

FEniCS in Linux Containers

Jack S. Hale (University of Luxembourg), Liza Li (University of Minnesota), Garth N. Wells (University of Cambridge)

jack.hale@uni.lu, lixx1445@umn.edu, gnw20@cam.ac.uk

Introduction

Docker allows application developers to build, ship and run their application nearly anywhere. We have built three Docker images with FEniCS that are suitable for both end-users and developers:

1. `fenicsproject/stable-ppa`: The latest version from the Ubuntu PPA.
2. `fenicsproject/dev-env`: An environment for building FEniCS from source.
3. `fenicsproject/dev`: FEniCS compiled weekly from master.

With one command, you can launch a lightweight container that provides a consistent and performant environment for developing or using FEniCS.

You will like using FEniCS with Docker because you can:

- Run the same image on any platform.
- See precisely how an image is built.
- Add to or modify the image, or use them as a base for your own custom image.
- Tag an image and send it to your colleagues so that they can run your complex application immediately.
- Push your image to your favourite commercial cloud computing service and run it there.
- Distribute your image with your paper so that others can reproduce your results.

Try it yourself

Docker is available on most platforms, including Linux, Mac OS X and Windows. You can install it by following the instructions for your platform at <http://docker.com>. You can then launch a FEniCS environment using the command:

```
docker run -ti fenicsproject/stable-ppa:latest
```

How do I . . . ?

Share a folder with the container

When you launch the container for the first time you can share any directory on the host into the container:

```
docker run -ti -v $(pwd):/shared \
fenicsproject/stable-ppa:latest
```

This will share the current working directory into the container.

Launch another terminal in a running container

First, check which containers are running

```
docker ps
```

Using the CONTAINER ID field, launch a new bash session using:

```
docker exec -ti b8f51c sudo -u fenics /bin/bash
```

Build FEniCS from source

We provide an image with a ready-to-go development environment for building FEniCS, including an easy-to-modify script to build FEniCS from the master repositories.

Launch a development container using the command:

```
docker run -ti fenicsproject/dev-env:latest
```

Then update the FEniCS build in the container using:

```
$ update_fenics
```

FEniCS is built against up-to-date and optimised stable versions of PETSc and SLEPc.

See how you built the images

Our Docker images are automatically built in the cloud on Dockerhub directly from the Dockerfile, the ‘recipe’ that defines every Docker image. The FEniCS Dockerfiles are maintained at <https://github.com/FEniCS/docker>. Every time we change the Dockerfile, a new image is automatically built. If you visit <https://registry.hub.docker.com/u/>

[fenicsproject/stable-ppa/dockerfile/](https://github.com/fenicsproject/stable-ppa/dockerfile/) and click on ‘Dockerfile’ you can see precisely which steps were used to build the image that you are running.

Install more software into the container

You have two options. The first is to manually install the software into the container, stop the container, then commit and tag the a new image. The second way is to learn how to write your own Dockerfile. For brevity, we only cover the first approach here. If you want to learn about writing your own Dockerfile using ours as a base then take a look at <http://docker.com>.

Begin by launching a container with a name:

```
docker run -ti --name="mycontainer" \
fenicsproject/stable-ppa:latest
```

Now install the software you want and exit the container:

```
$ sudo apt-get install python-pandas
$ exit
```

Commit the image:

```
docker commit mycontainer
```

The above command gives you a hash of the commit. You can tag this hash so that it is easy to remember.

```
docker tag f749 jhale/mycontainer:latest
```

Now you can push it to Dockerhub so your colleague can use it too:

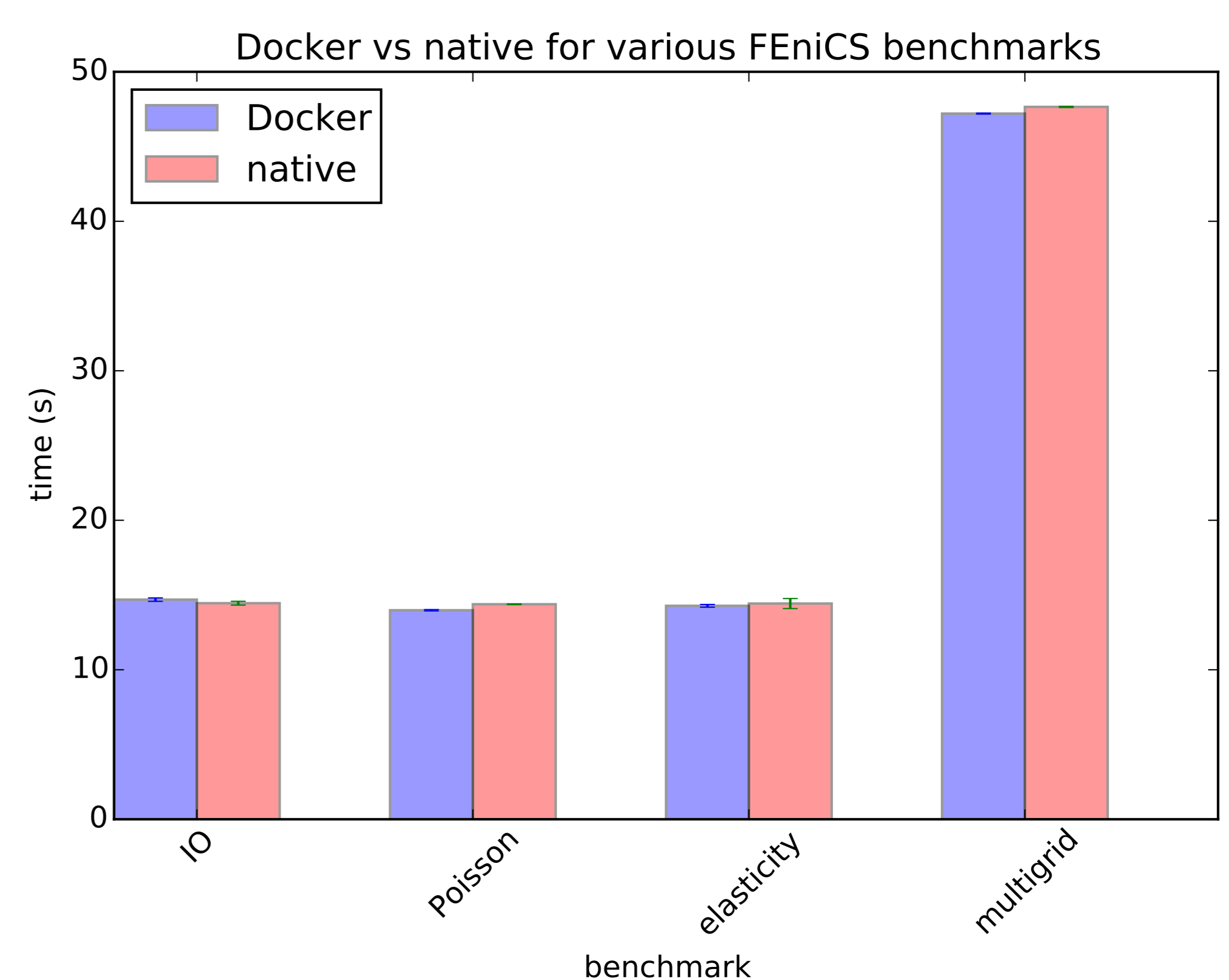
```
docker push jhale/mycontainer:latest
```

Your colleague just has to run:

```
docker run -ti jhale/mycontainer:latest
```

No performance penalty

We tested our Ubuntu-based Docker image against a bare-metal Ubuntu build, with both compiled against the same system packages on the same Intel Xeon-based machine. For a range of benchmarks that tested assembly, linear solvers and IO, we have found almost no measurable performance difference between the Docker and bare-metal builds, in serial or in parallel using MPI.



Because of Docker’s layered filesystem, additional containers take up only a few kilobytes of extra space and launch almost instantly. This means you can happily have hundreds of containers, one for every project if you like, on your development laptop.

Conclusions

We have covered a small number of ways that you can integrate Docker into a powerful workflow for repeatable, easy-to-use and portable scientific computing using FEniCS. For further reading we recommend the Docker documentation at <http://docker.com> and the Docker cheat sheet at <https://github.com/wsargent/docker-cheat-sheet>.

