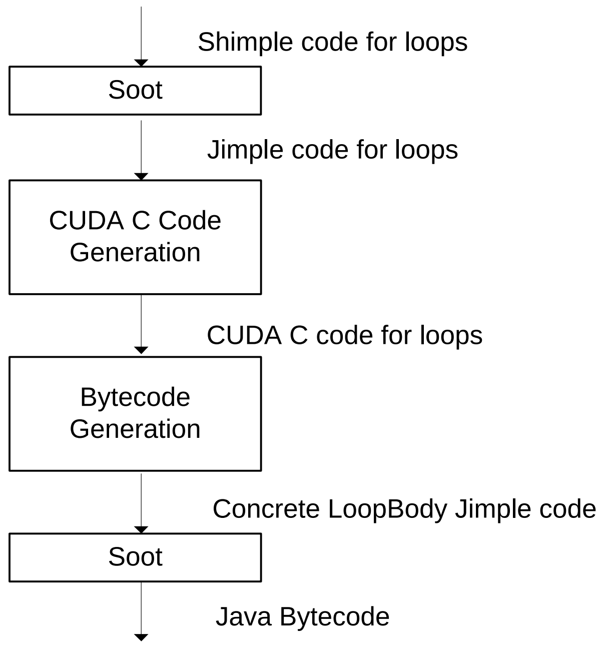
CUDA C Code Generation Transformation Flow



Appendix D. Complete Java+CUDA C Vector Addition Code

This appendix lists all the generated and transformed code for a vector addition example. The Shimple code is listed in Java for simplicity. Figure 47 is the vector addition Java Source. Figure 48 lists the transformed vector addition Java Source. Figure 49 shows a Concrete LoopBody. Figure 50 shows a Concrete GcObjectVisitor. Finally, figure 51 displays generated CUDA C code.

```java
package edu.syr.pcpratts.javaautogpu.runtime.test;

public class VectorAddExample {

  int[] x;
  int[] y;
  int[] ret;

  public void add(){
    for(int i = 0; i < x.length; ++i) {
       ret[i] = x[i]+y[i];
    }
  }
}
```

Figure 1 - Vector Addition Java Source

```java
package edu.syr.pcpratts.javaautogpu.runtime.test;

public class VectorAddExample {

  int[] x;
  int[] y;
  int[] ret;

  public void add(){
    QueueManager manager = QueueManager.v("0");
    for(int i = 0; i < x.length; ++i){
      LoopBody0 body = new LoopBody0(this, i);
      manager.enqueue(body);
    }
    manager.run();
    Iterator<LoopBody> iter = manager.iterator();
    while(iter.hasNext()){
      LoopBody0 curr_body = (LoopBody0) iter.next();
      //any inter-loop dependent code below the parallel portion
      //of the loop would be executed here. in this example there
      //is none.
    }
  }
}
```

Figure 2- Transformed Vector Addition Java Source

```java
package edu.syr.pcpratts.javaautogpu.generated;

import edu.syr.pcpratts.javaautogpu.runtime.LoopBody;
import edu.syr.pcpratts.javaautogpu.runtime.test;
import edu.syr.pcpratts.javaautpgpu.GcObjectVisitor;
import edu.syr.pcpratts.javaautogpu.runtime.Memory;
import edu.syr.pcpratts.javaautogpu.runtime.gpu.GcHeap;
```

```

public class LoopBody0 extends LoopBody {

  public VectorAddExample r0;
  public int i0_1;

  public LoopBody0(VectorAddExample r0, int i0_1){
    this.r0 = r0;
    this.i0_1 = i0_1;
  }

  public void run(){
    r0.ret = r0.x[i0_1] + r0.y[i0_1];
  }

  public String getCode(){
    StringBuilder ret = new StringBuilder();
    //note there is a maximum size of a String in a java class file
    //so the string is split up
    ret.append("<compiled CUDA C code part 1>\n");
    //...
    ret.append("<compiled CUDA C code part N>\n");
    return ret.toString();
  }

  public GcObjectVisitor getVisitor(Memory mem, GcHeap heap){
    return new LoopBody0GcObjectVisitor(mem, heap);
  }
}
```

