

长安大学

高性能计算平台调度系统
使用手册

V1.0

目 录

第一章 高算平台介绍	4
第二章 命令行使用方式	4
2.1. 平台登录方式	4
2.1.1 通过 ssh 登录	4
2.1.2 通过 web 门户登录	4
2.2. 数据上传方式	5
2.3. 景行资源管理与调度软件简介	5
第三章 调度命令行使用方式	8
3.1 加载指定工具	11
3.2 提交通用作业	12
3.3 提交 R 并行作业	13
3.4 提交 mpi 并行作业	13
3.5 提交 tensorflow 作业	13
3.5 提交 gpu 通用作业	14
第四章 门户应用使用方式	14
4.1 Fluent 计算任务提交	14
4.2 fluent 交互模式计算任务提交	19
4.3 Matlab 集群配置	22
4.4 仿真设计应用方式	25
第五章 门户其他功能	25
3.1 文件传输工具安装介绍	25
3.3 登录	29
3.4 作业管理使用	29
3.4.1 查看作业信息	31
3.4.2 挂起作业	34
3.4.3 继续作业	34
3.4.4 终止作业	34
3.5 我的会话	35
3.6 数据管理	39
3.7.1 用户家目录	39

3.7.2 作业数据区	40
3.7.3 共享数据区	41
3.7 注销.....	42
3.8 Rstudio 使用.....	42
3.8.1 打开和登录	42
3.8.2 会话管理.....	44
第四章 注意事项.....	45
4.1 支持的浏览器版本.....	45
4.2 上传下载打不开的原因.....	45
4.3 提交作业选择 CPU 核数.....	45

第一章 高性能计算平台介绍

长安大学新高性能计算平台由 2 台登录节点、2 台管理节点、75 台双路计算节点、2 台大内存四路计算节点、5 台 8 卡 GPU 计算节点和 4 台 4 卡的高性能显卡图形节点组成，集群采用 HDR Infiniband 网络，并配置一套 1.3PB 统一命名空间的高性能并行文件存储系统。配置信息如下：

第二章 命令行使用方式

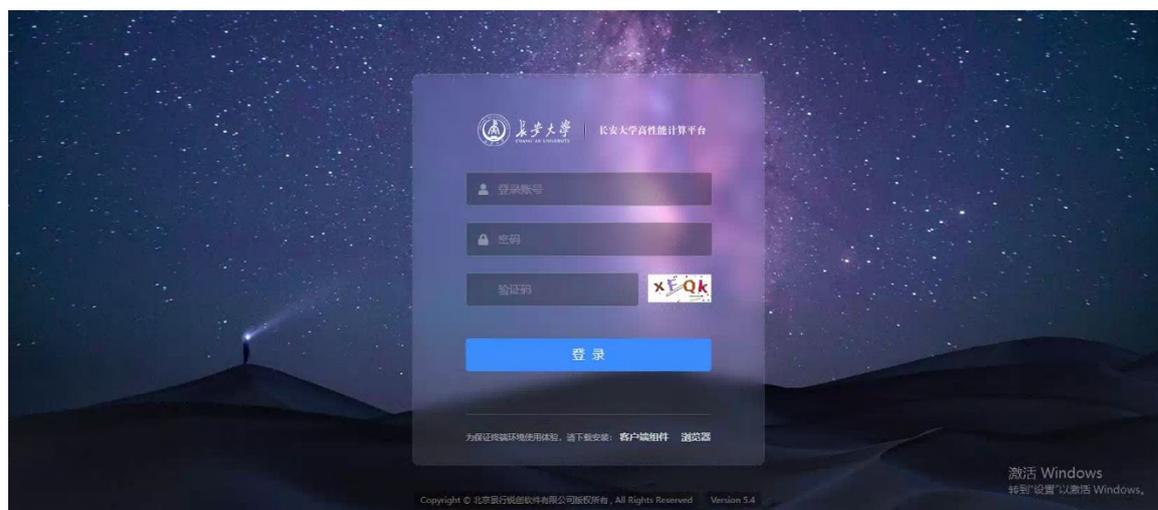
2.1. 平台登录方式

2.1.1 通过 ssh 登录

用户可通过第三方 ssh 工具（xshell、putty 等）登录高算平台。集群配置了 2 台登录节点，ssh 登录端口为 22，两台登录节点的 IP 地址分别为 10.131.0.202 以及 10.131.0.203。两台登录节点没有任何区别。

2.1.2 通过 web 门户登录

用户可通过浏览器（推荐 firefox）登录 HPC 集群门户，门户登录地址为 <https://10.131.0.199>，登录用户名和密码与 ssh 登录的账户信息一致，如下图所示：



用户输入与 ssh 登录信息一致的账户和密码后即可登录门户页面，可通过网

页的方式对自己的数据、作业进行管理。

2.2. 数据上传方式

用户可通过远程拷贝（scp、rsync 等命令）或通过第三方工具 winscp、sftp 的方式上传/下载数据。

2.3. 景行资源管理与调度软件简介

云计算资源管理：景行资源管理与调度软件可以将网络上多台异构类型的计算机资源整合为一个集群应用服务平台。应用程序资源的调用不再只局限于个人工作站的资源，也不需要为了满足个性化的资源需求而手动改动应用程序，只需要设置一些简单的应用脚本和命令就能够使用应用服务平台上的计算资源。景行资源管理与调度软件还可以根据节点主机的负载条件和应用程序的资源需求，从整个集群应用服务平台中选择最合适的计算节点。

执行作业：对于景行资源管理与调度软件管理的应用服务系统，远程和本机执行作业的行为一致。对于用户而言是开放和透明的，即使是最复杂的终端交互控制作业，也如同作业是在本地执行似的。当作业获得所需要的合适的软硬件资源，或当应用服务平台的系统负载较轻时，景行资源管理与调度软件会根据预定的策略自动执行作业，并可以根据资源负载情况，对作业进行完全控制。如：挂起作业和恢复作业运行。景行资源管理与调度软件能够以交互式作业或批处理作业的形式执行串行或并行的应用程序。当作业在一组资源负载较轻或空闲的资源节点上执行时，作业的执行效率会大大提高。

控制系统资源的访问：对于景行资源管理与调度软件管理的应用服务系统，系统管理员可以轻易地控制资源的访问，例如：

- 谁可以提交作业，这些作业可以使用哪些主机；
- 某个用户或者某个用户组最多可以同时执行多少个作业，最多可以使用多少种应用资源；
- 提交到指定队列的作业的资源限制；
- 每个计算单元可执行作业的时间窗口；
- 在某种负载条件下指定的计算单元可以接收作业或者挂起一些低优先级的作业。
- 可对提交的用户作业，进行运行内存的限制等。

资源和作业记账：景行资源管理与调度软件提供了资源和作业记账的机制。这些信息可以帮助管理员分析资源的使用情况和作业的执行情况，以及系统一天或者一周内的负载情况，同时帮助管理员确定是否有资源过载情况发生，为合理化系统配置或扩展和升级系统提供详尽的数据支持。

应用：绝大多数的应用都可以通过景行资源管理与调度软件的接口访问由其管理的应用服务系统。不需要直接与之发生交互，也不需要为了使用由景行资源管理与调度软件管理的应用服务器系统而刻意的修改程序。目前，由景行资源管理与调度软件管理的应用服务系统，可支持大多数的 Linux 或者 Windows 的命令和第三方应用程序。

容错：一旦有计算资源请求，景行资源管理与调度软件会通过内置高效的调度策略寻找可用计算资源，并保证计算资源的请求被及时的派发后执行。在应用服务系统中，只要还有一台服务器在运行，景行资源管理与调度软件就能够继续接收计算资源请求。如果计算资源请求执行失败，景行资源管理与调度软件就会主动的把该请求重新派发到另一个可满足资源需求的可用服务器上执行。景行资源管理与调度软件将整个应用服务系统的状态保存在文件数据库中。只要该事务日志文件可访问，景行资源管理与调度软件就能保证执行所有的计算资源请求。并可为该事务日志文件设置镜像备份，以足够保证当主文件服务器停止工作时，景行资源管理与调度软件可以根据镜像的事务日志文件继续执行作业的操作，为整个集群应用服务系统提供了执行作业的容错能力。

异构系统的支持：景行资源管理与调度软件是连接不同操作系统的中心枢纽。其通用的架构使得景行资源管理与调度软件非常容易支持多种类型的操作系统。它不仅支持 Linux 和 Windows 操作系统，还可支持 Linux 和 Windows 之间的相互操作。

并行处理：景行资源管理与调度软件支持 MPI (Message Passing Interface)。它既是集群资源的分配者又是管理者，并可为每个并行作业模块寻找最佳适合的主机。

调度策略：景行资源管理与调度软件提供了快速、高效的调度策略，保证了受其管理的应用服务系统的正常运行。用户可以根据不同的需要使用不同的策略，例如，用户可以设置队列级的公平共享调度策略，控制和管理对应用资

源的需求冲突。景行资源管理与调度软件还内置了许多其它队列级别的调度策略，如最基本的先来先服务、抢占式和独占式策略。

资源预留：是指对某个作业或者队列强制预留资源。资源预留保证了正在运行的作业有足够的可用资源（通过对资源提前占位，可以很好的解决作业运行过程中因需求资源动态的变化而不够的问题）。

作业记账：应用服务系统记录作业的大量信息，比如说：

- 提交节点和执行节点。
- 提交、派发、执行作业和结束作业时间；
- 执行作业的资源开销；
- CPU 时间、作业整体周转时间和自然时间等；

所有这些数据都存储在一个作业记账文件中。

作业数组：作业数组延伸了作业的概念，作业从单个输入文件、单例执行的应用程序延伸为多个并行文件、多例执行的应用程序。许多现实世界的问题，如渲染一场动画或者是在批处理数据转换时，需要多次输入不同的数据参数来多次执行同一队列应用程序。使用景行资源管理与调度软件的作业数组功能，可以允许用户提交单个数组作业，而该数组作业可以使用不同的数据参数来多次执行同一应用程序。

共享资源：共享资源是指由景行资源管理与调度软件管理的集群中所有节点上的可用资源，这些资源可以在节点组之间共享。如应用程序的浮动许可证就是一个典型的可共享资源，集群中的所有节点都可以通过自动获取到的浮动许可证来执行该软件。景行资源管理与调度软件保证了在作业派发到各执行节点前获得有效的许可证后执行作业，从而使得该浮动许可证资源在各执行节点间得到合理化的应用。

并行作业的处理器预留：在同一个景行资源管理与调度软件管理的集群中执行并行和普通应用程序时，因为普通作业只需要一个 CPU 而并行作业程序则需要等待多个空闲的 CPU，所以并行作业程序所需要的 CPU 总会被普通应用程序先占用。并行作业处理器的预留功能，允许并行作业在排队期间将所需的空闲处理器（作业槽 slots）预留一段时间而不被其它的普通作业使用。

Job Starter：每一个景行资源管理与调度软件队列都可以配置一个 Job

Starter。Job Starter 是一个脚本或者是可执行程序，用来创建作业执行的环境。通过 Job Starter，景行资源管理与调度软件管理员可以自定义作业执行的环境。例如：

- 配置作业的输入/输出缓存和重定向
- 配置执行仿真作业的运行环境

可配置的作业控制方式：可通过景行资源管理与调度软件的作业控制，改变作业在集群系统中的运行状态。通常情况下，作业先是进入 PEND 状态，然后进入 RUN 状态，作业执行完成后显示为 DONE 状态。有时会在作业的生命周期内，被系统挂起进入 SSUSP 状态，或者是被用户挂起而进入 USUSP 状态。景行资源管理与调度软件给管理员提供了配置控制作业时所触发的一系列动作，当作业状态改变时，这些自定义的触发动作将会被执行。

基于数据库的调度框架：景行资源管理与调度软件提供了开放的基于数据库的调度框架。用户可以根据该框架的要求，定义设置调度策略，从而更高效地利用集群的资源，实现对调度策略的深度定制。

GPU 调度：景行资源管理与调度软件提供了 GPU 调度功能，该功能会自动检测节点 GPU 信息，并将 GPU 信息管理起来，用于调度使用。GPU 调度支持两种模式，分别是基础 GPU 调度和 BIND GPU 调度，基础 GPU 调度是用户可以将 GPU 定义为资源，再写一个用来收集该自定义资源的 ELIM 脚本，这样就可以将 GPU 作为一种资源来调度。BIND GPU 调度是以基础 GPU 调度为基础，添加了给作业绑定 GPU 的功能，使作业独占被分配到的 GPU。

第三章 调度命令行使用方式

作业提交命令行说明：

```
jsub -n <CPU 数> -q <队列名> [-J <作业名> -o output.%.%J -e error.%.%J] ./commandLine
```

-n <CPU 数> 指定执行该作业所需的 CPU 数

-q <队列名> 指定该作业运行时使用的队列，所有的作业都会被提交到队列里在资源不足时排队，在资源满足时派发并执行。队列由管理员分配使用，所以用户在提交作业时确认下自己是否有权限使用该队列(jqueues -u <自己账户名>来列出自己可用的队列)。

-J <作业名> 指定作业名,方便辨识作业

-o output.%J 指定输出文件,这个输出指的是调度系统的输出,如果作业的输出没有重定向到应用自己的log文件里也会输出在这个文件里。%J表示的作业号,及最终会产生一个output.<作业号>的文件。

作业查看

jjobs [[-a] [-l] <作业号>]

jjobs 不使用参数,显示目前正在排队或运行的作业

```
[jhctest@ln01 Linpack]$ jjobs
```

JOBID	USER	STAT	QUEUE	FROM_HOST	EXEC_HOST	JOB_NAME	SUBMIT_TIME	ORDER
12829	jhctest	RUN	normal	ln01	64*cnode*	*_test_1node	Jul 04 14:24	-

jjobs -a 显示最近完成的作业及正在运行或排队的作业

```
[jhctest@ln01 Linpack]$ jjobs
```

JOBID	USER	STAT	QUEUE	FROM_HOST	EXEC_HOST	JOB_NAME	SUBMIT_TIME	ORDER
10572	jhctest	DONE	normal	ln01	64*cnode*	*_test_1node	Jul 02 21:37	-
12829	jhctest	RUN	normal	ln01	64*cnode*	*_test_1node	Jul 04 14:24	-

jjobs <作业号> 显示指定的作业状态

```
[jhctest@ln01 2nodes]$ jjobs 12829
```

JOBID	USER	STAT	QUEUE	FROM_HOST	EXEC_HOST	JOB_NAME	SUBMIT_TIME	ORDER
12829	jhctest	RUN	normal	ln01	64*cnode*	*_test_1node	Jul 04 14:24	-

jjobs -l <作业号> 显示指定作业的详细信息

```
[jhctest@ln01 2nodes]$ jjobs -l 12829
```

Job <12829>, Job Name <hpl_test_1node>, User <jhctest>, Project <default>, Status <RUN>, Queue <normal>, Application <default>, APS Priority <5000>, Command <#!/bin/bash;#JSUB -q normal;#JSUB -n 64;#JSUB -e error.%J;#JSUB -o output.%J;#JSUB -J hpl_test_1node;###>>> The following content is generally not modified. <<<###;ncpu=`echo \$JH_HOSTS | awk '{print \$2}'`;###>>>

