Efficient Execution of Microscopy Image Analysis on CPU, GPU, and MIC Equipped Cluster Systems

Guilherme Andrade\textsuperscript{1}  George Teodoro\textsuperscript{3} Leonardo Rocha\textsuperscript{2}
Joel H. Saltz\textsuperscript{4} Tahsin Kurc\textsuperscript{4} Renato Ferreira\textsuperscript{1}

\textsuperscript{1}Federal University of Minas Gerais
\textsuperscript{2}Federal University of São João del Rei
\textsuperscript{3}University of Brasília
\textsuperscript{4}Stony Brook University
**Motivation**

**Data Science**

- Extracting meaningful knowledge from large datasets collected
- Applications consist of several stages of data extraction and transformation
  - Large variety of complex algorithms
  - Large volumes of the raw data
- **High demand for computing power**

**Microscopy imaging studies:**

- Pathology image analysis used to investigate brain cancer morphology
- Several executions of the processing are necessary
- **Significant computing power need to harvested**
Motivation

Computer Systems Architecture

- Utilization of large clusters
  - High-end processors associated with compute-intensive,
  - Massively parallel co-processors (GPUs, Intel Xeon Phi.)
- Massively parallel architectures privileging ALU operations over I/O and control flow operations.

Boost to computer system research

- Frameworks providing high level programming abstractions
- Allow efficient execution on current hardware
- Extreme DataCutter
  - Applications are represented by hierarchical dataflows and
  - Each stage of the application may be replicated and assigned for computation in several nodes of a hybrid distributed memory machine
Goals

Proposal

- Explore the cooperative execution of a pathology image analysis application
- Hybrid systems equipped with CPUs, GPUs, and MICs.
- Propose, implement and evaluate performance aware scheduling techniques
- Make the pathology image analysis application computational feasible

Outline

- Motivating Application
- Extreme DataCutter
- Cooperative Execution on Hybrid Systems
- Experimental Evaluation
Motivating Application

Characteristics

- Find the classification of brain tumors
- Analysis of high resolution whole tissue slide images (WSIs) (*Whole Tissue Slide Images* - WSI)

The basic workflow in our application:

1. preprocessing as color normalization
2. segmentation of cells and nuclei (objects)
3. extraction of a set of shape and texture features per object
4. classification based on object features.
Motivating Application

Execution Tasks Profile

- **Comparative Performance Analysis of Intel Xeon Phi, GPU, and CPU: A Case Study from Microscopy Image Analysis** IPDPS 2014

- Implemented and evaluated the performance of these operations on CPUs, GPUs, and on the Intel Xeon Phi (MIC)

- The operations were implemented on all devices using the same parallelization strategy and level of optimizations.

![Graph showing relative speedup for different tasks between GPU and MIC](image-url)
Extreme DataCuter (EDC)

Characteristics

- Runtime system for heterogeneous environment supports dataflow applications.
- Application stage implemented as another dataflow of operations.
- Coarse-grain stages assigned to nodes and the fine-grain stage to Processors.
- In each node: A task dependency resolution scheme and task scheduling
- Manager-Worker model
Extreme DataCuter

Characteristics

- Worker uses all processors units
- Multiple stage instances may be assigned to the same Worker concurrently
- The maximum number of Coarse-grain stages: Worker Request Window Size
- The WRM creates a thread for each process unit (Demand-driven)
- The device manager threads notify the WRM once they become available, and the WRM scheduler selects one of the tasks with dependencies resolved for execution.
Cooperative Execution on Hybrid Systems

Providing efficient strategies

- These facts suggest that different processors are more efficient for particular types of operations
  1. there exists a high variability on the speedups achieved by the same processor as different operations are considered
  2. the relative performance among processors (CPU, GPU, and MIC) varies according to the operation executed.

- Efficient scheduling for cooperative execution on hybrid systems should take into account these performance variabilities to maximize the aggregate computing power of a heterogeneous machine.

Proposal

1. Performance-Aware Dequeue Adapted Scheduler (PADAS)
2. Performance-Aware Multiqueue Scheduler (PAMS)
Performance-Aware Dequeue Adapted Scheduler

PADAS

Considerations
- Each processing unit retrieves the task with greater speedup for its architecture
- CPU receives tasks that have low performance in accelerators

PADAS
- Global Queue
- Compare speedups (GPU e MIC) and inserts decreasingly
Performance-Aware Dequeue Adapted Scheduler

PAMS

- Multiple queues
- Inserts task copies in all queues
- Descending sorting by speedup
- $1/\max(GPUSp, MICSp)$.

Considerações

- Each processing unit retrieves the task with greater speedup for its architecture
- CPU receives tasks that have low performance in accelerators
- Task copies are deleted
Evaluation Description

Approach

- Hybrid System: 1 GPU, 1 MIC, 14 CPU cores
- 800 slices of the input image, which generate 10,800 fine-grain operations to be executed
Performance of the schedulers

Considerations

- Windows size varied from 16 to present no variation
- Windows size: impact on the decision space
- PAMS better results ($1.16 \times$ ao HEFT)
Understanding the Behavior of schedulers
Understanding the Behavior of schedulers

(a) FCFS

(b) PADAS

(c) HEFT

(d) PAMS
Considerations

- CPU-MIC e CPU-GPU: PADAS e PAMS same behavior
- GPU always impacts the performance increase: CPU-GPU is $1.27 \times$ faster than CPU-MIC

![Graph showing time comparison between different processor configurations and scheduling algorithms](image-url)
Sensibility to Inaccurate Estimations

Considerations

- Variation of error incrementing or decrementing the time.
- Speedup: little impacted with errors (PAMS e PADAS)
- Time: Significant impact (HEFT)
- PAMS 0% is just $1.07 \times$ faster than PAMS 100%
Scalability Results

**Considerações**

- 6,379 4K×4K image tiles as input
- PAMS are about 2.2× faster than the CPU-only executions
- PAMS is nearly 1.2× faster than HEFT

![Graph showing scalability results](image-url)
Conclusions and Future Directions

- PAMS scheduler achieved shorter execution times than the other polices, being 1.16× faster than HEFT.
- Our proposed performance-aware strategies, PADAS and PAMS, are less sensitive to inaccuracy in performance estimation data.
- The application scales well with EDC and the best cooperative executions, using PAMS, are about 2.2× faster than the CPU-only executions.

Future Directions

- Improve our proposed performance-aware PAMS and PADAS taking to further enhance the efficiency of pathology image analysis application.
- Evaluate PAMS and PADAS in others dataflow applications.
Questions?