# MDBenchmark

A tedious task — made straightforward

Michael Gecht, Marc Siggel, Max Linke

MAX-PLANCK-GESELLSCHAFT

mpibp
max-planck-institut
für biophysik

## Computing resources are a limiting factor

- Computational power is not abundant and expensive, …
- … but many interesting questions require lots of computational resources

Optimal usage of computational resources is mandatory

# More computers ≠ more performance

- Performance of simulations does not increase linearly with computational power
- Many factors to optimise:
  - Number of nodes
  - CPUs or GPUs
  - Cluster specifics (MPI, OpenMP)
  - Software specific (PME, cut-off, …)

## How does MDBenchmark help?

- Number of nodes
- CPUs or GPUs
- Cluster specifics (MPI, OpenMP)
- Software specific (PME, cut-off, …)

# How does MDBenchmark help?

- Number of nodes ✓
- CPUs or GPUs ✓
- *Cluster specifics (MPI, OpenMP)* 🔍
- *Software specific (PME, cut-off, …)* 🔍

Handles scaling studies of MD simulations

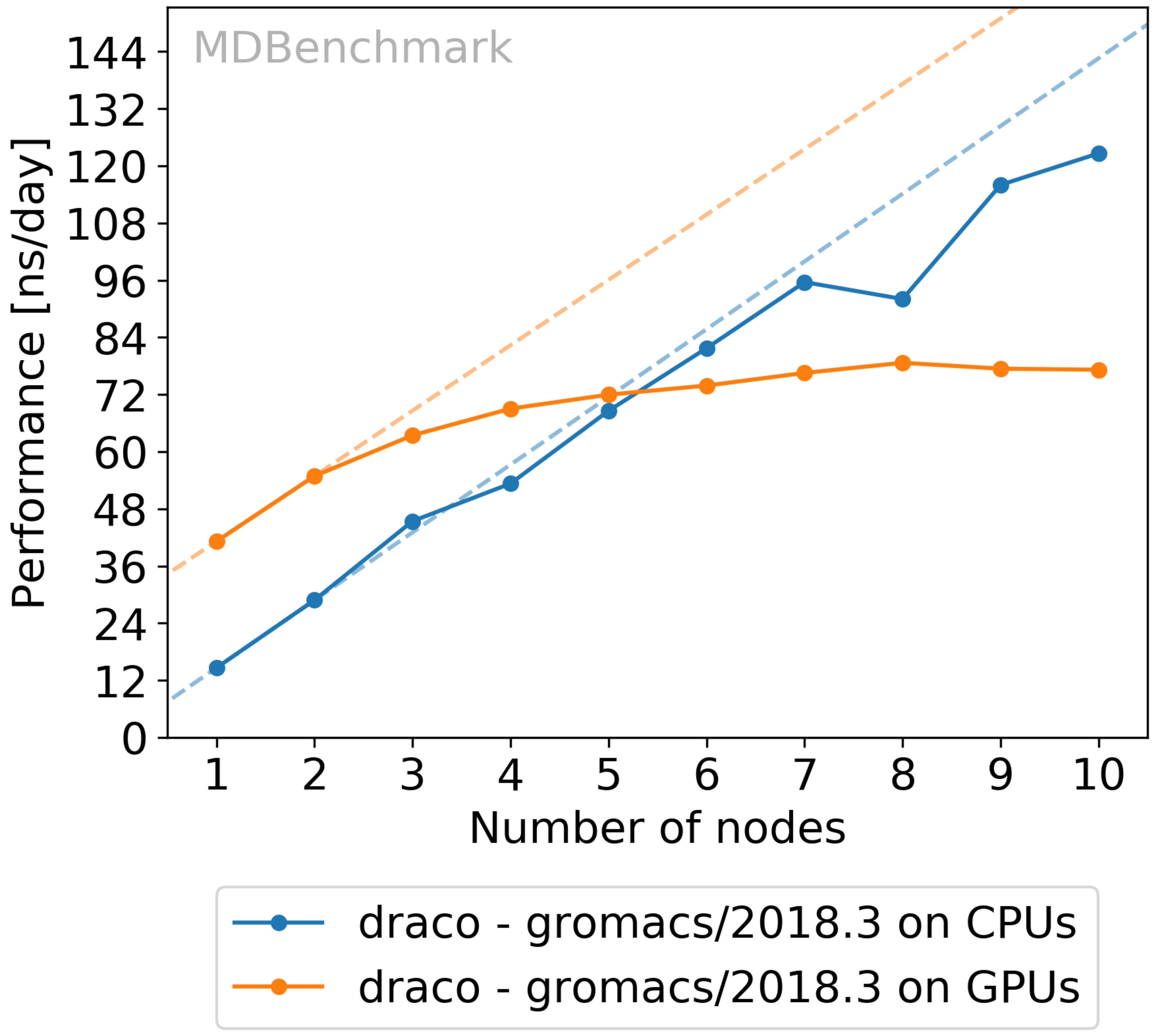
## What is MDBenchmark?

- Command-line interface (CLI) written in Python
- Open source and free (GPLv3), hosted on GitHub
- Documentation covers all usage patterns
- Installable on any computer running Python

# What does it do for you?

1. Create benchmarks with different parameters
2. Submit benchmarks to queuing system
3. Analyze running/finished benchmarks
4. Plot different benchmarks for comparison

# What does it do for you?



# Installation with your favourite Python package manager

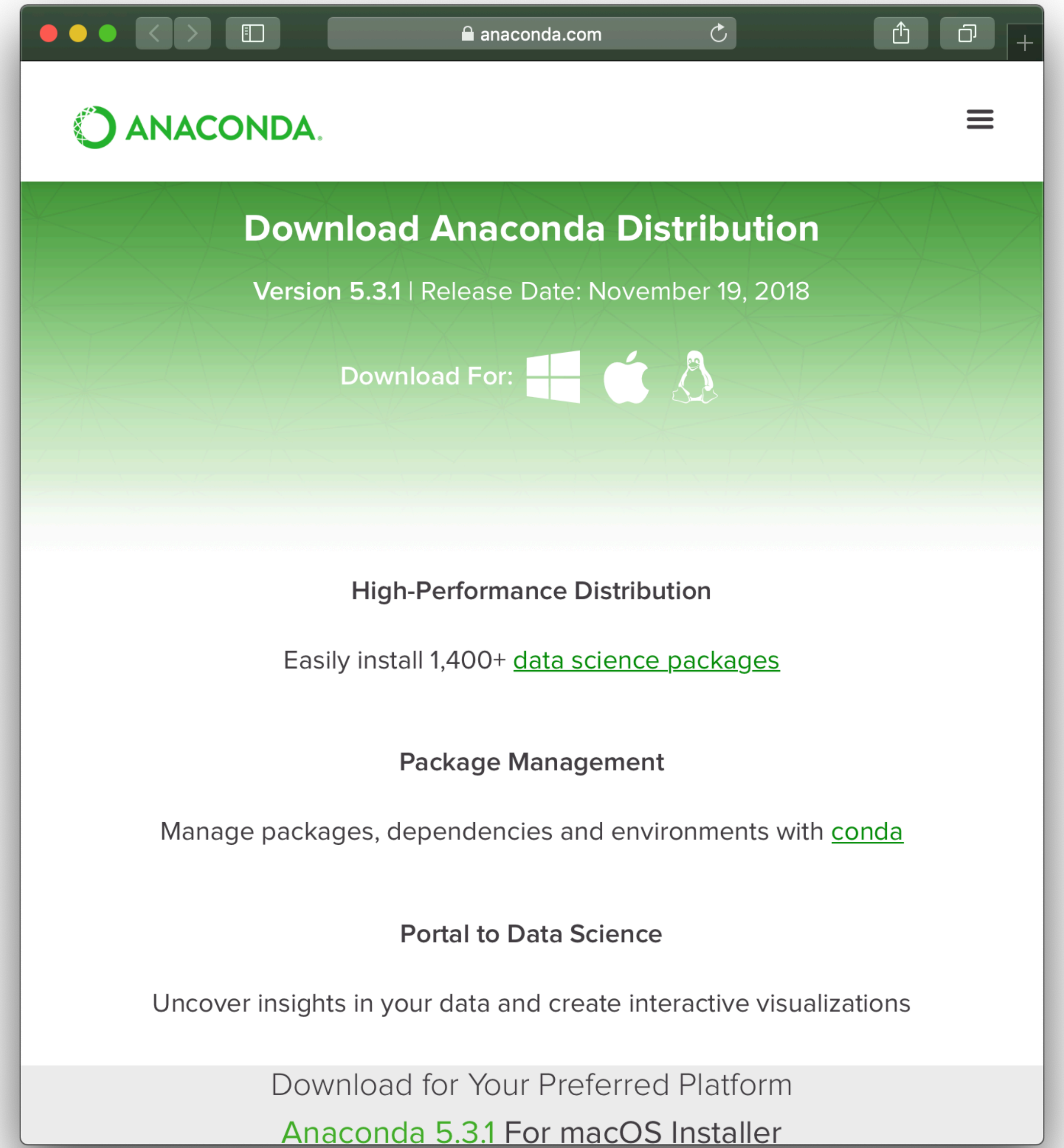
Find, install and publish Python packages with the Python Package Index

159,291 projects 1,137,286 releases 1,578,800 files 279,193 users

The Python Package Index (PyPI) is a repository of software for the Python programming language.

PyPI helps you find and install software developed and shared by the Python community. Learn about installing packages.

Package authors use PyPI to distribute their software. Learn how to package your Python code for PyPI.