The above content is generally not modified. <<<###;##

apps run command;module load intel-OneAPI-2022.2.0;EXEC=/storage/public/home/jhctest/Linpack/xhpl_intel64_dynamic;

```

OMP_NUM_THREADS=32 mpirun -np 2 $EXEC>
Tue Jul 04 14:24:25: Submitted from host <ln01>, CWD </storage/public/home/
jhtest/linpack/1node>, Output File <output.12829>, Error F
ile <error.12829>, 64 Processors Requested;
Tue Jul 04 14:24:25: Started on 64 Hosts/Processors <64*cnode122>, Execution Ho
me </storage/public/home/jhtest>, Execution CWD </storage/
public/home/jhtest/linpack/1node>, Execution user <jhtest>
;
Tue Jul 04 14:29:20: Resource usage collected.

The CPU time used is 4183.74 seconds.

IDLE_RATE(cputime/slots/runtime): 0.2267

MEM: 476 Gbytes; SWAP: 0 Bbytes

PGIDs:1249876 1249961 1249965 1249966; PIDs:1249876
1249877 1249880 1249944 1249955 1249960 1249961 1249965
1249966

SCHEDULING PARAMETERS:

      r15s   r1m   r5m   r15m   ut   pg   io   Ls
LoadSched -   -   -   -   -   -   -   -
LoadStop  -   -   -   -   -   -   -   -

      it   tmp   swap   mem
LoadSched -   -   -   -
LoadStop  -   -   -   -

```

作业终止

jctl kill <作业号>

```

[jhtest@ln01 ~]$ jctl kill 12829
Job <12829> is being terminated

```

3.1 加载指定工具

Module 是一个用于管理用户不同应用的工具，可以动态的加载、移除所安装的应用。工具的 modulefiles 文件编写完毕后，module 工具就会识别，此时使用 module avail, 即可显示所有可用模块：

```
[jhtest@ln01 ~]$ module avail
----- /storage/public/apps/modulefiles/compiler -----
compiler-rt go-1.20.5 intel-compiler-2021.6.0 intel-OneAPI-2022.2.0
tbb
----- /storage/public/apps/modulefiles/mathlib -----
fftw-3.3.8-gcc831 fftw-3.3.8-intel gsl-2.7-gcc831 htlib-1.17-gcc831
intelmkl-2022.1.0 openblas-0.3.21-gcc831 scalapack-2.2.1-gcc831
----- /storage/public/apps/modulefiles/mpi -----
intelmpi-2021.6 openmpi-4.1.5-gcc831 ucx-1.14.1
----- /storage/public/apps/modulefiles/software -----
AdmixTools-7.0.2      beast-2.7.4          cmake-3.26.4
eig-7.2.1            hic-pro-2.11.4      lastz-1.04.15
panacus-0.2.1_py310  samtools-1.9
```

使用 module load 可以加载模块，如 module load R-4.3.0-gcc831, 就可以加载 R 工具：

```
[jhtest@ln01 ~]$ module load R-4.3.0-gcc831
Loading R - 4.3.0 : /storage/public/apps/software/R/R-4.3.0/
```

查看是否加载成功：

```
[jhtest@ln01 ~]$ which R
/storage/public/apps/software/R/R-4.3.0/bin/R
```

此时使用 module list 可以查看目前已经加载的工具：

```
[jhtest@ln01 ~]$ module list
Currently Loaded Modulefiles:
1) R-4.3.0-gcc831
```

使用 module rm 就可以取消加载工具，如 module rm R-4.3.0-gcc831 ,

可以取消加载 R 工具：

```
[jhstest@ln01 ~]$ module rm R-4.3.0-gcc831
Removing R - 4.3.0
Use 'module list' to view any remaining dependent modules.
```

如果想取消所有工具的加载，使用 `module purge` 命令。

注意，如果加载某个软件后，在实际运行时报错提示“xxx command not found”，这可能是由于环境变量冲突导致，需使用 `module purge` 命令清除冲突的环境变量，然后再使用 `module load` 命令加载某个软件的环境变量，命令举例如下：

```
[jhstest@ln01 ~]$ module purge ; module load R-4.3.0-gcc831
```

3.2 提交通用作业

向一个队列提交作业，只要队列的状态是打开的，用户就可以向该队列提交。只有当队列是激活状态时，作业才能被派发执行。如果一个作业不属于并行作业，则提交到默认的队列中进行计算。

提交命令：

```
[jhstest@ln01 ~]$ jsub -q normal -n 1 -m cnode1 -e error.%J -o output.%J -J my_job .
/test.sh
Job <12833> is submitted to queue <normal> and default application <default>
```

参数详解：

<jsub>：提交作业

<-q normal>：指定 normal 队列

<-n 1>：指定该作业所需 1 核

<-m cu01>：指定该作业运行在 cnode1 节点（一般无需指定）

<-e error.%J>：错误信息输出到 error.%J 文件中

<-o output.%J>：正确信息输出到 output.%J 文件中

<-J my_job>：指定作业名字为 my_job

<test.sh>：执行的脚本

直到作业执行的所有条件都满足之前，作业一直处于等待状态。每个队列都有它自己的执行条件，该条件应用于队列中所有的作业。当用户提交作业时，可以指定其它附加的条件。

注意：如果待提交的作业需要在单台计算节点上执行（即不跨节点执行），则需在提交命令中加入“-R ” span[hosts=1]”关键字，无需通过“-m”参数指定特定节点，且申请的核心数不能大于队列中单台计算节点的物理核心数。命令举例如下：

```
[jhstest@ln01 ~]$ jsub -q normal -n 64 -R " span[hosts=1]" -e error.%J -o output.%J  
-J my_job ./test.sh
```

3.3 提交 R 并行作业

cd 到测试文件所在目录后，当前目录下存在 test.r (r 脚本)

```
[jhstest@ln01 ~]$ jsub -n 64 -q normal -e error.%J -o output.%J test.r
```

3.4 提交 mpi 并行作业

cd 到测试文件所在的目录后，当前目录下存在 test 脚本（脚本中为 mpi 命令）；

在 test 脚本中 mpirun 或 mpiexec 命令行的上一行添加 mpi 环境变量加载命令；

集群中提供了两种 mpi，分别为 Intel 编译器套件中的 IntelMPI 和开源的 OpenMPI，可分别通过如下命令加载：

```
[jhstest@ln01 ~]$ module load openmpi-4.1.5-gcc831  
[jhstest@ln01 ~]$ module load intelmpi-2021.6
```

提交作业：

```
[jhstest@ln01 ~]$ jsub -n 64 -q normal -e error.%J -o output.%J ./test
```

3.5 提交 tensorflow 作业

cd 到测试文件所在的目录后，当前目录下存在 test.py

提交到 cpu 队列

```
[jhctest@ln01 ~]$ jsub -q normal -n 2 -m cu001 -e error.%J -o output.%J -J mjob test.py
```

提交到 gpu 队列

```
[jhctest@ln01 ~]$ jsub -q gpu -n 1 -gpgpu 1 -m gpu05 -e error.%J -o output.%J test.py
```

3.5 提交 gpu 通用作业

提交到 gpu 的队列中

```
[jhctest@ln01 ~]$ jsub -q gpu -n 1 -gpgpu 1 -m gpu05 -e error.%J -o output.%J test.py
```

第四章 门户应用使用方式

这里以提交一个 fluent 作业为例来描述提交高性能作业的过程。

4.1 Fluent 计算任务提交

登录门户后，点击“应用仓库”中 Fluent 图标，系统会弹出如下提交页面：



Fluent(fluent)

目前集群中可用于计算【fluent】作业的CPU核数为4480 历史表单 保存表单

作业名:

CAS 文件*:

DAT 文件:

Dual-time 时间步数:

迭代步数*:

自动保存步数:

CPU核数:

维数:

计算结果名:

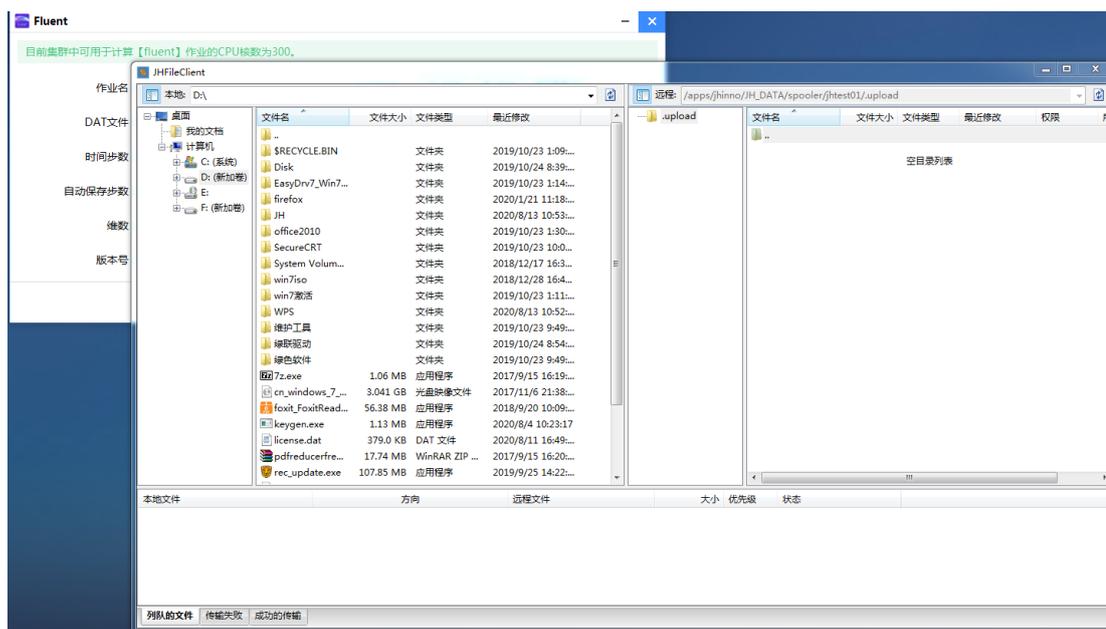
版本号:

