JobServer and TaskServer Administration and Configuration

Contents

1 Introduction

The JobServer is the middle tier of *MedeA*. It executes the jobs that are launched by the *MedeA* graphical user interface (the first tier). *MedeA* jobs encompass one or a series of *MedeA* tasks, i.e. the actual calculations. *MedeA* tasks are executed by the *TaskServer*. This section describes the JobServer and TaskServer management, administration, and configuration.

2 Checking, Starting, and Stopping JobServer and TaskServer

2.1 Checking JobServer and TaskServer

Before configuring your JobServer and TaskServer, first check whether they are up and running.

1. Start a web browser (e.g. the Internet Explorer, Firefox) and point the address bar to http://localhost:32000 to see if the JobServer on your local computer is up and running.

   Hint: "localhost" can be replaced with a remote JobServer's IP address or its hostname.

2. Open a browser and point the address bar to http://localhost:23000 to see if the TaskServer on your local computer is up and running.
3. Open the *MedeA* GUI, click Jobs >> Select Server, and verify that **local** is chosen - this is the JobServer on the local computer.
4. Click View and Control Jobs to see if the JobServer automatically opens.

   Note: You can switch to other JobServers to examine if they are up and running

5. On computers without the *MedeA* GUI or a web browser (or X-window) installed, such as HPC headnodes, you can use the following command-line options to check the JobServer and TaskServer statuses:

- JobServer

  ```
  wget localhost:32000
  ```

  with *MedeA* 3.1 or newer:

  ```
  sudo systemctl status mdjobserver.service
  ```

  or with *MedeA* 3.0 or older:

  ```
  sudo /etc/init.d/mdjobserver status
  ```

- TaskServer

  ```
  wget localhost:23000
  ```

  with *MedeA* 3.1 or newer:

  ```
  sudo systemctl status mdtaskserver.service
  ```

  or with *MedeA* 3.0 or older:

  ```
  sudo /etc/init.d/mdtaskserver status
  ```

2.2 Installing JobServer and TaskServer with MDMaintenance

If the JobServer and/or TaskServer had not been installed, follow these steps with *MedeA* 3.1 or newer:

- Windows:
  - Open the Windows Start Menu fo find "*MedeA* Maintenance" in the the "Materials Design folder"
  - Right-click on the program >> More >> Run as administrator .
  - Mark the checkbox of **Manage MD services** and click the Start button.
  - Click Create Service buttons for JobServer and TaskServer, and then click Continue .
  - Open the URL http://localhost:32000 in a web browser to check whether the JobServer frontpage appears. If the JobServer frontpage is visible, then the JobServer is up and running.
  - Submit a job from *MedeA* GUI.
- Linux with X-window:
  - Change to the <*install_dir*>*/MD/Linux-x86_64* directory and find `"MDMaintenance.x`
  - Execute the program **without** sudo: `./MDMaintenance.x`
  - Mark the checkbox of **Manage MD services** and click the Start button.
  - Enter your "Sudo password", and click Check
  - Click the Create Service buttons for JobServer and TaskServer, and then click Continue .
  - Open the URL http://localhost:32000 in a web browser to check whether the JobServer frontpage appears. If the JobServer frontpage is visible, then the JobServer is up and running.
  - Submit a job from *MedeA* GUI.
- Linux with command-line:

  Please refer to Subsection Command-line installation on Linux of the Section Installation from ISO.

2.3 Starting and Stopping the JobServer and TaskServer

If the JobServer and/or TaskServer had been installed but are down because of various reasons, follow these steps:

- Windows:
  - Follow the steps in the previous subsection, but
  - Click Start Service buttons for JobServer and TaskServer, and then click Continue .
  - Open the URL http://localhost:32000 in a web browser to check whether the JobServer frontpage appears. If the JobServer frontpage is visible, then the JobServer is up and running.
  - Submit a job from *MedeA* GUI.
- Linux with X-window:
  - Follow the steps in the previous subsection, but
  - Click the Start Service buttons for JobServer and TaskServer, and then click Continue .
  - Open the URL http://localhost:32000 in a web browser to check whether the JobServer frontpage appears. If the JobServer frontpage is visible, then the JobServer is up and running.
  - Submit a job from *MedeA* GUI.
- Linux with command-line:
  - To start and stop the JobServer with *MedeA* 3.1 or newer:

