TAU + MPI

Profiling and Tracing the SUMMA Algorithm with TAU

What and Why?

Agenda

- Define Profiling vs Tracing
- SUMMA Review
  - The algorithm
  - Data movement + costs
- TAU + Application:
  - Compile
  - Profile
  - Trace

Why?

- Performance analysis is tedious:
  - Instrumenting code by hand is repetitive
  - Adding new function-calls to code requires re-instrumenting
  - Setting up profiling and tracing code takes time
- TAU:
  - Instruments our code automatically at compile time
  - Re-written code can be re-instrumented by re-compiling
  - Puts time to instrument code back in the programmer's hands

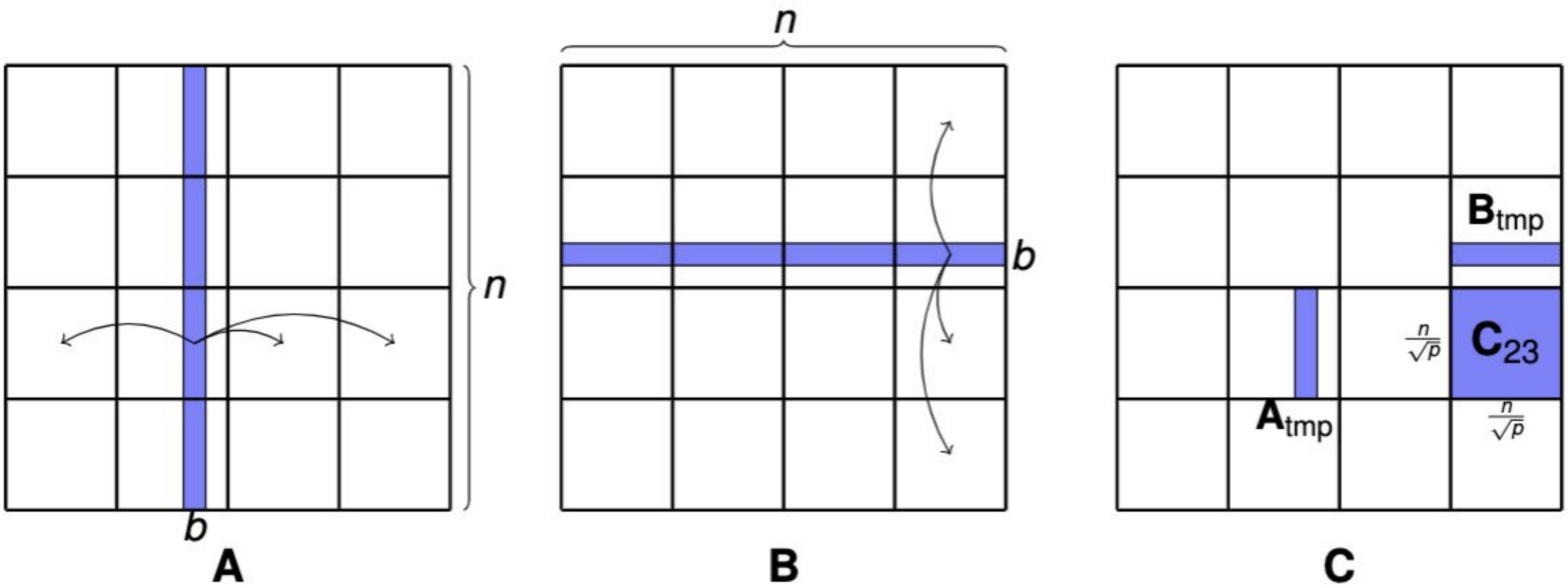
Profiling vs. Tracing

Profiling vs. Tracing

Profiling is about how much time/resources a given program and subdivisions of said program uses. It focuses on research usage.

Tracing is about the flow of control of a program and its subdivisions: what procedures are in control at any given time, when does context switching occur, and when do system events happen across processes.

