GeoDict

High Performance Computing

User Guide

GeoDict release 2024

Published: April 15, 2024

GeoDict

https://doi.org/10.30423/userguide.geodict

Citation:
Sven Linden, Andreas Wiegmann, Jürgen Becker, Aaron Widera, Barbara Planas. GeoDict 2024 User Guide. GeoDict – High Performance Computing handbook. Math2Market GmbH, Germany, doi.org/10.30423/userguide.geodict

The handbooks in the User Guide series of Math2Market GmbH can be obtained from:

Math2Market GmbH
Richard-Wagner-Strasse 1
67655 Kaiserslautern
Germany

Phone: +49 631 205 605 0
Fax: +49 631 205 605 99
Email: info@math2market.de
Web: www.math2market.de

Introduction to High Performance Computing

Three-dimensional computer tomography images can consist of 2000^3 or even more voxels. GeoDict allows to compute flow, thermal, electrical or mechanical properties on such large datasets. In these computations, multiple floating point values per grid cell must be computed and stored. As 2000^3 already means 8 billion grid cells, such computations need strong computational resources. These simulations require:

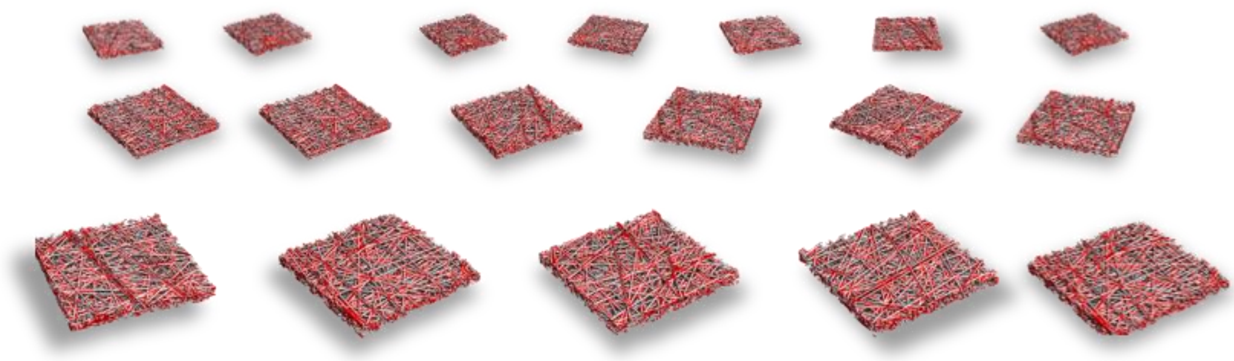
1. a large amount of memory (RAM),
2. a high number of cores to be computed in a short amount of time and
3. a large hard drive to store the simulation results.

These requirements can be met by:

1. large shared memory machines, or
2. computer clusters.

Often, such computational resources are not available locally, and there is a need for on-demand availability of computer resources in the cloud.

Large computational resources are also required in parameter studies. In parameter studies, a single computation may not require extraordinary resources and may run on a standard computer. However, there is often the need to vary many parameters. As an example, optimize a nonwoven filter material by creating various 3D structure models in FiberGeo and determine the filtration efficiency and pressure drop with FilterDict for each of them.



Assuming that a FilterDict simulation on such a 3D model takes 2 days to accomplish, and that 100 different parameter sets are to be tested, it would take 200 days on a single computer to perform the whole parameter study. However, by using computational resources in the cloud, it is possible to start all 100 simulations at once and finish the complete study in just 2 days.

Shared memory Machines

Large shared memory machines are single computers with a large amount of RAM, many CPU cores and one or more large hard drives. These machines have the advantage that they are relatively cheap to purchase and easy to maintain. They allow to perform and visualize simulations on very large structures. Shared memory machines are the best option for most of the application cases. Recommendations about the hardware configurations for different use cases can be found at:

https://www.geodict.com/service-support/technical-support/system-requirements.html

The operating system on these machines may be Windows or Linux. A GeoDict installation on such a shared memory machine may be used by several users. You can use a remote desktop connection to connect to this computer, or alternative use GeoDict's built-in job queue system to submit compute jobs from a local desktop to this computer.

Be aware that multiple licenses are required if several users want to use GeoDict at the same time or if a single user wants to use GeoDict on several computers at the same time (local desktop and remote server).

Remote Login

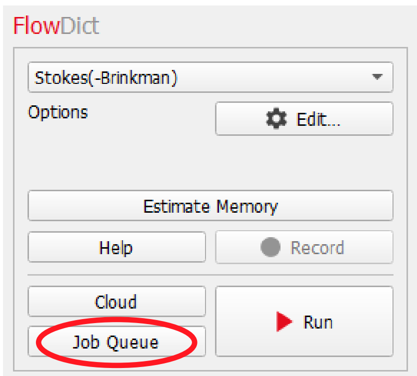
Several users can login to this computer at the same time with a remote desktop connection. The most used combinations are:

1. Local Windows to remote Windows can be done with Microsoft's built-in *Remote Desktop Connection* tool.
2. Local Linux to remote Linux can be done with the built-in *SSH* command line tool or TurboVNC which is a graphical remote desktop connection tool.
3. Local Windows to remote Linux can also be done with TurboVNC.

The graphical remote desktop connection with TurboVNC is a very convenient way to perform simulations on shared memory machines, but TurboVNC is not a built-in tool. On page 11, we describe the installation and usage of this tool.

Job Queue

Instead of a remote login to the compute server, it is also possible to work with GeoDict on your local machine and send compute jobs to the remote computer. For this, you can install and start a GeoDict job queue server on the remote computer. This server can then collect and execute the compute jobs from several users one after the other.



This is a new feature in GeoDict 2024 and requires a separate license. It is explained in detail on page 15.

Computer Clusters

Sometimes structures are very large (e.g., 4096^3) and even large shared memory machines do not have enough memory to perform a flow or mechanical simulation. In this situation a computer cluster with many compute nodes is required.

A computer cluster typically consists of compute nodes with same hardware configuration, and they are communicating with each other over a very fast interconnection (e.g. InfiniBand). With a floating license, it is possible to use GeoDict with its GUI on such clusters but typically a simulation script is submitted into a job queue management system.

A job queue scheduler is a computer application for controlling unattended background program execution of jobs. This is commonly called batch scheduling, as execution of non-interactive jobs is often called batch processing. Two commonly used job schedulers are:

1. Portable Batch System (PBS)
2. Slurm Workload Manager (SLURM)

Both schedulers can be used to submit and perform GeoDict simulation jobs.

PBS is the name of computer software that performs job scheduling. Its primary task is to allocate computational tasks, i.e., batch jobs, among the available computing resources. It is often used in conjunction with UNIX cluster environments.

The Slurm Workload Manager is a free and open-source job scheduler for Linux and Unix-like kernels, used by many of the world's supercomputers and computer clusters.

On page 23, we describe how to submit jobs on both systems.

Cloud Services

Cloud computing describes a model that provides shared computer resources as a service on demand: for example, in the form of servers, data storage or applications.

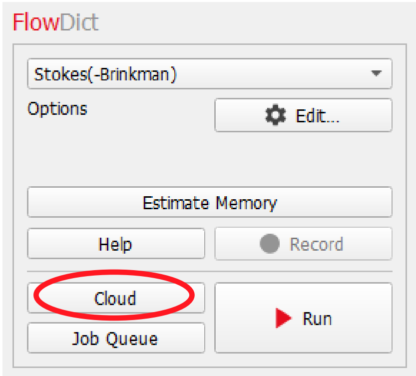
Usually, it is possible to lease large shared memory machines or cluster-like environments. These shared memory machines can also be used as virtual machines that allow remote desktop connections. The prices depend on the hardware configuration. Cloud computing can be a good alternative to purchasing an own computer system depending on the total usage per year. Especially for peak times it can make sense to supplement the computer resources by cloud computing.

To use GeoDict in a cloud environment of the user’s choice, a GeoDict cloud floating license is required. The floating license server must be set up such that it is accessible from within the cloud environment of the user. In this scenario, the user must purchase cloud resources directly from a cloud service provider, and install and set up GeoDict in the cloud himself.

To simplify cloud access, Math2Market offers two cloud solutions directly, the **GeoDict Cloud** and the **Math2Market Cloud**. In these setups, the user does not need to install GeoDict in the cloud, and access to cloud resources is provided by Math2Market.

GeoDict Cloud

GeoDict 2024 gives users the choice to run computations locally or in the cloud. Besides the **Run** button which starts the computation on the local computer, the **Cloud** button allows to submit the simulation into the cloud.



In 2020, Math2Market started to offer cloud solutions to customers together with KaleidoSim. This cloud solution frees the user from the constraints of limited local resources.

The GeoDict cloud offers a simple web framework that allows users to create parameter studies, run the simulations there, and retrieve the data again for detailed analysis locally.

GPU compute instances are offered as well, so that the AI capabilities of GeoDict (e.g., GeoDict-AI, FiberFind-AI, ImportGeo-VOL) can also be fully utilized.

This solution is described in detail on page 27.

Math2Market Cloud

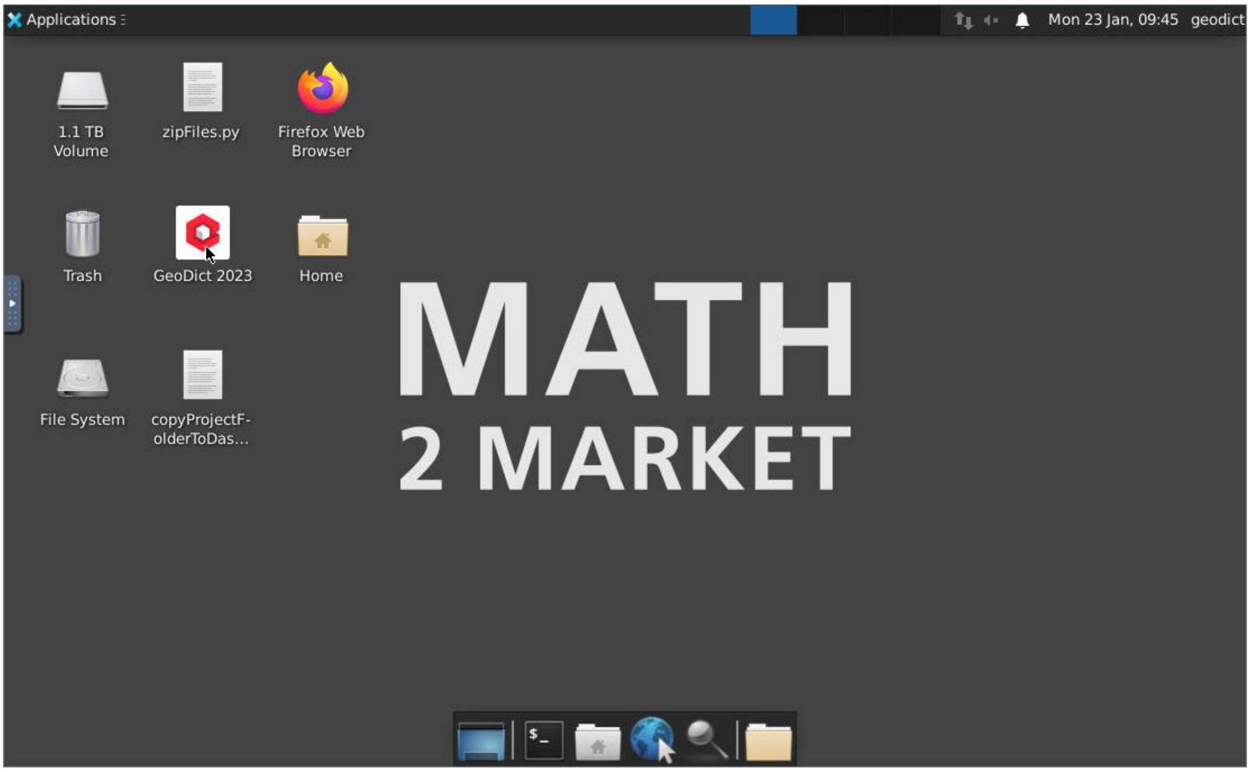
The Math2Market Cloud solution offers the full experience of GeoDict without any installation or hardware requirements. Only a web browser and an internet connection are needed to use GeoDict.

Though your web browser, you will be able to log into your account and access your individual Math2Market Cloud Dashboard. On this dashboard, you may start a virtual machine in the Math2Market Cloud, where GeoDict is already installed and ready-to-use.

A virtual machine (VM) is a virtual workstation in the Math2Market Cloud that is simply created with a few clicks through your Math2Market Cloud dashboard.

Connection to the VM occurs via this dashboard and then, the desktop of the VM is reached without leaving your browser. The VM desktop has the look and feel of a real workstation: simply click on the GeoDict icon found on the desktop to start a GeoDict instance on your VM.

This solution is described in detail on page 37.

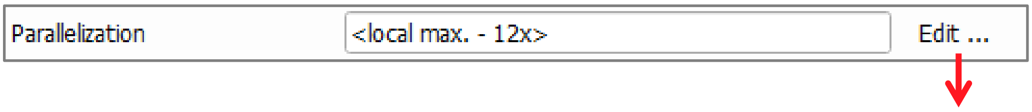

Applications
Mon 23 Jan, 09:45 geodict
1.1 TB Volume
zipFiles.py
Firefox Web Browser
Trash
GeoDict 2023
Home
File System
copyProjectF-olderToDas...
MATH
2 MARKET

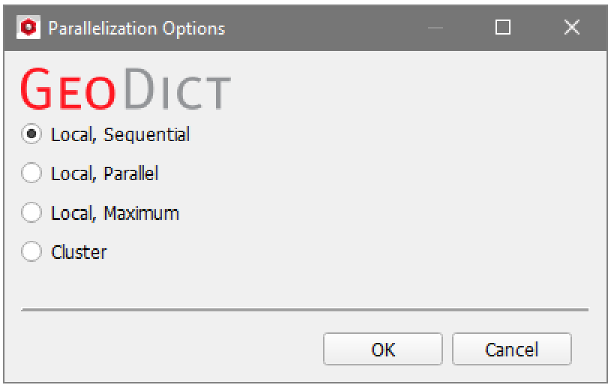
Parallelization and Configuration of MPI

The solvers in GeoDict can make use of powerful hardware by parallelization of simulations to decrease the runtime.

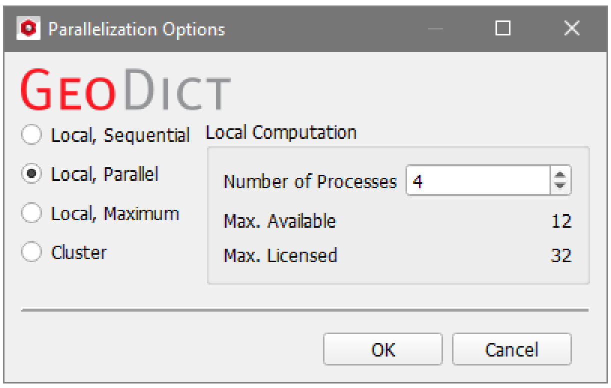
Parallelization Options

The parallelization can be configured in the parallelization dialog. This dialog is available in the solver tabs of all **Dict** modules:

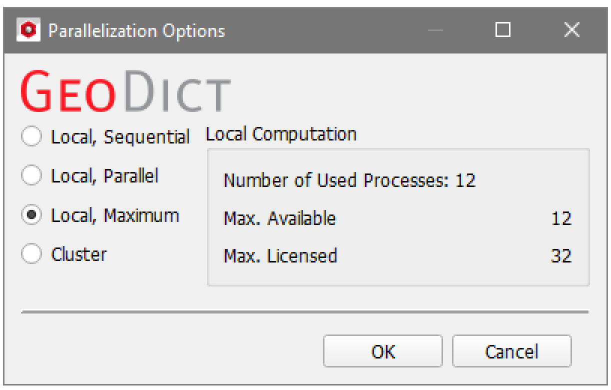




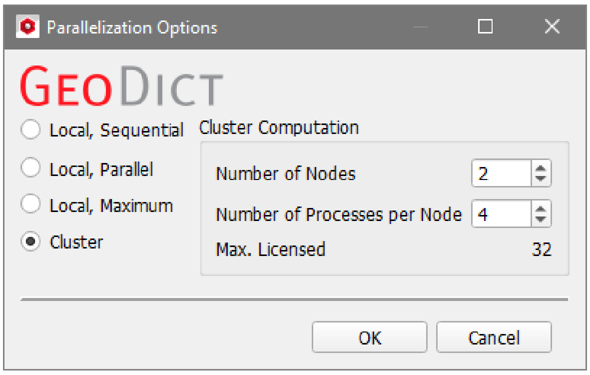
When **Local, Sequential** is selected, no further parameters are needed, and the solver runs sequential without parallelization. This option can be useful for very small structures.