Figure 3- Generated LoopBody0 Java Source

```
package edu.syr.pcpratts.javaautogpu.generated;

import edu.syr.pcpratts.javaautpgpu.GcObjectVisitor;
import edu.syr.pcpratts.javaautogpu.runtime.Memory;
import edu.syr.pcpratts.javaautogpu.runtime.gpu.GcHeap;

public class LoopBody0GcObjectVisitor extends GcObjectVisitor {

  public LoopBody0GcObjectVisitor(Memory mem, GcHeap heap){
    super(mem, heap);
  }

  public int doWriteToHeap(Object o, boolean write_data){
    if(o instanceof int[]){
      return GcArrayMover.write((int[]) o, write_data);
    }
    if(o instanceof VectorAddExample){
      VectorAddExample vec = (VectorAddExample) o;
      int heap_end_ptr = mHeap.getHeapEndPtr();
      mMem.writeByte(3);
      mMem.writeByte(0);
      mMem.writeByte(11);
      mMem.writeByte(0);
      mMem.writeInt(20);
      mHeap.incrementHeapEndPtr(20);
      mMem.pushAddress();
      mMem.incrementAddress(12);
      int ref_addr1 = writeToHeap(vec.ret, false);
      int ref_addr2 = writeToHeap(vec.x, true);
      int ref_addr3 = writeToHeap(vec.y, true);
      int top_address = mMem.topAddress();
      mMem.popAddress();
      mMem.writeInt(ref_addr1);
      mMem.writeInt(ref_addr2);
      mMem.writeInt(ref_addr3);
      mMem.setAddress(top_address);
      return heap_end_ptr;
    }
    if(o instanceof LoopBody0){
      LoopBody0 body = (LoopBody0) o;
      int heap_end_ptr = mHeap.getHeapEndPtr();
      mMem.writeByte(1);
      mMem.writeByte(0);
      mMem.writeByte(7);
      mMem.writeByte(0);
      mMem.writeInt(16);
      mHeap.incrementHeapEndPtr(16);
      mMem.pushAddress();
      mMem.incrementAddress(8);
      int ref_addr1 = writeToHeap(body.r0, true);
      int top_address = mMem.topAddress();
      mMem.popAddress();
      mMem.writeInt(ref_addr1);
      mMem.writeInt(body.i0_1);
      mMem.setAddress(top_address);
      return heap_end_ptr;
    }
    throw new RuntimeException("unknown type");
  }

  public Object doReadFromHeap(Object o, boolean read_data){
    int type = readType();
    if(type == 3){
      return GcArrayMover.read((int[]) o);
    }
    if(type == 11){ //VectorAddExample
      mMem.incrementAddress(3);
      byte ctor_used_on_gpu = mMem.readByte();
      if(ctor_used_on_gpu == 1){
        o = new VectorAddExample(Sentinal.instance());
      }
      VectorAddExample vec = (VectorAddExample) o;
      mMem.incrementAddress(4);
      int address_of_ret = mMem.readInt();
      mMem.pushAddress();
      mMem.setAddress(address_of_ret);
      vec.ret = readFromHeap(vec.ret);
      mMem.popAddress();
      mMem.incrementAddress(8);
      return vec;
    }
    if(type == 7){ //LoopBody0
      mMem.incrementAddress(3);
      byte ctor_used_on_gpu = mMem.readByte();
      if(ctor_used_on_gpu == 1){
        o = new LoopBody0(Sentinal.instance());
      }
      LoopBody0 body = (LoopBody0) o;
      mMem.incrementAddress(4);
      int address_of_vec = mMem.readInt();
      mMem.pushAddress();
      mMem.setAddress(address_of_vec);
      body.r0 = readFromHeap(body.r0);
      mMem.popAddress();
      mMem.incrementAddress(4);
      return body;
    }
    throw new RuntimeException("unknown type");
  }
}
```

