CST Studio Suite 2024

Cluster Integration Guide

3DEXPERIENCE®

Version 2024.0 - 6/28/2023

DASSAULT SYSTEMES | The 3DEXPERIENCE® Company

3DS.com/SIMULIA

Table of contents

1. Introduction

Cluster systems are specialized environments which provide access to High Performance Computing (HPC) resources. They way software is installed and used on such a system is usually very different as compared to a normal workstation. Often, a cluster does not offer a graphical login, software needs to be started in a non-interactive way (batch mode), and the selection of cluster resources reserved for a certain job is usually done by a queuing system. Setting up a cluster environment from scratch is a task which should only be performed by an experienced system administrator and such a setup is beyond the scope of this document. This document is meant to provide some guidelines when it comes to the installation of CST Studio Suite into a properly working cluster environment and, if necessary, helps you when you want to implement the proper integration into a third party queuing system used to organize the workload on the cluster.

Each installation of CST Studio Suite contains a simple batch queuing system called "Distributed Computing" system. This allows you to submit jobs to a simple FIFO queue without further configuration effort. It includes a file transfer mechanism, and provides some monitoring functionality for your cluster. This is a simple lightweight alternative to an external batch queuing system. However, as its functionality is limited you may want to start your simulations using one of the available external job queuing systems. This is especially true if some of the following points apply:

- The hardware is not dedicated to CST simulation jobs only. Other software tools run on the hardware as well and all different jobs should be organized within one job queue.
- Multiple users are using the hardware and each user needs to have their own user account and wants to submit their jobs independently from other users.
- You need advanced management options for your job queue such as prioritization, request special hardware resources for each different simulation etc.

The intent of this document is to demonstrate how CST Studio Suite simulations can be submitted to and started by several mainstream free and commercial job queuing systems. Additionally, it provides some hints on how to configure your queuing system properly such that CST Studio Suite can work seamlessly together with the system. However, this document will not show you how to perform the basic installation and configuration of your job queuing system. Please refer to the manuals provided by the vendor of your queuing system to learn how to install the system correctly for your environment. We assume that you have set up a computational environment, which runs an already configured queuing system. Currently, we have tested the following queuing systems with CST Studio Suite:

1. PBSPro
2. Torque
3. Load Sharing Facility (LSF)
4. OpenLava
5. Oracle Grid Engine (OGE), formerly known as Sun Grid Engine (SGE)
6. Microsoft HPC Job Scheduler
7. SLURM
8. HTCondor

2. Nomenclature

Execution Host	A node on which a job (e.g. a simulation process) is started by a queuing system.
Submit Host	The machine on which the job submission happens.
`command`	Commands you have to enter either on a command prompt (cmd on Windows or a terminal window on Linux) are written using this font.
`<…>`	Within commands the sections you should replace according to your environment are enclosed in "<…>", e.g., `<CST_INSTALLPATH>` should be replaced by the path to the directory where CST Studio Suite is installed.

3. CST installation set

Before you can integrate the CST software in the queuing system running on your cluster you need to install the required CST software components on the execution hosts. If you want to start your simulations in batch mode, you need to install the CST front-end (CST DESIGN ENVIRONMENT) on all machines which may be used as execution hosts. The required component is called “Program Files” on Windows and “Graphical front-end and command line interface” on Linux. Please note that a CST installation can be shared by all execution hosts of a cluster, i.e., you need to install the software just once on a fileserver such that it is visible on all execution hosts (e.g., via NFS). If you want to use such a central installation, you need to ensure that the mountpoints where the CST installation can be accessed is identical on all execution hosts. Please note that you do not need and should not select other components from the available install sets apart from the ones mentioned above.[1] On a cluster system running the Microsoft Windows operating system, additional software components (redistributables) need to be installed in case of a shared installation (these components can be found in the subfolder ClusterUtilities of your CST installation).

In the following sections, it is assumed that the CST software is installed properly on your cluster system.

[1] Please note that additional steps are needed to use MPI on Windows though (installation of MPI service, registration of user credentials). Please refer to the CST online help (section Simulation Acceleration→MPI Computing) to find out how to install MPI on Windows.

4. Queuing system configuration for CST

The following section contains special configuration requirements for the different queuing systems needed to interact properly with CST Studio Suite.