SUMMA Review



SUMMA Communication

SUMMA Algorithm

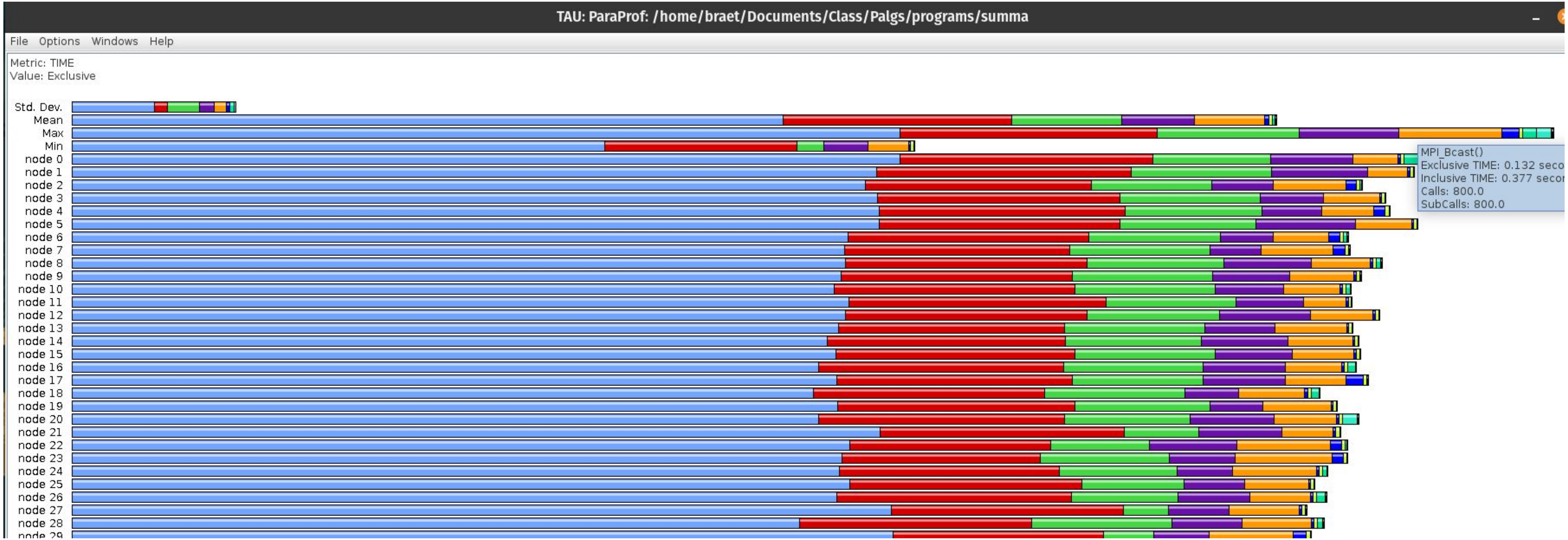
```
double* Alocal; // Column-wise random nxn matrix (local block of A)
double* Blocal; // Row-wise nxn identity matrix (local block of B)
double* Clocal; // Column wise nxn zero matrix (local block of product)
double* Atemp;  // (n/sqrt(p)) x b broadcast matrix for A
double* Btemp;  // b x (n/sqrt(p)) broadcast matrix for B
int mloc, nloc = n/sqrt(p); // size of local matrices

for (int i = 0; i < sqrt(p); i++) {
    for (int j = 0; j < m / (b*sqrt(p)); j++) {
        memcpy(Atemp, Alocal+b*j* mloc, b*mloc*sizeof(double));
        memcpy(Btemp, Blocal+b*j*nloc, b*nloc*sizeof(double));
        MPI Bcast(Atemp, b*mloc, MPI DOUBLE, i, row comm);
        MPI Bcast(Btemp, b*nloc, MPI DOUBLE, i, col_comm);
        local_gemm(Atemp, Btemp, Clocal, mloc, b);
    }
}
```

$$\gamma \cdot O\left(\frac{n^3}{p}\right) + \alpha \cdot O\left(\frac{n}{b}\log p\right) + \beta \cdot O\left(\frac{n^2}{\sqrt{p}}\right)$$

