

Assignment 4

MPI – Matrix Multiplication

Tammo Johannes Herbert (319391), Rico Jasper (319396), Erik Rudisch (343930)

19.06.2013

Exercise 1 – Going for Speed

TODO

Exercise 2 – Sparse Matrices

(a) Sparse Matrix Vector Multiplication

Wir summieren für jede Zeile der Eingabematrix A bzw. des Ausgabevektor c die Summe von Produkten typisch für eine Matrix-Vektor-Multiplikation. Jedoch werden pro Zeile nur die nicht-null Elemente der Matrix A mit dem Eingabevektor b multipliziert. Daher läuft die innere Schleife nur von `row_ptr[i]` nach `row_ptr[i+1]-1`. Die Spalte wird entsprechend über `col_ind[j]` bestimmt.

```
1  for i in 0 ... n-1                                // for each row
2      c[i] = 0
3      for j in row_ptr[i] ... row_ptr[i+1]-1        // for each column
4          c[i] += val[j] * b[col_ind[j]]
```

(b) Parallelization

Für eine Parallelisierung ist es wünschenswert, dass jede Recheneinheit möglichst dieselbe Menge an Arbeit verrichtet. Die Arbeit pro Zeile ist aber abhängig von der Anzahl der nicht-null Elementen und kann daher variieren.

Die zeilenweise parallelisierte Variante erfordert außerdem das replizieren des Eingabevektors b . Im Falle der CRS-Matrix A kann dieser Vektor allerdings in Relation jedoch groß ausfallen, da die Dichte von nicht-null Elementen recht dünn sein kann.

Hinsichtlich des ersten Problems kann man die Lasten versuchen auszugleichen. Beispielsweise könnten mehrere Zeilen einer Recheneinheit zugeteilt werden, um die Gesamtverteilung auszugleichen. Denkbar ist jedoch auch das Aufbrechen von „großen“ Zeilen in kleinere Stücke.

(c) Embedding into a FEM simulation

TODO