4.1. Configuration of LSF on Windows execution hosts

CST Studio Suite needs to open its user interface during a simulation run. As the LSF system prevents programs from opening graphical user interfaces (GUI) by default, there are some special steps required to allow the CST software to open its GUI on a special desktop provided by LSF. Otherwise, the simulation job will get stuck. The following steps are known to solve this problem:

1. Add the following lines to your `<LSF_INSTALL_DIR>/conf/lsf.conf`:

    ```
    #---------------------------
    # Allows CST GUI to open on virtual desktop[2]
    LSB_LOGON_INTERACTIVE=y
    # Allows empty spaces on the batch job submission command line[3]
    LSB_API_QUOTE_CMD=y
    #---------------------------
    ```

2. Run the following commands:

    ```
    lsadmin reconfig
    badmin mbdrestart
    ```

3. Submit the job with **tssub** instead of **bsub**. The **tssub** command is a wrapper around the **bsub** command and only submits jobs to hosts that have the MSTS resource. It sets the LSB_TSJOB and LSF_LOGON_DESKTOP environment variables (see LSF documentation for more information), which are then transferred to the execution host. This allows CST Studio Suite to start in a virtual desktop environment.
4. Please note that all hosts must have Windows Remote Desktop services installed and enabled. For more information how to configure the Remote Desktop Services for LFS, please refer to LSF documentation[4] *Using IBM Platform LSF on Windows*, chapter 6 *Displaying a Graphical User Interface in LSF with Microsoft Terminal Services*.

[2] See https://support.sas.com/rnd/scalability/platform/PSS10.1/lsf10.1_readme.pdf for details about LSB_LOGON_INTERACTIVE.
[3] See https://www.ibm.com/support/pages/job-command-space-cannot-be-recoganized-when-submitted-bsub for details about LSB_API_QUOTE_CMD.
[4] See https://www.ibm.com/support/knowledgecenter/en/SSWRJV_10.1.0/lsf_windows_using/ms_terminal_services_lsf.html for details.

4.2. Configuration of the Microsoft HPC Scheduler

CST Studio Suite needs a Windows console session on the execution host where the front-end is started. The job scheduler can be configured such that it creates a new Windows console session on an execution host. To enable this feature you need to enter the following two keys into the registry of all potential execution hosts of your cluster[5]:

- `HKLM\SYSTEM\CurrentControlSet\Services\HpcNodeManager\HpcConsoleSupport`
- `HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System\SoftwareSASGeneration`

Both should be created as ”DWORD” with value ”1”. This can be done automatically for all cluster nodes using the following commands

- `clusrun reg add HKLM\SYSTEM\CurrentControlSet\Services\HpcNodeManager /v HpcConsoleSupport /t REG_DWORD /d 1 /f`
- `clusrun reg add HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System /v SoftwareSASGeneration /t REG_DWORD /d 1 /f`

As CST Studio Suite uses all available CPU cores on each MPI node by default please configure the ”Type of resource to request” to ”Node”, (see fig. 1) unless you limit the number of threads of the solver using the ”-numthreads” command line option.