SUMMA Cost

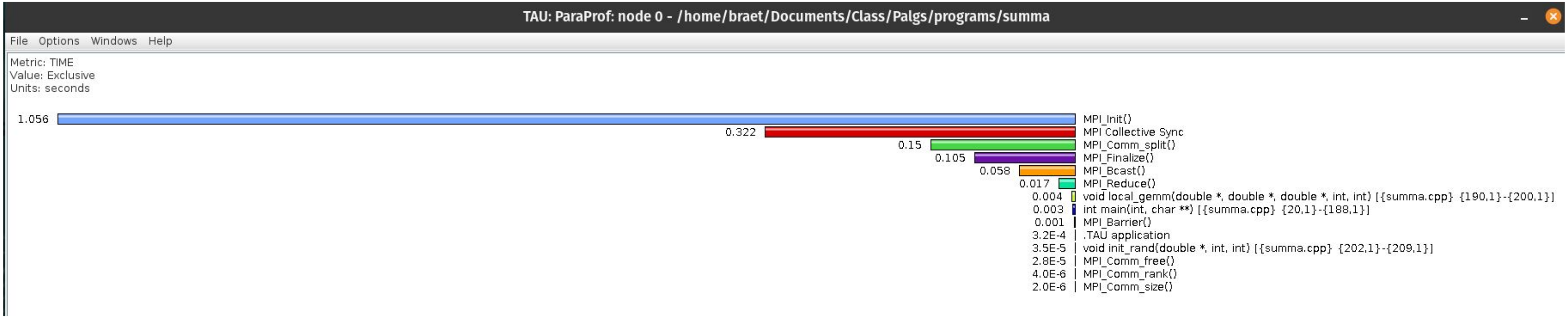
Compile + Profile + Trace

paraprof



TAU: ParaProf: /home/braet/Documents/Class/Palgs/programs/summa
File Options Windows Help
Metric: TIME
Value: Exclusive
Std. Dev.
Mean
Max
Min
node 0
node 1
node 2
node 3
node 4
node 5
node 6
node 7
node 8
node 9
node 10
node 11
node 12
node 13
node 14
node 15
node 16
node 17
node 18
node 19
node 20
node 21
node 22
node 23
node 24
node 25
node 26
node 27
node 28
MPI_Bcast()
Calls: 800.0
SubCalls: 800.0

paraprof


TAU: ParaProf: node 0 - /home/braet/Documents/Class/Palgs/programs/summa
File Options Windows Help
Metric: TIME
Value: Exclusive
Units: seconds
1.056 MPI_Init()
0.322 MPI Collective Sync
0.15 MPI_Comm_split()
0.105 MPI_Finalize()
0.058 MPI_Bcast()
0.017 MPI_Reduce()
0.004 void local_gemm(double *, double *, double *, int, int) [{summa.cpp} {190,1}-{200,1}]
0.003 int main(int, char **) [{summa.cpp} {20,1}-{188,1}]
0.001 MPI_Barrier()
3.2E-4 .TAU application
3.5E-5 void init_rand(double *, int, int) [{summa.cpp} {202,1}-{209,1}]
2.8E-5 MPI_Comm_free()
4.0E-6 MPI_Comm_rank()
2.0E-6 MPI_Comm_size()

