2. Application layer

The Application Layer provides Computing API, RemoteApp API, and Storage API for User Interface Layer. Before accessing the resources, users' authentication is needed and their permissions are detected. Anonymous access is not supported. When users are using the underlying resources, Metering and Billing are managed in real-time to generate users' resource use bills.

The Application Layer also provides an application management module for adding and maintaining simulation applications, including the following information:

1) Job submission parameters of the simulation computing applications, such as the number of the input parameters, the type of each parameter, and input constraints. When adding simulation computing applications, these contents should be filled in to generate the interface for job submission.
2) Monitoring strategies of simulation computing jobs, such as setting the refresh frequency of job status, and synchronizing the result files in real-time, etc.
3) RemoteApp mapping of Pre-processing and Post-processing. Establishing the mapping between the RemoteApp interface and services so that users can start the application by clicking on the application icon in the graphical interface. API requests are delivered to the Advanced Message Queuing Protocol (AMQP) Service of the Middle Layer through Message Broker.

4.3. Middle layer

Middle Layer provides AMQP Services and Storage Proxy to realize the transmission of job operation instructions and data. It logically isolates the Application Layer and the Resource Layer, and ensures the security of the underlying system. The Computing Worker is responsible for monitoring and processing job operation instructions, such as job submission, job suspension, job canceling, etc. It also verifies and converts job-related operation instructions, then send them to the HPC Adaptors for execution. HPC Adaptors adapt supercomputers with different scheduling systems such as Slurm, PBS, etc., and convert the operation instructions to scripts for execution as shown in **Figure 5**.

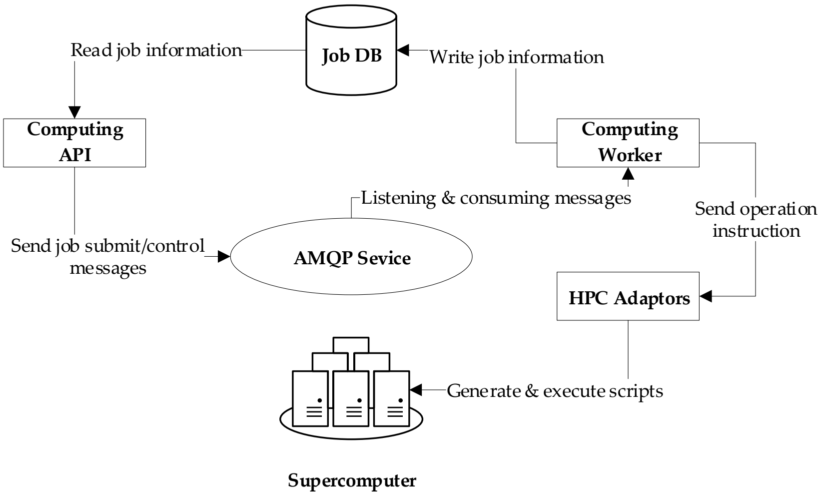


Figure 5. Operations of job in the Middle Layer.

The RemoteApp Worker is responsible for monitoring and handling RemoteApp related operation instructions, such as start, add, remove, etc., and sends them to the RemoteApp Adaptors to execute after verification and conversion. The RemoteApp Adaptors adapt different RemoteApp server clusters, such as Citrix DaaS, VMware Horizon, and convert the operation instructions to script for execution. Its operation procedure is similar to job operations.

Storage Proxy establishes encrypted transmission channel between Cloud Storage and Supercomputer internal storage. The operation is only allowed to be initiated by the FileSync application, which is used to synchronize the input and output files of a compute job. Users' data storage in FSPF is centered on Cloud Storage. The RemoteApp can store the Pre-processing files into the Cloud Storage, or read files from Cloud Storage for Post-processing. File Sync application can automatically synchronize the input file from Cloud Storage to Supercomputer when submitting the computation job. After completing the computation job, File Sync application will automatically synchronize the output file to Cloud Storage. The whole process is shown in **Figure 6**.