    ```
    sudo systemctl start mdjobserver.service
    sudo systemctl stop mdjobserver.service
    ```

  - To start and stop the TaskServer with *MedeA* 3.1 or newer:

    ```
    sudo systemctl start mdjobserver.service
    sudo systemctl stop mdjobserver.service
    ```

  - To start and stop the JobServer with *MedeA* 3.0 or older:

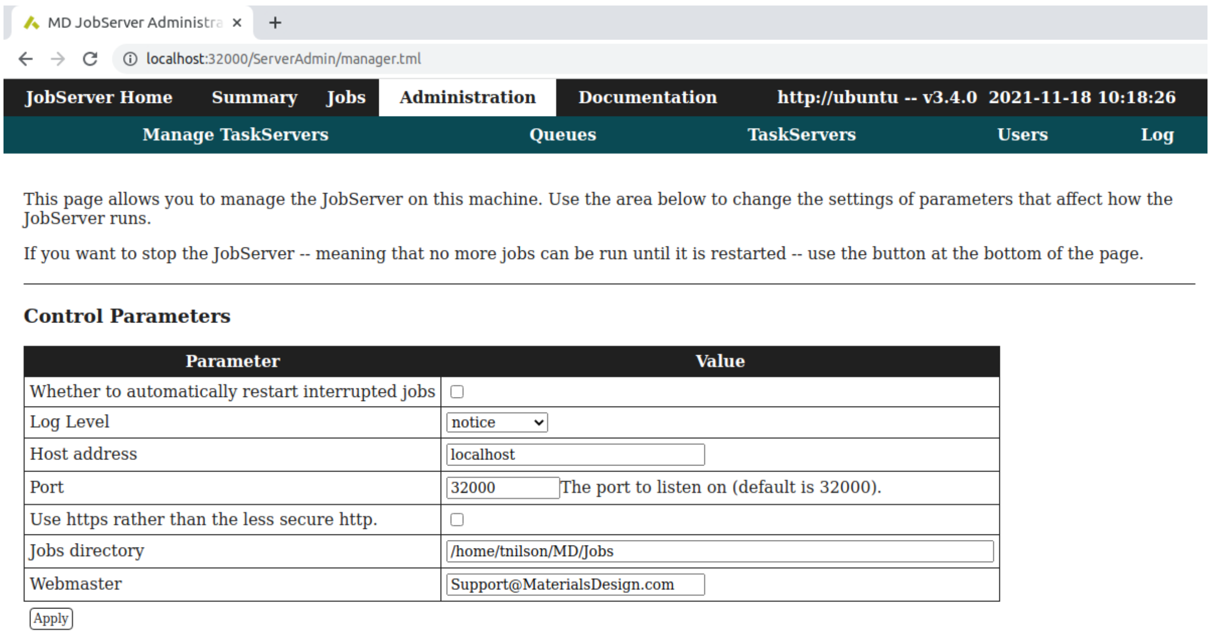
    ```
    sudo /etc/init.d/mdjobserver start
    sudo /etc/init.d/mdjobserver stop
    ```

  - To start and stop the TaskServer with *MedeA* 3.0 or older:

    ```
    sudo /etc/init.d/mdtaskserver start
    sudo /etc/init.d/mdtaskserver stop
    ```

3 JobServer Administration and Configuration

The JobServer web page is primarily used for monitoring jobs. Please see the Monitoring a Running Job section for more information. Once on the JobServer page click the Administration tab to change the settings of the JobServer.



MD JobServer Administra ×

localhost:32000/ServerAdmin/manager.tml

JobServer Home | Summary | Jobs | Administration | Documentation | http://ubuntu -- v3.4.0 2021-11-18 10:18:26

Manage TaskServers | Queues | TaskServers | Users | Log

This page allows you to manage the JobServer on this machine. Use the area below to change the settings of parameters that affect how the JobServer runs.