When **Local, Parallel** is selected, the **Number of Processes** can be entered. Then the maximum number of available processors and the maximum number of licensed parallel processes is shown in the dialog. The solver runs parallel with the specified number of processes / threads.



When **Local, Maximum** is selected, the maximum number of available and licensed processors / threads is used for the simulation. This is the default option and in most cases the best choice.



The choice of **Cluster** is for users of Linux clusters. Then the number of compute nodes and the number of processes per node can be entered. The solver runs parallel and distributes the simulation over different compute nodes.

Not all solvers support all parallelization options, as shown in the following table:

Solver	Parallelization Option			
	Local, Sequential	Local, Parallel	Local, Maximum	Cluster
EJ solver	✓	✓	✓	✓
SimpleFFT solver	✓	✓	✓	✓
LIR solver	✓	✓	✓	✗
FeelMath solver	✓	✓	✓	✓
Particle tracker	✓	✓	✓	✓
BEST solver	✓	✓	✓	✗

The parallelization of the solvers is done with three technical methods:

1. Local-MPI parallelization,
2. Distributed-MPI parallelization, and
3. Local-Thread parallelization.

MPI stands for Message Passing Interface and is a standardized and portable message-passing standard designed by a group of researchers from academia and industry to function on a wide variety of parallel computing architectures. Solvers (e.g. EJ or SimpleFFT) that use MPI are started multiple times as different instances. Each instance performs a simulation on a sub-volume of the whole structure. MPI is used to send data of intermediate results between the running instances. Local-MPI parallelization works within the same computer while Distributed-MPI works across different computers. When Local-MPI is used then the sending of data can be done very efficiently because its running on the same machine while Distributed-MPI has to use Ethernet or fast interconnection like InfiniBand to send data between different computers.

The LIR solver and BEST solver use Local-Thread parallelization. These solvers are started only once per simulation and data does not have to be sent between instances because there is only one. But these solvers cannot use multiple compute nodes of a cluster to distribute the simulation data.

The following table shows the support of both parallelization methods:

Solver	Parallelization method	
	MPI Parallel	Thread Parallel
EJ solver	✓	✗
SimpleFFT solver	✓	✗
LIR solver	✗	✓
FeelMath solver	✓	✓
Particle tracker	✓	✓
BEST solver	✗	✓

The **Cluster** parallelization requires that the solver supports the MPI parallelization method. The **Local, Parallel** parallelization can be done with MPI parallelization or Thread parallelization.

An important difference between thread and MPI parallelization is that thread parallelization does not need any special installation procedure. It is available on all operating systems and hardware architectures. MPI parallelization requires the installation of an MPI software package.

Installation of MPI

GeoDict MPI parallel solvers support three MPI software packages:

1. MPICH-3.2 for Linux (https://www.mpich.org/),
2. OpenMPI-1.10.7 for Linux (https://www.open-mpi.org/), and
3. Microsoft MPI for Windows.

The Microsoft MPI is automatically installed during the GeoDict installation. MPICH3 or OpenMPI are often available on Linux system but sometimes it has to be manually installed by the user.

Installation of MPI under Linux is possible using the command line or you can trigger the installation from the GeoDict GUI.

To install MPI from the command line open a terminal and change the current directory into the GeoDict installation folder. The installation folder contains a shell script with the name *setupMPI.sh*. Execute this script with the command:

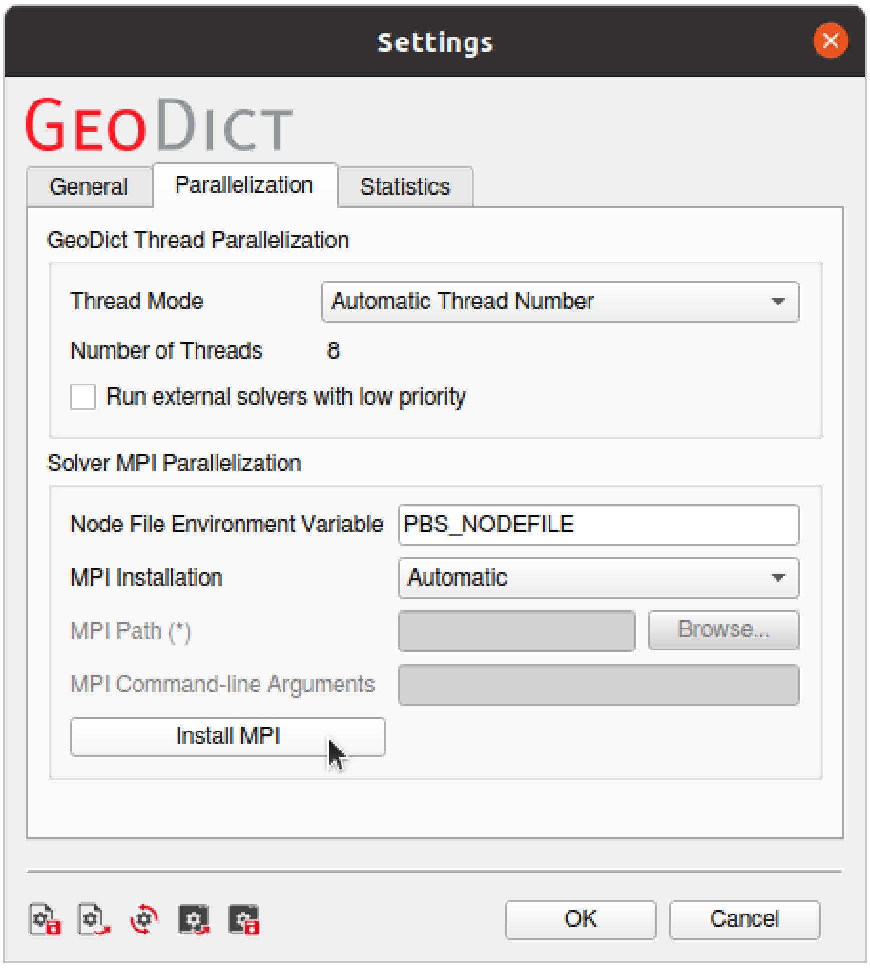
```
./setupMPI.sh
```

The script compiles MPICH-3.2 and OpenMPI-1.10.7 and installs them in the GeoDict installation folder in the sub-folder *MPI*. Root permission is not required to perform this installation procedure, but the compiled MPI packages are available for the current GeoDict installation only. It is also possible to make the MPI packages available for the whole system. This can be done by using the command line argument *root*, i.e.

```
sudo ./setupMPI.sh root
```

Here, root permission is required and the MPI packages are installed in */usr/local/*. GeoDict then automatically detects und uses MPI for parallelization after installation.

It is also possible to trigger the script ./setupMPI.sh from GeoDict's user interface, and in this case MPI will be installed locally (the root version is not available from the GUI). For this, choose **Settings** → **Settings** from the main menu of GeoDict. To install, click on **Install MPI** in the **Solver MPI Parallelization** panel.

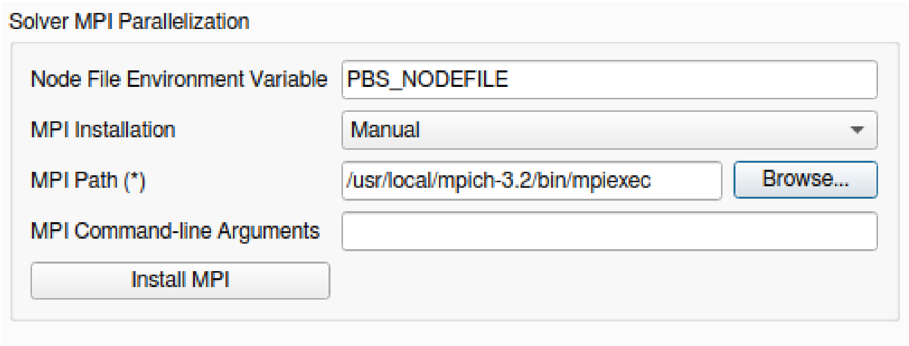


The compilation and installation of MPI will take approximately 15 to 30 minutes. After the installation, check if the OpenMPI and MPICH3. executables have been created, as the script will not return an error message if one of the installations fails.

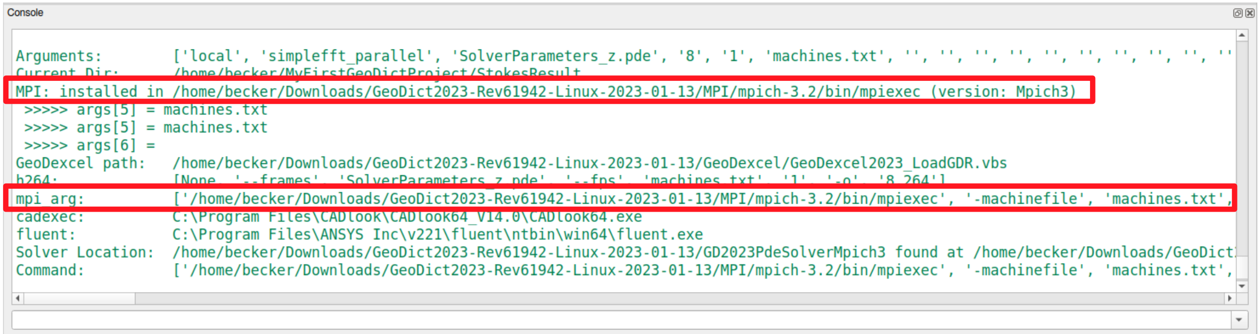
Switch between different MPI packages

MPICH-3.2 is used by default in Linux but sometimes OpenMPI-1.10.7 might be a better choice in computer cluster environments. Many clusters have InfiniBand interconnection between the compute nodes. Unfortunately, MPICH-3.2 cannot use this interconnection natively and therefore the scaling behavior might not be ideal for more than four compute nodes. OpenMPI-1.10.7 can use InfiniBand interconnections natively and provides a good scaling behavior beyond four compute nodes.

The choice of the used MPI package can be changed by setting **MPI Installation** to **Manual** and browsing to the corresponding **mpiexec** executable of the selected MPI version.



If set as above, GeoDict solvers will use MPICH-3.2 installed in /usr/local for parallelization. It is possible to check if the desired MPI package is used by inspection of the command line output of GeoDict when an MPI parallel solver is started. For MPICH-3.2 the command line output should look similar to:



It is possible to add additional command line arguments to the MPI call by entering them in the MPI Command-line Arguments box in the Settings dialog. These additional command line arguments may be used to configure proper usage of the InfiniBand adapter or to show more debug information in the command line output. As an example for MPICH-3.2, the usage of the Internet Protocol over InfiniBand (IPoIB) feature can be configured with the command line argument: "-iface ib0".

Graphical remote desktop connection to Linux computer

A graphical remote desktop connection to a remote Linux computer can be used to run GeoDict on it. A GeoDict installation has to be available on the remote Linux computer, but not necessarily on the local Windows computer. Since GeoDict completely runs on the Linux computer all benefits from the Linux computer capabilities in each step, e.g., structure generation, property prediction, visualization and so on are available.

Follow these steps to use the remote desktop for GeoDict. Steps 1 to 3 are to set it up, and carried out only once. Steps 4 to 6 are done every time remote computations are performed:

1. Install TurboVNC on the remote Linux computer. TurboVNC is used as X proxy and video server. Further information of TurboVNC can be found here:

   http://www.turbovnc.org/
2. Install and set up VirtualGL on the remote Linux computer. VirtualGL is used for hardware OpenGL support. Further information of VirtualGL can be found here:

   http://www.virtualgl.org/
3. Install and set up TurboVNC on the local Windows computer.
4. Start TurboVNC server on the remote Linux computer.
5. Connect the Windows computer to the remote Linux computer.
6. Start and run GeoDict on the remote Linux computer.

Install TurboVNC on the remote Linux computer

The following description is valid for Ubuntu, and it might be slightly different for other Linux operating systems.

1. Download the appropriate TurboVNC binary package for your system from

   http://sourceforge.net/projects/turbovnc/files/2.2.6/

   e.g.

   https://sourceforge.net/projects/turbovnc/files/2.2.6/turbovnc_2.2.6_amd64.deb
2. Install TurboVNC by switching (cd) to the directory where you downloaded TurboVNC and issuing the following command:

   ```
   sudo dpkg -i turbovnc*.deb
   ```

Install VirtualGL on the remote Linux computer

The following description is valid for Ubuntu running LightDM, and might be slightly different for other Linux operating systems.

1. Download the appropriate VirtualGL binary package for your system from:

   https://sourceforge.net/projects/virtualgl/files/2.6.5/

e.g.

https://sourceforge.net/projects/virtualgl/files/2.6.5/virtualgl_2.6.5_amd64.deb

2. Install VirtualGL by switching (cd) to the directory where you downloaded VirtualGL and issuing the following command:

   sudo dpkg -i virtualgl*.deb

3. Configure the server:
   a. Shut down the x server as follows (or restart computer later):

      sudo /etc/init.d/lightdm stop

   b. Run the following:

      sudo /opt/VirtualGL/bin/vglserver_config

   c. Select option 1 to configure the server.
   d. Choose ***no*** for restriction.
   e. Choose ***yes*** for disabling XTEST.
   f. Start the x server again as follows (or restart computer):

      sudo /etc/init.d/lightdm start

4. In order to get the Nvidia driver loading for remote computers, without having a monitor plugged in, open /etc/X11/xorg.conf with a text editor and add the option "ConnectedMonitor" "CRT" to the section "Device" just below the Driver "nvidia" line.

```
Section "Device"
    Identifier        "Device0"
    Driver            "nvidia"
    ConnectedMonitor  "CRT"
    VendorName        "NVIDIA Corporation"
EndSection
```

   In some linux distributions, the file /etc/X11/xorg.conf does not exist by default anymore. In such cases, please check where such an option should be added in your linux distribution.

Install TurboVNC on the local Windows computer

Download and install TurboVNC-2.2.6-x64.exe from:

https://sourceforge.net/projects/turbovnc/files/2.2.6/

Further information can be found here:

https://cdn.rawgit.com/TurboVNC/turbovnc/master/doc/index.html

Follow the instructions that are shown in the installation wizard (click “next” button).

START TURBOVNC SERVER ON THE LINUX COMPUTER

To set up the VNC server:

1. Open a ssh connection to the remote Linux computer, for example with **PuTTY**.
2. Enter: */opt/TurboVNC/bin/vncserver* or for non-default windows size use the parameter *–geometry* <width>x<height>,

For example, enter:

/opt/TurboVNC/bin/vncserver –geometry 1900x950

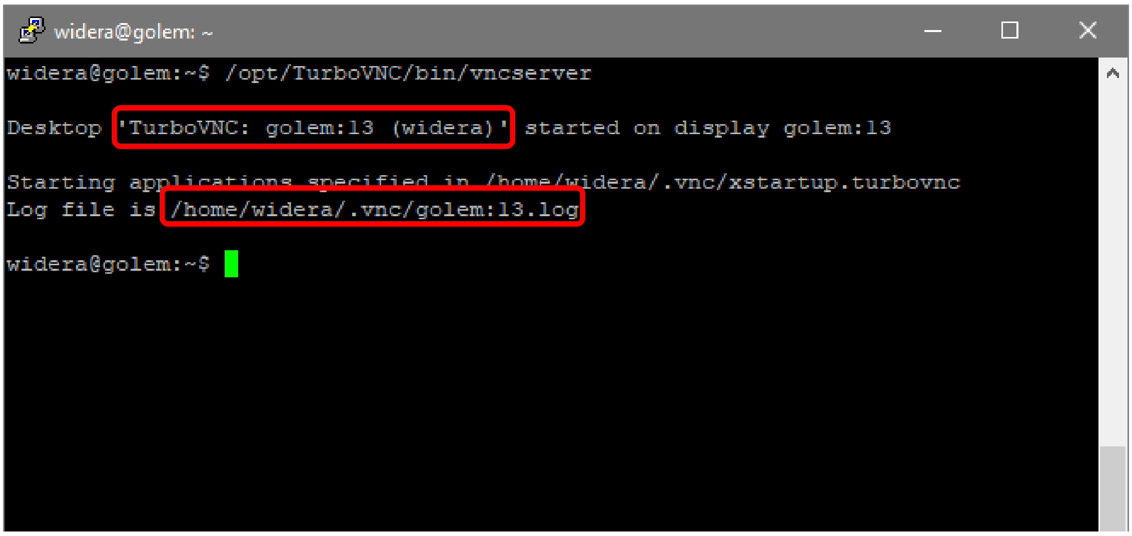
a. A VNC password is requested when started for the first time.
   - Enter a VNC password.
   - Verify the password.

b. Passwords can be changed later through:

   /opt/Turbo*VNC*/bin/vncpasswd.

c. The connection information is returned, e.g.