Download Anaconda Distribution

Version 5.3.1 | Release Date: November 19, 2018

High-Performance Distribution

Easily install 1,400+ data science packages

Package Management

Manage packages, dependencies and environments with conda

Portal to Data Science

Uncover insights in your data and create interactive visualizations

Download for Your Preferred Platform

Anaconda 5.3.1 For macOS Installer

```
$ pip install mdbenchmark
```

```
$ conda install -c conda-forge \
  mdbenchmark
```

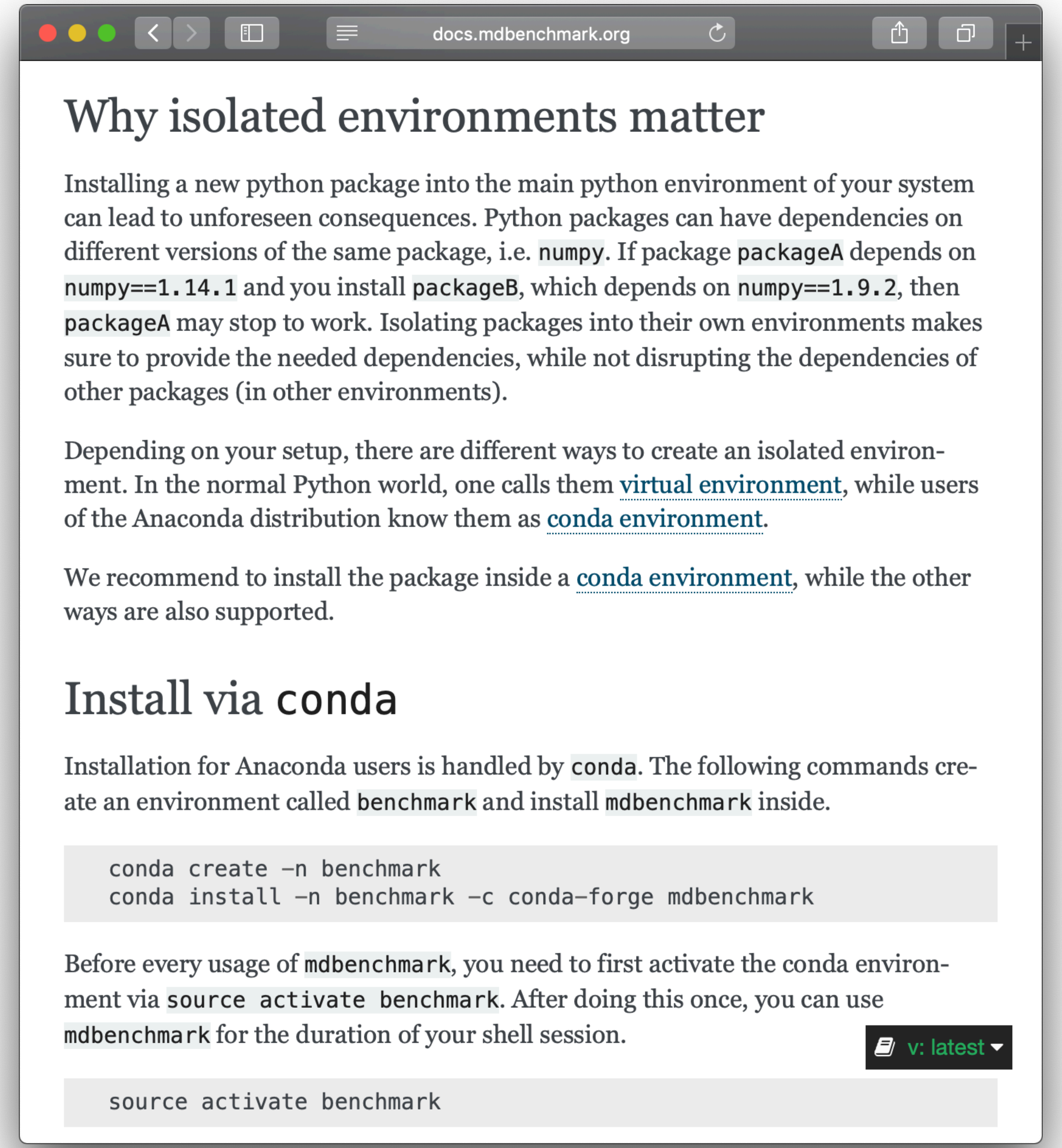
# Setting up MDBenchmark on your machine

1. Create isolated environment

2. Install MDBenchmark in an isolated Python environment

3. Activate the environment

4. Use `mdbenchmark` from your terminal



## Why isolated environments matter

Installing a new python package into the main python environment of your system can lead to unforeseen consequences. Python packages can have dependencies on different versions of the same package, i.e. `numpy`. If package `packageA` depends on `numpy==1.14.1` and you install `packageB`, which depends on `numpy==1.9.2`, then `packageA` may stop to work. Isolating packages into their own environments makes sure to provide the needed dependencies, while not disrupting the dependencies of other packages (in other environments).

Depending on your setup, there are different ways to create an isolated environment. In the normal Python world, one calls them virtual environment, while users of the Anaconda distribution know them as conda environment.

We recommend to install the package inside a conda environment, while the other ways are also supported.

## Install via `conda`

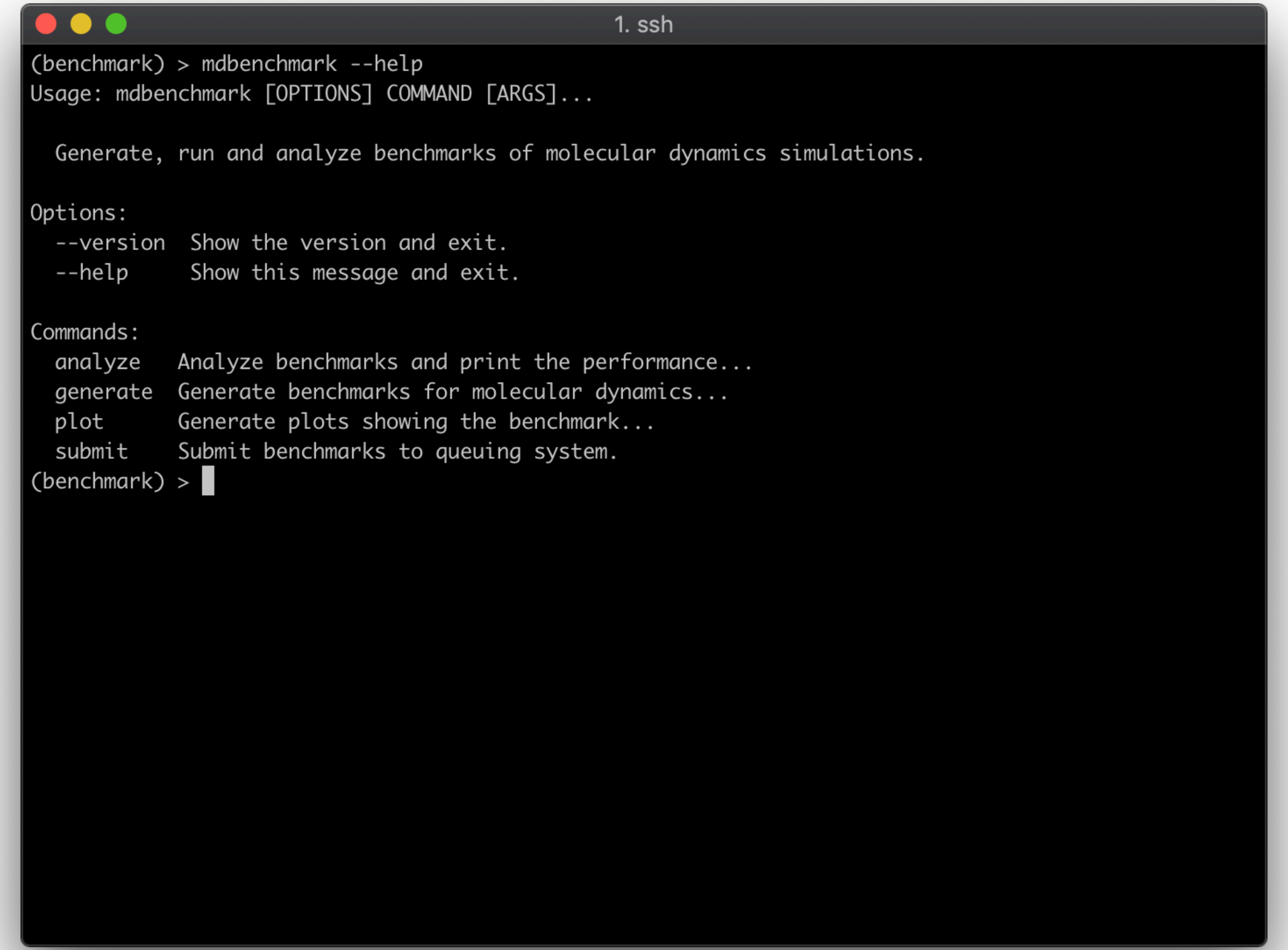
Installation for Anaconda users is handled by `conda`. The following commands create an environment called `benchmark` and install `mdbenchmark` inside.

```
conda create -n benchmark
conda install -n benchmark -c conda-forge mdbenchmark
```

Before every usage of `mdbenchmark`, you need to first activate the conda environment via `source activate benchmark`. After doing this once, you can use `mdbenchmark` for the duration of your shell session.

