Design Study - Team 10

Project2: Heated Earth – October 21, 2014

Timothy Laurent (tlaurent3)
Prabhakar Kallakuri (pkallakuri3)
John Swenson (jswenson8)

In this design study we examine the affects of different control structures on the performance of a representation of a simulated Earth, as well as the affect of buffer size on the throughput of the application. The program will examine having the data producer control the data consumer, the consumer control the producer, and a master class control both the producer and consumer. The program will also investigate the buffer between the consumer and producer, and how it's size affects the program throughput.

Context

To investigate these issues we will examine the calculation of the heat fluctuations on a simulated Earth as it rotates over time and is heated by the Sun. The simulation is initialized with the Sun over the Prime Meridian and the Earth at a uniform temperature of two hundred and eighty eight Kelvin. The heating of the Earth is then simulated at discrete time intervals, specified by the user. The calculation is simplified by treating the Earth as if it were spherical and not tilted, all area on the Earth reacts in an identical way to heat, and the Earth rotates uniformly once per twenty four hour period.

Throughout this study we examine the affects of initiative, considered the control of a program, on that program's performance. We also examine the affects of threading, defined as having a component on a separate execution thread from the main program, on the program's performance. Between the data consumer and producer exists a buffer which holds data moving between the two areas of control. We examine the affects of the size of that buffer on the program performance.

Research Questions:

1. What is the affect of different initiative configurations on the memory usage of the Heated Earth Simulation?
2. What is the affect of different initiative configurations on the execution time of the Heated Earth Simulation?
3. What is the affect of different initiative configurations on the structure and elegance of the Heated Earth Simulation architecture?
4. What is the affect of using threads to provide concurrent computation on the execution time of the Heated Earth Simulation?
5. What is the affect of using threads to provide concurrent computation on the memory usage of the Heated Earth Simulation?
6. What is the affect of different initiative configurations on the structure and elegance of the Heated Earth Simulation architecture?

7. What is the affect of buffer length on the idle time of the data producer and data consumer of the Heated Earth Simulation?
8. What is the affect of buffer length on the throughput of the Heated Earth Simulation?
9. What is the affect of the initial conditions on the amount of time it took for the Heated Earth Simulation to stabilize?

Subjects:

The Heated Earth simulation is instantiated through the command line, and differing command line arguments instantiate different architectures. The program consists of the Simulation, the calculation of temperatures across the Earth as a function of time, the Presentation, the display of these temperatures to the user, the buffer, which transmits data from the Simulation to the Presentation, and the main thread. The user can specify that the Simulation and or the Presentation be executed in a separate thread, as well as the size of the buffer.

Experimental conditions:

The system was tested on a 64-bit Ubuntu 14.04 LTS virtual machine using VMWare Fusion. The system was accessed to a processor core with 1024 MB of memory. The host machine runs a Intel Core i7 processor at 2.5 GHz with 16 GB of 1333 MHz DDR3 RAM on OS X 10.9.4.

Java 1.6 was used with no compiler options besides specifying the classpath. The execution parameters varied according to individual test run; these values are provided in the method section. During execution the virtual machine was the only user started process on the host machine.

Variables:

Variable Name	Type	Metric
Grid Spacing	independent	degrees (1-180)
Buffer Size	independent	items (1+)
Initiative	independent	mode (Simulation, Presentation, or other)
Threading Configuration	independent	true for false for Simulation, Presentation or both.
Initial Earth Temperature	independent	degrees kelvin
Memory Usage	dependent	bytes
Performance	dependent	seconds
Interdependence's	dependent	references
Lines of Code	dependent	lines
Time to Sabilization	dependent	millisecond
Throughput	dependent	items per millisecond

Metrics

1. Grid Spacing: The number of degrees that each grid cell covers.
2. Bytes: a grouping of eight bits of data.
3. Millisecond: a thousandth of a second.
4. Reference: A class is defined as having a reference for every other class to which it either instantiates or uses.
5. Line: A line of code is defined as a non comment line of executable java code.

Variable Summary:

Question #	Independent Variables	Dependent Variables
1	Initiative	Memory Usage
2	Initiative	Performance
3	Initiative	Lines of Code, Interdependencies
4	Threading Configuration	Performance
5	Threading Configuration	Memory Usage

6	Threading Configuration	Lines of Code, Interdependencies
7	Buffer Size	Idle Time
8	Buffer Size	Throughput
9	Buffer Size	Time to Sabilization

Method:

The simulation programs be invoked using commands with the following format:

java EarthSim.Demo -s -p -r -t -b <Integer>

s indicates that the Simulation should run in its own thread

p indicates that the Presentation should run in its own thread

t: indicates that the Simulation, after producing an updated grid, should instruct the to consume the data

r: indicates that the Presentation, after completing the display of a grid ,should instruct the Simulation to produce another

b:indicates the size of buffer.

Analysis Techniques

The primary stage in the data analysis is done during data collection. As mentioned in the previous section, an effort is made to eliminate outliers that may cause difficulties in calculating correlations. Each run of 5 measurements must fall with in the specified standard deviation threshold to be accepted. Otherwise, the average of the five runs is used as a value for plotting.

