Leveraging HPX on a Cluster of Raspberry Pis

Jesse Goncalves[1], Dr. Hartmut Kaiser[2]
[1]Department of Mathematics, Seattle University
[2]Center for Computation & Technology, Louisiana State University

Abstract

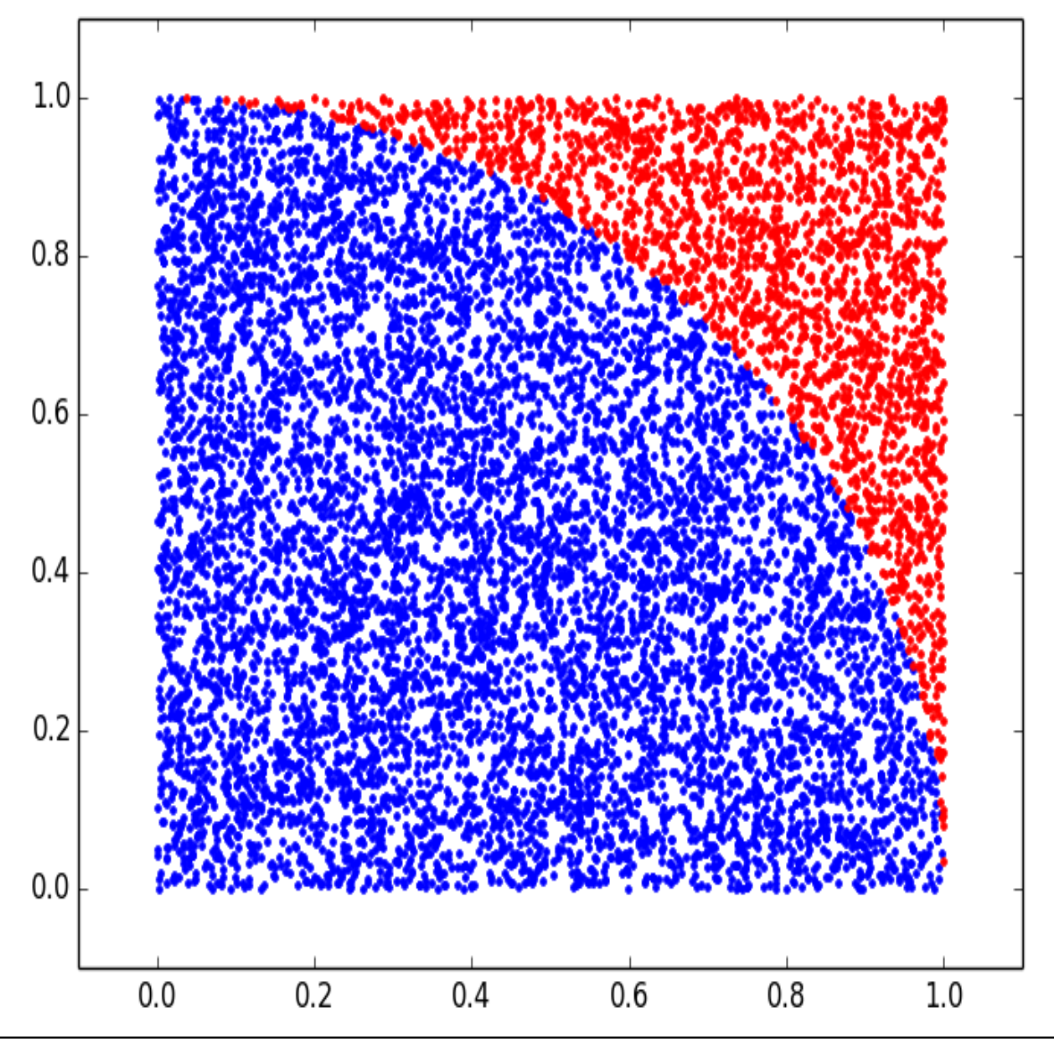
HPX, a C++ runtime system for parallel and distributed programming, is designed to enhance computational efficiency on platforms ranging from multicore PCs to supercomputers. To showcase the portability and performance of HPX, we sought to build and run HPX applications on a cluster of Raspberry Pis. Running on the ARM architecture of a Raspberry Pi and accelerating computation on that architecture shows that HPX can be used to take advantage of even the simplest parallel and distributed hardware structures. Indeed, the HPX applications we tested, which estimate π using a Monte Carlo simulation, showed excellent parallel and distributed scaling on the Pi cluster.