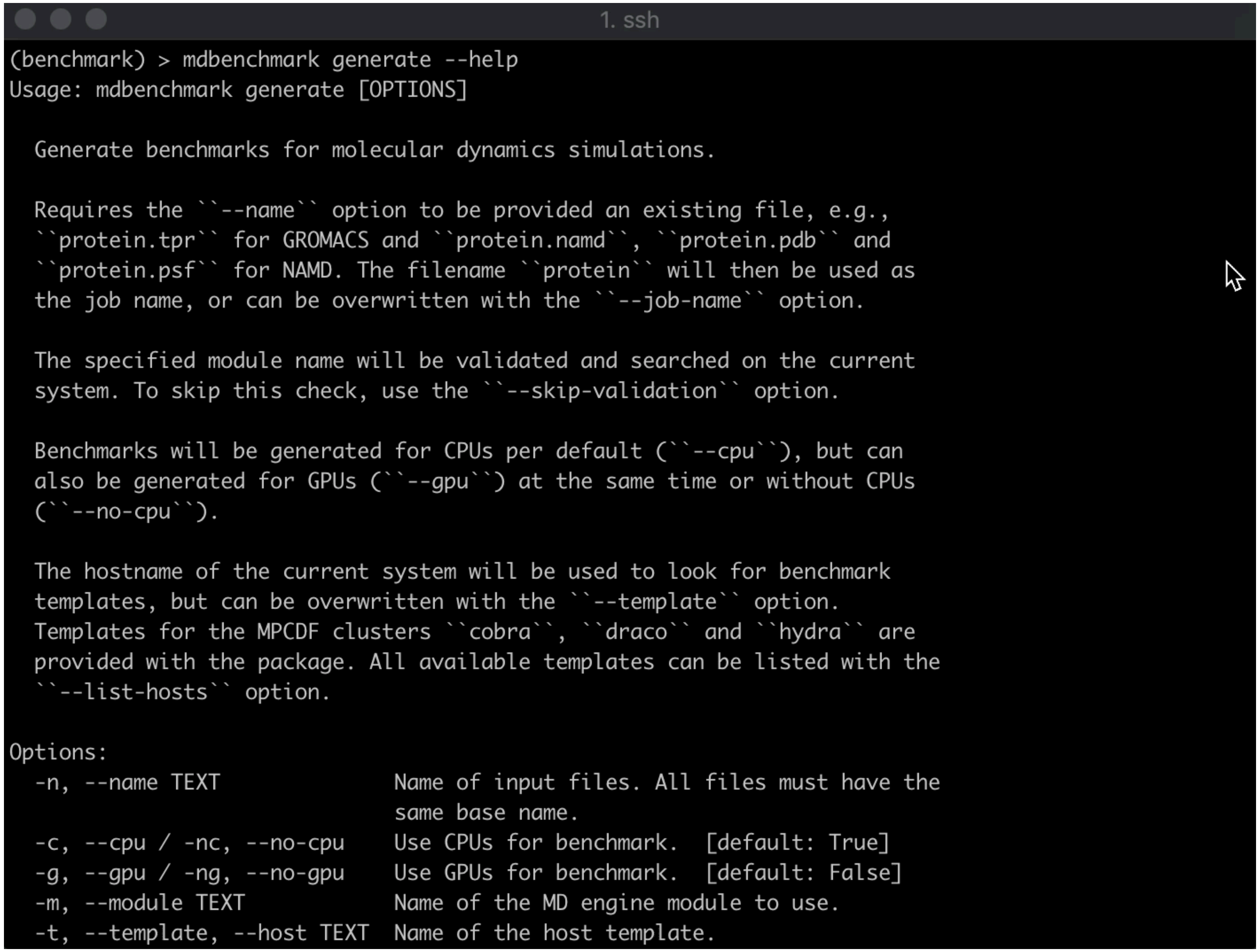
```
source activate benchmark
```

# MDBenchmark: CLI



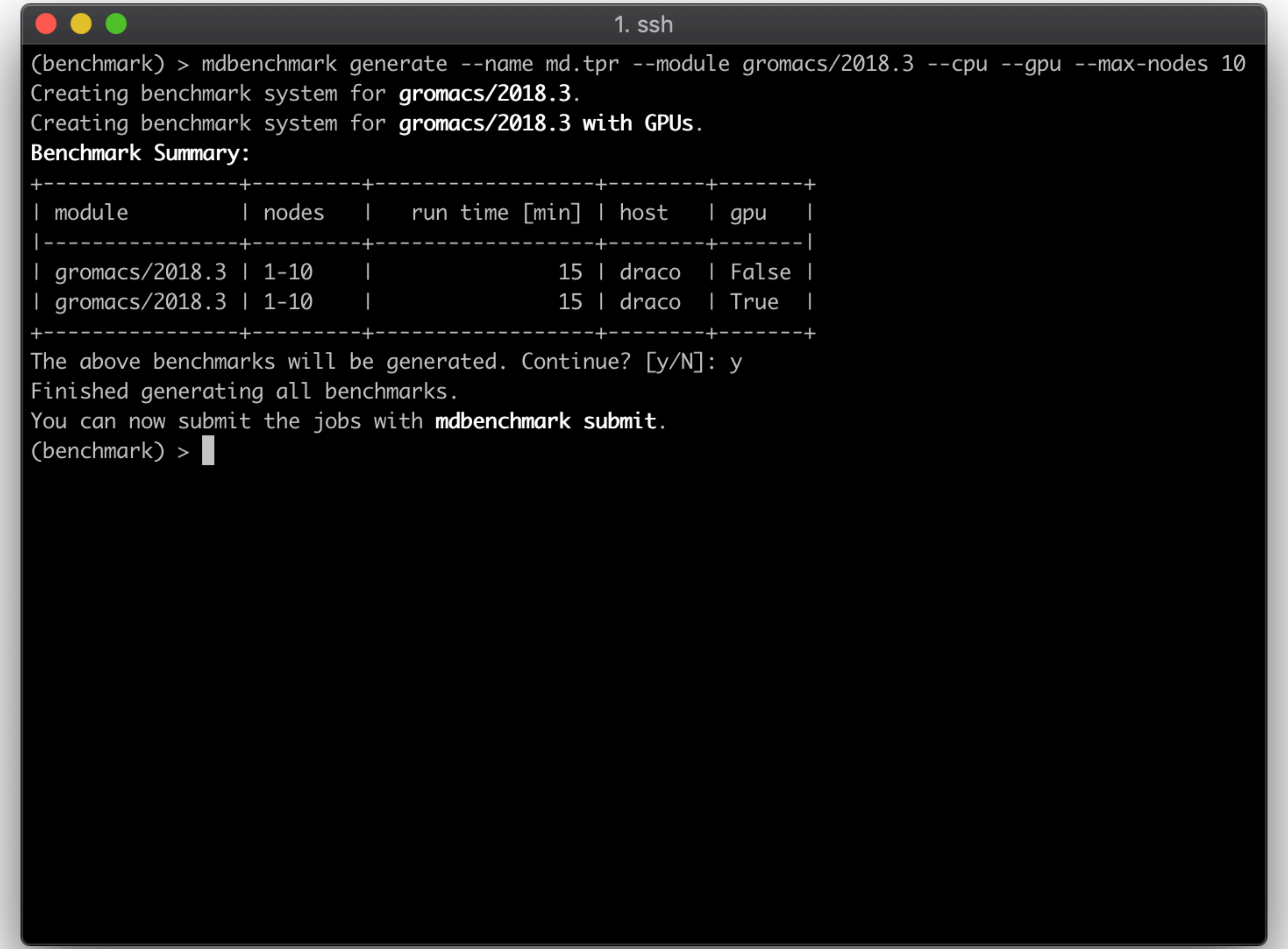
```
(benchmark) > mdbenchmark --help
Usage: mdbenchmark [OPTIONS] COMMAND [ARGS]...

  Generate, run and analyze benchmarks of molecular dynamics simulations.

Options:
  --version  Show the version and exit.
  --help     Show this message and exit.

Commands:
  analyze   Analyze benchmarks and print the performance...
  generate  Generate benchmarks for molecular dynamics...
  plot      Generate plots showing the benchmark...
  submit    Submit benchmarks to queuing system.
(benchmark) > 
```

