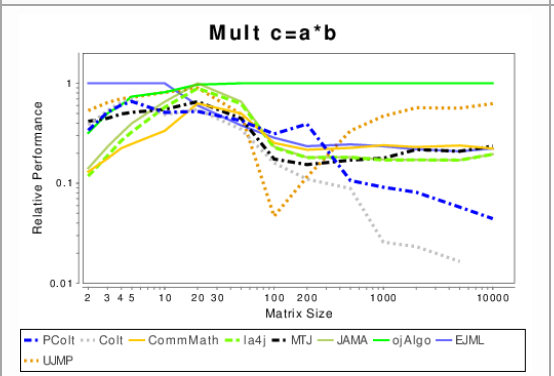
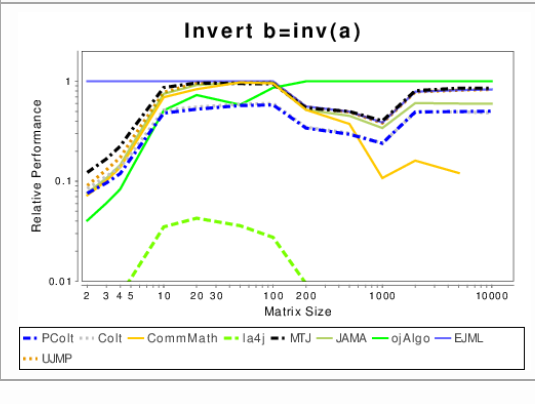
Java Matrix Libraries Benchmark <https://github.com/optimatika/ojAlgo/wiki/Java-Matrix-Benchmark>

For Simple Matrix(Dense)





**Here we are performed Fast Matrix Multiplication for sparse matrix using libraries that support sparse matrix format and stores and commute result in efficient way.**

For Elapsed time calculation we used java System.nanoTime().

<https://memorynotfound.com/calculating-elapsed-time-java/>

**Data(Sparse Matrixes)**

All are non-singular Matrix

Each of these matrix have only 10 to 40 non zero values

**Matrix 1** (10000 x 10000) : Values(0/1)

**Matrix 2** (5000 x 5000) : Values(0/1)

**Matrix 3** (10000 x 10000) : Values(0/1/2/3)

**Libraries**

<https://java-matrix.org/>

**Supporting Sparse Matrix**

Smile Math

La4j

Apache Commons Math

MTj

parallel colt

Jsci

Jscience

**OBSERVATIONS**

**Smile Matrix Library:**

* Have constructor of sparse matrix accepting double[][] 2D array
* Have various functions including multiply.
* Doesn’t have function to compute inverse for sparse or else we can use the Dense Matrix to compute inverse.

Matrix1: 81361327

Matrix2: 52352554

Matrix3: 125874595

**La4j:**

Initially

Matrix1: 1999267840

Matrix2:983641036

Matrix3:4615507452

After That

Out of Space

**Apache Common Math:**

* Only have interface for sparse Matrix
* Implemented using OpenMapRealMatrix

For Matrix2 12333114501

For Matrix 1 and matrix 3 Out of space

**Mtj**

* Have Sparse Matrix (CRM and CCM)
* Constructer accepting only integer 2D array.
* Doesn’t have inverse function.
* Using Linkedsparsematrix

For Matrix2 time in nanoseconds = 10619446

Matrix 1 and Matrix 3: Out of space

**Parallel Colt**

For Matrix 2:14960925878

For Matrix 1:Java heap space error