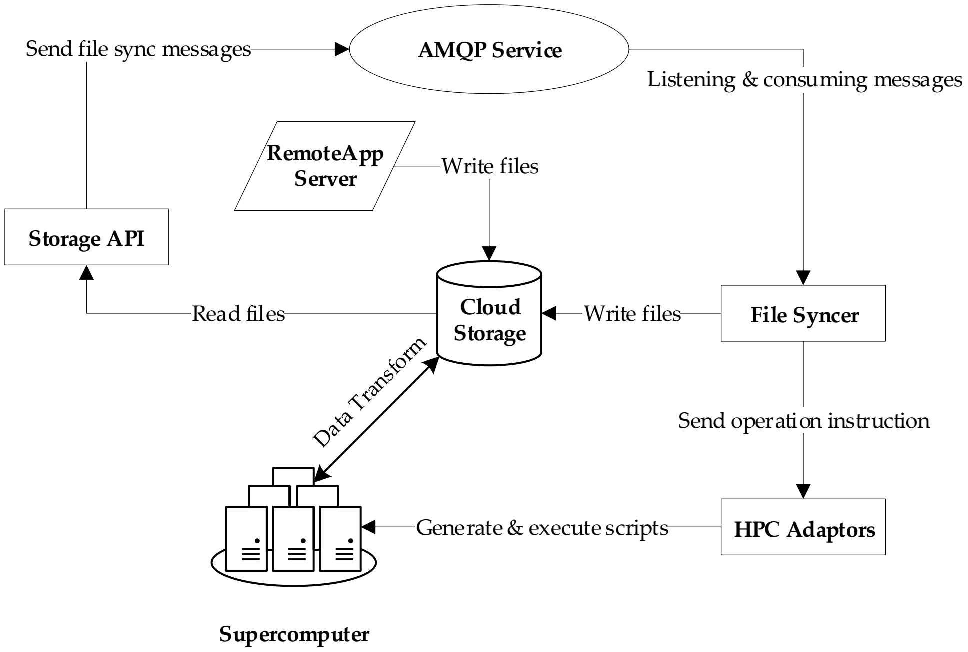


Figure 6. File processing in the Middle Layer.

4.4. Resource layer

The Resource Layer provides access to the Supercomputer, RemoteApp server cluster and Cloud Storage of the FSPF. Among them, there is an encrypted data transmission channel connection between the Cloud Storage and Supercomputer for synchronizing input and output data files. Generally, Supercomputer needs a set of resource management system to realize computing resource and job scheduling management. Slurm [33] is one of the common open-source software and there is many other software in current. The FSPF supports many kinds of HPC resource management software through the HPC Adaptors so as to support different supercomputer systems.

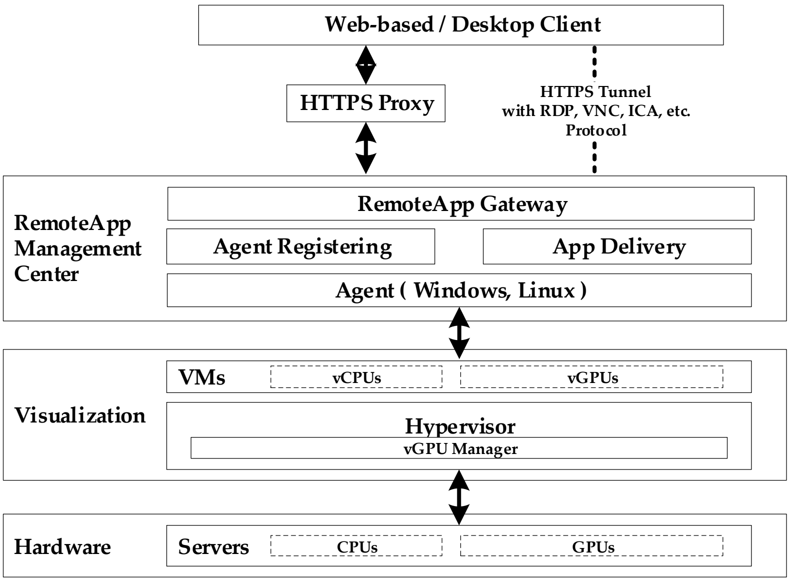


Figure 7. Architecture of RemoteApp service.

To provide RemoteApp service works. Firstly, we need a GPU-equipped server (to improve the efficiency of model rendering and to process larger and more complex models). Next, in addition to conventional CPU, memory, and other virtualization, GPU virtualization (such as NVIDIA, GRID and VGPU technology) is required to create VMs that include VGPU. GPU virtualization technology can significantly improve model rendering efficiency and enhance support for more complex models by allowing multiple virtual machines or users to run simultaneously on a shared physical GPU. This technology not only optimizes the performance of graphics-intensive applications, but also enables high-performance computation and graphics processing in resource-constrained environments, thus driving technological advances in large-scale, highly complex simulation efforts. Then install the Agent on such VMs to register with the RemoteApp Management Center and publish the installed applications. Finally, the client connects to the RemoteApp Gateway through the HTTPS Tunnel and hosts transport protocols such as RDP, VNC, ICA, etc., to realize the display and operation of RemoteApp. The architecture of the RemoteApp service is shown in **Figure 7**.

5. Discussion

5.1. TSCP: The implementation of the FSPF

We developed the FSPF and implemented the TSCP based on the Tianhe-1A and Tianhe exascale supercomputers and THCloud. HPC Adaptor adapts Slurm, the scheduling systems of Tianhe-1A and Tianhe exscale, and implements a series of execution scripts to support the execution of various operations. RemoteApp Adaptor adapts Citrix XenApp [31] installed in the TH Cloud, and realizes the addition, deletion, launch and other operations of RemoteApp. Cloud Storage uses the Shared File System services provided by the TH Cloud.

The implementation architecture of the TSCP is shown in **Figure 8**. Users connect to the TSCP web service via WLAN/HTTPS, and use the various functions of the platform in the web portal after authentication and access permission. The web portal calls the resources such as back-end compute nodes, storage nodes, and RemoteApp services through the RESTful application interfaces. Cloud storage is used to store users' data files, share file access with RemoteApp services, and synchronize files with supercomputers.