# Help is available for every command



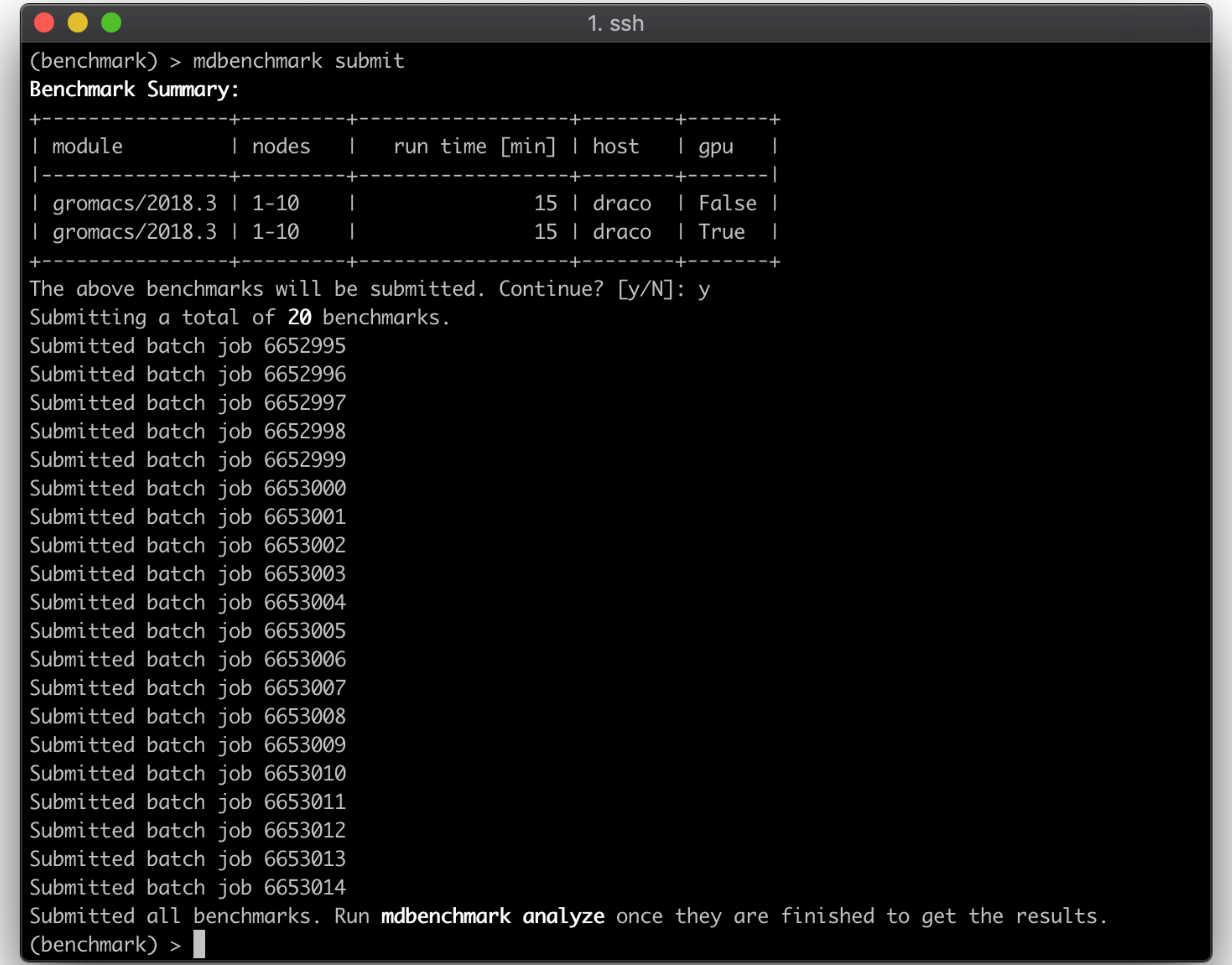
```
(benchmark) > mdbenchmark generate --help
Usage: mdbenchmark generate [OPTIONS]

  Generate benchmarks for molecular dynamics simulations.

  Requires the ``--name`` option to be provided an existing file, e.g.,
  ``protein.tpr`` for GROMACS and ``protein.namd``, ``protein.pdb`` and
  ``protein.psf`` for NAMD. The filename ``protein`` will then be used as
  the job name, or can be overwritten with the ``--job-name`` option.

  The specified module name will be validated and searched on the current
  system. To skip this check, use the ``--skip-validation`` option.

  Benchmarks will be generated for CPUs per default (``--cpu``), but can
  also be generated for GPUs (``--gpu``) at the same time or without CPUs
  (``--no-cpu``).

  The hostname of the current system will be used to look for benchmark
  templates, but can be overwritten with the ``--template`` option.
  Templates for the MPCDF clusters ``cobra``, ``draco`` and ``hydra`` are
  provided with the package. All available templates can be listed with the
  ``--list-hosts`` option.

Options:
  -n, --name TEXT              Name of input files. All files must have the
                               same base name.
  -c, --cpu / -nc, --no-cpu    Use CPUs for benchmark.  [default: True]
  -g, --gpu / -ng, --no-gpu    Use GPUs for benchmark.  [default: False]
  -m, --module TEXT            Name of the MD engine module to use.
  -t, --template, --host TEXT  Name of the host template.
```

# Generating benchmarks



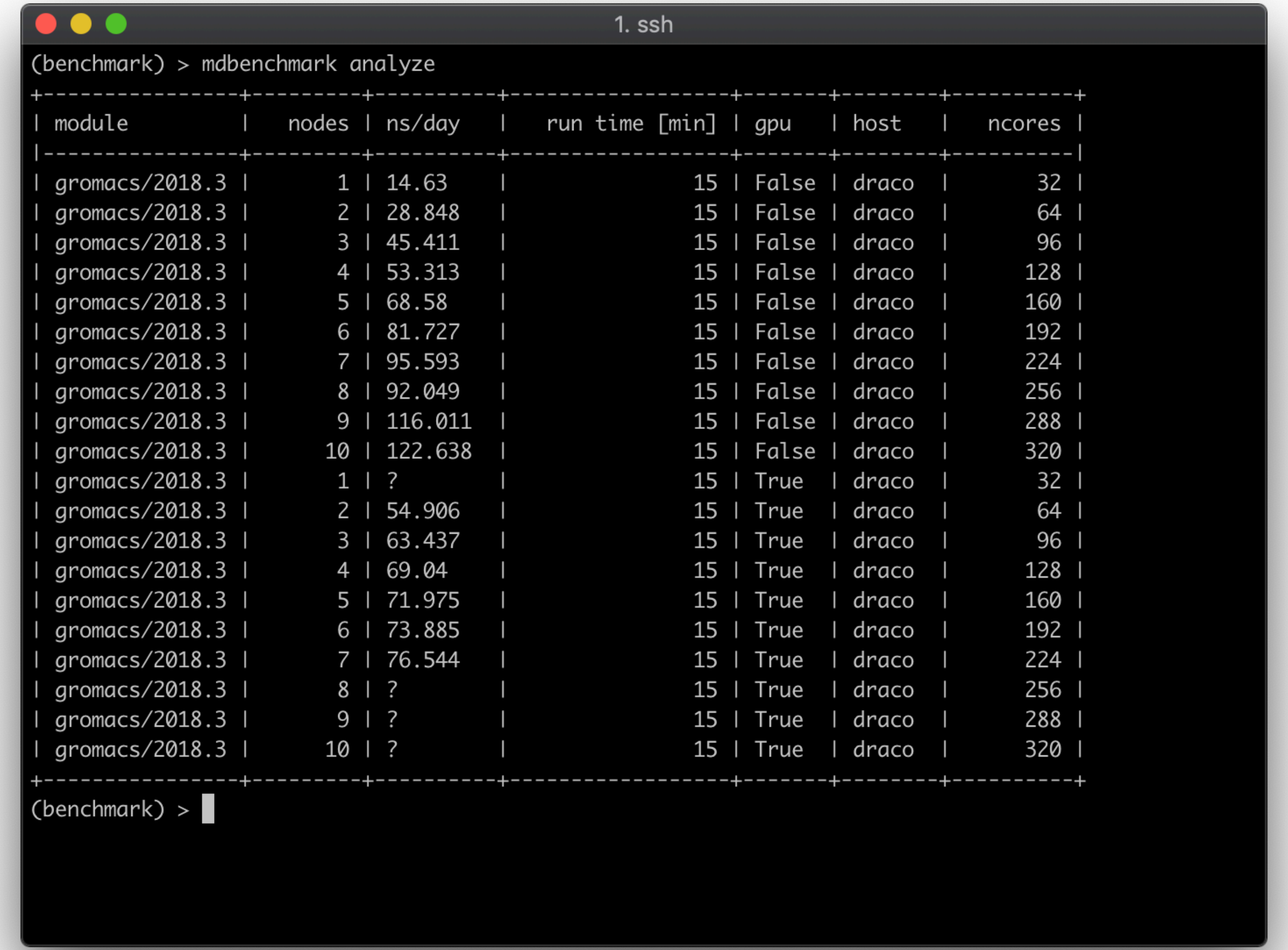
```
(benchmark) > mdbenchmark generate --name md.tpr --module gromacs/2018.3 --cpu --gpu --max-nodes 10
Creating benchmark system for gromacs/2018.3.
Creating benchmark system for gromacs/2018.3 with GPUs.
Benchmark Summary:
+----------------+---------+------------------+--------+-------+
| module         | nodes   |   run time [min] | host   | gpu   |
|----------------+---------+------------------+--------+-------|
| gromacs/2018.3 | 1-10    |               15 | draco  | False |
| gromacs/2018.3 | 1-10    |               15 | draco  | True  |
+----------------+---------+------------------+--------+-------+
The above benchmarks will be generated. Continue? [y/N]: y
Finished generating all benchmarks.
You can now submit the jobs with mdbenchmark submit.
(benchmark) >
```

