# Large-Scale Climate/Weather Statistical Modeling and Prediction with MVAPICH2

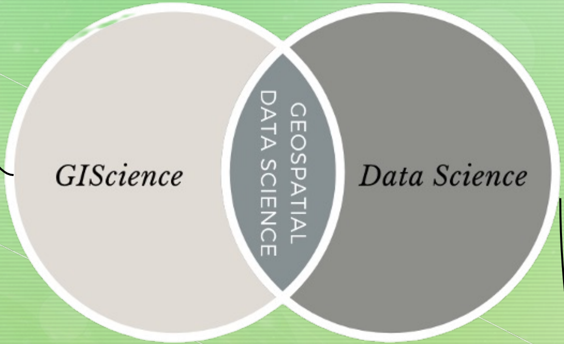
Sameh Abdulah
Research Scientist

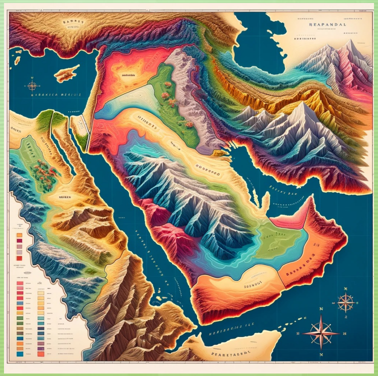
Extreme Computing Research Center (ECRC),
King Abdullah University of Science and Technology (KAUST)

# Geospatial Data Science

- The discipline that specifically focuses on the spatial component of the data science

The scientific study of geographic concepts, applications, and systems.

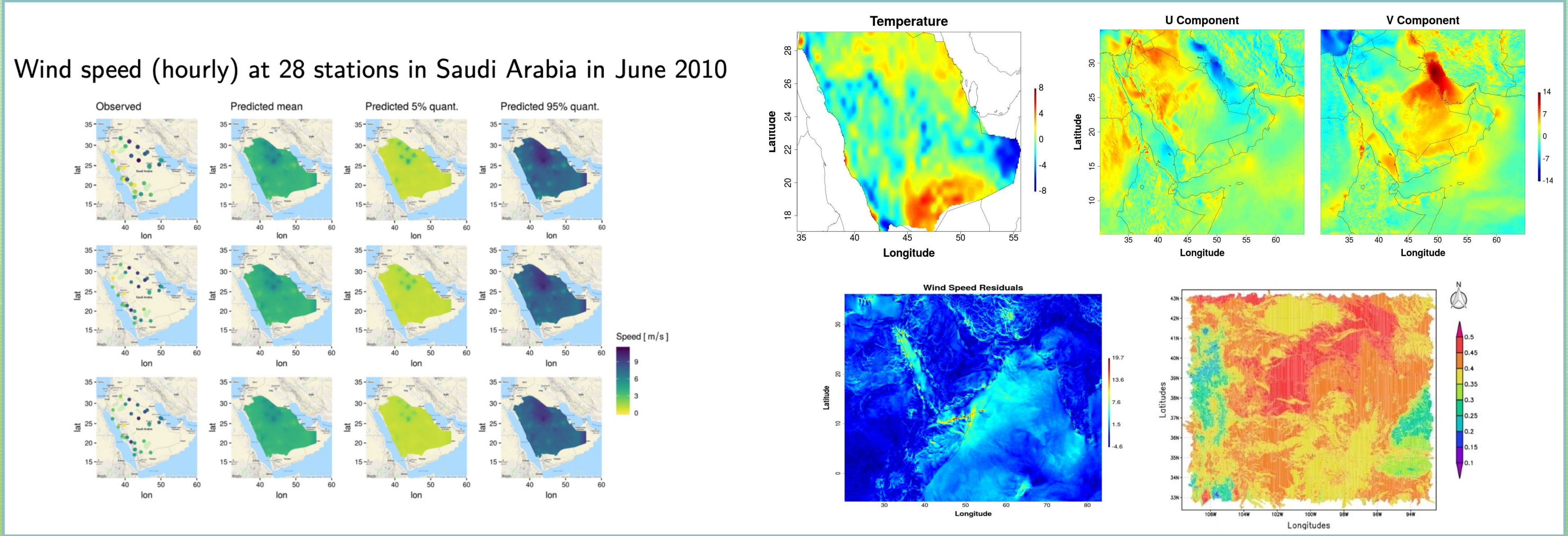


- Foster interdisciplinary field.
- Extract knowledge from structured and unstructured data.
- Apply the extracted knowledge to applications.
- Leverage techniques from mathematics, statistics, information science, and domain knowledge.