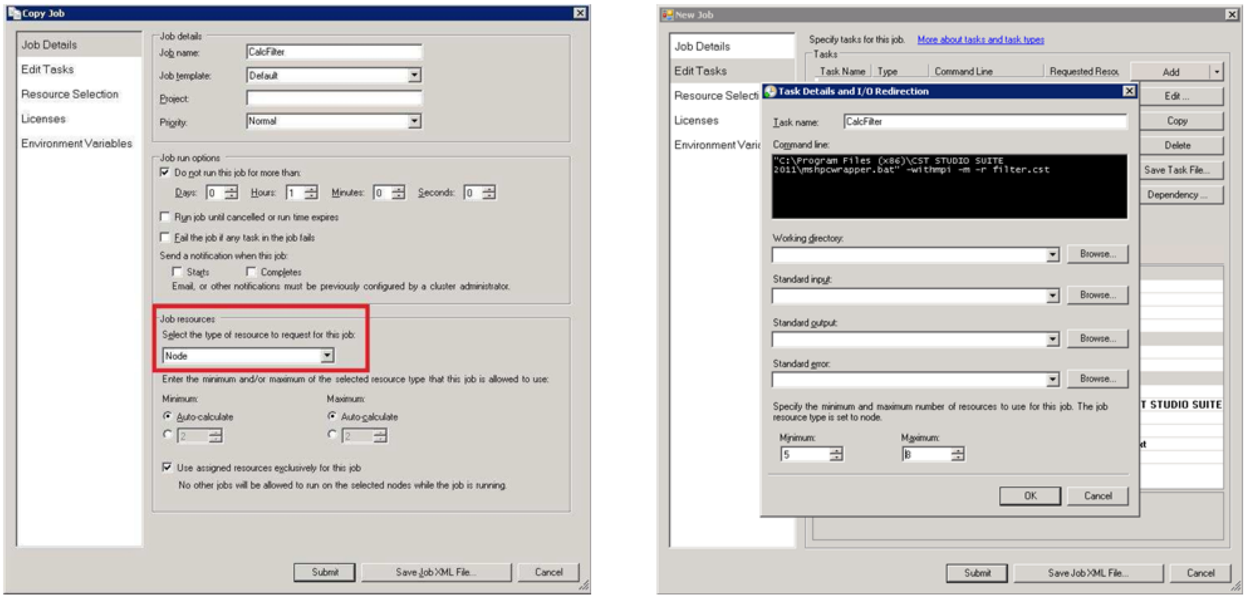

Figure 1: Select the resource type "node".

[5] See http://technet.microsoft.com/en-us/library/gg247477(WS.10).aspx for details.

Additionally, you need to set the environment variable ”HPC_CREATECONSOLE” to ”TRUE” in the environment of your process. This tells Windows to create a console session for the job, which allows CST Studio Suite to open the GUI, which is required for the simulation (see fig. 2).

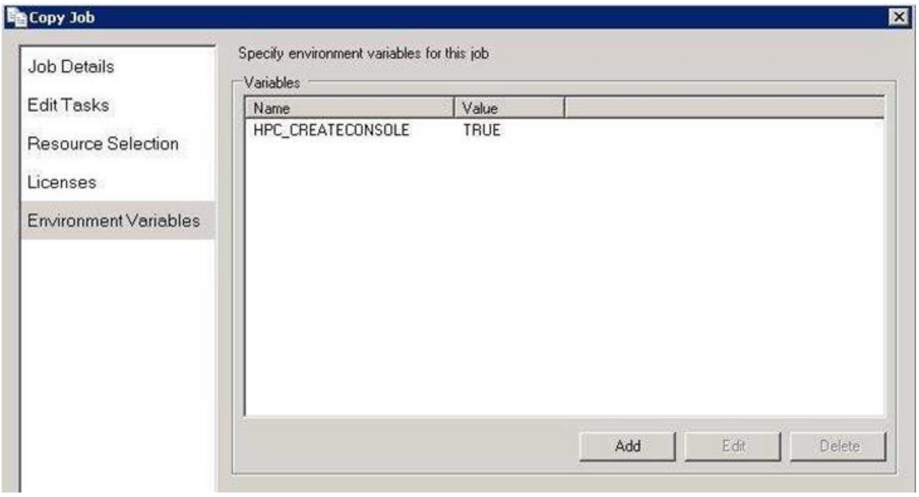

Figure 2: Environment variable, which allows the job to open a GUI in the console session of the execution host.

5. Job submission for CST Studio Suite simulations

CST Studio Suite can be started in batch mode using appropriate command line options to select a solver and acceleration options. The online help documents (General Features → Command Line Options) list all command line options that can be used to configure a batch run of CST Studio Suite.

5.1. Script collection for Linux clusters

For Linux cluster systems a set of convenient scripts, the so-called **Cluster Integration Scripts**, is a part of the standard installation and available in the CST Studio Suite directory: “<CST_INSTALLPATH>/ClusterUtilities/”.

These scripts can be used to provide a simple and robust interface for job submission. It is strongly recommended to use these scripts instead of writing own wrapper scripts which is time consuming and usually error prone. Please read the “README” file included in the archive to find out how the scripts can be adapted to your environment.

The scripts can also be integrated into the cluster user interfaces provided by third party software vendors. A ready-to-use plugin for the EnginFrame portal solution offered by NICE can be provided to CST customers free of charge. For integration into other cluster management solutions please contact us directly.

5.2. Submit from within CST Studio Suite

For both Linux cluster systems and Microsoft HPC Scheduler managed Windows clusters a macro is available allowing easy submission of simulation jobs from the CST Studio Suite front-end to the scheduler system.

The submission macro can be accessed from within the CST Studio Suite “Macros” menu in the “Home” ribbon as shown in Figure 3: Macros → Solver → High Performance Computing → Cluster Utilities.

