UNIVERSITY OF BATH

HPCBYTES

RESEARCH COMPUTING NEWSLETTER

ISSUE 4 MAY 2024

In This Edition

Verifying the Results of a Dynamical Quantum Simulator: Using HPC to Explore the Relaxation to Equilibrium of a One-Dimensional Bose Gas

Paul Secular (PhD Researcher, Departments of Mathematical Sciences and Physics)
Email: p.m.secular@bath.edu

Paul is a PhD candidate within the Departments of Mathematical Sciences and Physics who successfully defended his thesis in April 2024. During his PhD, his research focused on optimising and parallelising matrix product state algorithms for quantum many-body physics.

Simulating quantum many-body systems numerically is a difficult problem in general since quantum states are described by multidimensional arrays of numbers (tensors) that scale exponentially with system size. Because of this, physicists have started turning to experimental quantum simulators and quantum computers. These devices have the potential to revolutionise our knowledge of complex quantum systems but are still in their infancy. It is therefore important to verify their accuracy and to test claims of *"quantum advantage"*.

In a recent pre-print [1], Paul revisits a landmark Nature Physics paper [2] that describes *"the first dynamical quantum simulator"* [3]. This experiment used ultracold atoms in an optical lattice to simulate the relaxation towards equilibrium of an interacting one-dimensional Bose gas. Using Bath's high-performance computing (HPC) resources, Paul decided to investigate whether these results could be verified on a non-quantum (i.e. classical) supercomputer. To do this, he represented the state of the system in a compressed format known as a tensor train or matrix product state (MPS). This allowed him to use a parallel implementation [4] of the quasi-exact time-evolving block decimation (TEBD) algorithm [5]. Paul's TEBD code is based on Ref. [6], which uses the Message Passing Interface (MPI). Figure 1 shows the comparison of the simulation results with the experimental data.

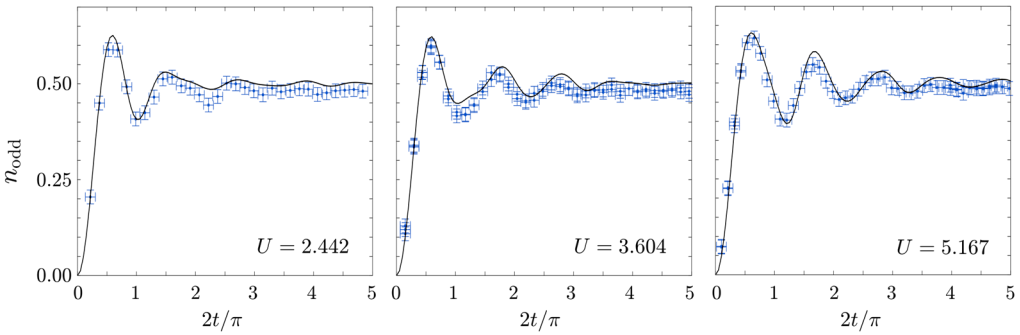


Figure 1: Relaxation with simulation time (*t*) of the mean density of odd lattice sites (n_{odd}) for various interaction strengths (*U*). Markers show the experimental quantum simulator data, while the solid curves are the parallel TEBD results.

For this problem, a weak scaling analysis was possible since different simulations involved differing numbers of particles *N*, where is an *N* odd integer. The effective system size is roughly equal to 2*N*, so one would expect the computation time to grow approximately linearly with *N*. For *N* < *11*, parallelisation was found to be unnecessary, so these simulations were carried out sequentially on a single compute node. For larger systems, Paul used *p* compute nodes, where *p = (N-7)/2*. For *N* between 19 and 39 (corresponding to 6 ≤ p ≤ 16), respectable weak scaling was found as shown in figure 2, with the wall times constant to within a few percent. All simulations were carried out on Balena's Dell PowerEdge C8220 nodes, comprising 64 GB of DDR3 SDRAM and two Intel E5-2650 v2 processors (20 MB cache, 2.60 GHz base frequency), giving a total of 16 cores per node. These cores provided an additional, lower level of shared-memory parallelisation for linear algebra operations via the Intel Math Kernel Library, although the multithreading efficiency was relatively poor.