图形界面支持: 关

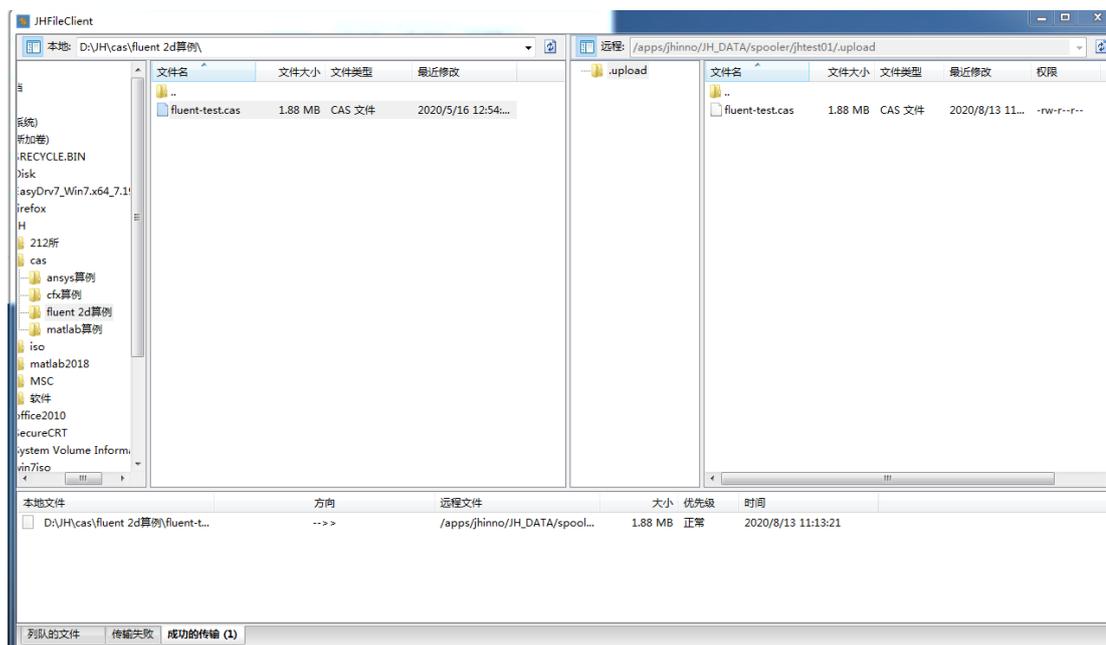
用户可以自主选择作业名称和选择 Fluent 的 CAS 算例文件

用户可以点击本地或服务器来选择文件。若点击“本地”，系统在安装客户

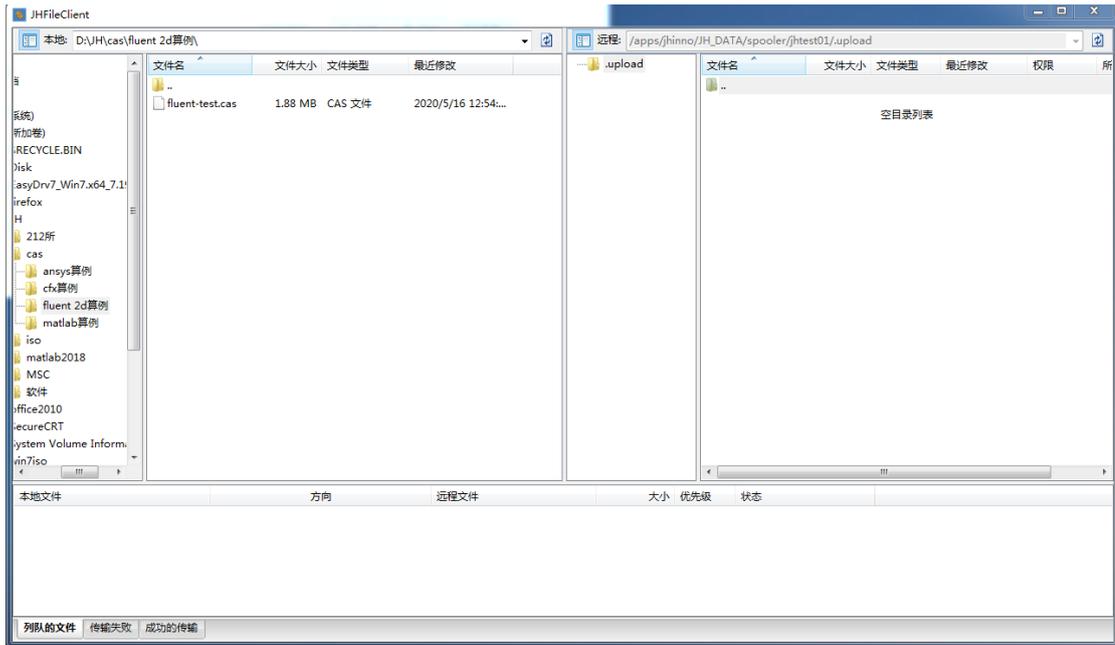
端软件后会自动弹出 JHFileClient 对话框：



用户可以通过左侧的本地选择对应的目录，如选择【D:\JH\cas\fluent 2d算例\】：



在选择好目录后，在要上传的文件处点击右键，选择上传，文件即开始上传到高性能服务器：



用户也可以点击“服务器端”来选择服务器端数据，进入服务器端后，选择需要计算的文件，点击“确定”。

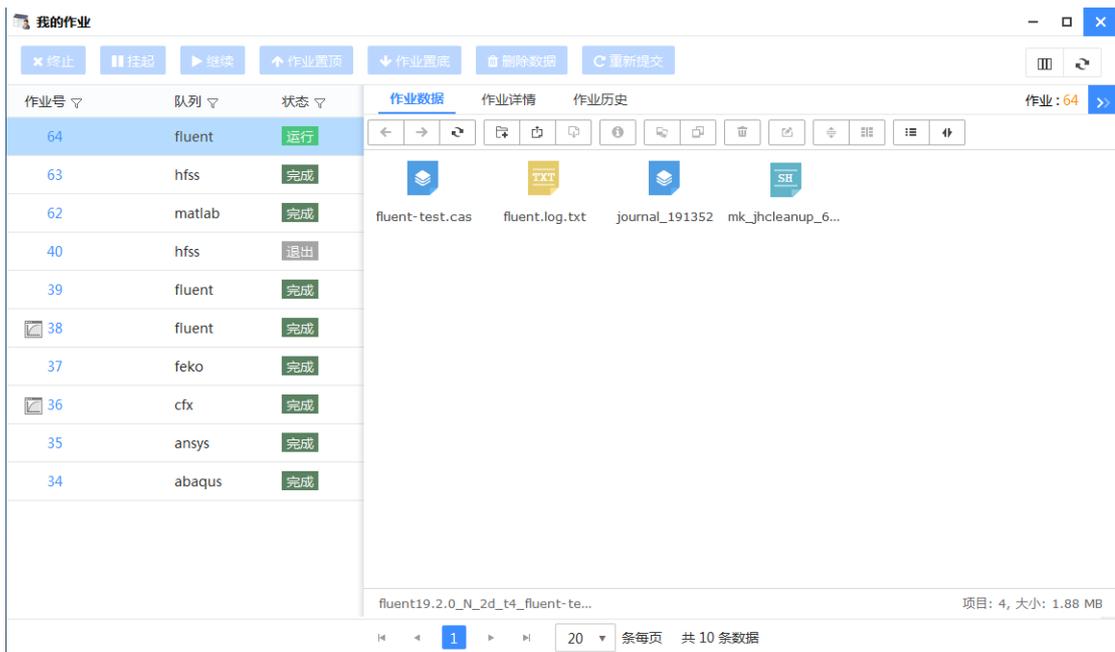
选择 CPU 核数；

填写迭代步数；

选择算例类型；

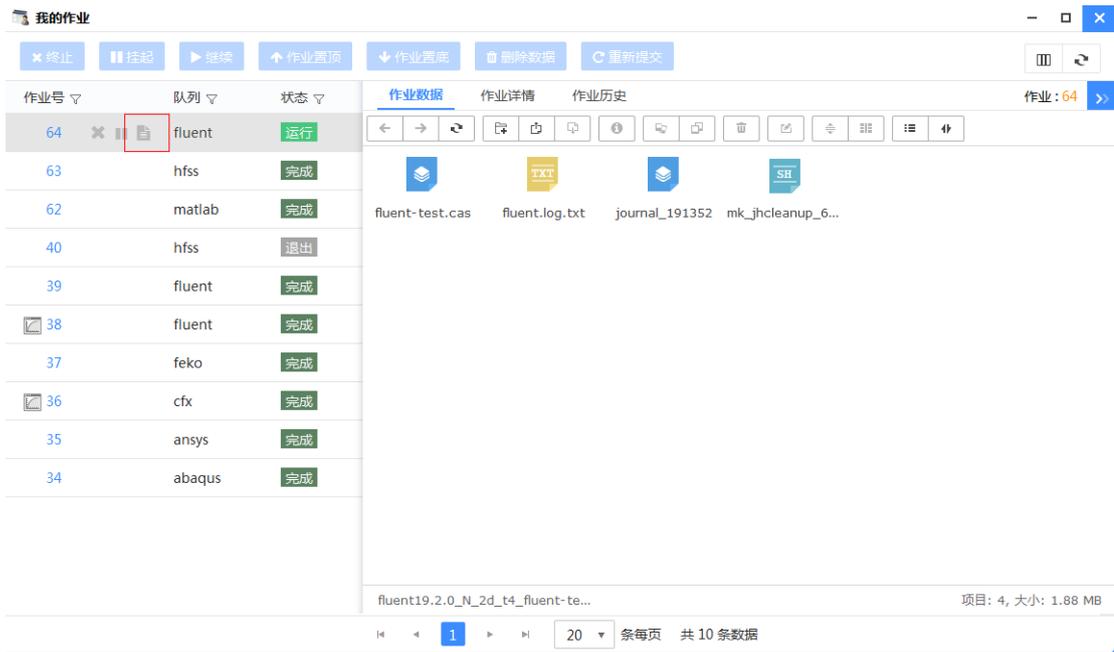
点击确定提交作业。

系统会自动跳转到作业详细页面，如下图所示：



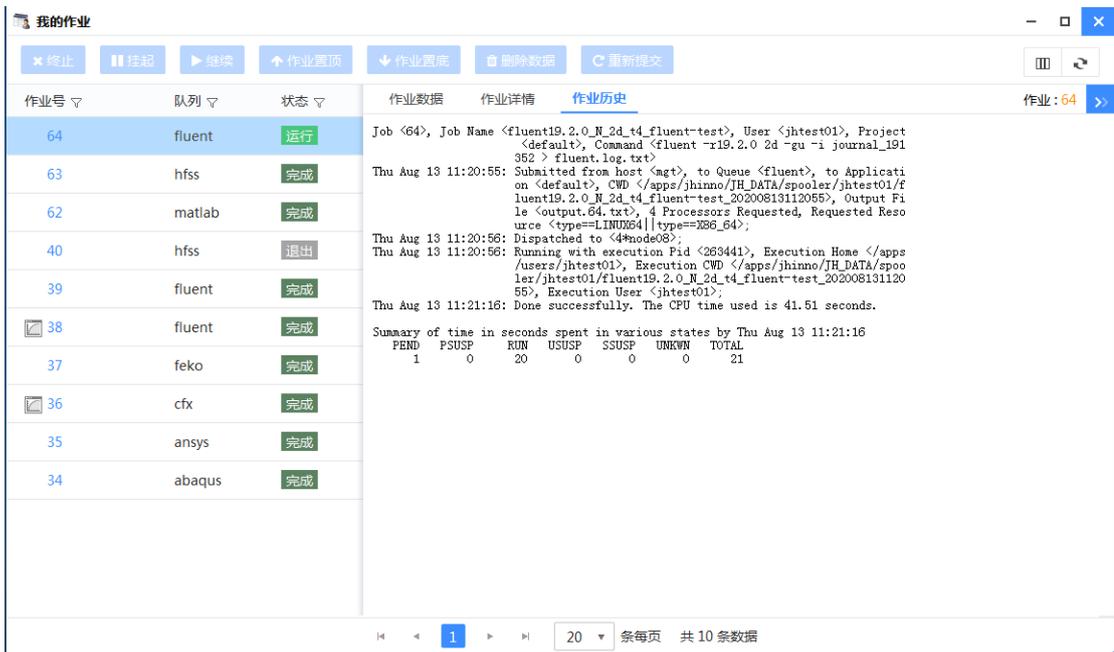
在这个页面中，我们可以点击“刷新”来刷新实时的数据情况和作业情况，

同时也可以点击左侧面板中的“详细信息”进入该作业的详细页面：

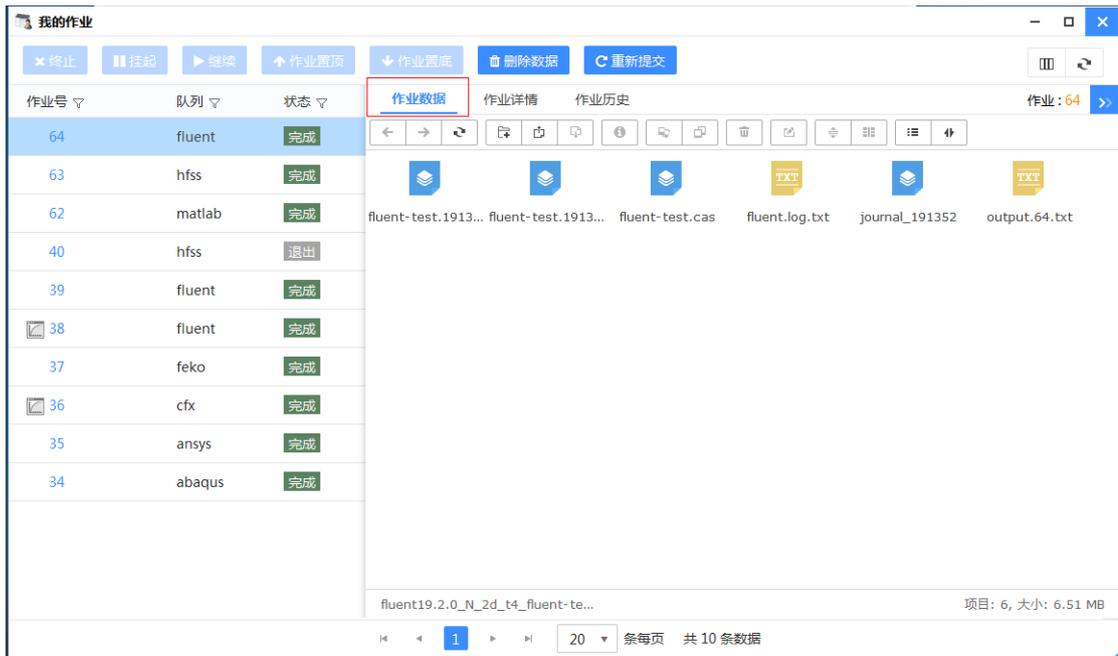


用户可以“查看作业输出”和“查看作业历史”。

点击“查看作业历史”：



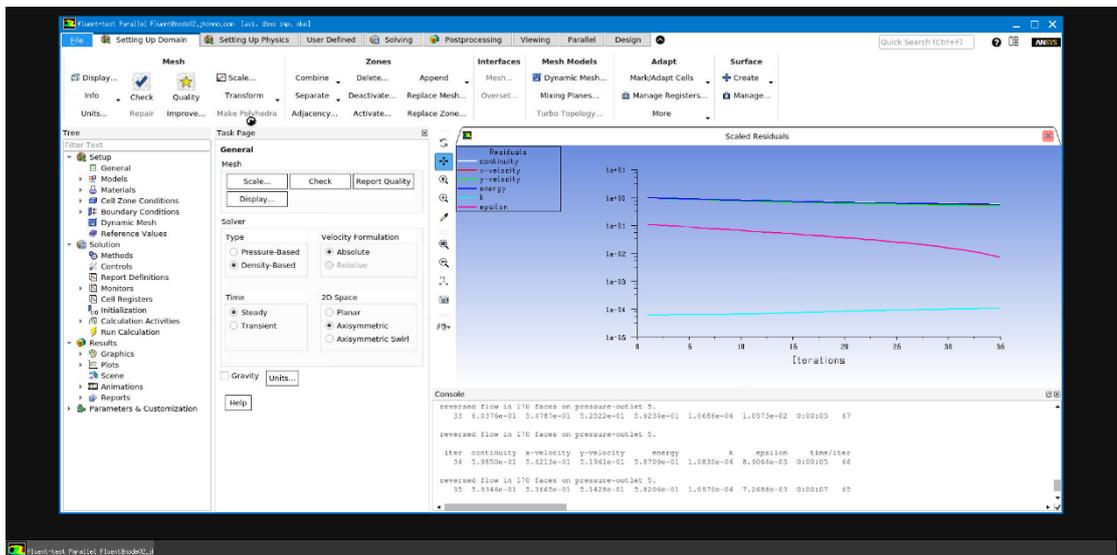
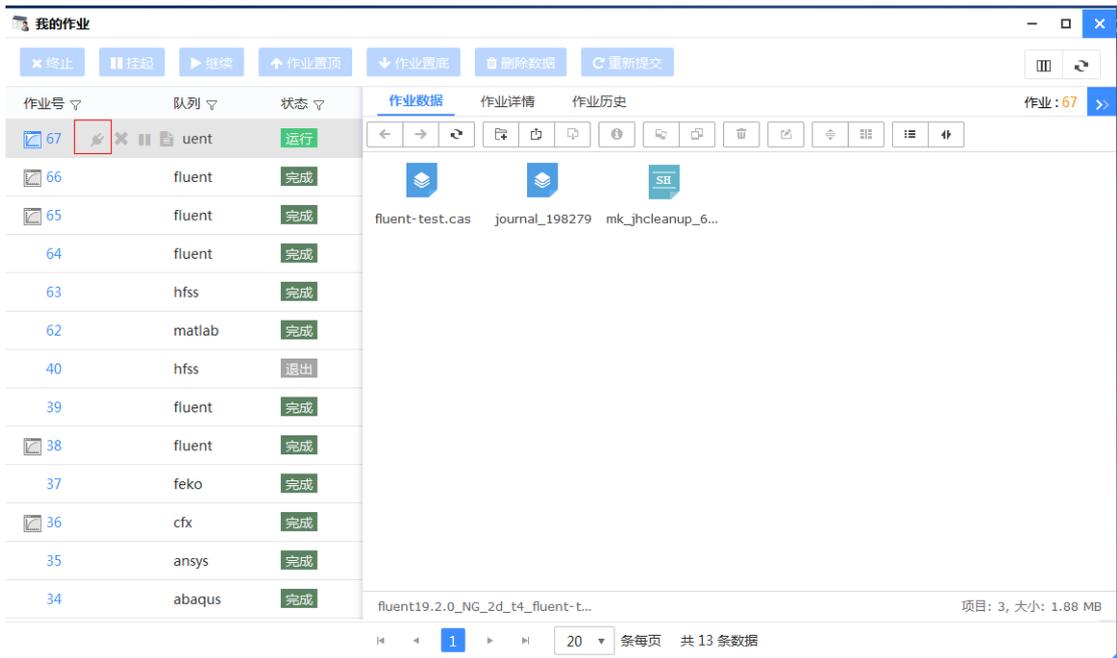
作业完成之后我们可以对作业进行后处理操作，点击作业数据按钮进入作业的数据列表页面。