Desktop 'TurboVNC: golem:13 (widera)'. The session number (here: *13*) is needed later. The IP address of this host is 192.168.1.39.



3. The ssh connection can be closed now.

This server stays open as long as you do not log-out the virtual session. A virtual session can be stopped by entering:

/opt/TurboVNC/bin/vncserver -kill :"your session number".

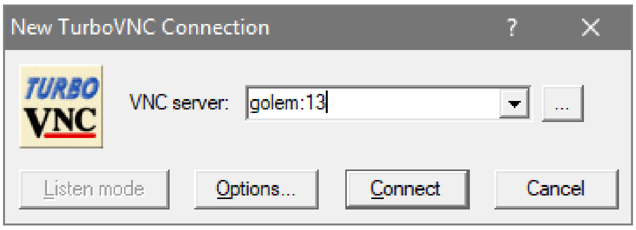
The active VNC servers can be listed by entering:

/opt/TurboVNC/bin/vncserver –list.

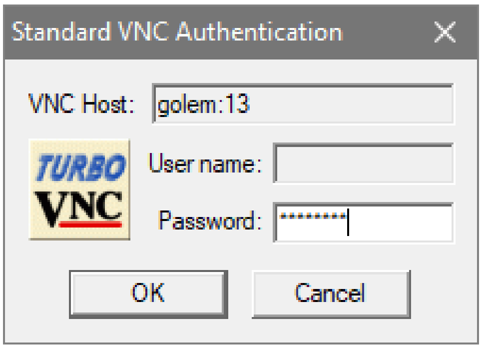
Connect Windows to a remote Linux computer

As long as a virtual session exists (i.e., you have followed these steps and have not logged-out or killed the session), the connection to the remote Linux computer is established as follows:

1. Start the TurboVNC Viewer.
2. Enter the session server, either by its hostname or IP address followed by the session number, e.g., 192.168.1.39:13, and click **Connect**.



3. Enter your VNC password and click **OK**.



A virtual desktop of the remote Linux computer opens.

Start and run GeoDict on the remote Linux computer

VirtualGL is used to redirect rendering to the remote graphics hardware, meaning that GeoDict's graphical user interface is available in the Linux computer.

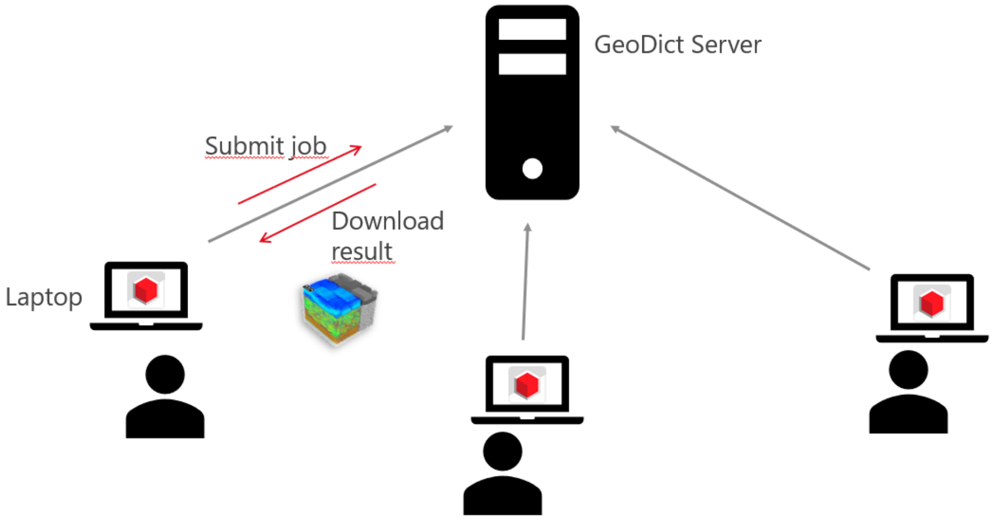
To start GeoDict, open a terminal and enter:

vglrun ./YourGeoDictPath/geodict2024

The started GeoDict session on the remote Linux computer is maintained until the user logs out or kills the VNC session. Otherwise, the user can close the remote desktop dialog box (by clicking on the x in the upper right corner), turn-off the local Windows computer, go home, and restart the remote session later, finding that the started GeoDict jobs are still running.

Job Queue Server

If multiple users want to use GeoDict on a powerful server at the same time, it is advisable to run GeoDict's builtin job queue server on this machine. The server can then collect jobs from several users. It queues the submitted jobs and executes them one after the other. This makes sure that the simulations do not compete for CPU cores, available RAM and GeoDict licenses.

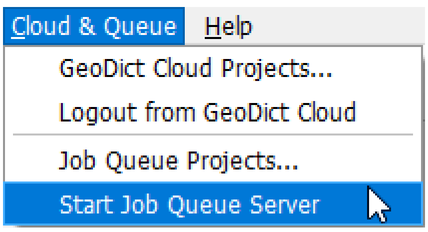


To run the job queue server, a GeoDict-Server license is required. The job queue server starts each submitted job in a separate GeoDict instance (similar to the execution of a macro in the command line) For this, it requires a GeoDict-Base license and licenses for the currently used module and processes. Users that want to submit jobs require another GeoDict license for their local machine.

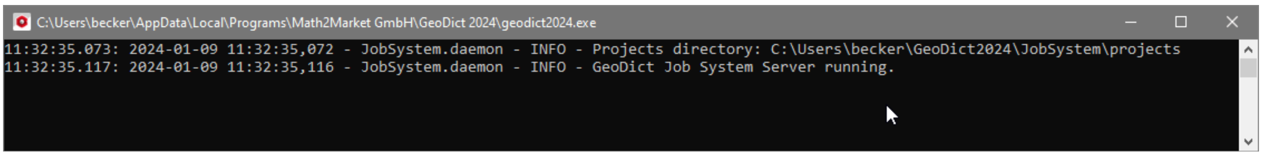
Attention: You must have at least Service Pack 1 of GeoDict 2024 installed to use this feature, both on the server machine and on all client machines.

Start Job Queue Server

A job queue server can be started by any user who has purchased a GeoDict server license. Administrator privileges are not required.



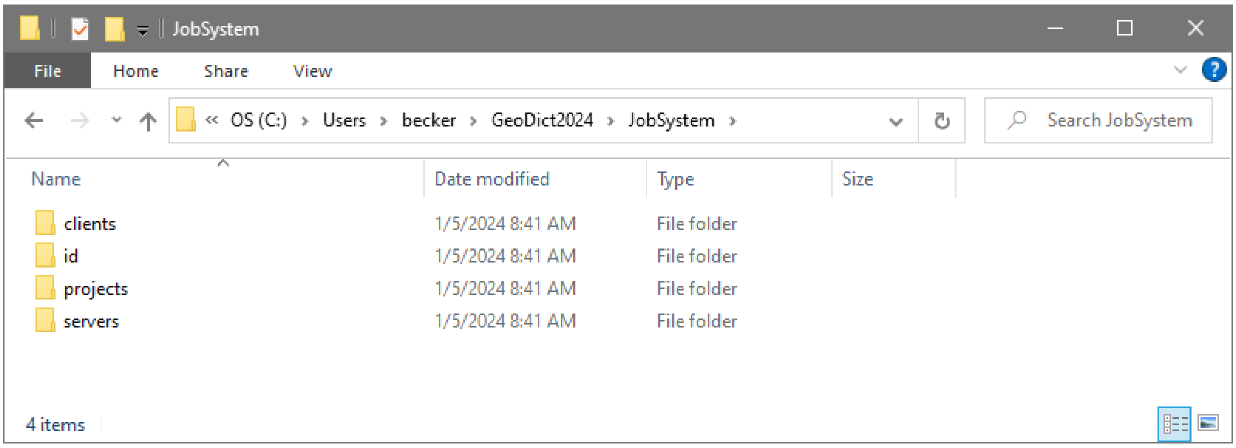
In the main menu, simply click on **Start Job Queue Server**. Then, a terminal window opens in which the server runs.



Alternatively, it is also possible to start the job queue server by starting GeoDict from the command line with the option: `--job-server=<project folder>`

The started job queue server continues to run after GeoDict is closed. Closing the terminal window terminates the job queue server.

The server stores data in the user's GeoDict setting folder, which (for a user named *becker*) is C:\Users*becker*\GeoDict2024\JobSystem\ in Windows or /home/*becker*/.geodict2024/JobSystem/ in Linux. The JobSystem folder contains four subfolders:



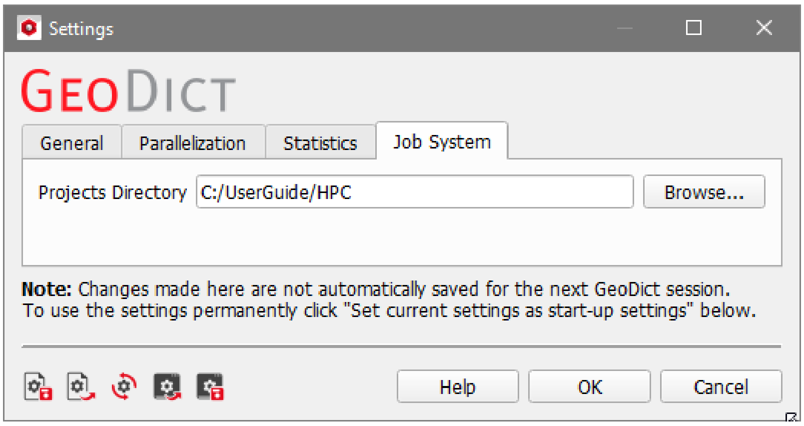
The **id** folder contains two files: *username*@*hostname*_cert.pem and *username*@*hostname*_key.pem, which are used to identify this host. (where *username* is the login name of the user and *hostname* is the name of the computer). GeoDict versions prior to 2024 SP1 used a different naming scheme for the certificates and are therefore incompatible to the current version.

The **servers** folder contains the *username*@*hostname*_cert.pem files of all hosts that can act as a job system server for this client.

The **clients** folder contains the *username*@*hostname*_cert.pem files of all hosts that can act as a client for this server.

The **projects** folder is the default location for all data and computational results of a running job queue server.

Thus, a huge amount of data may be stored in the projects folder. You may therefore select another location for this folder. In the main menu, select **Settings** → **Settings**, and set a different location on the **Job System** tab. Be aware that you have to do this before the job queue server is started, otherwise the default location is used. If the job queue server is started from the command line, you must always specify a location in the option --job-server=<project folder>, there is no default location.



The four subfolders should be created automatically by GeoDict. If this is not the case, and the JobSystem folder is empty, it is not possible to start or use a job queue server.

In that case, create the subfolders `id`, `projects`, `clients` and `servers` manually and restart GeoDict. This resolves “No such file or directory” or “Specified projects directory does not exist” error messages that appear upon startup of the server.

Enable Remote Access

If you have simply clicked on **Start Job Queue Server**, the server will only be accessible locally. You can only send jobs to the job queue server from the same host. It is not possible to send jobs from another host to this server.

Typically, you would want to access the server also from another computer (e.g. your laptop). To enable this, you have to do some preparations before starting the job queue server:

1. Collect the *`username@hostname`*`_cert.pem` files from the `JobSystem/id/` folder of each client and copy them into the `JobSystem/clients/` folder on the server.
2. Copy the *`username@hostname`*`_cert.pem` file from the `JobSystem/id/` folder of the server to the `JobSystem/servers/` folder of each client.
3. Start the GeoDict job queue server on the server machine.

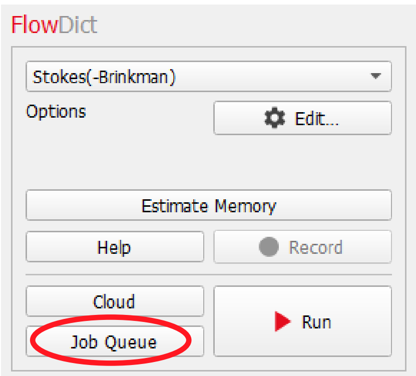
Now, you will be able to submit jobs to the queue from every client computer.

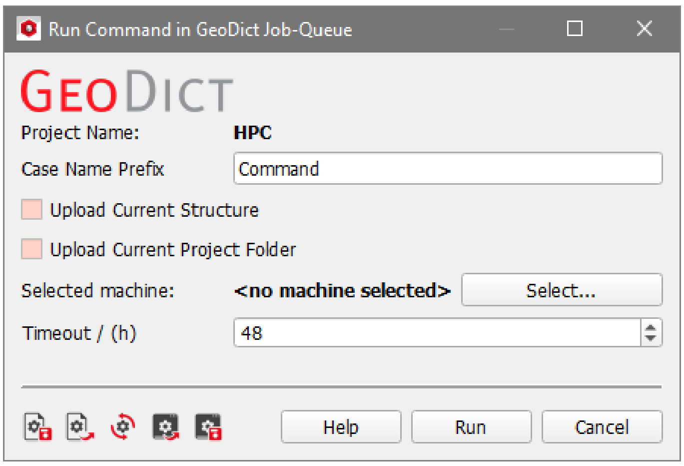
It is also possible to add client certificates to a running job queue server by adding the certificates to the servers `JobSystem/clients/` folder.

Submit Jobs

Submitting a single command

Instead of starting a simulation directly, you can submit it from the client computer into the job queue. Simply click the **Job Queue** button instead of the **Run** button to open the **Run Command in GeoDict Job-Queue** dialog

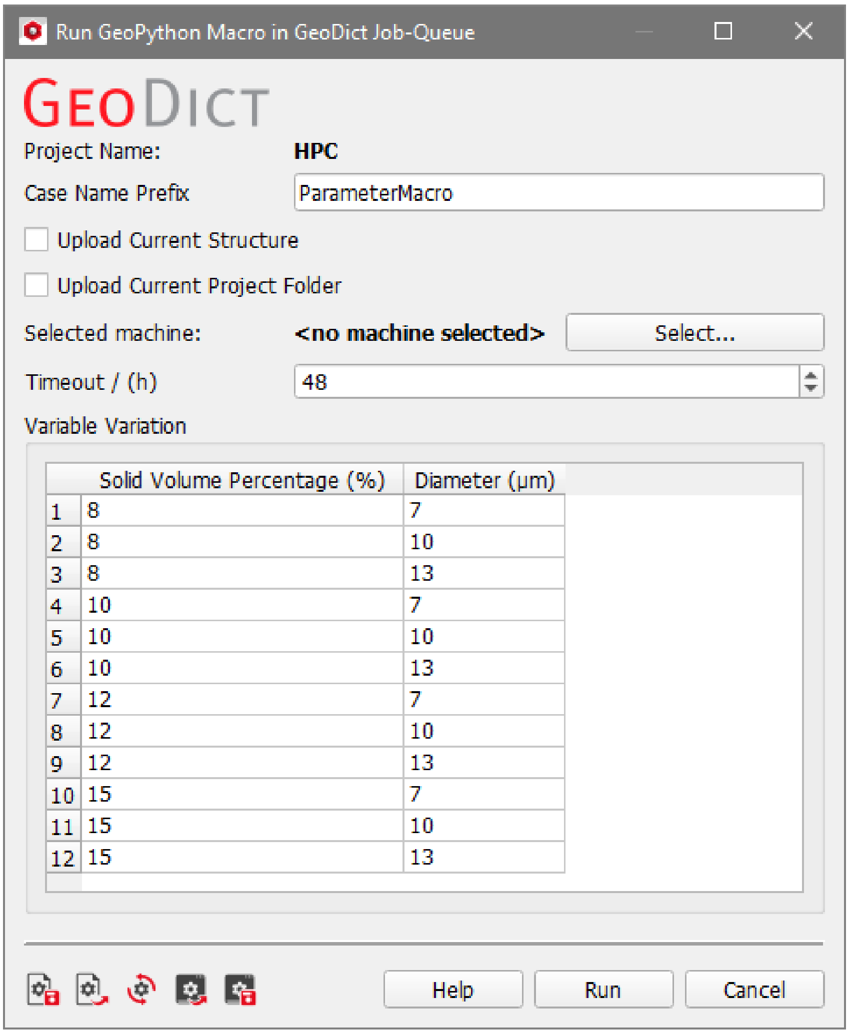




Submitting a macro

Similarly, instead of executing a macro directly, you can submit it into the job queue, too. In the Macro Execution Control dialog, click on **Run in Job Queue** instead on **Run**.

This opens the **Run GeoPython Macro in GeoDict Job-Queue** dialog.



Select Project Name and Case Name

The **Project Name** is the name of the current Project Folder, e.g. if the project folder is `C:\UserGuide\HPC\` the **Project Name** is HPC.

