**Stanford Streaming Supercomputer Software System**

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

**Brook**
- C syntax with Brook extensions
- Streams: Contiguous 1D view of records which can be operated on in parallel, stream float/float, 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 of 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;`  
  - `shared int x;`  
  - `release shared int x;`  
  - `unshared semantics`  
- User inserts barriers/sync commands
- `#pragma upc strict/release
pragma upc barrier;`  
-Pragma determines default characteristics

**Brooktran**
- Allows 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

**Open64 IR Levels**

<table>
<thead>
<tr>
<th>Language</th>
<th>C/C++</th>
<th>F77/F90</th>
<th>Brook/Brooktran</th>
</tr>
</thead>
<tbody>
<tr>
<td>Very High WHIRL</td>
<td>Lower Aggregates, Unnest Calls, Lower COMMAS, RCOMMAS</td>
<td>Stream allocation, Double buffering, Work partitioning, Data partitioning, Kernel conditional</td>
<td>Streaming phases</td>
</tr>
<tr>
<td>High WHIRL</td>
<td>Lower Arrays, Lower Complex Numbers, Lower high level control flow</td>
<td>Memory allocation, Synchronization, Brook operators, Stream allocation, Data partitioning, Kernel conditional</td>
<td>Stream Compiling</td>
</tr>
<tr>
<td>Mid WHIRL</td>
<td>Lower intrinsics to calls, Map data to segments, Lower load/store to final form</td>
<td>Memory allocation, Synchronization, Brook operators, Stream allocation, Data partitioning, Kernel conditional</td>
<td>Global optimizations</td>
</tr>
<tr>
<td>Low WHIRL</td>
<td>Code generation</td>
<td>Memory allocation, Synchronization, Brook operators, Stream allocation, Data partitioning, Kernel conditional</td>
<td>Scheduling</td>
</tr>
<tr>
<td>Very Low WHIRL</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Stream Compiling**
- Brook frontend
- Brooktran frontend
- SVM Frontend
- SVM Back-end
- BRT (Brook/ Brooktran Runtime)
- Stream Compiling

**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