fluent 可以实时查看作业收敛条件，如果需要在提交 fluent 的同时查看收敛，需勾选下图的图形界面支持。



提交成功后，点击作业号前面的打开应用图形，就会自动跳转到 fluent 计算的 web 界面。



同时，我们在作业详细页面中可以看到：



4.2 fluent 交互模式计算任务提交

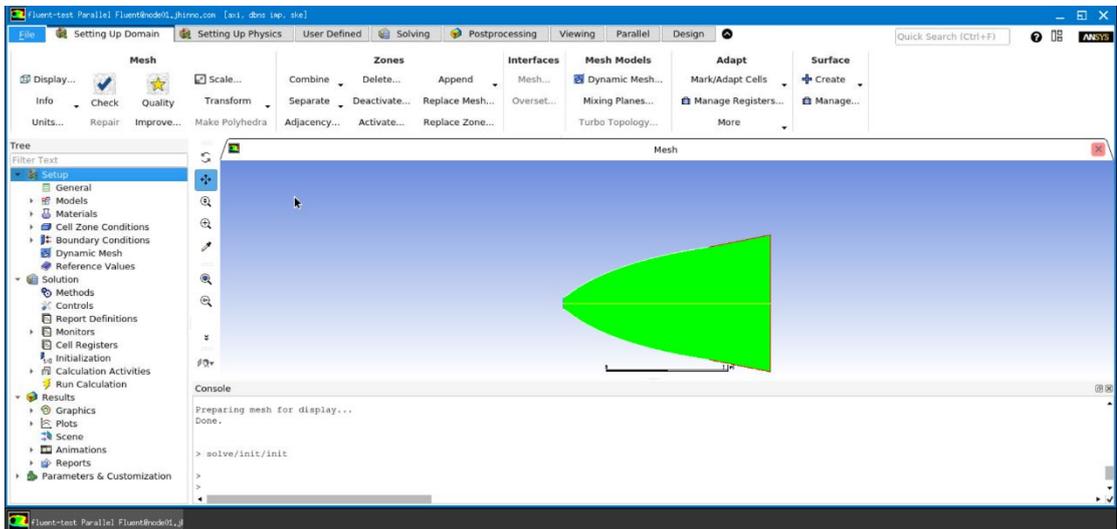
在登陆的主页面可以点击仿真计算的 Fluent 交互模式图标 ，

点击之后会弹出 Fluent 交互模式的提交页面，如下

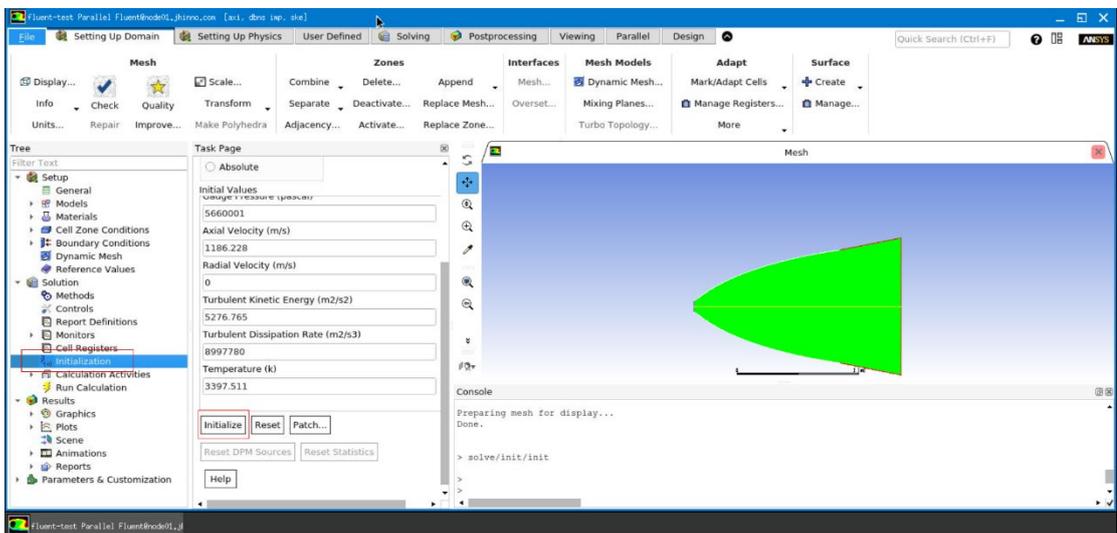


The screenshot shows the 'Fluent交互模式(fluent_interactive)' dialog box. At the top, it indicates '目前集群中可用于计算【fluent_interactive】作业的CPU核数为4480'. Below this, there are several input fields and dropdown menus: '作业名:' (Job Name), 'CAS文件*:' (CAS File) with '本地' (Local) and '远端' (Remote) options, 'DAT文件:' (DAT File) with '本地' (Local) and '远端' (Remote) options, 'CPU核数:' (CPU Cores) set to 1, '维数:' (Dimension) set to 3d, and '版本号:' (Version) set to 23.1. At the bottom, there are '取消' (Cancel) and '确定' (OK) buttons.

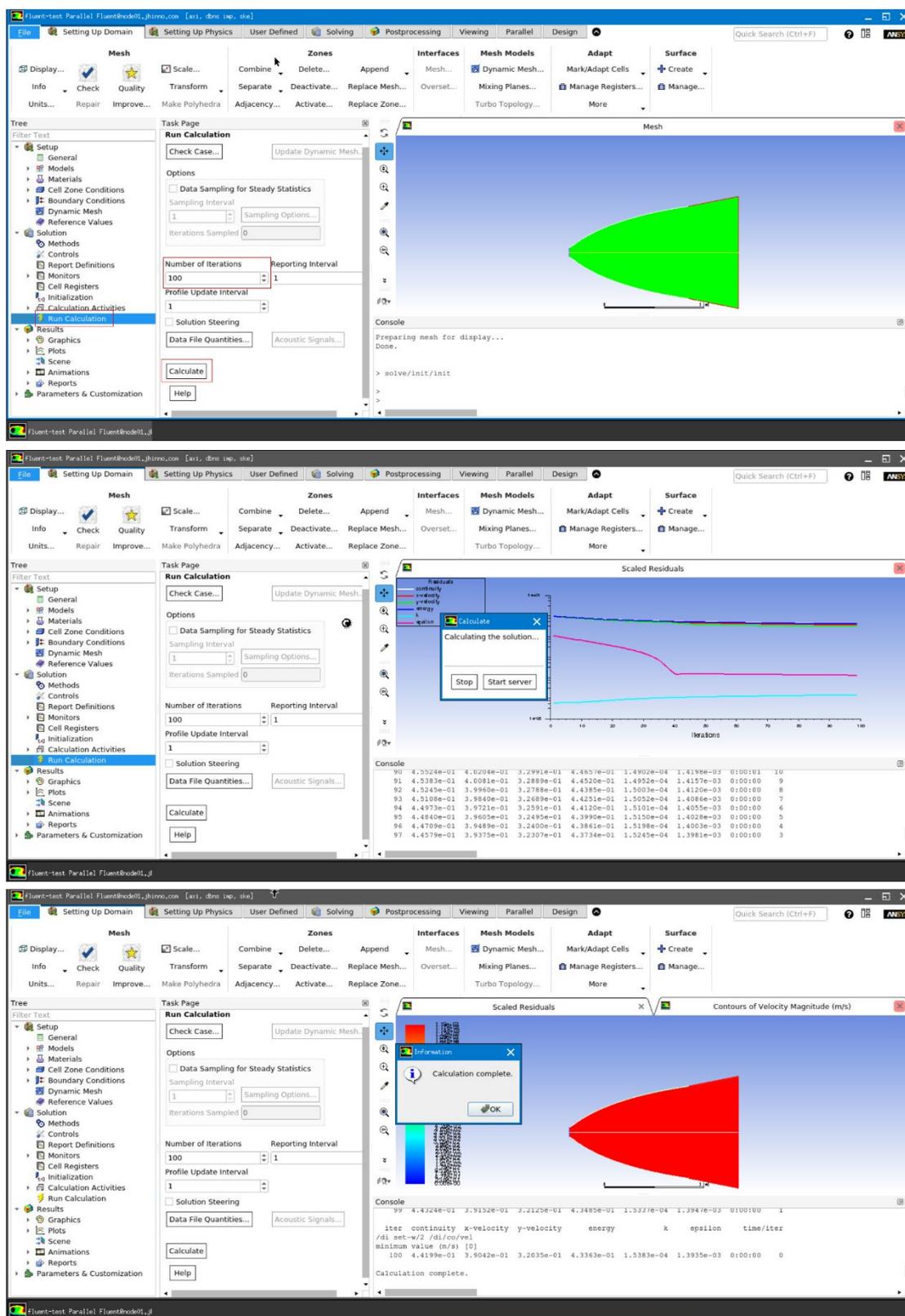
将图中各选项填写完成（其中带*的是必填项），点击确定提交作业，页面跳转到该作业的详细信息页面，在该页面可以查看作业的详细信息。并且可以在作业信息页面打开图形界面，图形界面上可以显示 Fluent 作业的运行过程：



选择左列 Initialization，点击初始化



设置好迭代步数后点击运行计算



完成计算，点击右上角关闭按钮退出。

4.3 Matlab集群配置

高算集群matlab软件linux版已正版化，可以正常使用。第一次使用请注意：

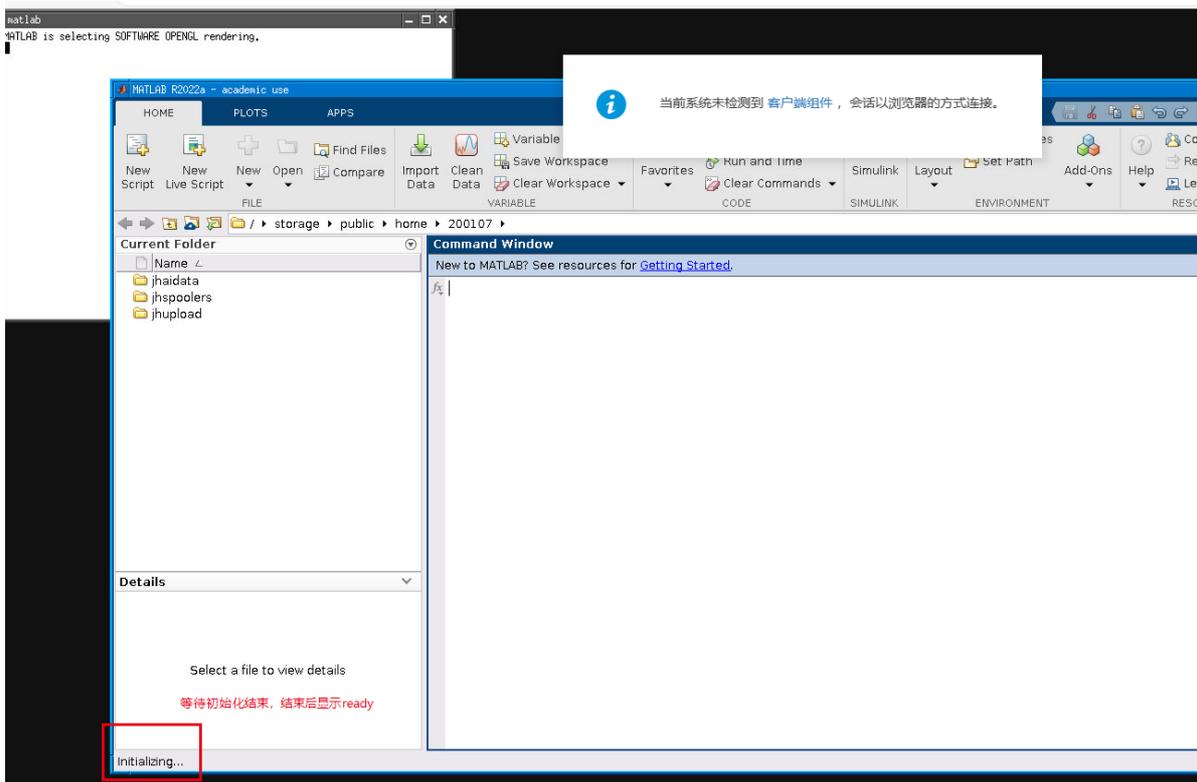
正版的matlab每个账户第一次使用的时候都要设置一下matlab集群配置，否则就只能在后台1个核跑。