Select a **Case Name Prefix** used to identify this simulation.

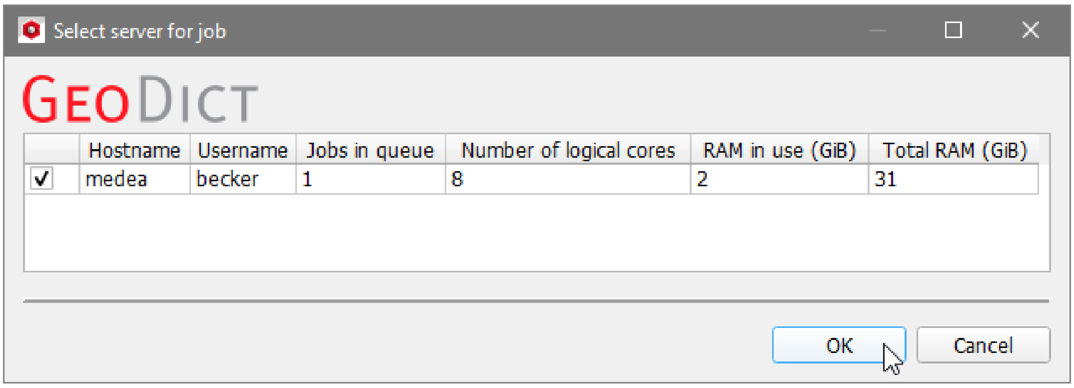
Upload Current Structure and Upload Current Project Folder

If you submit a simulation directly form the module, both check boxes are filled out already. For structure generators (Geo modules), typically no upload is needed. For analysis and prediction (Dict modules) an upload of the structure is required.

If you submit a macro to the job queue server, you have to select manually if data must be uploaded. Check **Upload Current Structure** if the macro requires the current structure to run. Check **Upload Current Project Folder**, if the macro requires data from the current project folder to run.

Selected machine

Click on **Select...** to open a dialog:



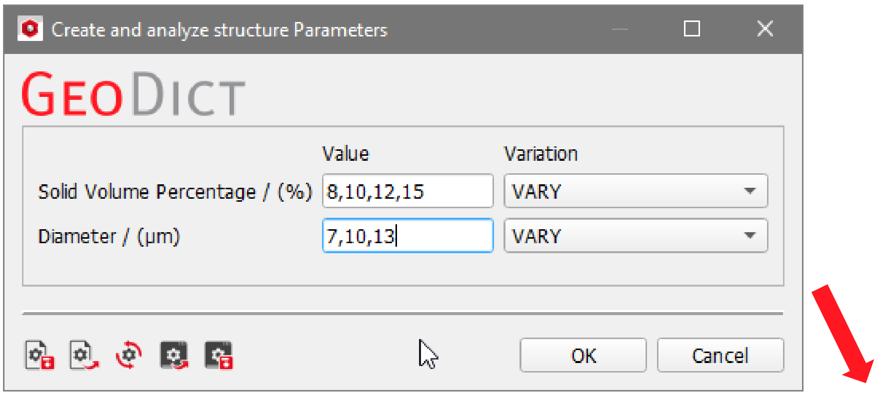
In the dialog, all possible job queue servers are shown. Be aware that the server only shows up if it is running, the server certificate file is present on the client computer and the client certificate file is present on the server (see the Enable Remote Access section above).

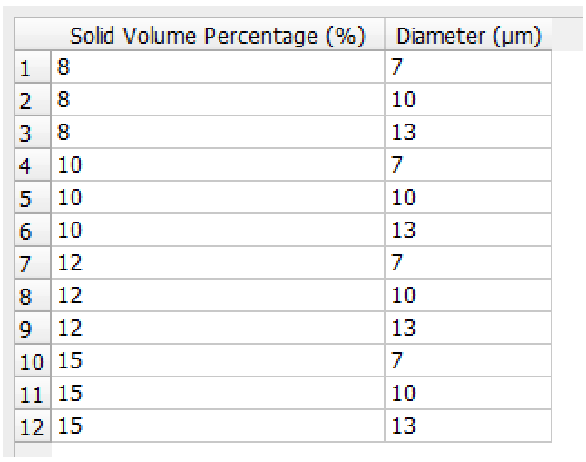
Timeout

Set a timeout to stop a simulation automatically when the given timeout is reached. You can set the timeout to a maximum of 720 h (30 days) per case. If set to 0, each simulation runs until it is finished, terminated or crashed.

Variable Variation

In case of parameter macros, this table gives an overview over all parameter variations that will be executed.

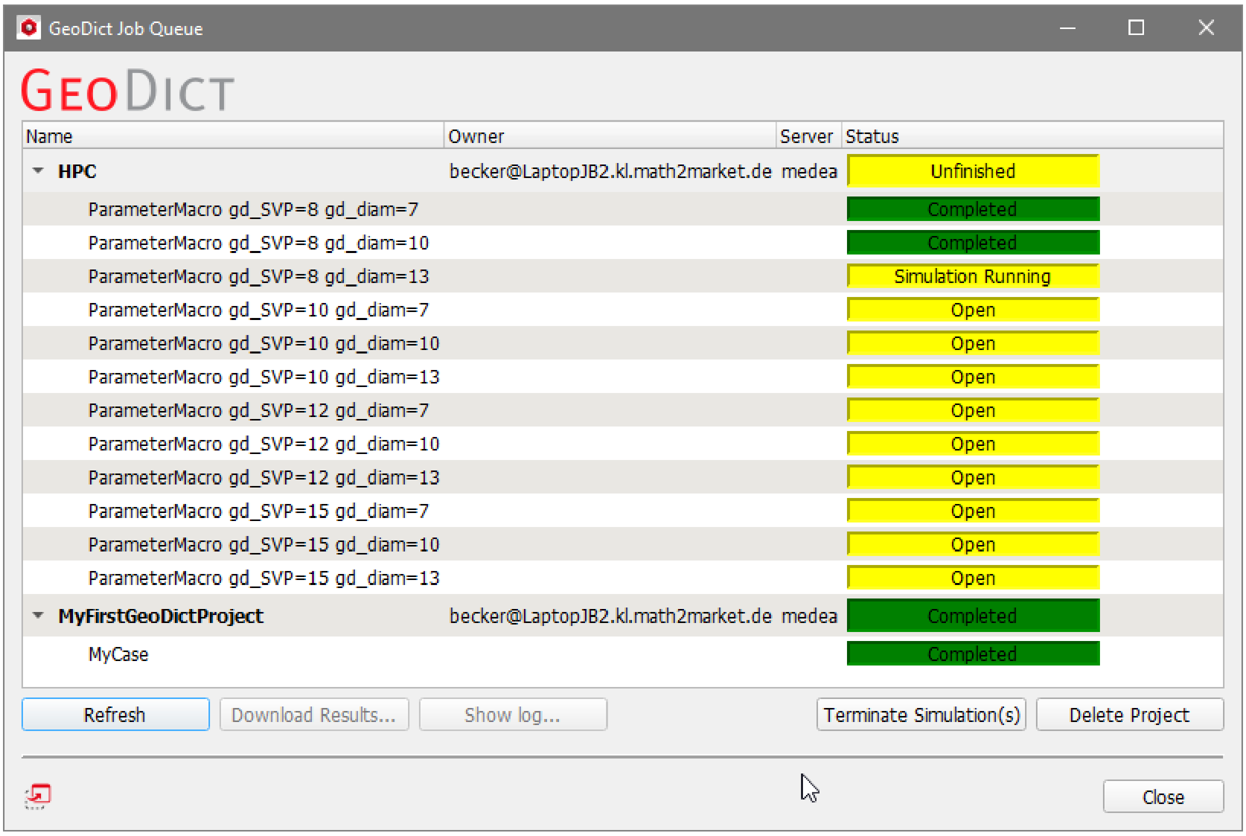




	Solid Volume Percentage (%)	Diameter (µm)
1	8	7
2	8	10
3	8	13
4	10	7
5	10	10
6	10	13
7	12	7
8	12	10
9	12	13
10	15	7
11	15	10
12	15	13

The GeoDict Job Queue dialog

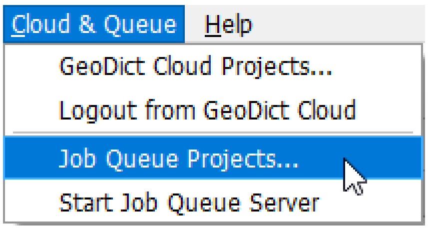
Clicking **Run** submits the job and the **GeoDict Job Queue** dialog opens.



The dialog shows project name (here HPC and MyFirstGeoDictProject) and the names of the cases below. For parameter macros, multiple cases are created. Otherwise, there will be only one case per project.

You may close this dialog now and continue to work in GeoDict, create new structures and run other simulations. You can even close GeoDict locally, and the queued jobs will still be executed on the server.

This dialog can be open again anytime later in GeoDict from the **Cloud & Queue** menu by selecting the **Job Queue Projects...** dialog.

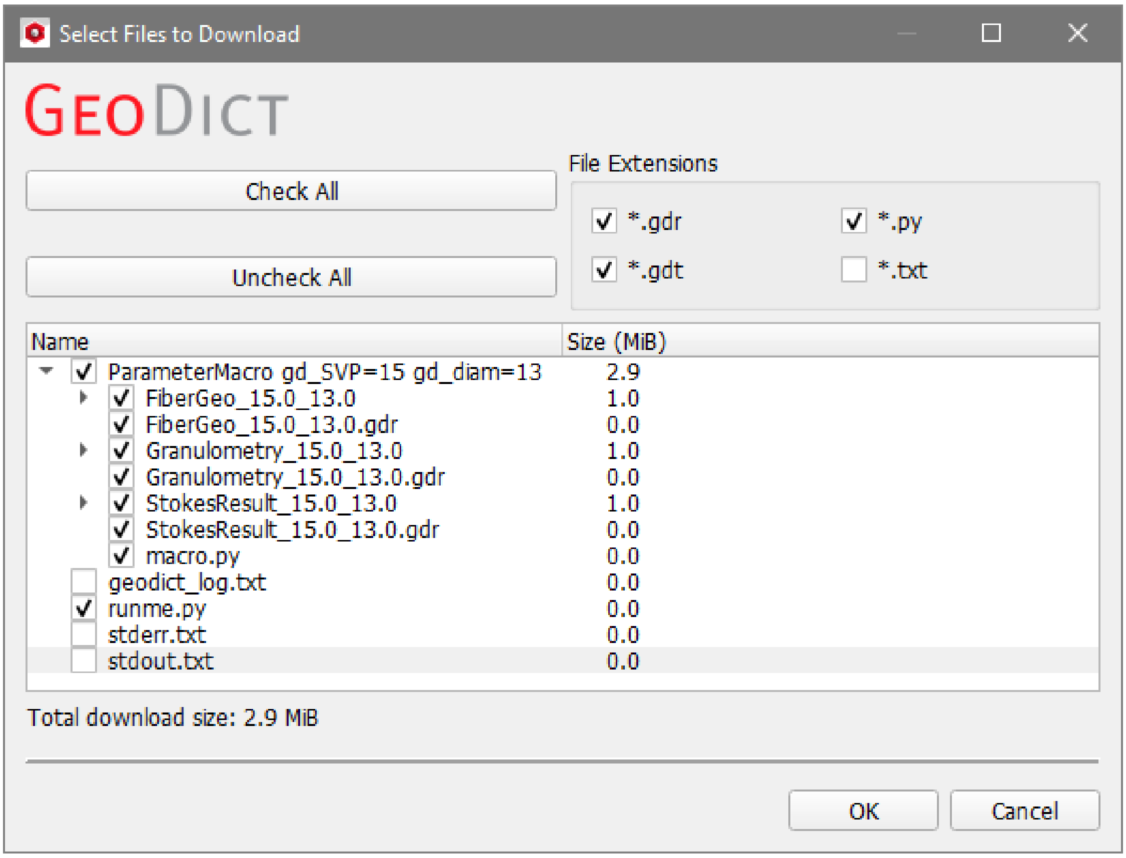


Refresh

Clicking the **Refresh** button updates the **Status** section. The **Status** section is not updated while the dialog is open if the button is not clicked.

Download Results

The **Download Results...** button gets active when a simulation is finished. Select either a project or a case and then click on **Download Results...** If a project was selected, the files for all subcases are included.



Select the files you want to download (copy) from the server to your local computer. Click **OK** and select a destination in the file dialog. All selected files and folders are copied to this location.

All files are copied (not moved) to the selected location on the client computer. To delete them on the server, you have to use **Delete Project**.

Show log

Select a case and click on **Show log...** to view the last lines of the log file of this case. To view the complete log file, use **Download Results...** instead.

Terminate Simulation(s)

Select a case and click on Terminate Simulation(s) to stop a computation. Partial results will be retained and can be downloaded afterwards.

Delete Project

All data is stored on the job server computer in the folder

C.\Users\Username\GeoDict2024\JobSystem\projects

of the user who started the job server. Data of individual jobs is stored in subfolders with encoded folder names. To free disk space on the server, select a project and click on **Delete Project**. Make sure that you have downloaded all results needed before.

Terminate Job Server

Closing the terminal window in which the server runs terminates the job queue server.

The job queue server is not stopped, if one of the submitted jobs causes GeoDict to crash. The server is a separate executable that starts each submitted job in another instance of GeoDict.

Cluster computing on Linux clusters

Many prediction modules (e.g. FlowDict) allow to perform computations on Linux clusters. This allows to utilize many compute nodes at the same time to massively parallelize large computations.

In this case, GeoDict needs to be installed on each compute node or it has to be installed on a shared file systems such that each compute node can access it. A floating license installed on a license server is needed and each compute node must have access to the license server. Node-locked licenses do not work for cluster computing.

Three steps are needed to start large simulations on a Linux cluster:

1. Enable password-less login on cluster compute nodes.
2. Prepare a cluster simulation script.
3. Submit a simulation on the cluster.

Enable password-less login

When using SSH to login to (other) cluster nodes, the system typically asks for a password. This is bothersome for multi-process job startup procedures. However, the SSH configuration can be changed to allow password-less login to cluster compute nodes. The script *enablePasswordLessLogin.sh* configures that for you:

1. Switch to the GeoDict installation folder.
2. Execute ./enablePasswordLessLogin.sh

This script has to be executed when logged in to your Linux account on the cluster. It can also be done on your "local" Linux computer, if your local computer and the cluster share the same account. After execution of the script, it is possible to login to cluster node without password input.

Prepare a cluster simulation script

A submission shell script is needed to start a simulation on a cluster. This shell script contains control information for the job submission system (e.g. number of nodes) and calls GeoDict with a floating license and a simulation script. A template for such a script is available in the GeoDict installation folder (Linux version only) with the name :

- *PBSClusterSimulationTemplate.sh* for PBS, and
- *SLURMClusterSimulationTemplate.sh* for SLURM.

The following description considers PBS but is very similar for SLURM.

```
1 #!/bin/bash
2
3 #################################################################################################
4 # The following lines starting with PBS are not comments but internal commands for the queue-system #
5 #################################################################################################
6
7 # If an error happens do not restart the job
8 #PBS -r n
9
10 # Name of the job in the queue
11 #PBS -N GeoDictClusterSimulation
12
13 # Name of the Std-Out-Files
14 #PBS -o GeoDictClusterSimulation.out
15
16 # Name of the Std-Err-Files
17 #PBS -e GeoDictClusterSimulation.err
18
19 # 8 Nodes with 4 processes per node for 48 hour (adjust according to your needs)
20 #PBS -l walltime=48:0:0
21 #PBS -l nodes=8:ppn=4
22
23
24 #################################################################################################
25 #      The following lines need to be adjusted according to your GeoDict and MPI location         #
26 #      EXECUTE THIS SHELL SCRIPT WITH:   qsub PBSClusterSimulationTemplate.sh                     #
27 #################################################################################################
28
29 # Location of the MPI installation that should be used for the simulation
30 export PATH=~/GeoDict2024/MPI/mpich-3.2/bin/:$PATH
31
32 # GeoDict call script (adjust according to your GeoDict installation)
33 GEODICT=~/GeoDict2024/geodict2024
34
35 # GeoDict licence file (adjust according to your licence location)
36 LICENSE=~/.geodict2024/geodict2024.lic
37
38 # Your simulation script that should be performed on the cluster
39 SIMULATIONSCRIPT=~/GeoDictClusterSimulation.py
40
41 #################################################################################################
42 #          The following lines start GeoDict and perform a cluster simulation                     #
43 #################################################################################################
44
45 $GEODICT $LICENSE $SIMULATIONSCRIPT
```

1. Adjust the following lines of the submission script according to your GeoDict installation and cluster settings:
   - Line 20: Maximum runtime of the simulation. If the simulation last longer than the specified runtime than the job is cancelled.
   - Line 21: Number of nodes and processes per nodes (ppn).
   - Line 30: Path to your MPI installation.
   - Line 33: Path to your GeoDict installation.
   - Line 36: Path to your GeoDict floating license.
   - Line 39: Path to your simulation script.
2. Create a simulation script that performs the simulation. Make sure that the parallelization settings in the simulation script use the cluster, e.g.:

```
1  'Parallelization' : {
2    'Mode'              : 'CLUSTER',
3    'NumberOfNodes'     : 8,
4    'ProcessorsPerNode' : 4,
5  },
```

   - Line 3: Number of nodes. This number has to be smaller or equal to the number of nodes in the submission script.
   - Line 4: Processes per node. This number has to be smaller or equal to the number of processes per node in the submission script.

