A Synergistic Approach for Abstracting Hardware Heterogeneity and Reducing Algorithmic Complexity: Powering HiCMA with oneAPI for HPC Applications

Hatem Ltaief

Principal Research Scientist
Extreme Computing Research Center

Intel oneAPI DevSummit at SC21 (Live Virtual)
November 14, 2021

KAUST oneAPI

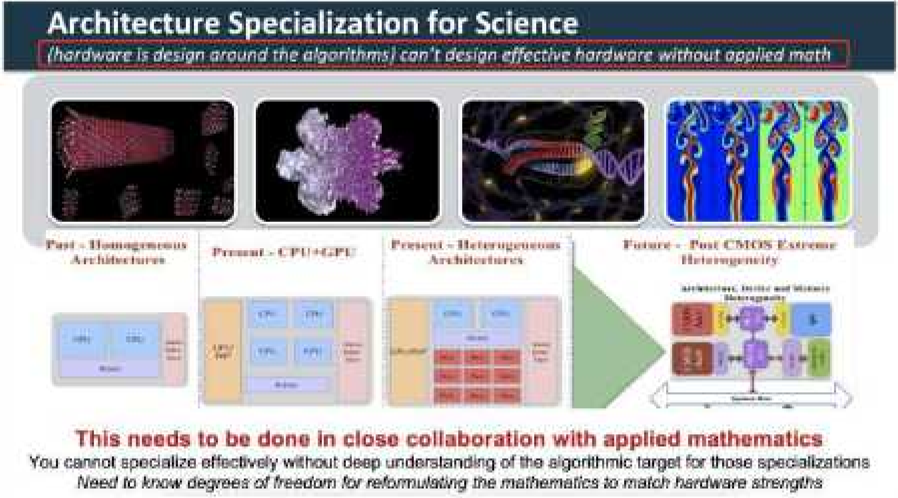
Acknowledgements

- ECRC: *S. Abdulah, K. Akbudak, R. Alomairy, M. Al Farhan, M. Genton, Y. Hong, D. Keyes, M. Ravasi, and Y. Sun*
- Paris Observatory: *F. Ferreira, D. Gratadour, A. Sevin*
- Research School of Astronomy & Astrophysics, Australian National University: *J. Cranney*
- Intel: *H. Ibeid, P. Thierry, A. Al-Jeshi, A. Abduljabbar*
- KAUST Supercomputing Lab: *S. Feki, B. Hadri, and S. Kortas*

HPC Recipe For Exascale Computing

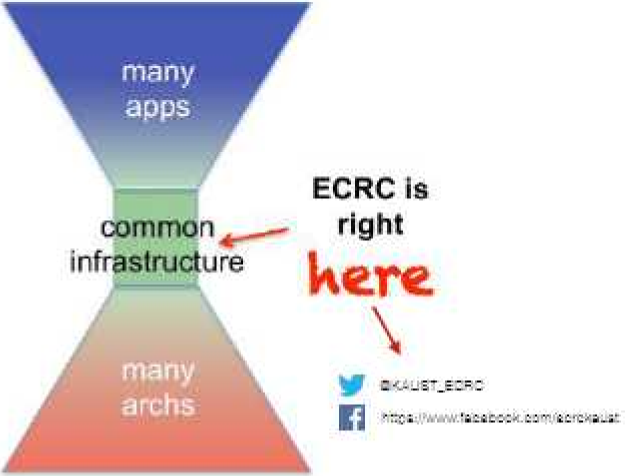
HPC Recipe For Exascale Computing

A Hostile Hardware Landscape



Slide courtesy from John Shalf, LBNL

The Hourglass Revisited

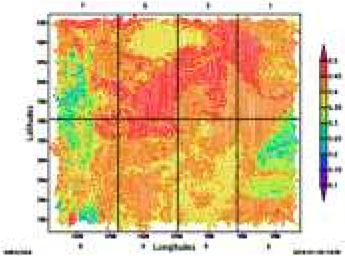


Performing Climate/Weather Forecasting Simulations

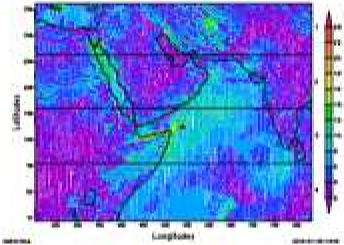
- Computational statistics: multivariate large spatial data sets in climate/weather modeling:

$$\ell(\boldsymbol{\theta}) = -\frac{1}{2}\mathbf{Z}^T\Sigma^{-1}(\boldsymbol{\theta})\mathbf{Z} - \frac{1}{2}\log|\Sigma(\boldsymbol{\theta})|$$

