

What is the SVM?

Mattan Erez

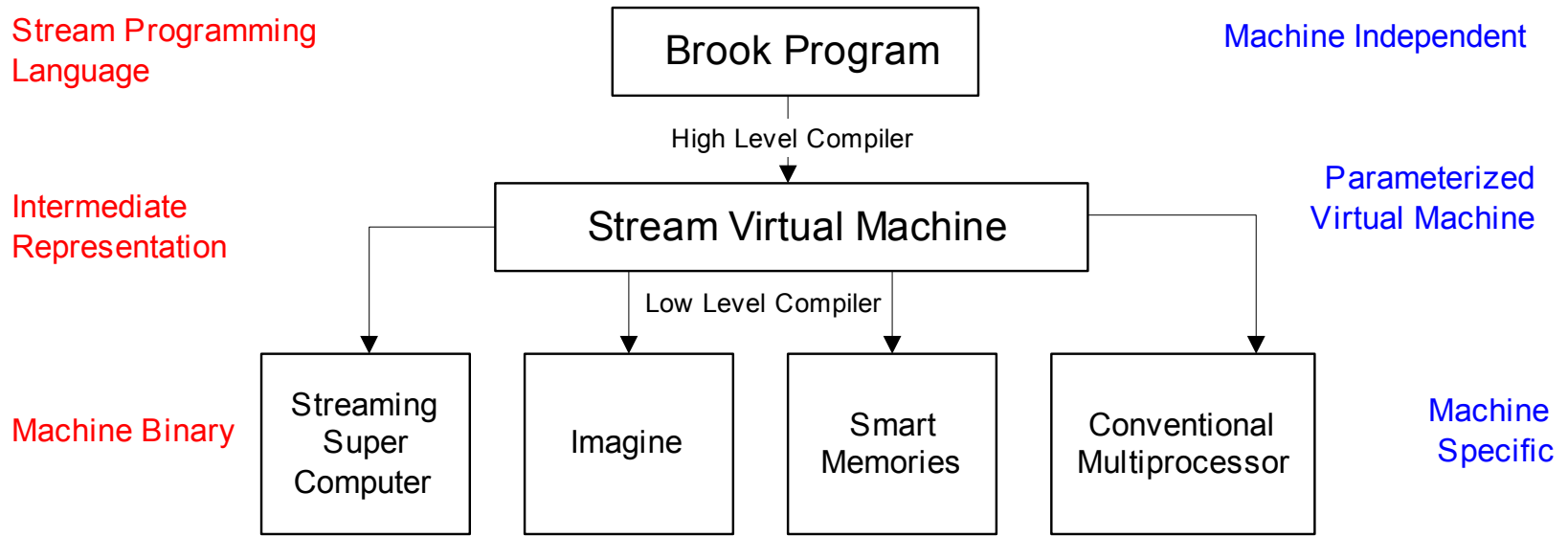
Francois Labonte

Outline of Discussion

- New Parameters
- Stream Machine Features
- Simulator
- Multinode
- What is the SVM?