# Perform Climate/Weather Forecasting Simulations

- Applications for climate and environmental predictions are among the most time-consuming simulations workloads running on HPC facilities
- Computational statistics: univariate/multivariate large spatial datasets in climate/weather modeling



# Pop Stats for Big Geodata

- An upsurge in generated geodata has been noted, yet the techniques for processing millions of observations have fallen behind
- Implementations that work with irregularly spaced observations are rare
- Various approximation methods have been proposed in the literature to ease the computation and memory burden
- HPC can be a game changer allowing dense computation for big geodata!
- However, other scientific fields show be included: linear algebra, optimization, data science, supervised learning

# Perform Climate/Weather Forecasting Simulations

- Applications for climate and environmental predictions are among the most time-consuming simulations workloads running on today's supercomputer facilities
- Today, weather and climate data are usually huge!
  - A set of univariate/multivariate Z observations at given n locations on one or more time slots
  - Z observations could be temperature, precipitation, ... etc

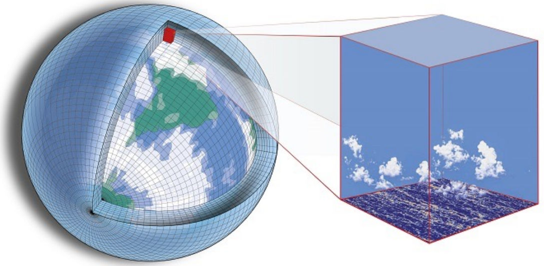
- Maximum Likelihood Function: An important statistical technique for modeling data in climate and environmental applications
  - Prohibitive computational Cost and memory requirements:

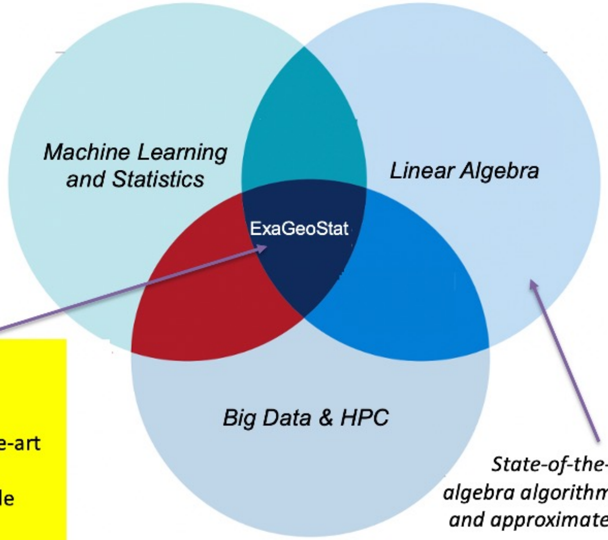
$$l(\theta) = -\frac{1}{2} Z^T \Sigma^{-1}(\theta) Z - \frac{1}{2} \log|\Sigma(\theta)| \qquad \boldsymbol{\Sigma(\theta)}$$

  - The linear solver and log-determinant involving floating point operations on n-by-n covariance matrix $\boldsymbol{\Sigma}(\boldsymbol{\theta})$ with $O(n^3)$ complexity and $O(n^2)$ memory footprint
  - For instance: $10^6$ locations require 8TB Memory!

# ExaGeoStat in a Nutshell

- *ExaGeoStat* for:
  1. Likelihood inference/learning for Matérn covariance function (among others)
  2. Spatial kriging (interpolation)
  3. Random field simulations
  4. Multivariate Gaussian probabilities
  5. Robust spatial inference






Exascale Geostatistics (ExaGeoStat) is A multidisciplinary software which exploits **machine learning**, **statistical modeling and forecasting**, the state-of-the-art **linear algebra algorithms**, **and supercomputing simulations** to handle large-scale Geostatistics data.