Process

1. Write a serial C++ application to estimate the value of π using a Monte Carlo simulation.
2. Build HPX for the Raspberry Pi and Raspbian OS by directly compiling dependencies and then cross-compiling HPX.
3. Write a parallel implementation of the Monte Carlo method to calculate π using HPX.
4. Test the scalability of the parallel HPX application on a single Raspberry Pi.
5. Write a parallel and distributed HPX application to estimate π.
6. Build a cluster of four Raspberry Pi 3s.
7. Measure the scalability of the parallel and distributed application on the Pi cluster.

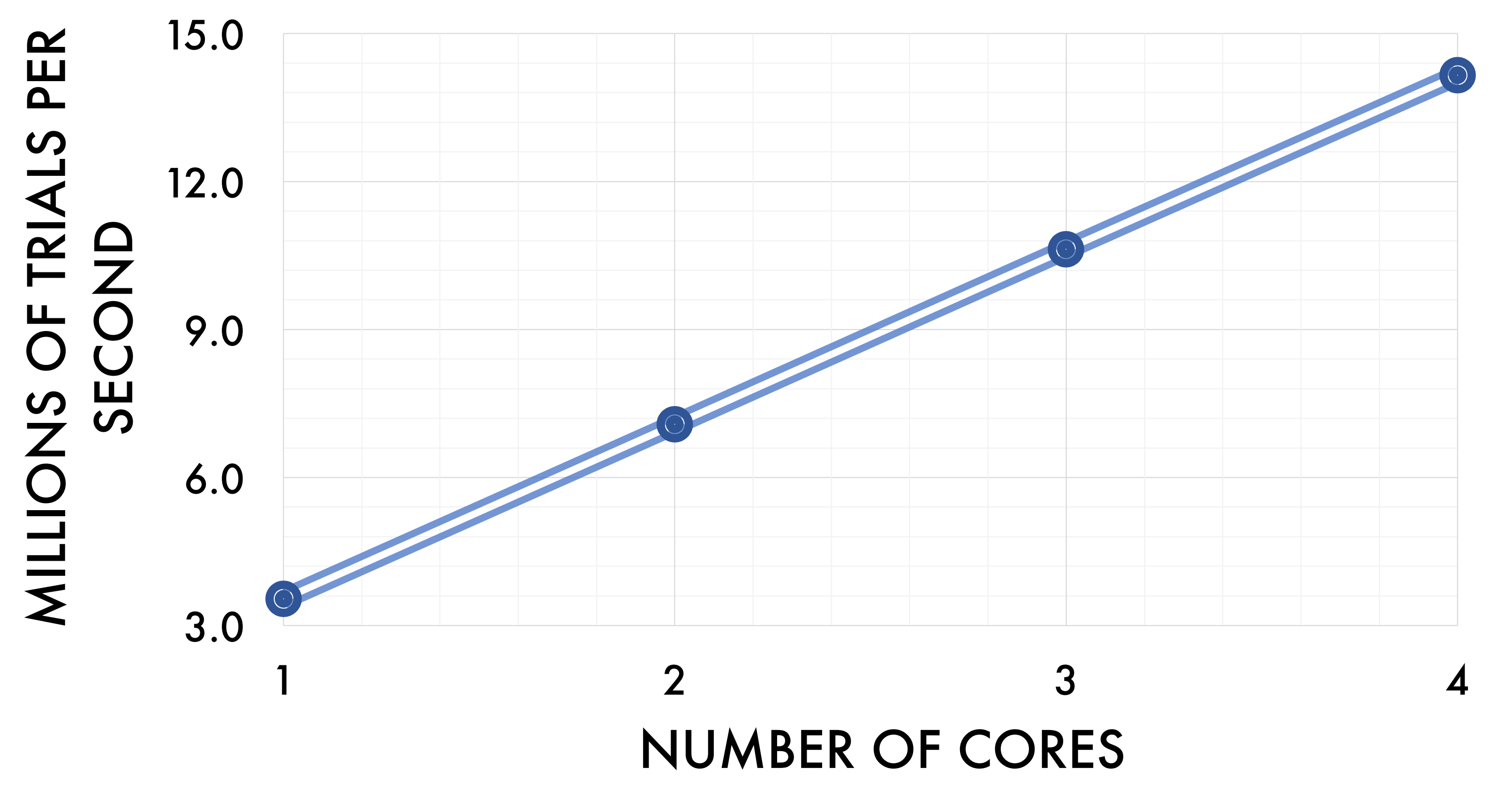
Estimating π

Monte Carlo simulations can be used to estimate the value of π. By generating uniformly distributed points in a 1x1 square, we can estimate the area of a quarter circle with radius 1 by dividing the number of these random points lying inside of the circle by the total number of points (see figure to the right). Since we know the area of a unit circle to be π, the area of the quarter circle is $\pi/4$, so we can multiply the estimate for the area by four to obtain an estimate of π.

