

3DEQUALIZER 4

3DE4/MHP Manual

Network Calculation

Science-D-Visions, 2018-11-12

Contents

- 1 About 3DE4/MHP**
 - 1.1 Basic principle
 - 1.2 Versions
- 2 Quick guide for 3DE4 users**
 - 2.1 Preferences**
 - 2.1.1 *Linux, OSX*
 - 2.1.2 *Windows*
 - 2.2 Parameter Adjustment**
 - 2.3 Debugging / Support**
- 3 Quick guide for administrators**
 - 3.1 Static network**
 - 3.1.1 *Adding more nodes to the network*
 - 3.2 Automatic network**
 - 3.2.1 *Adding more nodes to the network*
- 4 Reference manual**
 - 4.1 The executable tde4mhp**
 - 4.1.1 *Starting the node*
 - 4.1.2 *Terminating the node*
 - 4.1.3 *Command line options, environment variables, configuration file*
 - 4.1.4 *Important options*
 - 4.1.5 *Less important options*
 - 4.1.5.1 *Configuration file*
 - 4.1.6 *Summary*
 - 4.2 Requirements and recommendations**
 - 4.2.1 *The owner of tde4mhp*
 - 4.2.2 *Environment variables \$PATH and \$HOME*
 - 4.2.3 *Shell commands*
 - 4.2.4 *Default route*
 - 4.2.5 *TCP, UDP, Ports, Firewall*
 - 4.2.5.1 *CentOS*
 - 4.2.5.2 *Ubuntu*
 - 4.3 Limitations**
 - 4.4 The html-interface**
 - 4.4.1 *The page hosts.html*
 - 4.4.2 *The page properties.html*

4.4.3 *The pages processes.html and processes_network.html*

4.4.4 *The page load.html*

4.5 Lens distortion plugins

5 Miscellaneous

5.1 Todo

5.2 An incomplete list of useful shell commands

5.3 Multi-host administration

1 About 3DE4/MHP

3DE4/MHP stands for *3DE4 Multi Host Processing*. It is a program which allows 3DE4 to do parameter adjustment on many hosts instead of a single host. Parameter adjustment in 3DE4 can take a long time, when many parameters have to be determined in a brute-force method. On a single host, this means a 3DE4-project has to be processed over and over again with modified parameters. This is usually done by multiple threads. In organizations with many seats, machines are often not used to their capacity, because for many operations only one CPU core is needed, while the remaining ones do little or nothing. These free capacities are utilized by 3DE4/MHP for parameter adjustment. Depending on the number of available machines this leads to a considerable performance gain. This manual describes how to set up and use 3DE4/MHP.

1.1 Basic principle

In practice, 3DE4/MHP is an executable `tde4mhp` which is either launched and controlled by means of 3DE4's GUI, or it runs a standalone program. A running instance of `tde4mhp` is called a node. Once started, the node tries to find other nodes in the local area network and form a network on the application layer, which we call the **MHP-network**. The nodes in this network communicate with each other over TCP/UDP. Each node also provides an http-interface, so that it can be inspected and modified by the administrator or 3DE4 user by means of a common webbrowser.

1.2 Versions

The version number of `tde4mhp` has the form

a . b . c

c is increased in order to indicate bug fixes and minor changes. b indicates changes in the underlying core technology of 3DE4. Nodes with same major version a but different subversion number b can be managed by the same chief. However, 3DE4 will only send jobs to nodes with matching major version and matching subversion number. A change in a indicates a severe change in protocols, which requires that each node is upgraded. We will try to avoid this. In this section we keep track of the development of `tde4mhp`.

Version	3DE4 Rel.	Released	Changes
1.19.0	3DE4r6b2		Bugfixes in 3DE4's core
1.18.0	3DE4r6b1		Adaptive load balancing, better handling of small projects
...			Changes in 3DE4 retraced
1.5.0			Bugfixes in 3DE4's core
1.4.0			Bugfixes in 3DE4's core
1.3.0			Lens models which depend on focal length and focus distance; point constraints
1.2.0			Bugfix: Sync + Lock Channels
1.1.0	3DE4r4b2	2014-05-09	Bugfix: <code>tde4mhp</code> more robust when host resolution is blocking. Bugfixes in 3DE4's core, most of them related to stereo and sync

Version	3DE4 Rel.	Released	Changes
			projects.
1.0.69	3DE4R4b1	2014-04-02	First release

Versions of this document:

Version	Released	Changes
5	2018-11-07	firewall commands under Linux, bugfixes, appearance
4	unpublished	multi-admin tool mentioned
3	unpublished	minor changes in design
2	2014-05-09	minor changes, typos
1	2014-04-09	first release

2 Quick guide for 3DE4 users

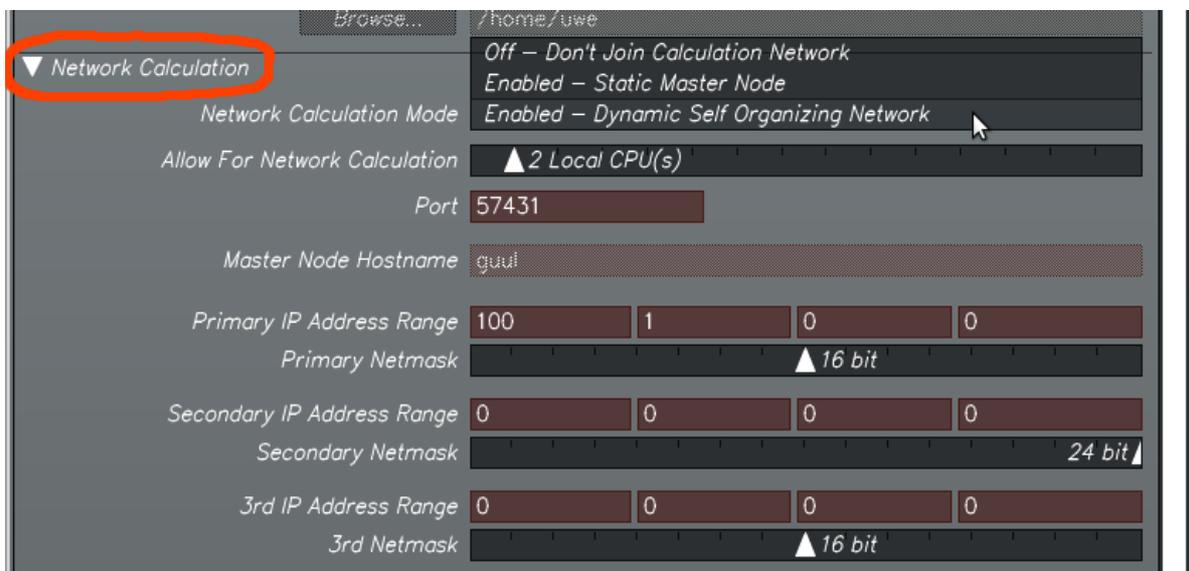
2.1 Preferences

2.1.1 Linux, OSX

Network calculations are enabled in 3DE4's preferences. If they are enabled, a `tde4mhp` node will run on your machine. There are two ways of connecting your node to nodes on other machines:

- *Enabled - Static Master Node* - In this mode you enter the name or ip-address of the chief node in the text field *Master Node Hostname*. If your organization is using dedicated hosts for network calculations, the system administrator will give you the details.
- *Enabled - Dynamic Self Organizing Network* - In this mode your node will try to find other nodes by scanning the local area network. In the ideal case this will take a few seconds, but it could also take much longer. We have put some effort into finding other nodes without burdening your network, so we think it's worth a try.