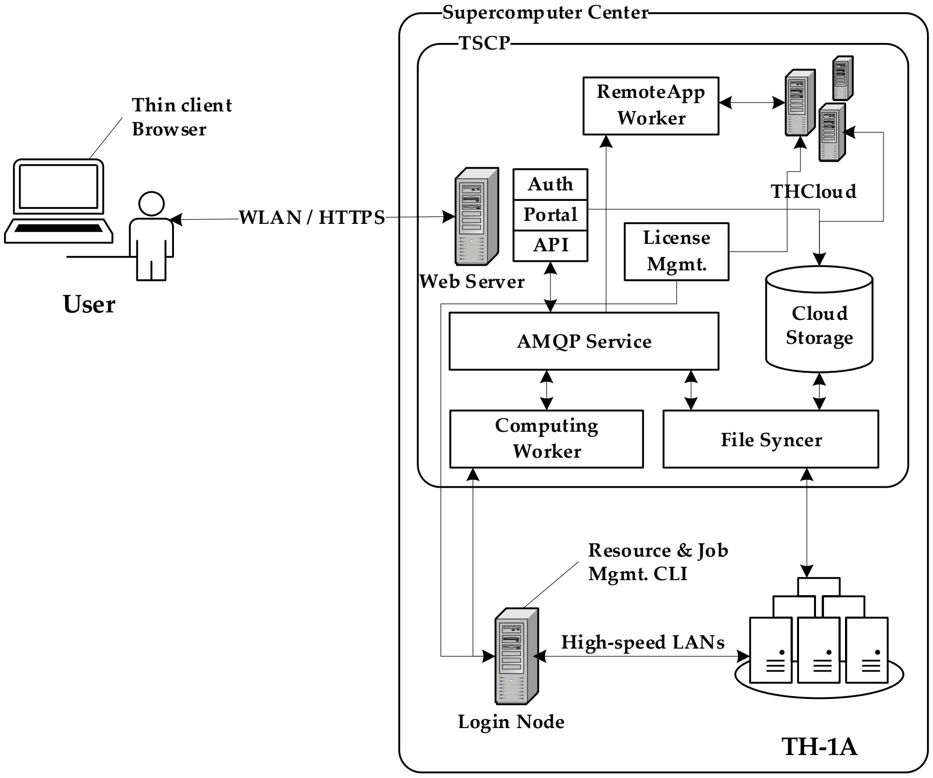
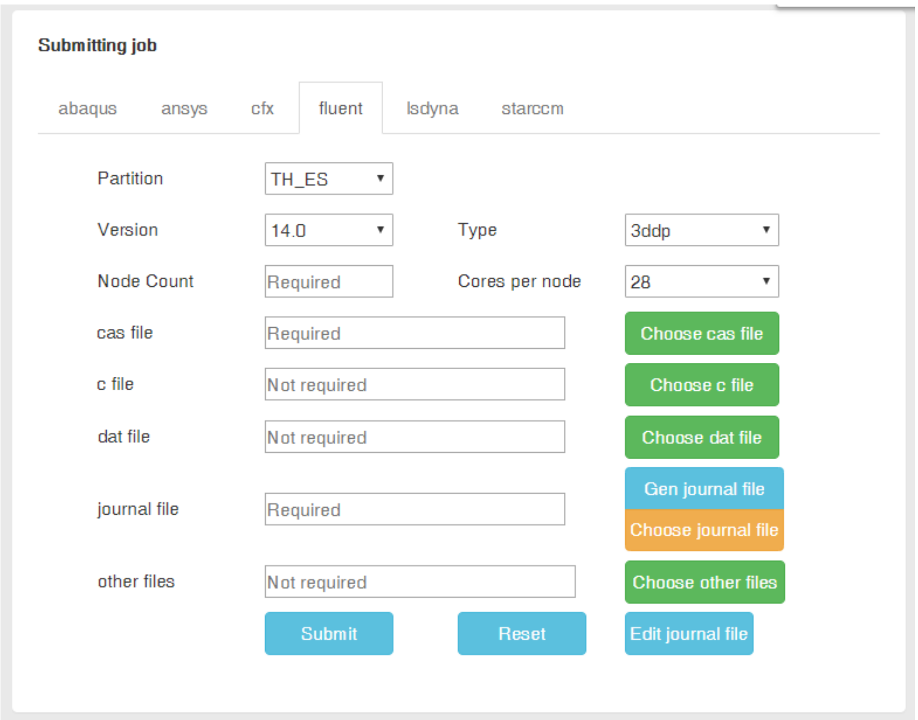


Figure 8. The implementation architecture of the TSCP.

The implementation architecture of the TSCP is shown in **Figure 8**. Users connect to the TSCP web service via WLAN/HTTPS, and use the various functions of the platform in the web portal after authentication and access permission. The web portal calls the resources such as back-end compute nodes, storage nodes, and RemoteApp services through the RESTful application interfaces. Cloud storage is used to store users' data files, share file access with RemoteApp services, and synchronize files with supercomputers.

We used the Django framework [34] to implement the Wed portal of the TSCP. The User Interface Layer provides web-based graphical interfaces, including:

1) **Figure 9** and **Figure 10** shows the job simulation of Submission. Also provides job submission interface and job status monitoring list of multiple simulation software.



