

Context-aware optimisation and run-time adaptation of sequential libraries for multi-core systems

Anton Lokhmotov¹ Grigori Fursin² Paul H.J. Kelly¹

¹Imperial College London

²INRIA Saclay

Adaptive Compilation Cluster Meeting
3 June 2008

Using sequential libraries on multi-core systems

- Many programs use carefully optimised libraries such as ATLAS, FFTW, Intel Math Kernel Library, AMD Core Math Library, *etc.*
- Some libraries switch at runtime between multiple versions of functions depending on input data features (typically only on the size of input data)
- System and program behaviour at invocation time is usually ignored
 - Not a big deal on single-threaded systems: context-switching is relatively infrequent (*e.g.* 10ms for a typical Linux timeslice, 1us for a cache refill)
 - Important when multiple threads compete for shared resources

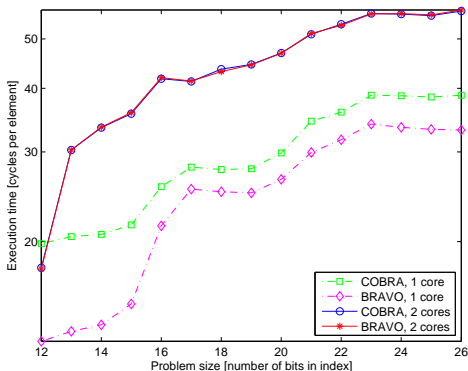
Using sequential libraries on multi-core systems

- Many programs use carefully optimised libraries such as ATLAS, FFTW, Intel Math Kernel Library, AMD Core Math Library, *etc.*
- Some libraries switch at runtime between multiple versions of functions depending on input data features (typically only on the size of input data)
- System and program behaviour at invocation time is usually ignored
 - Not a big deal on single-threaded systems: context-switching is relatively infrequent (*e.g.* 10ms for a typical Linux timeslice, 1us for a cache refill)
 - Important when multiple threads compete for shared resources

- Caches (important even for simultaneous multithreading)
- Memory bandwidth (see an example on next slide)
- ...

Bit-reversal on dual-core 2.5GHz PowerPC 970FX

Easy to parallelise but only makes the matters worse!



- COBRA: Carter/Gatlin (FOCS'98, HPCA'99)
- BRAVO: Lokhmotov/Mycroft (SPAA'07)

Context includes:

- System behaviour
 - workload
 - resource contention
- Program behaviour
 - performance (hardware counters?)
 - program phases
- Dynamic input features
 - size
 - patterns

Questions

- How to characterise and cluster different contexts?
- How to decide which library functions to make versions of (and in which way), to cover context clusters?
- Which context features are important for effectively predicting which function version to use at run-time with low-overhead?
- How to modify current iterative compilation environments to systematically explore library optimisations for different contexts?

- Iterative compilation of Imperial's 'active' libraries
- Learning how to adapt to different contexts. . .