Department of Information and Computer Science
Utrecht University

INFOB3CC: Data Parallelism

Trevor L. McDonell

January 18, 2024

Introduction

Use these tasks to practice the topics of the lectures. You may have to do some research to read up on terms or topics not (yet) covered in the lectures.

Questions

1. What is data parallelism? How does it differ from task parallelism?
2. Algorithmic skeleletons
   (a) What is the purpose of using programming patterns—or algorithmic skeletons—to talk about our code?
3. In the lecture we discussed the stencil operation and separable filters. To gain some more understanding on the topic here is some additional reading material:
   - Computerphile: Separable Filters and a Bauble
     `https://edu.nl/8kjc6`

   (a) What is a convolution? How can they be implemented in terms of the stencil operator?
   (b) What does it mean to implement the Gaussian blur as a separable filter? Why do we do this?
   (c) When can other stencil operators be implemented as separable filters?
   (d) Explain why implementing the `stencil` operator in-place is difficult. Are there ways in which we can work around this limitation?
   (e) In a stencil computation, what is the *halo* region? How can the halo region be used when computing a stencil over multiple processors? In particular, think about if each processor has their own memory region which is not shared with the other processors.
   (f) What is the tiling optimisation in the `stencil` pattern?
4. Data parallelism
   (a) What is the `map` pattern?
   (b) Give an example operation which can be implemented using `map`, and an example which can not.
   (c) How does `map` differ from the `stencil` operation?
   (d) Are the `map` and/or `stencil` operations embarrassingly parallel? Why?
5. Data motion and layout
   (a) What is the difference between the forward and backward permutation patterns?
   (b) For efficient code execution, the layout of data in memory must be considered. What is the difference between the AoS and SoA representations?
   (c) Complex numbers are typically stored in the AoS representation. Give an example computation where this is a good representation, and an example where this is not the ideal representation.

6. Parallel scan
   (a) What is the scan operation? What is the difference between the inclusive and exclusive scan?
   (b) How would you implement a parallel scan operation using multiple processors?

7. Matrix-vector multiplication
   (a) Matrix-vector multiplication is an important operation in many physics and engineering applications. For example, a point p in three-dimensional space can be transformed into a different basis by multiplying by a matrix $\mathbf{A}$, to yield a new point $p' = \mathbf{A}y$, defined as:

$$\begin{bmatrix} ax + by + cz \\ dx + ey + fz \\ gx + hy + iz \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

   Describe how this operation could be implemented in terms of the data-parallel combinators we have discussed in class.
   (b) What is *nested* data parallelism? Is your solution to previous question expressed as a nested or flat data-parallel style? Write it using only flat data-parallelism (in Accelerate).
   (c) A sparse matrix is one in which only the non-zero elements of the matrix are stored. Sparse matrices appear in many applications; see the following site for same example datasets, along with some very nice visualisation: `https://sparse.tamu.edu/about`

   For example, the Hardesty3 dataset, which arose out of a computer vision application, consists of only 0.000065% non-zero entries. To store every value explicitly would require around 250 terabytes, but the non-zero values require only 160 megabytes.

   In the lectures we described a method for computing a sparse-matrix vector multiply. Suppose we want to parallelise this operation by having threads individually compute each value of the output; that is, thread one computes the first element of the result vector, thread two computes the second element, et cetera. How do we determine the range of values in the sparse matrix each thread should operate over?

8. Segmented operations
   (a) A segmented `scan` can be defined in terms of a regular (non-segmented) `scan`, by the use of an operator lifting function $\oplus^s$. Anneke wishes to implement the segmented plus operator as follows:

$$(f_x, x) \ +^s \ (f_y, y) = \text{if } f_y \text{ then } (true, y) \text{ else } (false, x + y)$$

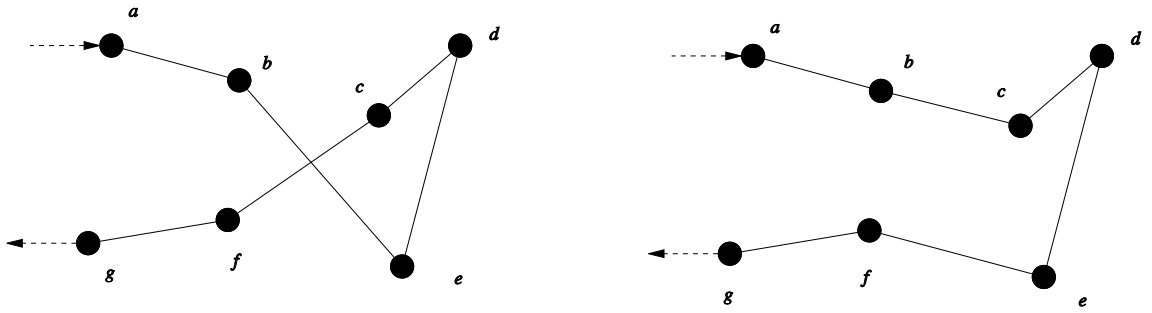
   where x and y are the array values and f_x and f_y are the corresponding flag values. The operator to the `scan` function should satisfy some mathematical property or properties. Which property/properties does/do not hold for this implementation, but should?
   (b) Explain (or draw) a parallel execution of (segmented) `scan` using Anneke's operator, which demonstrates that the operator does not work correctly.
   (c) Can you construct a segmented version of the `map` operator, using only a single regular (non-segmented) `map`? Motivate your response.
   (d) Can you construct a segmented version of the `fold` operator, using only a single regular (non-segmented) `fold`? Motivate your response.
   (e) Run length encoding is a form of lossless data compression in which sequences of the same data value (the runs) are stored as a single value together with the count of how many times that value appeared. For example, the string `MMMOOOOOOOOOOWWWWWWOOOOWWW` can be encoded as the array `[(3,'M'), (9,'O'), (6,'W'), (4,'O'), (3,'W')]`.