Figure 4 - Generated LoopBody0GcObjectVisitor Java Source

```
__device__ int
edu_syr_pcpratts_gc_get_id(edu_pcpratts_gc_info gc_info){
  return blockIdx.x * blockDim.x + threadIdx.x;
}

__device__ edu_pcpratts_gc_info
edu_syr_pcpratts_gc_init(char * gc_info_space, char * to_space,
  char * from_space, char * to_handle_map, char * from_handle_map,
  int to_space_free_ptr, int space_size, int cache_assoc){

  mCacheAssoc = cache_assoc;
  *((int *) (&gc_info_space[0])) = (int) to_space;
  *((int *) (&gc_info_space[4])) = (int) from_space;
  *((int *) (&gc_info_space[8])) = (int) to_handle_map;
  *((int *) (&gc_info_space[12])) = (int) from_handle_map;
  *((int *) (&gc_info_space[16])) = to_space_free_ptr;
  *((int *) (&gc_info_space[20])) = space_size;

  __syncthreads();

  return gc_info_space;
}

__device__ char *
edu_syr_pcpratts_gc_deref(edu_pcpratts_gc_info gc_info, int handle){
  char * to_space = edu_syr_pcpratts_gc_get_to_space_address(gc_info);
  return &to_space[handle];
}

__device__ void
edu_syr_pcpratts_gc_assign(edu_pcpratts_gc_info gc_info,
  int * lhs_ptr, int rhs){
```

```
  *lhs_ptr = rhs;
}

//no cache
__device__ int
edu_syr_pcpratts_cache_get_int(edu_pcpratts_gc_info gc_info,
  int address){
  char * to_space = (char *)
edu_syr_pcpratts_gc_get_to_space_address(gc_info);
  return *((int *) &to_space[address]);
}

__device__ void edu_syr_pcpratts_javaautogpu_generated_LoopBody0_run(
edu_pcpratts_gc_info gc_info, int thisref){
  int this0 = -1;
  int r0 = -1;
  int i0_1;
  int $r2 = -1;
  int $r3 = -1;
  int $i2;
  int $r4 = -1;
  int $i3;
  int $i4;
  edu_syr_pcpratts_gc_assign(gc_info, & this0 ,  thisref );

  r0  = instance_getter_edu_syr_pcpratts_javaautogpu_generated_
        LoopBody0_r0( gc_info, this0);

  i0_1  = instance_getter_edu_syr_pcpratts_javaautogpu_generated_
          LoopBody_i0_1(gc_info, this0);

  $r2  = instance_getter_edu_syr_pcpratts_javaautogpu_runtime
         _test_VectorAddExample_ret(gc_info, r0);

  $r3  = instance_getter_edu_syr_pcpratts_javaautogpu_runtime_test
         _VectorAddExample_x(gc_info, r0);
  $i2  = int__array_get_cached(gc_info, $r3, i0_1);;

  $r4  = instance_getter_edu_syr_pcpratts_javaautogpu_runtime_test
         _VectorAddExample_y(gc_info, r0);

  $i3  = int__array_get_cached(gc_info, $r4, i0_1);;

  $i4  =  $i2   +   $i3  ;
  int__array_set(gc_info, $r2, i0_1,  $i4 );

  return;
}

__device__ void
edu_syr_pcpratts_javaautogpu_runtime_RuntimeBasicBlock_run(
  edu_pcpratts_gc_info gc_info, int thisref){

}

__device__ void
invoke_edu_syr_pcpratts_javaautogpu_generated_LoopBody0_run(
  edu_pcpratts_gc_info gc_info, int thisref){
  char * thisref_deref = edu_syr_pcpratts_gc_deref(gc_info, thisref);
  GC_OBJ_TYPE_TYPE derived_type =
    edu_syr_pcpratts_gc_get_type(thisref_deref);
  if(0){}
  else if(derived_type == 7){
    edu_syr_pcpratts_javaautogpu_generated_LoopBody0_run(gc_info,
      thisref);
  }
  else if(derived_type == 8){