(a) Problem Definition.



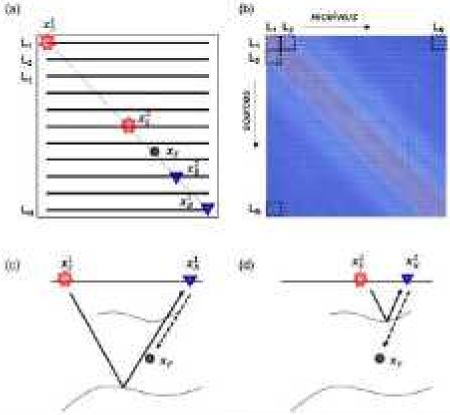
(b) Soil moisture.



(c) Wind speed.

Revealing the Underground Layers with Seismic Imaging

- Transitioning to new energy technologies requires an even higher level of details to make informed decision
- Monitoring permafrost degradation
- Seismic redatuming is key!



Outsmarting the Atmospheric Turbulence in AO

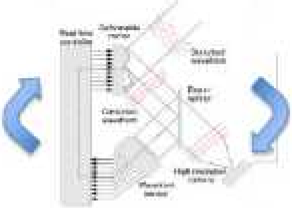
(a) VLT.

(b) Subaru.

(c) E-ELT.

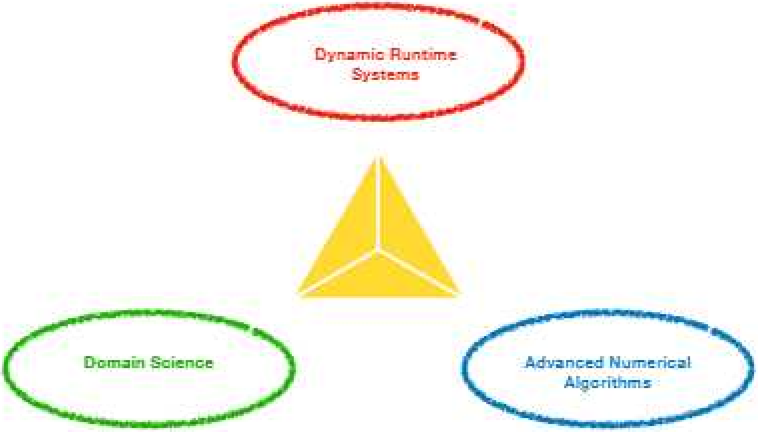
(d) ScExAO.

(e) MAVIS.



(f) Soft/Hard RTC.

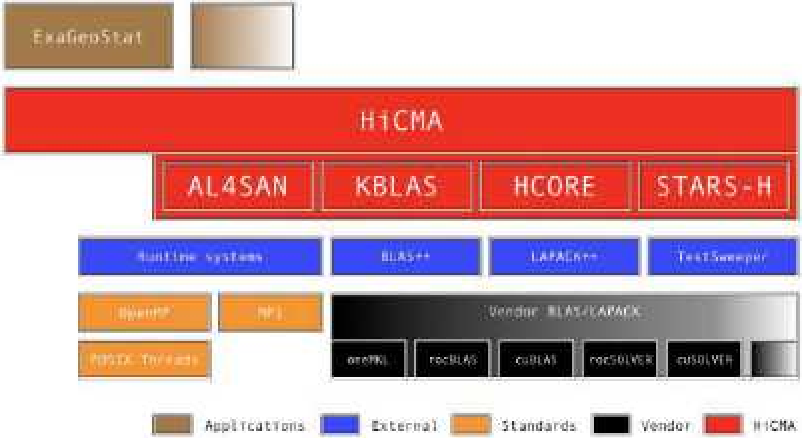
An Effective Approach Based on a Separation of Concerns



