Analysis of Processing Advantages in Parallel Cluster-Computing in Molecular Dynamic Applications

Kyle Xiao

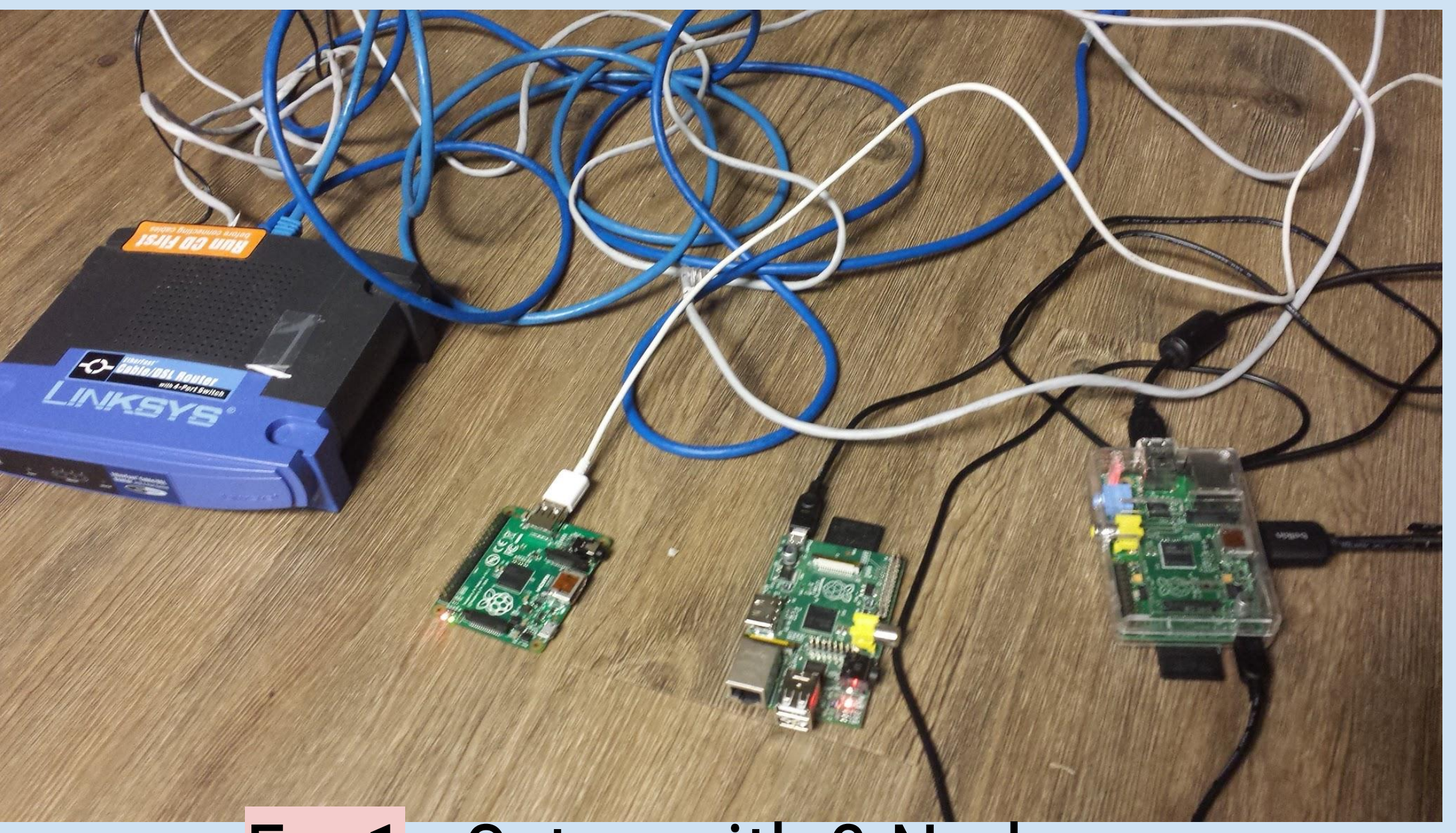
Abstract:

Parallel computing is a method of computation utilizing multiple processors to carry out multiple calculations simultaneously. This framework for supercomputing is commonly used in applications such as topology, discrete mathematics, and molecular dynamics. An example of such a setup is the Beowulf cluster, which are network systems that comprise of usually inexpensive, commodity grade nodes compiled into Local Area Networks (LAN).