# Submit to any queuing systems



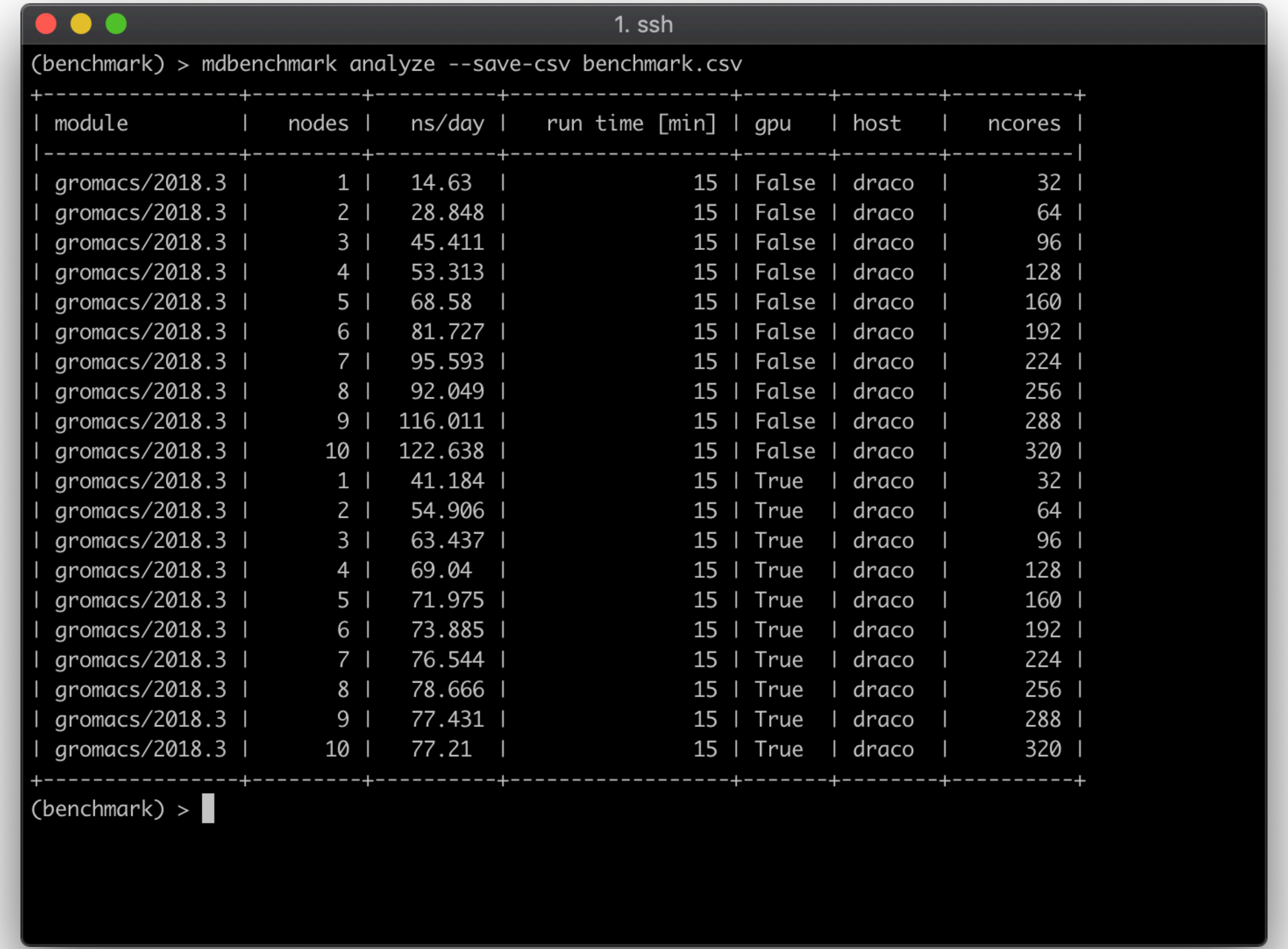
```
(benchmark) > mdbenchmark submit
Benchmark Summary:
+----------------+---------+-----------------+--------+-------+
| module         | nodes   |   run time [min] | host   | gpu   |
|----------------+---------+-----------------+--------+-------|
| gromacs/2018.3 | 1-10    |              15 | draco  | False |
| gromacs/2018.3 | 1-10    |              15 | draco  | True  |
+----------------+---------+-----------------+--------+-------+
The above benchmarks will be submitted. Continue? [y/N]: y
Submitting a total of 20 benchmarks.
Submitted batch job 6652995
Submitted batch job 6652996
Submitted batch job 6652997
Submitted batch job 6652998
Submitted batch job 6652999
Submitted batch job 6653000
Submitted batch job 6653001
Submitted batch job 6653002
Submitted batch job 6653003
Submitted batch job 6653004
Submitted batch job 6653005
Submitted batch job 6653006
Submitted batch job 6653007
Submitted batch job 6653008
Submitted batch job 6653009
Submitted batch job 6653010
Submitted batch job 6653011
Submitted batch job 6653012
Submitted batch job 6653013
Submitted batch job 6653014
Submitted all benchmarks. Run mdbenchmark analyze once they are finished to get the results.
(benchmark) >
```

# Grab performance from log files



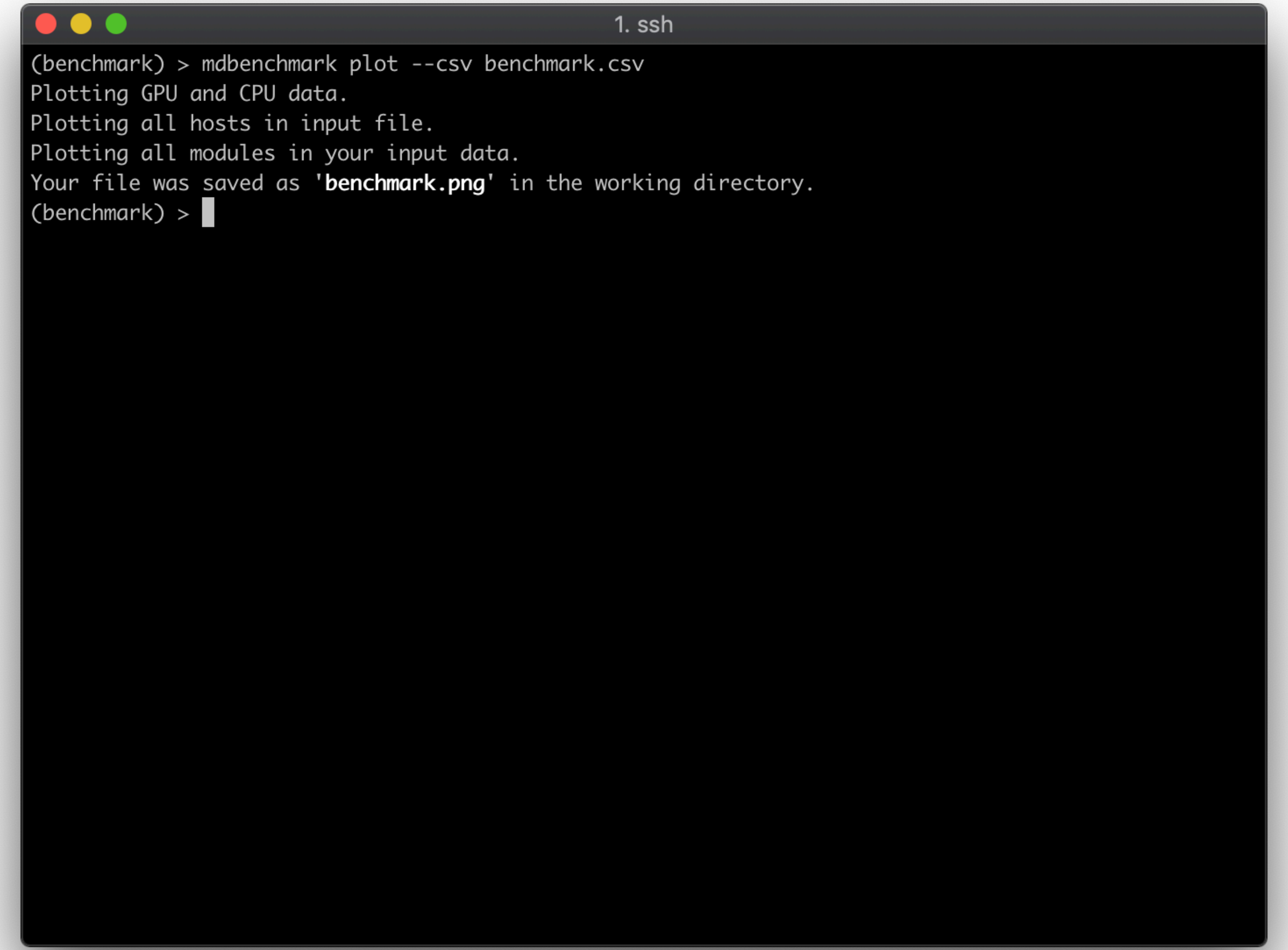
```
(benchmark) > mdbenchmark analyze
+----------------+---------+----------+------------------+-------+--------+----------+
| module         |   nodes | ns/day   |   run time [min] | gpu   | host   |   ncores |
|----------------+---------+----------+------------------+-------+--------+----------|
| gromacs/2018.3 |       1 | 14.63    |               15 | False | draco  |       32 |
| gromacs/2018.3 |       2 | 28.848   |               15 | False | draco  |       64 |
| gromacs/2018.3 |       3 | 45.411   |               15 | False | draco  |       96 |
| gromacs/2018.3 |       4 | 53.313   |               15 | False | draco  |      128 |
| gromacs/2018.3 |       5 | 68.58    |               15 | False | draco  |      160 |
| gromacs/2018.3 |       6 | 81.727   |               15 | False | draco  |      192 |
| gromacs/2018.3 |       7 | 95.593   |               15 | False | draco  |      224 |
| gromacs/2018.3 |       8 | 92.049   |               15 | False | draco  |      256 |
| gromacs/2018.3 |       9 | 116.011  |               15 | False | draco  |      288 |
| gromacs/2018.3 |      10 | 122.638  |               15 | False | draco  |      320 |
| gromacs/2018.3 |       1 | ?        |               15 | True  | draco  |       32 |
| gromacs/2018.3 |       2 | 54.906   |               15 | True  | draco  |       64 |
| gromacs/2018.3 |       3 | 63.437   |               15 | True  | draco  |       96 |
| gromacs/2018.3 |       4 | 69.04    |               15 | True  | draco  |      128 |
| gromacs/2018.3 |       5 | 71.975   |               15 | True  | draco  |      160 |
| gromacs/2018.3 |       6 | 73.885   |               15 | True  | draco  |      192 |
| gromacs/2018.3 |       7 | 76.544   |               15 | True  | draco  |      224 |
| gromacs/2018.3 |       8 | ?        |               15 | True  | draco  |      256 |
| gromacs/2018.3 |       9 | ?        |               15 | True  | draco  |      288 |
| gromacs/2018.3 |      10 | ?        |               15 | True  | draco  |      320 |
+----------------+---------+----------+------------------+-------+--------+----------+
(benchmark) >
```