If you want to stop the JobServer -- meaning that no more jobs can be run until it is restarted -- use the button at the bottom of the page.

Control Parameters

Parameter	Value
Whether to automatically restart interrupted jobs	☐
Log Level	notice
Host address	localhost
Port	32000 The port to listen on (default is 32000).
Use https rather than the less secure http.	☐
Jobs directory	/home/tnilson/MD/Jobs
Webmaster	Support@MaterialsDesign.com

Apply

On this page, you can set the Log Level. The default level is *notice*; however, this can be changed to *debug* if you are having issues configuring the JobServer. The Host address is the hostname or the IP address of the JobServer machine/computer and this name/address must be reachable from the TaskServer machine/computer. On local laptops and workstations, setting this variable to `localhost` or `127.0.0.1` is sufficient. For HPC headnodes, this variable is recommended to be the full hostname (i.e. with the domain name) or IP address. The Port is the port number via which the JobServer communicates. Though the default port number is `32000`, you can change it to other numbers.

Warning: The port for the JobServer needs to be open through the firewall settings of the JobServer computer/machine. The JobServer has to be able to contact TaskServers via this port and TaskServers should be able to contact the JobServer via this port.

The Jobs directory is the directory where all the job folder and files are stored by their job id number. The default directory of the JobServer is the *Jobs* folder in the root of the installation directory, but you can change it to anywhere else. All the input and output files will be stored here after all the tasks have finished, see the next section *TaskSever* to learn how to administer tasks.

All of the above settings are saved in the configuration file (*JobSever.options*) located in the *JobServer* directory of the root installation directory. You can make changes to this file that will directly change the JobServer settings and appear on the JobServer page after the JobServer is restarted.

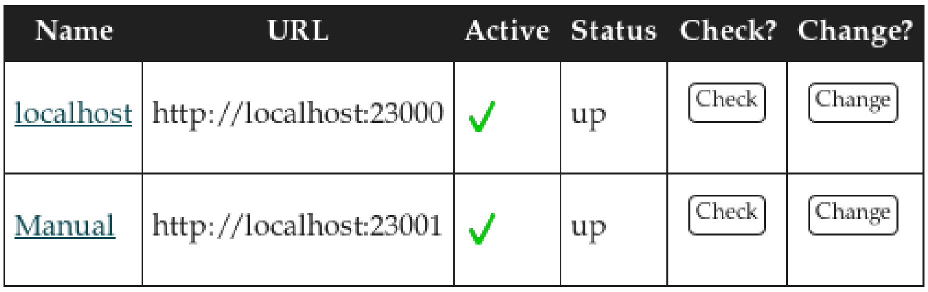
Warning: Incorrect settings in the *JobSever.options* will cause the JobServer to produce an error upon start and a JobServer malfunction. Please contact MD Support [1] before changing this file.

MedeA stores job information in the *MDJobs.db* database file which is located in the *Databases* folder of the root installation directory. This database contains all the information displayed on the Jobs page. Jobs can be added to this database using the Import/Reconnect Jobs function of the Materials Design Maintenance program, see the section Using the Materials Design Maintenance Program for more information.

In the green bar are 5 sub-tabs:

- Manage TaskServers : This page allows you to add and manage the TaskServers that this JobServer uses. Listed in a table at the bottom of the page are the registered TaskServers:

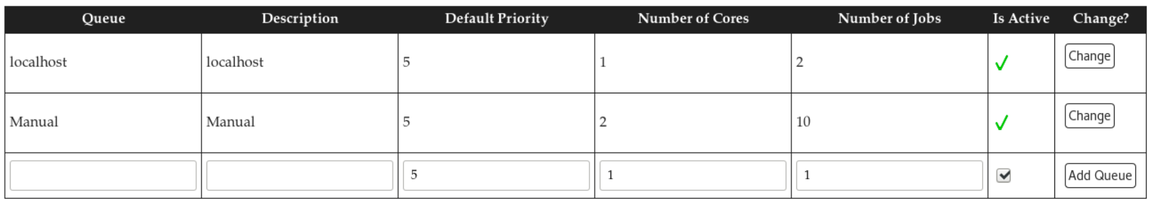
[1] support@materialsdesign.com



Name	URL	Active	Status	Check?	Change?
localhost	http://localhost:23000	✓	up	Check	Change
Manual	http://localhost:23001	✓	up	Check	Change