By the slider *Allow For Network Calculation* you specify how many of your CPU cores are provided to the community so that they can work on jobs submitted by other users in the MHP-network. The remaining ones are reserved for jobs initiated by yourself. It's not easy to give a reasonable default value for this. On machines with more than 4 CPU cores we recommend to keep at least 4 CPU cores private.





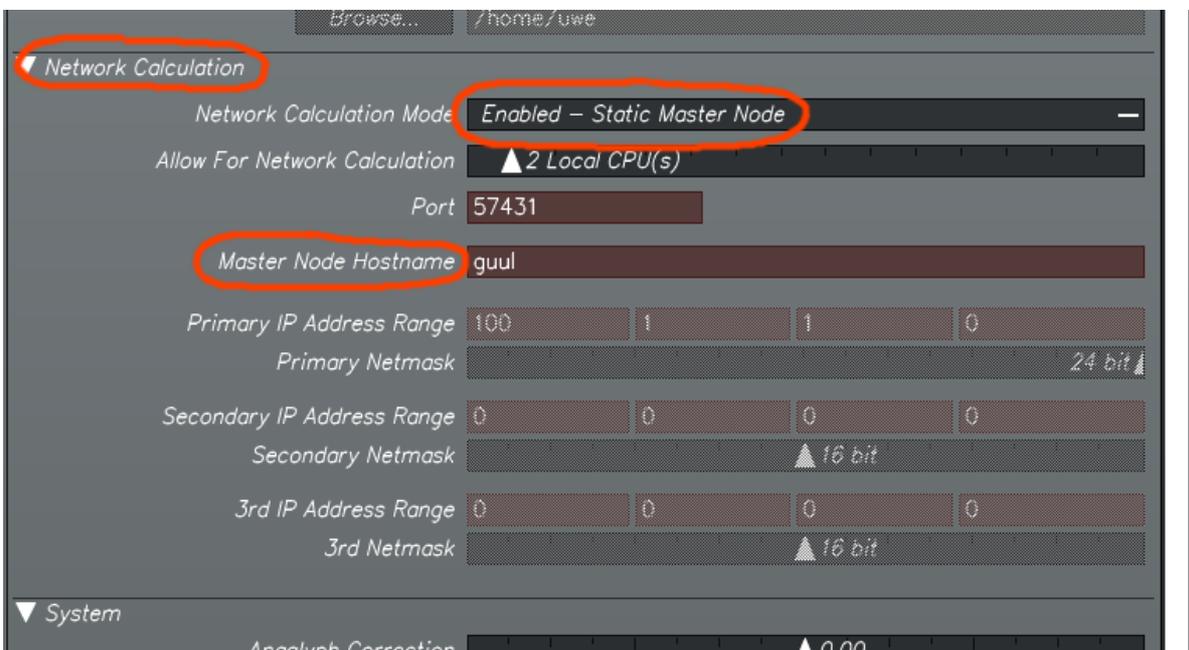
The preferences allow to specify up to three address ranges to scan for other nodes. The primary ip address range should already contain reasonable default values, namely the netmask of the local area network your host belongs to. Typical default values here are:

- 10.0.0.0, 8 bit
- 172.16.0.0 to 172.31.0.0, 12 bit
- 192.168.0.0, 16 bit
- 192.168.0.0 to 192.168.255.0, 24 bit

Generally, there should not be reason to modify these values.

2.1.2 Windows

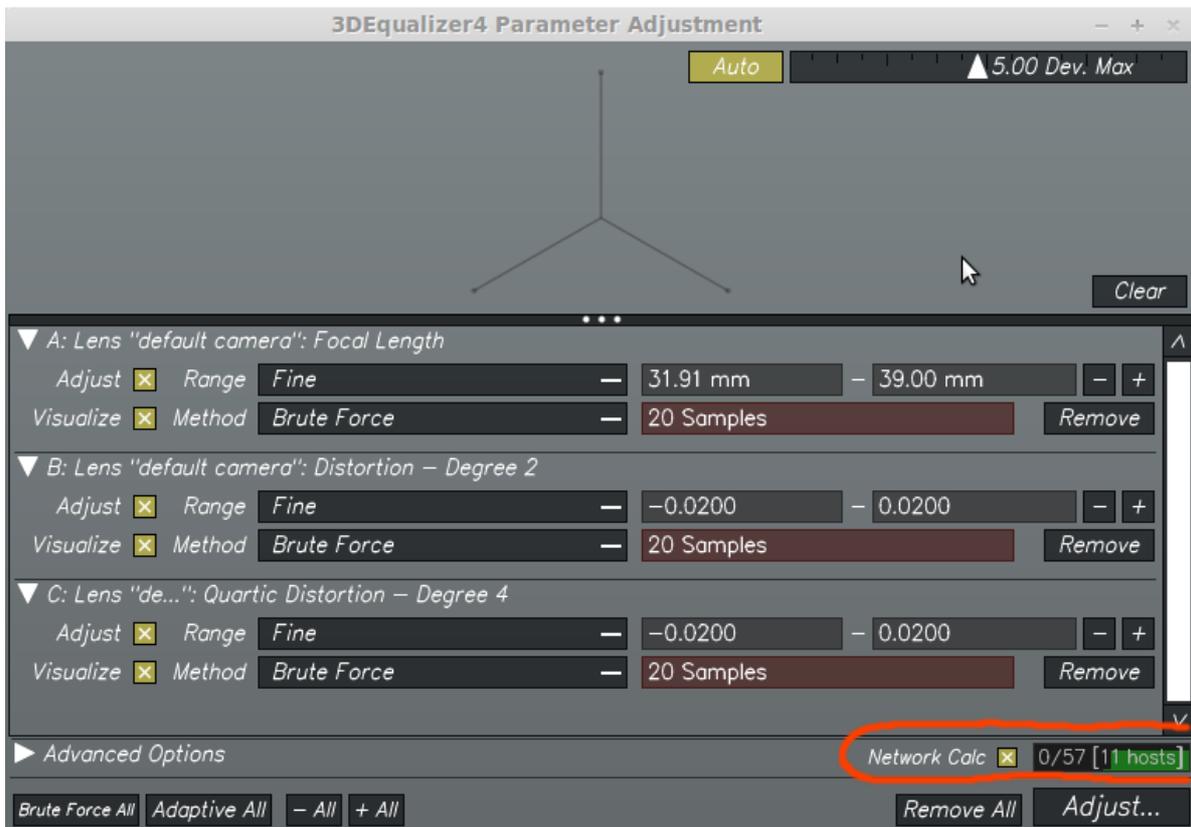
In the current release of 3DE4/MHP there is no `tde4mhp` node running on Windows. Nonetheless, 3DE4 users working on Windows can connect to existing MHP-networks. If your company decides to use 3DE4/MHP your system administrator should set up a static MHP-network on Linux and/or OSX machines. Once this network is up and running, you only need to enter the name or ip-address of any of the hosts participating in the network in the text field *Master Node Hostname* in 3DE4's preferences:



For calculations, 3DE4 will use all community threads available in the MHP-network plus all CPU cores of the local host you are working on.

2.2 Parameter Adjustment

The main purpose of 3DE4/MHP is to accelerate parameter adjustment. In the *Parameter Adjustment Window* you see a small section named *Network Calc* underneath the parameter entries. The toggle button enables or disables the network calculation. If it is turned off, other users will still be able to use the common CPU cores on your machine. If it is turned on, any parameter adjustment you start will be submitted to the MHP-network. The numbers (here "0/57 [11 hosts]") indicate how many threads are working on your job (here: 0) and how many threads there are at your service (here: 57), provided by 11 hosts. The second number is the number of public threads / CPU cores plus your private threads / CPU cores. The green bar shows that someone else is currently doing calculations in the network.



