Paramathics Interview Task II

Martin D. Pham

January 2021

1 Problem

Given a sparse lower triangular matrix $L \in \mathbb{R}^{n \times n}$ and a vector $b \in \mathbb{R}^n$, we apply forward substitution to solve for the system:

$$Lx = b$$

Storing L in compressed column format allows a naive forward substitution implementation to skip iterating over zero-valued contributions given by the zero-valued entries of L.

We can further optimize the serial algorithm using the strategy shown in Sympiler. Steps are as follows: (1) construct adjacency graph $DG_L = (V, E)$ of L representing dependencies between columns in triangular solve; (2) using the sparsity of b, i.e. $\beta = \{i | b_i \neq 0\}$, perform depth-first search on DG_L starting from β to determine $Reach_L(\beta)$; (3) only the columns in $Reach_L(\beta)$ contribute to the non-zero RHS entries and so all other columns, i.e. $V \backslash Reach_L(\beta)$, can be skipped during iteration.

No parallel optimizations were successful. Some naive attempts at using reduction clause with + operator to sum entries but updating for triangular solve became an issue regarding untangling loop dependency when working in CCS format. Seems like motivation for decoupled symbolic analysis?

2 Results

OS	Ubuntu 18.04.5 LTS
Memory	31.2GiB
Processor	Intel® Core™ i7-8550U CPU @ 1.80GHz × 8
Compiler	g++ (Ubuntu 7.5.0-3ubuntu1 18.04) 7.5.0

Table 1: Machine details (System 76 Galago Pro 2018 Model)

Linear system	Naive compressed column	Adjacency graph	Speedup
torso1	0.057025	0.034461	x1.6548
TSOPF	0.223503	0.223382	x1.0005

Table 2: Average wall time in seconds using *omp_get_wtime* over 10 runs between naive forward substitution compared to adjacency graph optimization.

Linear system	$\|V\|$	$\|Reach_L(\beta)\|$	$\|Reach_L(\beta)\|/\|V\|$
torso1	116158	34314	0.30
TSOPF	35696	35414	0.99

Table 3: Total number of columns compared to number of contributing columns that need to be included in forward substitution. $torso1$ skips 70% of columns while $TSOPF$ only skips 1%.

The adjacency optimization improved performance for $torso1$ but not $TSOPF$. Looking at the relative sizes of $Reach_L(\beta)$ for both systems, we find that $torso1$ is able to skip a significant number of columns while $TSOPF$ barely benefits. The relative residuals in different norms also suggest that the matrix in $TSOPF$ is ill-conditioned.

References

Resources links attached to assignment.