   Using the parallel array functions discussed in the course, provide an implementation of how a run-length encoded string, encoded in an array of (count, value) pairs as shown above, can be decoded into the full uncompressed string. A high-level overview or pseudocode implementation is sufficient. Describe your solution, and include the operator used with the array function(s) of your answer.

9. In this question we consider the schedule of a truck which delivers parcels. The route is stored in an array, where the first element of the array is the parcel to deliver first, the second element is the parcel to deliver second, and so on.

   In the following questions consider which of the parallel array combinators discussed in the course can be used to implement the described functionality.

   (a) Some routes can be optimised using the procedure 2-Opt. This method flips the delivery order for part of the route. For example, consider the following delivery schedule:



   The route on the left can be optimised using 2-Opt to the one on the right, that is:

$$[a, b, e, d, c, f, g] \xrightarrow{2-\text{Opt}} [a, b, c, d, e, f, g]$$

   If the start and end indices of the part of the route to flip are given (in the example above, the indices for c and e), which function can be used to perform 2-Opt? Two answers are possible; select them both!

   A. `fold`
   B. `gather`
   C. `map`
   D. `scanl`
   E. `scanr`
   F. `scatter`
   G. `stencil`
   H. `zipWith`

   (b) Is there a reason to prefer one of the approaches in the previous question to the other? Motivate your answer.

   (c) Using the function `drivingTime :: Parcel -> Parcel -> Time` we can compute the time to drive between any two parcel delivery addresses.[1] Which of the following operations can be used to compute an array where each element contains the driving time from the previous address? For example, the fourth element of the array will contain the driving time from the third parcel delivery point to the fourth delivery point. Select the correct response.

   A. `fold`
   B. `gather`
   C. `map`
   D. `scanl`
   E. `scanr`
   F. `scatter`
   G. `stencil`
   H. `zipWith`

   (d) It is important for the driver to know how long the deliveries will take. Given an array containing the time to drive between each parcel delivery address, what is the most efficient way to compute the total driving time for the route? Select the correct response.

[1] In the Accelerate library, as used in the third practical, the function `drivingTime` would have an `Exp` type; we ignore this detail for brevity, here and through the remainder of the paper.

A. `fold`
B. `gather`
C. `map`
D. `scanl`
E. `scanr`
F. `scatter`
G. `stencil`
H. `zipWith`

(e) We want to inform clients of the expected time at which their parcel will be delivered. Given an array containing the time to drive between each parcel delivery address, how do we compute the expected delivery time of each parcel? Select the correct response.

A. `fold`
B. `gather`
C. `map`
D. `scanl`
E. `scanr`
F. `scatter`
G. `stencil`
H. `zipWith`

(f) As the parcels can be very heavy, the total mass of the delivery truck should be taken into account when computing the driving time between delivery locations. Given the total mass of all of the parcels in the delivery truck, the function `slowdownFactor :: Mass -> Double` returns a multiplicative factor which determines how much slower the delivery truck is compared to an empty truck. For example, a value of 1.0 is the normal driving speed, while a value of 2.0 means the delivery will take twice as long (the truck must drive at half the speed due to the extra mass). After each parcel is delivered, the truck is lighter, and thus for subsequent deliveries can drive faster. Taking this new information into account, how do we compute:

i. The expected delivery time for each parcel; and
ii. The total time to deliver all parcels on the route?

Provide an implementation using the array functions discussed in the course (`fold`, `gather`, `map`, `scanl`, `scanr`, `scatter`, `stencil` and `zipWith`). A high-level overview or pseudocode implementation is sufficient. You may assume a function `parcelMass :: Parcel -> Mass`, which gives the mass of each parcel. Describe your solution, and include the operator used with the array function(s) of your answer.

10. A furniture shop has developed an application to manage the stock in their warehouse. Each product consists of one or more parts. A part may also be used for different products. For instance, different tables may use the same legs combined with a different table top.

    To further increase the capacity of the system, the furniture shop considers to use data parallelism to process orders in bulk. Instead of checking the stock level for each product one at a time, the system will now check whether every part for every product of the entire order is in stock at once.