3DE4/MHP is clearly focused on brute-force calculations. For brute-force adjustment, network calculations are accelerated considerably, although we do not have reliable numbers on the performance scaling behaviour. For adaptive adjustment, network calculations are possible, but they are accelerated only between 10% to 70%, sometimes 100%, much less than brute-force adjustments.

2.3 Debugging / Support

The most critical part of 3DE4/MHP on the client side, i.e. in connection with 3DE4 is the startup and shutdown of the node and connecting it to other nodes. 3DE4 examines the environment variable **TDE4MHP_CLIENT_LOG** and writes debugging data to the file specified there. If you send us a file with these data, this will help us a lot getting it running. If you are curious, set this to **/dev/tty** on Linux/OSX or to **CON** on Windows...

3 Quick guide for administrators

First of all, there should be as little work as possible for the administrator. The entire distributed computing business is useless if all resources (i.e. processing time) saved in one part of the company need to be invested somewhere else (administration time). But if work needs to be done, there should be appropriate tools. For this reason, we have equipped the nodes with a simple webservice functionality, so that the current state of each node or the entire MHP-network can be examined and modified in a common webbrowser. Most of the time during development, we have been using Firefox, but Chrome, Opera or Safari should do as well. The browser interface is explained in detail in section 4.4.

Although this is meant to be the quick guide, we will have to define a few things. When the user starts 3DE4 with network calculations enabled, a **tde4mhp** node will start as well. This node is controlled via 3DE4's GUI, and therefore we call it a controlled node. On the other hand, it might be interesting to increase computational power by adding more machines without running 3DE4. Nodes running on hosts without 3DE4 are controlled over the browser interface or by means of scripts. We will call these nodes standalone nodes in the following.


```

| flcd_port          | 57423          | default  |
+-----+-----+-----+
| num_cores         | 8              | system   |
| num_threads_comm  | 8              | default  |
| num_threads_priv  | 0              | default  |
+-----+-----+-----+
| install_dir       | /guul/server/software/linux64/tde4mhp | installation |
| plugin_dir        | <install_dir> / | undefined  |
| tmp_dir           | /tmp/          | default   |
| home_dir          | /home/uwe/.3dequalizer/ | default   |
| config_file       |                | default   |
| logfile           | /tmp/tde4mhp.u753.log | default   |
| loglevel          | LOG_NOTICE     | default   |
+-----+-----+-----+
| executable path   | ./tde4mhp     | installation |
| full logfile path | /tmp/tde4mhp.u753.log |
+-----+-----+-----+
uwe@hellboy software/linux64/tde4mhp>

```

Since there is no error message the daemon should be running now. Let's have a look:

```

uwe@hellboy software/linux64/tde4mhp> ps -ef | grep tde4mhp
uwe      10160    1  0 15:16 ?        00:00:00 ./tde4mhp
uwe      10170  10160  0 15:16 ?        00:00:00 ./tde4mhp
uwe      10174  10090  0 15:20 pts/1    00:00:00 grep tde4mhp

```

In our example there are two processes, with ids 10160 and 10170. The daemon has spawned a child process (10170) which receives messages from other nodes. We can ignore that. If we try to start the daemon once more, we get an error message, because only one daemon is supposed to run at a time.

```

uwe@hellboy software/linux64/tde4mhp> ./tde4mhp
There already seems to be a node running on this host
with process id 10160. Please remove this process first
by 'kill 10160' and then try again.

```

From the daemon's output we see that hellboy has eight CPU cores, and all of them are dedicated to the community. This makes sense for a standalone node. Without any 3DE4 user working directly on the machine there is no need for private CPU cores. What we also see is, that the network mode is explicit which means, the node will not try to find other nodes. Our node now forms a simple network, being the chief itself, and other nodes can check in. That's it in principle. It should be mentioned, that the chief has to master more network traffic than other nodes. It should be a machine with high availability, like the license server.

3.1.1 Adding more nodes to the network

Once we have a running chief, it is easy to add more standalone nodes. We start a daemon on a machine named `smaug` and connect it to the chief node on `hellboy`:

```

uwe@smaug /home/uwe> cd /server/software/linux64/tde4mhp/
uwe@smaug software/linux64/tde4mhp> ./tde4mhp -chief_host hellboy

```

Again, the daemon will output a lot of information, but the interesting part is this one:

```

...
| chief_host        | hellboy        | cmdline   |
| chief_port        | 57431         | default   |
...
| num_cores         | 8              | system    |
| num_threads_comm  | 8              | default   |
| num_threads_priv  | 0              | default   |
...

```

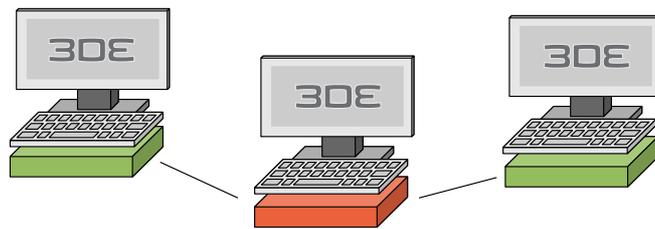
Our network now consists of two machines, with 16 public CPU cores. At this point, although this is still the quick guide, it's a good idea to know about the browser interface. At address **100.1.1.34:57431/hosts.html** we see the following now:

host:port	hostname	proc	msg	launched as	role	threads	uptime	version	perf	ops
100.1.1.34:57431	hellboy	daemon	ok	standalone	chief	8 / 0	1h 5m	1.0.69	0.0%	🔄 ⏻
100.1.1.40:57431	smaug	daemon	ok	standalone	loyal	8 / 0	11m 9s	1.0.69	0.0%	🔄 ⏻ ⚡

Starting at this point you can browse through your MHP-network, configure and repair it, if necessary. For details please read section 4.4.

3.2 Automatic network

When the 3DE4 users start their program, a `tde4mhp` node is started as well. The characteristic property of the automatic network is, as the name indicates, that in principle it does not require any other setup. However, in practice it makes sense to check if all nodes are running as they are supposed to. Also, even in an automatic network you as the administrator can still add standalone nodes in order to increase computational power.



Let us assume that there are three instances of 3DE4 now running on machines with 8 CPU cores each, all have enabled network calculation in their preferences and provide 4 CPU cores to the community. Then, in the *Parameter Adjustment Window* the users should see the following:



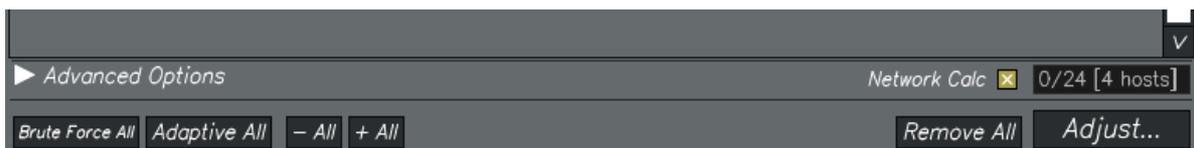
The display shows 16 CPU cores: 4 from each of the three machines plus 4 private cores from the machine we took this snapshot from. Now let's add some computational power to this network.

