Cross-Layer Driven Computer Architecture Optimizations

Per Stenström

Chalmers University of Technology
Göteborg, Sweden
Viking Yale

...“Ö”...
(...Island ahead...)
Columbus and Huffman (unfairly) got Credit for Viking Discoveries

Source of inspiration: Yale Patt
We know that Vikings were on Mars
But Vikings Landed on the Moon before Americans did too
Yale’s Transformation Hierarchy

- Problem
- Algorithm
- Program
- ISA
- Microarchitecture
- Circuits
- Electrons
Scaling (as we know it) is ending soon…

A radical new way of how we think about compute efficiency is needed

Yale 80 Celebration – Per Stenstrom © 2019
Per’s Transformation Hierarchy

Parallel programming model plus semantic information

ISA + primitives

Resource Management

Microarchitecture
Task-based Dataflow Prog. Models

- Programmer annotations for task dependences
- Annotations used by runtime system for scheduling
- Dataflow task graph constructed dynamically

**Convey semantic information to runtime for efficient scheduling**
Runtime Management of Cache Hierarchies

**View:** Runtime is part of the chip and responsible for management of the cache hierarchy
- in analogy with OS managing virtual memory

**Runtime-assisted cache management:**
- Runtime-assisted cache coherence optimizations
- Runtime-assisted dead-block management
- Runtime-assisted global cache management
Runtime-Assisted Coherence Management [IPDPS 2014]

Dependency annotations allow for optimizations with high accuracy (like in message passing)

Bulk data transfer

Prefetching

Migratory sharing optimization
Impact on Memory Stall Time

**Cholesky**

**Matmul**

**SparseLU**

**Qr**

Cross-layer coherence management can yield significant gains
Runtime-Assisted Dead-Block Management – Motivation

• Most Last-Level-Cache blocks are dead consuming precious cache space

• State-of-the-art schemes: prediction based on past behavior

• RADAR’s approach: Semantic information + prediction
Overview of RADAR [HPCA 2017]

Three schemes:

- **Look Ahead** (LA): Use data-dependence graph for prediction
- **Look Back** (LB): Use past region access statistics
- **Combined**: LA ∩ LB and LA∪LB

Evict (demote to LRU) region from LLC

Specify Task and its Region Accesses

Is Region dead after current access?
Memory bound apps provide significant gains
Outline

Background

Runtime-Assisted Cache Management

Runtime-Assisted Power Management

Concluding Remarks
QoS-driven Resource Management – Background [IPDPS 2019]

By using QoS targets, we can throttle processor resources to minimize energy consumption.

Goal: Trade off resources to minimize energy consumption while meeting QoS targets.
Overview

Every monitoring interval (on each core)

Phase Change

(Remember Management Algorithm)

RMA

Configure

SW HW

Cache Miss Profile

(IPC)

Core1 (App1)

Core2 (App2)

Partitioning Bitmask

Partitioning Bitmask

DVFS

Perf. Counters

DVFS

Perf. Counters

Last Level Cache

MeCCA

erc

CHALMERS

Yale 80 Celebration – Per Stenstrom © 2019
Searching the Configuration Space

RMA

Global Optimization Algorithm

ATD

Phase Change

Misses

Perf. Model

QoS Function

Minimum Frequency

Energy Model

EPI

MeCCa ERC

CHALMERS

Yale 80 Celebration – Per Stenstrom © 2019
Energy Saving Results (Relaxed QoS)

- Partitioning (Perfect Model)
- Partitioning (Realistic Model + OH)
- Combined (Perfect Model)
- Combined (Realistic Model + OH)

Save up to 28% of energy with 30% reduced IPS target (AVG: 20%)
Outline

1. Background
2. Runtime-Assisted Cache Management
3. Runtime-Assisted Power Management
4. Concluding Remarks
Concluding Remarks

Programming model plus semantic information
ISA + primitives
Resource Management
Microarchitecture

MECCA team members: Alexandra Angerd, Mohammad Waqar Azhar, Nadja Holtryd, Madhavan Manivannan, Mehrzad Nejat, Vasileios Papaefstathiou, Risat Pathan, Miquel Pericàs and Petros Voudouris