Exploiting the Data Sparsity of these Matrices

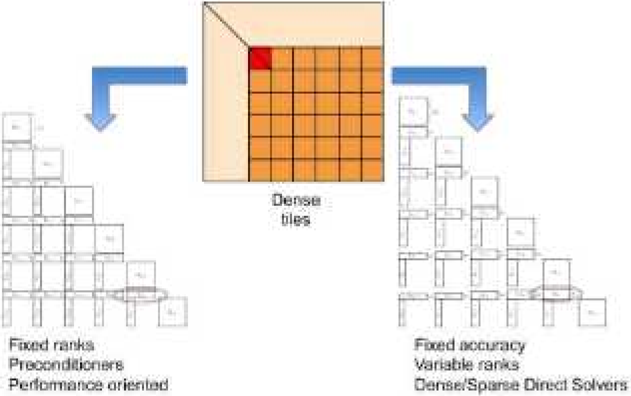
- Ubiquitous in computational science and engineering
- Symmetric, positive-definite matrix structure
- Apparently dense matrices but often data-sparse
- Decay of parameter correlations with distance

The HiCMA Library



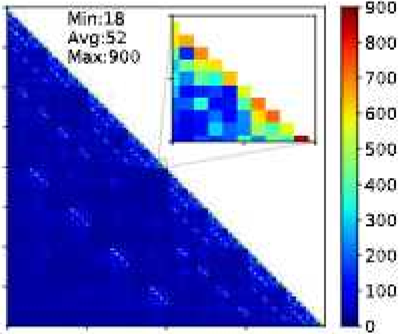
Available at http://github.com/ecrc/hicma

Tile Low-Rank Data Format as a Pragmatic Approach

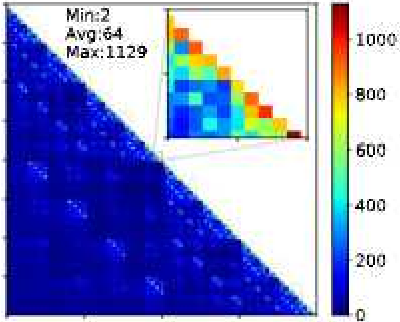


T. Mary, PhD Dissertation, Block Low-Rank multifrontal solvers: complexity, performance, and scalability, 2017.
C. Weisberger, PhD Dissertation, Improving multifrontal solvers by means of algebraic Block Low-Rank representations, 2013.

Rank Heatmaps of 3D Exponential Kernel with $N = 1\text{M}$, $b = 2700$, and $acc = 1e-8$



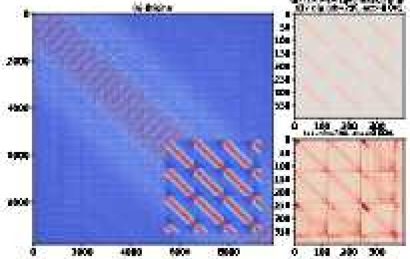
(a) Initial ranks.



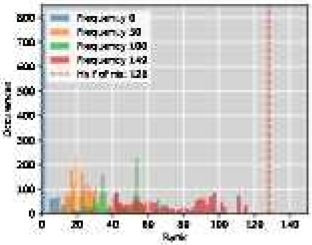
(b) Final ranks.