*State-of-the-Art linear algebra algorithm for both exact and approximated computation*

# The ExaGeoStat Software Stack


X86 CPU


AArch64


Fujitsu A64FX

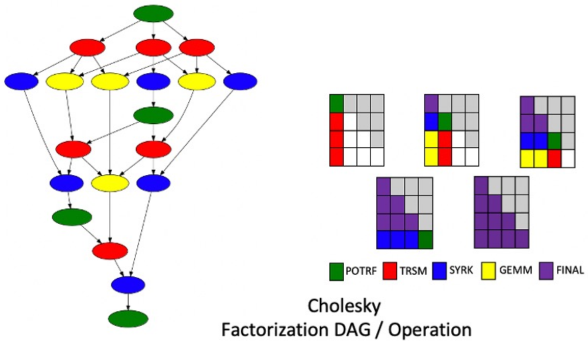
NVIDIA V100


AMD MI250X

Climate/Weather Applications

Modeling/Inference

Optimization Library (e.g., **NLopt**)




Cholesky
Factorization DAG / Operation
(4 X 4 Tile Matrix)

StarPU/PaRSEC Dynamic Runtime System

Drivers (e.g., Pthreads, CUDA, OpenCL, MPI)

| Shared Memory Systems | Distributed Memory Systems |
| --- | --- |

| X86 CPU | AArch64 CPU | GPU |
| --- | --- | --- |

#1 Frontier


#4 Fugaku

#7 Summit



#42 HAWK

#141 Shaheen-II

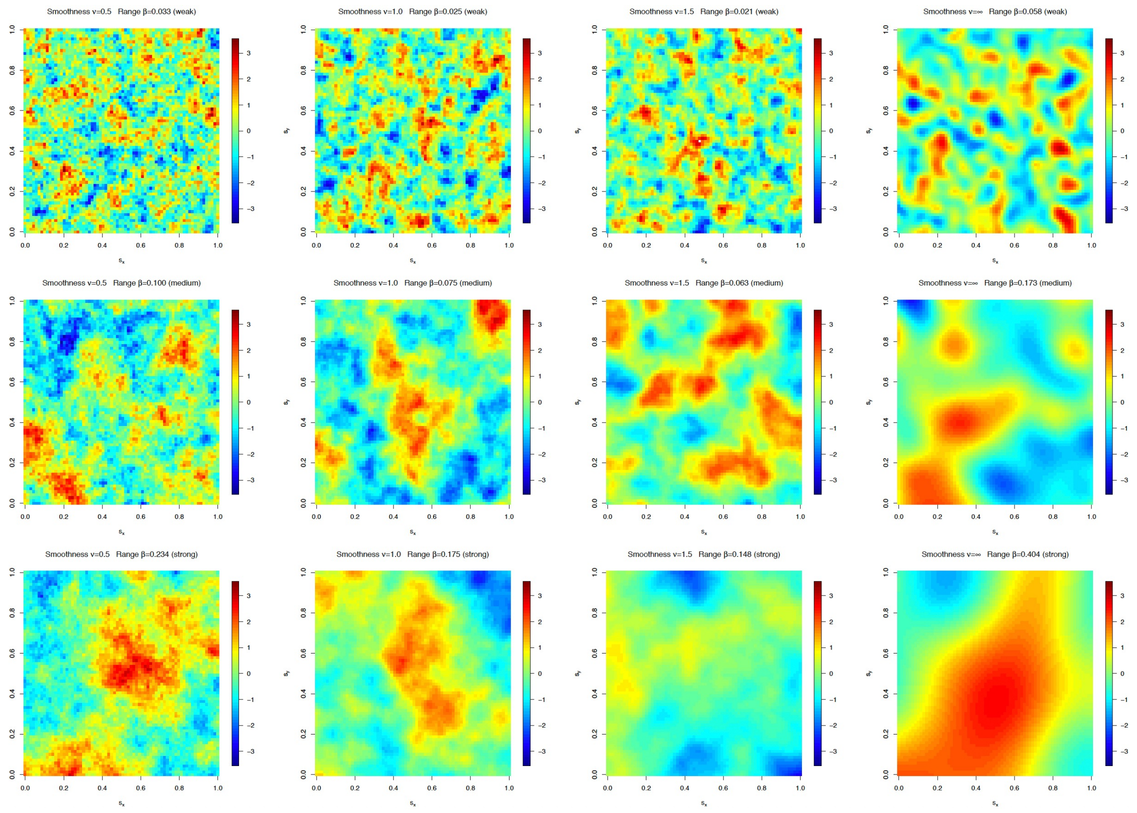
# Matérn Covariance Function

- In the Gaussian random field, different covariance functions can be used to generate $\boldsymbol{\Sigma}(\boldsymbol{\theta})$

$$cov\{Z(\boldsymbol{s}_i), Z(\boldsymbol{s}_j)\} = \sigma^2 \frac{2^{1-\upsilon}}{\Gamma(\upsilon)} \left(\frac{||\boldsymbol{s}_i - \boldsymbol{s}_j||}{\beta}\right)^{\upsilon} K_{\upsilon}\left(\frac{||\boldsymbol{s}_i - \boldsymbol{s}_j||}{\beta}\right) + \tau^2 \mathbb{I}_{\{i=j\}}$$