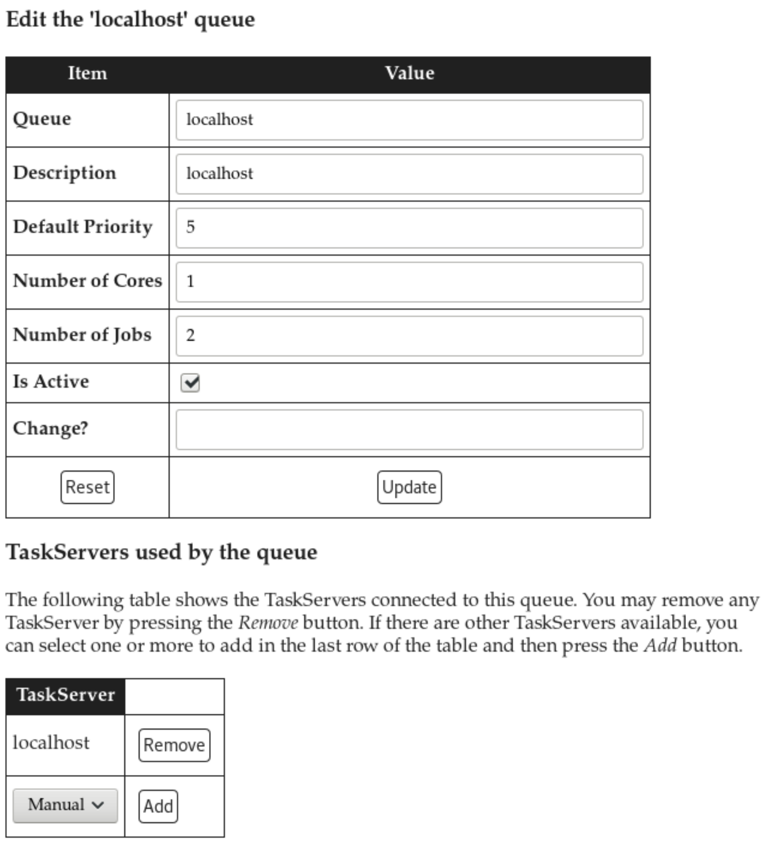
Click the Check button to check the communication between the JobServer and this particular TaskServer. Click the Change button to change the TaskServer domain name/IP address, port number, or make it active/inactive. You can also delete this TaskServer from the list.

- Queues : This page controls the available *MedeA* queues on this JobServer to submit your jobs to:



Queue	Description	Default Priority	Number of Cores	Number of Jobs	Is Active	Change?
localhost	localhost	5	1	2	✓	Change
Manual	Manual	5	2	10	✓	Change
		5	1	1	✓	Add Queue

Click the Change button to manage the settings:



Edit the 'localhost' queue

Item	Value
Queue	localhost
Description	localhost
Default Priority	5
Number of Cores	1
Number of Jobs	2
Is Active	✓
Change?	
Reset	Update

TaskServers used by the queue

The following table shows the TaskServers connected to this queue. You may remove any TaskServer by pressing the *Remove* button. If there are other TaskServers available, you can select one or more to add in the last row of the table and then press the *Add* button.

TaskServer	
localhost	Remove
Manual	Add

- **Queue**: Name of this *MedeA* queue (this name will appear upon submitting jobs from the *MedeA GUI* to this JobServer)
- **Description**: Description of this *MedeA* queue only for your information
- **Default Priority**: 5 is the default value, 10 is the highest priority, while 1 is the lowest

Note: The priority is only effective within *MedeA* and does not affect the priority of queuing systems such as PBS, LSF, SLURM, GridEngine, etc.

- **Number of Cores**: The default number of cores for new jobs. You can change the number of cores upon job submission to larger of smaller values.

Note: The maximum number of compute cores that a calculation, i.e. *MedeA* task is permitted to use, is determined by the TaskServer option **Number of Parallel Cores**, see the Section *TaskServer Administration* for more information.