    In the following questions answer using the data parallel operations discussed in the course (`map`, `zipWith`, `fold`, `scanl`, `scanr`, `permute`, `backpermute`, and `stencil`). Nested data parallelism is not allowed. Write code in Haskell.

    (a) An order is given as an array of products. Write a function which computes an array of the parts required to fulfil the order.

    You may use $n(i)$ to denote the number of parts required for product i, and $f(i, k)$ to denote the k-th part of item i, where $0 \leq k < n(i)$.

    (b) Given an array of parts required for an order, we need to compute the total number of each part which is required. Write a function which computes an array such that the value at index i of the array contains the count of the number of parts of kind i required for this order.

    (c) Given the output of the previous question, and an array containing the current stock level for every part in the warehouse, where both are represented as an array where the i-th element denotes the count for parts of kind i. Write a function which determines whether everything for the order is in stock.

(d) If all of the parts are in stock for the order, how can we compute the new stock levels, after reserving the parts required for this order?

11. The scale and connectivity of the global air travel network increases the risk for infectious diseases to spread. Epidemic control measures can be applied to air travel networks to minimise the risk of large-scale contagion. In order to design the most effective outbreak control measures, we must build a model of the dynamics of infection which can then be used to evaluate the impact of various outbreak control policies.

We will use an SIR model[2] to represent the evolution of an outbreak in a given population over time, using a set of differential equations specifying the proportion of the population in each possible state an individual can assume: susceptible (S), infected (I), and recovered (R). We have the following discrete form equations:

$$S_{i,t+1} = S_{i,t} - \frac{\beta_i I_{i,t} S_{i,t}}{N_{i,t}} + \sum_{j \in \theta(i)} S_{ji,t} - \sum_{j \in \theta(i)} S_{ij,t} \tag{12}$$

$$I_{i,t+1} = I_{i,t} + \frac{\beta_i I_{i,t} S_{i,t}}{N_{i,t}} - \gamma I_{i,t} + \sum_{j \in \theta(i)} I_{ji,t} - \sum_{j \in \theta(i)} I_{ij,t} \tag{13}$$

$$R_{i,t+1} = R_{i,t} + \gamma I_{i,t} + \sum_{j \in \theta(i)} R_{ji,t} - \sum_{j \in \theta(i)} R_{ij,t} \tag{14}$$

where $\theta(i)$ is the set of nodes (cities) which have a direct connection to node i, β is the probability of contact between a susceptible and an infected person causing infection, and γ the proportion of infected people recovering per day (the inverse of this value is the number of days it takes for an infected individual to recover). In this question we consider the spread of influenza, for which we can treat these last two terms as constants: $\beta = 0.15$ and $\gamma = 1/7$.

For example, equation (12) says that the number of susceptible individuals in city i at time $t+1$ is given by the number of susceptible individuals at time t (first term), minus the number of susceptible individuals which became infected through contact with an infected individual (second term), then modified by the number of susceptible individuals entering (third term) and leaving (fourth term) the city.

In the following questions you will sketch how this model can be implemented using the data parallel operations discussed in the course (`map`, `zipWith`, `fold`, `scanl`, `scanr`, `permute`, `backpermute`, and `stencil`). Nested data parallelism is not allowed. Write code in Haskell.

(a) At each time step t, the number of individuals in each category (S, I, and R) is modified by the rate at which individuals travel between cities. We can model this with an $N \times N$ adjacency matrix `travelFlows`, where the value at index `Z :. j :. i` represents the number of passengers travelling from city i to city j, where $0 \leq i < N$ and $0 \leq j < N$.

Give a function which computes in data parallel:

i. The total number of individuals departing each city

ii. The total number of individuals arriving at each city

(b) Given a vector N (the total population at each node i) and vectors S, I, and R containing the current number of individuals in each category respectively at time t, write a function to compute in data parallel each of the vectors S, I, and R at time $t + 1$.

(c) The *infection attack rate* is the percentage of the population which contracts the disease in an at-risk population during a specified time interval. Write a function to compute in data parallel the attack rate at each city for a given time step.

(d) In order to model the dynamics of the infection, we iteratively apply equations (12)–(14). You are given matrices S', I', and R' containing the number of individuals in each category for every time

[2] Chen, Nathan, Rey, David, and Gardner, Lauren. Multiscale Network Model for Evaluating Global Outbreak Control Strategies. In *Journal of the Transporation Research Board*, 2017. `http://dx.doi.org/10.3141/2626-06`

step of the simulation. Write a function to compute in data parallel the *peak prevalence* of the infection in each city.

(e) Infectious disease spread is an example of exponential growth because the number of cases on a given day is proportional to the number of cases in the previous day. However, exponential growth like this can not continue forever, and must start slowing down at some point. Given the matrices S', I', and R' from the previous question, write a function to compute in data parallel the inflection point of the curve, the time t at which the *growth factor* first drops to (at or below) one.