BLIS-like Library Instantiation Subprograms (BLIS)

The Basic Linear Algebra Subprograms (BLAS) are a set of low-level subroutines that perform common linear algebra operations. BLIS is a software framework for instantiating high-performance BLAS-like dense linear algebra libraries. BLIS [1] was chosen over GoTOBLAS, ATLAS, etc., due to its portable micro-kernel architecture and active user base.

BLIS features:
- ISO C99 code with flexible BSD license.
- Support for BLAS APIs calling conventions.
- Competitive performance [2].
- Multi-core friendly.
- Multi-layer API and code identifying and isolating a key set of computational kernels.
- Modularity and extensiveness.
- Portability (x86, x64, TIC66x, PowerPC, etc.) that doesn’t impede high performance [3].
- Foundation for mixed precision (experimental).

Level-3 BLAS using BLIS on Myriad

BLIS defines three Level-3 micro-kernels. Implementation of the fused GEMM-TRSM kernel is optional.

GEMM and TRSM operations

The xGEMM and xTRSM routines are the typical benchmarks of the Level-3 BLAS performance of an implementation. Basic information on the operations and the computational complexities of these two routines are presented below.

Mapping of matrix blocks on CMX (SGEMM)

The BLIS ISO C99 code allowed straightforward compilation for Myriad. Following optimizations consisted of:
- micro-kernel implementation in SHAVE assembler,
- and memory management/allocation.

Memory focused optimizations

- Double/triple buffering of arguments.
- Buffers shared by all SHAVES.
- Data passing using pointer arithmetic.
- Overlapped DMA accesses.

SGEMM and STRSM performance

Results

Conclusions

Acknowledgement

The work funded by the SHAVE Team and Movidius Ltd.

The Myriad media-processor SoCs

Myriad architecture prioritises power-efficient operation and area efficiency. In order to guarantee sustained high performance and minimise power the proprietary SHAVE (Streaming Hybrid Architecture Vector Engine) processor was developed. Data and instructions reside in a shared Connection Matrix (CMX) memory block shared by all Shave processors. Data is moved between peripherals, processors and memory via a bank of software-controlled DMA engines.

Myriad 1 architecture highlights [4]

- 66nm ultra-low power architecture
- Hardware support for SIMD, matrix transpose, sparse data, sqrl@fp16, predicated execution...
- Heterogeneous SoC: 1 Leon3@fp64 + 16 Shaves@fp32.
- 32KB LRAM, 1MB CMX, 16/64MB DDR, DMAs.
- Power efficiency of 1TOPS/W (max 8-bit equivalent).

Myriad 2 architecture highlights [4]

- 28nm ultra-low power (< 0.5W/600MHz) with 17 power islands.
- Extended hardware support over Myriad 1: clock-gating, hard-wired configurable accelerators for imaging and vision, etc.
- Heterogeneous SoC: 2 Leon4@fp64 + 12 Shaves@fp32.
- 256+32KB LRAM, 2MB CMX, DDR3 support, DMAs.
- Power efficiency of 2TOPs/W (max 16-bit equivalent).

Acknowledgement

This work has been carried out under the HiPEAC PhD industrial internship programme and the Movidius Ltd. project (www.movidius.com) while being co-funded by Movidius and the European Union under the grant 609720 (HiPEAC) and 611183 (HiPEAC2).