- **Number of Jobs**: The number of jobs this *MedeA* queue should simultaneously process by creating input files and queueing scripts and submitting calculations to TaskServers
- **Is Active**: Whether this *MedeA* queue is active
- TaskServers used by the queue: Manage the TaskServers for this queue so tasks submitted to this *MedeA* queue are executed by this TaskServers. If there is no TaskServer associated with this queue, then jobs submitted to this queue will not run and stay pending.

• TaskServers

Advanced settings on JobServer and TaskServer security. See Section the Sharing and Security for more information.

Note: To see a list of TaskServers for this JobServer, use the Manage TaskServer option

• Users

Advanced settings on JobServer and TaskServer security. See Section the Sharing and Security for more information.

• Log

Contains the JobServer log files which can be useful for debugging JobServer issues.

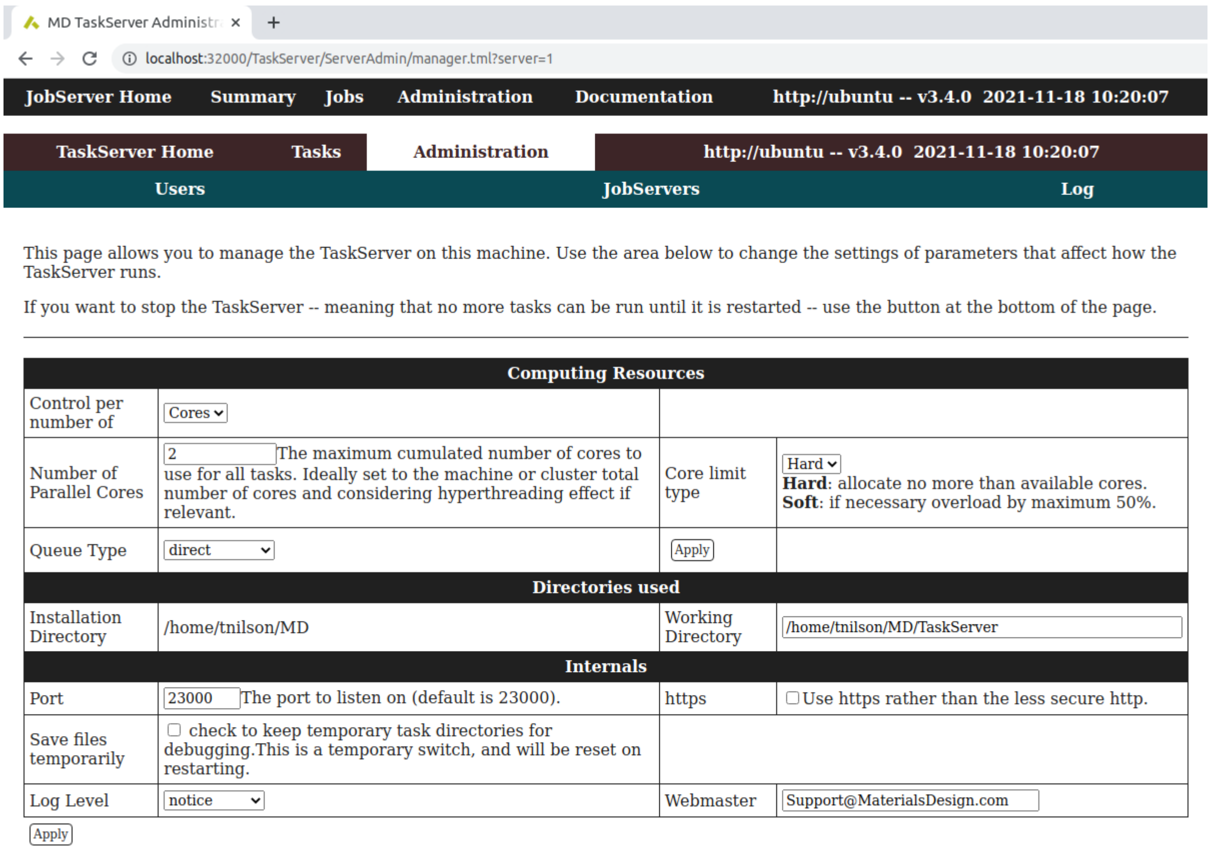
3.1 Administrative Rights

It is recommended to have *MedeA* JobServer and TaskServer running in the background. To install *MedeA* JobServer and TaskServer accordingly requires administrative rights during installation. Windows provides a mechanism, called Services, to accomplish that. On Linux, these service scripts are saved in the directory */etc/systemd/system*. You can ask for temporary administrative rights or have your administrator install *MedeA* JobServer and TaskServer for you. see Section Run as Administrator for information about running programs as an administrator. Without administrative rights, JobServer and TaskServer can be started manually as interactive programs, see Section Running JobServer and TaskServer Scripts.

4 TaskServer Administration

The TaskServer is the end tier of *MedeA* that executes all the individual computations and calculations (tasks) needed to complete jobs. The *MedeA* TaskServer has several options on how to run your computational tasks, e.g. you can choose to run in serial, in parallel, or through a queuing system. You can limit the number of cores used on a specific TaskServer machine. This section describes what options are available to configure how tasks run.

