Accelerate Geostatistical Modeling with Mixed-Precision Cholesky: A High-Productivity Approach with PaRSEC

Yu Pei
SIAM CSE
March 1st 2021

In collaboration with KAUST

Qinglei Cao, Yu Pei, George Bosilca, Jack Dongarra from UTK
Sameh Abdullah, Hatem Ltaief, Marc Genton, David Keyes, Ying Sun from KAUST
Motivations and Contributions

- Environmental data from climate and weather applications in some region. E.g. soil moisture, temperature, humidity
- Modeled as a realization of Gaussian spatial random field
- For any two locations $s$ and $s'$ in a region, define covariance function: $C(|s - s'|, \theta)$
- Usually estimate $\theta$ by maximizing log-likelihood function:

$$\ell(\theta) = -\frac{n}{2} \log(2\pi) - \frac{1}{2} \log |\Sigma(\theta)| - \frac{1}{2} Z^\top \Sigma(\theta)^{-1} Z.$$ 

- With estimated $\theta$ we then can predict measurements at other locations
- ExaGeoStat is such a framework that can generate synthetic data, evaluate likelihood function and predict new values
- Generation/real data, optimization, and prediction
- One of the main computation kernels is the evaluation of Gaussian log-likelihood function, which translates into solving a symmetric positive definite covariance matrix (Cholesky).

Sample measurements from Midwest USA
Motivations and Contributions

The main contributions are

- extending ExaGeoStat in PaRSEC runtime and expediting Cholesky factorization with multi-precisions, double-, single- and half-precisions;
- optimizing the performance of proposed mixed-precision Cholesky by shepherding the task execution order and balancing the GPU workloads;
- validating accuracy via synthetic datasets and real datasets; and
- performing large-scale proposed mixed-precision Cholesky factorization on AMD-based, Intel-based and IBM-based multi-GPU systems with up to 196,608 cores, 131,072 cores and 768 GPUs respectively.
Mixed-precision Cholesky

- Constructing a corresponding covariance matrix $\Sigma(\theta)$ for a set of given locations in MLE modeling or prediction operations requires defining a covariance function to describe the correlation over a given distance matrix.

- The resulting Cholesky factorization takes advantage of the covariance matrix structure, which under a proper ordering [4] clusters the most significant information around the diagonal.

- BAND_SIZE_DP and BAND_SIZE_SP (the number of bands/sub-diagonals) are utilized to control the floating-point format for each tile in terms of memory storage and the operated numerical kernel.

- Data flows causing communication:
  - POTRF broadcasts to TRSM
  - TRSM broadcasts to GEMM in a row
  - TRSM broadcasts to GEMM in a column
  - TRSM send to SYRK

- The datatype of the data encapsulated in a data-flow is determined by the sender side.
Mixed-precision Cholesky

- PaRSEC supports multiple languages, including PTG, DTD and TTG, here we used to parameterize task graph (PTG) approach

- PaRSEC separates (1) data and data distribution, (2) flow and kernel (3) CPU/GPU device selection

- Flows (JDF) are the same as dense Cholesky; changes are only the data encapsulated in the flow and numerical kernels

Features we used in PTG code to enable mixed precisions and optimizations while keeping the same execution graph: addition of control flow, heterogeneous devices managed, data transfer type/shape/size specification
Runtime Optimizations

- Because of the mixture of three precisions in a Cholesky factorization, imbalance happens in terms of computation and communication.
  - **Computation.** The hardware’s computational capacity for different precisions varies, resulting in time-to-solution discrepancies between D|S|HGEMM, where the most operations come from in Cholesky factorization, leading to compute imbalance in tasks.
  - **Communication:** data stored in double-precision send $b^2 \times \text{sizeof (double)}$, $b$ is the tile size. While data stored in single-precision send $b^2 \times \text{sizeof (float)}$. 
Lookahead To Follow The Critical Path

- Possibility of tasks operating on tiles far away from the critical path may be scheduled first, e.g., tiles with magenta boundary.
- The concept of control dependency between tasks in PaRSEC, which guides the task execution order and priorities by adding an empty dependency (without data encapsulated) and extends the “lookahead” technology.
- In panel factorization k, control dependencies are applied between the top DGEMM and TRSMs with \(m - k > \text{lookahead}\) in the same panel factorization.
- Expediting the discovery of tasks on/near the critical path, and enough workloads could be guaranteed.
GPU Load Balancing

- Data transfers between devices, i.e., CPU and GPU here, along with execution are automatically managed by the runtime system.
- Advanced usages for users to predefine the data’s residence on GPU to relieves the runtime’s burden.
- GPU_ID = (n × NT + m)%g, where (m, n) is the tile index, NT is the number of tiles in a dimension, and g is the number of GPUs in a process.