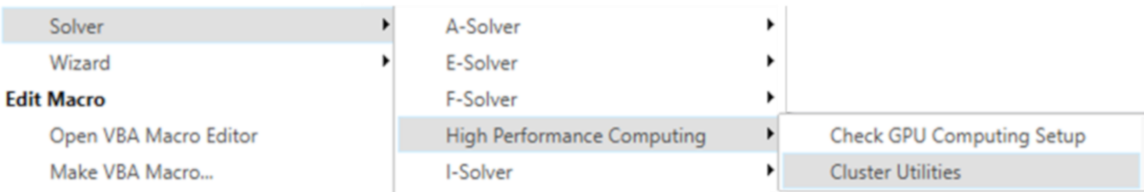

Figure 3: Macro menu with macro package installed.

On the first startup, a configuration window will come up. Please choose the desired scheduler system and, depending on this choice, the cluster settings necessary for job submission. All fields have to be filled in. The configuration dialog box can be accessed in the main submission window as well using the “Cluster Settings…” button.

Depending on the selected scheduler, you will either be asked to choose a queue for Linux cluster systems or a template for Microsoft HPC Scheduler managed clusters. Next, a window allowing the adjustment of the simulation settings will open up. Here the desired solver and acceleration options including the number of GPUs and cluster nodes (in case of DC or MPI) can be selected.

5.2.1. Submission to Microsoft HPC Scheduler

Submission to the Microsoft HPC Scheduler requires an installed Microsoft HPC Pack on the cluster itself as well as on the submit host. In case MPI should be used, the submitting user needs to be registered on all MPI nodes (refer to the online help documents of CST Studio Suite, section: Simulation Acceleration → MPI Computing for more information). **Additionally, the CST project files have to be stored on a network share accessible to all cluster nodes**[6].

First, the dialog box settings for the Microsoft HPC scheduler will be discussed. Below in Figure 4 the cluster settings dialog box is shown.

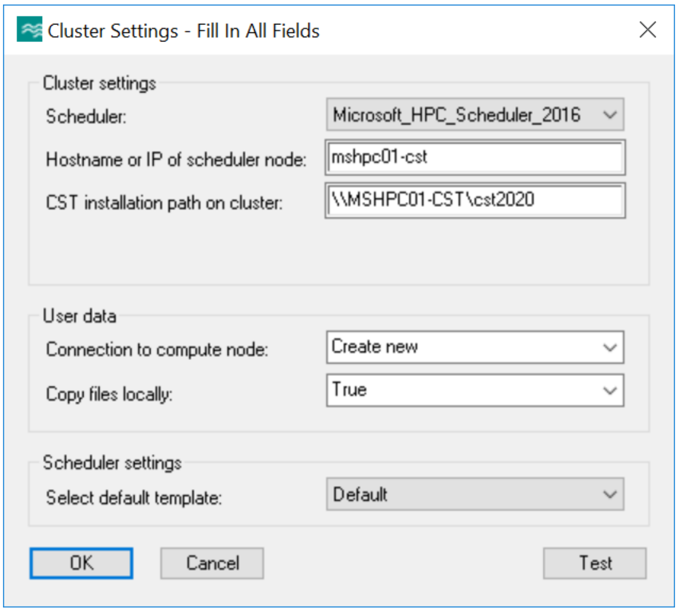

Figure 4: Configuration Wizard for Windows schedulers.

Since CST Studio Suite 2022 SP 1, three versions of Microsoft HPC pack have been supported:

- MS_HPC_Scheduler_2012 (in previous versions: MS_HPC_Scheduler)
- MS_HPC_Scheduler_2016
- MS_HPC_Scheduler_2019

Most of the dialog box settings are self-explanatory and briefly described in Table 1. Only the setting “Connection to compute node” will be discussed in detail in the next paragraph.

[6] In case of a Windows cluster the models need to be stored on a file server accessible by all nodes by the user account specified for submission. In case of a Linux cluster the working directory needs to be located on a shared file system.

Field Name	Explanation/Options	Comment
Scheduler	Linux_Scheduler or Microsoft_HPC_Scheduler	For the Linux cluster system using the script collection or for managed Windows clusters
Hostname or IP of scheduler node	Hostname the scheduler of the cluster system	
CST installation path on cluster	CST installation path on cluster system	For Linux cluster systems please make sure that the script collection is installed in this directory as well (see section 5.2.2).
Working directory	Path to directory where the project data will be stored	Linux only. Commonly, this is the user's home directory.
User name	Username of submit user	Linux only. This user account must exist on all cluster nodes.
Password	Password of submit user	Linux only. Password will be stored in encrypted form.
Connection to compute node	Create new or Attach	Windows only. Choose whether to create a new console if the user is not connected to the simulation host or, in case the user is logged into the simulation host, attach to an existing console session.
Copy files locally	True/False	Windows only. Activate to copy the files to the simulation host before simulation. Files are transferred back to a subfolder of the current project-directory.
Select default queue/template	<Queues as set up on cluster system>	The macro displays the queues or templates available on the cluster.