配置步骤如下：

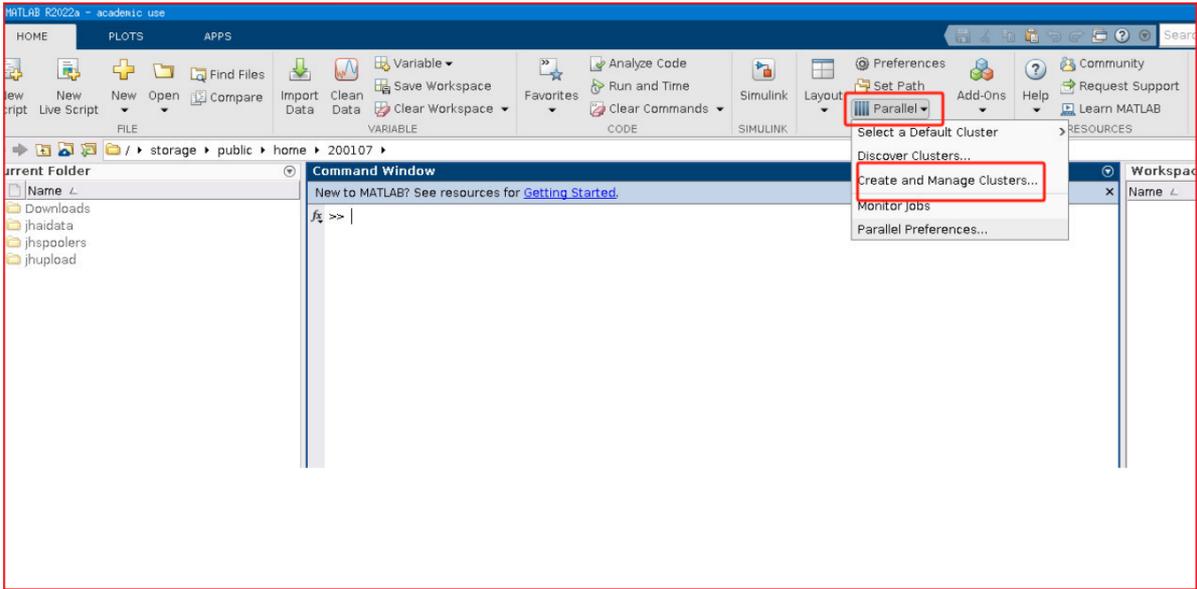
step 1: 打开平台 matlab 软件



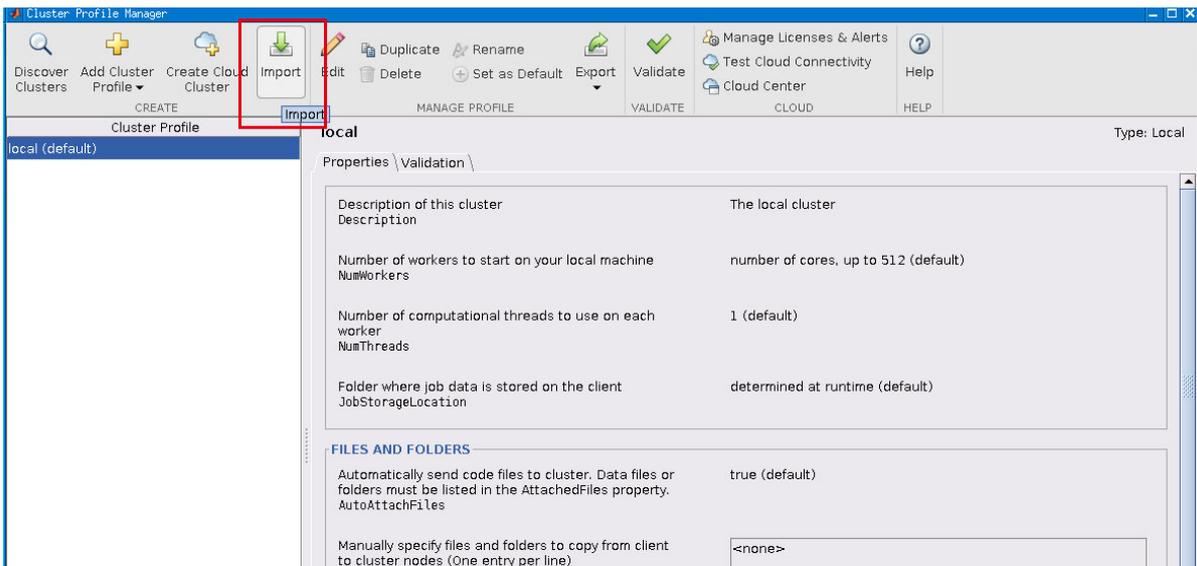
step 2: 等待 matlab 初始化完成



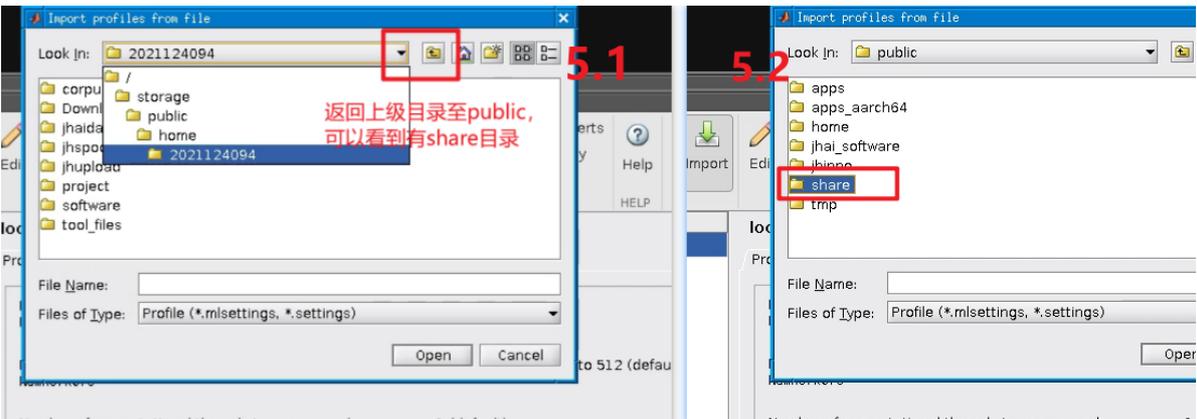
step 3: 依次选择 “Parallel” 、 “Create and Manage Clusters...”



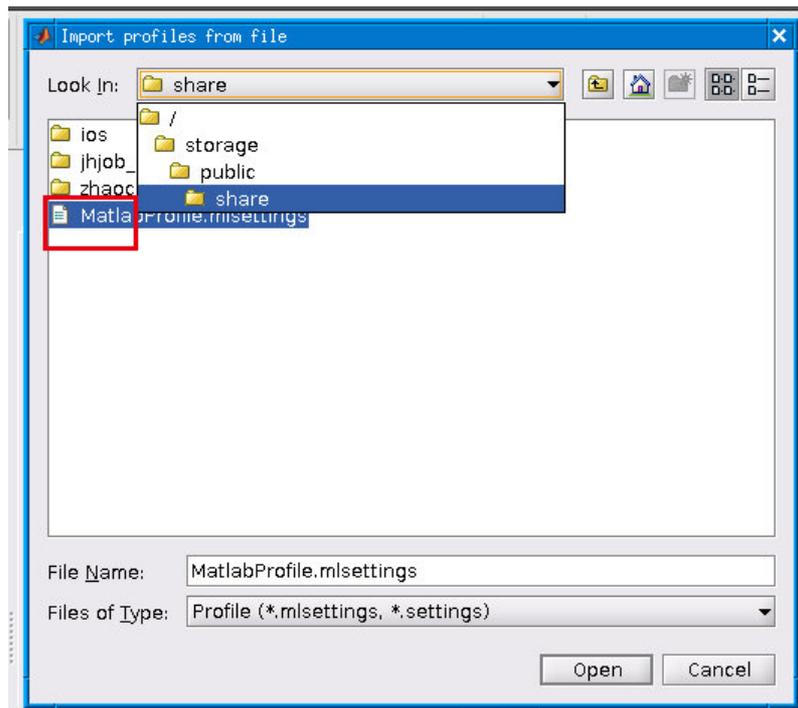
step 4: 待页面加载完成后，单击 import 按钮



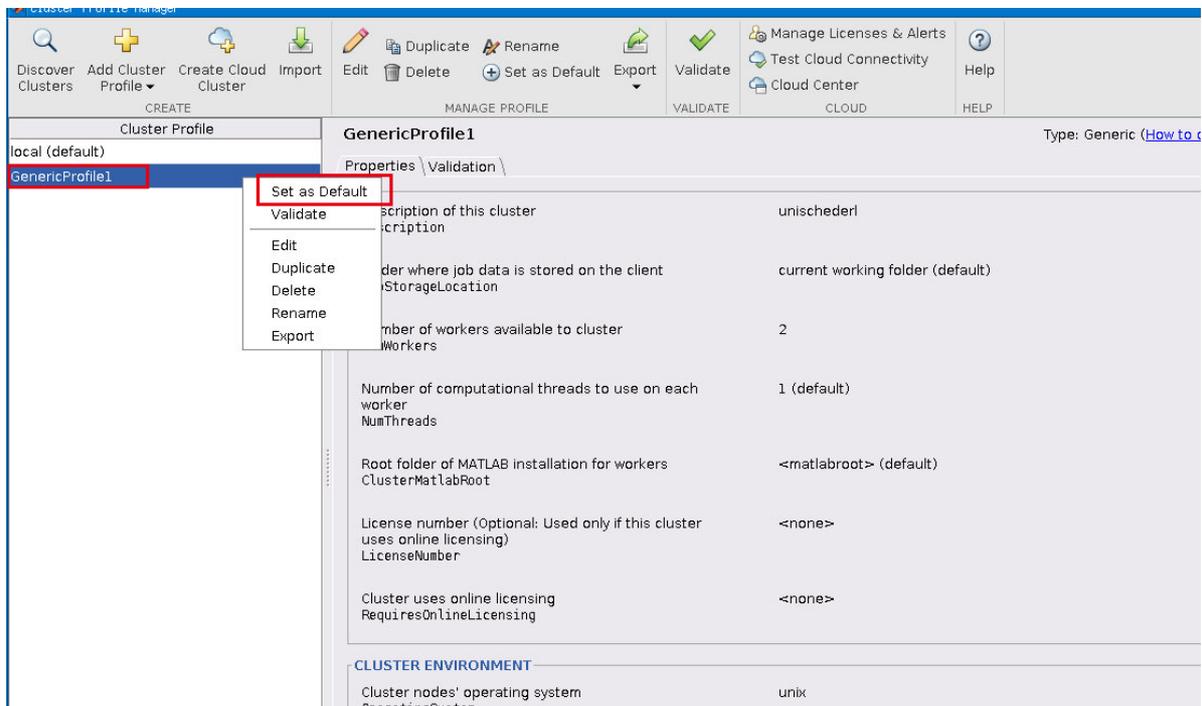
step 5: 返回上级目录，至 public 目录下，可以看到有 share 目录



step 6: 打开share目录，然后选中文件 MatlabProfile.mlsettings 并将其打开（open）



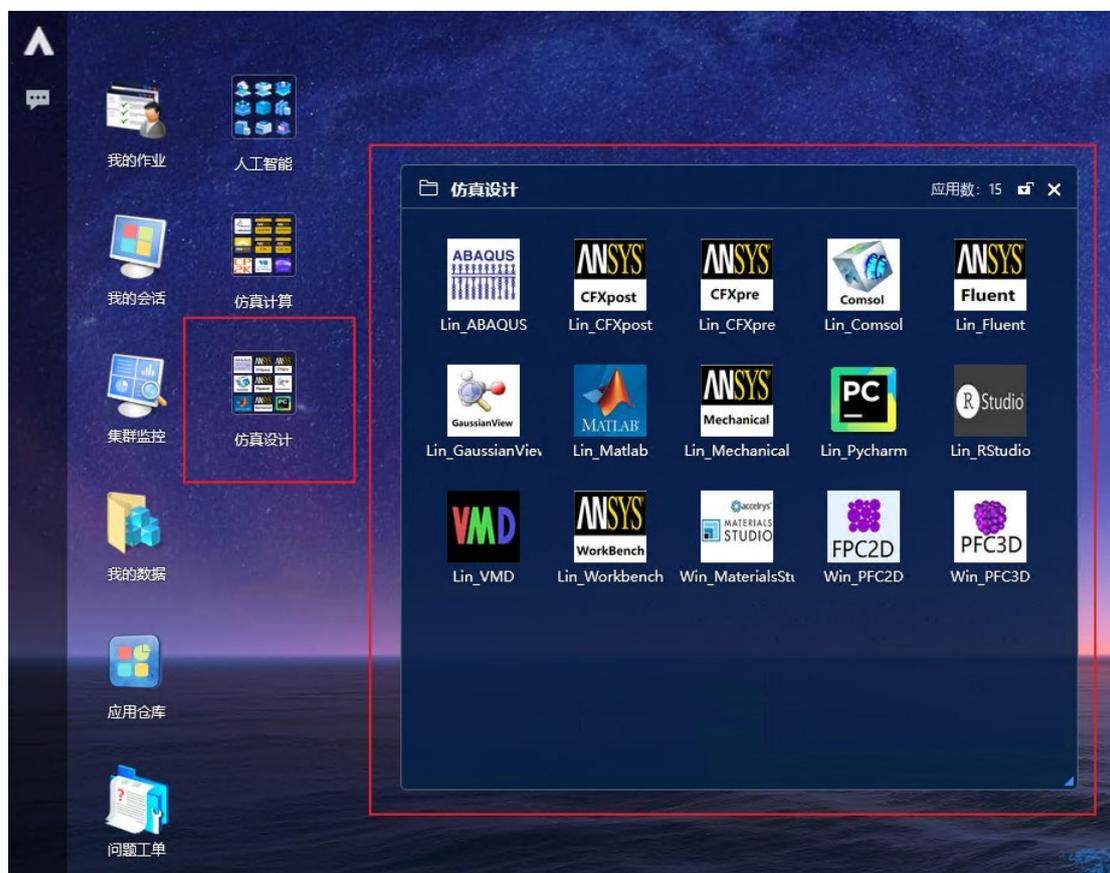
step 7: 选中 GenericProfile1，然后鼠标“右键单击”，选择“Set as Default”单击即可



上述设置步骤设置完成后，就可以关闭此页面，matlab就可以正常使用了

4.4 仿真设计应用方式

用户登录门户后，可在仿真设计菜单中选择需要的软件打开，常用软件如下所示：



使用鼠标双击软件图标后会弹出该软件的 UI 图形会话。更多会话使用和管理请参考“我的会话”小节。

第五章 门户其他功能

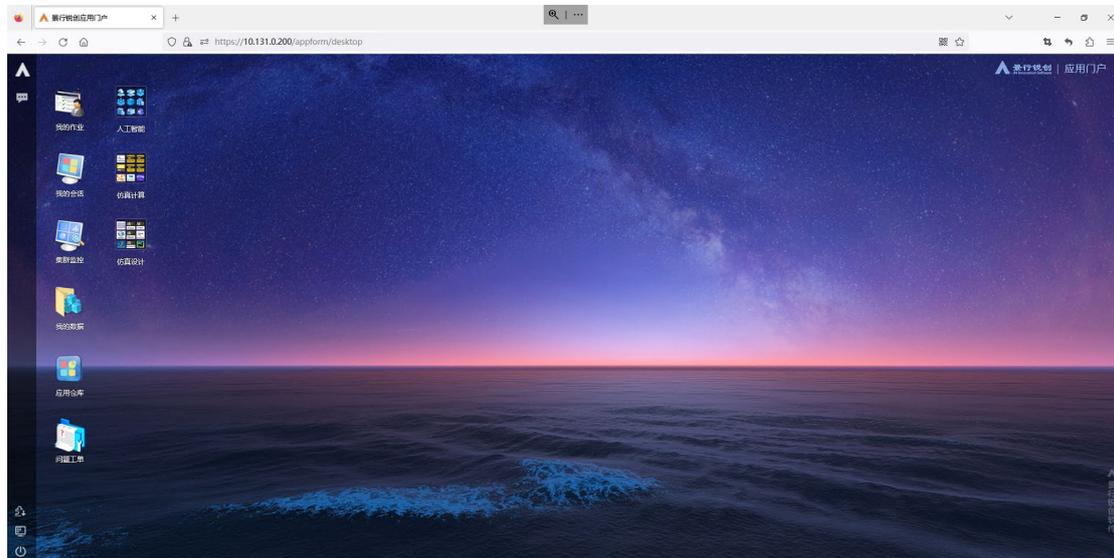
3.1 文件传输工具安装介绍

jhfileclient 主要是用来做文件的上传、下载。可以上传几十 G 的大文件，并且可以实现断点续传。访问 <https://10.131.0.199> 进入高算平台的登录页面，在页面上下载客户端组件并安装。