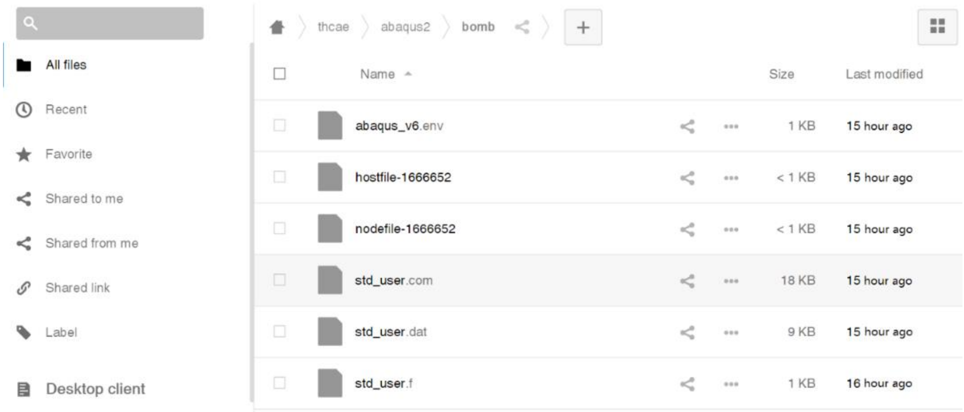
Submitting job

Partition: TH_ES
Version: 14.0
Type: 3ddp
Node Count: Required
Cores per node: 28
cas file: Required
c file: Not required
dat file: Not required
journal file: Required
other files: Not required

Figure 9. Simulation job submission interface.

2) Web-based RemoteApp client is shown in **Figure 10**, which provides a diverse set of remote simulation software applications that can be opened and used through the web browsers.

Figure 10. RemoteApp interface.



☐	Name	Size	Last modified
☐	abaqus_v6.env	1 KB	15 hour ago
☐	hostfile-1666652	< 1 KB	15 hour ago
☐	nodefile-1666652	< 1 KB	15 hour ago
☐	std_user.com	18 KB	15 hour ago
☐	std_user.dat	9 KB	15 hour ago
☐	std_user.f	1 KB	16 hour ago

Figure 11. File browser interface.

3) Web-based file browser is shown in **Figure 11**, users can directly access the file list, upload files, download files, manage files and make other operations through the web browser.

5.2. User experience of TSCP

As explained in Section 5.1, the TSCP provided a web-based graphic user interface which is very easy to get started without many computer skills required. It allows engineers to focus more on the simulation work itself, without paying attention to how to deploy and manage software, VPN networks, computer systems, etc. Thereby providing a good user experience.

After registered as a user of the TSCP, the permissions and resources could be applying for use. The operation flow of TSCP for the simulation work is shown in **Figure 12**, which includes typical operations in the whole process of simulation works.