Table 1: Overview of all cluster settings that have to be filled in.

The setting “Connection to compute node” is not very intuitive and it will be explained in detail.

If the submitting user is connected by Windows remote desktops (RDP) **to all compute nodes** that will be used for the simulation, **“Attach”** has to be selected. Microsoft HPC Pack will then attach to the existing terminal sessions that have been created due to the RDP connection. Internally the environment variable HPC_ATTACHTOSESSION is used. This mode is very helpful to debug and investigate what is happening on the compute nodes.

For **standard usage** and the daily work, the recommendation is to use the mode **“Create new”**. In this case, there **must be no RDP connection** to any of the compute nodes. Otherwise, the simulation will fail. In this mode, Microsoft HPC Pack will create new console sessions to submit the job. Internally the environment variable HPC_CREATECONSOLE is used.

Mixed configurations, where only a few of the needed compute nodes are connected by RDP, are not allowed. For an MPI simulation that means, all nodes must be connected by RDP or none. This configuration is required by the Microsoft HPC Pack and independent of CST Studio Suite.

Our recommendation: Use RDP to establish a connection to the submission host and configure Microsoft HPC Pack in a way, that the submission host cannot be used to run a job. Set “Connection to compute node” to “Create new” and do not connect by RDP to any of the compute nodes.

5.2.2. Submission to Linux clusters

For submission to Linux cluster systems, properly configured cluster integration scripts (see section 5.1.) and a working SSH connection from the submit host to the Linux cluster system are mandatory. The cluster integration scripts are available in the subdirectory ClusterUtilities of the CST Studio Suite installation. Please make sure that the scripts can submit jobs to your queuing system. Adjust the scripts if needed. Additional information can be obtained from the online available MPI Computing Guide: https://updates.cst.com/downloads/MPI_Computing_Guide_2021.pdf.

For the Linux scheduler all setting are self-explanatory and will not be discussed in detail, see Table 1. The simulation data will be temporarily stored in the working directory.

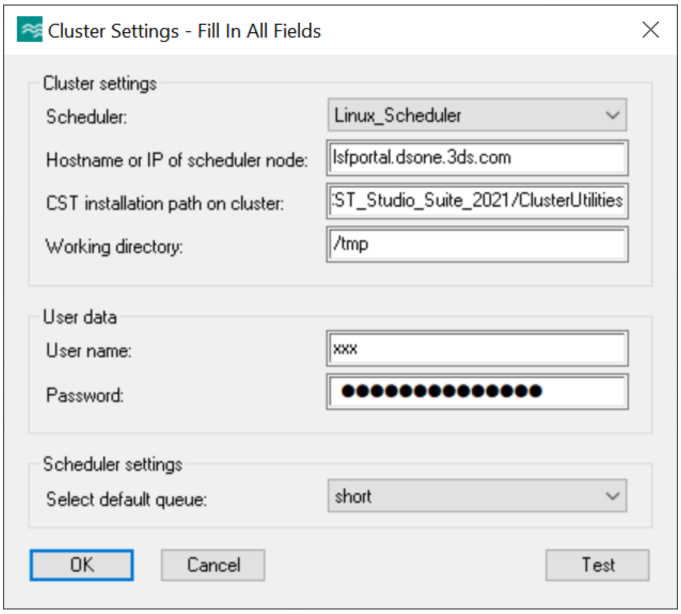

Figure 5: Configuration Wizard for Linux schedulers.

If all settings are entered correctly, the next macro execution will show a different dialog box. For Linux cluster systems, one of the actions (Submit a job, Fetch a job, Cluster settings…) can be chosen.

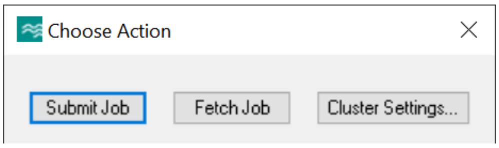

Figure 6: Action dialog box.