pprof

```
FUNCTION SUMMARY (mean):
---------------------------------------------------------------------------------------
%Time    Exclusive    Inclusive       #Call      #Subrs  Inclusive Name
              msec   total msec                          usec/call
---------------------------------------------------------------------------------------
100.0        0.355        3,293           1           1    3293428 .TAU application
100.0           87        3,293           1        3013    3293073 int main(int, char **)
 59.7          468        1,964        2000        2000        982 MPI_Bcast()
 45.7        1,505        1,505        2001           0        752 MPI Collective Sync
 27.9          920          920           1           0     920400 MPI_Init()
  3.3          109          109           2           0      54883 MPI_Comm_split()
  3.0           98           98           1           0      98979 MPI_Finalize()
  2.7           88           88        1000           0         89 void local_gemm(double *,
  0.6           10           19           1           1      19782 MPI_Reduce()
  0.1            2            2           1           0       2173 MPI_Barrier()
  0.0         0.24         0.24           1           0        240 void init_rand(double *,
  0.0       0.0434       0.0434           2           0         22 MPI_Comm_free()
  0.0      0.00308      0.00308           3           0          1 MPI_Comm_rank()
  0.0      0.00152      0.00152           1           0          2 MPI_Comm_size()
➜  summa git:(main) ✗
```

