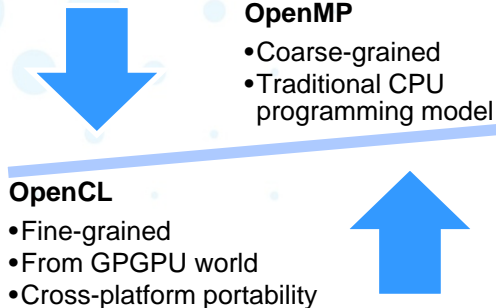


Performance Comparison between OpenCL and OpenMP on Multi-core CPUs

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

j.shen@tudelft.nl

1. Programming on Multi-core CPUs



Optimize the math !

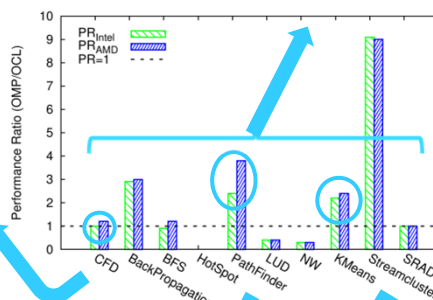
Using fast-math compiler option works!
Make sure accuracy is still OK!

Vectorize your code !

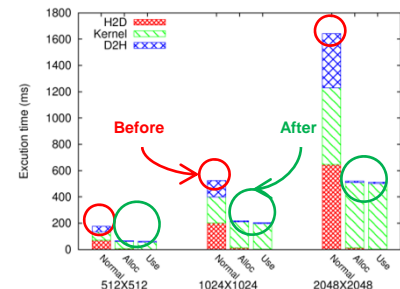
Performance Ratio	Aircraft wing (200K) vs. Missile (193K)
OpenCL	1.89
OpenMP	1.38

Be careful with data-dependent branches!

2. Diverse Performance Gaps

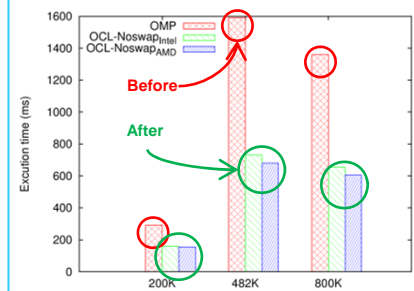


Use zero-copies !



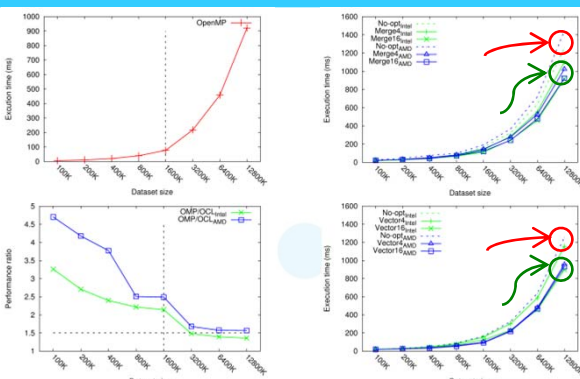
Replacing H2D and D2H with zero-copy works!

Figure out your memory access patterns !



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!

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 !