3.2.1 Adding more nodes to the network

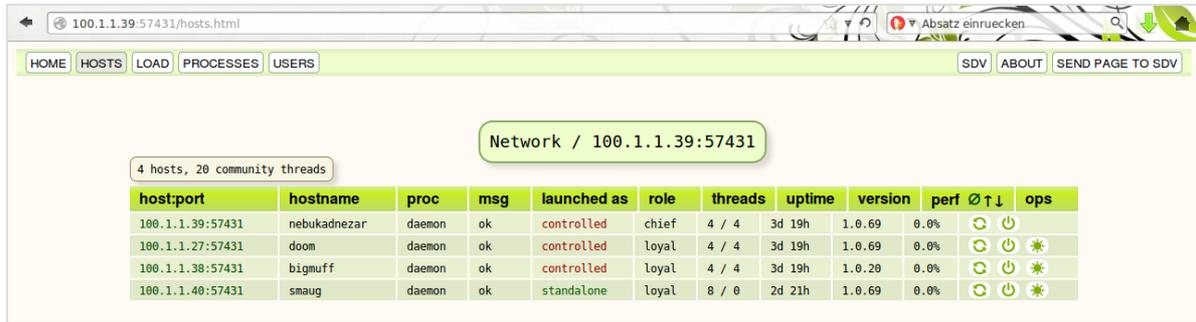
The license server FLCD is well-informed about the 3DE4 instances. We can benefit from this by passing the FLCD host (doom in the example below) to `tde4mhp` on the command line. FLCD will send a list of hosts running 3DE4. From this list the node to be started will find out who is chief and connect `tde4mhp` directly. The port specification could be omitted, since it's the default for FLCD.

```
uwe@smaug> ./tde4mhp -network_mode automatic -flcd_host doom -flcd_port 57423
```

Assuming that smaug offers 8 CPU cores to the community, 3DE4's *Parameter Adjustment Window* should display now 24 CPU cores on 4 hosts.



In the hosts overview of the browser interface we now see the following.

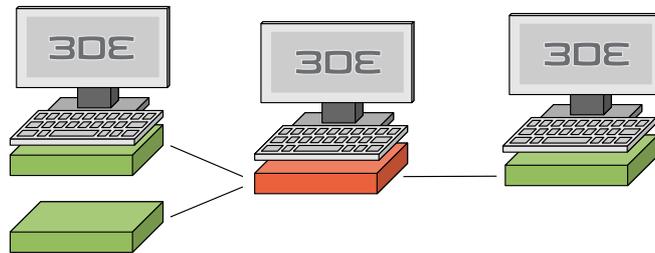


Network / 100.1.1.39:57431

4 hosts, 20 community threads

host:port	hostname	proc	msg	launched as	role	threads	uptime	version	perf	ops
100.1.1.39:57431	nebukadnezar	daemon	ok	controlled	chief	4 / 4	3d 19h	1.0.69	0.0%	⏪ ⏩ ⏻
100.1.1.27:57431	doom	daemon	ok	controlled	loyal	4 / 4	3d 19h	1.0.69	0.0%	⏪ ⏩ ⏻ ⚙
100.1.1.38:57431	bigmuff	daemon	ok	controlled	loyal	4 / 4	3d 19h	1.0.20	0.0%	⏪ ⏩ ⏻ ⚙
100.1.1.40:57431	smaug	daemon	ok	standalone	loyal	8 / 0	2d 21h	1.0.69	0.0%	⏪ ⏩ ⏻ ⚙

Or graphically:



In principle you can build up a large network like this, but in that case it might be good idea to bypass the network scanning procedure and run the chief as described in section 3.1 instead.

4 Reference manual

4.1 The executable tde4mhp

The executable `tde4mhp` is located in 3DE4's installation directory at

`${3DE4}/bin/tde4mhp`

in the same directory as the 3DE4 executable itself. In order to be started as a controlled node by 3DE4 it must reside in this directory. For standalone nodes you can move it wherever you like. If you invoke `tde4mhp` and pass `help` or `-help`, the node will only write an explanation text about its options and exit.

4.1.1 Starting the node

Let us consider controlled nodes first. When 3DE4 is started with network calculations enabled, `tde4mhp` is started as well. When 3DE4 is terminated, the node keeps on running until it is terminated by intervention of the administrator. Usually, a node controlled by 3DE4 is only terminated, if network calculations are disabled in 3DE4's preferences.

The case of standalone nodes is more interesting for the administrator, so we shall discuss this case in the following. We do this by means of examples, so let us assume our current directory contains the executable of `tde4mhp`. The following command will start 3DE4/MHP as a daemon. It will form a monadic network, consisting only of itself (unless other nodes connect to it). This is the common way to start the chief node in a static setup.

```
uwe@hellboy> ./tde4mhp
```

The following will start the node in foreground mode. This could be interesting for testing and debugging purposes. When started this way, the node can be terminated by `Ctrl` `C`. The process mode can be combined with any other option, so we won't list all the combinations here.

```
uwe@smaug> ./tde4mhp -process_mode foreground
```

The following command will start the node as daemon and connect it to the chief hellboy. Since 57431 is the default port, the option `-port` can be omitted here.

```
uwe@smaug> ./tde4mhp -chief_host hellboy -chief_port 57431
```

Assume, you have no idea, on which host the chief is running (and you're not in the mood to find out). Then you can use the automatic network mode and consult the 3DE4 license server FLCD (in our case it's called doom). Again, the option `-flcd_port` could be omitted, since 57423 is the default value for FLCD.

```
uwe@smaug> ./tde4mhp -network_mode automatic -flcd_host doom -flcd_port 57423
```

By the following command, the node will scan the network address range 192.168.32.1 to 192.168.63.254 in order to find the chief or other nodes.

```
uwe@smaug> ./tde4mhp -network_mode automatic -subnet_a 192.168.32.0/20
```

This command will start the node as daemon, read the configuration file and set the thread numbers to 8 and 0, regardless of the values given in the configuration file.

```
uwe@smaug> ./tde4mhp -config_file /server/etc/tde4mhp/tde4mhp.conf -num_threads_comm 8 -num_threads_priv 0
```

4.1.2 Terminating the node

There are several ways to terminate a running node. A node is terminated by setting the network mode to *Off* in 3DE4's preferences (if it is a controlled node), or you terminate the node via the web interface. In both cases, the MHP-network will keep consistent. If you are terminating the chief node, the chief role will be transferred to another node, so that the network keeps consistent and operational. Ideally, this would be transparent for the user of 3DE4. Please have a look at section 4.4.2 for details.

You can also terminate 3DE4/MHP from the command line, but in this case the chief role (if your node is chief) is not transferred automatically to another node. Let us assume, your node is not chief. Then a node running in process mode foreground is simply terminated by `Ctrl C`. This raises the signal SIGTERM, which allows the node to check out at the chief and clean up whatever needs to be cleaned up. A node running in process mode daemon has no connection to any terminal. At command line you can terminate it by

```
kill <pid>
```

which by default sends SIGTERM to the process. A more severe method, which cannot be refused by the process and does not allow the node to clean up is

```
kill -s KILL <pid>
```

or equivalently

```
kill -9 <pid>
```

This might leave the MHP-network in an inconsistent state and should be avoided.

4.1.3 Command line options, environment variables, configuration file

tde4mhp can be configured in three different ways: by passing options at the command line, by defining environment variables and by configuration file. These three channels can be combined according to your needs. Command line options have the highest priority; any parameter given as environment variable or configuration file is overruled by its command line version.

The names of most command line options, environment variables and configuration file option names are related as in the following example:

Environment variable	Command line option	Configuration file entry
TDE4MHP_NETWORK_MODE	-network_mode	network_mode

There are only a few exceptions: `-help`, `-config_file` and `-install_dir` cannot appear in the configuration file (since that makes little sense). Whenever we talk about an option in the following, we will address it by the command line version. Implicitly, by doing so we also refer to the environment variable and the configuration file entry.

