CS267 Assignment 0

David Eliahu

BIO

I am a first-semester Master's student in EECS at UC Berkeley, and I received my Bachelor of Science degree in EECS from UC Berkeley in December 2012. I have performed research on the topic of communication-avoiding recursive rectangular matrix multiplication[1,2], and I implemented an algorithm that attains communication lower bounds for rectangular matrix multiplication[3].

My future research plan is to develop a framework for solving "arbitrary" recursive algorithms efficiently. This framework will handle BFS/DFS interleaving[4], parallelization, and autotuning. By leveraging SEJITS[5,6], the framework will allow developers to create specializers to efficiently solve their unique recursive algorithm by simply implementing the framework's API. This is initial-stage research being performed in conjunction with Professor James Demmel, Professor Armando Fox, Shoaib Kamil, Benjamin Lipshitz, Oded Schwartz, and Omer Spillinger.

I entered Berkeley as a Civil Engineering major, and I switched into EECS after thoroughly enjoying E7 (Introduction to Computer Programming for Scientists and Engineers). I have had experience programming in JAVA, C, Ruby, Python, and MATLAB. In this class, I am hoping to learn about a variety of different parallel programming models, techniques, and tools. I would like to increase my efficiency and flexibility when it comes to implementing parallel programs, as well as deepen my understanding of low-level details that affect parallel performance and how to hide these details from productivity-layer developers.

[1] http://www.eecs.berkeley.edu/Pubs/TechRpts/2012/EECS-2012-205.pdf
[2] http://www.cs.berkeley.edu/~odedsc/papers/CARMA%20Poster-SC12
[3] https://github.com/dose78/CARMA
[4] http://www.eecs.berkeley.edu/Pubs/TechRpts/2012/EECS-2012-32.pdf
[5] http://www.eecs.berkeley.edu/Pubs/TechRpts/2010/EECS-2010-23.pdf
[6] http://parlab.eecs.berkeley.edu/sites/all/parlab/files/Bringing%20Parallel%20Performance%20to%20Python_0.pdf

Application: Earthquake Prediction

Earthquake prediction "is usually defined as the specification of the time, location, and magnitude of a future earthquake within stated limits"[7]. This is not equivalent to earthquake *forcasting* (the probabilistic assessment of *general* earthquake hazards), nor is it analogous to earthquake *warning systems* (which provide a few seconds of warning to neighboring areas after the initial detection of ground shaking)[8]. Earthquake prediction must be precise (in terms of time and location) as well as reliable in order to be useful to society.

One theory for earthquake prediction, developed by Professor X. C. Yin in China, is called the "Load-Unload Response Ratio (LURR) Method"[9]. The LURR is a measure of a material's deterioration due to load and strain. This weakening depends on the load acting on the material; as the load increases, the material will pass through three phases: elastic, damage, and failure. In the damage phase, the loading response is different from the unloading response (the loading and unloading moduli differ), and the deformation process is not reversible. Conceptually, the LURR (Y) is a measurement of the difference between the loading and unloading responses in the damage (inelastic) phase.

Zhuang and Yin define a critical value of LURR (Y_c) that depends on the number and relative locations of recent earthquakes. The ratio Y/Y_c is hypothesized to be an indicator of imminent earthquake risk[10].

As prediction areas grow, time horizons lengthen, and longitudinal/latitudinal measurements become more precise, computation of the LURR ratio becomes infeasible on a single CPU. The most time-consuming component of the LURR procedure is the calculation of the distance between the epicenter of a previous earthquake and a given location. This distance is computed with the following equation:

$$d = R_e \times \sqrt{\cos^2(\mathrm{lat}_z) + \cos^2(\mathrm{lat}_c) - 2\cos(\mathrm{lat}_z)\cos(\mathrm{lat}_c)\cos(\mathrm{lon}_z - \mathrm{lon}_c) + (\sin(\mathrm{lat}_z) - \sin(\mathrm{lat}_c))^2},$$

where R_e is the Earth's radius, lon_c and lat_c are the longitude and latitude of the epicenter, and lon_z and lat_z are the longitude and latitude of the given location. Another expensive operation is the calculation of the loading and unloading responses at a given location:

$$E = \sqrt{10^{(11.8+1.5M)}} = 10^{(11.8+1.5M)/2} = 10^{5.9+0.75M},$$

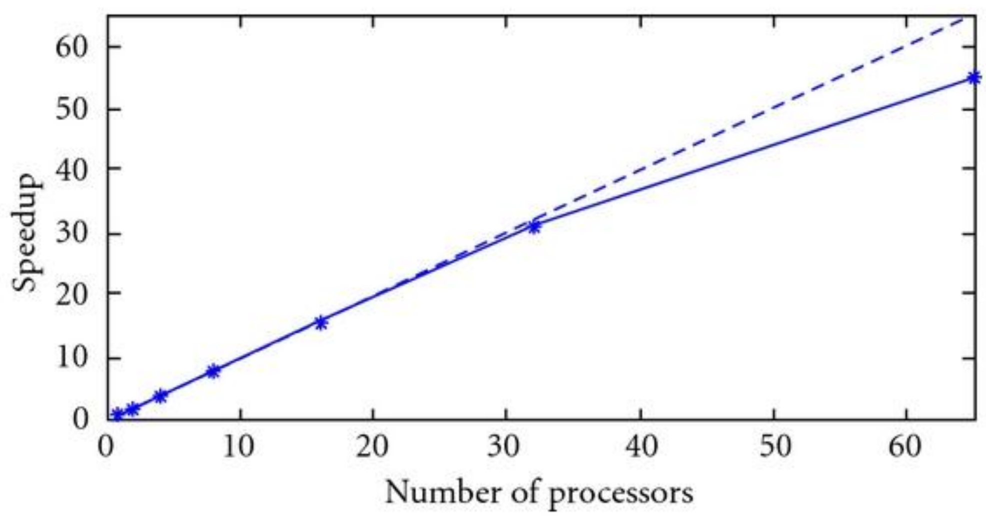
where M is the Richter magnitude scale.

[7] http://moho.ess.ucla.edu/~kagan/Geller_et_al_1997.pdf
[8] http://en.wikipedia.org/wiki/Earthquake_prediction
[9] X. C. Yin, "A new approach to earthquake prediction," *Earthquake Research in China*, vol. 3, pp. 1–7, 1987.
[10] J. C. Zhuang and X. C. Yin, "Random distribution of the Load/Unload response ration(LURR) under assumptions of poisson model," *Earthquake Research in China*, vol. 15, pp. 128–138, 1999.

There has been one published description of the parallelization of the LURR method, performed by Yangde Feng, Guoliang Ji, and Wenkai Cui in China[11]. They parallelized the computation by partitioning the spatial region into P regions and assigning each region to one processor. Neighboring regions do not overlap, and region boundaries are selected to reduce communication across regions. Each step of longitude and latitude scanning requires that each processor individually compute values for its region and store the results in temporary files. At the end of the computation, one processor writes the final output.

The researchers used MPI on the DeepComp 7000 Cluster, which has 12216 total cores and ranked #293 in the top500 list at 102.8 TFlop/s (as of November 2012)[12]. They tested the LURR with earthquake data from 1990 to 2004 in mainland China. The diagram below shows the strong scaling plot. As the researchers explain, scaling suffers as the number of processors exceeds 32 due to communication costs.

Speedup (Strong Scaling)



Although the parallelization of this algorithm was moderately successful for a small number of processors, it is clear that scaling will become an issue at larger scale. The researchers should consider communication-avoiding techniques that will reduce communication and increase performance. For example, the researchers may consider overlapping regions to allow for more computation to be performed before synchronization and communication are required.

[11] Yangde Feng, Guoliang Ji, and Wenkai Cui, "Parallel Computing for LURR of Earthquake Prediction," *International Journal of Geophysics*, vol. 2012, Article ID 567293, 3 pages, 2012. doi:10.1155/2012/567293
[12] http://top500.org/system/9868

It is important to note that the LURR method's ability to predict earthquakes has been formally challenged. Despite Professor X. C. Yin's validation of the LURR method, Andrea Donnellan, Peter Mora, and Mitsuhiro Matsu'Ura refute its effectiveness in their book *Computational Earthquake Science, Part 2*[13]. They write:

> *Although the weight of evidence does not support a general correlation between seismicity and tides, it is still possible that this correlation may develop locally before a large earthquake as proposed by YIN et al. (1995). Called the Load/Unload Response Ratio (LURR) effect, it proposes that seismic activity in a region surrounding a future earthquake becomes relatively greater during periods of tidal loading than during periods of tidal unloading in the year or so before the event...Because of the important implications for the predictability of large earthquakes, we undertook a re-analysis of five major earthquakes in California for which significant LURR effects bad been published…We found that fluctuations in the LURR function were primarily controlled by a small numbers of moderate earthquakes, that the results were not robust with respect to choice of area or time interval, and that the choice to plot the LURR ratio on a linear scale biases the display of results.* ***It is our opinion that the reported anomalous behavior of LURR prior to large earthquakes is of no predictive significance.***

[13] Donnellan, Andrea, Peter Mora, and Mitsuhiro Matsu'Ura. *Computational Earthquake Science, Part 2*. N.p.: Springer, 2004.