Implementing simplex on DFE

Jurij Mihelič, Uroš Čibej

August 27, 2015

Overview

- Definition of the problem
- Overview of the simplex algorithm
- Two implementations on the DFE
- Results

Problem definition

n variables

$$x = [x_1, x_2, \ldots, x_n]$$

These variables are bounded by a set of m constraints, given in matrix form as:

$$Ax \leq b.$$

Out of these feasible solutions, the ultimate goal is to find a vector x, which maximizes a linear goal function

$$cx.$$

Data layout

The input parameters are organized in a matrix called the simplex tableau in the following way:

-cx	c
b	A

Simplex revisited

Data: Simplex tableau T
while $\exists j : c[j] > 0$ **do**

$$j_p = \arg\max_j c[j];$$

$$i_p = \arg\min_{i, T[i][j_p] > 0} \frac{b[i]}{T[i][j_p]};$$

$$p = T[i_p][j_p];$$

$$row = T[i_p][*];$$

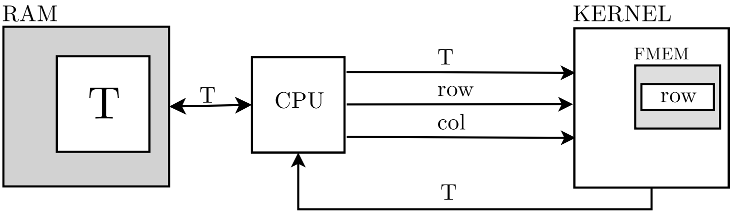
$$col = T[*][j_p];$$

$$T[i][j] = \begin{cases} \frac{1}{p} & i = i_p \wedge j = j_p \\ \frac{T[i][j]}{p} & i = i_p \\ -\frac{T[i][j]}{p} & j = j_p \\ T[i][j] - row[i] \times col[j] \times \frac{1}{p} & \text{otherwise} \end{cases};$$

end

Streaming from main memory

In the first implementation, the tableau is kept in the main memory and each iteration is streamed to the kernel.



Streaming from LMEM I.

Due to technological constraints, when streaming from LMEM, some changes need to be done. The CPU code is the following:

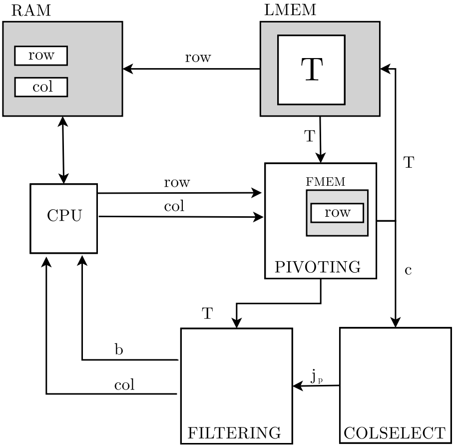
Data: Simplex tableau T

$col = select\_col(T)$;
$row = select\_row(T)$;
$copy\_to\_LMEM(T)$;
while $col[0] > 0$ **do**
    $pivoting\_DFE(col, row, new\_col)$;
    $select\_row(b, col)$;
    $row = copy\_row\_from\_LMEM()$;
    $col = new\_col$;
end

Streaming from LMEM II.

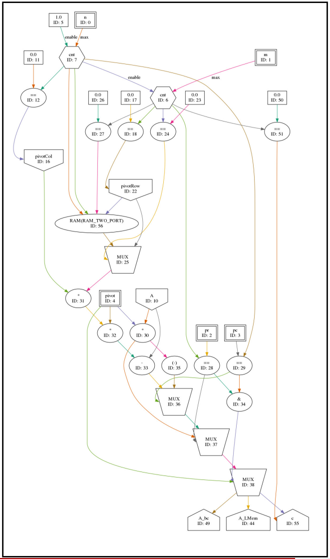
A different design is required to extract the column while performing the pivoting.

- Pivoting kernel (transformation of the matrix)
- Column select kernel (select the maximal c coefficient)
- Filtering kernel (extract the column from the matrix)

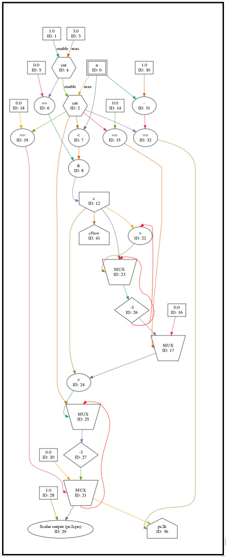


Kernels in more detail

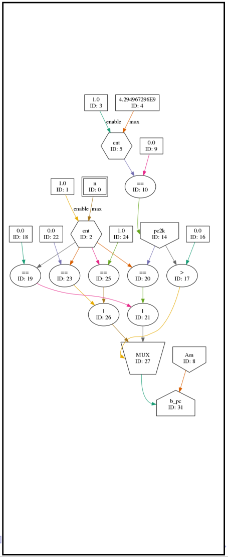
Pivoting kernel



Max. kernel

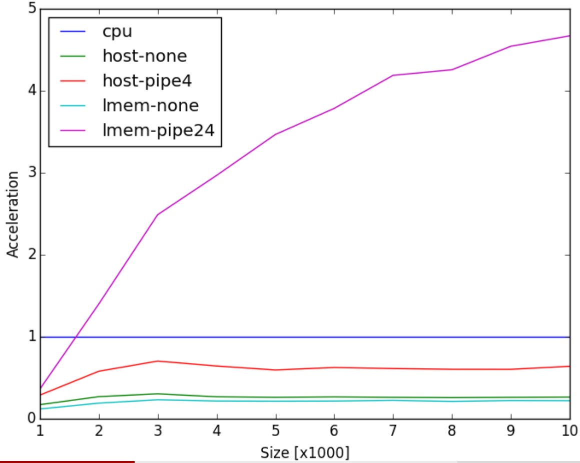


Filtering kernel



Results

The accelerations for different implementations, for prblem sizes from 10^3 to 10^4.



Conclusions and future work

- The results show a large improvement over the CPU implementation
- The bottleneck for both implementation is the bandwidth

$$(L)MEM \Longleftrightarrow DFE$$

- For dense linear programs it outperforms state-of-the-art solvers
- For very sparse linear programs new approaches need to be implemented