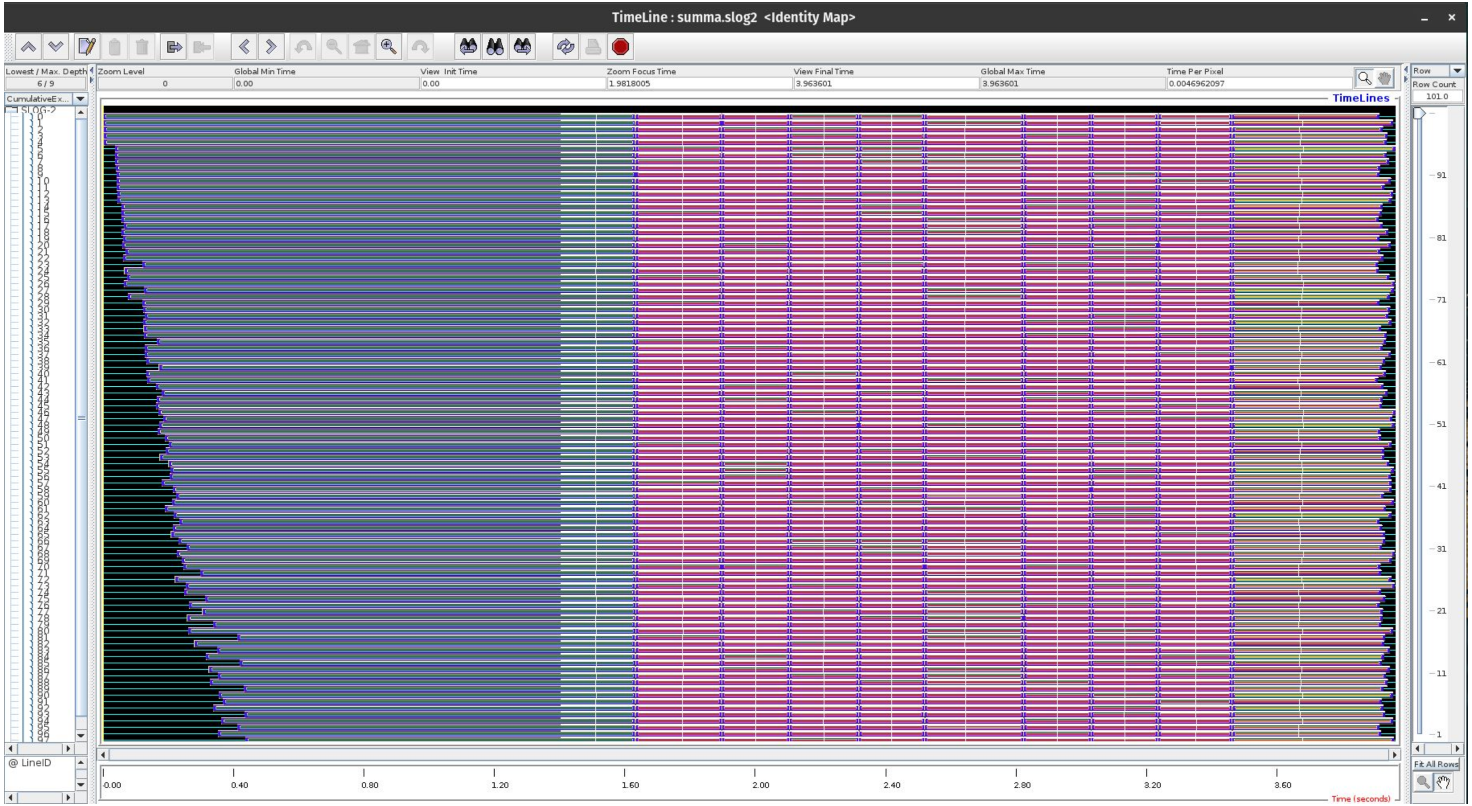
pprof

```
FUNCTION SUMMARY (mean):
---------------------------------------------------------------------------------------
%Time    Exclusive    Inclusive       #Call      #Subrs  Inclusive Name
              msec   total msec                          usec/call
---------------------------------------------------------------------------------------
100.0        0.245        1,926           1           1    1926183 .TAU application
100.0        0.938        1,925           1         313    1925937 int main(int, char **)
 49.8          958          958           1           0     958707 MPI_Init()
 36.8          184          708         200         200       3544 MPI_Bcast()
 27.3          526          526         201           0       2617 MPI Collective Sync
  5.9          113          113           2           0      56971 MPI_Comm_split()
  4.5           86           86           1           0      86533 MPI_Finalize()
  2.5           48           48         100           0        484 void local_gemm(double *
  0.3            5            5           1           0       5952 MPI_Barrier()
  0.1         0.22            2           1           1       2473 MPI_Reduce()
  0.0        0.258        0.258           1           0        258 void init_rand(double *,
  0.0       0.0362       0.0362           2           0         18 MPI_Comm_free()
  0.0      0.00327      0.00327           3           0          1 MPI_Comm_rank()
  0.0      0.00141      0.00141           1           0          1 MPI_Comm_size()
➜  summa git:(main) ✗
```