# Save results to a CSV



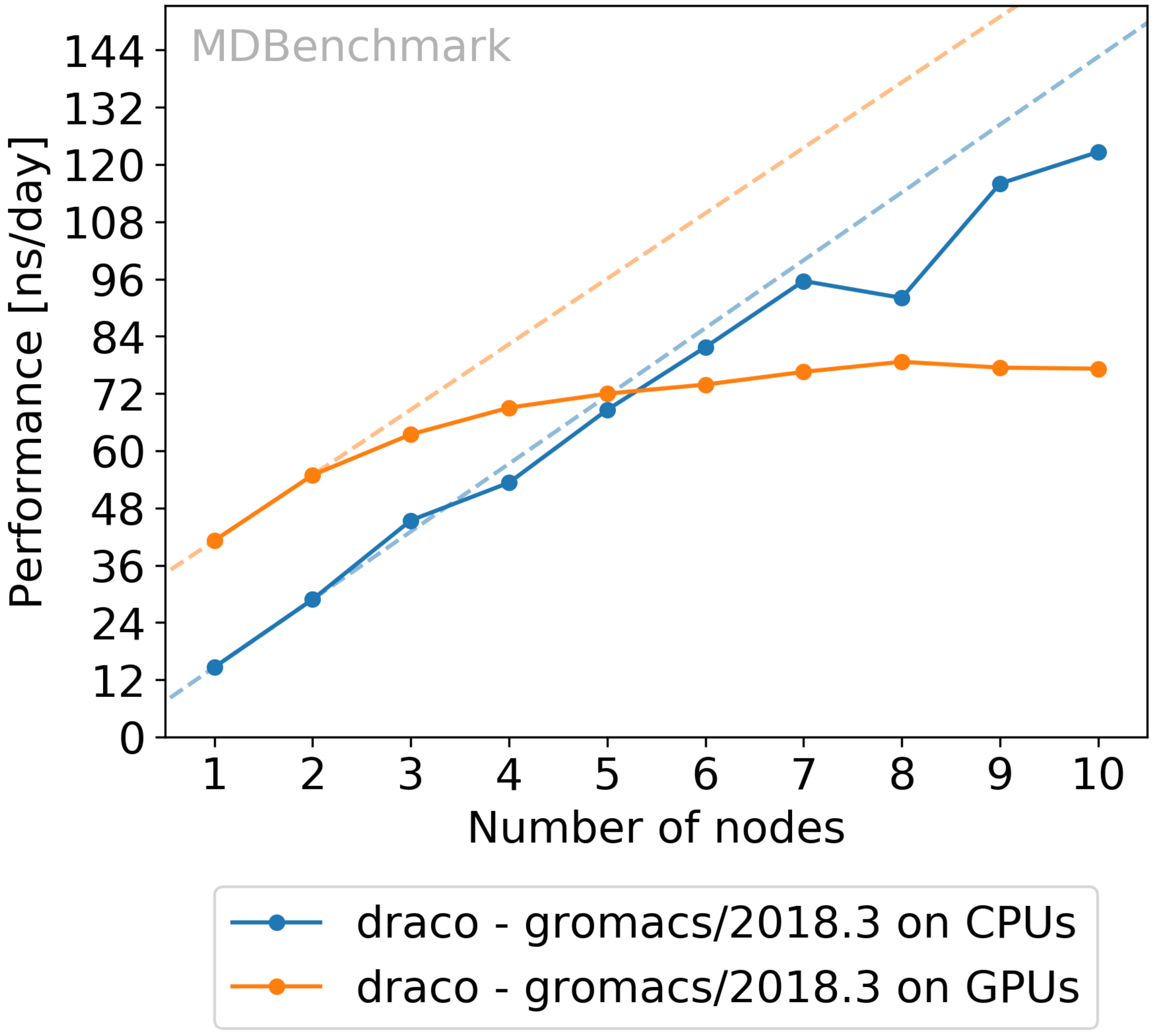
```
(benchmark) > mdbenchmark analyze --save-csv benchmark.csv
+---------------+--------+---------+----------------+-------+-------+----------+
| module        |  nodes |  ns/day |  run time [min] | gpu   | host  |   ncores |
|---------------+--------+---------+----------------+-------+-------+----------|
| gromacs/2018.3 |      1 |  14.63  |             15 | False | draco |       32 |
| gromacs/2018.3 |      2 |  28.848 |             15 | False | draco |       64 |
| gromacs/2018.3 |      3 |  45.411 |             15 | False | draco |       96 |
| gromacs/2018.3 |      4 |  53.313 |             15 | False | draco |      128 |
| gromacs/2018.3 |      5 |  68.58  |             15 | False | draco |      160 |
| gromacs/2018.3 |      6 |  81.727 |             15 | False | draco |      192 |
| gromacs/2018.3 |      7 |  95.593 |             15 | False | draco |      224 |
| gromacs/2018.3 |      8 |  92.049 |             15 | False | draco |      256 |
| gromacs/2018.3 |      9 | 116.011 |             15 | False | draco |      288 |
| gromacs/2018.3 |     10 | 122.638 |             15 | False | draco |      320 |
| gromacs/2018.3 |      1 |  41.184 |             15 | True  | draco |       32 |
| gromacs/2018.3 |      2 |  54.906 |             15 | True  | draco |       64 |
| gromacs/2018.3 |      3 |  63.437 |             15 | True  | draco |       96 |
| gromacs/2018.3 |      4 |  69.04  |             15 | True  | draco |      128 |
| gromacs/2018.3 |      5 |  71.975 |             15 | True  | draco |      160 |
| gromacs/2018.3 |      6 |  73.885 |             15 | True  | draco |      192 |
| gromacs/2018.3 |      7 |  76.544 |             15 | True  | draco |      224 |
| gromacs/2018.3 |      8 |  78.666 |             15 | True  | draco |      256 |
| gromacs/2018.3 |      9 |  77.431 |             15 | True  | draco |      288 |
| gromacs/2018.3 |     10 |  77.21  |             15 | True  | draco |      320 |
+---------------+--------+---------+----------------+-------+-------+----------+
(benchmark) >
```

# Create a performance plot

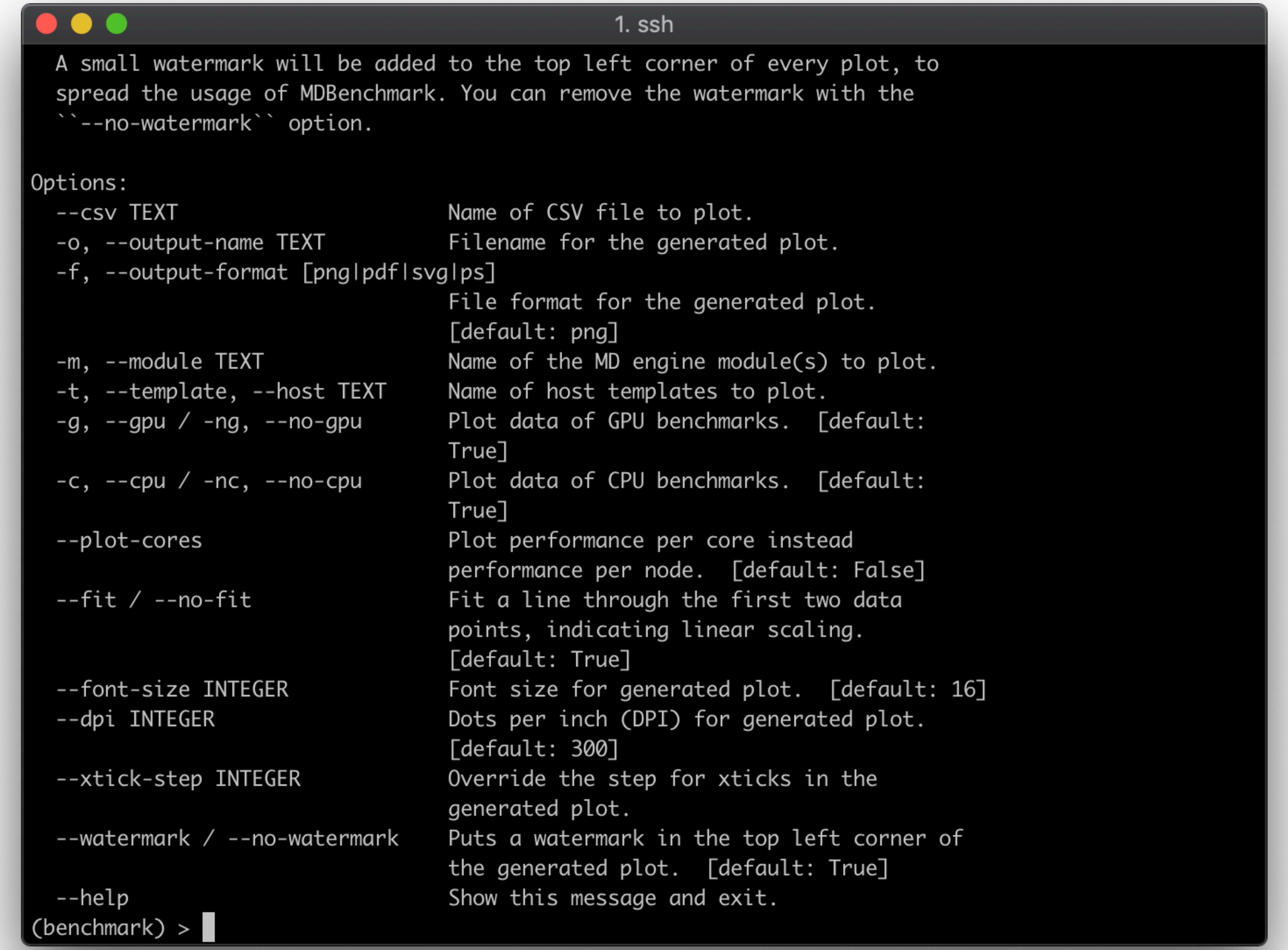


```
(benchmark) > mdbenchmark plot --csv benchmark.csv
Plotting GPU and CPU data.
Plotting all hosts in input file.
Plotting all modules in your input data.
Your file was saved as 'benchmark.png' in the working directory.
(benchmark) >
```