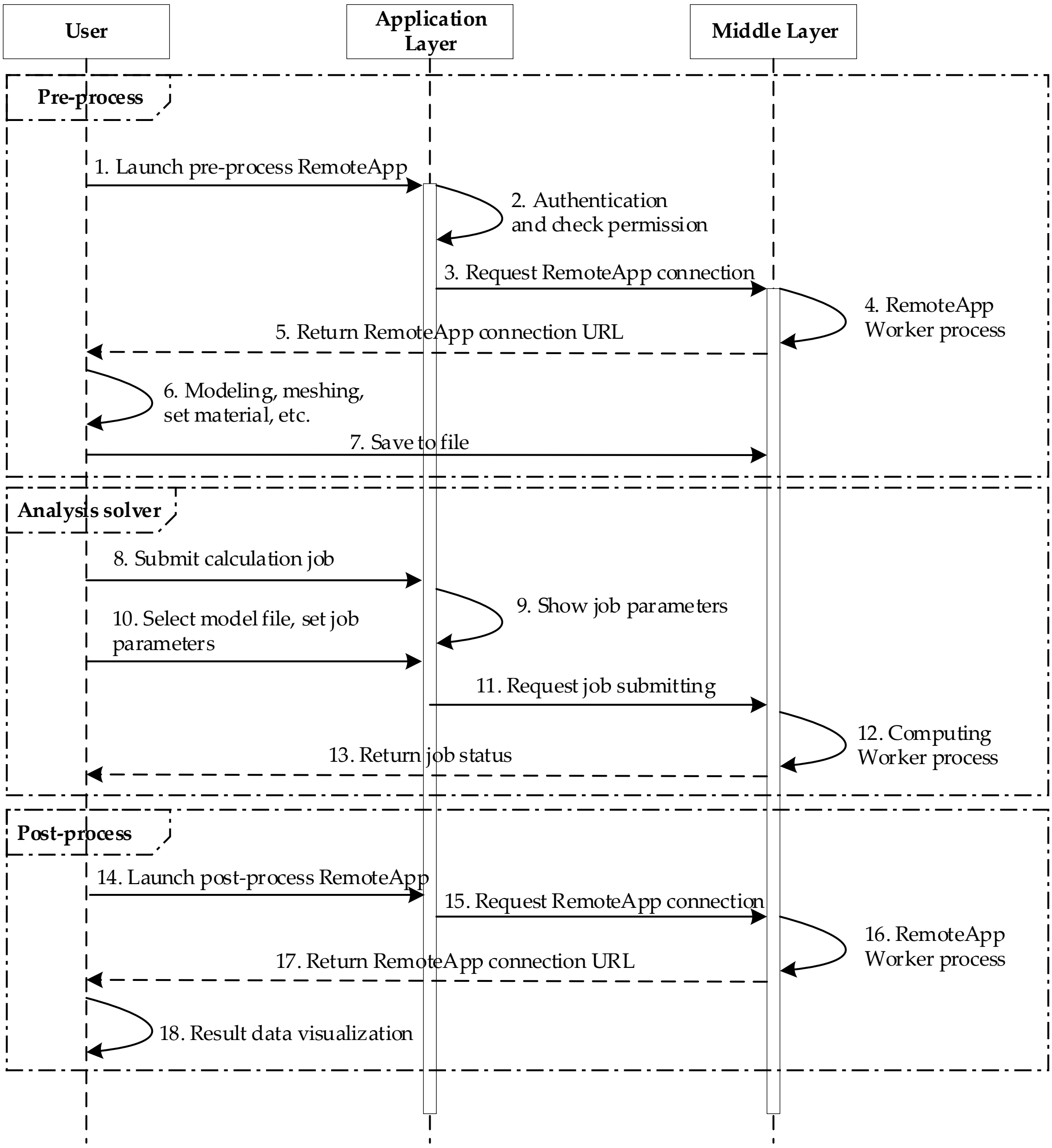


Figure 12. The user operation procedure of using TSCP.

Users could work as follows:

1) Pre-process: Enter the RemoteApp Center, click to start the Pre-processing application and after a series of back-end processing (shown in **Figure 13** specifically), then enter the interface of the application. One can do modeling, meshing and other work as usual locally. When completed, the file will be saved to the specified cloud storage path.
2) Analysis solver: Enter the job submission interface, select the simulation solver, set the input files and parameters, click the submit button, and then submit the job to the supercomputer for computation. The operation process of Computing Worker has been evaluated in Section 4, which will not be repeated here. One can see the operation status of the job through the job monitoring interface to know whether the job is completed. During the operation of the job, the intermediate files can also be manually synchronized to see the computation and solution.
3) Post-process: The operation process is similar to the Pre-processing.

5.3. Comparison with other platforms

A comparison of other similar platforms is shown in **Table 1**, which shows that the platform architecture proposed in this article （TSCP） can integrate HPC and cloud platform resources, support full-process simulation, and has better scalability.

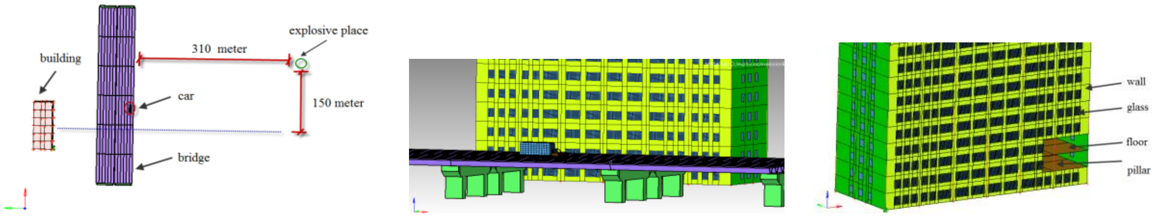
Table 1. Comparison with other platforms.

Features	**Sub Features**	**CloudSME[26]**	**LincoSim[27]**	**CloudPass[28]**	**MCX Cloud[29]**	**TSCP**
Usage	Web-based	✓	✓	✓	✓	✓
Computing resource architecture	Cloud	✓		✓	✓	✓
	HPC		✓			✓
	GPU			✓	✓	✓
Application areas	General	✓		✓		✓
	Special		✓		✓	
Simulation process	Pre-process	✓	✓			✓
	Analysis solver	✓	✓	✓	✓	✓
	Post-process	✓	✓			✓
Scalability	Experimental scale（nodes）	2	6	32	10	64
	Max nodes per Job	<100	<100	<100	<100	<2000

From the above table, it can be seen that the platform architecture (TSCP) proposed in this paper is able to integrate HPC and cloud platform resources, support full process simulation, and have better scalability. Compared with other similar platforms such as CloudSME, LincoSim, CloudPass, and MCX Cloud, TSCP performs well in terms of computational resource architecture, application domain, simulation process, and scalability. Specifically, TSCP supports Web-based usage, has cloud computing and HPC resource architectures, supports GPUs, is suitable for general and specific application domains, is able to cover the entire simulation process of preprocessing, analytical solver, and post-processing, and outperforms other platforms in terms of experimental scale and maximum number of nodes per job.

5.4. Experiment Analysis of TSCP

Taking the damaging effect of an urban area level explosion on the surrounding environment as an example, a demonstration is set to show the effect of the TSCP in improving the simulation efficiency. The air model analyzed in this paper adopts the regional range of 600*400*400m. The Euler grid is coupled with the police station, buildings, containers, bridges and automobiles through the treatment method of fluid-solid coupling to analyze the effect of explosion pressure wave on the structure, as shown in **Figure 13**.



(a) **(b)** **(c)**

Figure 13. Model assembly of urban area level explosion simulation. **(a)** Overall model assembly; (b) model assembly of car, bridge and building; **(c)** Building model details.

We carry out this simulation in three ways, including:

1) All in local using a local workstation in the whole process;
2) Local + remote HPC using a local workstation in pre- and post-process, and using remote HPC in analysis solver;
3) TSCP using the TSCP in the whole process.

Working conditions and environment settings are shown in **Table 2**.

Table 2. Environments of three working ways.