The performance and memory data then will be placed into a scatter plot and fitted with a curve. We hypothesize that both performance and memory usage will scale exponentially, so each data set will be exponentially regressed using tools provided by Google Charts. The analysis was primarily to identify the program with the lowest exponential coefficient, that is, as more dimensions are added, the program requires less time or memory than others.

The curve fit and error rate will be measured by the correlation coefficient of the regressed curve. If the R^2 value is greater than 95%, we accept the proposed curve as illustrative of the data.

Recommendation:

When the system used to run the Heated Earth Simulation has multiple cores available having both the simulation and presentation modules on separate threads is the recommended configuration. When three or more logical cores are not available the number of separate threads should be matched to the number of cores, specifically if there are two cores the simulation should have it's own thread, and if there is only one core neither the simulation or presentation should have it's own thread.

Coarseness of the lattice:

The cell size varied between one degree and one hundred and eighty degrees. As the cell size decreased the complexity of the simulation increased in an exponential manner. The smaller cell size allowed for more precision in the calculation, as each actual point on Earth was closer to the point at which the calculation was made to determine it's temperature.

Initial conditions:

Buffering: The size of the buffer has a direct impact on the idle time and throughput. With higher buffer, idle time between simulator and presentation process is significant because the presentation thread has to wait for the buffer to fill before it can consume the data. Also simulation thread would have to wait for long time until the buffer is emptied by presentation thread before it can fire the simulation process for the next cycle.

Since both presentation thread and simulation thread wait on each process to complete, the overall throughput of the application is slow.

Results:

Adaptability:

Functional Requirements

Scenario	Description	Effect on Design
Add Algorithm or domain model for calculation of temperature of earth	Adding another algorithm implementation that simulates heating of earth.	The code which runs simulations would have to be modified to execute the new algorithm to calculate heating of earth.
3D Heat Earth	Add an algorithm that functions on a three dimensional grid .	The GUI would have to be adapted to show temperature of earth in form of 3 Dimensional representation. The user

		would need to execute simulation threads accordingly.
Displaying the temperature at physical location along with the physical location name and address	This feature would enable the user to see temperatures at specific location upon placing the cursor i.e by city ,country	The simulation would calculate the temperatures at all locations around the earth and display the physical location upon placing the cursor on the presentation screen.
Displaying the toggle or simulation options in 3 dimensional or pictorial way by showing execution time and user friendly messages	Currently, the user does not have enough information on the elapsed time after executing a particular command .Instead of this, we would display enough messages on screen asking user to either continue or pause or stop execution .	The GUI would have to be modified to specify a user friendly message or information about time spent. For the simplest case, after every 10 seconds a user friendly message would be thrown asking the user either to continue or stop the execution process. But, in a more advance version, multiple messages would be displayed and handled accordingly.

Non -Functional Requirements

Scenario	Description	Effect on Design
Increase Usability of GUI	Currently, the GUI does not provide any method of determining specific cell values. One possibility for addressing this would be to show the value of a cell in a tooltip when the user hovers over it. Also, feedback from user studies could be obtained to make the GUI	Adding tooltips or otherwise changing the GUI ought to usually be fairly straightforward and not have deep architectural implications. Most of the effort would go into getting feedback from users and reconciling conflicting advice.

	otherwise more user friendly.	
Increase Extensibility Via Plug-ins	The proposal here is to increase extensibility by having new algorithms implement a common interface and performing a standard installation procedure.	To enable this, the current implementation would have to be carefully scrutinized for various implementations. Currently ,we are using multiple threads to calculate temperature.This can be extended for multiple algorithms and results can be analysed to identify optimum algorithm

Conclusion:

Because we were unable to complete the project we could not generate data in order to conclusively answer our research questions. However, due to our experience in the field and ability to apply the knowledge we've gain in this course we can posit the answers to the questions.

The different initiative configurations should not contribute to the memory usage of the Heated Earth Simulation. Because of the way the program is structured there is no difference on the amount of items in memory due to the initiative configuration. Similarly there should be only slight differences on execution time because of different initiative configurations, and only when threading is used. When the initiative is neither with the presentation or the simulation, and both the presentation and simulation have separate threads, the execution time is decreased. This is because the workload of maintaining the initiative does not impact the calculations. There was no affect on code elegance as we measured when comparing different initiatives. This was a failing on our data to accurately capture the differences. Because we were only using one program we were not able to use the same methods of comparison as in the previous project.

When multiple cores are present using multiple threads should decrease the execution time of the Heated Earth Simulation because of the loadsharing. Multiple cores also should decrease the memory usage of the Heated Earth Simulation, but will not affect the structure and elegance of the architecture.

Up to a point having a larger buffer length should decrease the idle time of the data producer and data consumer. After that break even point there should not be any affects from increasing the buffer length. A similar situation holds true for the throughput of the Heated Earth Simulation. The farther the initial temperature of the Earth, or the initial conditions, is from the temperature generated by the Sun radiating the Earth the longer the simulation should take to stabilize.