The whole setup described above has to be done only once and will be stored in the Windows registry. To be able to change the cluster settings, go back to the Cluster Settings dialog box by clicking on the button Cluster Settings…

In case of Microsoft HPC Scheduler managed clusters, the submission window will be displayed directly. The submission window will let you choose the desired simulation setup. Solver choices made in the CST Studio Suite front-end will be reused. During the macro run, information will be written to the message window of CST Studio Suite. In case of an error in the communication with Linux machines, a command window will open showing the messages from the Linux system.

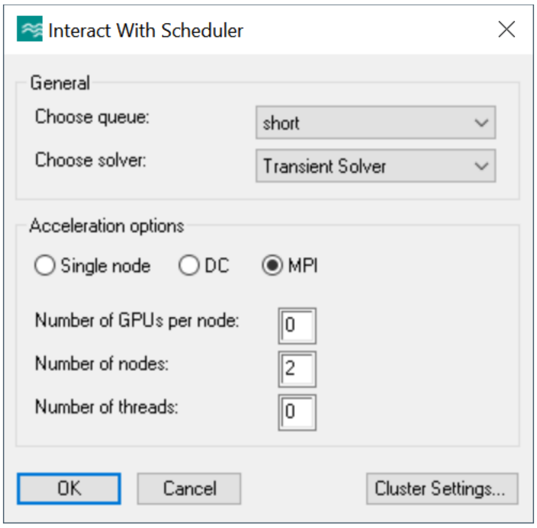

Figure 7: Configuration dialog box for submission to scheduler system.

6. Resource reservation for CST Studio Suite simulations

6.1. Single core vs. full machine/CPU device

Most of the compute intensive parts of CST Studio Suite are efficiently multithreading parallelized, i.e., the software can efficiently use the computational resources of a complete physical machine which typically has many CPU cores. In order to get the best possible performance, it is strongly recommended though to request computing resources (CPUs) which are physically close in the cluster system, i.e., prefer CPU cores, which are part of the same CPU device or the same cluster node before spawning MPI processes onto multiple distributed resources.

The same applies to parallel processes spanning multiple execution hosts using MPI computing: Starting MPI processes on different CPU devices requires an additional license feature called “acceleration token” where the number of required tokens depends on the number of CPU devices involved in the simulation. Because of this licensing scheme the most efficient way (in terms of licensing cost) to run a parallel simulation using MPI is to start one MPI process per CPU device/CPU socket which can then again use multithreading parallelization to make use of all computational resources (CPU cores) found on the CPU socket. This may differ from the licensing scheme of other software vendors who sometimes license their parallelized solvers per CPU core.

6.2. Licensing considerations

As CST Studio Suite is protected by the FlexNet or DSLS licensing system, there are some points you need to be aware of when configuring your queuing system for starting CST simulations as well as when you submit your simulation jobs:

1. If you are using a node locked license, the CST DESIGN ENVIRONMENT must be started on a certain execution host. Namely, the one for which the license was issued (if your license is based on a MAC address) or on the node that has your dongle attached, respectively. The usual way to ensure this is to assign a so-called resource to the node and then request this resource when submitting your job. If you have installed a LAN license, your CST simulation jobs can be started on any node equipped with an installation of CST Studio Suite.

2. As the queuing system cannot know which features your CST license contains, you need to configure the system such that it knows which and how many features are available and, when submitting a job, which and how many of those features your job will require. The typical way to configure this is to define ”consumable resources” describing your CST license. A ”consumable resource” is no longer available as soon as it has been acquired by one job. Depending on your type of license this resource should be assigned at the queue level (in case of a LAN license) or to a node (in case of a node locked license).

The following is an example configuration for the PBSPro queuing system. Let us say you have a LAN license for CST Studio Suite that contains 2 simulation processes (i.e. two simulations at a time may run), 2 T-solver licenses, 1 F-solver license, 1 I-solver license, and 2 Acceleration Tokens. Then you could define a consumable resource for each of these solvers/features at the queue level. The following example configuration shows how the custom resources for the queue could look like. Please consult the manual of your queuing system to learn the details about how to define such custom resources.

```
resources_available.CST_ACCTOKEN = 2
resources_available.CST_MWS_F_SOLVER = 1
resources_available.CST_MWS_I_SOLVER = 1
resources_available.CST_MWS_T_SOLVER = 2
resources_available.CST_SIM_PROCESS = 2
```

Please note that not all queuing systems support consumable resources (e.g. Torque does not support such a setup).