4.1.4 Important options

- `-port` - The port the node is listening to. The default value is 57431. Since 3DE4 does not allow to specify different ports for its own, controlled node and for the chief node, the port number will be the same for all nodes within an MHP-network.
- `-chief_host`, `-chief_port` - A node can scan the network in order to find other nodes or it can connect directly to the chief node. By this option the chief host and port are specified. Since the default value for the chief port is 57431, this option can often be omitted.
- `-num_threads_comm`, `-num_threads_priv` - A host has a certain number of CPU cores, say n . Assuming that under full load one core corresponds to one thread, threads are divided into community threads and private threads. Community threads will work on jobs sent to the node by any user participating in the MHP-network, while private threads are reserved for the user working on the machine (i.e. using 3DE4 or other software). For controlled nodes it will depend on company guidelines how many threads users should provide to the community, maybe 25%, maybe up to 50%. For standalone nodes it makes sense to have n community threads and 0 private threads. On some OSX machines with 16 cores (8 "real" and 8 "virtual") it makes sense not to use all cores, but only 8 of them, because additional cores will not increase performance. We have verified this on several Mac Pros running OSX 10.6.
- `-process_mode` - The process mode determines how the process is running within the UNIX environment, in the foreground (i.e. connected to a terminal, responding to keyboard events, with standard output and error streams), background (still with standard output and error stream) or daemon (detached from the terminal and without standard streams). For testing and debugging, process mode foreground is often helpful. In a fully operational network the process mode should be daemon (which is the default).
- `-network_mode` - This option can assume the following values: `explicit` (the default) and `automatic`. In explicit mode the node is either started as chief and forms a monadic network or the chief node the node is supposed to connect to is specified by means of `-chief_host` and `-chief_port`. In automatic mode the node will scan the network and try to find other nodes in order to form a network. This scanning process is supported by other options like `-known_hosts` and (most important) `-flcd_host` and `-flcd_port` (see below).
- `-flcd_host`, `-flcd_port` - This option is only relevant in automatic network mode. Without any hint the node is forced to scan the entire network for other nodes, which is done very slowly in order not to overburden the network. By this option you allow the node to connect to 3DE4's license server FLCD and fetch a list of starting points to scan. If any 3DE4 user already has a running `tde4mhp` node it will be found immediately due to this option.
- `-plugin_dir` - This option is relevant if you are using your own lens distortion models (see section 4.5).

4.1.5 Less important options

- `-subnet_a`, `-subnet_b`, `-subnet_c` - These options correspond to *Primary/Secondary/3rd IP Address Range* in 3DE4's preferences. They allow you to specify address ranges for the automatic network mode. By default, `subnet_a` refers to the entire local area network of your host. As mentioned, scanning the entire net can take a long time. For this reason it makes sense to restrict the set of ip-addresses to be scanned. Values for this option are given in the standard dot notation, e.g. `192.168.45.0/24`, where `/24` indicates that the first 3 numbers (24 bit) are fixed while the last one is

iterated through. The `tde4mhp` node will first scan address range *a*, then *b* and then *c*. Another example: Let us assume your local area network has the network address `172.16.0.0/12`. Your camera department including 3DE4 seats is located in the subnet `172.16.32.0/20` and you wish to add nodes in this subnet. Then, during startup the node will determine the default value for `subnet_a` as `172.16.0.0/12`, but it makes sense to set it to `172.16.32.0/20` which is scanned within a few minutes.

- `-tmp_dir` - The directory for temporary data. It should point to a directory on a local drive, not on an NFS-mounted volume. The default is `/tmp`, and we recommend to leave it as it is.
- `-logfile` - The path to the logfile. By default this is located in `/tmp`. Alternatively, you could place it in `/var/tmp`, but wherever you choose it to be, it should be on the local drive of your host, **not(!)** on an NFS-mounted volume. Also, it must be a regular file, not a special one like `/dev/tty` or the like. The node will truncate the logfile automatically as soon as its size exceeds 1MB. The precise default value is `/tmp/tde4mhp.u{uid}.log` where `{uid}` stands for the id of the user running the node. We recommend to leave this as it is.
- `-loglevel` - Logged messages have a level of "severity". For 3DE4/MHP we adapted the official UNIX levels ranging from `LOG_DEBUG` up to `LOG_EMERG`, see `man syslog` for details. Default is `LOG_NOTICE`. Only messages with the given severity level or higher are logged, all others are discarded. In case of trouble, if you need support from us, we might ask you to set this to `LOG_DEBUG` so that we have a better chance to solve the problem.
- `-launched_by` - 3DE4 passes some string here to indicate that the node is controlled. If you start `tde4mhp` as a standalone node, please leave this empty.
- `-fileservers_base` - This is an option for future applications running on 3DE4/MHP which need access to a file server.
- `-known_hosts` - This is another hint for the automatic network mode. 3DE4 uses this option and passes a comma-separated lists of hosts obtained from the license server FLCD. You can use this option for standalone nodes as well, but in general it will be easier for you to pass the FLCD host by option `-flcd_host` instead.
- `-host` - When a node is started, it will determine its own ip-address by analyzing network devices and the routing table. If this should fail for what reason ever, you can bypass this mechanism by this option. In that case please specify the ip-address here.
- `-install_dir` - As you have noted, some options specify paths in the file system, e.g. the directory for lens distortion plugins or the location of a configuration file. These paths can be absolute (i.e. starting with `"/"`) or relative. If they are relative, they are interpreted as relative to the "installation directory" which can be specified by `-install_dir`. If you do not provide this option, the path to the executable `tde4mhp` is used as base for relative paths. Our suggestion in order to keep it simple, is not to use this option and use absolute paths wherever paths are required.

4.1.5.1 Configuration file

The configuration filename is specified by passing option `-config_file <path/to/configfile>` or by defining the environment variable `TDE4MHP_CONFIG_FILE`. A configuration file has the following simple format:

```
<option>      <value>
<option>      <value>
...
<option>      <value>
```

Lines starting with `#` are ignored, i.e. comments are marked by leading `#`.

4.1.6 Summary

Environment variables	Command line options	Configuration file entry	Possible values
<code>TDE4MHP_HOST</code>	<code>-host</code>	host	ip address e.g. 192.168.1.32

Environment variables	Command line options	Configuration file entry	Possible values
TDE4MHP_PORT	-port	port	port number, default is 57431
TDE4MHP_CHIEF_HOST	-chief_host	chief_host	ip address
TDE4MHP_CHIEF_PORT	-chief_port	chief_port	port number
TDE4MHP_PROCESS_MODE	-process_mode	process_mode	"foreground", "background", "daemon"
TDE4MHP_NETWORK_MODE	-network_mode	network_mode	"automatic", "explicit"
TDE4MHP_SUBNET_A	-subnet_a	subnet_a	net address e.g. 192.168.1.0/24
TDE4MHP_SUBNET_B	-subnet_b	subnet_b	net address e.g. 192.168.1.0/24
TDE4MHP_SUBNET_C	-subnet_c	subnet_c	net address e.g. 192.168.1.0/24
TDE4MHP_PLUGIN_DIR	-plugin_dir	plugin_dir	path/to/plugin/dir/
TDE4MHP_TMP_DIR	-tmp_dir	tmp_dir	/tmp, /var/tmp
TDE4MHP_LOGFILE	-logfile	logfile	/some/local/file
TDE4MHP_LOGLEVEL	-loglevel	loglevel	LOG_DEBUG, LOG_INFO, LOG_NOTICE, LOG_WARNING, LOG_ERR, LOG_CRIT, LOG_ALERT, LOG_EMERG
TDE4MHP_NUM_THREADS_COMM	-num_threads_comm	num_threads_comm	integer number
TDE4MHP_NUM_THREADS_PRIV	-num_threads_priv	num_threads_priv	integer number
TDE4MHP_LAUNCHED_BY	-launched_by	launched_by	string
TDE4MHP_FILESERVER_BASE	-fileserver_base	fileserver_base	/mount/point /of/fileserver
TDE4MHP_HOME_DIR	-home_dir	home_dir	directory, like \$HOME/.3dequalizer
TDE4MHP_KNOWN_HOSTS	-known_hosts	known_hosts	comma-separated list of hostnames
TDE4MHP_FLCD_HOST	-flcd_host	flcd_host	hostname
TDE4MHP_FLCD_PORT	-flcd_port	flcd_port	port number, default is 57423
TDE4MHP_CONFIG_FILE	-config_file	- n.a. -	path
TDE4MHP_INSTALL_DIR	-install_dir	- n.a. -	base for relative paths
- n.a. -	-help / help	- n.a. -	- n.a. -

Environment variables	Command line options	Configuration file entry	Possible values
LD_PLUGINS_3DE4	- n.a. -	- n.a. -	Additional directory for lens distortion plugins

4.2 Requirements and recommendations

4.2.1 The owner of tde4mhp



We recommend, **not** to run tde4mhp as user root.