    edu_syr_pcpratts_javaautogpu_runtime_LoopBody0_run(gc_info,
      thisref);
  }
}

__device__ int
instance_getter_edu_syr_pcpratts_javaautogpu_generated_LoopBody0_r0(
  edu_pcpratts_gc_info gc_info, int thisref){

  char * thisref_deref = edu_syr_pcpratts_gc_deref(gc_info, thisref);
  return *((int *) &thisref_deref[8]);
}

__device__ int
instance_getter_edu_syr_pcpratts_javaautogpu_generated_LoopBody0_i0_1(
  edu_pcpratts_gc_info gc_info, int thisref){

  return edu_syr_pcpratts_cache_get_int(gc_info, thisref+12);
}

__device__ int
instance_getter_edu_syr_pcpratts_javaautogpu_runtime_test_
  VectorAddExample_ret(edu_pcpratts_gc_info gc_info, int thisref){

  char * thisref_deref = edu_syr_pcpratts_gc_deref(gc_info, thisref);
  return *((int *) &thisref_deref[8]);
}

__device__ int
instance_getter_edu_syr_pcpratts_javaautogpu_runtime_test_
  VectorAddExample_x(edu_pcpratts_gc_info gc_info, int thisref){

  return edu_syr_pcpratts_cache_get_int(gc_info, thisref+12);
}

__device__ int
instance_getter_edu_syr_pcpratts_javaautogpu_runtime_test_
  VectorAddExample_y(edu_pcpratts_gc_info gc_info, int thisref){

  return edu_syr_pcpratts_cache_get_int(gc_info, thisref+16);
}
__device__ void
instance_setter_edu_syr_pcpratts_javaautogpu_runtime_test_
  VectorAddExample_y(edu_pcpratts_gc_info gc_info, int thisref,
  int parameter0){

  char * thisref_deref = edu_syr_pcpratts_gc_deref(gc_info, thisref);
  edu_syr_pcpratts_gc_assign(gc_info, (int *) &thisref_deref[16],
    parameter0);
}

__device__ int int__array_get(edu_pcpratts_gc_info gc_info,
  int thisref, int parameter0){
  char * thisref_deref = edu_syr_pcpratts_gc_deref(gc_info, thisref);
  return *((int *) &thisref_deref[12+(parameter0*4)]);
}

__device__ int int__array_get_cached(edu_pcpratts_gc_info gc_info,
  int thisref, int parameter0){
  return edu_syr_pcpratts_cache_get_int(gc_info,
    thisref+12+(parameter0*4));
}

__device__ void int__array_set(edu_pcpratts_gc_info gc_info,
  int thisref, int parameter0, int parameter1){
  char * thisref_deref = edu_syr_pcpratts_gc_deref(gc_info, thisref);
  *((int *) &thisref_deref[12+(parameter0*4)]) = parameter1;
}

__global__ void entry(char * gc_info_space, char * to_space,
  char * from_space, char * to_handle_map,
  char * from_handle_map, int * handles, int * to_space_free_ptr,
  int space_size, int cache_assoc, int iters){

  edu_pcpratts_gc_info gc_info =
  edu_syr_pcpratts_gc_init(gc_info_space,
    to_space, from_space, to_handle_map, from_handle_map,
    *to_space_free_ptr, space_size, cache_assoc);
  int loop_control = edu_syr_pcpratts_gc_get_id(gc_info);
  if(loop_control < iters){
    int handle = handles[loop_control];
    edu_syr_pcpratts_javaautogpu_generated_LoopBody0_run(gc_info,
      handle);
  }
  __syncthreads();
  *to_space_free_ptr =
    edu_syr_pcpratts_gc_get_to_space_free_ptr(gc_info);
  __syncthreads();
}
```

Figure 5 - Generated CUDA C code (abbreviated)