The emerging and continuing use of multi-core architectures and graphics processing units requires changes in the existing software and sometimes even in a redesign of the established algorithms in order to take advantage of now prevailing parallelism. Parallel Linear Algebra Software for Multi-core Architectures (PLASMA) and Matrix Algebra for GPU and Multicore Architectures (MAGMA) are two projects that aim to achieve high performance and portability across a wide range of multi-core architectures and hybrid systems respectively.

**THE PARALLEL LINEAR ALGEBRA SOFTWARE FOR MULTICORE ARCHITECTURES (PLASMA) PROJECT**

PLASMA aims to create software frameworks that enable programmers to simplify the process of developing applications that can achieve both high performance and portability across a range of new architectures. PLASMA uses a programming model that allows asynchronous, out-of-order scheduling of operations in order to achieve a scalable yet highly efficient software framework for Computational Linear Algebra applications.

**PERFORMANCE RESULTS**

- **Double Precision**
  - CPU: Intel Xeon E5-2698 v2, 2.2 GHz, 8 cores, s/d gemm peak 128/65 GFlop/s
  - NVIDIA GeForce GTX 1080, 1536 CUDA cores, 11.25 TFlop/s
  - Intel Xeon E5-2698 v2, 2.2 GHz, 8 cores, s/d gemm peak 128/65 GFlop/s
  - NVIDIA GeForce GTX 1080, 1536 CUDA cores, 11.25 TFlop/s

**MAGMA**

The Matrix Algebra for GPU and Multicore Architectures (MAGMA) project aims to create a new generation of linear algebra libraries that achieve the fastest possible time to an accurate solution on hybrid/multicore architectures, starting with current multicore-aware GPU systems. To address the challenges stemming from the heterogeneity of these systems, the need for parallelism, and the gap in computation vs. CPU-GPU communication speeds, MAGMA research is based on the idea that optimal software solutions with Hybrid/heterogeneous hardware, balancing the strengths of different algorithms within a single framework.

**HYBRID ALGORITHMS**

- ESSL-based or hybrid algorithms that match algorithms requirements to the architectural strengths of the system and its components.
- Small non-parallelizable tasks, often on the critical path, are scheduled on the CPU, and larger parallelizable algorithms, often Level 3 BLAS, are scheduled on the GPU.

**MAGMA BLAS**

A complementary to CUBLAS subset of CUDA BLAS that features for MAGMA’s needs.

- **Double Precision**
  - CPU/GPU BLAS
  - CPU/CPU BLAS
  - CPU/CPUs
  - Hybrid algorithms on GPUs

**CURRENT RESEARCH**

- Symmetry and Positivity: Non-Symmetric Eigenvalue Problems
- Singular Value Decomposition
- Mixed-precision algorithms

- **SINGLE PRECISION**

- **DOUBLE PRECISION**

**PERFORMANCE RESULTS**

- **CPU**
  - Intel Xeon E5-2698 v2, 2.2 GHz, 8 cores, s/d gemm peak 128/65 GFlop/s
  - NVIDIA GeForce GTX 1080, 1536 CUDA cores, 11.25 TFlop/s
  - Intel Xeon E5-2698 v2, 2.2 GHz, 8 cores, s/d gemm peak 128/65 GFlop/s
  - NVIDIA GeForce GTX 1080, 1536 CUDA cores, 11.25 TFlop/s

- **GPU**
  - NVIDIA GeForce GTX 1080, 1536 CUDA cores, 11.25 TFlop/s
  - Intel Xeon E5-2698 v2, 2.2 GHz, 8 cores, s/d gemm peak 128/65 GFlop/s
  - NVIDIA GeForce GTX 1080, 1536 CUDA cores, 11.25 TFlop/s
  - Intel Xeon E5-2698 v2, 2.2 GHz, 8 cores, s/d gemm peak 128/65 GFlop/s

**Matrix algebra for GPU and Multicore Architectures (MAGMA) Project** aims to create a new generation of linear algebra libraries that achieve the fastest possible time to an accurate solution on hybrid/multicore architectures, starting with current multicore-aware GPU systems. To address the challenges stemming from the heterogeneity of these systems, the need for parallelism, and the gap in computation vs. CPU-GPU communication speeds, MAGMA research is based on the idea that optimal software solutions with Hybrid/heterogeneous hardware, balancing the strengths of different algorithms within a single framework.

**CURRENT RESEARCH**

- Symmetry and Positivity: Non-Symmetric Eigenvalue Problems
- Singular Value Decomposition
- Mixed-precision algorithms

**PERFORMANCE RESULTS**

- **CPU**
  - Intel Xeon E5-2698 v2, 2.2 GHz, 8 cores, s/d gemm peak 128/65 GFlop/s
  - NVIDIA GeForce GTX 1080, 1536 CUDA cores, 11.25 TFlop/s
  - Intel Xeon E5-2698 v2, 2.2 GHz, 8 cores, s/d gemm peak 128/65 GFlop/s
  - NVIDIA GeForce GTX 1080, 1536 CUDA cores, 11.25 TFlop/s

- **GPU**
  - NVIDIA GeForce GTX 1080, 1536 CUDA cores, 11.25 TFlop/s
  - Intel Xeon E5-2698 v2, 2.2 GHz, 8 cores, s/d gemm peak 128/65 GFlop/s
  - NVIDIA GeForce GTX 1080, 1536 CUDA cores, 11.25 TFlop/s
  - Intel Xeon E5-2698 v2, 2.2 GHz, 8 cores, s/d gemm peak 128/65 GFlop/s

**PHOTOS**

- A “bar-and-board” performance plot for “SSSYR” vs. the number of CUDA cores.
- Reduction in message-passing time for eigenvalue computations (double precision).

**DOWNLOAD THE LIBRARY AT**

- [http://icl.eecs.utk.edu/magma/]()