In the following, please have a TaskServer installed that can be contacted through the JobServer links Home >> Administration >> Manage TaskServers or directly through the direct URL *http://<taskserver>:23000* where <*taskserver*> can be *localhost*, your machine's full hostname, or your machine IP address. Both, the sequence of links and the direct URL should open the following page:



MD TaskServer Administr × +

localhost:32000/TaskServer/ServerAdmin/manager.tml?server=1

JobServer Home | Summary | Jobs | Administration | Documentation | http://ubuntu -- v3.4.0 2021-11-18 10:20:07

TaskServer Home | Tasks | Administration | http://ubuntu -- v3.4.0 2021-11-18 10:20:07

Users | JobServers | Log

This page allows you to manage the TaskServer on this machine. Use the area below to change the settings of parameters that affect how the TaskServer runs.

If you want to stop the TaskServer -- meaning that no more tasks can be run until it is restarted -- use the button at the bottom of the page.

Computing Resources			
Control per number of	Cores		
Number of Parallel Cores	2 The maximum cumulated number of cores to use for all tasks. Ideally set to the machine or cluster total number of cores and considering hyperthreading effect if relevant.	Core limit type	Hard **Hard**: allocate no more than available cores. **Soft**: if necessary overload by maximum 50%.
Queue Type	direct	Apply	
Directories used			
Installation Directory	/home/tnilson/MD	Working Directory	/home/tnilson/MD/TaskServer
Internals			
Port	23000 The port to listen on (default is 23000).	https	☐ Use https rather than the less secure http.
Save files temporarily	☐ check to keep temporary task directories for debugging.This is a temporary switch, and will be reset on restarting.		
Log Level	notice	Webmaster	Support@MaterialsDesign.com

Apply

Note: To make any change of the TaskServer configuration options effective, click on the Apply button.

Click on the Administration tab in the brown navigation bar to proceed to the page with TaskServer configuration options:

The *Control per number of* drop-menu has two options:

- *Cores*

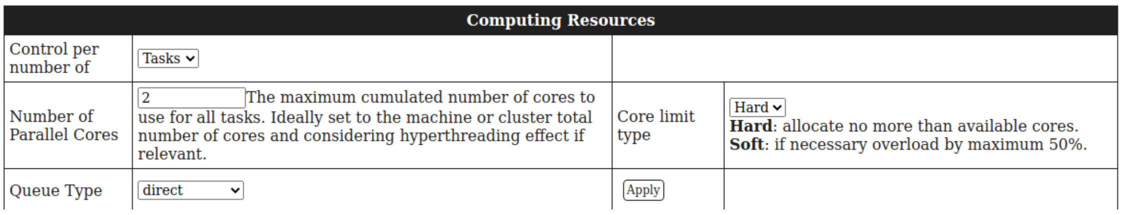
  Follow-up options are:

  – *Number of Parallel Cores*: The maximum number of cores to use for all tasks.

  – *Core limit type*: *Hard* or *Soft* which allocates no more than available cores (*Hard*) or overload by a maximum of 50% (*Soft*), if necessary.