"HPC aids Paul's research by allowing him to run parallel tensor train algorithms on multiple compute nodes in order to verify quantum simulation experiments"

In contrast to the numerics in Ref. [2], which reportedly took about five weeks of wall time per simulation [7], Paul's parallel calculations required less than 50 hours.

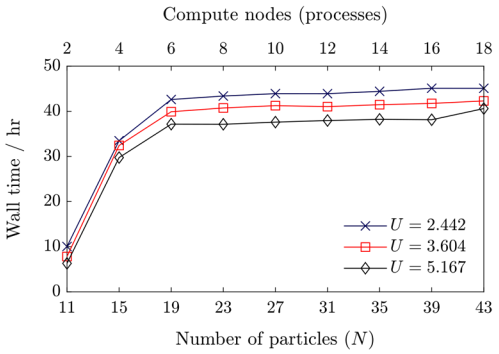


Figure 2: Weak scaling plots of Paul's TEBD code.

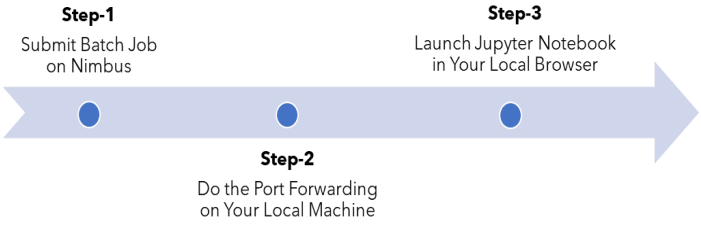
More importantly, the use of HPC allowed Paul to extend the numerics to later times, finding that it is possible to accurately compute expectation values up to the point at which they appear equilibrated experimentally.

References

1. ***P. Secular***, *arXiv:2401.05301 (2024) https://doi.org/10.48550/arXiv.2401.05301.*
2. *S. Trotzky, et al. Nat. Phys. 8, 325 (2012) https://www.nature.com/articles/nphys2232.*
3. *Ludwig-Maximilians-Universität München, Theoretical Nanophysics, https://www.theorie.physik.uni-muenchen.de/lsschollwoeck/.*
4. *M. Urbanek & P. Soldán, Comput. Phys. Commun. 199, 170 (2016) https://doi.org/10.1016/j.cpc.2015.10.016.*
5. *G. Vidal, Phys. Rev. Lett. 93, 040502 (2004) https://doi.org/10.1103/PhysRevLett.93.040502.*
6. *C. Goodyer, Technical Report (NAG Ltd, 2013) http://www.hector.ac.uk/cse/distributedcse/reports/UniTNT/.*
7. *J. Eisert, IBM Research THINKQ Conference on "Approximate quantum computing: from advantage to applications" (2017) https://www.youtube.com/watch?v=m0MOlgn11QA.*

Running Interactive Jupyter Notebook from Your Local Machines on Nimbus

Jupyter Notebook is a useful tool for shortening the feedback loop from coding and results visualization in an interactive way. By launching a "headless" Jupyter Notebook as a batch job, users can tap into Nimbus's robust computing resources while enjoying the familiar Graphical User Interface (GUI) on their local machines' web browsers—be it a desktop or a laptop. This will allow the User to use the Nimbus compute resources without the need of installing any Python packages on the local machine to run the Jupyter Notebook.