Q. Cao, Y. Pei, K. Akbudak, G. Bosilca, H. Ltaief, D. Keyes, and J. Dongarra: Leveraging PaRSEC Runtime Support to Tackle Challenging 3D Data-Sparse Matrix Problems, IPDPS'21

Rank Analysis for Seismic Frequency Matrices



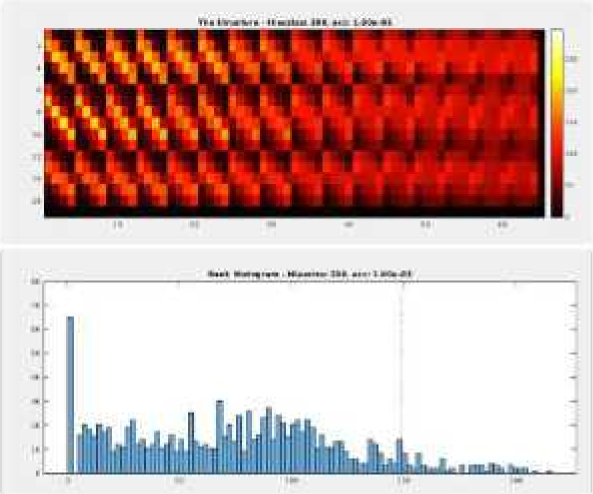
(a) Rank Heatmaps.



(b) Rank Distribution.

Y. Hong, H. Ltaief, M. Ravasi, L. Gatineau, D. Keyes, Journal of Supercomputing Frontiers and Innovations'21

Rank Analysis for MAVIS Instrument on the Very Large Telescope



H. Ltaief, J. Cranney, D. Gratadour, Y. Hong, L. Gatineau, D. Keyes, SC'21

Implementation Details: Programming Models

OpenMP

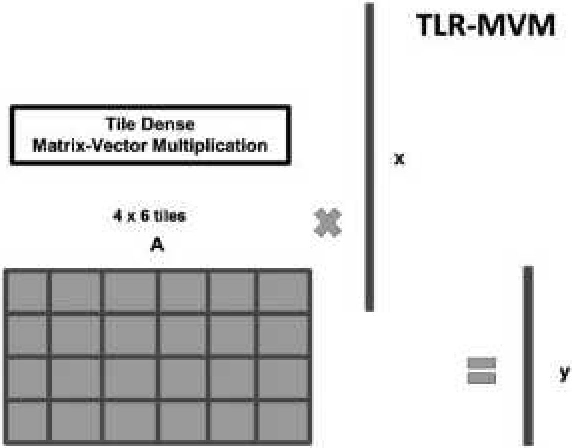
oneAPI

Implementation Details: TLR Cholesky Factorization

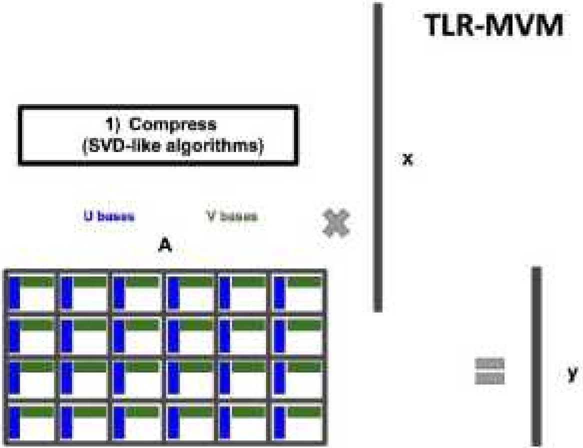
Algorithm 1 HiCMA_DPOTRF(HicmaLower, D, U, V, N, nb, rank, acc)

```
p = N / nb
for k = 1 to p do
    #pragma omp task depend(inout:D(k,k))
    hcore_dpotrf(HicmaLower, D(k,k), rank, acc)
    for i = k+1 to p do
        #pragma omp task depend(in:D(k,k)) depend(inout:U(i,k))
        hcore_dtrsm(V(i,k), D(k,k), rank, acc)
    end for
    for j = k+1 to p do
        #pragma omp task depend(in:U(j,k)) depend(in:V(j,k)) depend(inout:D(j,j))
        hcore_dsyrk(D(j,j), U(j,k), V(j,k), rank, acc)
        for i = j+1 to p do
            #pragma omp task
                depend(in:U(i,k)) depend(in:V(i,k))
                depend(in:U(j,k)) depend(in:V(j,k))
                depend(inout:U(i,j)) depend(inout:V(i,j))
            hcore_dgemm(U(i,k), V(i,k), U(j,k), V(j,k), U(i,j), V(i,j), rank, acc)
        end for
    end for
end for
```