3.3 登录

访问 <https://10.131.0.199> 进入高算平台的登录页面，如下：

输入用户名密码登录，登录进去的主页面如下：



3.4 作业管理使用

作业管理主要是指查看作业信息、挂起作业、继续作业、终止作业四个功能。还可以按作业号、状态、队列、提交时间、执行节点、作业名对作业进行排序。

管理员可以在作业管理页面查看到所有用户的作业，并对这些作业进行操作。而其他用户仅能在该页面上查看到自己提交的作业。

作业管理页面如图所示：

作业号 ▾	状态 ▾	队列 ▾	执行节点 ▾	提交时间 ▾	执行时间 ▾	作业名 ▾
239	DONE	tensorflow_gpu	2*gpu01	11-04 15:22:10	11-04 15:22:10	test3.py
238	EXIT	tensorflow_gpu	2*gpu02	11-04 15:20:02	11-04 15:20:04	test3.py
237	DONE	tensorflow_gpu	2*gpu02	11-04 15:16:17	11-04 15:16:18	test3.py
230	DONE	qgpu	gpu02	11-04 11:59:50	11-04 11:59:50	./run.sh
229	DONE	rsnow	30*cu04	11-04 11:33:20	11-04 11:33:21	testjob1
228	DONE	rsnow	30*cu04	11-04 11:29:16	11-04 11:29:18	test_local_var.R
227	EXIT	rsnow	30*cu04	11-04 11:25:06	11-04 11:25:08	test_local_var.R
226	EXIT	rsnow	30*cu04	11-04 10:57:31	11-04 10:57:33	test_local_var.R
225	EXIT	rsnow	30*cu04	11-04 10:55:02	11-04 10:55:05	test_local_var.R
224	DONE	rsnow	30*cu04	11-04 10:46:35	11-04 10:46:37	testjob1
223	DONE	rsnow	30*cu04	11-04 10:45:12	11-04 10:45:12	testjob1
222	DONE	rsnow	30*cu04	11-04 10:44:05	11-04 10:44:06	testjob1
221	EXIT	rsnow	30*cu04	11-04 10:35:08	11-04 10:35:08	testjob1

我的作业页面

作业管理页面上显示了以下元素：

- (1) 作业号：默认显示；
- (2) 状态：默认显示，其中作业的状态包含有：RUN、PEND、UNKWN、UNKNOW、PSUSP、USUSP、SSUSP、ZOMBI、DONE、EXIT；
- (3) 队列：默认显示，显示作业运行的队列名称；
- (4) 执行节点：默认显示；
- (5) 提交时间：默认显示；
- (6) 执行时间：显示的是作业执行的时间点；
- (7) 作业名：显示的是作业的名称

作业管理页面对每一列提供了过滤设置，并支持个性化手动设置过滤条件。用户可以通过在每一列右边的漏斗状图标，就可以进行每一列元素的过滤。这些过滤条件会自动保存起来，下次访问该页面的时候，会执行自己设置的过滤条。

其中每一列的过滤条件有：

- (1) 作业号：提供等于、大于、和小于三个过滤条件。过滤展开框如图所示：

作业号	状态	队列	执行节点	提交时间	执行时间	作业名
239		tensorflow_gpu	2*gpu01	11-04 15:22:10	11-04 15:22:10	test3.py
238		tensorflow_gpu	2*gpu02	11-04 15:20:02	11-04 15:20:04	test3.py
237		tensorflow_gpu	2*gpu02	11-04 15:16:17	11-04 15:16:18	test3.py
230	DONE	qgpu	gpu02	11-04 11:59:50	11-04 11:59:50	./run.sh
229	DONE	rsnow	30*cu04	11-04 11:33:20	11-04 11:33:21	testjob1
228	DONE	rsnow	30*cu04	11-04 11:29:16	11-04 11:29:18	test_local_var.R
227	EXIT	rsnow	30*cu04	11-04 11:25:06	11-04 11:25:08	test_local_var.R
226	EXIT	rsnow	30*cu04	11-04 10:57:31	11-04 10:57:33	test_local_var.R
225	EXIT	rsnow	30*cu04	11-04 10:55:02	11-04 10:55:05	test_local_var.R
224	DONE	rsnow	30*cu04	11-04 10:46:35	11-04 10:46:37	testjob1
223	DONE	rsnow	30*cu04	11-04 10:45:12	11-04 10:45:12	testjob1
222	DONE	rsnow	30*cu04	11-04 10:44:05	11-04 10:44:06	testjob1
221	EXIT	rsnow	30*cu04	11-04 10:35:08	11-04 10:35:08	testiob1

作业号过滤框（我的作业页面）

(2) 状态：提供等于、大于和小于三个过滤条件，并提供一个选择框选择作业号与两个过滤条件组合使用。过滤展开框如图所示：

作业号	状态	队列	执行节点	提交时间	执行时间	作业名
239		tensorflow_gpu	2*gpu01	11-04 15:22:10	11-04 15:22:10	test3.py
238		tensorflow_gpu	2*gpu02	11-04 15:20:02	11-04 15:20:04	test3.py
237		tensorflow_gpu	2*gpu02	11-04 15:16:17	11-04 15:16:18	test3.py
230	DONE	qgpu	gpu02	11-04 11:59:50	11-04 11:59:50	./run.sh
229	DONE	rsnow	30*cu04	11-04 11:33:20	11-04 11:33:21	testjob1
228	DONE	rsnow	30*cu04	11-04 11:29:16	11-04 11:29:18	test_local_var.R
227	EXIT	rsnow	30*cu04	11-04 11:25:06	11-04 11:25:08	test_local_var.R
226	EXIT	rsnow	30*cu04	11-04 10:57:31	11-04 10:57:33	test_local_var.R
225	EXIT	rsnow	30*cu04	11-04 10:55:02	11-04 10:55:05	test_local_var.R
224	DONE	rsnow	30*cu04	11-04 10:46:35	11-04 10:46:37	testjob1
223	DONE	rsnow	30*cu04	11-04 10:45:12	11-04 10:45:12	testjob1
222	DONE	rsnow	30*cu04	11-04 10:44:05	11-04 10:44:06	testjob1
221	EXIT	rsnow	30*cu04	11-04 10:35:08	11-04 10:35:08	testiob1

状态过滤框（我的作业页面）

(3) 队列：提供筛选的过滤条件。过滤展开框如图所示：

我的作业

× 终止 | 挂起 | 继续 | 作业置顶 | 作业置底 | 删除数据 | 重新提交

作业号	状态	队列	执行节点	提交时间	执行时间	作业名
239	DONE	tens		11-04 15:22:10	11-04 15:22:10	test3.py
238	EXIT	tens		11-04 15:20:02	11-04 15:20:04	test3.py
237	DONE	tensorflow_gpu	2*gpu02	11-04 15:16:17	11-04 15:16:18	test3.py
230	DONE	qgpu	gpu02	11-04 11:59:50	11-04 11:59:50	./run.sh
229	DONE	rsnow	30*cu04	11-04 11:33:20	11-04 11:33:21	testjob1
228	DONE	rsnow	30*cu04	11-04 11:29:16	11-04 11:29:18	test_local_var.R
227	EXIT	rsnow	30*cu04	11-04 11:25:06	11-04 11:25:08	test_local_var.R
226	EXIT	rsnow	30*cu04	11-04 10:57:31	11-04 10:57:33	test_local_var.R
225	EXIT	rsnow	30*cu04	11-04 10:55:02	11-04 10:55:05	test_local_var.R
224	DONE	rsnow	30*cu04	11-04 10:46:35	11-04 10:46:37	testjob1
223	DONE	rsnow	30*cu04	11-04 10:45:12	11-04 10:45:12	testjob1
222	DONE	rsnow	30*cu04	11-04 10:44:05	11-04 10:44:06	testjob1
221	EXIT	rsnow	30*cu04	11-04 10:35:08	11-04 10:35:08	testiob1

1 2 3 4 20 条每页 共 68 条数据

队列过滤框（我的作业页面）

(4) 执行节点：提供过滤设置，过滤展开框如图所示：

我的作业

× 终止 | 挂起 | 继续 | 作业置顶 | 作业置底 | 删除数据 | 重新提交

作业号	状态	队列	执行节点	提交时间	执行时间	作业名
239	DONE	tensorflow_gpu	2*gpu01		11-04 15:22:10	test3.py
238	EXIT	tensorflow_gpu	2*gpu02		11-04 15:20:04	test3.py
237	DONE	tensorflow_gpu	2*gpu02	11-04 15:16:17	11-04 15:16:18	test3.py
230	DONE	qgpu	gpu02	11-04 11:59:50	11-04 11:59:50	./run.sh
229	DONE	rsnow	30*cu04	11-04 11:33:20	11-04 11:33:21	testjob1
228	DONE	rsnow	30*cu04	11-04 11:29:16	11-04 11:29:18	test_local_var.R
227	EXIT	rsnow	30*cu04	11-04 11:25:06	11-04 11:25:08	test_local_var.R
226	EXIT	rsnow	30*cu04	11-04 10:57:31	11-04 10:57:33	test_local_var.R
225	EXIT	rsnow	30*cu04	11-04 10:55:02	11-04 10:55:05	test_local_var.R
224	DONE	rsnow	30*cu04	11-04 10:46:35	11-04 10:46:37	testjob1
223	DONE	rsnow	30*cu04	11-04 10:45:12	11-04 10:45:12	testjob1
222	DONE	rsnow	30*cu04	11-04 10:44:05	11-04 10:44:06	testjob1
221	EXIT	rsnow	30*cu04	11-04 10:35:08	11-04 10:35:08	testiob1

1 2 3 4 20 条每页 共 68 条数据

节点过滤框（我的作业页面）

(5) 提交时间：作业的提交时间的过滤条件。过滤展开框如图所示：

我的作业

[× 终止](#)
[|| 挂起](#)
[▶ 继续](#)
[⏴ 作业置顶](#)
[⏵ 作业置底](#)
[🗑 删除数据](#)
[↶ 重新提交](#)

作业号	状态	队列	执行节点	提交时间	执行时间	作业名
239	DONE	tensorflow_gpu	2*gpu01	11-04 15:16:17	11-04 15:22:10	test3.py
238	EXIT	tensorflow_gpu	2*gpu02	11-04 15:16:17	11-04 15:20:04	test3.py
237	DONE	tensorflow_gpu	2*gpu02	11-04 15:16:17	11-04 15:16:18	test3.py
230	DONE	qgpu	gpu02	11-04 11:59:50	11-04 11:59:50	./run.sh
229	DONE	rsnow	30*cu04	11-04 11:33:20	11-04 11:33:21	testjob1
228	DONE	rsnow	30*cu04	11-04 11:29:16	11-04 11:29:18	test_local_var.R
227	EXIT	rsnow	30*cu04	11-04 11:25:06	11-04 11:25:08	test_local_var.R
226	EXIT	rsnow	30*cu04	11-04 10:57:31	11-04 10:57:33	test_local_var.R
225	EXIT	rsnow	30*cu04	11-04 10:55:02	11-04 10:55:05	test_local_var.R
224	DONE	rsnow	30*cu04	11-04 10:46:35	11-04 10:46:37	testjob1
223	DONE	rsnow	30*cu04	11-04 10:45:12	11-04 10:45:12	testjob1
222	DONE	rsnow	30*cu04	11-04 10:44:05	11-04 10:44:06	testjob1
221	EXIT	rsnow	30*cu04	11-04 10:35:08	11-04 10:35:08	testjob1

1 2 3 4 20 条每页 共 68 条数据

提交时间过滤框（我的作业页面）

(6) 执行时间：作业的执行时间的过滤条件。过滤展开框如图所示：

我的作业

[× 终止](#)
[|| 挂起](#)
[▶ 继续](#)
[⏴ 作业置顶](#)
[⏵ 作业置底](#)
[🗑 删除数据](#)
[↶ 重新提交](#)

作业号	状态	队列	执行节点	提交时间	执行时间	作业名
239	DONE	tensorflow_gpu	2*gpu01	11-04 15:16:17	11-04 15:22:10	test3.py
238	EXIT	tensorflow_gpu	2*gpu02	11-04 15:16:17	11-04 15:20:04	test3.py
237	DONE	tensorflow_gpu	2*gpu02	11-04 15:16:17	11-04 15:16:18	test3.py
230	DONE	qgpu	gpu02	11-04 11:59:50	11-04 11:59:50	./run.sh
229	DONE	rsnow	30*cu04	11-04 11:33:20	11-04 11:33:21	testjob1
228	DONE	rsnow	30*cu04	11-04 11:29:16	11-04 11:29:18	test_local_var.R
227	EXIT	rsnow	30*cu04	11-04 11:25:06	11-04 11:25:08	test_local_var.R
226	EXIT	rsnow	30*cu04	11-04 10:57:31	11-04 10:57:33	test_local_var.R
225	EXIT	rsnow	30*cu04	11-04 10:55:02	11-04 10:55:05	test_local_var.R
224	DONE	rsnow	30*cu04	11-04 10:46:35	11-04 10:46:37	testjob1
223	DONE	rsnow	30*cu04	11-04 10:45:12	11-04 10:45:12	testjob1
222	DONE	rsnow	30*cu04	11-04 10:44:05	11-04 10:44:06	testjob1
221	EXIT	rsnow	30*cu04	11-04 10:35:08	11-04 10:35:08	testjob1

1 2 3 4 20 条每页 共 68 条数据

执行时间过滤框（我的作业页面）

(7) 作业名：作业名的过滤条件。过滤展开框如图所示：

作业号	状态	队列	执行节点	提交时间	执行时间	作业名
239	DONE	tensorflow_gpu	2*gpu01	11-04 15:22:10	11-04 15:22:10	test3.p
238	EXIT	tensorflow_gpu	2*gpu02	11-04 15:20:02	11-04 15:20:04	test3.p
237	DONE	tensorflow_gpu	2*gpu02	11-04 15:16:17	11-04 15:16:18	test3.py
230	DONE	qgpu	gpu02	11-04 11:59:50	11-04 11:59:50	./run.sh
229	DONE	rsnow	30*cu04	11-04 11:33:20	11-04 11:33:21	testjob1
228	DONE	rsnow	30*cu04	11-04 11:29:16	11-04 11:29:18	test_local_var.R
227	EXIT	rsnow	30*cu04	11-04 11:25:06	11-04 11:25:08	test_local_var.R
226	EXIT	rsnow	30*cu04	11-04 10:57:31	11-04 10:57:33	test_local_var.R
225	EXIT	rsnow	30*cu04	11-04 10:55:02	11-04 10:55:05	test_local_var.R
224	DONE	rsnow	30*cu04	11-04 10:46:35	11-04 10:46:37	testjob1
223	DONE	rsnow	30*cu04	11-04 10:45:12	11-04 10:45:12	testjob1
222	DONE	rsnow	30*cu04	11-04 10:44:05	11-04 10:44:06	testjob1
221	EXIT	rsnow	30*cu04	11-04 10:35:08	11-04 10:35:08	testiob1