Step-1: Submit the Jupyter Notebook Job on Nimbus

```
# The job script to launch Jupyter. Change cores/partition as per your needs and account.
#!/bin/bash
#SBATCH --job-name=jupyter-test
#SBATCH --output=jupyter-test.out
#SBATCH --error=jupyter-test.err
#SBATCH --account=CA-CS1HGN-0XX
#SBATCH --qos=paygo-fsv2-1
#SBATCH --nodes=1
#SBATCH --ntasks-per-node=1
#SBATCH --partition=paygo-fsv2-1
#SBATCH --time=01:00:00

NOTEBOOK_LOGFILE=jupyterlog.out
# get port forwarding info into a file
node=$(hostname -s)
user=$(whoami)
cluster="nimbus.hpc.bath.ac.uk"
port=9300

echo -e "Run the following command from your local machine terminal with local machine port YYYY:
$ ssh -N -f -L YYYY:${node}:${port} ${user}@${cluster}" > port_forwarding.txt

module purge
source /apps/build/easy_build/scripts/id_instance.sh
source /apps/build/easy_build/scripts/setup_modules.sh
module load Anaconda3/2022.10

# Run the Jupyter notebook
jupyter notebook --no-browser --ip=${node} --port=${port} > ${NOTEBOOK_LOGFILE} 2>&1
```

Step-2: Once the job is running, do the port forwarding on your local machine

- Check the "port_forwarding.txt" file made after the job started running:

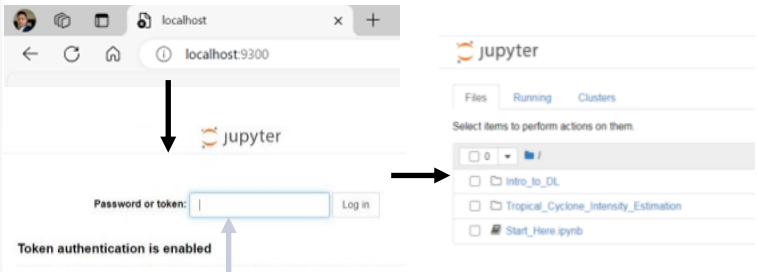
```
[user@nimbus-1-login-1~]$ cat port_forwarding.txt

Run the following command from your local machine terminal with local machine port YYYY:
$ ssh -N -f -L YYYY:nimbus-1-paygo-xxxxxxxx:9333 user@nimbus.hpc.bath.ac.uk
```

- Run the port forwarding command from your local machine terminal (Linux/putty/mobaxterm etc.). Remember to replace YYYY with a local port no. which could be same as Jupyter port on Nimbus. This may ask for your nimbus password.

```
[@localmachinexxx]$ ssh -N -f -L 9300:nimbus-1-paygo-xxxxxxxx:9300 user@nimbus.hpc.bath.ac.uk
```

Step-3: Open web browser in local machine and type localhost:YYYY



- Get the token from jupyterlog.out file and paste

```
[user@nimbus-1-login-1~]$ tailf jupyterlog.out
http://nimbus-1-xxxxxxx:9300/
token=93dd55ca40b2f463fxxxxxxxxxxxxx
```

HPC News and Updates

Isambard 3 coming soon

Isambard 2 will go into maintenance mode before its June end shutdown. From **May 1st, 2024, no new users or updates will be accepted**. Users should move their data to personal storage by June's end. Application process for new user accounts on Isambard 3 will be announced soon.

Tip of the Month

Select your partition on Nimbus

```
# list all available partitions
[user@login1 ~]$ sinfo
# Check details of a specific partition (Max. walltime and nodes/job, tot. nodes, default mem..)
[user@login1 ~]$ scontrol show partition partition_name
```

Acknowledgements

The Research Computing team would like to thank all contributors for the current issue of ***HPCBytes***.

- If you would like to contribute a case study or article to be featured in ***HPCBytes***, please get in touch with the Research Computing team.
- If you would like to hear more, please subscribe to the Research Computing mailing list here: https://forms.office.com/e/rF8rLWbakA

Contact us

Research Computing Team
Digital, Data and Technology
Email: *it-hpc@bath.ac.uk*