No resources accessed by tde4mhp require root privileges, and you are on the safe side if you run it as a standard user or even create a particular under-privileged user account.

4.2.2 Environment variables \$PATH and \$HOME

The node needs to know the full path of its executable file. This is determined by means of the environment variable **\$PATH**, which must be present in the shell from which tde4mhp is started. This is usually not a problem on UNIX-like systems in higher run levels (Linux terminology), but if this is not the case, you can alternatively start tde4mhp by its full, absolute path, so that the environment variable is not required.

tde4mhp reads the environment variable **\$HOME** and creates a directory **.3dequalizer** for non-temporary data. If for what reason ever, this environment variable does not exist, please use the option **-home_dir** and set it to **/some/existing/directory/.3dequalizer**. If **\$HOME** exists, please do not use this option.

4.2.3 Shell commands

A certain set of shell commands must be present, since they are invoked by tde4mhp. These commands are

- **awk** (POSIX.1)
- **grep** (POSIX.1)
- **killall**
- **kill** (POSIX.1)
- **ps** (POSIX.1)
- **uname** (POSIX.1)
- **wc** (POSIX.1)
- **/sbin/ifconfig** (required on OSX)
- **/sbin/ip** (required on Linux)
- **/sbin/route**

Not all of these commands seem to be mandatory by the POSIX standard, however, according to our experience they are present on Linux and/or OSX. An easy way to check if all commands are available, is to use the **which**-command. Under Linux:

```
which awk grep killall kill ps uname wc ifconfig ip route
```

and under OSX (without ip):

```
which awk grep killall kill ps uname wc ifconfig route
```

4.2.4 Default route

It is strongly recommended that each host running a node has a default route. Usually this is the case, nonetheless we mention it here for reasons of completeness. The default route is required because the nodes are identified by the ip-address of the interface associated to the default route. Example: `/sbin/route` generates the following output:

```
Kernel IP routing table
Destination Gateway Genmask Flags Metric Ref Use Iface
10.211.55.0 * 255.255.255.0 U 0 0 0 eth0
link-local * 255.255.0.0 U 0 0 0 eth0
loopback * 255.0.0.0 U 0 0 0 lo
default 10.211.55.1 0.0.0.0 UG 0 0 0 eth0
```

Then, from `/sbin/ip route get 10.211.55.1` the node will find out that `10.211.55.7` is the ip-address of the interface leading to the standard gateway:

```
10.211.55.1 dev eth0 src 10.211.55.7
cache mtu 1500 advmss 1460 hoplimit 64
```

so, the ID of the node will be `10.211.55.7`.

4.2.5 TCP, UDP, Ports, Firewall

Since communication can go in both directions between all nodes of the MHP-network (where the roles of server and client may change), all hosts must allow **incoming and outgoing** TCP and UDP connections for the port used in the MHP-network. Please make sure, the firewalls on all participating hosts allow this. On Linux methods for modifying the firewall may vary between distributions. In this section we will at least try to cover a few methods. Note that

 All methods in the following subsections require administrative permissions. Modifying the firewall will affect the security of the hosts in your network, so make sure you fully understand the consequences of these commands.

4.2.5.1 CentOS

On **CentOS** the firewall can be modified using the command `firewall-cmd`. The option `--permanent` ensures that the reconfiguration will survive a reboot of the host. The option `--zone` could look different in your case, please consult the documentation in case of trouble.

```
linux> sudo firewall-cmd --zone=public --add-port=57431/tcp --permanent
linux> sudo firewall-cmd --zone=public --add-port=57431/udp --permanent
linux> sudo firewall-cmd --reload
```

4.2.5.2 Ubuntu

On **Ubuntu** and its derivatives there should be a command `ufw` which comes in quite handy. Assuming you are authorized to use `sudo` you could try the following: 1. Check status of the firewall

```
linux> sudo ufw status
```

If the firewall is inactive (which is not recommended) all ports are open and mhp should work. If the firewall is active, then: 2. Open tcp/udp-port 57431 (or which port ever you are using for your mhp-network)

```
linux> sudo ufw allow 57431
```

4.3 Limitations

It is not clear how many nodes an MHP-network can have. We think there is an upper boundary of 256 hosts participating in the

network for technical reasons, which we are however not able to verify or falsify. Also, there are practical reasons which advise, that the number of hosts should be lower. For example, instead of building one large network of 256 machines we suggest to build two networks made of 128 hosts or four networks made of 64 hosts, each network running on a different port in order to avoid interference. Whatever you try, it makes sense to start with a small number of machines and increase it step by step. Your feedback is appreciated...

4.4 The html-interface

Each node responds to browser requests on the same port they use for communicating with each other. Some of the html-pages describe the node they were requested from, others will give you information about the MHP-network.

4.4.1 The page `hosts.html`

This page gives you information about the entire network.

host:port	hostname	proc	msg	launched as	role	threads	uptime	version	perf	ops
100.1.1.39:57431	nebukadnezar	daemon	ok	controlled	chief	2 / 14	25m 46s	1.0.69	21.5%	🔄 🔌 ⚡
100.1.1.20:57431	neo	daemon	ok	standalone	loyal	4 / 0	1m	1.0.69	0.0%	🔄 🔌 ⚡
100.1.1.23:57431	morpheus	daemon	ok	standalone	loyal	4 / 0	6s	1.0.69	0.0%	🔄 🔌 ⚡
100.1.1.24:57431	guul	daemon	ok	standalone	loyal	2 / 0	11m 6s	1.0.69	84.0%	🔄 🔌 ⚡
100.1.1.26:57431	lear.fritz.box	daemon	ok	standalone	loyal	2 / 0	10m 41s	1.0.20	47.0%	🔄 🔌 ⚡
100.1.1.27:57431	doom	daemon	ok	controlled	loyal	2 / 6	25m 49s	1.0.69	40.7%	🔄 🔌 ⚡
100.1.1.34:57431	hellboy	daemon	ok	standalone	loyal	8 / 0	8m 32s	1.0.69	80.3%	🔄 🔌 ⚡
100.1.1.35:57431	hydra.fritz.box	daemon	ok	standalone	loyal	8 / 0	15m 26s	1.0.20	12.7%	🔄 🔌 ⚡
100.1.1.36:57431	octopus	daemon	ok	standalone	loyal	8 / 0	15m 18s	1.0.69	70.7%	🔄 🔌 ⚡
100.1.1.38:57431	bigmuff	daemon	ok	controlled	loyal	2 / 2	25m 13s	1.0.20	51.7%	🔄 🔌 ⚡
100.1.1.40:57431	smaug	daemon	ok	controlled	loyal	2 / 6	25m 29s	1.0.69	76.8%	🔄 🔌 ⚡
100.1.1.4:57431	devil	daemon	ok	standalone	loyal	4 / 0	8m 56s	1.0.69	100.0%	🔄 🔌 ⚡
100.1.1.70:57431	schreck.local	daemon	ok	standalone	loyal	2 / 0	10m 10s	1.0.20	64.0%	🔄 🔌 ⚡

Column	Content
host:port	In this column you see the ip-address and the port number of each node. It is advisable to use the same port number for all hosts in the network. In 3DE4/MHP hosts are not identified by their hostname, instead we use the ip-address of the interface pointing to the standard gateway (the default route). This makes the network resistant against spontaneous renaming by the DHCP server or the like.
hostname	This column shows the human-readable hostname generated by the host (man 3 gethostbyname).
proc	The column "proc" displays how the node was started, as foreground process, as background process or as a daemon. Running a node in process mode "foreground" is helpful for learning and debugging puposes. For daily use, console output for debugging is no longer required, and nodes should be run as daemons.
msg	This field contains the last status report from network connection between chief node and all other nodes. Whenever hosts.html is requested, the chief will fetch properties from all nodes (including itself). After some timeout, all collected data are displayed. Failed connections lead to an error message here.
launched_as	This column shows if this is a controlled node or a standalone node.