  The **Number of Parallel Cores** determines how many cores are available for all tasks running on this TaskServer. The default, set during installation, is the number of physical cores (no hyper-threading) on the TaskServer computer/machine. This is the right value for using *mpi* or *direct* mode. If you are using an external queuing system you may increase **Number of Parallel Cores** to the number of available cores. The **Number of Parallel Cores** is an upper limit and can be raised at run time in the job submission dialog, which shows the value set for a given queue as the default. If you have 10 compute nodes each with 24 cores, the **Number of Parallel Cores** variable can be set to *240*.

- *Tasks*



Computing Resources			
Control per number of	Tasks		
Number of Parallel Cores	2 The maximum cumulated number of cores to use for all tasks. Ideally set to the machine or cluster total number of cores and considering hyperthreading effect if relevant.	Core limit type	Hard **Hard**: allocate no more than available cores. **Soft**: if necessary overload by maximum 50%.
Queue Type	direct	Apply	

Follow-up options are:

- *Number of Parallel Cores*: The maximum number of compute cores per task.
- *Simultaneous tasks*: How many *MedeA* tasks (calculations) to run simultaneously.

The *Queue Type* determines how the executables of of the compute engines LAMMPS, GIBBS, VASP, MOPAC, and Gaussian are executed by the TaskServer. The following options are available:

- **direct**: uses Intel MPI5 included in *MedeA* to run calculations (serial or parallel) *interactively*
- **mpi**: uses Intel MPI5 included in *MedeA* to run serial or parallel jobs with an automatically generated script
- **PBS**, **LSF**, **GridEngine**, **SLURM**, etc.: expects a file *<templateQUEUE>.tcl* in the directory *<md_install_dir>/TaskServer/Tools*
  - **PBS** (Linux): works with *PBS*, *OpenPBS*, or *Torque*
  - **LSF** (Windows/Unix/Linux)
  - **GridEngine**
  - **SLURM**
  - **LoadLeveler**
  - **HPC**: A placeholder for your queueing script
  - **manual**: supports running on remote systems with manual file transfer. See the Section Creating a Manual TaskServer and Queue for more information.

Attention: *MedeA* does not include any of the above external queuing systems. You will need to install, configure, and manage the external queuing systems. Furthermore, provide a working example batch script for the queuing system that you want to use. Send any working example batch script to MD Support [2] . In return, you will receive a file *<QUEUE>.tcl* that your TaskServers can use to submit calculations to the HPC via the queuing system that you want to use.

To run e.g. with *PBS*, copy the file *templatePBS.tcl* in *TaskServer/Tools* to *PBS.tcl* and set the queue type to **PBS** in the TaskServer Administration page. The file *PBS.tcl* provides several parameters that can be fine-tuned to adapt the script to your HPC queueing environment. The script is general so it works with all compute engines in a general sense.

The *Installation Directory* is where the TaskServer is installed and cannot be changed from this Administration page.

The *Working Directory* is where the TaskServer creates temporary folders and files to run the calculations. This is a scratch directory as all relevant data are exported back to the JobServer upon completion of tasks.

The *Port* is the TaskServer port number. It is not recommended to change the **Port** that the TaskServer uses to communicate with the JobServer, unless there is a specific and inevitable reason for doing so.

With marking the checkbox of *Save files temporarily* and confirming with Apply task directories are kept temporarily for debugging purposes, such as configuring and tuning the submission to the external queueing system. Please note that this is a temporary switch and will be reset upon restarting the TaskServer.

In the green banner there are 3 sub-tabs:

[2] support@materialsdesign.com

- Users

  Advanced settings on JobServer and TaskServer security. See Section the Sharing and Security for more information.

- JobServers

  Advanced settings on JobServer and TaskServer security. See Section the Sharing and Security for more information.

- Log

  Contains the TaskServer log files which can be useful for debugging TaskServer issues.

5 Managing JobServer and TaskServer Services on Linux

This section provides some additional notes on managing the JobServer and TaskServer services on Linux, including verifying automatic start of services on Linux, checking and managing firewall settings, and checking and setting system limits.