Working conditions	**Environment settings**
All in local	A workstation (configured with Intel Xeon Gold 6240 2.6G 18C/36T, Nvidia Quadro P6000, 256 GB Memory, 8 TB M.2 SSD, 1000 Mb Internet bandwidth)
Local + remote HPC	A workstation (configured same as All in local,100 Mb Internet bandwidth) Tianhe-1A and Tianhe exascale supercomputers
TSCP	A thin client (configured with Intel Celeron J3160 CPU, Intel HD 400 Graphics GPU, 4 GB Memory, 128 GB SSD Disk, 100 Mb Internet bandwidth) TSCP (deployed in THCloud)

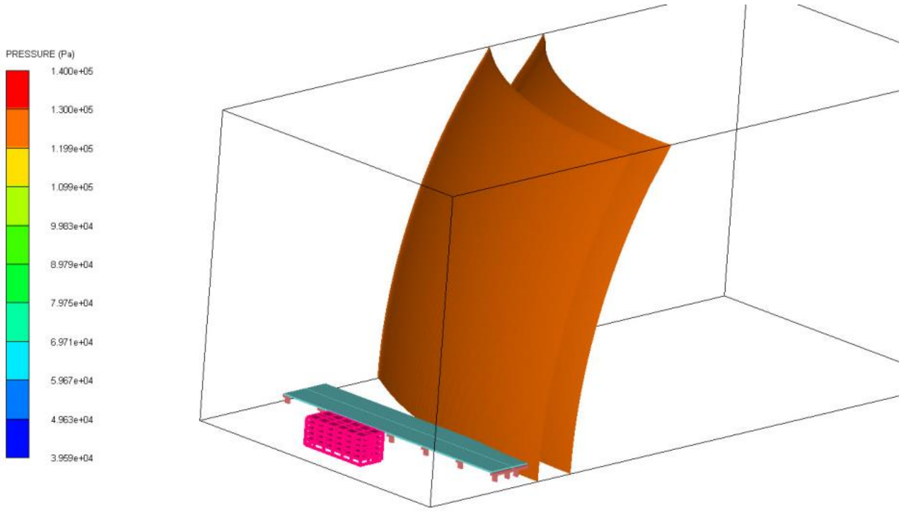
The final pre-process model file size is 1.6GB, and the result file is 400GB, and the post-process results are shown in **Figure 14**. The time spent is explained as follows:

Time spent on pre-processing includes modeling, meshing, and so on. These mainly are manual works, so the three ways take the same amount of time.

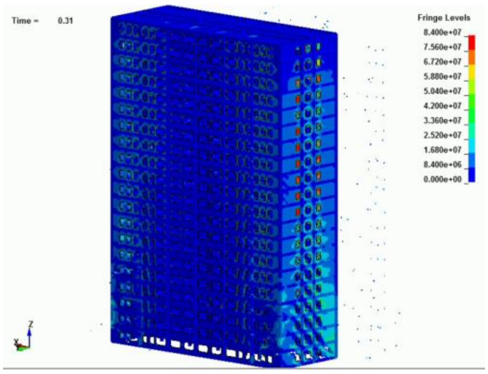
Time spent on the Analysis solver depends on computing power. The local workstation only has 36 CPU cores, so it took the longest time. This experiment used the parallel scale of 100 nodes and 2048 CPU cores of the Tianhe-1A and Tianhe exascale supercomputers that can complete the analysis solver very soon.

Time spent on post-process is similar to the pre-processing.

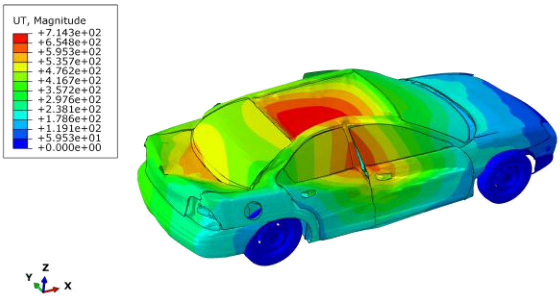
Time spent on data moving, file and data transfer, operations. The time spent on data moving mainly includes two parts: uploading the pre-process model to the HPC system (Tsend), downloading the calculation result file to the user’s local (Trecv) for result viewing. The length of time mainly depends on the network bandwidth the preprocessing model and the calculation result file size. The “All in local” and “TSCP” way is always a unified environment with graphic user interface that is user-friendly, so it won’t spend too much time. However, the “Local + remote HPC” way needs to switch between local and remote environments, login VPN, file transfer and use the command line interface (CLI) of HPC to process jobs, these may take lots of time especially for novices. Such as, the pre-proces model file size is 1.6GB and the result file is 400GB, and the theoretical time is 400 × 1024 × 8/100/3600 = 9.1 hours. In fact, the network bandwidth cannot be fully used, which means that the actual file transfer rate can only be 80% of the bandwidth.



(a)



(b)



(c)

Figure 14. Post-process results of urban area level explosion simulation. **(a)** Explosion shock wave front; **(b)** building damage effect; **(c)** vehicle damage effect.

The detail of the actual time spent is shown in **Table 3**.

Table 3. Time spent of three working ways in the experiment.

	All in local (Hours)	**Local + cloud (Hours)**	**TSCP (Hours)**
Time spent of pre-process	40	40	40
Time spent of analysis solver	1482	26	26
Time spent of post-process	3.5	3.5	3.5
Time spent on data moving	0	9.1	0
Total time spent	1525.5	78.6	69.5

According to the data in **Table 3**, the time spent using TSCP is the least one in this experiment. It saved more than 97% time against that of all in local, and 11.6% time against that of Local + Remote HPC. The TSCP can effectively shorten the time spent of simulation works, and simplify the operations, thus significantly improving the simulation engineers' working efficiency.