Protein folding is the process by which amino acids collapse into a minimum-energy shape. The study of protein folding has been studied extensively by molecular biologists as the shapes of macromolecules have profound effects on biological systems. Protein shapes and interactions have been studies extensively in research for Parkinson's, Alzheimer's, and cancer. By utilizing parallel processing, this group believes it can devise and analyze a way to simulate protein folding for research more efficiently.

Introduction:

Over the course of 3 weeks, this group attempted to find and quantify data on processing advantages in Beowulf cluster network setups as opposed to typical standalone processing units acting individually in terms of concrete applications., preferably a high processing intensity one. The group chose to focus on protein folding simulation, a heavily intensive processing application. The group ran Md5 hash cracking algorithms, specifically dictionary attacks, on the network and utilized processing speeds as benchmarks for performance. Afterward, Molecular Modeling Toolkit (MMTK) in conjunction with RCSB Protein Data Bank was used to construct and manipulate proteins in specific environments while accessing diagnostic information to evaluate levels of performance achieved with parallel computing setups.



Ex. 1 - Setup with 3 Nodes

Methodology:

1. Download an operating system compatible with the Raspberry Pi™ computers, such as Raspbian, on the SD card.
2. Configure the system settings to enable SSH, autologin at startup, and overclock.
3. Install MPI and MPI4PY. Copy the OS image onto a computer.
4. Transfer the image onto each SD card in the node. Connect each pi to a local area network.
5. Using a molecular dynamics library such as OpenMM, MMTK, or MSMbuilder, write a python script to run the simulation. Make sure to include diagnostics such as execution time.
6. On the master pi, install nmap and scan the network for the ip addresses of the other pis. SSH into them and subsequently, change their hostname and copy the script.
7. Create a machinefile with all the IP addresses and execute MPI on the master pi.
8. Record output and analyze further with graphcs viewer such as PyMol.