- Tasks that operate in higher precision and near/on the critical path will play a crucial role especially when the majority portion is in half-precision.
- To address this imbalance, GPU ID for tile operating in half-precision is still determined by the formula above, while GPU ID for tile (m, n) in double- or single- precision is defined differently by ID = (m/P)%g, where P is the row process grid size and g is the number of GPUs in a process.

Both process placement and GPU placement are in 2 dim block cyclic.

Replace critical tasks GPU placement with user defined placement.
Performance Settings

- Three different HPC clusters with various kinds of architectures are used to evaluate efficiency.
  - Shaheen II at KAUST: an Intel-based Cray XC40 system with 6,174 compute nodes, each of which has two 16-core Intel Haswell CPUs running at 2.30 GHz and 128 GB of main memory.
  - HAWK at HLRS: an AMD-based system with 5,632 compute nodes, each of which has two 64-core AMD EPYC 7742 CPUs running at 2.25 GHz and 256 GB of main memory.
  - Summit at ORNL: an IBM-based system with 4,356 compute nodes, each of which has two 22-core Power9 CPUs running at 3.07 GHz and 256 GB of main memory, and each CPU is deployed with three NVIDIA Tesla V100 GPUs.
Correctness: Synthetic Datasets

We fix the variance parameter ($\theta_0$) to 1.5 and we use two levels of smoothness ($\theta_2$), 0.6 (rough field), and 1.5 (smooth field). We use the rough field with the three correlation lengths and give one example of smooth and strong correlated data (The four columns respectively)

one may experience accuracy loss with highly correlated data when using lower precisions. Moreover, when comparing the 3rd/4th columns with rough / smooth fields and strong correlations, smooth fields seem to require higher precision accuracy to properly estimates the model parameters.

Use Monte Carlo simulations (100 samples) on 2D 40K locations to show the impact of changing the precision of the covariance matrix using the proposed three precisions approach.
Correctness: Real Datasets

1M 2D soil moisture data at the top layer of the Mississippi River basin, U.S.

116K 2D Spatial images of the wind speed dataset in the Arabian Sea
**Correctness: Real Datasets**

1M 2D Soil Moisture Dataset: The estimation of the model parameters are close to the pure DP MLE, except for the 1D:99H variant. We observe that this dataset has medium correlated data with an average smooth field. This corroborates the analysis made with synthetic datasets that concludes on the effectiveness of the mixed-precision MLE for such data characteristics.

<table>
<thead>
<tr>
<th>Variant</th>
<th>Variance($\theta_1$)</th>
<th>Range ($\theta_2$)</th>
<th>Smoothness ($\theta_3$)</th>
<th>Log-Likelihood (llh)</th>
<th>MSE</th>
<th>Prediction Uncertainty</th>
<th>Iterations</th>
</tr>
</thead>
<tbody>
<tr>
<td>DP100%</td>
<td>0.7223</td>
<td>0.0933</td>
<td>0.9983</td>
<td>-59740.65974</td>
<td>0.044926</td>
<td>4.734439e+03</td>
<td>180</td>
</tr>
<tr>
<td>DP30%SP70%</td>
<td>0.7230</td>
<td>0.0935</td>
<td>0.9982</td>
<td>-59740.66579</td>
<td>0.044919</td>
<td>4.741754e+02</td>
<td>206</td>
</tr>
<tr>
<td>DP10%SP90%</td>
<td>0.7314</td>
<td>0.0953</td>
<td>0.9969</td>
<td>-59741.37532</td>
<td>0.044933</td>
<td>4.736149e+03</td>
<td>207</td>
</tr>
<tr>
<td>DP10%SP30%HP60%</td>
<td>0.7229</td>
<td>0.0936</td>
<td>0.9982</td>
<td>-59740.6552</td>
<td>0.044927</td>
<td>4.734435e+03</td>
<td>244</td>
</tr>
<tr>
<td>DP10%SP10%HP80%</td>
<td>0.7328</td>
<td>0.0947</td>
<td>0.9983</td>
<td>-59741.03423</td>
<td>0.044926</td>
<td>4.733337e+03</td>
<td>207</td>
</tr>
<tr>
<td>DP5%SP5%HP90%</td>
<td>0.7106</td>
<td>0.0927</td>
<td>0.9967</td>
<td>-59741.35348</td>
<td>0.044935</td>
<td>4.736572e+03</td>
<td>204</td>
</tr>
<tr>
<td>DP1%HP99%</td>
<td>0.9330</td>
<td>0.1286</td>
<td>0.9863</td>
<td>-59867.53239</td>
<td>0.044980</td>
<td>4.750953e+03</td>
<td>159</td>
</tr>
</tbody>
</table>