3. If you do not want to implement the license resources into your setup, it is possible to allow just CST DESIGN ENVIRONMENT to wait until it gets the license required to run the simulation. This can be achieved by setting the environment variable `CST_WAIT_FOR_LICENSE` to 1.

7. Preserving intermediate results in terms of a timeout

By default, CST Studio Suite assembles its results after a solver run has been completed. Most often, it is unknown how long a solver run will take *a priori*. As in a typical configuration of a queuing system, you will need to specify the maximum walltime for your job. This may lead to loss of results when the system kills CST Studio Suite while the simulation is still running because the walltime limit is exceeded. Therefore, it is often advisable to start your simulation with the additional command line option "`-dump <n>`". In this case, the solver will save intermediate results for the running job and even if the job is terminated, the results right before the job termination will be available. The parameter `<n>` sets the time interval in minutes for this operation, e.g. "`-dump 5`" will perform this operation every five minutes.

8. Viewing simulation progress

The following section describes different ways how you can check the progress of a running simulation. The methods are listed with growing complexity of the setup.

8.1. Progress file and message file

If the CST front-end is started in batch mode, it automatically writes a progress file. This file can be found in <PROJECT_FOLDER>/Result/progress.log. The <PROJECT_FOLDER> has the same name as your CST file and contains all results related to your project. This file has the following format:

```
Header:string:<Operation>
Text:string:<Progress Information>
Progress:int:<Status Bar>
```

The contained information is related to the progress bar that you can see in CST DESIGN ENVIRONMENT in the following way:

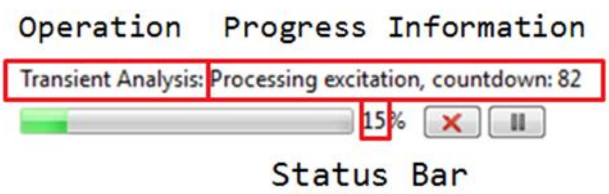

Figure 8: Progress information of CST Studio Suite.

For the example in the picture shown above the content of the file would be:

```
Header:string:Transient Analysis:
Text:string:Processing excitation, countdown: 82
Progress:int:15
```

Additionally, a message file is written during the simulation. This file is stored as <PROJECT_FOLDER>/Result/output.txt and contains all information displayed in the message window of the CST front-end.

The script collection for Linux clusters as mentioned in section 5.1 contains a tool to access conveniently the files mentioned above and to display their content on a terminal.

8.2. Accessing the CST front-end of the running simulation

A CST simulation is always controlled by a graphical front-end (CST DESIGN ENVIRONMENT). However, in a cluster environment it is a non-trivial task to get access to this graphical front-end to display the simulation progress and intermediate results. Whether this can work at all depends on the configuration of your environment and on the operating system you are using.

8.2.1. Microsoft HPC Scheduler

On a cluster system controlled by the Microsoft HPC scheduler, the CST Studio Suite front-end opens in the console session of the execution host. Access to this session is limited to system administrators. Unless your user account has administrative permissions, it is impossible to login to this session. If you belong to the group of administrators, you may simply establish a RDP connection to the console session of the execution host and you will be able to see the CST front-end running there.

8.2.2. LSF Scheduler

On a cluster system controlled by LSF, the CST Studio Suite front-end can be opened in a new RDP session on the execution host. In this case, a user may connect to his job using the `tspeek` tool, which is provided with the LSF installation.

8.2.3. Linux operating system

When running in batchmode on a Linux system the CST front-end is started on a virtual frame buffer by default (Xvfb). The virtual frame buffer is a non-interactive X-server and accessing it as an interactive display is not recommended.

However, using the script collection mentioned in section 5.1, it is possible to start the CST graphical front-end on an interactive X-server (e.g., Xvnc) which can then be accessed either using a tool from the script collection (please refer to the README file of the script collection to learn more) or other mechanisms available to access an interactive X-server.

If an already running X-server is available in the context of the job which can be used to display the graphical user interface of CST you can specify the option "`--use-external-display`" as command line option of the job submit script contained in the script collection (see section 5.1). Then, CST Studio Suite will try to use the X-server specified by the `DISPLAY` environment variable to open its graphical user interface.

8.3. Third party portal solutions

CST Studio Suite can be integrated into portal solutions commonly used for job submission and job management on cluster systems. A ready-to-use plugin for the EnginFrame portal solution offered by NICE can be provided to CST customers free of charge. For integration in other cluster management solutions please contact us directly.