pprof

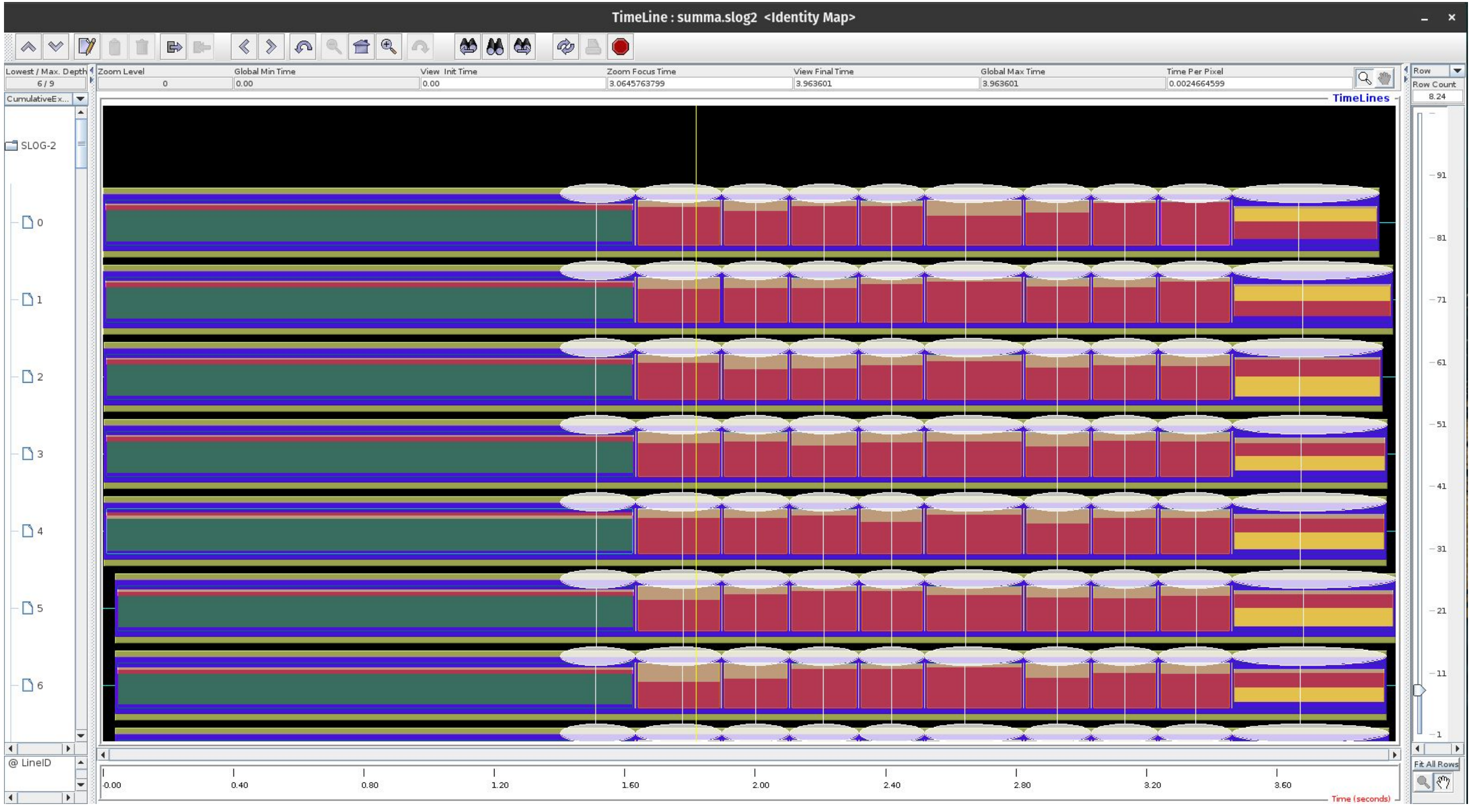
```
➜  summa git:(main) ✗ bash tau2csv.sh
1866,1865,849,778,586,89,87,49,7,2,0.249,0.0337,0.00328,0.00139
➜  summa git:(main) ✗
```

