ETH zürich High Performance Computing for Science and Engineering II

P. Hadjidoukas, C. Papadimitriou
ETH Zentrum, HG E5/G3
CH-8092 Zürich

Spring semester 2017

Set 4 - Data Inference, MPI Datatypes

Issued: March 20, 2017

Question 1: Using MKL and DAAL for data inference

Intel® Math Kernel Library is a collection of highly optimized mathematical routines for the Intel architecture, while Intel ® Data Analytics Acceleration Library focuses on algorithms specific to data science.

If you are using an Intel architecture on this platform, you can get a student licensed software called Intel Parallel Studio[1], which includes both libraries, as well as the powerful `icc` compiler. Otherwise, use Euler, and run before doing anything else: `module load intel/16.0.0`.

In this exercise, we will build an artificial data set using a vector $\beta \in \mathbb{R}^p$ of our choosing. We will generate N random points $x_i \in \mathbb{R}^p$ and calculate y_i by perturbing the linear relationship with Gaussian noise $\varepsilon_i \sim \mathcal{N}(0,1)$:

$$y_i = \mathbf{x}_i^\top \boldsymbol{\beta} + \varepsilon_i. \tag{1}$$

We have already seen that the maximum likelihood estimate of $\boldsymbol{\beta}$ is the least squares solution:

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y}, \tag{2}$$

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_1^\top \\ \vdots \\ \mathbf{x}_N^\top \end{bmatrix}, \ \mathbf{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_N \end{bmatrix}. \tag{3}$$

While mathematically the problem is solved, it requires some computational care. Direct solving might cause issues, see for example the Läuchli matrix in [2, p. 239]. We instead use methods which exploit the structure of the problem beyond merely treating it as a general system of linear equations.

a) QR decomposition

Any matrix $\mathbf{A} \in \mathbb{R}^{m \times n}, m > n$ can be decomposed in the following way: [1, p. 82]:

$$\mathbf{A} = \mathbf{QR}, \ \mathbf{Q} \in \mathbb{R}^{m \times n}, \ \mathbf{R} \in \mathbb{R}^{n \times n}, \tag{4}$$

$$\mathbf{Q}^\top \mathbf{Q} = \mathbf{I}, \tag{5}$$

and $\mathbf{R}$ is an upper triangular matrix.

i) Show that $\hat{\boldsymbol{\beta}} = \mathbf{R}^{-1} \mathbf{Q}^\top \mathbf{y}$

ii) Consult the documentation for Intel MKL and list the necessary routines. Note that you should not explicitly store $\mathbf{R}^{-1}$ or $\mathbf{Q}^\top$.

[1] https://software.intel.com/en-us/qualify-for-free-software/student

iii) Implement the method using the MKL routines. Run multiple experiments:
- Validate your kernel by comparing it to the direct least squares solver `LAPACKE_dgels`. How does the error $||\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}||$ behave, $\hat{\boldsymbol{\beta}}$ the LS estimate and $\boldsymbol{\beta}$ the true parameter? Let $N = 2^1, \ldots, 2^{15}$.
- Measure the computational runtime $N = 2^1, \ldots 2^K$. Set K initially to 15 but try increasing it to see how high you can go, before experiencing any issues, if any.
- Investigate if the runtime decreases if you allow multiple threads – that is, investigate what options exist for multi-threading.

Pick $p = 30$.

Question 2: MPI Diffusion

If we want to spread the heat beyond a single node we need message passing. In this exercise we will parallelize the 2D diffusion equation with MPI.

a) Parallelize the serial 2D diffusion code found in `q2/diffusion2d_serial.cpp` with MPI. Divide the 2D domain with an MPI cartesian grid and use appropriate MPI datatypes for the data transfer.

b) Estimate (on paper) the communication overhead of the 2D grid decomposition compared to the horizontal strips that we used in the previous semester. Which one achieve better scaling?

c) Make a strong scaling plot up to 48 cores.

References

[1] A. Quarteroni, R. Sacco, and F. Saleri. *Numerical Mathematics (Texts in Applied Mathematics)*. Springer, 2010.

[2] J. Stoer. *Introduction to Numerical Analysis*. Springer, New York, 2002.