116K 2D Wind Speed Dataset: This dataset comes from a highly smooth field ($\theta_2$). Thus, the estimation of the model parameters is impacted starting from the first mixed-precision 30D:70S variant and further deteriorates with lower precision configurations.

<table>
<thead>
<tr>
<th>Variant</th>
<th>Variance($\theta_1$)</th>
<th>Range ($\theta_2$)</th>
<th>Smoothness ($\theta_3$)</th>
<th>Log-Likelihood (llh)</th>
<th>MSE</th>
<th>Prediction Uncertainty</th>
<th>Iterations</th>
</tr>
</thead>
<tbody>
<tr>
<td>DP100%</td>
<td>0.8407</td>
<td>0.0751</td>
<td>1.9905</td>
<td>241480.9994</td>
<td>1.752914E-02</td>
<td>2.2855E+00</td>
<td>666</td>
</tr>
<tr>
<td>DP30%SP70%</td>
<td>0.9925</td>
<td>0.1794</td>
<td>1.9757</td>
<td>239908.9983</td>
<td>1.766191E-02</td>
<td>2.9221E+00</td>
<td>108</td>
</tr>
<tr>
<td>DP10%SP90%</td>
<td>0.9924</td>
<td>0.1794</td>
<td>1.9757</td>
<td>239908.1004</td>
<td>1.766194E-02</td>
<td>2.9170E+00</td>
<td>91</td>
</tr>
<tr>
<td>DP10/SP50/HP40</td>
<td>0.9911</td>
<td>0.1810</td>
<td>1.9754</td>
<td>239884.4173</td>
<td>1.766318E-02</td>
<td>2.9313E+00</td>
<td>89</td>
</tr>
<tr>
<td>DP10/SP30/HP60</td>
<td>0.9761</td>
<td>0.1804</td>
<td>1.9576</td>
<td>232783.9932</td>
<td>1.765651E-02</td>
<td>5.2836E+00</td>
<td>94</td>
</tr>
<tr>
<td>DP10/SP10/HP80</td>
<td>0.9774</td>
<td>0.1802</td>
<td>1.9588</td>
<td>233438.8691</td>
<td>1.765624E-02</td>
<td>5.1232E+00</td>
<td>107</td>
</tr>
</tbody>
</table>
Comparison With State-of-the-art

Performance on shared memory of four V100 GPUs in [1]

Strong scalability on distributed memory of matrix 640K X 640K in [2]

---


Performance Evaluation at Scale

1536 nodes on HAWK

4096 nodes on Shaheen II

Shaheen II: about 3.2 Pflop/s for DP100 which is about 88% of Linpack performance, and constantly about 1.56× speedup of DP10SP90 to DP100 and 2.05× speedup of SP100 to DP100 when matrix size is bigger than 2.4M.
Performance Evaluation at Scale

Efficiency: DP100 is about 82.8% of Linpack performance.

Scalability: performance is doubled on 128 nodes compared to 64 nodes (not shown)

Mixed-precision effect: speedup to DP100 has positive correlations with matrix size and percentage of lower precision (SP and HP) with up to 2.7× speedup.

Optimization's effect
Conclusion and Future Work

• Extended ExaGeoStat in PaRSEC runtime system and expedited the most expensive operation in terms of computation and memory footprint, Cholesky factorization, by mixing double-, single- and half-precision.

• While maintaining the application-expected accuracy on synthetic and real datasets, implementation along with runtime-level optimizations showed great efficiency and scalability on multiple architectures.

• Investigate performance scalability issue when scale to more than 2k nodes on Summit (12000+ GPUs)

• Fine-grained tile-based decision instead of band-based for the precision format.

• Combine this approach with low-rank approximation technique to further speed up the workload