The platform combines cloud computing, high-performance computing, and other computing infrastructure that require a large investment, as well as relatively expensive commercial software in the industrial field. Users only need a PC connected to the Internet to carry out research and development tasks, without having to invest in computing resources and commercial software, which can greatly reduce the user's initial capital investment. And the platform is based on the "Tianhe" series of supercomputers, and can provide services for applications of different computing scales, allocate on demand, and meet the needs of users with different resource requirements.

6. Conclusions

The FSPF provides a generic full-process supported platform framework of cloud simulation by using supercomputer, RemoteApp, and cloud storage. It integrated the framework of cloud computing and supercomputer that can be adapted to common commercial environments. The TSCP which is an implementation of FSPF made it possible to provide convenient, effective, and safe simulation SaaS services. By using the TSCP, engineers can more easily to startup, and save the time spent of simulation works, thereby improving product development efficiency. Moreover, the use of local thin client and TSCP, can avoid purchasing and maintaining local high-performance workstations and permanent software licenses, thus greatly saving investment costs.

Studies have shown the FSPF is a modern simulation framework that can be more suitable for large-scale, complex system simulation works. The platform needs to have an Internet bandwidth of more than 10Mb. The platform adopts the web service model, and users mostly use drop-down menus and parameter configuration modes when using it. In subsequent work, user support documents will be further improved to reduce user usage costs. Our goal is to promote it to more users and constantly improve efficiency.

Future work considers extending the generic ability of FSPF that adapts to various open-source and commercial environments. Such as web-based modeling and auto-meshing tools, open-source simulation solvers, etc. And the concept of workflow would be added to the FSPF, to make the simulation work more automatically.

In conclusion, the FSPF platform has demonstrated its versatility and efficiency in supporting a wide range of research and development tasks. Its ability to provide scalable computing resources on demand, coupled with the cost-effectiveness of utilizing the “Tianhe” supercomputers, positions it as a valuable asset for both academia and industry. The platform’s user-friendly interface and robust middle layer architecture ensure a seamless experience for users, while the resource layer’s intelligent allocation mechanism caters to diverse computational needs. The significance of this research lies not only in the fact that it provides a powerful tool for researchers and developers, but also in the fact that it promotes the popularization and ease of use of high-performance computing resources. With the continuous advancement of technology and growing application demands, the continuous optimization and upgrading of the FSPF platform will offer the possibility of solving more complex scientific problems, as well as providing a solid foundation for technological advancement and innovation in related fields. In the future, with more users and developers contributing to the platform, FSPF is expected to become an important force in promoting scientific and technological progress.

Author contributions: conceptualization, JF and HW; methodology, JF; software, JF and LW; validation, JF, HW and LW; data curation, WH; writing—original draft preparation, JF; writing—review and editing, JF and HW; visualization, LW; project administration, HW; funding acquisition, JF. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by National Key Research and Development Program, grant number 2017YFB0202204.

Ethical approval: Not applicable.

Conflict of interest: The authors declare no conflict of interest.

References

1. De Gregoriis, D., et al. (2019). Development and validation of a fully predictive high-fidelity simulation approach for predicting coarse road dynamic tire/road rolling contact forces. Journal of Sound and Vibration, 452, 147–168.
2. Yang, Y. B., Jiang, Y. L., & Kong, Q. X. (2019). The Arrow-Hurwicz iterative finite element method for the stationary magnetohydrodynamics flow. Applied Mathematics and Computation, 356, 347–361.