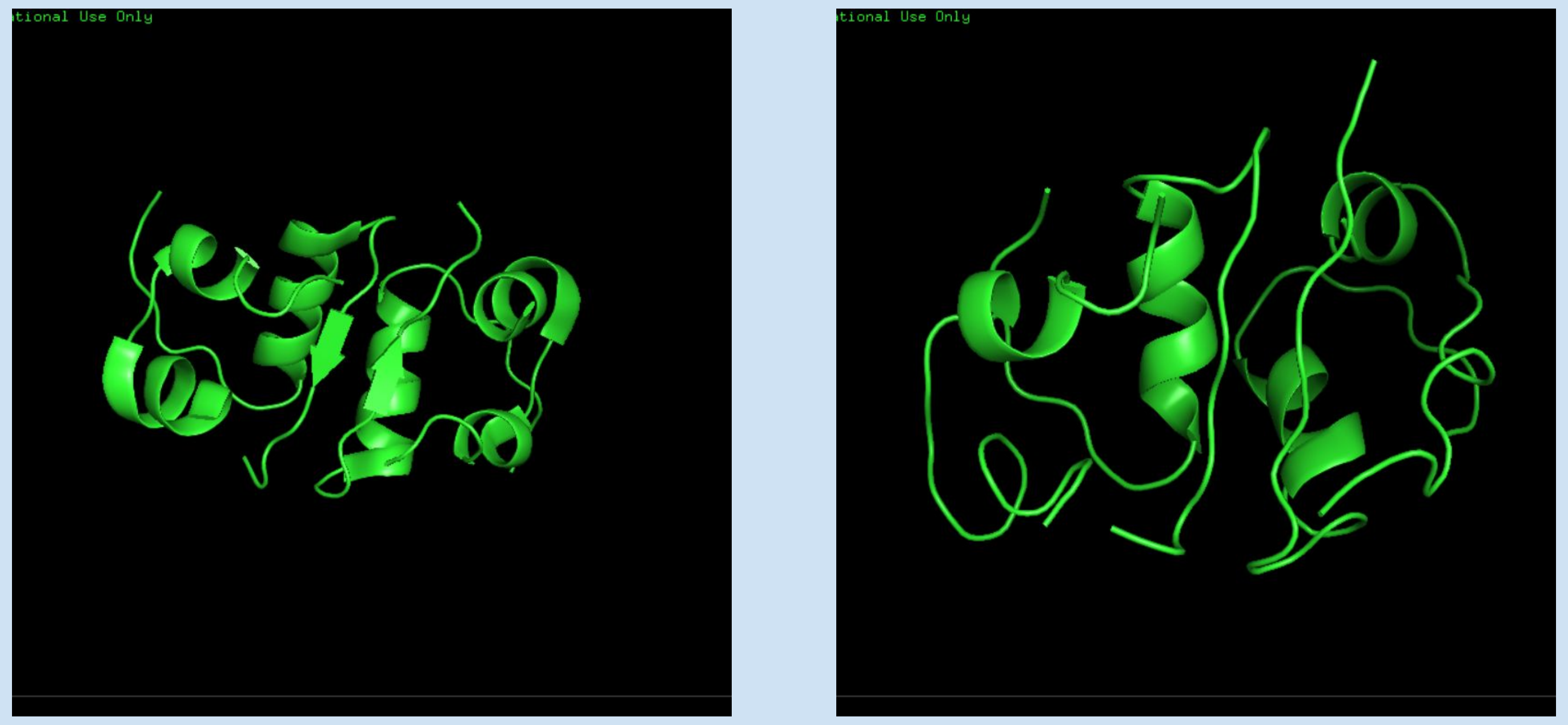


Fig 1.3 -A Comparison of Insulin in Standard and Non-Standard Conditions

Conclusion:

I. The results of the experiment showed a general increase in processing rate as the number of nodes increased, but also seemed to level off at higher numbers of nodes specifically in the dictionary attack on Md5 hashes. We also observed the hypothesized decreasing gains; whether or not this fits the √n hypothesis has yet to be determined due to the fact processing power hit a ceiling as early as 3 node networks. This asymptote is likely due to the inherent latency of the network becoming the primary limiting factor as opposed to the actual computing power. To gain a greater understanding of the processing power function, benchmarks of greater strain must be conducted and analyzed in the future.

II. Furthermore, the protein folding simulation yielded expected results. After running the protein simulation in non-standard conditions such as in vacuum at 50C, the protein underwent significant conformational changes such as beta sheet decomposition and denaturation. Similarly, the group observed great gains in processing power and speed; a protein folding task that normally would have taken several hours took only about 480 seconds using a cluster network.

Results:

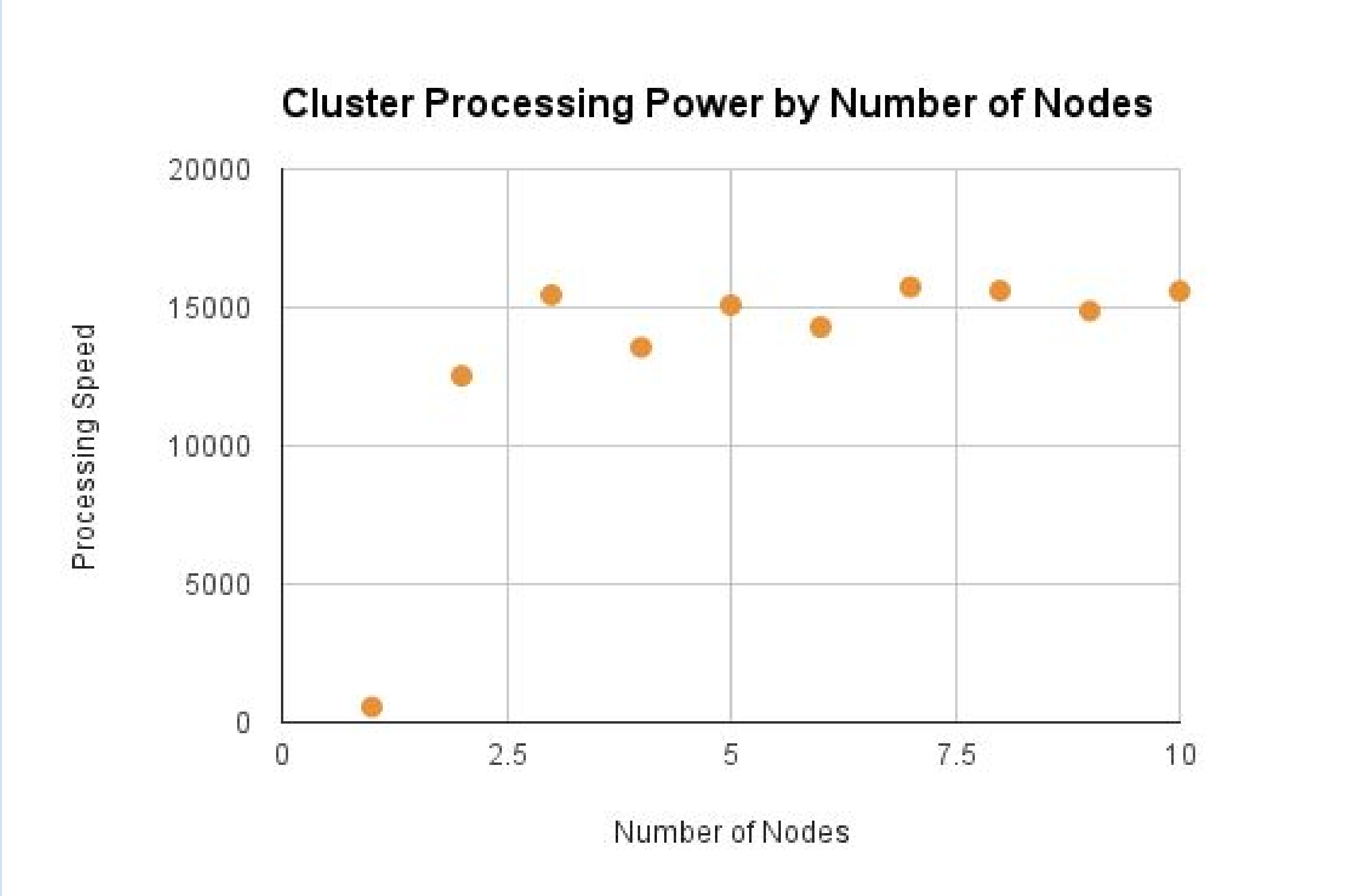


Fig 1.1 - Cluster Processing Power by Nodes

Insulin Folding Energy States			
Step	Time	Potential Energy (eV)	Kinetic Energy (eV)
0	0	-3971.9794	969.7141
100	0.1	-8603.3926	1952.1219
200	0.2	-8546.7829	2904.5617
300	0.3	-7521.3270	3907.7725
400	0.4	-6555.9801	4858.7455
500	0.5	-5376.8842	5833.5975
600	0.6	-5075.6153	5848.3133
700	0.7	-5154.8047	5873.2208
800	0.8	-4976.9578	5884.7813
900	0.9	-4892.8904	5864.1871
1000	1	-5100.7692	5833.4447

Fig 1.2 - Insulin Folding Energy States

References:

Bader, M. (2014). Parallel computing : Accelerating computational science and engineering (CSE).

Chodera, J. D., & Noe, F. (n.d.). Markov state models of biomolecular conformational dynamics. *ScienceDirect*.

Hinsen, K. (n.d.). The Molecular Modeling Toolkit: A New Approach to Molecular Simulations. *Centre De Biophysique Mol´eculaire*.

H.M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T.N. Bhat, H. Weissig, I.N. Shindyalov, P.E. Bourne (2000) The Protein Data Bank

MPI: A Message-Passing Interface Standard. (2012, September 12). *Message Passing Interface Forum*.