9. Viewing simulation results

After a simulation run has been completed, the results need to be accessed. Sometimes, the amount of data created by the simulation prevents copying the data out of the cluster environment. In this case accessing the results requires remote graphical login to a machine, which has high-speed access to the storage holding the result data of the simulation. The simulation data itself is not transferred over a potentially slow LAN/WAN network connection but only the picture rendered on the remote side is transferred using a remote access protocol such as VNC, RDP, etc. (see Figure 9).

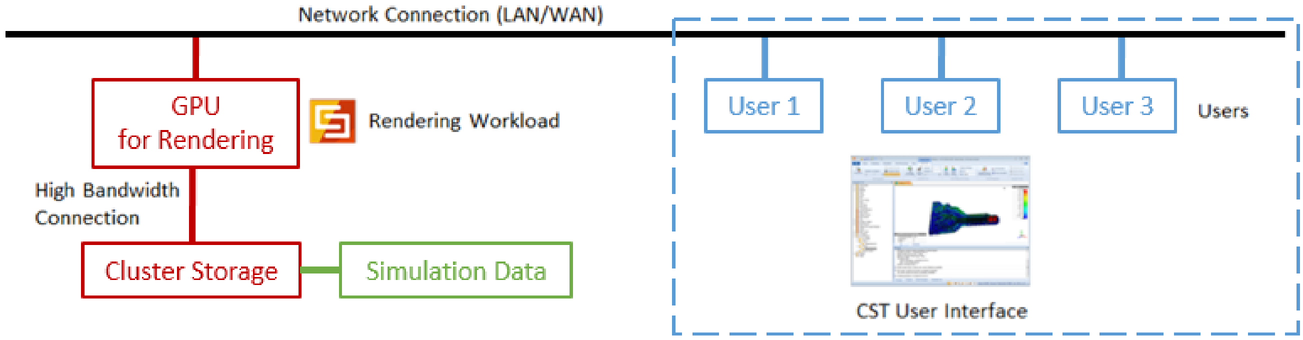

Figure 9: Remote access to simulation data using GPU accelerated rendering and remote access protocol. The picture shows the principle setup and the involved system components.

In order for such a setup to work properly and with reasonable performance, it is essential to understand which system components influence the performance.

1. **Simulation Data Access.** The machine running CST Studio Suite to display the simulation data needs to access the data, which usually resides on a cluster storage / file server. The bandwidth of the network link to this storage system affects the performance of CST Studio Suite access and loading of the data.
2. **Rendering Workload.** When CST Studio Suite displays the simulation data, it needs to render a 3D picture using OpenGL (in case field data is displayed). Although OpenGL workloads can be processed by a CPU, a GPU is optimized for this kind of workload and is therefore easily an order of magnitude faster as compared to CPU based rendering.
3. **Network Connection to Client.** The data of the rendered picture needs to be transferred to the client system of the user and, depending of the reliability and bandwidth of the LAN/WAN network this can influence the performance and user experience significantly.

CST Studio Suite has been tested to be compatible with different graphical remote access protocols that can be used in the described scenario:

- NICE Desktop Cloud Visualization (DCV) (Windows and Linux)
- Windows Remote Desktop (RDP) Protocol (Windows)
- VNC with VirtualGL (Linux)

A CST plugin to the NICE EnginFrame portal solution is available on request free of charge.

10. Pitfalls and troubleshooting

10.1. Installation components

When you submit a job to a scheduling system, be aware of the fact that the job might start on an arbitrary node of your cluster. The CST Studio Suite front-end must be available on this node in order to be able to start the simulation. Please refer to the explanations in section 3.

10.2. Simulation gets stuck on Windows execution hosts

CST Studio Suite will try to open its graphical user interface (GUI) on the execution host. Most of the queueing systems prevent this by default and therefore the program will get stuck at this point. Please refer to the manual of your queueing system to find out how to allow a program to show a GUI. Please refer to section 4 to find out how to configure LSF and Microsoft HPC scheduler such that CST Studio Suite is allowed to open its GUI on a virtual desktop. Alternatively, the GUI can be suppressed to allow no user interaction during the simulation.

10.3. File locking on parallel file systems

Parallel file systems like Lustre provide a POSIX interface, which disables file locking by default. However, all CST Studio Suite processes require file locking to be enabled.

In order to enable file locking for POSIX, add the option **flock** to the mount command in the configuration file **/etc/fstab**. Just changing the mount option is not enough, it is necessary to remount or reboot each of the cluster nodes afterwards.