- A Generic covariance function can be directly used (Matérn function):
  - $\sigma^2 > 0$ (Variance)
  - $\beta > 0$ (Spatial Range, larger values → strong correlation)
  - $\upsilon > 0$ (Smoothness, larger values → smoother field)
- Apparently, dense matrices arising in climate/weather applications,
  - Rely on leading-edge parallel architectures
  - Compress the dense covariance matrix, e.g., tile low-rank approximation
    - Huge performance improvement via cutting down flops
    - Preserving the accuracy requirements of the scientific application
  - Reduce the precision accuracy of the given covariance matrix

# Simulated Gaussian Random Fields using Matérn CF




Weak Correlation

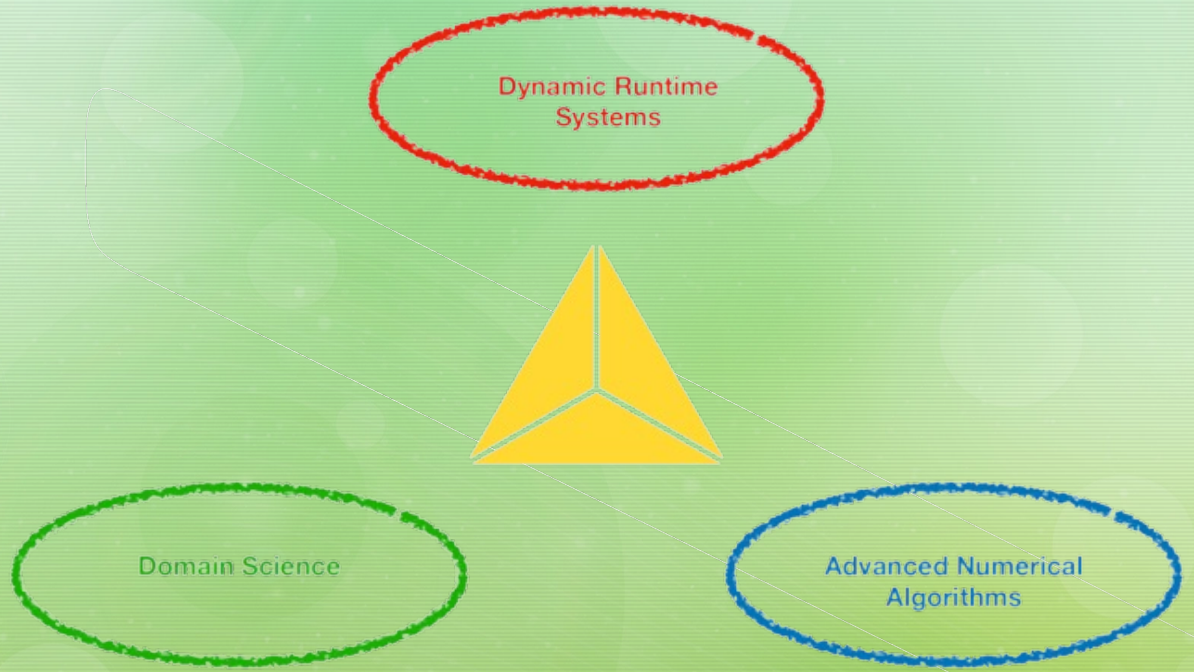
Medium Correlation

Strong Correlation

# Software Functionality

- Synthetic Dataset Generator
  - Generates large-scale geospatial datasets which can be separately used as benchmark datasets for other software packages
- Maximum Likelihood Estimator (MLE)
  - Evaluates the maximum likelihood function on large-scale geospatial datasets
  - Supports full machine precision (full-matrix), Tile Low-Rank (TLR) approximation, low-precision approximation accuracy
- ExaGeoStat Predictor
  - Predicts unknown measurements at known geospatial locations by leveraging the MLE estimated parameters