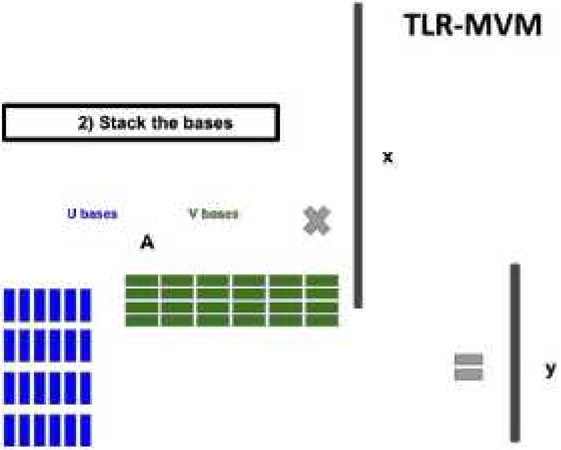
Implementation Details: TLR-MVM for MAVIS Instrument and Seismic Redatuming



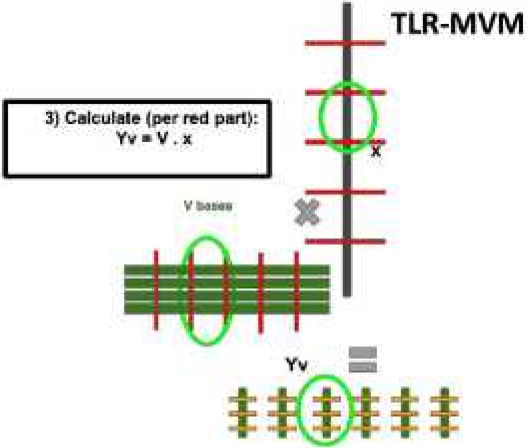
Implementation Details: TLR-MVM for MAVIS Instrument and Seismic Redatuming



Implementation Details: TLR-MVM for MAVIS Instrument and Seismic Redatuming

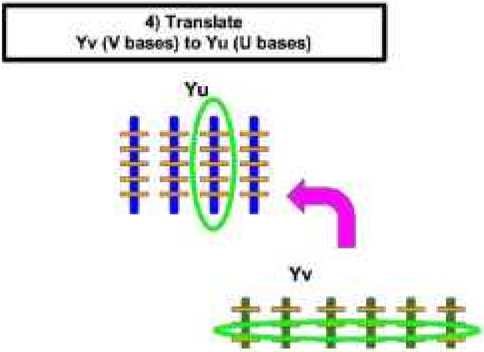


Implementation Details: TLR-MVM for MAVIS Instrument and Seismic Redatuming



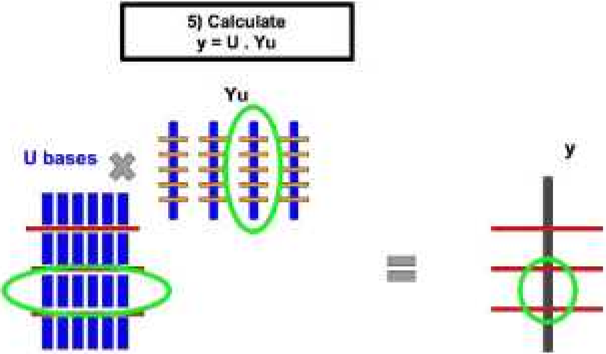
Implementation Details: TLR-MVM for MAVIS Instrument and Seismic Redatuming