作业名过滤框（我的作业页面）

下面将详细介绍作业管理的四个主要功能：查看作业信息、挂起作业、继续作业、终止作业

3.4.1 查看作业信息

作业信息主要包括用户、作业号、队列、状态、命令、提交节点、执行节点、提交目录、执行目录、作业槽数，提交时间、执行时间、结束时间、CPU 执行时间、作业历史信息、作业输出等。作业信息页面如图所示：

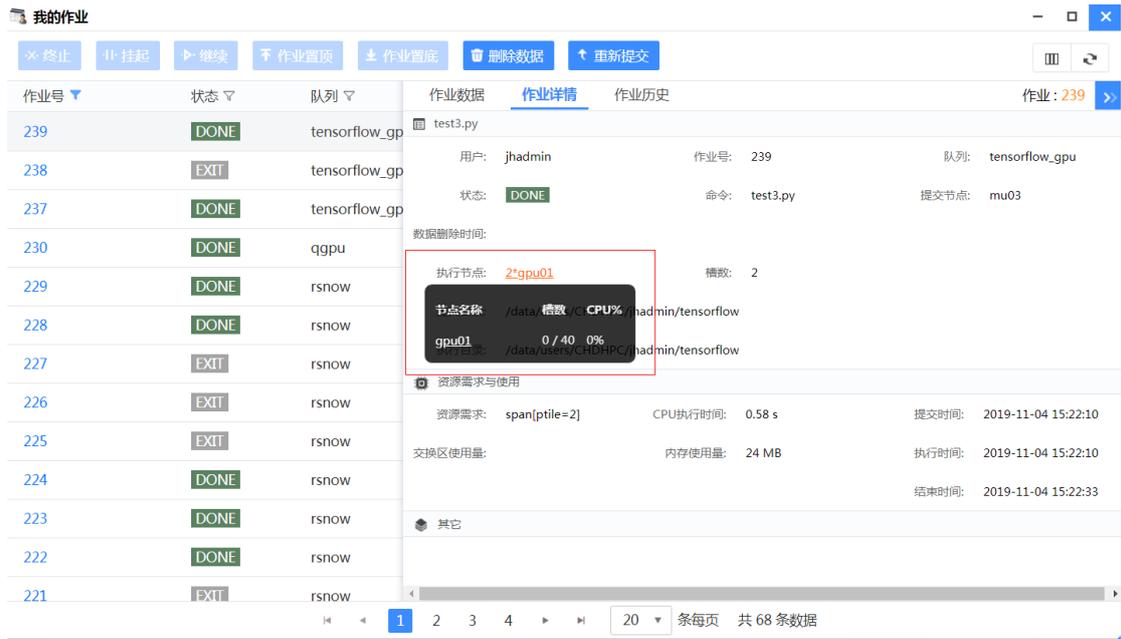
作业号	状态	队列	作业数据	作业详情	作业历史
239	DONE	tensorflow_gp	test3.py	用户: jhadmin	作业号: 239
238	EXIT	tensorflow_gp		状态: DONE	命令: test3.py
237	DONE	tensorflow_gp		数据删除时间:	提交节点: mu03
230	DONE	qgpu		执行节点: 2*gpu01	槽数: 2
229	DONE	rsnow		提交目录: /data/users/CHDHPC/jhadmin/tensorflow	
228	DONE	rsnow		执行目录: /data/users/CHDHPC/jhadmin/tensorflow	
227	EXIT	rsnow		资源需求与使用	
226	EXIT	rsnow		资源需求: span[ptile=2]	CPU执行时间: 0.58 s
225	EXIT	rsnow		提交时间: 2019-11-04 15:22:10	
224	DONE	rsnow		交换区使用量:	内存使用量: 24 MB
223	DONE	rsnow		执行时间: 2019-11-04 15:22:10	
222	DONE	rsnow		结束时间: 2019-11-04 15:22:33	
221	EXIT	rsnow		其它	

作业信息页面

作业详细信息页面统计了作业的资源需求与使用，以及作业的执行情况。详细信息页面统计了以下几项：

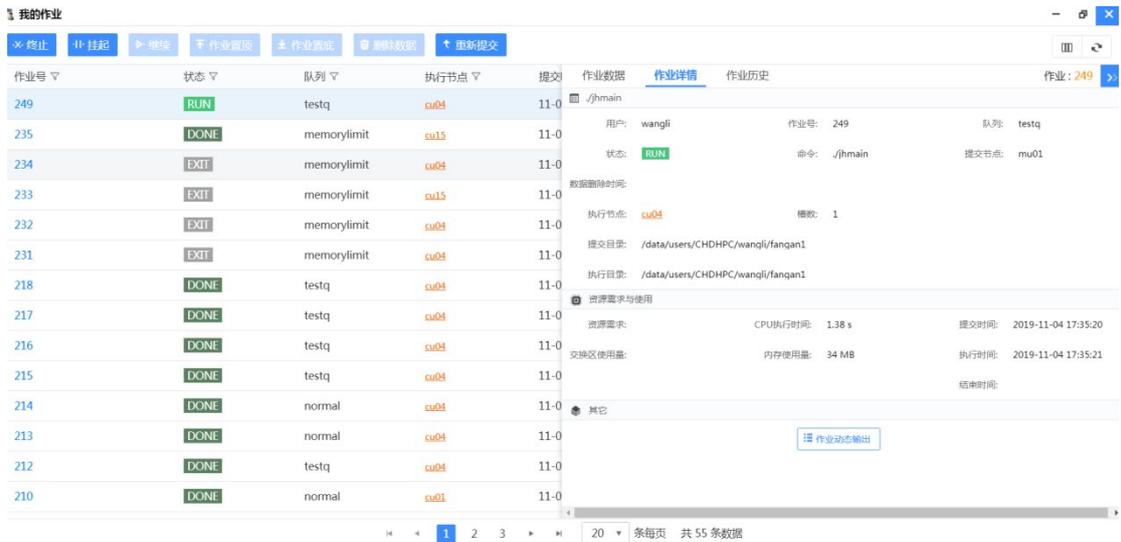
- 作业号
- 作业名
- 用户：作业的执行用户。
- 队列：作业的执行队列
- 项目：作业的项目名
- 状态：作业的实时状态
- 命令：作业的执行命令
- 提交节点/提交目录
- 执行节点/执行目录
- 作业槽数：若是 RUN 状态的作业，该参数指的是作业占用的槽数，若是 PEND 状态即为作业执行所需要的槽数。
- 资源需求：作业提交的请求资源串。
- CPU 执行时间：作业执行完成后所使用的 CPU 时间。在作业执行完成后才显示。
- 内存/交换区使用量：作业执行完成后所使用的内存和交换区使用。在作业执行完成后才显示。
- 提交/执行/结束时间

点击执行节点，会显示该节点的机器状态，如图所示：



作业提交节点信息

点击查看作业动态输出，可以在页面上直接看到作业的运行输出信息（只有在作业运行时才有输出），如图所示：



作业输出信息

点击查看作业历史，可以在页面上直接看到作业的历史信息，如图所示：

作业号	状态	队列	作业数据	作业详情	作业历史
239	DONE	tensorflow_gp	Job <239>, User <jhadmin>, Project <default>, Command <test3.py> Mon Nov 04 15:22:10: Submitted from host <mu03>, to Queue <tensorflow_gpu>, CWD </data/users/CHDHPC/jhadmin/tensorflow>, Output File <out.tt>, Error File <err.tt>, 2 Processors Requested, Each unique Host requests:2 GPUs <TeslaV100-SXM2-32GB>, Requested Resource :	Mon Nov 04 15:22:10: Dispatched to <2*gpu01>: Mon Nov 04 15:22:10: Running with execution Pid <110011>, Execution Home </data/users/CHDHPC/jhadmin>, Execution CWD </data/users/CHDHPC/jhadmin/tensorflow>, Execution User <jhadmin>: Mon Nov 04 15:22:33: Done successfully. The CPU time used is 0.38 seconds:	
238	EXIT	tensorflow_gp			
237	DONE	tensorflow_gp			
230	DONE	qgpu			
229	DONE	rsnow			
228	DONE	rsnow			
227	EXIT	rsnow			
226	EXIT	rsnow			
225	EXIT	rsnow			
224	DONE	rsnow			
223	DONE	rsnow			
222	DONE	rsnow			
221	EXIT	rsnow			

Summary of time in seconds spent in various states by Mon Nov 04 15:22:33

PEND	PSUSP	RUN	USUSP	SSUSP	UNKN	TOTAL
0	0	23	0	0	0	23

作业历史信息

3.4.2 挂起作业

仅能对 PENDING、RUN 状态的作业执行挂起操作。可以同时选择一个或多个作业进行挂起操作，也可以在作业信息页面对作业进行挂起操作。其中 PENDING 状态的作业挂起后状态变成 PSUSP，RUN 状态的作业挂起后状态变成 USUSP。

3.4.3 继续作业

仅能对 PSUSP、USUSP 状态的作业进行唤醒操作，使挂起的作业可以继续运行。可以同时选择一个或多个作业进行继续操作，也可以在作业信息页面对作业进行继续操作。其中 PSUSP 状态的作业继续后状态变成 PENDING，USUSP 状态的作业继续后状态变成 RUN。

3.4.4 终止作业

仅能对 PENDING、RUN、PSUSP、USUSP 状态的作业进行终止操作。可以同时选择一个或多个作业进行终止操作，也可以在作业信息页面对作业进行终止操作。其中，对作业进行终止后作业状态变成 EXIT。

3.5 我的会话

“我的会话”包含“三维图形会话”和“作业会话”两个标签页，其主要功能有查看会话详细信息，对会话进行连接、断开、注销、共享和设置密级。若集成了协同桌面软件，还可以管理三维的会话。未集成协同桌面软件，仅能管理二维桌面（即作业会话）。

此应用中能查看当前用户所启动的会话，还有别人共享给自己的会话。如图所示：



我的会话-三维图形会话页面



我的会话-作业会话页面

在“我的会话”页面可以执行的操作有：查看会话详情以及会话的连接、注

销、断开、共享、设置密级和删除。也可以根据会话名、状态、所有者、操作系统、共享、执行节点、启动模式、启动时间运行时长、密级、空闲时间和会话 ID，对会话列表进行排序和筛选，可以删除启动失败的会话记录。

在会话列表的会话名右侧，还提供了“连接”、“断开”、“注销”和“共享”快捷图标按钮，当鼠标移至任意会话所在行时会显示快捷键。

点击会话名或双击会话所在行，会从右侧滑出会话的详细信息页面，其中展示了会话状态、启动时间、作业号、执行节点、所有者、协议和进程名称，若此会话被共享，则在详细信息页面中还会显示可交互用户和可观察用户。

为了满足可视化的协作交流，在会话列表中，提供了“共享”按钮。在我的会话列表中，选中需要进行共享的会话所在行，点击“共享”按钮后，就可以调出会话共享的页面。

用户可以将自己的任意一个远程 Windows 和 Linux 或者是其他图形桌面系统同时共享给多个用户，供远程用户操作前后处理 CAD 建模或者是协同装配工作。

会话共享提供两种会话共享模式：默认共享后为观察模式，对方只能观看，不能对会话进行任何操作；参与者连接后，会话 owner 可以授权某个用户操作权限（交互模式），此用户可以直接在会话上任意操作，实现基于同一个会话的多人协同仿真计算和应用设计。

特定场景下，在给用户共享会话时，操作权可以传递给特定用户，用以操作会话，而没有操作权的用户仅能查看。

会话共享窗口的已共享列表中，当鼠标滑至用户名左侧的灰色图标上时，会有“设置操作权”的提示信息。



设置操作权的按钮

会话共享页面只能给一个用户设置操作权。点击灰色鼠标图标设置操作权，用户前鼠标图标变为绿色，表明将会话以交互模式共享给了用户。

会话共享窗口中有“操作权自由传递”的选项，勾选此项后，拥有操作权的参与者 A 可将操作权再次传递给其他用户如参与者 B，将操作权传递后，参与者 A 将只能以观察模式连接会话。



操作权传递

在会话共享页面，可以选择单位、研究室和用户进行会话共享，共享设置成功后，共享的用户可以在我的会话列表中查看或连接此会话。在会话共享页面可

以选中已共享的用户名，点击“<”或者选中用户，点击用户右侧“×”按钮来取消此用户加入会话共享，点击清空图标，可以清空加入共享的全部用户。

在我的会话应用中，点击“会话连接方式”按钮可自由选择三维图形会话和作业会话的连接方式，支持浏览器和图形客户端两种会话连接方式。在用户没有安装客户端组件的情况下，打开三维图形会话或上传下载文件时，将会自动切换为浏览器方式。



设置会话的默认连接方式

在会话列表中，如果是三维会话，当鼠标停留在会话名右侧的感叹号图标上时，会出现 tooltips 提示，例如：“执行节点 host1 的剩余内存不足 100M”。如果是带图形的 HPC 作业，只有运行节点的剩余内存达到阈值，才会加载提醒信息。



会话列表中的告警

3.6 数据管理

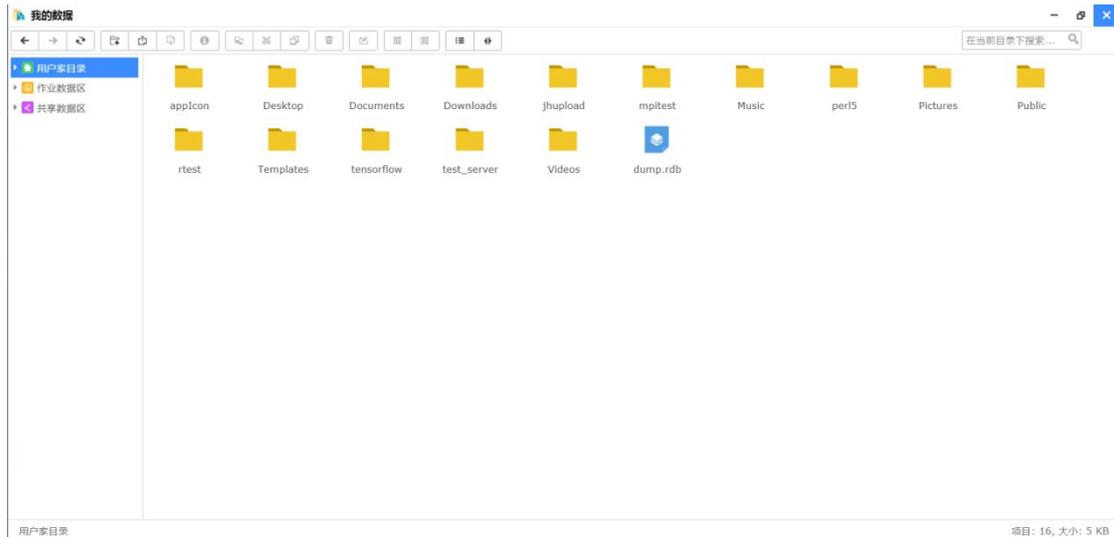
云端数据管理主要是对数据列表、云端家目录、云端工作区中的数据进行管理。对数据的操作主要有查看数据详细信息、删除数据、对数据列表进行排序。对文件的主要操作有新建文件夹、新建文本文件、上传文件、打开文件、下载文件、选择文件、查看文件信息、剪切文件、复制文件、粘贴文件、删除文件、重命名文件、对文件进行前后处理、对文件进行排序。

