

Introducing HPC to Young Students

Russell Feldhausen
Kansas State University
russfeld@ksu.edu

Scott Bell
Kansas State University
rsbell@ksu.edu

Daniel A. Andresen
Kansas State University
dan@ksu.edu

Introduction

Graphical development environments such as Scratch [4] allow beginning students to explore complicated computing concepts without some of the syntactical struggles often associated with typical first programming languages such as Java or C. As part of an outreach activity, we developed an application that simulates wind forecasting in Scratch, taking advantage of its multi-threading capability to demonstrate the power and benefits of this programming concept. This poster presents our presentation approach, the application, and the outreach activity.

The outreach program, which is designed for middle and high school girls, provides attendees with brief (40-50 minute) exposure to various STEM programs across campus. The goal is to spark an interest in STEM disciplines among the attendees in the hope that they will pursue a future in one of these areas. This program is sponsored by the Kansas State University office for the Advancement of Women in Science and Engineering (KAWSE) [1]. This year over 40 students attended the computing component of the program.

We chose to use Scratch based on our experiences using it in several other programs. It has been used successfully in 4th to 12th grade classrooms as part of our GK-12 INSIGHT program. [2] It has also been used in summer programs within the department of Computing and Information Sciences and is currently used in our introductory computer science course. The Scratch environment provides access to many complex computer science concepts but removes the need for students to write extensive syntax. Students are able to focus on problem solving rather than spending a large percentage of their time battling syntax errors. While we did not require students to build any code for this project, the intuitive structure of Scratch and the simple graphical interface made it an ideal choice for this situation.

Instructional Approach

Given our limited classroom time, the goal was to give students a brief glimpse of the ways that high performance computing is currently used, followed by a tangible experience with this technology that might help build their interest and confidence in becoming participants within the field. The presentation was divided into three components:

- We began each session with a brief overview of some ways high performance computing (HPC) is currently utilized in our society. This included several examples

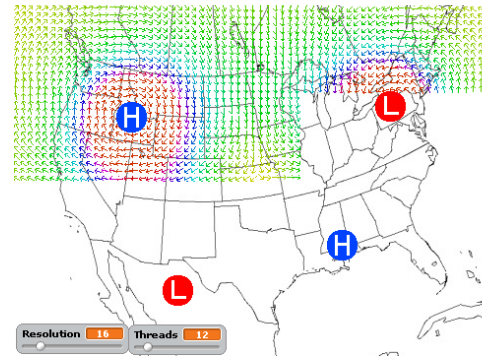


Figure 1: Application running, showing progression of 12 separate threads, each producing calculations for a row within the model

which demonstrate the power and usefulness of HPC. For example, the Watson computer system developed by IBM is now being used to help with decision making in the health care industry. Another example is the computer system used by the National Center for Atmospheric Research (NCAR) to model weather patterns and generate forecast materials used by meteorologists around the world.

- Next, students toured the Beocat cluster, [3] Kansas State's cluster computing system which is utilized by departments across campus for everything from genome research to watershed modeling and simulating the spread of plant pathogens. Students were able to experience the noise and heat produced by over 2000 cores: a glimpse at the scale and computing power needed to solve today's complex problems.
- We ended each session with a hands-on experiment using the multi-threaded wind forecast model built in Scratch. Users can vary the resolution of the model output as well as the number of threads that are utilized in computing that output. The application measures the amount of time it takes for the model to run to completion. Progress is shown on the screen via updates to a weather map showing wind vectors at each point on the map as can be seen in Figure 1.

The Application

The application we developed was a very simplified example to show how organizations such as the National Weather Service might approach the modeling of weather patterns.

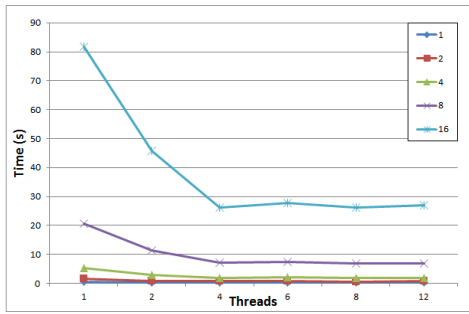


Figure 2: Application time to completion for a given number of threads is shown for each resolution tested.