START A SIMULATION ON THE CLUSTER

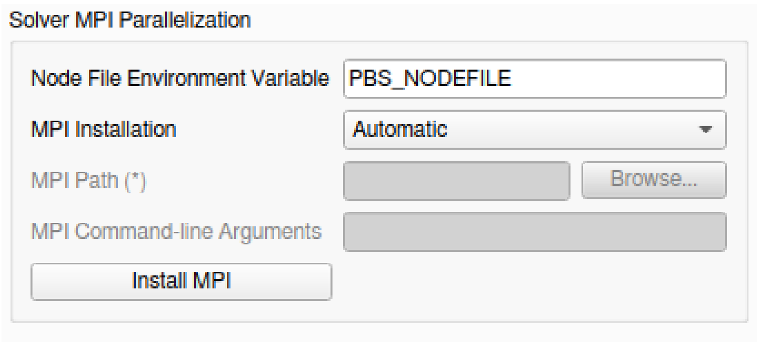
1. Login to the master node of the cluster (e.g. with putty or SSH)
2. Change the working directory to the submission shell script.
3. Make sure that enough disk space is available for temporary saving of flow fields
4. For PBS: Start the simulation with the command:

   qsub <Path to Shell Script Folder>/ *PBSClusterSimulationTemplate* .sh
5. For SLURM: Start the simulation with the command:

   sbatch <Path to Script Folder>/ *SLURMClusterSimulationTemplate*.sh
6. A log file for the simulation run is created with the name **GeoDictClusterSimulation.out**

CHOICE OF COMPUTE NODES

When using a job submission system, it is not a priori known which compute nodes will be used for the computation, the job system will make that decision when starting the job depending on which nodes are available at that moment. The mechanism used by PBS and SLURM to pass the information about the assigned compute nodes to MPI is as follows. The job system will create a file containing a list of the signed compute nodes. The path to this file will be set as an environment variable, typically named PBS_NODEFILE. If another job submission system is used, which may use a different variable name, GeoDict allows to change the name of the environment variable in the **Settings → Settings** dialog.



This mechanism can also be exploited to allow for a manual choice of the compute nodes, e.g. when no job submission system is used:

1. You have to create your own "node" file. Create a text file and fill it with the names of the computers that should be used for the computation, e.g.
   ```
   node001
   node002
   node003
   ```
   The names are separated by newlines.
2. Set the path to the node file as value for the PBS_NODEFILE variable. This can be done with the Linux command: export PBS_NODEFILE=*<path to node file>* This command has to be executed before GeoDict is started.
3. Then follow the instructions described above to start a distributed simulation.

HINTS

The setup of simulations on a cluster can be complex and some things can go wrong. Here is a list of hints to consider before setting up a simulation on a cluster:

- Currently, OpenMPI 1.10.7 does not work on a cluster in combination with the FeelMath solver. Use MPICH3.2 instead of OpenMPI 1.10.7.
- The **Local, Parallel** parallelization option of the EJ and SimpleFFT do not work on cluster nodes. Here, you will get the error message "-1815 „Invalid parent process ID"". Use the **Cluster** parallelization option instead of **Local, Parallel**. Even if just one node is used for the computation.

CLOUD COMPUTING

GEODICT CLOUD

GeoDict gives users the choice to submit computations to the cloud. using the **Cloud** button.

REGISTRATION

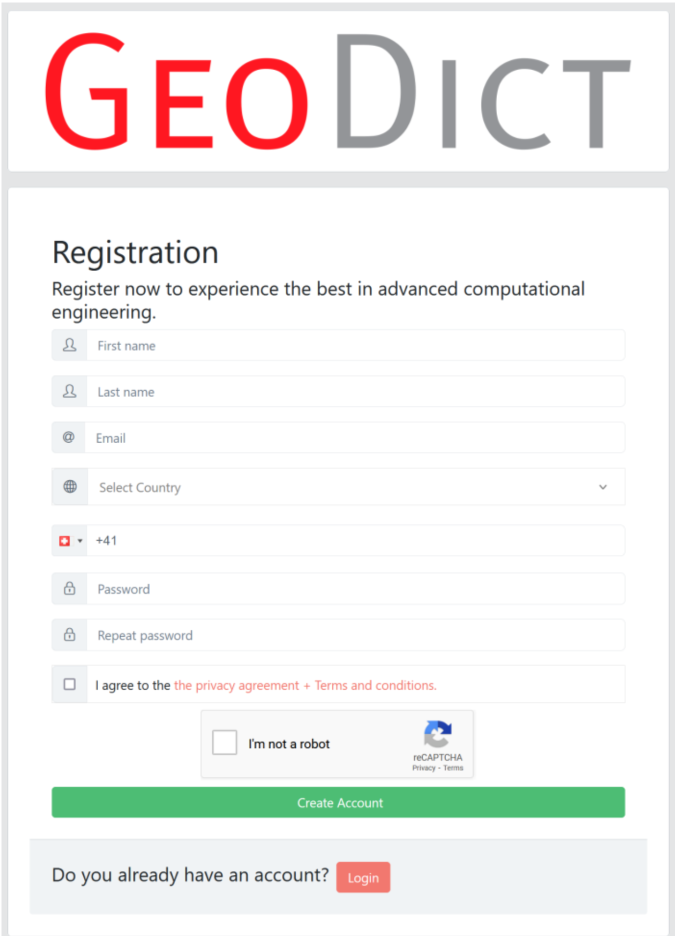
To register for this cloud service, please contact our support at support@math2market.de for more information.

Before you can use the **Run in Cloud** button, an account must be created for your organization.

When you are added as user in your organizations account, you will receive an invitation email. After clicking on the link in the email, a registration form will open in your web browser.

Enter your data to register in your organizations account. After you have created your account, you are set to use the **Run in Cloud** button from GeoDict.

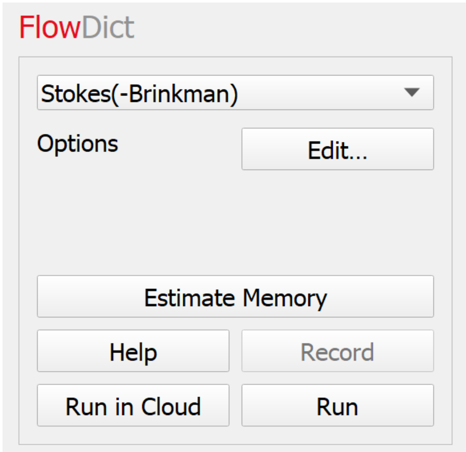
Be aware that additional pay-per-use costs are associated with using the GeoDict cloud.



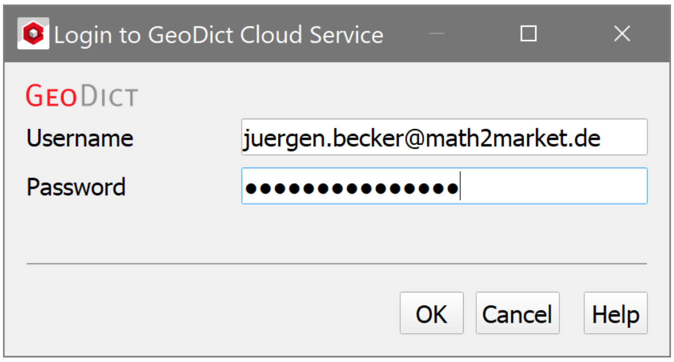
Running a Single Simulation in the Cloud

To start a simulation in the cloud, enter the simulation parameters as usual in GeoDict by clicking on the Options **Edit...** button.

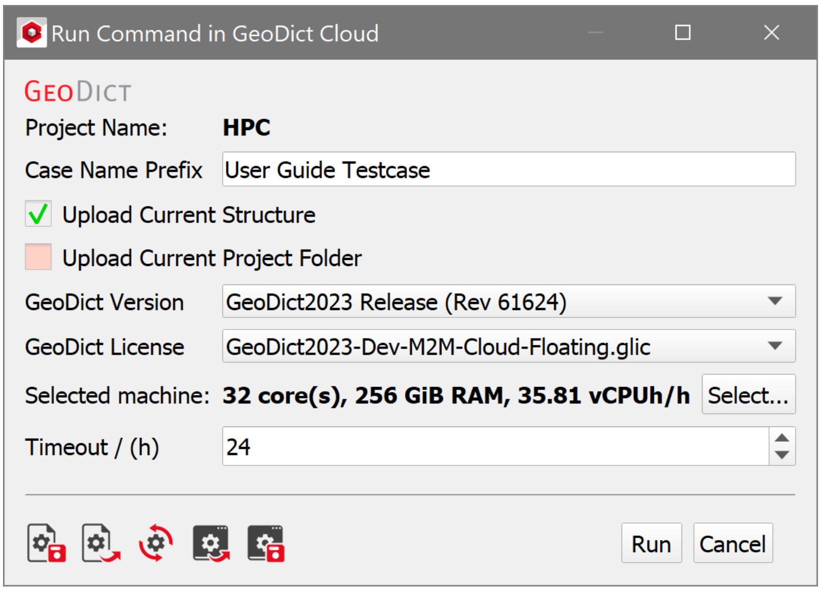
After setting up of the parameters, click the **Run in Cloud** button instead of the **Run** button.



Then, enter your credentials in the pop-up dialog. The Username is your mail address, and the Password is the one you selected when registering.



After entering your credentials, the **Run Command in GeoDict Cloud** dialog opens.



The **Project Name** is the name of the current project folder in GeoDict.

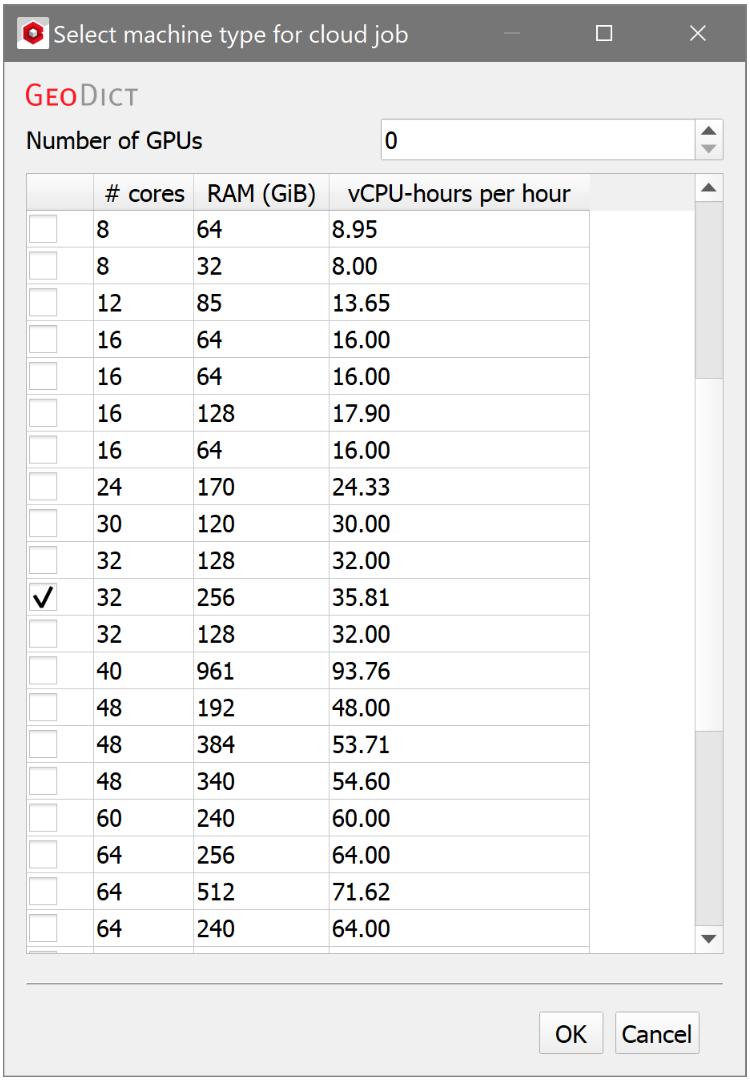
Enter a **Case Name Prefix** that is used to identify the simulation later.

The two checkboxes **Upload Current Structure** and **Upload Current Project Folder** are selected or un-selected automatically depending on the simulation to run.

Select the **GeoDict Version** (should be same as local GeoDict version) and the **GeoDict License**. Only licenses that Math2Market has made available for the specific user are shown here. Typically, just a single license will be available for a user.

Select a Machine

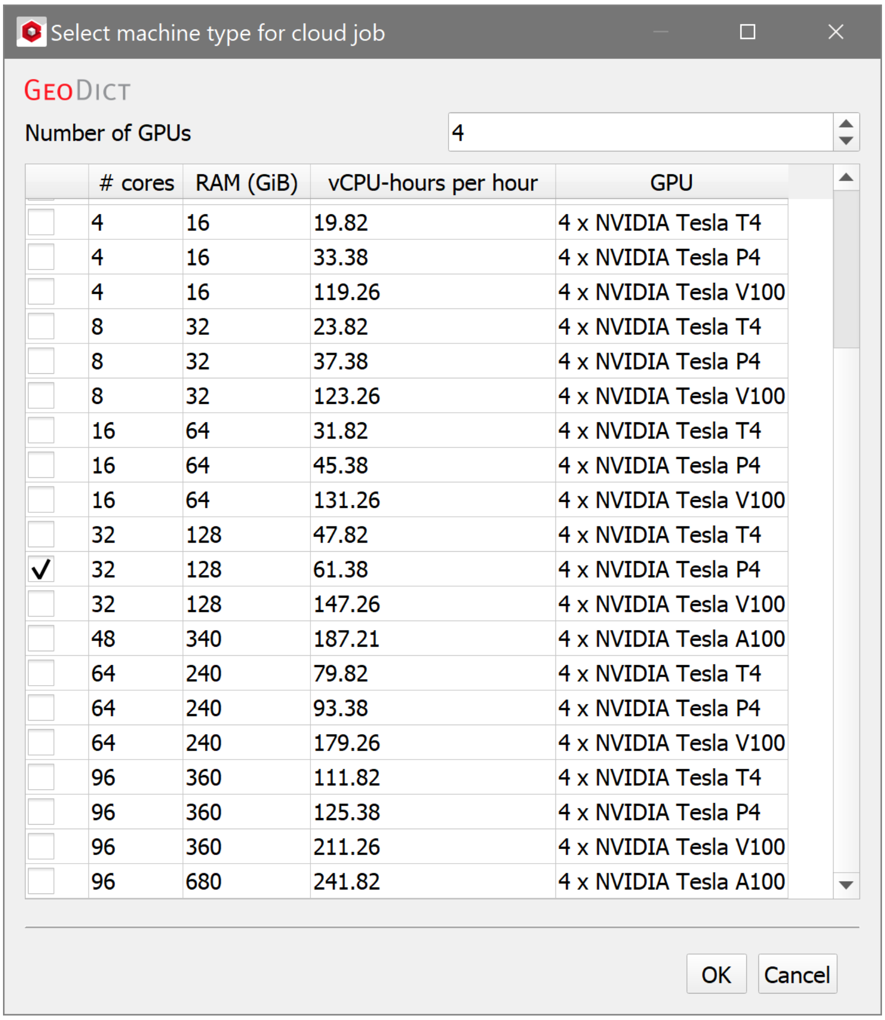
Click on **Select...** to open a dialog that let's you select a machine that is sufficient for your job.



Select a machine that has enough **RAM (GiB)** available to run the job. If you are in doubt how much memory is required, you may run **Estimate Memory** prior to **Run in Cloud**. If the machine does not have enough RAM, executing the job will fail.

Select **# cores** available to run the job. You must select at least as many as selected in the Parallelization dialog. Be aware that the **# cores** shown includes cores available through the use of hyperthreading. That means, a machine with # cores equal to 32 consists of 16 real cores. Many GeoDict jobs cannot efficiently use hyperthreading, their speed is limited by the ability of the processor to process large amount of data, and not by the ability to do many arithmetic operations. Therefore, if you select a machine with 32 cores from the list, the runtimes may be similar for using 16x parallelization or 32x parallelization, as cores available through hyperthreading do not add a significant speedup.

