Exploring the capabilities of multi-core DSPs for dense linear algebra

Francisco D. Igual
Texas Advanced Computing Center
Murtaza Ali
Texas Instruments

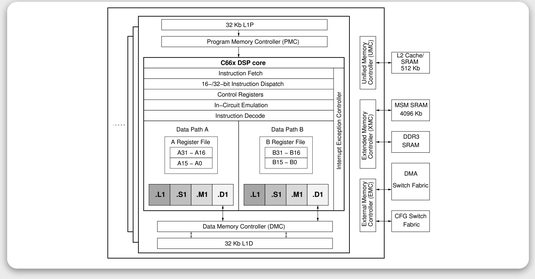
Introduction

DSPs: ubiquity, high performance, low power
- **Ubiquity:** Present in **millions** of commodity devices.
- **High performance:** Recently added **floating-point** capabilities to support 4G networks.
- **Low power:** Promising **10 W per chip**.

Why dense linear algebra (DLA)?
- Representative of the potential performance of architectures.
- Base for higher-level libraries and applications.

Related previous work
- First attempt to port DLA to DSPs
- **libflame's SuperMatrix** previously ported to multi-core and multiGPU. DSP port as a proof of flexibility.
- Same methodology, same codes for all architectures.

Texas Instruments C66x



Core capabilities (up to 8 cores)

Floating point capabilities:
- 16 flops/cycle (single precision).
- SIMD support up to 128 bits.

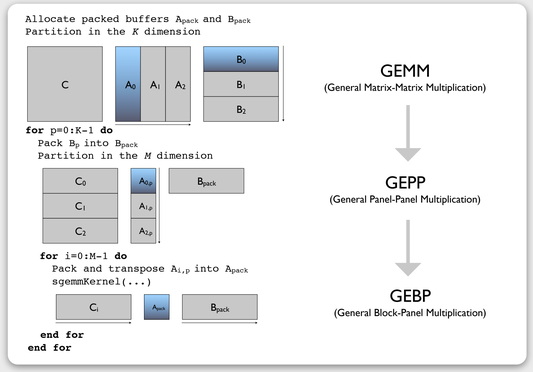
Eight functional units, two register files, two data paths:
- .M perform all multiply operations.
- .S and .L perform general arithmetic, logical and branch.
- .D perform load and store.
- Two general-purpose register files with 32 32-bit registers.

Memory capabilities
- 32 Kb of L1P and L1D cache.
- 4096 Kb of L2 memory (512 Kb per core).
  - Usable as SRAM, cached or mixed.
- 4096 Kb of shared MSM (Multicore Shared Memory).
  - Usable as SRAM or L2.
- 64-bit DDR3 external memory interface at 1600 Mhz with ECC.
- Full flexibility to view caches as pure scratchpad memories.

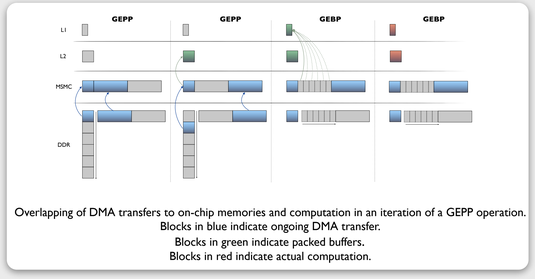
Programmability and multi-thread support
- Native C/C++ code is supported by the TI C/C++ compiler.
- Intrinsics and vector datatypes to improve performance.
- Code Composer Studio as IDE.
- Multi-thread support:
  - OpenMP 3.0.
- No cache coherency between cores.

GEMM on a single core

GotoBLAS approach
- High performance implementation of BLAS on multiple architectures.
- High-level multi-layer approach to take benefits of multiple cache levels.
- We map this approach to the TI DSP at the single core level.

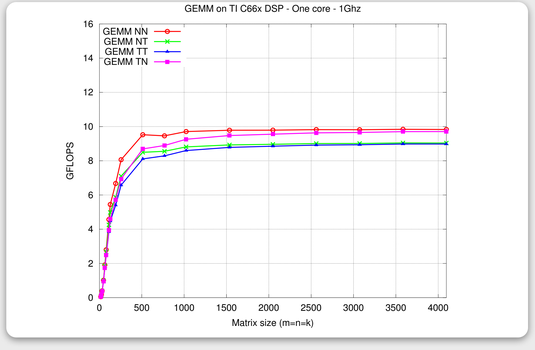


Mapping GotoBLAS to the TI C66x DSP
- TI DSP allows allocation of buffers in L1, L2, MSMC and DDR.
- Use of DMA to overlap computation and transfer between memory layers.
- Highly tuned inner kernel using vector intrinsics.