3. Bertozzi, B., et al. (2019). On the interactions between airflow and ice melting in ice caves: A novel methodology based on computational fluid dynamics modeling. Science of the Total Environment, 669, 322–332.
4. Wu, H., Wu, P. B., & Li, F. S. (2019). Fatigue analysis of the gearbox housing in high-speed trains under wheel polygonization using a multibody dynamics algorithm. Engineering Failure Analysis, 100, 351–364.
5. Lin, C. K., et al. (2019). Mechanical durability of solid oxide fuel cell glass-ceramic sealant/steel interconnect joint under thermo-mechanical cycling. Renewable Energy, 138, 1205–1213.
6. Petit, S. (2015). Current challenges in simulations of HPC systems. In 2015 International Conference on High Performance Computing & Simulation (HPCS), Amsterdam, Netherlands, 20–24 July, 2015, 653–655.
7. Ahmed, K., et al. (2017). A brief history of HPC simulation and future challenges. In 2017 Winter Simulation Conference (WSC), Las Vegas, USA, 03–06 , 419–430.
8. Kumar, D., Memon, S., & Thebo, L. A. (2018). Design, Implementation & Performance Analysis of Low Cost High Performance Computing (HPC) Clusters. In 2018 12th International Conference on Signal Processing and Communication Systems (ICSPCS), Cairns, Australia, 17–19.
9. Ochôa, P., Groves, R. M., & Benedictus, R. (2019). Systematic multiparameter design methodology for an ultrasonic health monitoring system for full-scale composite aircraft primary structures. Structural Control & Health Monitoring, 2340.
10. Zarchi, M., & Attaran, B. (2019). Improve design of an active landing gear for a passenger aircraft using multi-objective optimization technology. Structural and Multidisciplinary Optimization, 59, 1813–1833.
11. Wang, K., Shi, T. T., & He, Y. R. (2019). Case analysis and CFD numerical study on gas explosion and damage processing caused by aging urban subsurface pipeline failures. Engineering Failure Analysis, 97, 201–219.
12. Fang, P. C., Jiang, Y. C., & Zhong, R. Y. (2018). Real-time monitoring of workshop status based on internet of things. In The 48th International Conference on Computers and Industrial Engineering (CIE 48), Auckland, New Zealand, 2–5.
13. Jiang, Y. C., Pang, X. L., & Li, C. S. (2018). Design and development of Tianhe render cloud platform considering the background of industrial internet. In The 48th International Conference on Computers and Industrial Engineering (CIE 48), Auckland, New Zealand, 1–8.
14. Li, R. Z., et al. (2016). A Cost-Effective Approach of Building Multi-tenant Oriented Lightweight Virtual HPC Cluster. In 2016 7th International Conference on Cloud Computing and Big Data (CCBD), Macau, China, 16–18, 219–224.
15. Mancini, M., & Aloisio, G. (2015). How advanced cloud technologies can impact and change HPC environments for simulation. In 2015 International Conference on High Performance Computing & Simulation (HPCS), Amsterdam, Netherlands, 667–668.
16. Taylor, S. J. E., et al. (2018). The CloudSME simulation platform and its applications: A generic multi-cloud platform for developing and executing commercial cloud-based simulations. Future Generation Computer Systems, 88, 524–539.
17. Ieshkin, A., et al. (2015). Computer simulation and visualization of supersonic jet for gas cluster equipment. Nuclear Instruments and Methods in Physics Research A, 795, 395–398.
18. Hollman, J. A., Martí, J. R. (2003). Real Time Network Simulation With PC-Cluster. IEEE Transactions on Power Systems, 18, 563–569.
19. Borštnik, U., et al. (2012). Implementation of the Force Decomposition Machine for Molecular Dynamics Simulations. National Institutes of Health (NIH) Public Access, 38, 243–247.
20. O'Leary, P., et al. (2015). HPCCloud: A Cloud/Web Based Simulation Environment. In Proceedings of the 2015 7th IEEE International Conference on Cloud Computing Technology and Science, Vancouver, BC, Canada, 25–33.
21. Guzzetti, S., et al. (2017). Platform and algorithm effects on computational fluid dynamics applications in life sciences. Future Generation Computer Systems, 67, 382–396.
22. Hanai, M., et al. (2014). An adaptive VM provisioning method for large-scale agent-based traffic simulations on the cloud. 2014 IEEE 6th International Conference on Cloud Computing Technology and Science. Singapore,130–137.
23. Anderson, K., Du, J., Narayan, A., El Gamal, A. (2014). GridSpice: A Distributed Simulation Platform for the Smart Grid. IEEE Transactions on Industrial Informatics, 10, 2354–2363.
24. Deelman, E., Vahi, K., et al. (2016). Pegasus in the cloud: Science automation through workflow technologies. IEEE Internet Computing, 20, 70–76.
25. Salvadore, F., Ponzini, R. (2019). LincoSim: a web based HPC-cloud platform for automatic virtual towing tank analysis. Journal of Grid Computing, 17(4), 771–795.

26. Song, Y., Chen, Y., Yu, Z., et al. (2020). CloudPSS: A high-performance power system simulator based on cloud computing. Energy Reports, 6, 1611–1618.
27. Fang, Q., Yan, S. (2022). MCX Cloud—a modern, scalable, high-performance and in-browser Monte Carlo simulation platform with cloud computing. Journal of Biomedical Optics, 27(8), 083008.
28. Wolstencroft, K., et al. (2013). The Taverna work-flow suite: designing and executing work-flows of Web Services on the desktop, web or in the cloud. Nucleic Acids Research, 41, 557–561.
29. Qi, Y., Song, W. (2023). A Cloud-based Environment for Collaborative Aircraft Sizing Supported by a Full Parametric Geometry. AIAA SCITECH 2023 Forum, 0979.
30. Deng, Z., Zhang, J., Yin, H. (2016). Architecture of cloud platform for CAE simulation in supercomputing environment. International Journal of High Performance Systems Architecture, 6(3), 131–142.
31. Alam, S. R., Gila, M., Klein, M., et al. (2023). Versatile software-defined HPC and cloud clusters on Alps supercomputer for diverse workflows. The International Journal of High Performance Computing Applications, 37, 288–305.
32. Wang, Y. B., Blache, R., Zheng, P., Xu, X. (2018). A knowledge Management System to Support Design for Additive Manufacturing Using Bayesian Networks. Journal of Mechanical Design, 140.
33. Yoo, A. B., Jette, M. A., Grondona, M. (2003). Slurm: Simple linux utility for resource management. Lecture Notes in Computer Science, 2862, 44–60.
34. Theodoropoulos, D., Pekridis, G., Miliadis, P., et al. (2023). Early Results of Mapping Industrial Applications on Heterogeneous HPC Systems: The OPTIMA Project. Proceedings of the 20th ACM International Conference on Computing Frontiers, 304–308.