下面将详细介绍数据管理的主要三项内容：

- 用户家目录
- 作业数据区
- 共享数据区

3.7.1 用户家目录

用户家目录页面主要显示了用户家目录下的文件。如图所示：

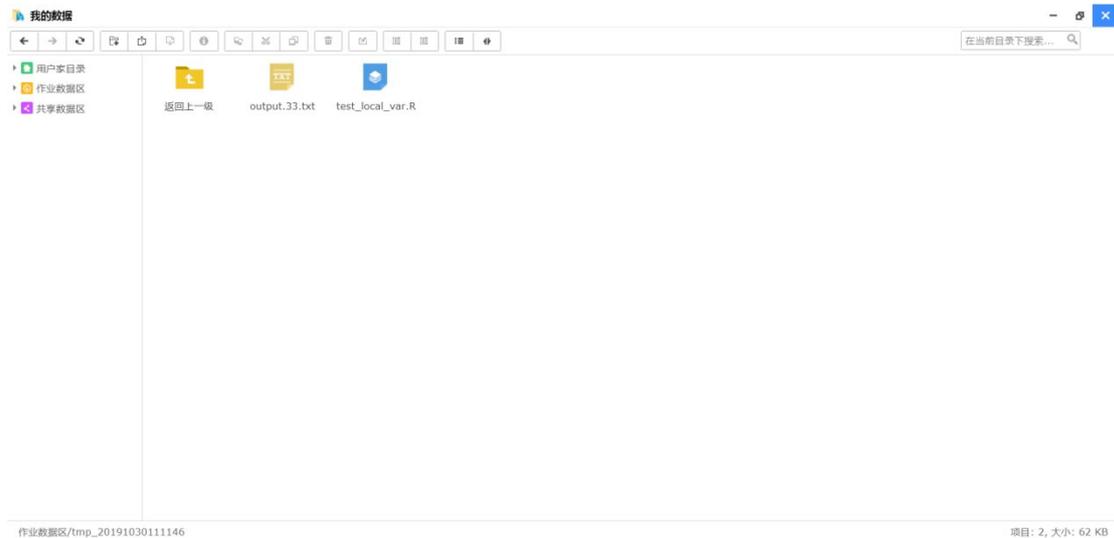


数据列表页面

云端家目录中列出的是服务器端用户 home 目录下（/data/users/CHDHPC/）的文件，可以对文件进行上传、打开、下载、剪切、复制、粘贴、压缩、删除等操作。

3.7.2 作业数据区

作业数据区页面主要显示了列表的形式显示了用户提交作业所产生的数据。如图所示：



作业数据区页面

作业数据区显示的是提交作业产生的数据信息，可以同时选择一个或多个数据进行删除操作，也可以根据作业数据名、项目、创建时间、删除时间对数据列表进行排序。点击作业数据目录名可以进入数据信息页面，在该页面可以查看数

据的详细信息。

3.7.3 共享数据区

为了方便用户之间进行文件共享，作业数据区提供了数据共享的功能，用户可以选择某个用户或某个组进行数据共享。用户登录门户后，点击“我的数据”应用图标，然后切换到共享数据区目录，即可访问共享数据区页面。

可以通过共享组管理，查看、新建、修改、删除共享组等功能。

在共享数据区右键新建共享组。



● 添加共享组

添加共享组时，共享组名称需要手动填写，不能为空。

共享组成员需要从左侧的单位、研究室、用户目录树中选取，可通过点击单位与研究室的向下箭头展开后进行选取操作，支持键盘的 ctrl 和 shift 多选。

支持模糊查找用户功能，缺省状态下可以查找所有用户。

左侧用户目录树中不显示当前创建共享组的用户，创建者默认为共享组的一员。

左侧用户选择完毕后，可以点击“添加”按钮向右侧容器中加入选择的用户；也可以点击选择右侧已加入的用户，将其移除出分享参与者的队

列。

当页面信息填写完毕后，点击页面右下角的“确定”按钮，创建共享组。

- **修改共享组**

创建者对自己建立的共享组具有修改权限，参与者不能修改他人创建的共享组。修改共享组，只能是更改当前组的成员，不能修改共享组的名称。

- **删除共享组**

创建者可以删除自己创建的共享组，参与者不能删除他人创建的共享组。删除操作会删除此共享组的数据库记录以及共享资源和目录。

3.7 注销



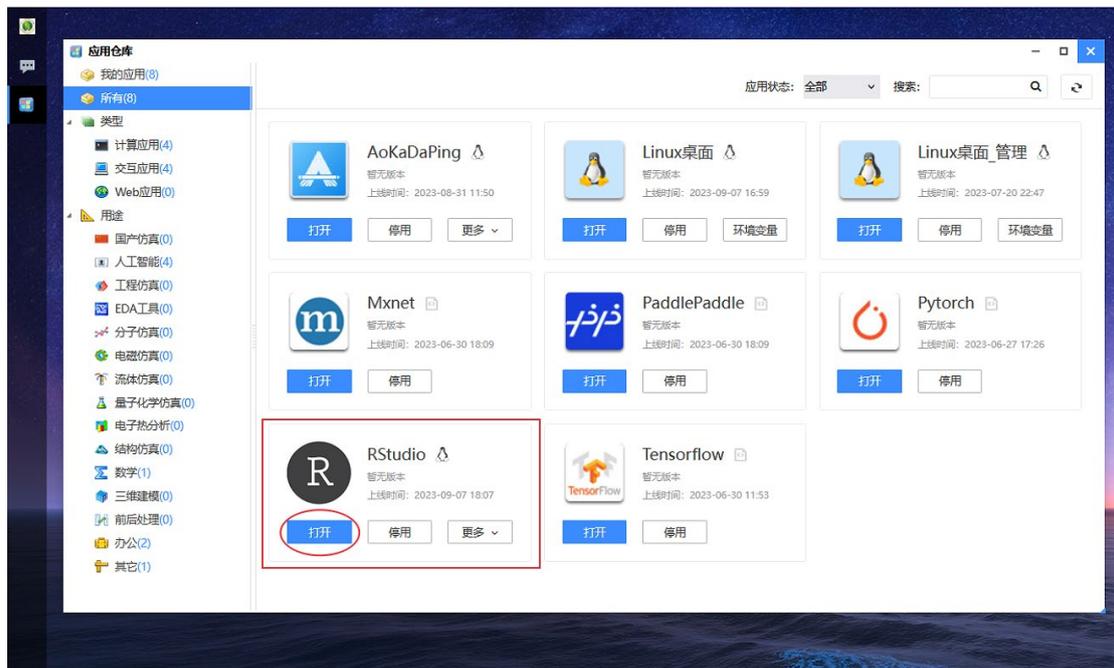
点击左下角的退出按钮可以退出当前登录用户。

3.8 Rstudio 使用

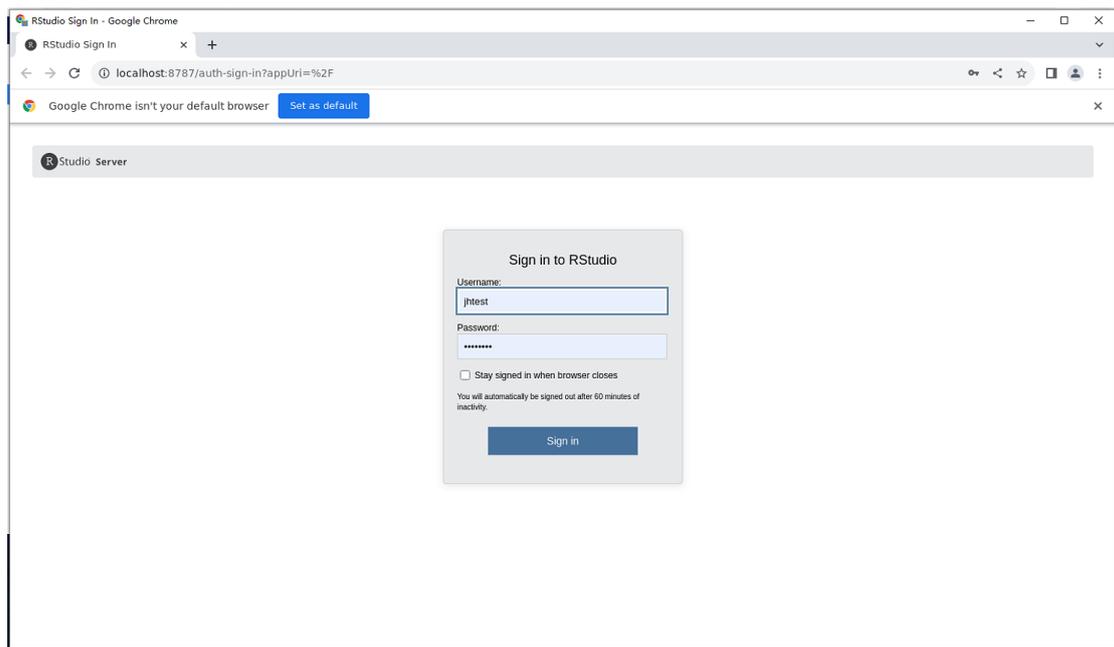
集群门户支持 Rstudio 应用，可以通过网页方便管理或处理 R 数据，具体操作步骤如下：

3.8.1 打开和登录

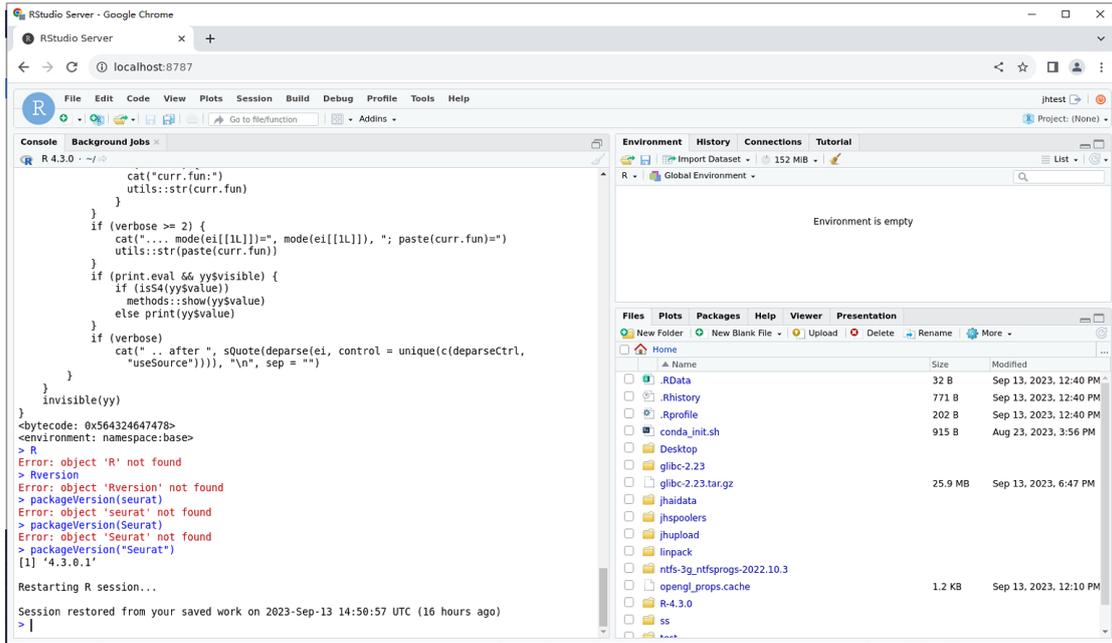
登录门户后打开“应用仓库”，在仓库中打开“RStudio”应用：



之后会弹出谷歌浏览器并默认打开 Rstudio 登录页面，账号和密码与集群中的信息一致：

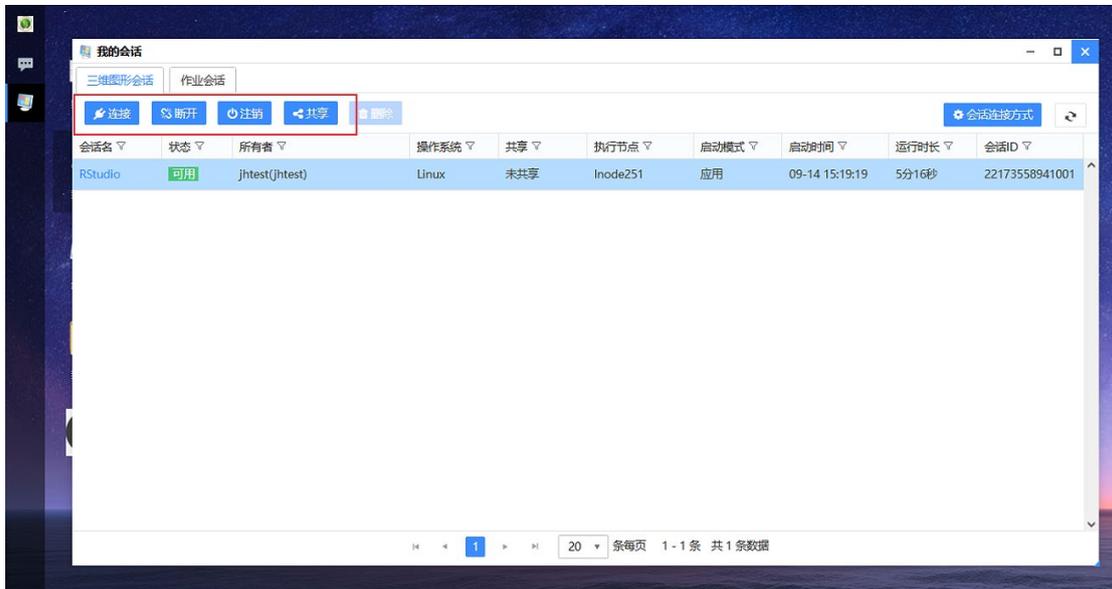


登录后即可使用 Rstudio 了，如下图所示：



3.8.2 会话管理

可在门户中“我的会话”中对已经打开的 Rstudio 会话管理，包含连接、断开、注销和共享操作，如下图：



第四章 注意事项

4.1 支持的浏览器版本

目前支持：IE8、firefox45、chrome50 以上。其他浏览器可以访问，但可能需要修改浏览器的安全配置项，建议使用 firefox。

4.2 上传下载打不开的原因

- 1) 浏览器的版本
- 2) JH 客户端是否安装，版本是否正确。

4.3 提交作业选择 CPU 核数

提交作业时选择的核数，只会决定调度会分配多少核，并不能决定程序能使用多少核，具体应根据自己的程序实际能使用到多少核进行选择，串行程序则选择一个核心数，并行程序则根据实际需要进行选择。