TLR-MVM

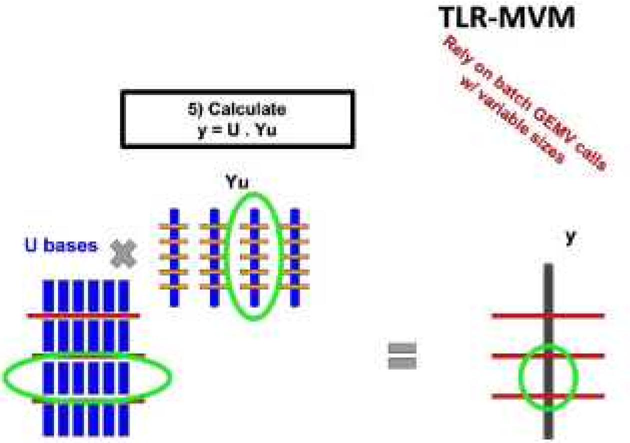


Implementation Details: TLR-MVM for MAVIS Instrument and Seismic Redatuming

TLR-MVM



Implementation Details: TLR-MVM for MAVIS Instrument and Seismic Redatuming



Implementation Details: TLR-MVM OpenMP for MAVIS Instrument and Seismic Redatuming

Algorithm 2 TLR-MVM_OpenMP(Utiles, Vtiles, YuYvMapping, x)

```
ntiles = N / nb, mtiles = M / nb
#pragma omp parallel for
for k = 1 to ntiles do
    cblas_gemv(Vtiles(k), x(k), yv(k)) // Phase 1: yv is output
end for
#pragma omp parallel for
for k = 1 to length of yv do
    yu(YuYvMapping(k)) = yv(k) // Phase 2: yu is output
end for
#pragma omp parallel for
for k = 1 to mtiles do
    cblas_gemv(Utiles(k), yu(k), y(k)) // Phase 3: y is output
end for
```

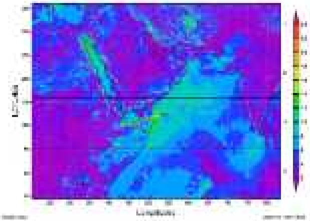
Implementation Details: TLR-MVM oneAPI for MAVIS Instrument and Seismic Redatuming

Algorithm 3 TLR-MVM_oneAPI(Utiles, Vtiles, YuYvMapping, x, devicequeue)

```
ntiles = N / nb, mtiles = M / nb
#pragma omp parallel for
for k = 1 to ntiles do
    oneAPI_gemv(Vtiles(k), x(k), yv(k), devicequeue) // Phase 1: yv is output
end for
devicequeue.wait()
devicequeue.parallell_for{yu(YuYvMapping(k)) = yv(k)} // Phase 2: yu is output
devicequeue.wait()
#pragma omp parallel for
for k = 1 to mtiles do
    oneAPI_gemv(Utiles(k), yu(k), y(k), devicequeue) // Phase 3: y is output
end for
devicequeue.wait()
```

Numerical Accuracy Assessment: TLR Cholesky for MLE

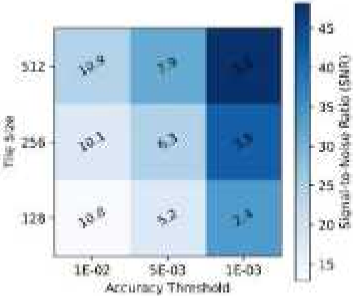
- Estimation of the Matérn covariance parameters for four geographical regions of wind speed dataset.

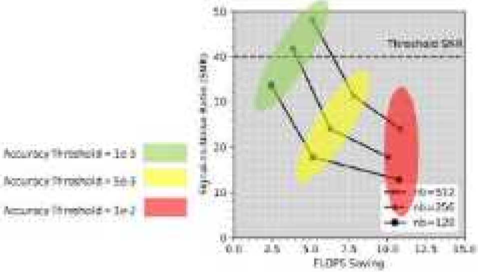


R	Matérn Covariance Variance (θ_1) TLR Accuracy				Spatial Range (θ_2) TLR Accuracy				Smoothness (θ_3) TLR Accuracy			
	10^{-5}	10^{-7}	10^{-9}	*Full-tile*	10^{-5}	10^{-7}	10^{-9}	*Full-tile*	10^{-5}	10^{-7}	10^{-9}	*Full-tile*
R1	7.406	9.407	12.247	8.715	29.576	33.886	39.573	32.083	1.214	1.196	1.175	1.210
R2	11.970	13.159	13.550	12.517	26.011	28.063	28.707	27.237	1.290	1.267	1.260	1.274
R3	10.588	10.944	11.232	10.819	18.423	18.783	19.114	18.634	1.418	1.413	1.407	1.416
R4	12.408	17.112	12.388	12.270	17.264	17.112	17.247	17.112	1.168	1.170	1.168	1.170