For AI applications using the Tensorflow python library, e.g. FiberFind-AI, it is possible to select a machine with one or several GPUs. Enter the **Number of GPUs** in the box above to receive a selection of machines with the requested number of GPUs.



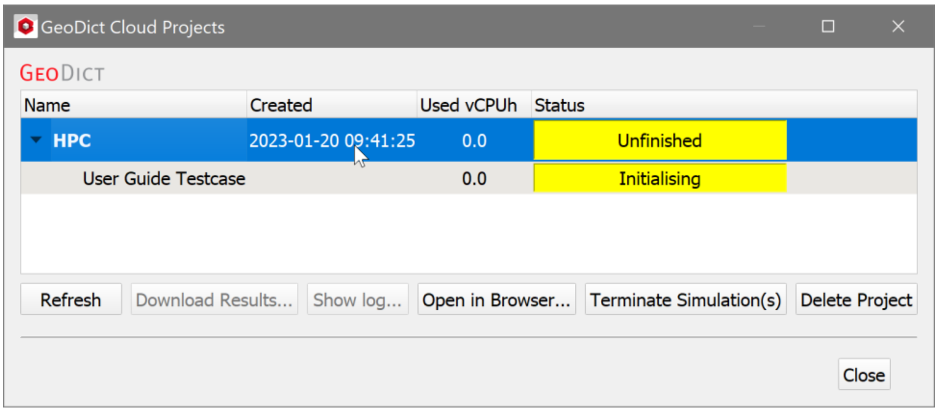
	# cores	RAM (GiB)	vCPU-hours per hour	GPU
☐	4	16	19.82	4 x NVIDIA Tesla T4
☐	4	16	33.38	4 x NVIDIA Tesla P4
☐	4	16	119.26	4 x NVIDIA Tesla V100
☐	8	32	23.82	4 x NVIDIA Tesla T4
☐	8	32	37.38	4 x NVIDIA Tesla P4
☐	8	32	123.26	4 x NVIDIA Tesla V100
☐	16	64	31.82	4 x NVIDIA Tesla T4
☐	16	64	45.38	4 x NVIDIA Tesla P4
☐	16	64	131.26	4 x NVIDIA Tesla V100
☐	32	128	47.82	4 x NVIDIA Tesla T4
☑	32	128	61.38	4 x NVIDIA Tesla P4
☐	32	128	147.26	4 x NVIDIA Tesla V100
☐	48	340	187.21	4 x NVIDIA Tesla A100
☐	64	240	79.82	4 x NVIDIA Tesla T4
☐	64	240	93.38	4 x NVIDIA Tesla P4
☐	64	240	179.26	4 x NVIDIA Tesla V100
☐	96	360	111.82	4 x NVIDIA Tesla T4
☐	96	360	125.38	4 x NVIDIA Tesla P4
☐	96	360	211.26	4 x NVIDIA Tesla V100
☐	96	680	241.82	4 x NVIDIA Tesla A100

The **vCPU-hours per hour** are the costs associated with the selected machine, and depend on the number of cores, Ram and GPUs available.

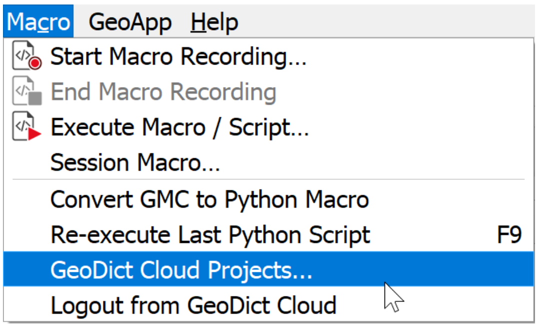
The selected machine will be stopped automatically, after the simulation run is finished. Setting a **Timeout** ensures that the simulation is cancelled and the selected machine is stopped after the selected time.

Start the simulation

Click on the **Run** button to start the simulation. The **GeoDict Cloud Projects** dialog appears:



The job is now tranferred to the cloud and solved on the designated machine. You may close the dialog and continue to work with GeoDict. Open the dialog any time by selecting **Macro – GeoDict Cloud Projects**… in the main menu.



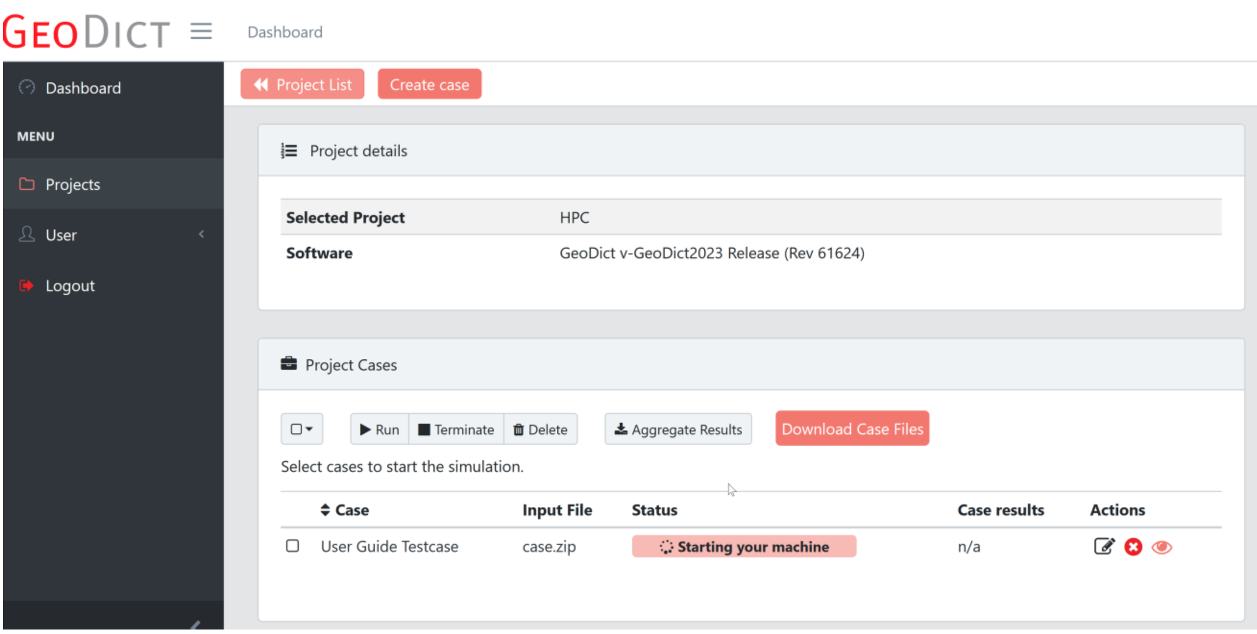
To update the shown status and used vCPUh information, it is necessary to click on the **Refresh** button.

Terminate Simulation(s) stops the selected simulation(s).

Delete Project removes a project from the list. The **GeoDict Cloud Projects** dialog always lists all projects accessible in the account of the user, including simulations and projects started in previous GeoDict sessions.

The Dashboard

Click on **Open in Browser…** to access the dashboard (This step is not necessary to run the simulation, the whole simulation process can also be managed from within the **GeoDict Cloud Projects** dialog). The dashboard gives direct acces to the user profile and the users projects.

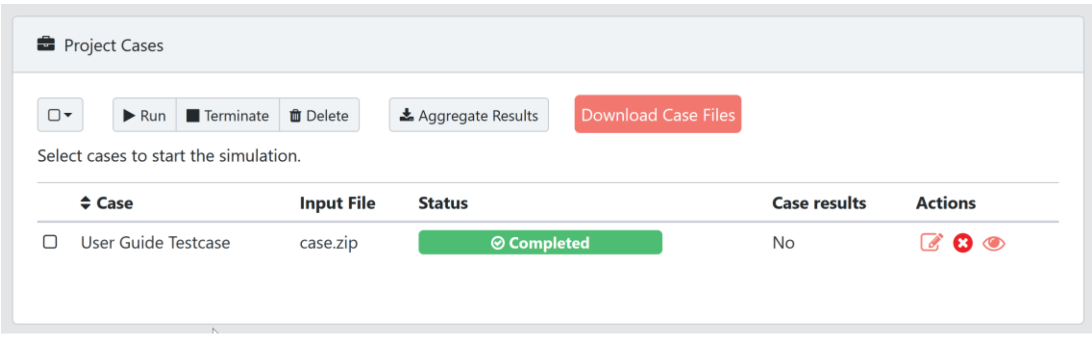


Click on **Projects** and select your project to access the **Project Cases** box. Inside the box, the current status of the simulation is shown. During a simulation run, it will move through various stages until finally arriving at "Completed".

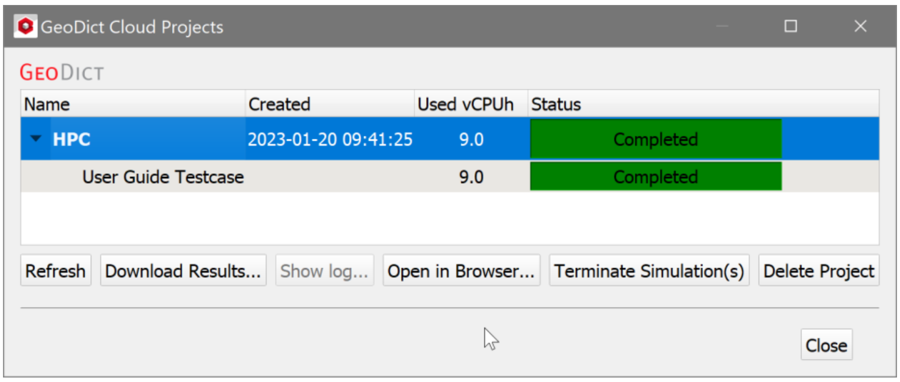
Clicking on the **Detail** toolbutton (the eye symbol) shows more details of the running simulation, including the current log output.

Download Results

When the simulation is completed, it will be shown both on the dashboard



and (after **Refresh**) in the dialog window:

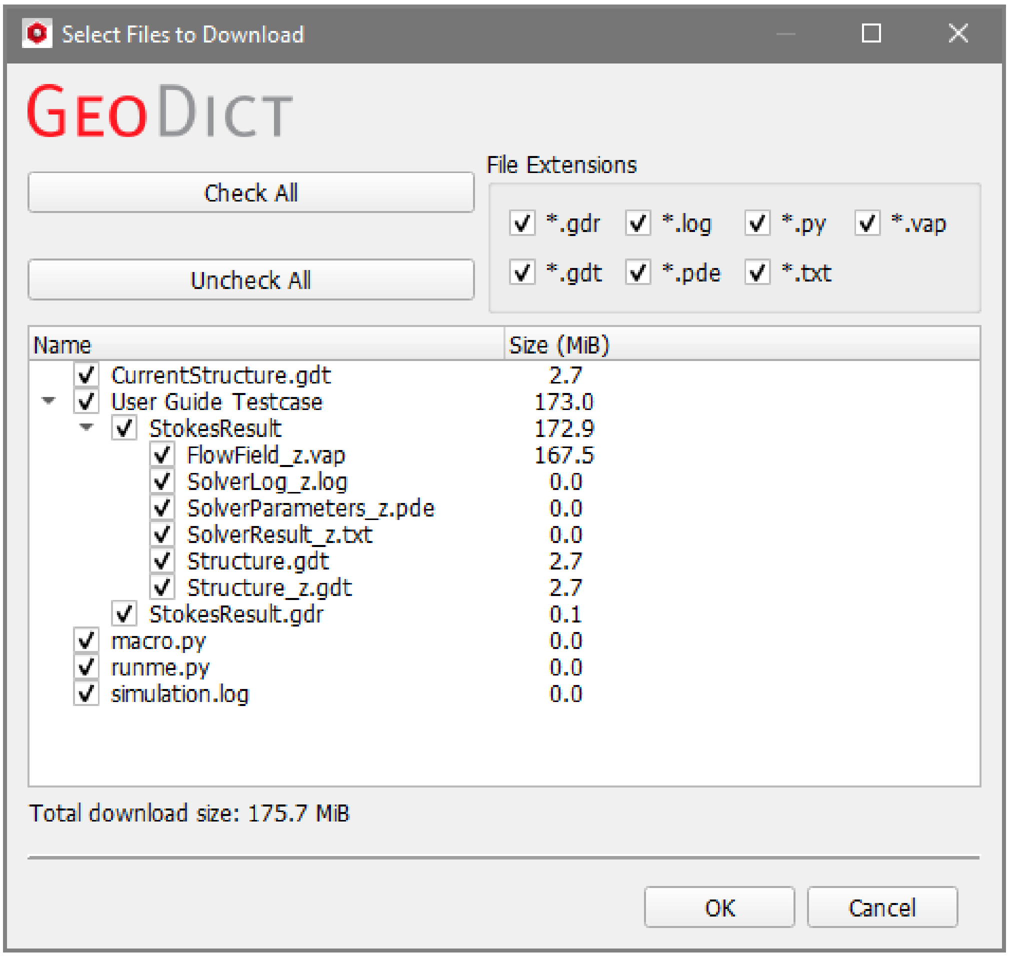


The dialog will also show the **Used vCPUh** of the simulation run, which determines the costs. When the simulation is completed, the simulation results are located in the

GeoDict cloud, not on the local machine. To download the results, click **Download Results...** and another dialog will open that lets you select which files to download.

It is advisable to download at least all .gdr files.

Select the files to download and the location on your local machine.

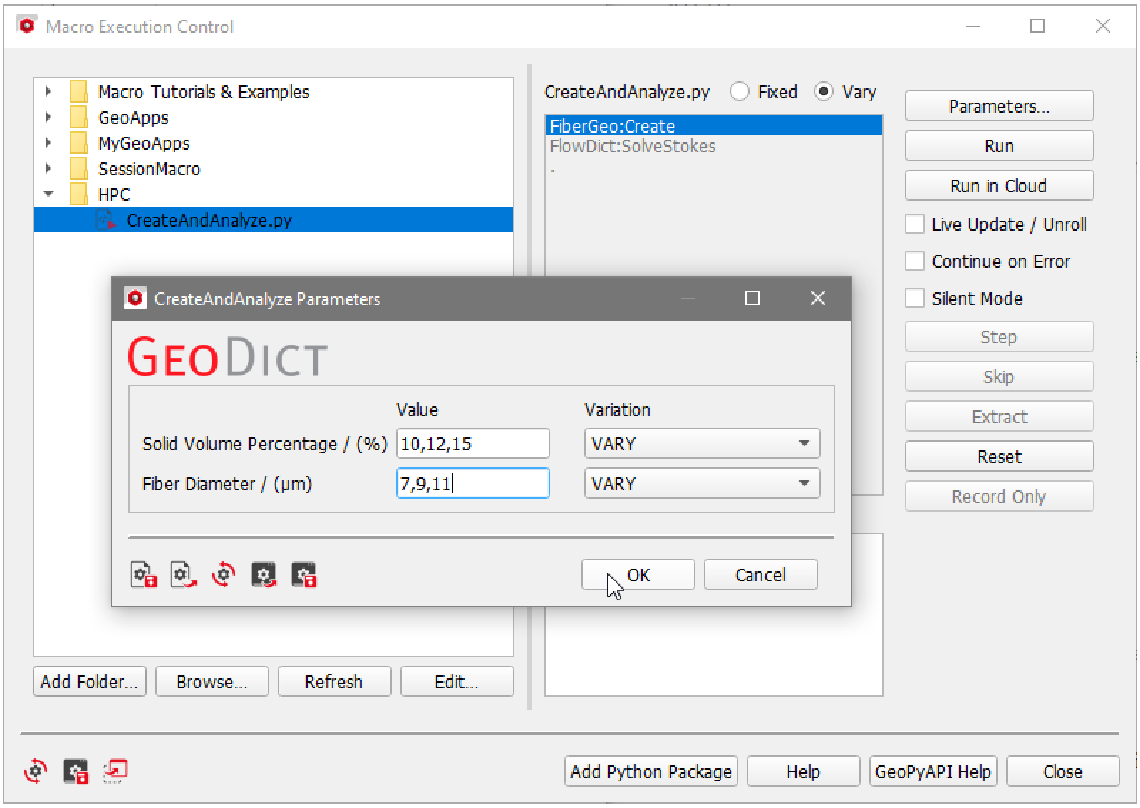


To open the result file locally, select **File → Open Results (*.gdr)** in GeoDict's main menu and browse to the downloaded gdr file. The results will then show in your local GeoDict. Remember, that some visualization options may only be available if additional files located in the corresponding subfolder are also downloaded, e.g., in the shown example, streamlines can only be visualized if the file *FlowField_z.vap* is also downloaded.

