



# Stanford Streaming Supercomputer Compiler

Bill Dally  
 Ben Chelf  
 Jayanth Gummaraju

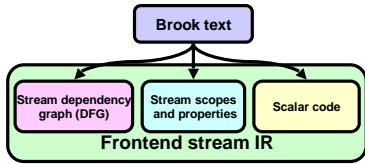
Pat Hanrahan  
 Abhishek Das  
 Ujval Kapasi

Mendel Rosenblum  
 Mattan Erez  
 Francois Labonte



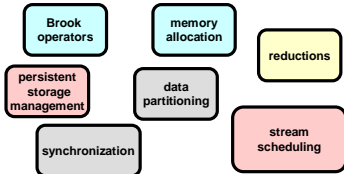
## Frontend

- Parse Brook text into machine readable form
- Extract semantic information



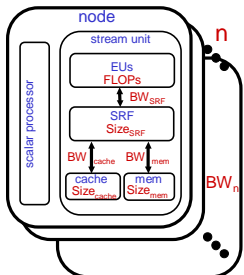
## Global Passes

- Analyze and produce high-level executables

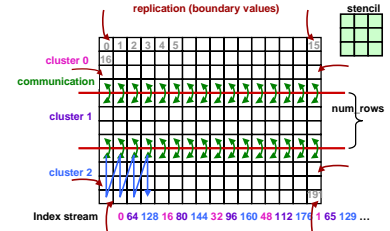


## Streaming Virtual Machine

- A parameterized model of a stream architecture
  - allows efficient high level compiler optimizations
  - common global compiler for multiple hardware platforms (SSS, Imagine, Clusters)

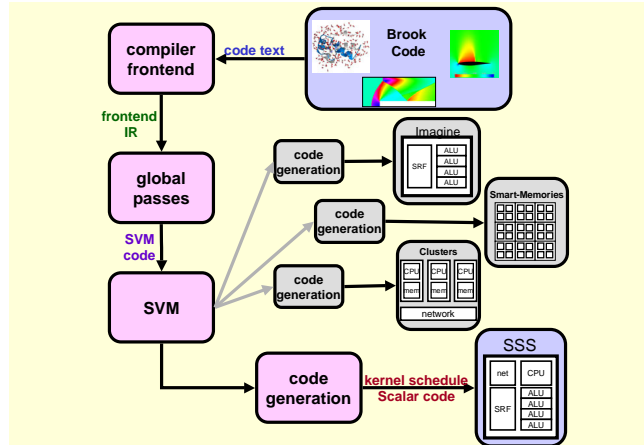


## 2D Stencil Example



## Status

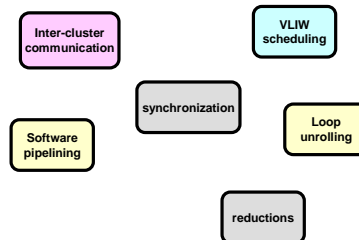
- Frontend built based on *Metacompilation*
- Simple analyses only
  - produces single-node code
  - relies on efficient implementation of Brook operators
- StreamScheduler and StreamC used for SVM
  - based on Imagine tools with support new architecture
  - produces SSS scalar code
  - produces SVM code
- ISCD and KernelIC used for code-generation
  - includes support for iterative operations
  - optimizes kernel code (SWP, unrolling, and scheduling)
  - produces SSS cluster VLIW code



The compiler relies on explicit information present in the Brook program to perform a set of global transformations and optimizations. Examples of these transformations include data partitioning, stream scheduling, and persistent storage management. The result of the global phase is *Stream Virtual Machine* code which can be targeted at different platforms using specific code generators.

## Code Generation

- Target code at specific SSS configuration
- Optimize kernels and SVM constructs
- Insert inter-cluster communication
- Finalize reductions and synchronization



## Research Directions

- |                                                                                                                                       |                                                                                                                                       |
|---------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------|
| Multi-node support                                                                                                                    | Locality enhancing transformations                                                                                                    |
| <ul style="list-style-type: none"> <li>data partitioning</li> <li>synchronization</li> <li>global data-movement scheduling</li> </ul> | <ul style="list-style-type: none"> <li>kernel partitioning</li> <li>multi-dimensional optimizations</li> <li>indexable SRF</li> </ul> |

## Evaluation

- analyze results with SSS simulator
- enhance and optimize accordingly