S. Abdulah, H. Ltaief, Y. Sun, M. Genton, and D. Keyes, IEEE Cluster'18

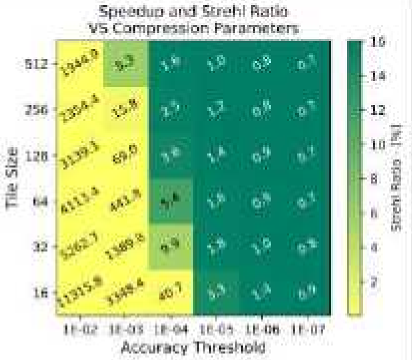
Numerical Accuracy Assessment: TLR-MVM for Seismic Redatuming

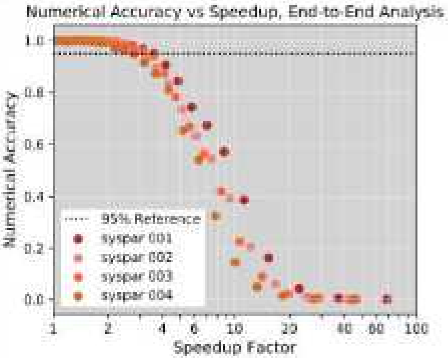




Y. Hong, H. Ltaief, M. Ravasi, L. Gatineau, D. Keyes, Journal of Supercomputing Frontiers and Innovations'21

Numerical Accuracy Assessment: TLR-MVM for MAVIS Instrument





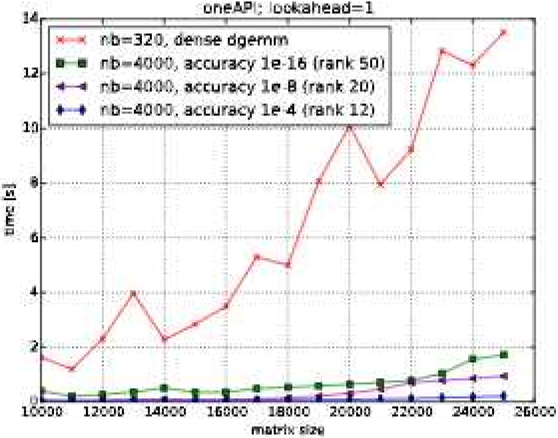
H. Ltaief, J. Cranney, D. Gratadour, Y. Hong, L. Gatineau, D. Keyes, SC'21

Environment Settings

Vendor	Intel
Family	Ice Lake
Model	Intel(R) Xeon(R) Gold 6330
Node(s)/Card(s)	1
Socket(s)	2
Cores	56
GHz	2.0
Memory Sustained BW	1000GB DDR4 320GB/s
LLC Sustained BW	84MB 1.1TB/s
Compiler	DPC++/C++ compiler 2021.3.0
BLAS library	Intel oneMKL 2021.3.0
MPI library	IntelMPI 2021.3.0

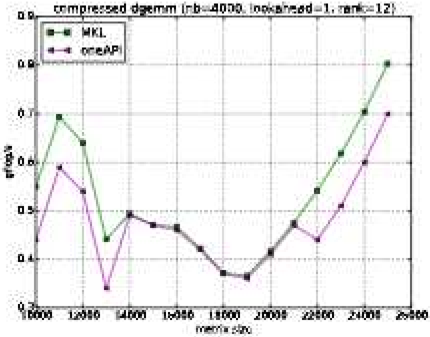
Preliminary Performance Results: TLR GEMM toward MLE Computations in Climate/Weather Prediction (1)