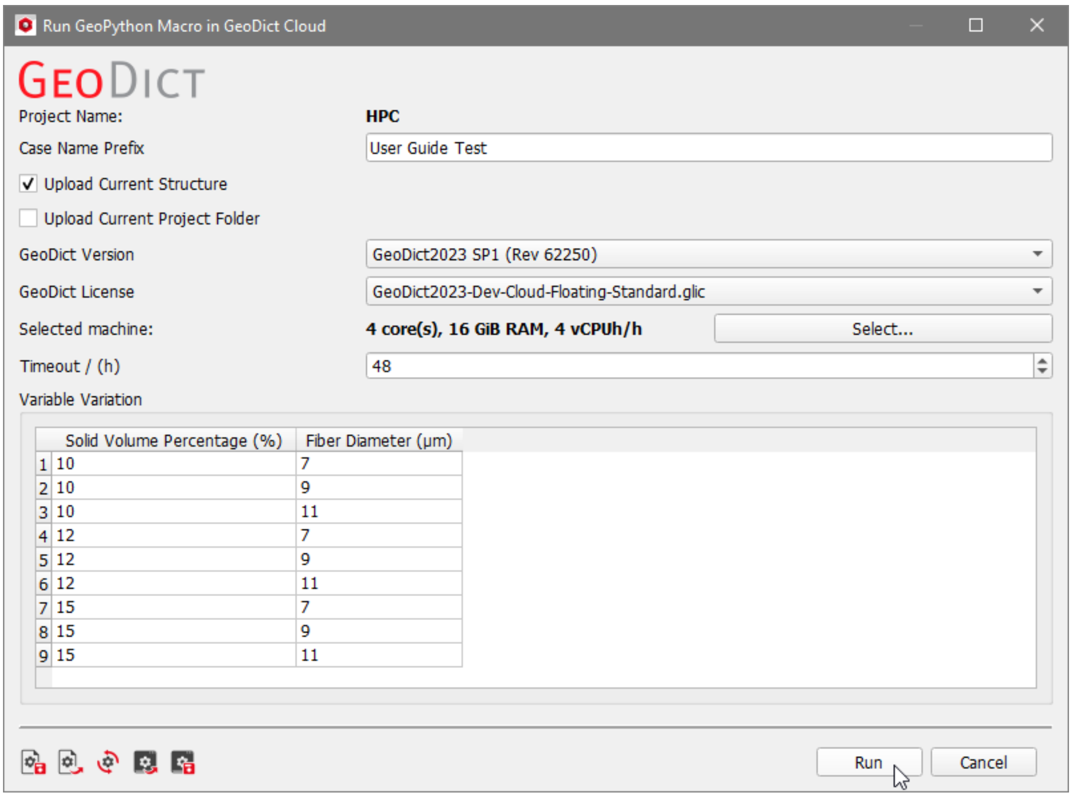
Clicking on **Show log...** opens the logfile in another dialog.

Running Parameter Studies in the Cloud

Besides running a single simulation in the cloud, it is possible to start several simulations at once in a parameter study. Select **Macro → Execute Macro / Script** in the main menu and open the **Macro Execution Control** dialog.



Select the macro to run (it must be a macro containing variables), check **Vary** and open the **Parameters** selection. After entering a list of parameters, click **Run in Cloud** and the **Run GeoPython Macro in GeoDict Cloud** dialog opens:



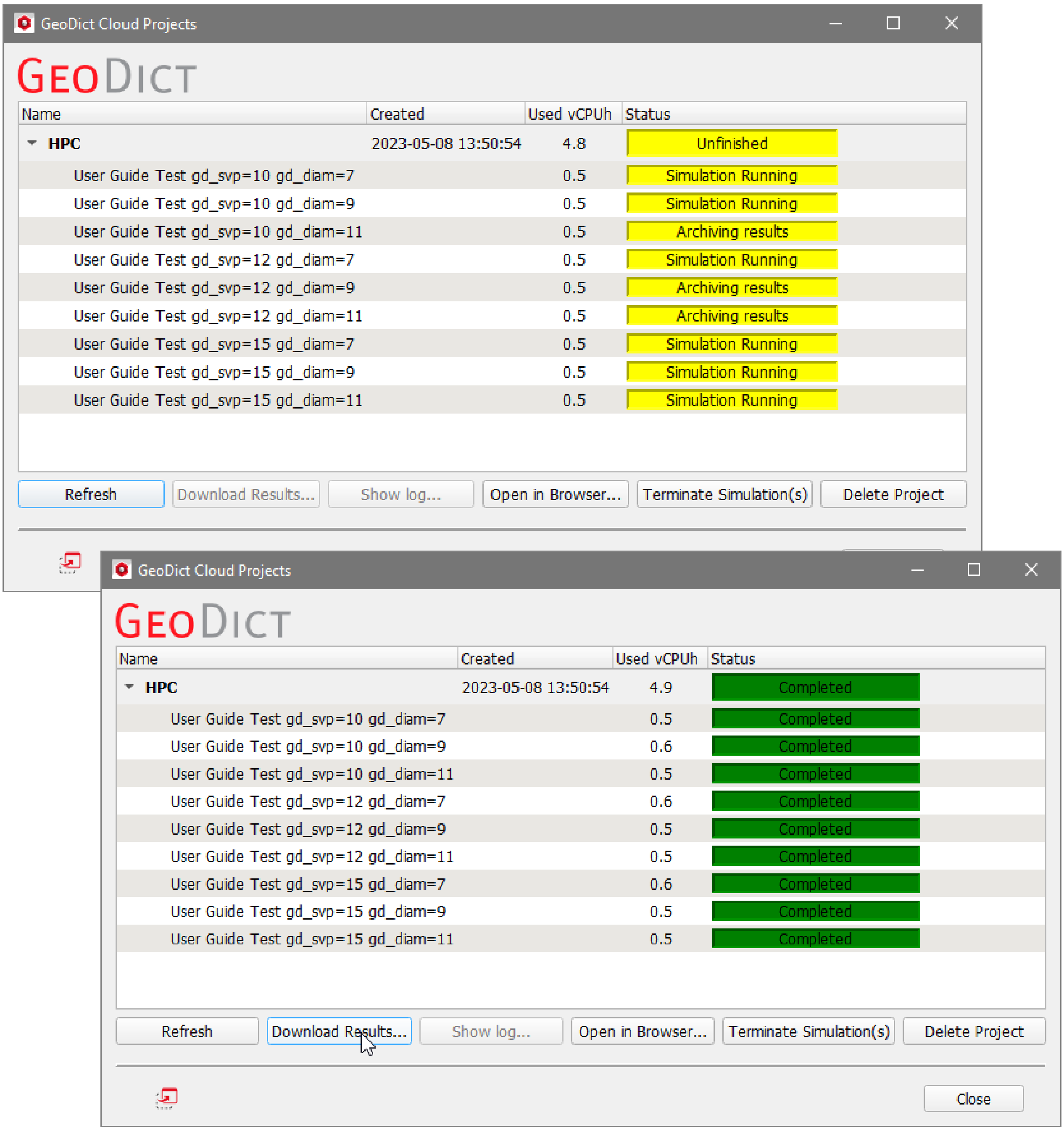
The dialog is similar to the **Run Command in GeoDict Cloud** dialog described on page 28. Here, the **Upload Current Structure** and **Upload Current Project Folder** boxes are selectable.

Select **Upload Current Structure** if the macro operates on the current structure. If all structure models are generated in the macro run itself, it is not necessary to upload the current structure.

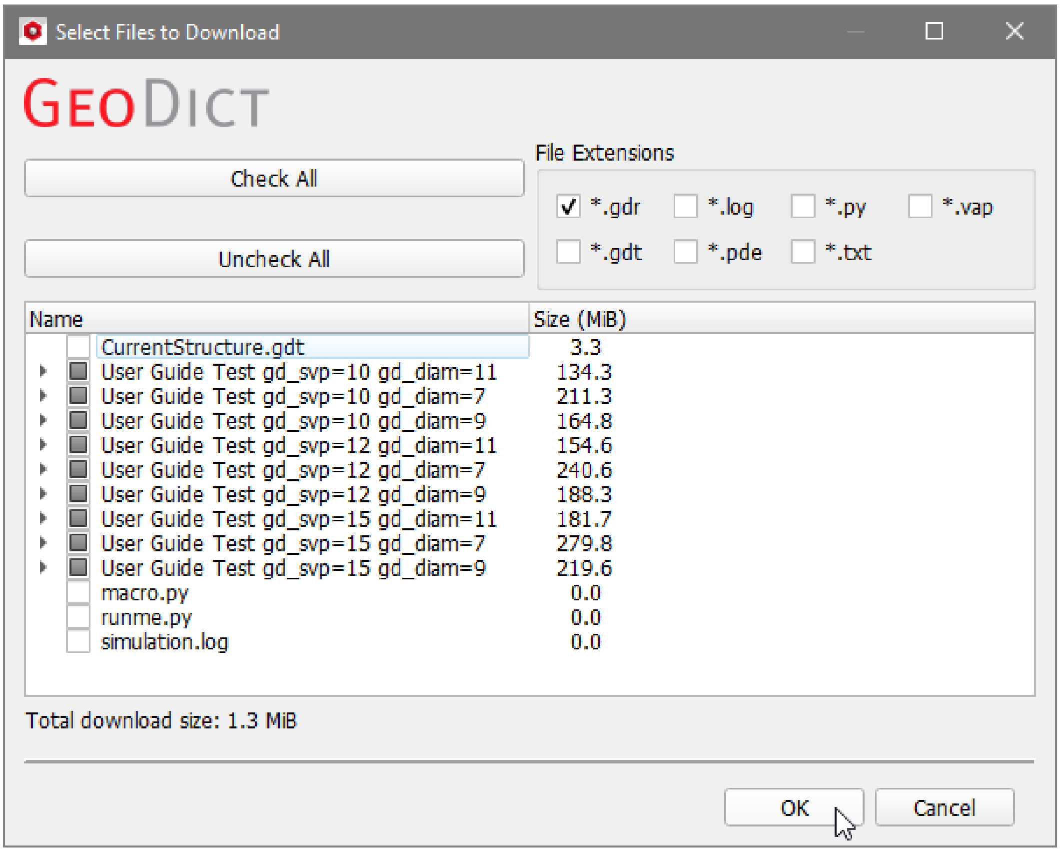
Select **Upload Current Project Folder** if the macro depends on files stored in the local project folder, e.g. structure files are loaded in the macro run.

At the bottom of the dialog, the **Variable Variation** table gives an overview over all simulations and the selected parameters.

Click **Run** to start the simulation in the cloud. The simulations will not be started sequentially on a single machine, but parallel on multiple machines of the selected type. The **GeoDict Cloud Projects** dialog shows the current state of each simulation:



When all simulations are finished, click **Download Results** and select which type of results you want to download to your local computer.



The downloaded files will appear unzipped in the selected folder.

MATH2MARKET CLOUD

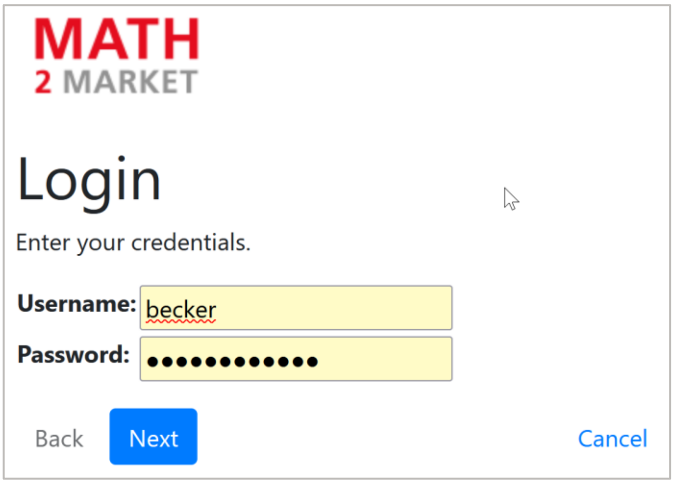
With the Math2Market Cloud solution, only a web browser and an internet connection are needed to use GeoDict. Though your web browser, you will be able to log into your account and access your individual Math2Market Cloud Dashboard. On this dashboard, you may start a virtual machine in the Math2Market Cloud, where GeoDict is already installed and ready-to-use.

Please contact support@math2market.de for more information.

LOG INTO YOUR M2M CLOUD DASHBOARD

You will receive an email from accounts@cloud.math2market.de with your login data (username and password). The link included in this email gives access to the Math2Market Cloud webpage.

In this webpage, enter the login data (username and password).



If multi-factor authentication is required for your account, a second identification (e.g. through a token generated with the Google authenticator app on your smartphone) will be necessary.

Afterwards, your individual Math2Market Cloud dashboard appears. Note that in case of high load, the log in process and the initial account setup may take a few seconds.

MANAGE DATA IN THE DASHBOARD

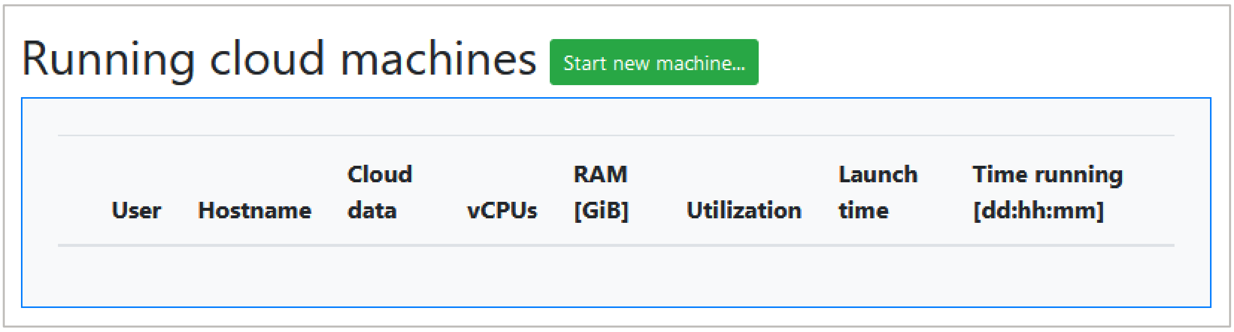
The dashboard contains four buttons in the upper right corner:

- Dashboard: this overview page.
- Change Password: page to change your password.
- MFA setting: page to handle multi-factor authentication settings.
- Logout.

and two panels below:

Running cloud machines

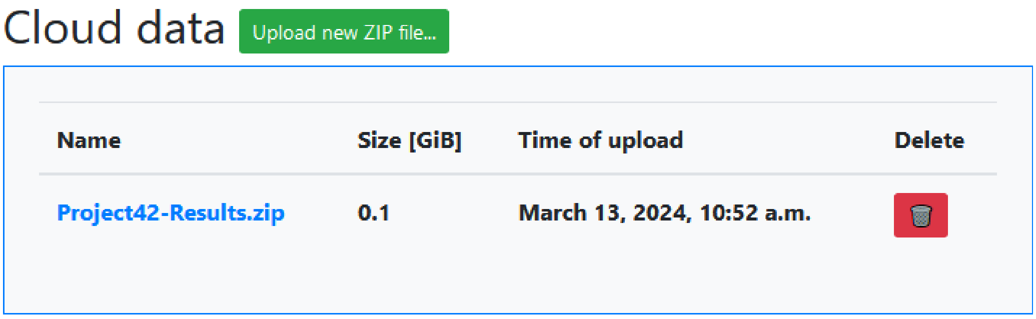
This panel contains a list of your currently running virtual workstations.



You can start a cloud workstation by clicking the green **Start new machine...** button.

Cloud Data

The panel contains a list of currently available data archives (.zip) in the Math2Market cloud.



Zipped data archives shown in this panel of the dashboard come from these sources:

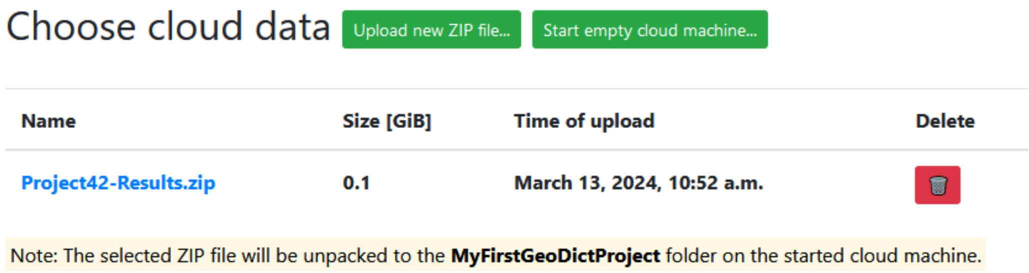
- From your local computer for upload to the cloud. This is done by clicking **Upload new ZIP file...** (in this panel) to select a zip file. Afterwards, the uploaded data is shown in the **Cloud data** panel.
- From the project folder of a running cloud machine. This is done by double-clicking the ***copyProjectFolderToDashboard*** script on the desktop of a running cloud machine. See page 41 for details.

