LIBRA: AFFINITY-AWARE WORK-STEALING FOR INTEGRATED GPU PROCESSORS
Naila Farooqui, Rajkishore Barik, Brian T. Lewis, Tatiana Shpeisman, Karsten Schwan
Georgia Tech, *Intel Labs

PROBLEM STATEMENT

- SVM support on today’s integrated GPU processors makes true CPU-GPU work-stealing possible, but effective work-stealing is challenging:
  - **Application**: Large performance gap between CPU vs GPU, based on runtime behavior, impacts tail-end execution
  - **Device**: CPUs and GPUs have different ‘costs’ of stealing

![Graph showing CPU vs GPU performance](image)

![Graph showing work stolen](image)

PROPOSED SOLUTION

- Augment classical work-stealing with:
  - Lightweight online profiling to incorporate device affinity based on application runtime behavior
  - Hierarchical stealing to incorporate architectural differences between CPU and GPU stealing costs

**LIBRA**: Compiler and Runtime Support for Affinity-Aware Work-Stealing

- **C++ Application**
- **LLVM-to-OpenCL JIT Compiler**
  - Dynamically inserts work-stealing code into OpenCL kernels

EXPERIMENTAL RESULTS AND NEXT STEPS

- Affinity-aware work-stealing outperforms shared-queue, online profiling, and classical work-stealing approaches

![Graph showing performance comparison](image)

Effective CPU-GPU work-stealing must consider:
- **Device architectural** characteristics
- **Application runtime behavior**

Future work will investigate energy-aware heuristics to improve both performance and energy

*Intel Core-M 5Y71 Broadwell Processor*