Performance Comparison between OpenCL and OpenMP on Multi-core CPUs

Jie Shen, Jianbin Fang, Ana Lucia Varbanescu, and Henk Sips

1. Programming on Multi-core CPUs

OpenMP
• Coarse-grained
• Traditional CPU programming model

OpenCL
• Fine-grained
• From GPGPU world
• Cross-platform portability

Optimize the math!
Using fast-math compiler option works!
Make sure accuracy is still OK!

Vectorize your code!
Replacing H2D and D2H with zero-copy works!
Row-major is CPU friendly, column-major is NOT. Transposing OR undoing the transposition works!

Check if fine(-grain) works fine!
Fine grain parallelism doesn’t work for all apps ...
... but increasing workload granularity might fix it!

2. Diverse Performance Gaps

Be careful with data-dependent branches!

Use zero-copies!
Replacing H2D and D2H with zero-copy works!

3. Performance Gaps DO EXIST!

Why do they appear?
• Incorrect use of OpenCL on multi-core CPUs
• Inherent OpenCL fine-grained parallelism
• OpenCL compilers are not fully mature

How to fix them?
• Remove GPU-like programming style
• Tune the parallelism granularity
• Optimize the (math) operations

... but they can be fixed with a few simple rules!