# Performance plot



# Plots can be customised



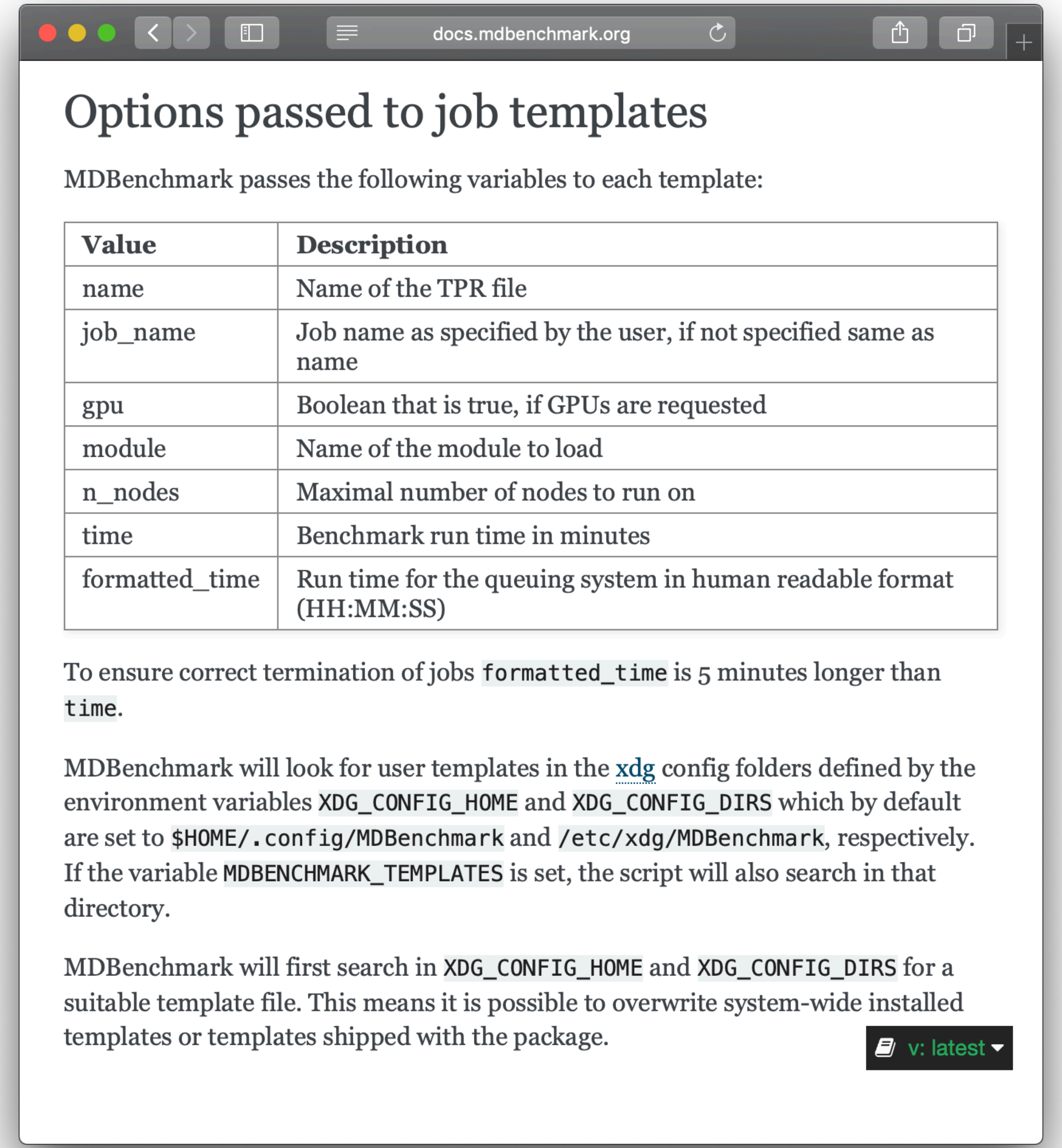
```
  A small watermark will be added to the top left corner of every plot, to
  spread the usage of MDBenchmark. You can remove the watermark with the
  ``--no-watermark`` option.

Options:
  --csv TEXT                      Name of CSV file to plot.
  -o, --output-name TEXT          Filename for the generated plot.
  -f, --output-format [png|pdf|svg|ps]
                                  File format for the generated plot.
                                  [default: png]
  -m, --module TEXT               Name of the MD engine module(s) to plot.
  -t, --template, --host TEXT     Name of host templates to plot.
  -g, --gpu / -ng, --no-gpu       Plot data of GPU benchmarks.  [default:
                                  True]
  -c, --cpu / -nc, --no-cpu       Plot data of CPU benchmarks.  [default:
                                  True]
  --plot-cores                    Plot performance per core instead
                                  performance per node.  [default: False]
  --fit / --no-fit                Fit a line through the first two data
                                  points, indicating linear scaling.
                                  [default: True]
  --font-size INTEGER             Font size for generated plot.  [default: 16]
  --dpi INTEGER                   Dots per inch (DPI) for generated plot.
                                  [default: 300]
  --xtick-step INTEGER            Override the step for xticks in the
                                  generated plot.
  --watermark / --no-watermark    Puts a watermark in the top left corner of
                                  the generated plot.  [default: True]
  --help                          Show this message and exit.
(benchmark) > 
```

# Writing your own templates

- Customisation for own clusters is done with host templates
- MDBenchmark parametrises host templates
- Host templates can be provided system-wide or on a per-user basis:

  ```
  /etc/xdg/MDBenchmark/my_hpc

  /home/user/.config/MDBenchmark/my_hpc
  ```

# Job templates are parametrised



## Options passed to job templates

MDBenchmark passes the following variables to each template:

| Value | Description |
|---|---|
| name | Name of the TPR file |
| job_name | Job name as specified by the user, if not specified same as name |
| gpu | Boolean that is true, if GPUs are requested |
| module | Name of the module to load |
| n_nodes | Maximal number of nodes to run on |
| time | Benchmark run time in minutes |
| formatted_time | Run time for the queuing system in human readable format (HH:MM:SS) |

To ensure correct termination of jobs `formatted_time` is 5 minutes longer than `time`.

MDBenchmark will look for user templates in the xdg config folders defined by the environment variables `XDG_CONFIG_HOME` and `XDG_CONFIG_DIRS` which by default are set to `$HOME/.config/MDBenchmark` and `/etc/xdg/MDBenchmark`, respectively. If the variable `MDBENCHMARK_TEMPLATES` is set, the script will also search in that directory.

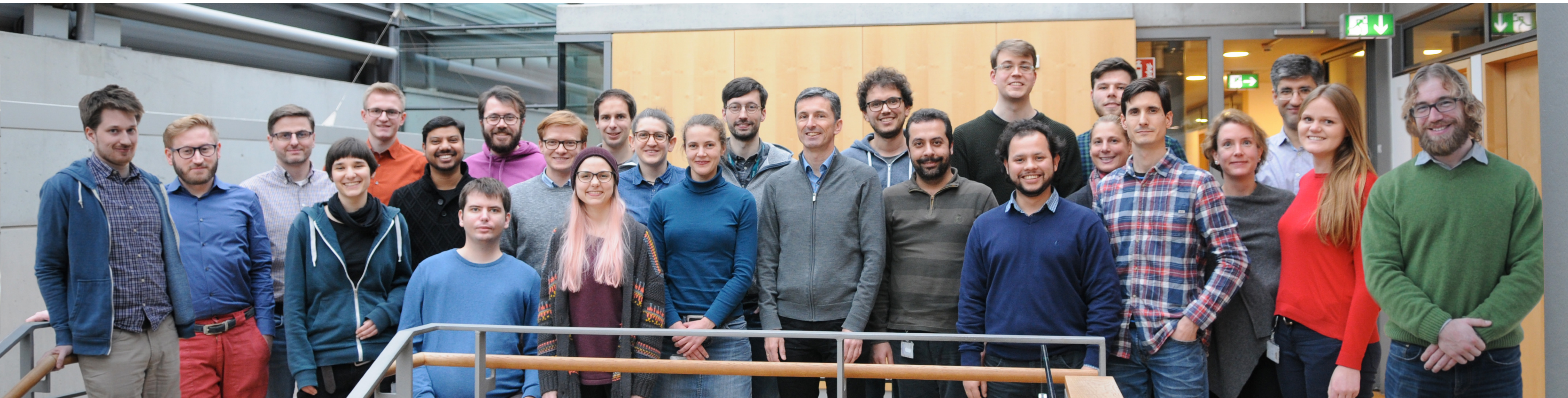
MDBenchmark will first search in `XDG_CONFIG_HOME` and `XDG_CONFIG_DIRS` for a suitable template file. This means it is possible to overwrite system-wide installed templates or templates shipped with the package.