jumpshot



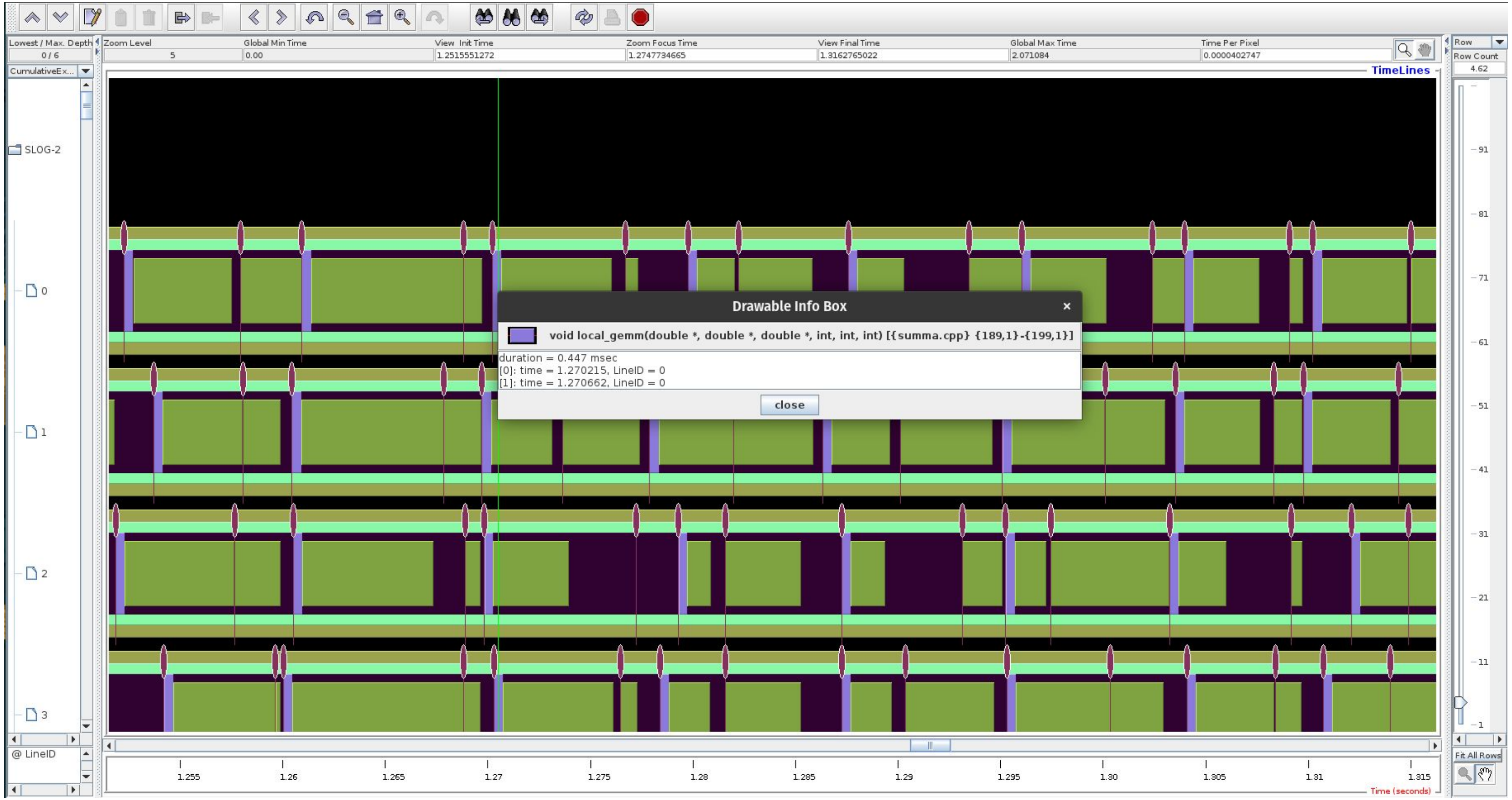
TimeLine : summa.slog2 <Identity Map>
Lowest / Max. Depth
6 / 9
Zoom Level
0
Global Min Time
0.00
View Init Time
0.00
Zoom Focus Time
1.9818005
View Final Time
3.963601
Global Max Time
3.963601
Time Per Pixel
0.0046962097
Row
Row Count
101.0
TimeLines
Fit All Rows
@ LineID
Time (seconds)

jumpshot



TimeLine : summa.slog2 <Identity Map>
Lowest / Max. Depth
6 / 9
Zoom Level
0
Global Min Time
0.00
View Init Time
0.00
Zoom Focus Time
3.0645763799
View Final Time
3.963601
Global Max Time
3.963601
Time Per Pixel
0.0024664599
Row
Row Count
8.24
TimeLines
SLOG-2
0
1
2
3
4
5
6
LineID
Fit All Rows
0.00
0.40
0.80
1.20
1.60
2.00
2.40
2.80
3.20
3.60
Time (seconds)

jumpshot



Lowest / Max. Depth
0 / 6
Zoom Level
5
Global Min Time
0.00
View Init Time
1.2515551272
Zoom Focus Time
1.2747734665
View Final Time
1.3162765022
Global Max Time
2.071084
Time Per Pixel
0.0000402747
Row
Row Count
4.62
TimeLines
SLOG-2
Drawable Info Box
void local_gemm(double *, double *, double *, int, int, int) [{summa.cpp} {189,1}-{199,1}]
duration = 0.447 msec
[0]: time = 1.270215, LineID = 0
[1]: time = 1.270662, LineID = 0
close
@ LineID
Fit All Rows
Time (seconds)

Q+A