



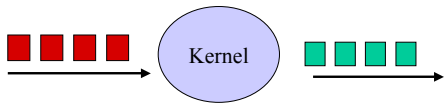
Stanford Streaming Supercomputer Software System



Bill Dally Pat Hanrahan Mendel Rosenblum
 Ian Buck Mattan Erez Massimiliano Fatica
 Jayanth Gummaraju Francois Labonte Alan Ray

Brook

- C syntax with Brook extensions
- Streams
 - Contiguous 1D view of records which can be operated on in parallel.
 - stream float floats; // Variable length stream of floats
- Kernels
 - Functions which operate only on streams
 - Arguments are read-only, write-only, or reduce (associative operations only)
 - Called on every element of the input streams
 - Limited communication between elements
 - No static variables
 - No global memory access



UPC Additions

- Initial Effort
- Same scalar code executed on all nodes
 - No mechanism for task parallelism (no UPC forall constructs, no MYTHREAD)
 - Each instance keeps a copy all non-shared memory (locals, heap memory, etc).
 - Read/Write to non-shared memory is resolved locally
 - Reads to shared memory is performed by all nodes
 - Writes to shared memory only occurs if memory resides on that node.

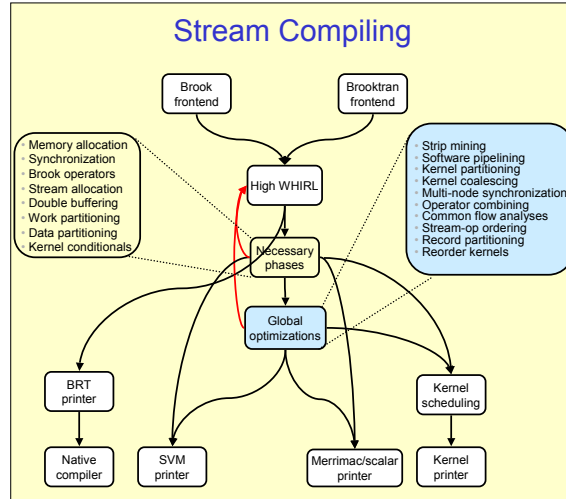
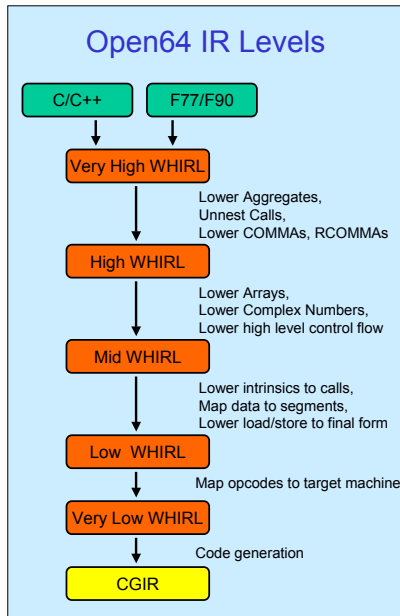
- Synchronization on Writes
 - Determined by UPC reference-type-qualifier:
 - strict shared int y;
 - Strict ordering semantics (code inserted by compiler/runtime)
 - Worst case: sync on every write
 - relaxed shared int x;
 - Unordered semantics
 - User inserts barriers/sync commands
 - upc_barrier();
 - #pragma upc strict/relaxed
 - Pragma determines default characteristics

Brooktran

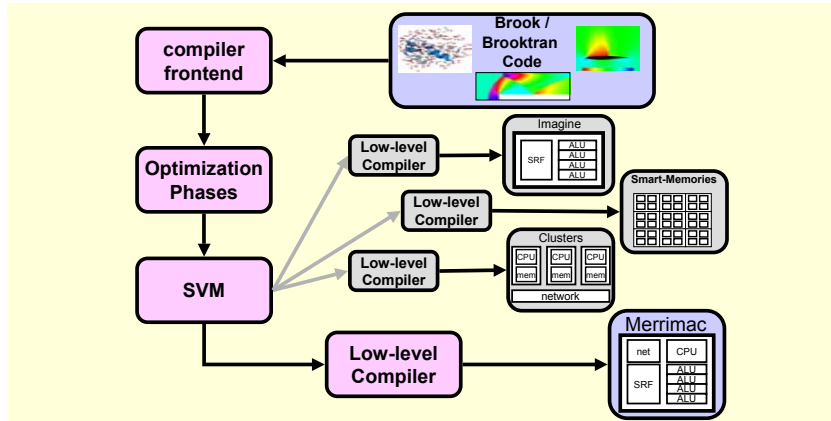
- Allow full streaming capabilities within F95
- Brooktran is a superset of F95
- Makes possible incremental porting of existing legacy codes
- F95 array syntax will be supported in streaming environment

```
stream, type(gridcell), dimension(:,:): a, b(2,2)
stream, type(real), dimension(:,:): c
.....
call streamSource(a, mesh, 2, nx+1, ny+1)
call streamStencil(b, a, STREAM_STENCIL_HALO, 2, 0, 1, 0, 1)
call ComputeMetric(b, c, volmin, volmax)
call streamSink(c, vol(1,1), nx, ny)

kernel subroutine ComputeMetric(grid, volume, volmin, volmax)
stream, intent(in), type(gridcell):: b(2,2)
stream, intent(out), type(real):: volume
real, intent(reduce):: volmin, volmax
.....
end subroutine ComputeMetric
```



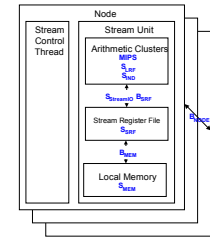
Brook and Brooktran are first compiled into a common IR (High WHIRL) by the frontends. High WHIRL is translated into BRT and then run by the native compiler. Several transformations are performed on High WHIRL that are necessary to translate into SVM code. For example, SVM code assumes that memory is allocated for all the streams. The SVM printer generates the code. Several global optimizations are performed to generate optimized SVM code and Merrimac code for multiple nodes. For example, kernels that have low arithmetic intensity are coalesced together to form a bigger kernel having higher arithmetic intensity. Finally, the optimized SVM code and Merrimac code is generated from High WHIRL. The kernels are scheduled after both the necessary phases and global optimizations.



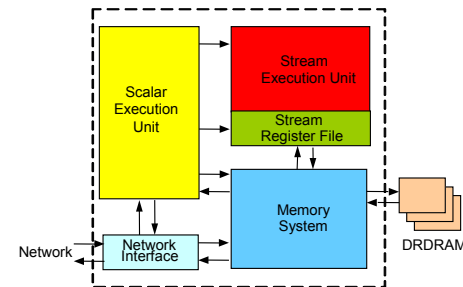
The compiler relies on explicit information present in the Brook/Brooktran program to perform a set of global transformations and optimizations. The result of the global phase is *Stream Virtual Machine* code which can be targeted at different platforms using specific low-level compilers.

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)



Merrimac



Streams are loaded from DRDRAMs or the network into the SRF (Stream Register file). The streams are then transferred to the Stream Execution Unit where a kernel operates on the stream elements, producing one or more output streams. The output streams are written back to the SRF which are then consumed by a subsequent kernel, and so on. The final results are sent to DRDRAMs and network.

Status / Future Work

- Front End Working
 - Brooktran to WHIRL
 - Brook to WHIRL
- SVM Initial Specification Complete
- Current Projects
 - SVM reference implementation
 - Brook to C reference implementation
 - Implementing necessary stream compiler phases