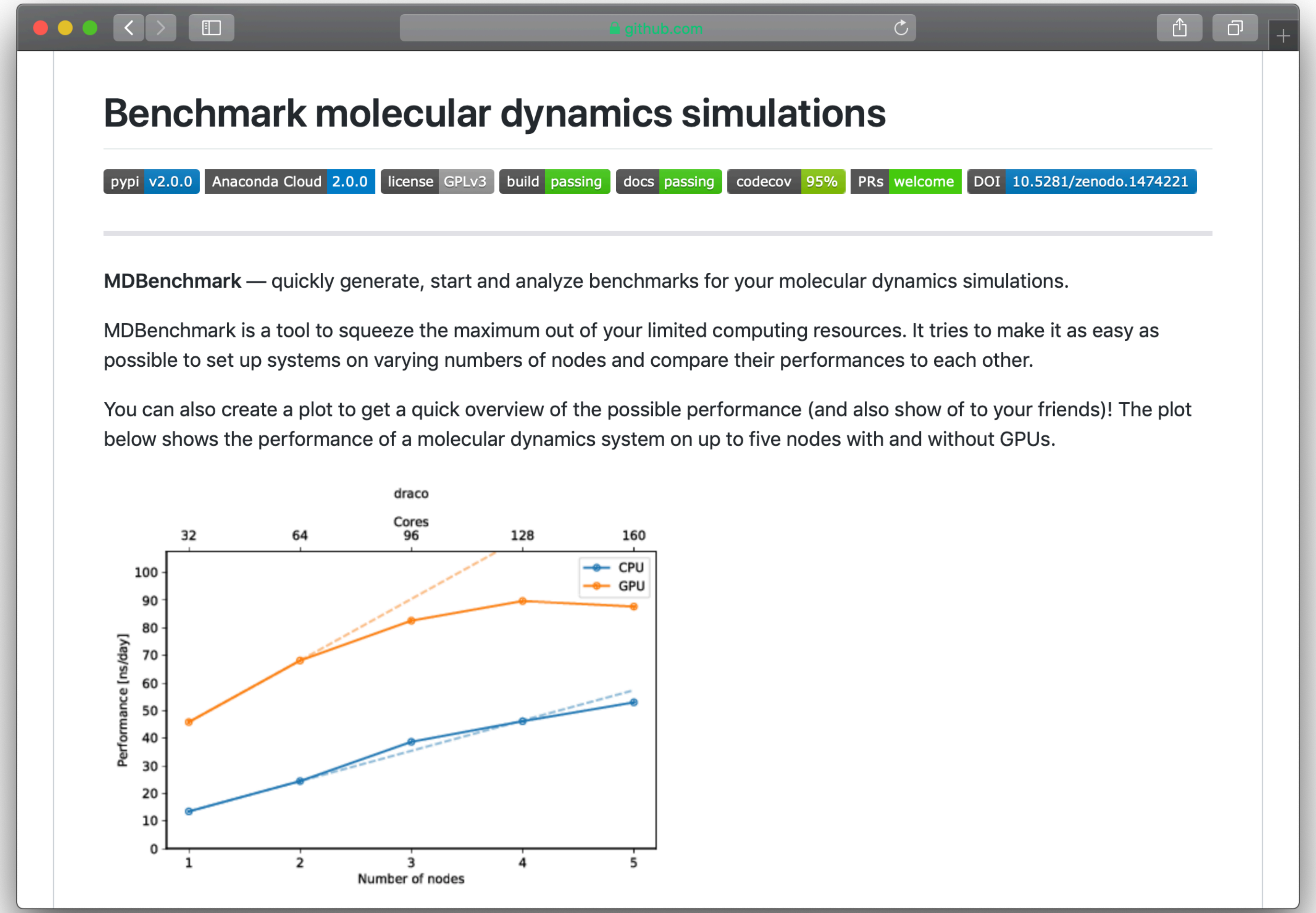
# Where do we go from here?

- Explore parameter space:
  - cluster specific options (MPI/OpenMP, hyper-threading, …)
  - Software specifics (PME, cut-offs, …)
- Provide API for programmatic usage
- Add missing MD engines (AMBER, LAMMPS)

# Acknowledgments

- Prof. Dr. Gerhard Hummer
- Dr. Jürgen Köfinger
- Max Linke
- Marc Siggel





# Benchmark molecular dynamics simulations

pypi v2.0.0 | Anaconda Cloud 2.0.0 | license GPLv3 | build passing | docs passing | codecov 95% | PRs welcome | DOI 10.5281/zenodo.1474221

**MDBenchmark** — quickly generate, start and analyze benchmarks for your molecular dynamics simulations.

MDBenchmark is a tool to squeeze the maximum out of your limited computing resources. It tries to make it as easy as possible to set up systems on varying numbers of nodes and compare their performances to each other.

You can also create a plot to get a quick overview of the possible performance (and also show of to your friends)! The plot below shows the performance of a molecular dynamics system on up to five nodes with and without GPUs.

## Code

https://github.com/bio-phys/mdbenchmark

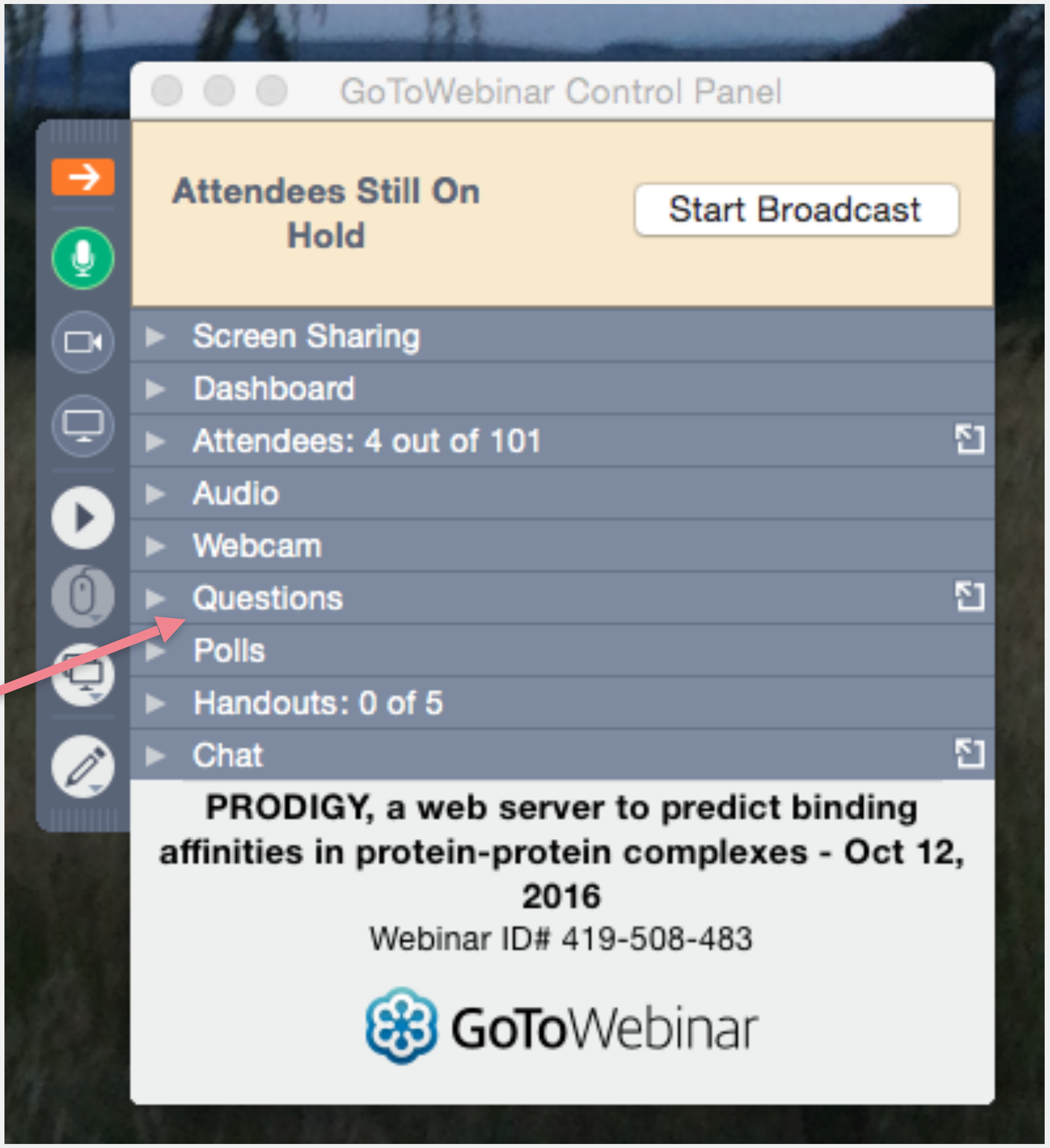
## Documentation

https://docs.mdbenchmark.org

## Audience Q&A session

Please use the **Questions** function in GoToWebinar application

Any other questions or points to discuss after the live webinar? Join the discussion the discussion at http://ask.bioexcel.eu.



PRODIGY, a web server to predict binding affinities in protein-protein complexes - Oct 12, 2016

Webinar ID# 419-508-483

GoToWebinar