Column	Content
role	This column tells you if the node is chief, or loyal (normal node). The third possible state is renegade which appears, when a node has left the network unexpectedly without checking out. Renegades should be removed or reintegrated into the network. When the network connection fails due to a timeout, the role is labelled unclear.
threads	This column displays the number of threads dedicated to the community and the number of private threads. For standalone nodes it makes sense to make all resources available to the community. Since 3DE4 is not running on these hosts, private threads are not needed.
uptime	This column displays the time the node has been running.
version	In the introduction we have already mentioned that only nodes having the same major version and subversion number can collaborate, so it's important to display the version here.
perf	After each parameter adjustment, the client collects the times for processing the jobs are averaged and mapped into a relative performance values. These values are sent to the chief and averaged over the previous ten projects. When the performance values are displayed, the fastest host is labelled with "100%" and all other hosts accordingly.
op	The operations column contains buttons for restarting or terminating a node. Loyal nodes have a button which turns the node into the chief. In mixed networks with controlled and standalone nodes, it makes sense to associate the chief role to a standalone node, in order to release the controlled node from network traffic.

4.4.2 The page `properties.html`

This page is requested for a specific host you wish to receive information about. It also provides functions to modify the structure of the network.

name	value	def'd by	mod
- Process -			
executable	tde4mhp		
version	1.0.0		
uptime	7,58451		
pid	13660		
process_mode	foreground	cmdline	
launched_by		default	
owned_by	uwe	default	
- Network -			
host	10.211.55.7	system	
port	57431	default	
chief_host	10.211.55.7	default	
chief_port	57431	default	
chief_status	ok		
is_chief	yes		
network_mode	explicit	default	

Button	Effect
RESTART	Restart the node. The node will keep its settings. If it has been the chief before restart, it will be chief after restart as well, so this operation should keep the network consistent. Technically, this is done by sending the signal SIGUSR1 to the node process. The operation

Button	Effect
	makes sense, when the software has been updated. There seem to be problems sometimes on OSX, when the executable is located on an NFS-mounted volume.
TERM	This will terminate the node regularly. When the node has been chief before termination, the chief role is moved to another node and all other nodes are relinked to the new chief. Computational child processes are not terminated, so ongoing computations will not be affected. These processes can appear as "undefined" in the process list, but should vanish when computations are done.
RELINK...	Your browser will ask you for an ip address and a port number. The node is then relinked from its current chief to the new chief given by the host data you have entered. If the node has been chief before the operation it will take all loyals over to the new chief, so this operation can be used to merge two smaller networks into a bigger one.
UNLINK...	The node is unlinked from its current chief and forms a network on its own (a "monadic" network). If the node is chief, the network will not be modified. This operation makes sense if you want to exclude the node from network computations without terminating it.
SCAN NW...	When the node is started in automatic network mode, a subprocess is started, which scans the local area network for other nodes. When this process is over (successfully or not) the scanner process terminates. By this button you can initiate the scanner process manually.
COLLECT...	When the chief is terminated unexpectedly, the network is left in an inconsistent state; the remaining nodes are checked in at a chief node that ceased to exist, yet they are not smart enough to re-organize themselves. Relinking each single node by hand would be a lot of work. In order to speed this up, you go to the properties page of the chief to be designated and press this button. The node will try to collect nodes in the inconsistent network and form a new, well-defined network. So, this is kind of a "repair" button, but it should only be pressed, if there is need to repair something.

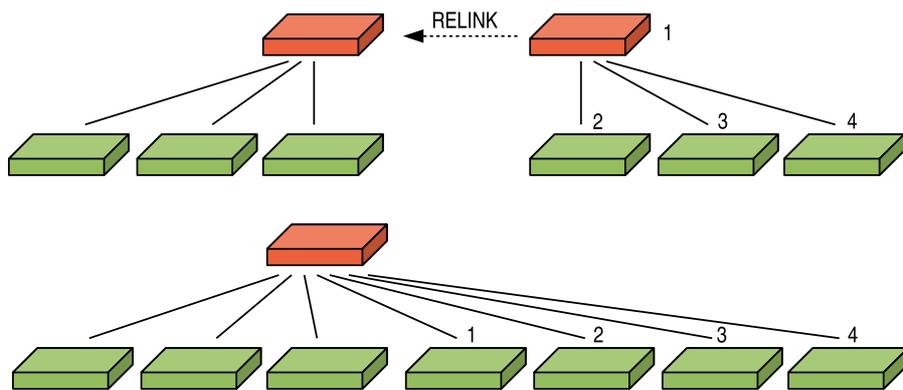


Figure 2: RELINK

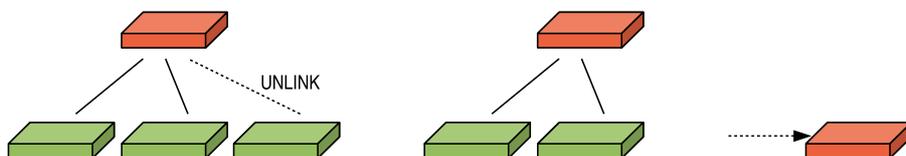


Figure 3: UNLINK





Figure 4: COLLECT

4.4.3 The pages `processes.html` and `processes_network.html`

A node creates other processes for performing various tasks. You can examine these processes in the browser interface, and terminate them in case of trouble. Processes can be examined for the entire network (as in the figure below) or for a single node. In the network view, responder processes are omitted.

host:port	sys/hostname	pid	ppid	category	pcpu	cputime	comm	op
		14497	13102	computational	733	00:00:22	tde4mhp	✖
100.1.1.38:57431	bigmuff	58128	1	parent	0,0	0:00.13	/server/software/osx64/3de/tde4r3_devel //bin/tde4mhp	
		65036	58128	computational	199,1	0:05.93	/server/software/osx64/3de/tde4r3_devel //bin/tde4mhp	✖
100.1.1.39:57431	nebukadnezar	10842	1	parent	0.0	00:00:00	tde4mhp	
		10858	10842	updater_udp	0.0	00:00:00	tde4mhp	✖
		12478	10842	news	0.0	00:00:01	tde4mhp	✖
		12480	10842	news	0.0	00:00:01	tde4mhp	✖
		12482	10842	news	0.0	00:00:01	tde4mhp	✖
		12484	10842	news	0.0	00:00:00	tde4mhp	✖
		16433	10842	news	0.0	00:00:00	tde4mhp	✖
		16472	10842	computational	901	00:00:27	tde4mhp	✖
100.1.1.4:57431	devil	20594	1	parent	0.0	00:00:00	tde4mhp	
		22953	20594	computational	290	00:00:11	tde4mhp	✖
100.1.1.40:57431	smaug	17970	1	parent	0.0	00:00:00	tde4mhp	
		21324	17970	computational	195	00:00:05	tde4mhp	✖

Figure 5: Processes running on hosts in the network

In the following table all process categories are listed.

