Exploring the Benefits of SEJIT Specialization for Software-Managed, Reconfigurable Memory Hierarchies

Henry Cook

Overview

SW-Reconfigurable HW
- Provides:
  - Energy efficiency
  - Performance predictability
  - Adaptability
- Needs:
  - Management provided by software routines

SEJIT Specialization
- Provides:
  - Optimizations based on productive abstractions
  - Autotuning of higher-order library routines
- Needs:
  - HW with predictable performance

Applications

SVM training
- Speaker diarization
- Bloodflow simulation
- Real-time audio proc

Pattern Compositions

Maps within maps
- Iterative maps and graph traversals
- Stencils and nonlinear solves
- SIMD within tasks

Specializations

Closures, Closed over objects
- Dimensions of features and mixtures
- Required propagation of flow data
- Deadlines, priorities of tasks

Specializations

Mapping of objects to memories
- Selection of algorithm variant
- Comm avoiding strategy
- Assignment of resources to task types

ELL/Runtime Implementations

_shared_vars, copies to them
- Templated CUDA kernels
- Data layout and solve parameters
- Interaction with QoS framework

Virtual Memory Hierarchies

- Last level cache distributed in banks
- Banks have non-uniform access time over on-chip network
- How should data be placed in such a structure?
- Many possible strategies for migration and replication

Virtualization

- Define a hierarchy of multiple virtual caches stored in same NUCA cache
- Each has own index hash
- Mig/rep equivalent to eviction/fill policies on VMH

Private Virtual Local Stores

- Configurability
  - Allows local store semantics on-demand for individual routines
  - Way partitioning for easy reconfiguration

- Predictability
  - No unexpected hardware actions under the covers
  - Stable target for overlapping comm and computation
  - Direct mapped, unique virtual addresses

On-chip Memory Management in Copperhead

Exploiting closures
- Data parallel maps implicitly identify objects shared by all instances
- Can be identified as “closed over”
- Can be identified from examining composition of maps

Speaker diarization

Bloodflow simulation

Real-time audio proc

NUCA Caches

Virtual Memory Hierarchies

Future Work

- More intelligent code variant selection mechanism, using ML
- Identification components reused by all specializers
- Composability of structural patterns: depends on communication between them as staged through the memory hierarchy
- Definition of HW feature set most beneficial to specializer writers
- Quantification of energy savings of specialized code, simpler HW