# Separate Areas of Interest



# Parallel Programming

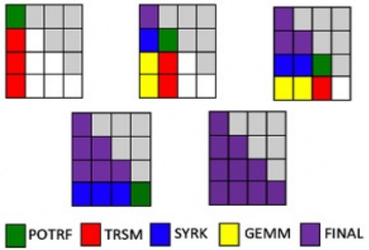
- Multicore programming: Essential for leveraging the power of multicore processors
  - Thread-based Programming: e.g. POSIX threads (pthreads)
  - OpenMP (Open Multi-Processing)
  - Threading Building Blocks (TBB)
- Accelerator Programming
  - OpenCL and CUDA
  - VHDL
  - OpenACC
  - SYCL
- Hyprid models (heterogenous systems): writing software that can run across a combination of different types of processors or cores
  - One solution is to use dynamic runtime system and task-based parallelism
    - The program is divided into tasks, which are the smallest units of work that can be scheduled and executed independently. You can understand the code flow though design a parallel task graph.

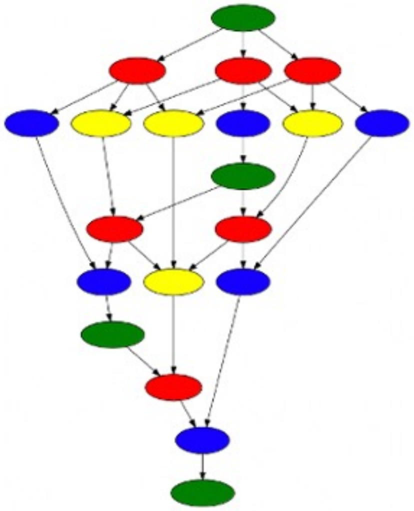
# Parallel Task Graphs

- Parallel task graphs: are a visual representation of tasks that can be executed in parallel, highlighting the dependencies between them
- They are a critical part of parallel computing and are used to optimize and manage the execution of tasks on parallel processors

**Algorithm 1.** Tiled Cholesky Factorization

**for** $k = 0$ to $n-1$ **do**
  $A_{k,k} \leftarrow POTRF(A_{k,k})$ $\{\mathrm{POTRF}_k\}$
  **for** $i = k+1$ to $n-1$ **do**
    $A_{i,k} \leftarrow TRSM(A_{k,k}, A_{i,k})$ $\{\mathrm{TRSM}_{i,k}\}$
  **end for**
  **for** $j = k+1$ to $n-1$ **do**
    $A_{j,j} \leftarrow SYRK(A_{j,k}, A_{j,j})$ $\{\mathrm{SYRK}_{j,k}\}$
    **for** $i = j+1$ to $n-1$ **do**
      $Ai, j \leftarrow GEMM(A_{i,k}, A_{j,k})$ $\{\mathrm{GEMM}_{i,j,k}\}$
    **end for**
  **end for**
**end for**



# Dynamic Runtime Systems

- Multicore programming: Essential for leveraging the power of multicore processors
  - Thread-based Programming: e.g. POSIX threads (pthreads)
  - OpenMP (Open Multi-Processing)
  - Threading Building Blocks (TBB)
- Accelerator Programming
  - OpenCL and CUDA
  - VHDL
  - OpenACC
  - SYCL
- Hyprid models (heterogenous systems): writing software that can run across a combination of different types of processors or cores
  - One solution is to use dynamic runtime system and task-based parallelism
    - The program is divided into tasks, which are the smallest units of work that can be scheduled and executed independently. You can understand the code flow though design a parallel task graph.

# Matrix Approximations / Data Structures

The picture can't be displayed.

Exact

Tile Low-Rank (TLR)

Mixed-precision (MP)

Mixed-precision (MP) / TLR

# Mixed-Precision (MP) Computation

The picture can't be displayed.

The picture can't be displayed.