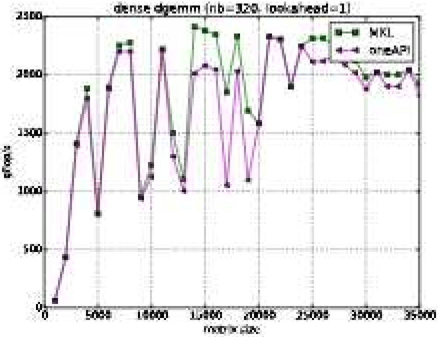
- Tile low-rank GEMM with OpenMP Task



Preliminary Performance Results: TLR GEMM toward MLE Computations in Climate/Weather Prediction (2)

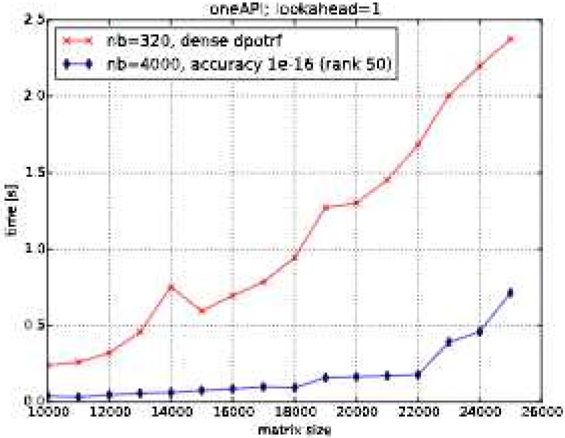
- Tile low-rank GEMM with OpenMP Task Vs oneAPI





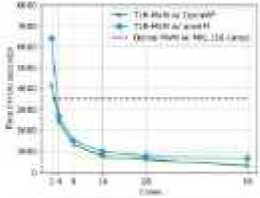
Preliminary Performance Results: TLR Cholesky toward MLE Computations in Climate/Weather Prediction (3)

- Tile low-rank Cholesky with oneAPI



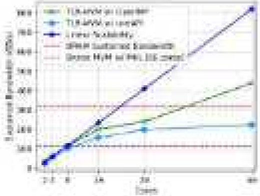
Preliminary Performance Results: TLR-MVM for MAVIS Instrument and Seismic Redatuming

Lower is better



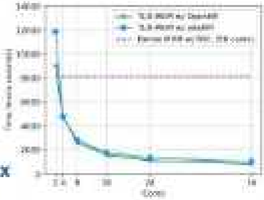
MAVIS matrix

Higher is better



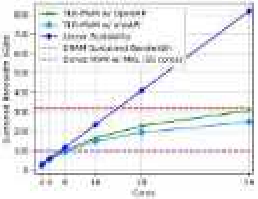
Data size: 152MB

Lower is better



Seismic Matrix (Frequency Index 100)

Higher is better

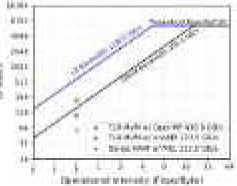


Data size: 260MB

Preliminary Performance Results: TLR-MVM for MAVIS Instrument and Seismic Redatuming

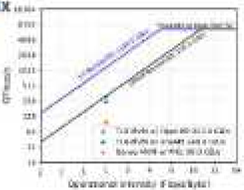
MAVIS Matrix

Higher is better



Seismic Matrix (Frequency Index 100)

Higher is better



Operational Intensity (OI) of TLRMVM:

$$OI(fp32) = \frac{FLOPS}{BYTE} = \frac{4Rb}{(n+m+4R+2Rb)\times B} \approx \frac{2}{B} \approx \frac{1}{2}$$

$$OI(complex) = \frac{FLOPS}{BYTE} = \frac{16Rb}{(n+m+4R+2Rb)\times B} \approx \frac{8}{B} \approx 1$$

- m, n : rows and columns of input Matrix A
- R : Total rank summation of Matrix A at a given threshold
- B : Bytes per element : 4 for MAVIS Matrix (single) 8 for Seismic Matrix (single complex)
- b : The tile size of small block matrix (256)

Conclusion and Future Work

- Describe a synergistic approach to leverage performance of HPC applications on heterogeneous hardware architectures
- Enhance user-productivity with *oneAPI* programming model
- Study the interoperability between OpenMP/oneAPI
- Plan to port on Intel FPGAs and GPUs

Thank You! Questions?