1. BACKGROUND

European Extreme Large Telescope (E-ELT) is a high priority project in ground based astronomy that aims at constructing the largest telescope ever built. MOSAIC is an instrument proposed for E-ELT using Multi-Object Adaptive Optics (MOAO) technique for astronomical telescopes, which compensates for effects of atmospheric turbulence on image quality, and operates on patches across a large FoV.

2. MOTIVATION

• Real time simulation needed.
• Extremely compute intensive.
• Core computation in constructing Tomographic Reconstructor (TR).
• Drives in real-time the Deformable Mirror (DM).
• Repeatedly computed by inverting huge dense symmetric covariance matrix (order of 100k x 100k).

3. CONTRIBUTION

• Regularize the numerically unstable problem for TR computation [1], by accounting for the natural noise from WFS.
• Positive definite invertible matrix, allowing for the less expensive direct Cholesky based matrix inversion.
• Operate on tile algorithms – exposing more concurrency – through MORSE dense linear algebra library [2].
• Accelerate computation using HPC (High Performance Computing) tools that can utilize the hardware resources (CPU cores and GPUs) to near optimal performance.
• Deploy concepts of dataflow programming model to facilitate data management, dynamic scheduling and synchronization-reducing flow of task execution. Established by pipelining different stages of computation through the dynamic runtime system StarPU [3].
• Unprecedented speed and scale on shared memory system.

4. IMPLEMENTATION

• TR computation of 4 main stages: Cholesky factorization (DPOTRF), Cholesky-based inversion (DTRTRI & DLAUUM), symmetric matrix-matrix multiplication (DSYMM); breaks down to smaller subtasks as shown in the DAG figure.
• Tile-based implementations by MORSE over both CPU & GPU.
• StarPU builds a DAG representing tasks and their dependencies.
• Asynchronously schedules these tasks out of order.
• Scheduling policy that is data aware, relies on history based performance models (calibrated and adapted to available hardware).
• High priority to tasks at the critical path.

5. RESULTS

Experiments run on a shared memory dual-socket Intel(R) Xeon(R) CPU E5-2650 of eight cores each, running at 2GHz, with a total of 64GB of main memory equipped with four NVIDIA Kepler K20c GPUs

6. DISCUSSION

• Close to 85% of efficiency, up to 13-fold speedup.
• Scheduler able to cope with hardware complexity, maximizing available resources usage, overlap of computation and data-transfer.
• 16-core performance hits close to the theoretical peak of x86 system.
• 1-Tflop/s scored with addition of 1-GPU.
• Scaling decreases with more accelerators, scheduler overhead.
• Memory congestion, out of core execution triggered, expensive memory footprint of Cholesky based inversion.

7. FUTURE WORK

• Enhance the scheduling policy to overcome memory congestion.
• Target flat shared memory system (8 – GPU system) to validate scalability.
• Port application to distributed architectures to gain bigger boost in performance.

REFERENCES