5.1 Verify Automatic Start of Services on Linux

Use your Linux distributions' administration tools to verify that JobServer and TaskServer will start automatically after reboot. Check that the system service *mdjobserver* (if Installed) Starts automatically. On *RPM-based* systems (e.g. Ubuntu and Debian) the following commands are important:

```
chkconfig --list mdjobserver
```

```
mdjobserver 0:off    1:off    2:off    3:off    4:off   5:off    6:off
```

In this case, the system service *mdjobserver* is not going to start automatically. Let's change this for run levels 3 and 5:

```
chkconfig --level 35 mdjobserver on
```

```
chkconfig --list mdjobserver
```

```
mdjobserver   0:off    1:off    2:off    3:on     4:off   5:on     6:off
```

Check that the system service *mdtaskserver* (if Installed) Starts automatically

```
chkconfig --list mdtaskserver
```

```
mdtaskserver   0:off    1:off    2:off    3:off    4:off   5:off    6:off
```

In this case, the system service *mdtaskserver* is not going to start automatically, change this for runlevels 3 and 5:

```
chkconfig --level 35 mdtaskserver on

chkconfig --list mdtaskserver
```

```
mdtaskserver   0:off    1:off    2:off    3:on     4:off   5:on     6:off
```

On *SuSE* Linux use the runlevel editor of *YaST*. With *Debian-based* systems such as *Debian*, *Ubuntu*, and *Gentoo* the installation of system services is part of installing the package *mysqld*.

5.2 Check Firewall Settings and Allow Connections for JobServer and TaskServer

If you have a firewall in place, make sure that the TaskServer can be reached from the JobServer. With *iptables* in place, the TaskServer port can be opened with the following commands.

```
iptables -I INPUT -p tcp --dport 23000 -j ACCEPT
```

You can restrict access to the TaskServer to a single JobServer by specifying a network address:

```
iptables -I INPUT -p tcp --dport 23000 -j ACCEPT -s <IP of JobServer>
```

Check the current set of rules in place - and please talk to your IT department:

```
iptables --list
```

Chain INPUT (policy ACCEPT)

target prot opt source destination

ACCEPT tcp – anywhere anywhere tcp dpt:23000

To keep the set of rules working after the next reboot, you need to save them. This depends on the Linux version and specifics of your installation.

- RPM-based (*RedHat/Centos/Fedora/SUSE*):

  ```
  /etc/init.d/iptables save
  ```

  Similarly, you need to allow access on the JobServer from your GUI:

  ```
  iptables -I INPUT -p tcp --dport 32000 -j ACCEPT
  ```

- Debian-based (*Ubuntu/Debian/Mint*):

  ```
  iptables-save > /etc/iptables/rules.v4
  ```

  or

  ```
  ip6tables-save > /etc/iptables/rules.v6
  ```

  to store IPv4 or IPv6 rules, respectively.

  To automatically load the rules upon booting computers, install the package *iptables-persistent* which is part of the default software repositories:

  ```
  sudo apt-get update

  sudo apt-get install iptables-persistent
  ```

5.3 Check System Limits

Depending on model size and type of calculation, VASP and LAMMPS calculations can demand considerable amounts of memory, especially when running in parallel across many compute nodes. Please check whether the following properties are large enough, you can invoke the command *ulimit -a* in bash. The output of the command should be similar to

- *fsize*: maximum file size: should be *unlimited*
- *memlock*: maximum locked-in memory: 80-90% of total main memory
- *rss*: maximum memory size: should be *unlimited*

- *nofile*: number of open files: at least 20 per core
- *stack*: (memory) stack size: 80-90% of total main memory
- *cpu*: cpu time that can be used: should be *unlimited*
- *nproc*: max. processed per user: set to at least 1000
- *as*: virtual memory: should be *unlimited*
- *Locks*: file locks: should be *unlimited*

Modify */etc/security/limits.conf* to raise the impinging limits, here are examples:

```
* soft memlock unlimited
* hard memlock unlimited
* soft nofile 10240
* hard nofile 10240
* soft nproc 3153919
* hard nproc 3153919
```