Brook: Summer Progress Report

Ian Buck
Mattan Erez, Peter Mattson, Eric Schwietz, Kenneth Mackenzie

Stanford University
Reservoir Labs
Brook Update

• June 19th
  – Goal: Completed Spec by August 22

• August 28th
  – Full draft finished

• October-November
  – Preliminary Reservoir non-optimizing compiler

• August 2004
  – Fully optimizing compiler
Streams

• Declarations
  – float s<>;
  – float t<200,300>;
  – float m<,300>;
  – float p<>[3][2];
  – mytype q<50>;

• Length & Shape
  – Length may be runtime dependent
  – Shape parameters must be constant expressions
    • May need to recompile code for different datasets
Streams

- **StreamRead**
  ```c
  float s<>;
  float a[3][5];
  streamRead(s, a, 0, 3, 0, 5);
  ```

- **StreamWrite**
  ```c
  streamWrite(s, a, 0, 3, 0, 5);
  ```

- **StreamReadAll**
  ```c
  streamReadAll(s, a);
  ```

- **StreamWriteAll**
  ```c
  streamWriteAll(s, a);
  ```
Brook C

- .br files containing streaming code
  - Called by .c code
- Limited Keywords
  - asm, auto, register, goto, volatile, static,
- No static storage
  - prevent visible side-effects from external code
  - globals and statics
- Heavily restricted pointers
  - permits pass by reference but not much else
- No recursion
- No precise exceptions
- Full sequential consistency
Brook C

- Brook C Function Declaration
  - Arrays are fully specified
  - mybrookfunction (float a[300][200][100])
    - May require recompiling of program for dataset
  - Streams not accessible or persist outside of .br files
Kernels

```c
void kernel foo (float a<>, out float b<>) { ... }
```

Variable Output Streams (vout)
```c
void kernel foo (... , vout[100] float c<> ) {...}
```
- “100”: max push count
- Optional but strongly encouraged
- Order preserving
Reductions

- Limited access variables
  ```c
  void kernel sum (float a<>, reduce float r) {
    r += a;
  }
  ```

- Operators
  - Native types: `+= *= |= &= ^=`
  - User defined reduction operator
    - Permits reductions on structs
    - Must be communitive, associative
  ```c
  void reduce MulFactor (float a<>, reduce float r, int factor) {
    r = r * factor * a;
  }
  ```
Stream Operators

- streamDomain
- streamGroup
- streamStencil
- streamFlatten
- streamStride
- streamReplicate
- streamRepeat
- streamCat
- streamMerge
- streamGatherOp
- streamScatterOp
Status

• Public version available September 15th
• Missing Stuff
  – Asynchronous operations
  – Structure filtering
  – Exception model
  – User defined ScatterOp/GatherOp
  – #parameter specifiers