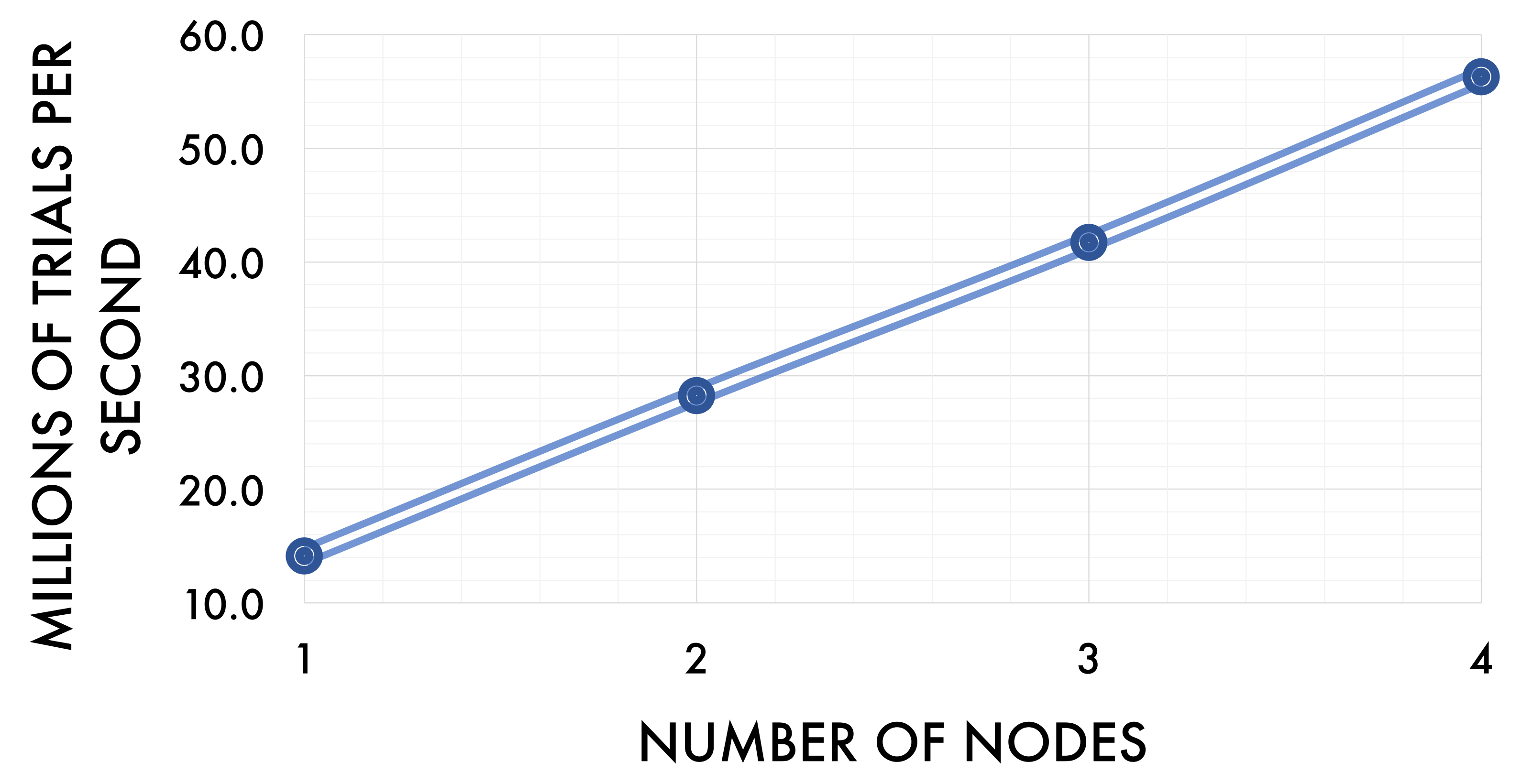


Results

HPX Parallel Scaling



HPX Distributed Scaling



Time Required to Estimate π to Percent Error

Percent Error	1.0E-02	1.0E-04	1.0E-06
Serial Time (s)	7.73	7.73E+04	7.73E+08
Parallel Time (s)	1.93	1.93E+04	1.93E+08
Parallel/Distributed Time (s)	0.49	4.85E+03	4.85E+07

