AML: Building Blocks for Explicit Memory Management

Swann Perarnau

Argonne National Laboratory

Overview

AML

C99 BSD3 Library for explicit memory management.

Overarching goal:
Abstract and Expose Data Management in Applications

Few core abstractions, dedicated to explicit management:
-> the application dictates the control flow of data operations.

1. Topology performance info (work in progress)
2. Data Layout
3. Work Decomposition
4. Movement/Transform
5. Resource Usage (work in progress)

Give users the tools to explore their performance design space

AML Itself

Abstractions (1)

Areas: how to allocate in virtual address space

- Think mmap/munmap + mbind
- Also works on CUDA
- Can be tuned for specific needs (interleave blocks of pages)

Layouts: how to dereference a buffer (shape of it)

- Contiguous or not
- Strides, pitches
- Padding, reshapes

Abstractions (2)

Tilings: how to decompose a buffer

- Slicing a layout into pieces
- Indexing tiles
- Padding, border tiles

DMAs: how to move data between memories

- From layout to layout
- Synchronous or not
- Offloaded to specific engine
- Transforms

Tutorial

Online here: **https://argo-aml.readthedocs.io/en/staging/**

Feel free to ask questions, suggest improvements, reports issues here:
https://xgitlab.cels.anl.gov/argo/aml

Work In Progress

2020: Topology queries, performance-based iterators over topology

2021: Tiling iterators, resource tracking over DMAs