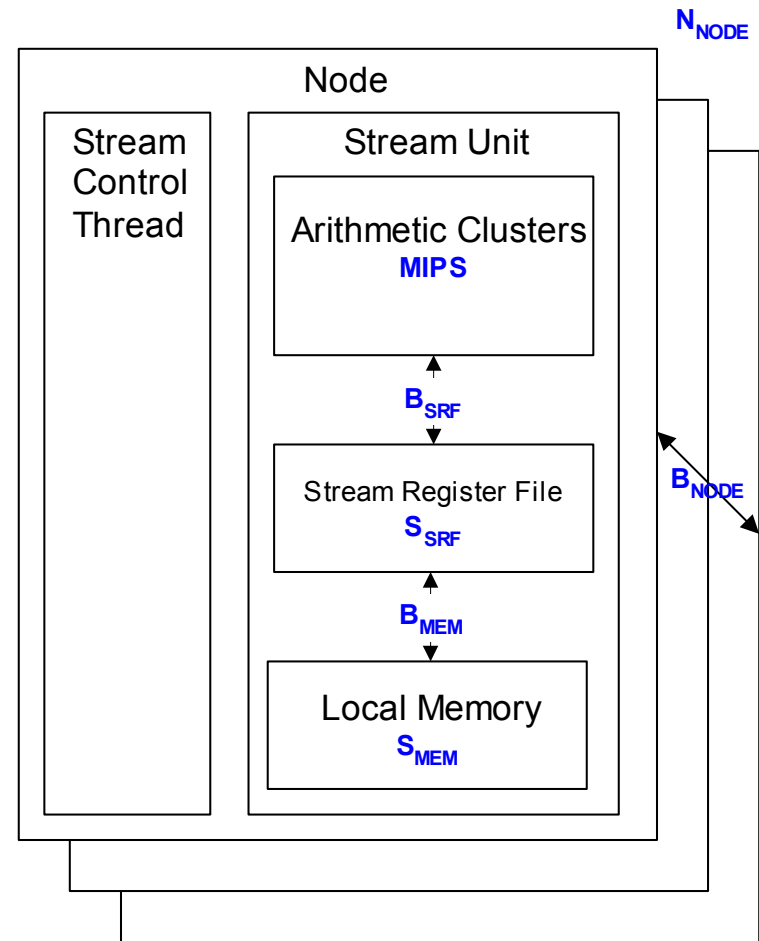
Original SVM

- Intermediate target for high level compiler to do resource allocation for various stream architectures
- Abstract and parameterize features of different stream machines



SVM Parameters

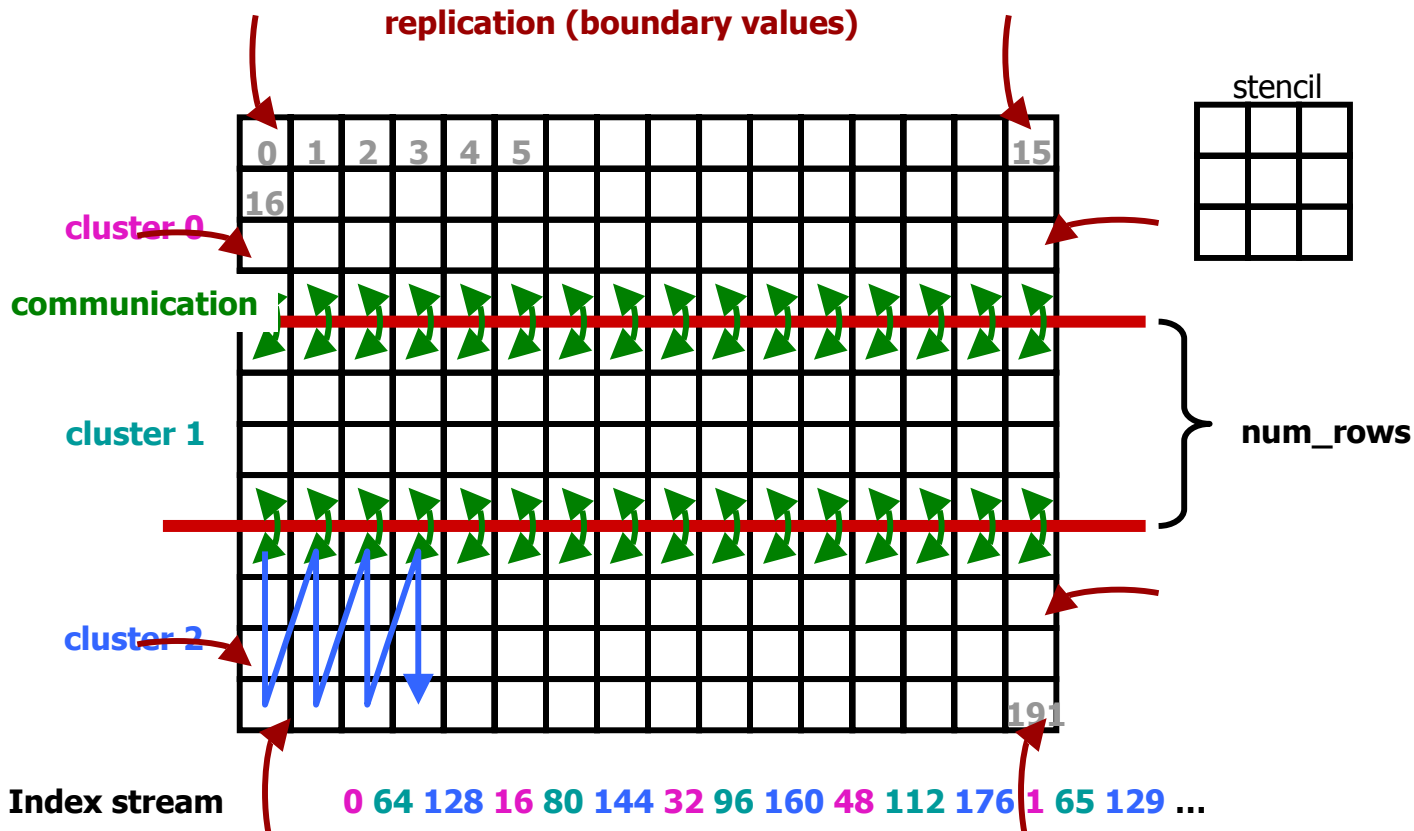
- Number of nodes
- Bandwidths between nodes, memory hierarchies
- Size of Memories – except local register files
- Computation of Arithmetic cluster (approximate, aggregate)



