COP 701 Software System LAB 1

Lab1 Assignment

2020MCS2456 DIPIKA TANWAR

Objective:

Design an efficient Data Struecture to store FixPoint numbers

To Analyze Floating Point Vs Fix Point Performance and Accuracy

Design Lex and Yacc (Grammer) to Parse Matrix given in text files

Lex Parser:

This Parser Parse Text file, Basic Tokes are NUMBERS, and SYMBOL

```
[+-]?[0-9]*[.]?[0-9]+   {yylval.fnum=atof(yytext); return NUMBER;}
[,\[\]]                 {return yytext[0];}
[ \t]                   ;
[\n]                    return 0;
.                       printf(" invalid \n");
```

YACC Grammer

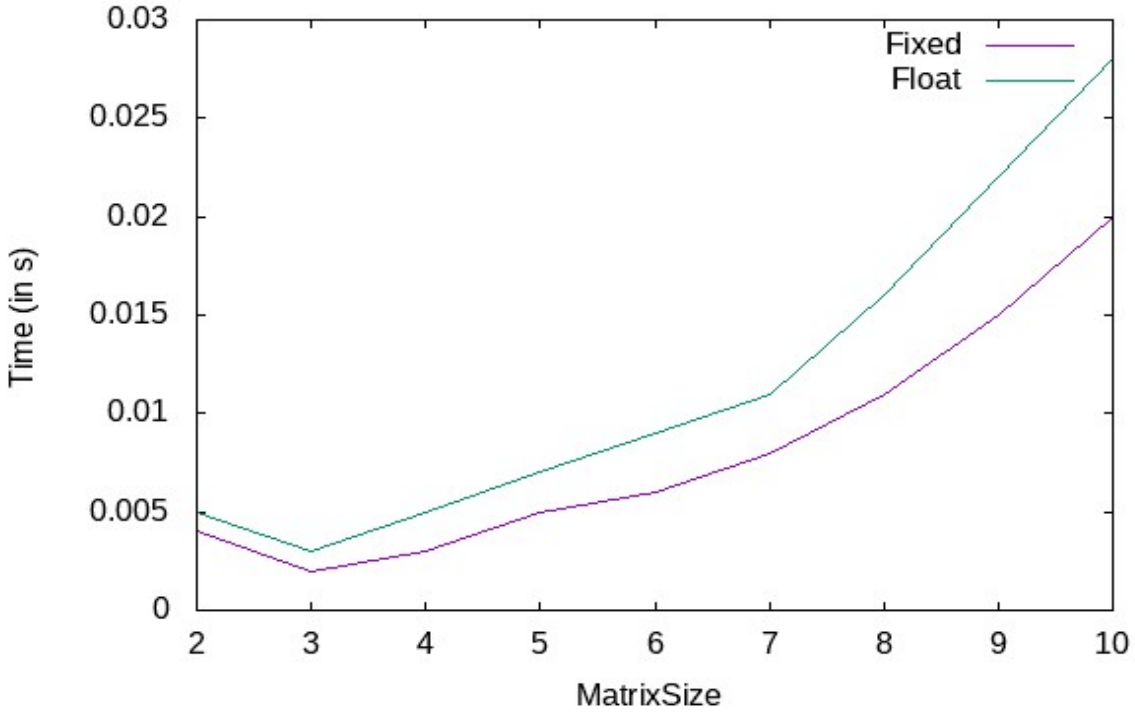
```
%start S
%union {float fnum;}
%token <fnum> NUMBER

%%

S       :matrix          {printf("File for Matrix Parse");}
        |line            {prinf("File For Exponential");}
        ;

line    :line NUMBER     {expo_pow = $2;}
        |NUMBER          {expo_base = $1;}
        ;

matrix  :'['arrays']'    {printf("\n matrix found\n");}
        ;

arrays  :array           {;}
        |arrays','array  {;}
        ;
array   :'['list']'      {m_row++;}
        ;

list    :list','NUMBER   { MATRIX[m_index++]=$3;}
        |NUMBER          { MATRIX[m_index++]=$1;}
        ;

%%
```

To Disable virtulization for Floating point in GCC use CFLAGS **-ffloat-store**

Performance Graph

Size vs Time graph for Matrix Multiplication

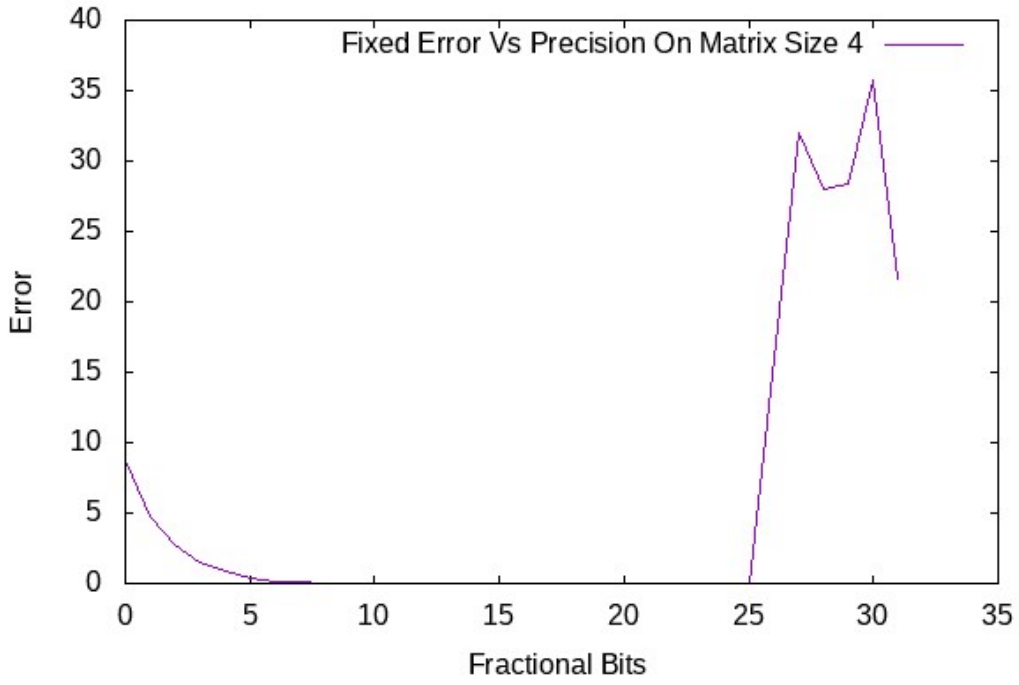
-For floating Point vectorization is disabled using -ffloat-store CFLAG



Precision Graph

To Show Presition, Mean Absolute Error metric is used

Mean absolute Error vs Frational Bits


Fixed Error Vs Precision On Matrix Size 4
Error
Fractional Bits
0
5
10
15
20
25
30
35
40