Process name	Function
parent	The parent process waits for incoming tcp-requests. When a tcp-connection is established, it creates a thread or a process.
responder	A responder process is created when an xml-, html- or any other http-related file is requested. Usually, processes of this type only exist for a less than a second.
computational	Each user doing computations on the node causes the creation of a computational process, i.e. there can be more than one computational at a time. Processes of this type create threads and use a lot of cpu-time. When no calculations are done in the MHP-network, there should not be any computational process. When the parent process is killed, computational processes will keep working, until their jobs are done.
news	The chief host spawns a news process or each instance of 3DE4 connecting to the MHP-network. The news process sends information about the current status of the network to 3DE4, which are then displayed in 3DE4's GUI.
updater_udp	This process is spawned by the chief. It creates an udp-socket and waits for messages from all nodes. The updater_udp process will run as long as the chief node is running (and as long as the node is chief). There can be only one updater_udp process per chief. When (temporarily) there are more processes of this type the additional ones should terminate after a few seconds.

Process name	Function
scanner	In network-mode automatic this process scans the tcp-network for a chief host and checks in at the node running there.
prefork	During startup, i.e. before accepting network connections, the process is called prefork. Processes with this label will generally not appear in process lists, but you may encounter them in the logfiles.
undefined	When a node is restarted, its subprocesses (e.g. computationals) keep on running. These subprocesses are unknown to the new node process and appear as undefined in process lists. They are no problem as long as they disappear after some time, say, when they are no longer needed for computations.

4.4.4 The page load.html

This page lists all nodes participating in the MHP-network. In each row you see the community threads and the private threads of each node. This snapshot was taken, when six users where doing computations in the network. 3DE4/MHP generates a colour for each user in order to visualize how threads are associated to the users (that's why it looks like Teletubbyland meets My Little Pony). You see that community threads are generally divided among some or all users, while the private threads are reserved for one single user working on 3DE4. The page is updated periodically after two seconds. Please note, that your browser will receive live data from the chief of the network. Each browser displaying this page will increase network traffic at the chief node (slightly).

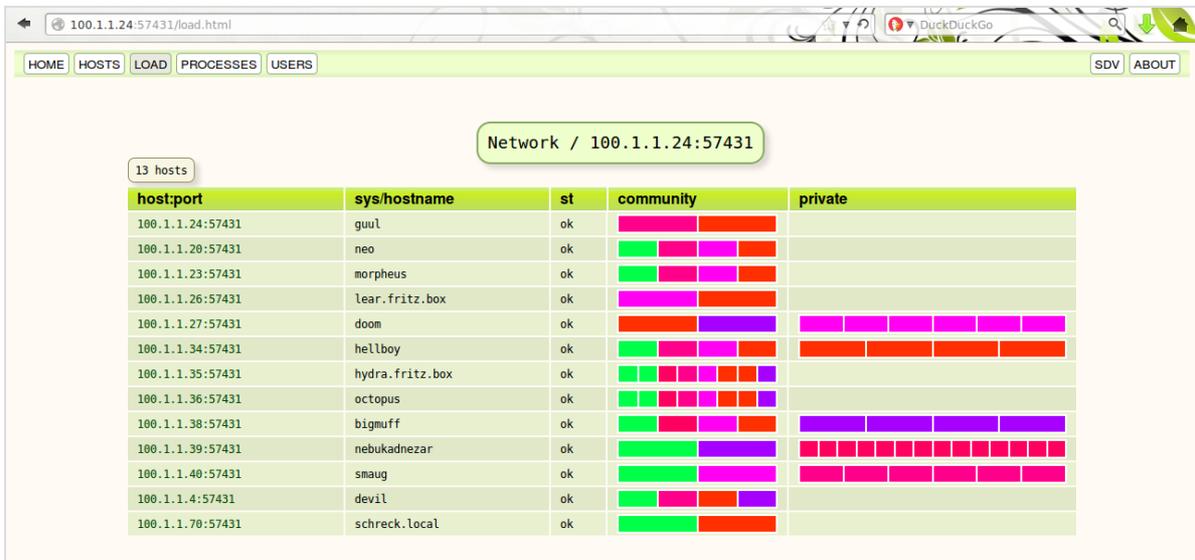


Figure 6: MHP at work

4.5 Lens distortion plugins

If you are using your own lens distortion plugins, tde4mhp needs to know where they are located. You can pass the directory as option `-plugin_dir` or as environment variable `TDE4MHP_PLUGIN_DIR` or as entry in the configuration file. Additionally, for reasons of compatibility to 3DE4 the directories given by the environment variable `LD_PLUGINS_3DE4` are scanned for plugins, where `LD_PLUGINS_3DE4` contains one or more directories, separated by ":". After starting tde4mhp it's a good idea to check in the HTML interface, if all plugins are loaded correctly (http://myhost:myport/ld_models.html). The version number in column "header" refers to the version number in the header `tde4_ld_plugin.h` containing the base class `tde4_ld_plugin`, from which you have derived your plugin class.



HOME | HOSTS | LOAD | PROCESSES | USERS SDV | ABOUT

100.1.1.39:57431

HOST PROPERTIES | LENS MODELS | LOGFILE | PROCESSES

name	linking	path	filename	header
3DE Classic LD Model	static	-	-	1.0.8
3DE4 All Parameter Types [Plugin]	plugin	/server/devel/sdv/tde4/uwe/source/tde4mhp	tde4_ldp_all_par_types.so	1.0.8
3DE4 Anamorphic - Standard, Degree 4	static	-	-	1.0.8
3DE4 Anamorphic, Degree 6	static	-	-	1.0.8
3DE4 Radial - Fisheye, Degree 8	static	-	-	1.0.8
3DE4 Radial - Standard, Degree 4	static	-	-	1.0.8

5 Miscellaneous

5.1 Todo



A command line based administration tool would be helpful.

5.2 An incomplete list of useful shell commands

UNIX-like operation systems provide a set of shell commands which can be helpful in build and maintaining networking applications like 3DE4/MHP. Some of them might be located in `/sbin` on your machine, or require administrator rights. If you are missing your preferred tool here, don't hesitate to write us, we will add it to the list.

Name	Purpose
curl	often available on OSX. non-interactive download of files (http,https,ftp).
ifconfig	does a lot, useful for examining network interfaces
ipcs	report interprocess communication facilities status. The shared memory acquired by tde4mhp will appear here.
kill	send a signal to a process. If you need to kill a tde4mhp process, please keep in mind that it's better to try -TERM (the default) instead of -KILL (-9). Only use -KILL if necessary, because tde4mhp won't be able to clean up.
killall	kill (more precise: send signal to) processes by name
lscpu	list cpu information
lsof	list open files. This one produces a lot of output. Some options are particularly interesting, like e.g. -i which lists processes with open sockets. Or you may try lsof grep tde4mhp , in order to see what tde4mhp is doing on your machine. Lens distortion plugins should appear in the output.
netstat	the Swiss knife in networking
nmap	(obsolete) a port scanner. Should be available as a package in most Linux distributions.
ping	check if host is up and reachable
ps	report process status
top	display processes
wget	often available on Linux. non-interactive download of files (http,https,ftp).

5.3 Multi-host administration

There is a number of administration tools for multiple hosts available. We are currently using `cluster-ssh` which comes in quite handy.



Document created with CDML from /server/dev/sdv/privat/uwe/source/tde4_doc/tde4_mhp/cdml/tde4_mhp.xml



© 2014 by Science-D-Visions.