The ZIP files in the **Cloud data** panel (thus, from the Math2Market cloud) can be downloaded to your local computer by clicking on their filename, shown in blue.

A ZIP archive from the **Cloud data** panel might be deleted by clicking the red trash that can appear on the right-hand side of the row.

Create a new M2M Cloud machine

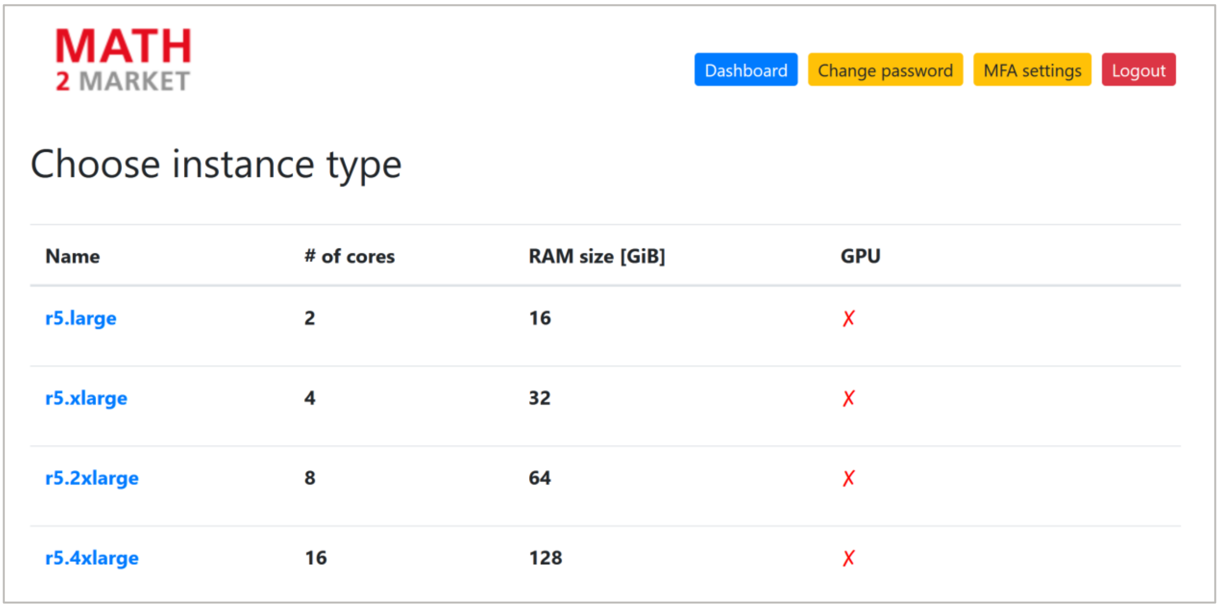
After clicking the **Start new machine...** button, decide if the new machine should already include some of the **Cloud data**, or if it should start without any data:



Name	Size [GiB]	Time of upload	Delete
Project42-Results.zip	0.1	March 13, 2024, 10:52 a.m.	

You can either start with one of the available zip files by clicking on the file name (in blue), or upload a new ZIP file from your local computer, or start an empty cloud machine.

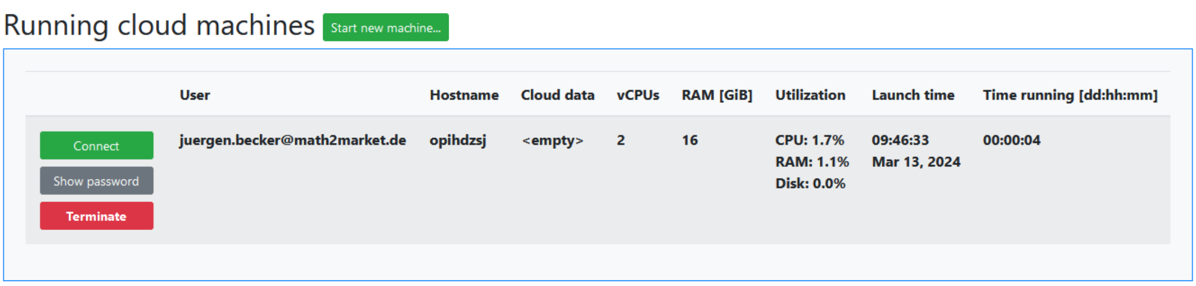
Next, you have to select the cloud machine type. Several different options are available, they differ in the number of available cores and the available RAM. Also, the cloud machine may have a GPU available or not.



Name	# of cores	RAM size [GiB]	GPU
r5.large	2	16	✗
r5.xlarge	4	32	✗
r5.2xlarge	8	64	✗
r5.4xlarge	16	128	✗

Select a machine by clicking on the blue type name. A new cloud machine is created. After a short loading time, a new row appears in the list of **Running cloud machines** panel. The creation of a Cloud machine may take several minutes.

When the Cloud machine is ready to use, the status changes from **Starting** to **Connect** and turns from orange to a green, clickable button.



	User	Hostname	Cloud data	vCPUs	RAM [GiB]	Utilization	Launch time	Time running [dd:hh:mm]
	juergen.becker@math2market.de	opihdzsj	<empty>	2	16	CPU: 1.7% RAM: 1.1% Disk: 0.0%	09:46:33 Mar 13, 2024	00:00:04

Access the created M2M Cloud machines

The newly created cloud machine will be accessed by clicking on the green **Connect** button.

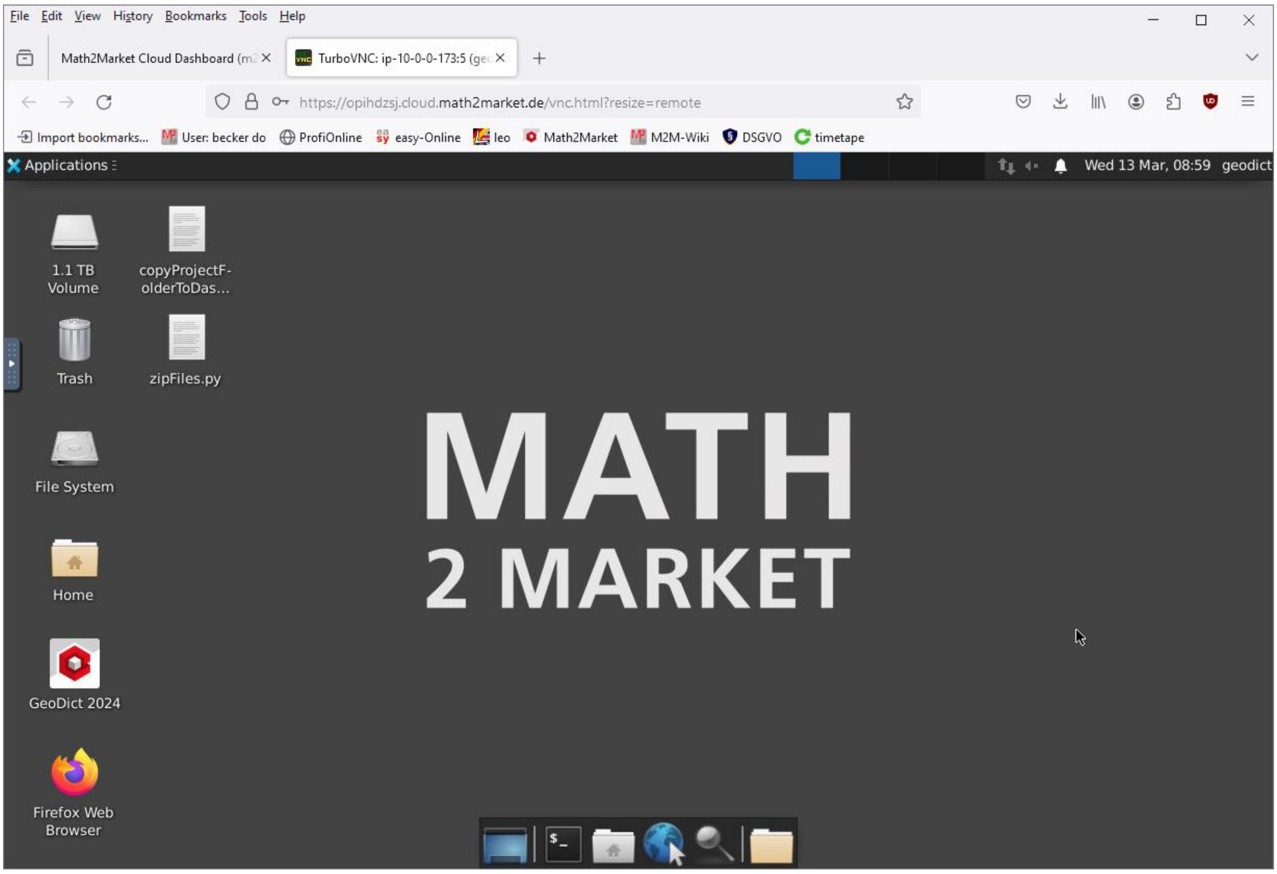
For this, a password will be needed to access the machine in the next step. This password is displayed in a small dialog after clicking the **Show Password** button. Select and copy this password to your clipboard (Ctrl+C) and click **Close** to close the dialog. It is recommended to copy the password before connecting to the machine.

Click now the green **Connect** button under **Status**.

In the new VNC browser tab, click **Connect**.

Paste the copied password (Ctrl+V) and click **Send Password** to connect to the desktop of your cloud machine.

You have now access to your running Math2Market Cloud machine.



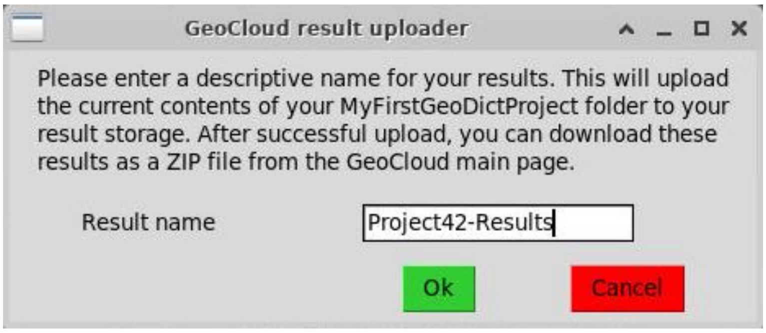
Start GeoDict in the new M2M Cloud machines

Start GeoDict by double-clicking the GeoDict-icon. Starting GeoDict may not be immediate, but take a few seconds.

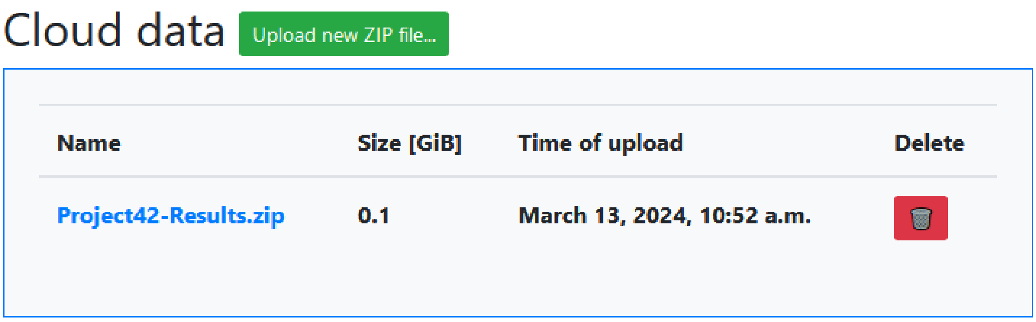
If you started the M2M Cloud machine with a ZIP archive from the **Project data** panel, the content of this archive can be found in /home/geodict/MyFirstGeoDictProject.

Copy Result Data to the Dashboard

To copy GeoDict results to the **Math2Market Cloud Dashboard**, double-click **copyProjectFolderToDashboard.py** on the desktop. You will be prompted for a name. After clicking **OK**, you should see a progress bar during the copy process.

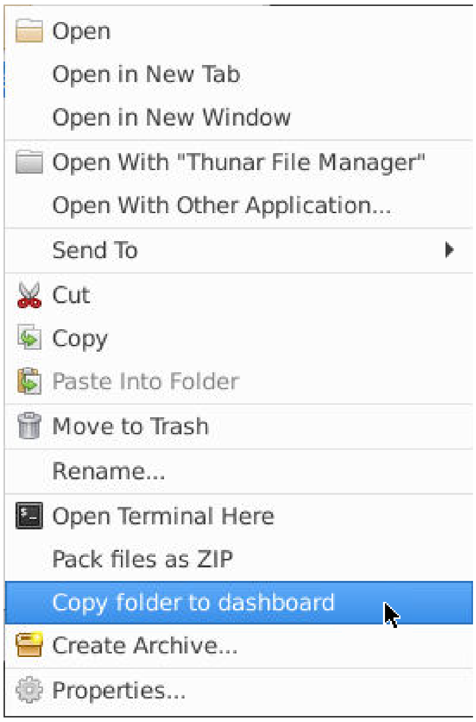


Once the window disappears, you should find these results as a ZIP file in the **Cloud data** panel shortly after. From here, again, you can download it to your machine or start another workstation in the future with that data.

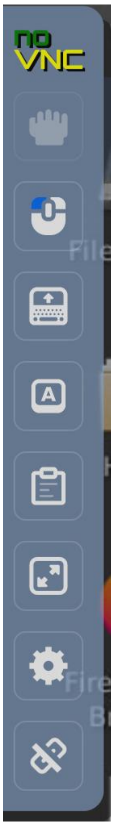


If you want to upload a specific folder instead of your whole project folder, simply right-click on the folder in the file manager and select **Copy folder to dashboard**. You will be prompted to enter a descriptive file name for the new archive to identify it later on.

The name should not contain any special characters (dot, comma, #, &, %, @, /, \, …)


Open
Open in New Tab
Open in New Window
Open With "Thunar File Manager"
Open With Other Application...
Send To
Cut
Copy
Paste Into Folder
Move to Trash
Rename...
Open Terminal Here
Pack files as ZIP
Copy folder to dashboard
Create Archive...
Properties...

Viewer settings for your running M2M Cloud machines



A tab on the left-hand side of your cloud machine browser page gives access to the viewer settings.

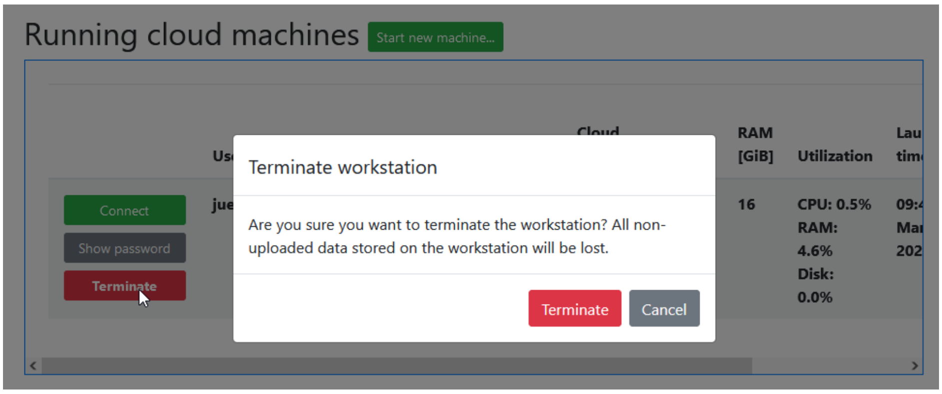
Open the tab by clicking on the arrow  symbol:

There, it is possible (for example) to change to full screen mode (press **Esc** to finish). This should be used if the virtual machine window size is not scaling with the browser window size.

Terminate the running M2M Cloud machine

A running cloud machine does not stop even after closing the browser tab. Therefore, to avoid additional costs, you should always terminate an unused cloud machine!

To do this, go back to the dashboard, to the **Running cloud machines** panel and click the **Terminate** button on the right-hand side. It will prompt you to be sure that you have saved all required files.



In the background, the status changes to **Terminating**. Terminating a cloud machine may take a minute or two. Now close the VNC browser tab.

https://doi.org/10.30423/userguide.geodict

Technical documentation:

Sven Linden

Andreas Wiegmann

Jürgen Becker

Aaron Widera

Barbara Planas

Math2Market GmbH

Richard-Wagner-Str. 1, 67655 Kaiserslautern, Germany
www.geodict.com