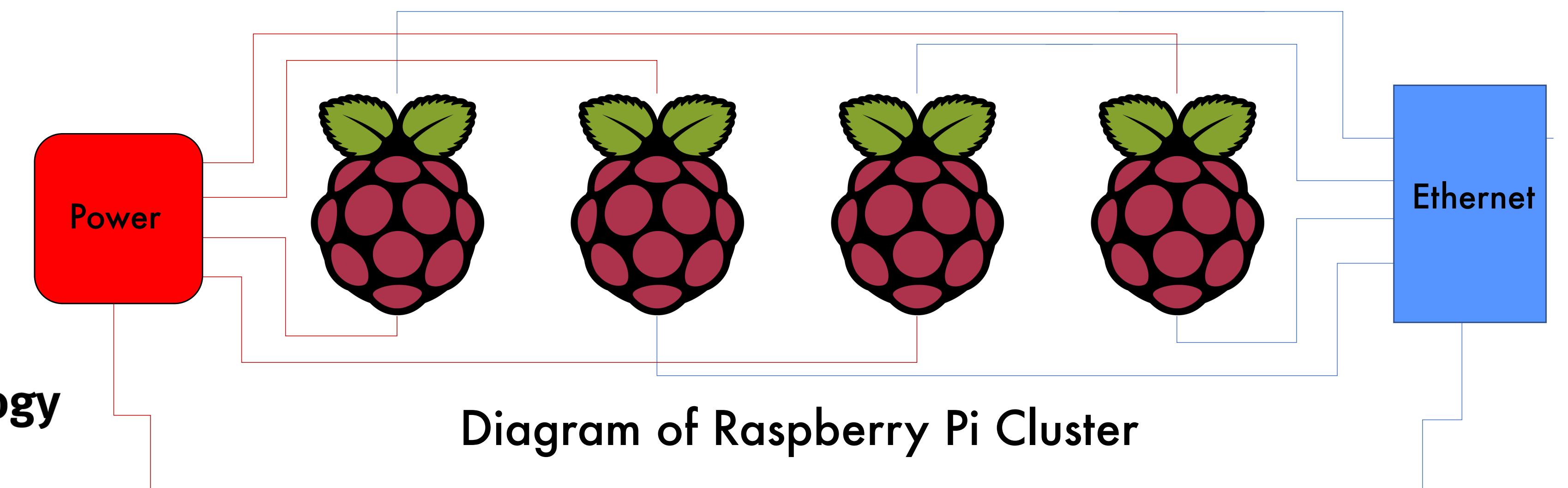


Diagram of Raspberry Pi Cluster

Discussion

Our Monte Carlo HPX applications showed excellent parallel and distributed scalability on the Raspberry Pis, and the scaling efficiency in each case remained above 99% for all of the tests we ran. The scalability of our HPX applications can be observed in the graphs of trials per second vs. number of cores/nodes, a generic measure of scaling. The table of computation times required to reach different percent errors in our calculation of π also showcases the scalability of HPX.

Building HPX for the Raspberry Pis proved to be the most challenging part of the project. We decided to cross-compile HPX because of how long it takes to build, which required a great deal of experimentation with a cross-compilation toolchain and the configuration options of HPX and its dependencies. The procedure we followed can certainly be improved upon in the future, but we have shown that HPX ports to the ARM architecture of the Pis.

Next Steps

- Add more Raspberry Pis to the cluster and test the distributed scaling of our HPX application on a larger number nodes.
- Optimize the process of building HPX for Raspbian so that others can easily follow the same procedure, which could then be added to the HPX documentation.
- Write and test more complicated HPX applications on the Pis.

Acknowledgements

- Thank you to Adrian Serio, Zach Byerly, and many other members of the CCT community for their support on this project.
- This material is based upon work supported by the National Science Foundation under award OCI-1560410 with additional support from the Center for Computation & Technology at Louisiana State University.

References

- Monte Carlo π estimation image: http://glowingpython.blogspot.com/2012/01/monte-carlo-estimate-for-pi-with-numpy.html
- Raspberry Pi logo: https://www.raspberrypi.org/app/uploads/2011/10/Raspi-PGB001.png

STE||AR GROUP LSU Center for Computation & Technology