Adaptive decision map for Matern 2D space on 1M matrix;
default dense double is 4356GB

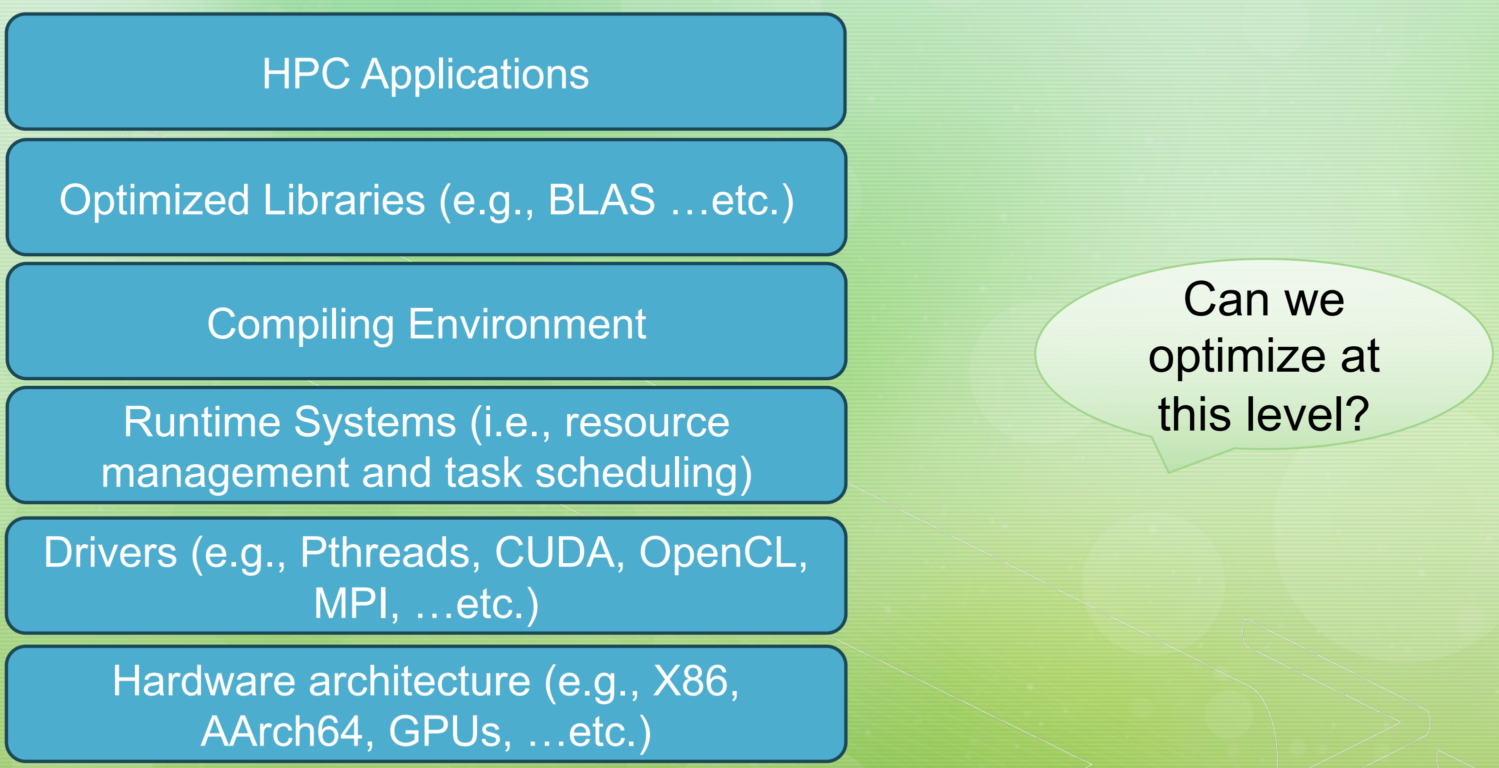
# Precision map of with different spatial statistics kernel

The picture can't be displayed.

# Performance of precision conversion strategies and efficiency on one GPU

The picture can't be displayed.