As Figure 1 shows, we created four high and low pressure centers which were superimposed on a map of the United States. Users are able to relocate these centers as they desire. The user can also select the model resolution they wish to use during the model run (we restrict this to a value between 1 and 16). This will determine the number of data points that must be calculated $n = (4 * res)^2$.

For each data point, a wind direction and relative strength is determined. We take great liberties in this calculation, not incorporating pressure values, only using simple interpolation based on the distance and direction from each pressure center to the point currently being calculated. We were more interested in students understanding the benefits of multi-threading and how this is used in solving real world problems than getting them to understand the complexities of true weather calculations.

The user is also able to select the number of threads they wish to utilize when the program runs. This number is restricted to values between 1 and 12. When the application runs, each thread calculates data points for several rows of values. Representations of these values are plotted on the screen as they are calculated, allowing the user to visualize the progress of the individual threads over time (see Figure 1). For example, when the number of threads is set to 12, students are able to see 12 arrows indicating wind vectors appear nearly simultaneously. Then, as each of the first 12 rows are completed, the next 12 rows begin appearing (showing that the 12 threads have each moved on to their next assigned row of calculations). In a very tangible way, this demonstrates how calculations can be parallelized yet the results can still be composed into a single, coherent output. When the model completes all calculations, it then displays the elapsed time.

The Experiment

Students worked in pairs, with each pair being assigned a computer with the application loaded. Each pair of students was given a specific resolution and asked to run the model multiple times, varying the number of threads used then recording the time to completion for each iteration. This data was plotted to show the change in time based on the number of threads for a given model resolution. As Figure 2 shows, the results are very consistent and interesting.

There is significant improvement in performance as we progress from 1 to 4 threads, but the performance then levels out,

showing only slight variations and no real improvement in performance after we reach 4 threads. We discussed possible reasons for this behavior with the students. It turns out that the systems we were running this software on were quad core machines. This test was very effectively showing us the benefits of having multiple cores and utilizing a multi-threaded application in order to distribute the computations over these resources! This idea tied together nicely with the initial presentation and the tour of Beocat, allowing the students to understand the computational power that a system with over 2000 cores might have if we can find a way to efficiently use all of the available resources.

Outcomes

Over 40 girls attended our sessions during the program, and everyone was able to complete their data collection within the limited time we had available. They were able to visually see the benefits of a multi-core system and also gain an understanding of some of the limitations to these benefits. Students were surveyed at the end of the GROW Summer Workshop, and 22 of 41 indicated that they had an interest in a job using HPC to solve problems while 27 of 37 felt they could learn to write computer programs.

Future Plans

We would very much like to have more time for this activity. Another 20 minutes would have allowed students to plot their own data and have a more in-depth discussion of the outcomes. For most sections, we had to rush to plot one set of data points on the board and discuss what the data was showing. While this is a great activity, it would have been useful for the students to have time to discuss and think about the results more thoroughly at the end.

We plan to incorporate this application into the introductory programming course in our department. The program will be modified to utilize more accurate calculations for modeling wind dynamics and the students will be required to actually implement the code for this calculation and possibly develop the code that coordinates the multiple threads. This will provide an excellent introduction to the concept of high performance computing for this course.

References

- [1] K-State Office for the Advancement of Women in Science and Engineering. <http://www.k-state.edu/kawse/>.
- [2] Kansas State University. INSIGHT an NSF GK-12 Program. <http://gk12.cis.ksu.edu/>.
- [3] Kansas State University Computing and Information Sciences. Beocat. <http://beocat.cis.ksu.edu/>.
- [4] MIT Lifelong Kindergarten. Scratch. <http://gk12.cis.ksu.edu/>.