What was Abstracted

- Size of Local register files
 - Register allocation is the last step of kernel compilation
 - Loop unrolling (depends on number and latency of ALUs) will affect number of live registers
- Number of clusters (parallel units)
 - Communication pattern

Why We Need More



$$\text{SRF allocation} = \text{num_rows} * \text{row_length} * \text{num_clusters}$$

What Else?

- Feedback paths from lower levels
 - Failed register allocation
 - Instruction store problems
 - Unrolling
- Split/Merge kernels
- Redo stream schedule

What we propose

- Number of cluster and local registers (per cluster) are parameters to be used for such things as stencils
- Loop unrolling and software pipelining (to a less extent) will change the number of available local registers.
- Each kernel has an “Unroll Factor” which accounts for registers taken for unrolling

Stream Architecture Features

- Ex: Not all architectures have indexable SRF... How will a single compiler exploit these features?
- Possible Solutions
 1. Different Machine models
 - SSS model, GPU model,.... Different compilers?
 2. True/False Features
 - Compiler chooses methods (need to implement brook functions using different methods)
 3. Features with cost
 - Emulate some features and give them a cost, so if there is another method to implement a brook function use the cheaper one

What we would prefer

- Probably a combination of Cost model and True/False features:
 - Some feature might not be possible to emulate (spills on Imagine)

SVM Simulator

- What's the purpose:
 - Run a multinode simulation
 - Timing counters for memory transfers
 - MEM to SRF
 - NODE to NODE
 - Timing counters for execution cluster
 - Timing counter for overall execution (when execution is waiting on memory and vice-versa)

Multinode

- Memory model
 - SSS has a segmented multinode view of memory
 - Other machines will not
- Memory Consistency model
 - SSS has user controlled consistency
- Memory Coherence
 - Machines with implicit memory coherence
- Communication
 - SRF to SRF communication

Multinode – cont.

- **Scalar – Stream Interactions**
- Synchronization
 - Lock, barrier => part of features
- Programming model
 - Consistent with Brook model

What is SVM

- It was a compilation target, a clean break, which abstracted the execution
- It is an intermediate representation with parameters for everything including execution.
- This is mostly a change in how we think about it, it still serves its purpose (hopefully will)