



Brook

A Streaming Programming Environment

Ian Buck, Mattan Erez, Ben Chelf, Pat Hanrahan



Streams

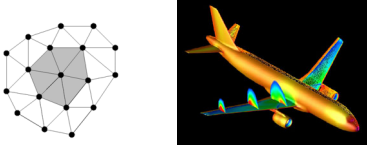
- Contiguous 1D view of records in memory
- Operated on in parallel
- Declarations:
 - stream float floats;
- Accessing stream values not permitted outside of kernels



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

Reference Streams



values:

references:

streamScatterOp (values, references, FLOAT_ADD)

memory:

Operations supported: addition, subtraction, multiplication, reciprocal, maximum, minimum, logical or, logical and.

Current Status

- Over 15,000 lines of Brook code
- 8 application designers
- Compiler support on Win32, Linux, MacOS X

Streaming Languages

- Compilation
- System Architecture
- Node Architecture
- StreamFlo
- StreamMD
- StreamFEM

Why Stream Programming

- High arithmetic intensity
- Hide memory latency
- Efficient use of VLSI resources
- Exploit parallelism

```

struct Vector { float x, y, z };
typedef struct Vector VectorT ;

kernel void UpdatePosition (
    stream VectorT pos,
    stream VectorT vel,
    float timestep,
    out stream VectorT new_pos) {
    new_pos.x = pos.x + timestep * vel.x ;
    new_pos.y = pos.y + timestep * vel.y ;
    new_pos.z = pos.z + timestep * vel.z ;
}

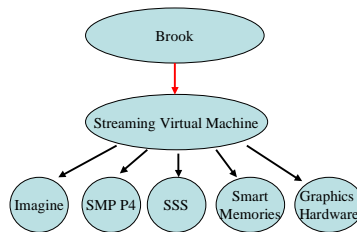
void main () {
    VectorT Pos[1024] = {...};
    VectorT Vel[1024] = {...};

    stream PositionT sPos, sVel, sNewPos;
    streamLoad (sPos, Pos, 1024);
    streamLoad (sVel, Vel, 1024);

    UpdatePosition (sPos, sVel, 0.2f, sNewPos);

    streamStore (sNewPos, Pos);
}
  
```

General Streaming Language



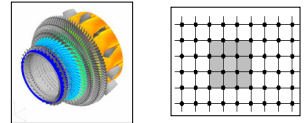
Stream Operators

- | | | |
|-----------|------------|-------------|
| Load | Group | Repeat |
| Store | Stencil | Replicate |
| Shape | StencilExt | Stride |
| GetLength | Group | SelfProduct |
| SetLength | GroupExt | Product |
| Domain | Flatten | MakeRef |
| Merge | | Gather |
| | | GatherOp |
| | | Scatter |
| | | ScatterOp |

Design Goals

- Encourage programmer write parallel code with high arithmetic intensity
 - Place restrictions on kernels and streams
- General purpose streaming language
 - No domain specific operators
- Easy to program and understand code
 - Minor extensions to C

Multidimensional Streaming



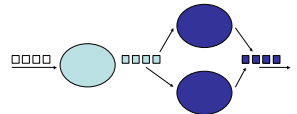
a: streamShape (a, 2, 3, 3)

a:

streamStencil (b, a, 2, HALO, 0, 1, 0, 1)

b:

Data & Task Parallelism



- Parallel execution hidden from programmer
- Task level parallelism
 - Multiple kernels executing at once
- Data level parallelism
 - Multiple stream elements processed simultaneously

Research Directions

- Multinode issues
- Memory synchronization primitives
- Blocking primitives