Iterative Stencil Computations in Ruby on GPUs

Matthias Springer  Peter Wauligmann  Hidehiko Masuhara
Department of Mathematical and Computing Science, Tokyo Institute of Technology

What is Ikra?
• RubyGem for array-based GPU computing
• Compiles Ruby code to C++/CUDA program
• Current focus: Iterative scientific computations
• Parallel map, reduce, stencil, new
• Data types inside parallel/host sections:
  primitive (int, float, bool, nil), array (read only), zipped, 
  object (partial support, incl. method calls), 
  union type (combination of above ones)

Design Decisions
• Modularity: Build complex programs from multiple 
  parallel sections using object-oriented programming
• Kernel Fusion: Combine parallel sections into single 
  GPU kernel, delay execution to the latest possible point
• Host Section: Avoid switching between Ruby 
  interpreter and generated C++ program

Ikra API: Example
result = Ikra.host_section do 
  arr = Array.pnew(10) do |i| i + 1 end 
  while arr.preduce(10) < 100 
    arr = arr.pmap do |i| i + 2 end 
  end 
  puts "Result is #{result.to_a}"
end

Symbolic exec. in Ruby interpreter: returns a command 
(contains all information for code generation + execution)

Code Generation
• C++ type for polymorphic expressions: union type struct
  struct union_t {
    union { int *; /* ... */ void *pointer; } data; 
    int class_id; 
  }
• Method call with polymorphic receivers: switch stmt.
• Parallel section: Data structure for command data
• Kernel launch: Generated only for run, [1], end of section
• Future work: Data sharing between multiple parallel 
  sections (avoid redundant comput.), escape analysis to 
  detect if it is safe to reuse the same memory location

Kernel Fusion in Loops via Symb. Execution

a1 = Arr.pnew(...) 
while (a2 = φ(a1, a3); a2.preduce[0] < 100) 
  a2 = a2.pmap do ... end 
end

Code Generation: High-level overview with 
kernel launches only

https://prg-titech.github.io/ikra-ruby/