TIMO SCHNEIDER <TIMOS@INF.ETHZ.CH>

Master Thesis: Distributed Memory Support for Dataflow Programming
Thesis Topic

• We developed a graph-based representation for parallel programs
• We currently look to multicore CPUs, GPUs, and FPGAs
• Your task would be: integrate support for distributed memory machines
• Idea is to use MPI RMA

• You already know about parallel programming, MPI, etc.
• You will work with a team of PhD students and post-docs
• You will work on a large code-base
The Big Picture

Domain Scientist

Problem Formulation
\[ \frac{\partial u}{\partial t} - \alpha \nabla^2 u = 0 \]

\[ @\textit{spcl} \text{.\texttt{map}} \]
\[ \text{def} \ \textit{hoarz}(z; \{0:0\}, \ y; \{0:0\}, \ x; \{0:0\}) \]
\[ a \ <\ \textit{A}(z,y,x) \]
\[ b \ >>\ \textit{B}(z,y,x) \]
\[ ... \]

Explicit Dataflow
Implicit Dataflow
High-Level Program

Performance Engineer

Data-Centric Intermediate Representation (SDFG, §3)

Graph Transformation

Interactive Optimization Environment (DIODE, §4)

System

Hardware Information
SDFG Compiler

CPU Binary
GPU Binary
FPGA Modules

Thin Runtime Infrastructure
DIODE: The Future of Performance Tuning?
DIODE: The Future of Performance Tuning?
# Stateful Dataflow Multigraphs (SDFGs)

<table>
<thead>
<tr>
<th>Primitive</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Turing-Complete Construction</strong></td>
<td></td>
</tr>
<tr>
<td>s0 &gt; iter &lt; N &gt; s1</td>
<td>State: State machine element.</td>
</tr>
<tr>
<td>Data</td>
<td>N-dimensional array container.</td>
</tr>
<tr>
<td>Transient Data</td>
<td></td>
</tr>
<tr>
<td>Tasklet</td>
<td>Fine-grained computational block.</td>
</tr>
</tbody>
</table>

**Dataflow and Concurrency**

| Stream | Streaming data container. |
| Map: [i=0:M, j=0:N] | Parametric graph abstraction for parallelism. |
| C[i,j] (CR: Sum) | Conflict Resolution: Defines behavior during conflicting writes. |

**Nesting and Subgraph Aliases**

| Invoke | Call a nested SDFG. |
| Reduce: [p=0], id: 0 | Reduction of one or more memlet axes. |
| Consume: [p=0], s0, [p=0], s1 | Dynamic mapping of computations on stream elements. |
Optimizing SDFGs via Graph Transformations

Video of DIODE in action (tuning MM): https://vimeo.com/301317247
So does this really work?
So does this really work?
What you need to do

• Data-nodes need to be annotated with a storage location
• When reading from non-local memory, generate MPI_Get()
• Add synchronization in the right place
• Collective “detection”
• Look at lots of example SDFGs and the generated code

• Details are flexible of course, if interested just contact us!

timos@inf.ethz.ch