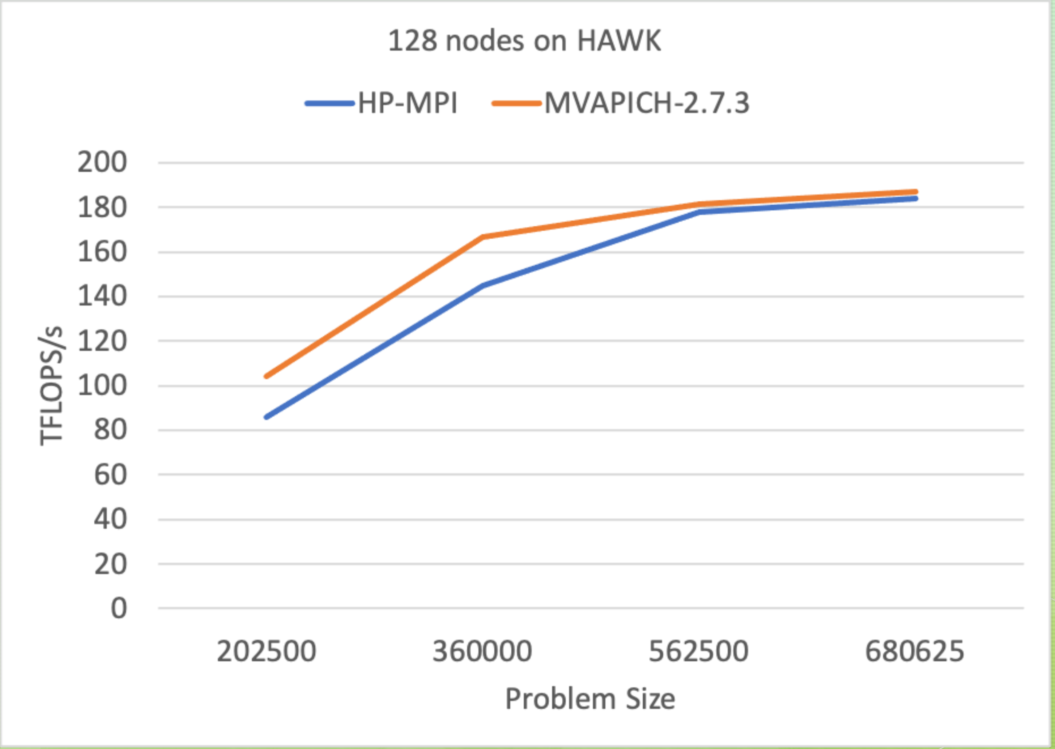
# Back to The Dynamic Runtime Systems



# HLRS HPE Apollo (Hawk) System

- Number of compute nodes: 5,632
  - CPU Type: AMD EPYC 7742
- Number of compute cores: 720,896
- System peak performance: 26 Petaflops
- Total system memory: ~1.44 PB

# Performance with HP-MPI and MVAPICH2



# Profiling with HP-MPI and MVAPICH2

- Experiments with MVAPICH2 version 2.3.7 and HP-MPI
- The operations are exclusively point-to-point,
- Only a single core from each node involved in the inter-node communication.
- Tests were run across 2, 4, and 16 nodes
- Notably, there was a significant increase in the time taken for the mpi_irecv operation when using HP-MPI, For example, with 16 nodes and 40K problem sizes, the mpi_irecv operation required 3.353 ms with HP-MPI, in contrast to 0.949 ms with MVAPICH2
- This speedup comes from the the optimized non-blocking P2P MPI operations in MVAPICH2

# ExaGeoStatCPP

- V 1.0.0 has been Released on Nov 12th 2023

- A cutting-edge C++ API designed for the ExaGeoStat framework. This new API is tailored for C++ developers, combining traditional programming practices with modern C++ elements like namespaces, templates, and exceptions to enhance functionality significantly

- Easier and Faster Installation: ExaGeoStatCPP is engineered to offer a more straightforward and quicker installation process for all dependencies across various systems. This enhancement is a game-changer, significantly improving users' productivity and streamlining the overall user experience

- https://github.com/ecrc/ExaGeoStatCPP

# Summary

- We tackle the complexity of computing the inverse of the covariance matrix $\boldsymbol{\Sigma}(\boldsymbol{\theta})$ in spatial data modeling and prediction by proposing a tile-centric approximation method that is able to take advantage of both tile-low rank and mixed-precision approximations
- We rely on StarPU runtime system to orchestrate data distribution and movement by scheduling asynchronously tasks operating on dense / TLR / mixed-precision data structure
- We also assess the performance of the mixed-precision approximation on NVIDIA GPUs, V100, A100, and H100
- We reduce data transfers by relying on an automated precision conversion strategy
- We evaluated the performance of ExaGeoStat on the HAWK system, comparing the system's standard HP-MPI library with MVAPICH2. Our findings revealed a notable performance boost when using MVAPICH2, attributable to its optimized non-blocking peer-to-peer operations, which are extensively used throughout the execution of our software.

The picture can't be displayed.