Overlapping of DMA transfers to on-chip memories and computation in an iteration of a GEPP operation.
Blocks in blue indicate ongoing DMA transfer.
Blocks in green indicate packed buffers.
Blocks in red indicate actual computation.

Experimental results on one core

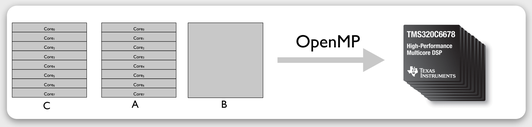


Retargeting to multi-core

Option 1: Manual OpenMP parallelization

Pros: No runtime overhead.

Cons: Programmability. Manually parallelize all BLAS/LAPACK routines.

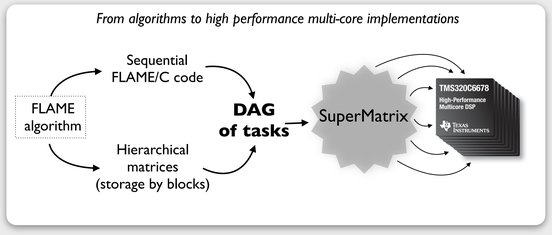


Option 2: Automatic runtime-based parallelization

libflame SuperMatrix

Pros: Programmability. Existing sequential DLA implementations parallelize automatically to the DSP.

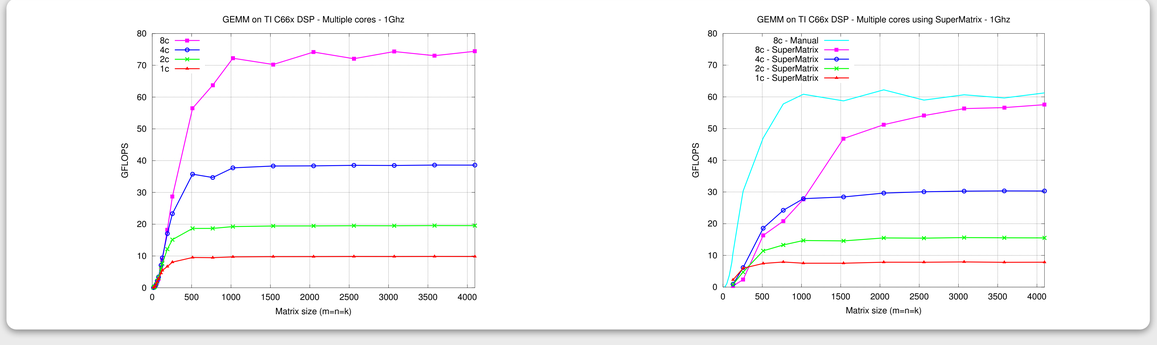
Cons: Runtime overhead, especially for small matrices.



Runtime features:
- Unlike previous solutions, no host is needed.
- Runtime running exclusively in the DSP cores.
- Shares infrastructure with multi-core and multiGPU libflame.

Same DLA codes target multiple architectures, including DSP.

Experimental results on multiple cores



Power efficiency

Architecture	GFLOPS	GFLOPS/Watt	Utilization
Core i7-960	96	1.14	95%
Nvidia GTX280	410	2.6	66%
Cell	200	5.0	88%
Nvidia GTX480	940	5.4	70%
Stratix IV	200	7	90+%
TI C66x DSP	**74**	**7.4**	**57%**

Conclusions

- **GotoBLAS** ideas can be directly ported to single core DSP.
  - Memory hierarchy flexibility and DMA transfers.
  - Excellent performance per core (10 GFLOPS).
  - Scalability to 8 cores (74 GFLOPS).
- **libflame + DSP.**
  - One code, one library, multiple architectures.
  - Successfully ported to multicore DSPs.
  - No need of host: purely standalone low-power DLA solution.
  - Competitive asymptotic performance.
  - Excellent GFLOPS/Watt ratio.

More information...

Visit http://www.cs.utexas.edu/~flame