Rootbeer:

Seamlessly using GPUs from Java

Phil Pratt-Szeliga, Jim Fawcett, Roy Welch.
Syracuse University, Syracuse, NY USA

Rootbeer Overview

- Rootbeer allows a developer to program GPUs in Java
- It does much more than Java language bindings
- Almost any existing Java program can easily be "cut" into CPU/GPU parts using Rootbeer
- With Rootbeer
    - Kernels are written in Java
    - Graphs of complex types are serialized for you

Rootbeer Programming Interface

- All gpu kernels implement the Kernel interface

```
public interface Kernel {
    void gpuMethod();
}
```

- A static analysis/transformation on the Java Bytecode is done (using Soot) starting at each class implementing the Kernel interface

(soot: http://www.sable.mcgill.ca/soot/)

Rootbeer Programming Interface

- Fields accessible from Kernels are copied to the GPU

```
public class ArraySum implements Kernel {
  private int[] source; private int[] ret; private int index;
  public ArraySum (int[] src, int[] dst, int i){
     source = src; ret = dst; index = i;
  }
  public void gpuMethod(){
     int sum = 0;
     for(int i = 0; i < array.length; ++i){
        sum += array[i];
     }
     ret[index] = sum;
  }
}
```

Rootbeer Programming Interface

- Then Kernel class objects are made and run

```
public class ArraySumApp {
  public int[] void sumArrays(List<int[]> arrays){
    List<Kernel> jobs = new ArrayList<Kernel>();
    int[] ret = new int[arrays.size()];
    for(int i = 0; i < arrays.size(); ++i){
      jobs.add(new ArraySum(arrays.get(i), ret, i));
    }
    Rootbeer rootbeer = new Rootbeer();
    rootbeer.runAll(jobs);
    return ret;
  }
}
```

Running the GPU program

- Run the static Rootbeer translator

  -
    ```
    $java -jar Rootbeer.jar InputJar.jar OutputJar.jar
    ```
- Run your program as normal

  -
    ```
    $java -jar OutputJar.jar <cmd line args>
    ```
- Execute built in tests

  -
    ```
    $java -Xmx20g -jar Rootbeer.jar -runtests
    ```

Java Language Features Supported

- Instance and Static Methods and Fields
- Array Types of any Dimension of any Base Type
- Composite Reference Types
- Arbitrary Object Graphs including Cycles
- Synchronized Static and Instance Methods
- Locking on an object
- Inner Classes
- Dynamic Memory Allocation
- Strings and Exceptions
- Null Pointer/ Out of Memory Exceptions are thrown

Unsupported Java Features

- Reflection
- Native Methods
- Sleeping inside a monitor
- Dynamic Method Invocation
- Garbage Collection (future work)

Rootbeer Testing

- Test Driven Development was used
  - "If you haven't tested it, it doesn't work guaranteed" [Weste93]

- 21k lines of product code
- 7k lines of test code (39 high level test cases)
- All tests pass on Windows and Linux

High Performance Serialization

- Every piece of data is read from a field before serialization to GPU memory
- Performance every possible way to read a Java field:

Method	Execution Time (ms) (10,000,000 reads)
JNI	247
Reflection	173
Pure Java	5

- Rootbeer generates Java Bytecode that can serialize objects you want on the GPU using pure java

Rootbeer Performance 1/3

Dense Matrix Multiplication: 67X faster

4096x4096 matrices

System	Time
Java Only	58 minutes
Java with Rootbeer	52 seconds

Details

Event	Time (ms)
Rootbeer Serialization	557
Rootbeer GPU Execution	51,687
Rootbeer Deserialization	20

Rootbeer Performance 2/3

Brute Force Fourier Transform: 54X faster

n = 114688

System	Time
Java Only	78 minutes
Java with Rootbeer	87 seconds

Details

Event	Time (ms)
Rootbeer Serialization	15
Rootbeer GPU Execution	87,220
Rootbeer Deserialization	20

Rootbeer Performance 3/3

Sobel Filter: 3.8X slower

1600x1200 pixel image

System	Time
Java Only	129 milliseconds
Java with Rootbeer	502 milliseconds

Details

Event	Time (ms)
Rootbeer Serialization	167
Rootbeer GPU Execution	125
Rootbeer Deserialization	210

Research with Rootbeer

- Rootbeer is open source and freely available (GNU/GPLv3)
  - http://chirrup.org/rootbeer/
  - user: hpcc_2012
  - pass: hpcc_2012
- More user-level documentation coming soon
- There is a native emulation mode
  - Can be used to arbitrarily compile any Java bytecode to x86/amd64 and launch the native code from within a Java process

Conclusions

- Rootbeer is highly tested and ready to be used by researchers who want to use the Java Programming language to program GPUs

- I plan to maintain Rootbeer until the Java Programming Language is not popular. If you find a bug and send me a testcase, I will fix it

Support

- Rootbeer is supported by the National Science Foundation

References

- [Weste93] - N. Weste and K. Eshraghian, Principles of CMOS VLSI Design - A Systems Perspective, Addison-Wesley, 1993.