**CPSC 501 Assignment 4 Report**

1. **Initial Base Program**

**Profiling-**

Flat profile:

Each sample counts as 0.01 seconds.

% cumulative self self total

time seconds seconds calls Ks/call Ks/call name

100.33 1784.78 1784.78 1 1.78 1.78 convolve(float\*, int, float\*, int, float\*, int)

0.00 1784.80 0.02 2 0.00 0.00 wavReader(float\*, char\*, int\*)

0.00 1784.82 0.02 1 0.00 0.00 scale(float\*, int)

0.00 1784.82 0.00 3 0.00 0.00 std::operator|(std::\_Ios\_Openmode, std::\_Ios\_Openmode)

0.00 1784.82 0.00 2 0.00 0.00 std::fpos<\_\_mbstate\_t>::fpos(long)

0.00 1784.82 0.00 1 0.00 0.00 \_GLOBAL\_\_sub\_I\_chunkId

0.00 1784.82 0.00 1 0.00 0.00 \_\_static\_initialization\_and\_destruction\_0(int, int)

0.00 1784.82 0.00 1 0.00 0.00 wavWriter(float\*, int, char\*)

Call graph (explanation follows)

granularity: each sample hit covers 2 byte(s) for 0.00% of 1784.82 seconds

index % time self children called name

<spontaneous>

[1] 100.0 0.00 1784.82 main [1]

1784.78 0.00 1/1 convolve(float\*, int, float\*, int, float\*, int) [2]

0.02 0.00 2/2 wavReader(float\*, char\*, int\*) [3]

0.02 0.00 1/1 scale(float\*, int) [4]

0.00 0.00 1/1 wavWriter(float\*, int, char\*) [15]

-----------------------------------------------

1784.78 0.00 1/1 main [1]

[2] 100.0 1784.78 0.00 1 convolve(float\*, int, float\*, int, float\*, int) [2]

-----------------------------------------------

0.02 0.00 2/2 main [1]

[3] 0.0 0.02 0.00 2 wavReader(float\*, char\*, int\*) [3]

0.00 0.00 2/3 std::operator|(std::\_Ios\_Openmode, std::\_Ios\_Openmode) [11]

0.00 0.00 2/2 std::fpos<\_\_mbstate\_t>::fpos(long) [12]

-----------------------------------------------

0.02 0.00 1/1 main [1]

[4] 0.0 0.02 0.00 1 scale(float\*, int) [4]

-----------------------------------------------

0.00 0.00 1/3 wavWriter(float\*, int, char\*) [15]

0.00 0.00 2/3 wavReader(float\*, char\*, int\*) [3]

[11] 0.0 0.00 0.00 3 std::operator|(std::\_Ios\_Openmode, std::\_Ios\_Openmode) [11]

-----------------------------------------------

0.00 0.00 2/2 wavReader(float\*, char\*, int\*) [3]

[12] 0.0 0.00 0.00 2 std::fpos<\_\_mbstate\_t>::fpos(long) [12]

-----------------------------------------------

0.00 0.00 1/1 \_\_libc\_csu\_init [21]

[13] 0.0 0.00 0.00 1 \_GLOBAL\_\_sub\_I\_chunkId [13]

0.00 0.00 1/1 \_\_static\_initialization\_and\_destruction\_0(int, int) [14]

-----------------------------------------------

0.00 0.00 1/1 \_GLOBAL\_\_sub\_I\_chunkId [13]

[14] 0.0 0.00 0.00 1 \_\_static\_initialization\_and\_destruction\_0(int, int) [14]

-----------------------------------------------

0.00 0.00 1/1 main [1]

[15] 0.0 0.00 0.00 1 wavWriter(float\*, int, char\*) [15]

0.00 0.00 1/3 std::operator|(std::\_Ios\_Openmode, std::\_Ios\_Openmode) [11]

-----------------------------------------------

Copyright (C) 2012-2016 Free Software Foundation, Inc.

Copying and distribution of this file, with or without modification,

are permitted in any medium without royalty provided the copyright

notice and this notice are preserved.

Index by function name

[13] \_GLOBAL\_\_sub\_I\_chunkId [2] convolve(float\*, int, float\*, int, float\*, int) [12] std::fpos<\_\_mbstate\_t>::fpos(long)

[14] \_\_static\_initialization\_and\_destruction\_0(int, int) [3] wavReader(float\*, char\*, int\*) [11] std::operator|(std::\_Ios\_Openmode, std::\_Ios\_Openmode)

[4] scale(float\*, int) [15] wavWriter(float\*, int, char\*)

**Summary –**

In this change, I implemented the standard Input side convolution. Because of how slow the algorithm is, we see that the total run time is 1784 seconds and 100% of our run time is spent inside of the convolve function.

1. **FFT Optimized Program**

**Profile –**

Flat profile:

Each sample counts as 0.01 seconds.

% cumulative self self total

time seconds seconds calls s/call s/call name

95.98 3.41 3.41 3 1.14 1.14 four1(float\*, int, int)

1.98 3.48 0.07 1 0.07 3.50 convolve(float\*, int, float\*, int, float\*, int)

0.85 3.51 0.03 2 0.02 0.02 wavReader(float\*, char\*, int\*)

0.56 3.53 0.02 1 0.02 0.02 four1Scale(float\*, int)

0.56 3.55 0.02 1 0.02 0.02 scale(float\*, int)

0.28 3.56 0.01 1 0.01 0.01 wavWriter(float\*, int, char\*)

0.00 3.56 0.00 3 0.00 0.00 std::operator|(std::\_Ios\_Openmode, std::\_Ios\_Openmode)

0.00 3.56 0.00 2 0.00 0.00 std::fpos<\_\_mbstate\_t>::fpos(long)

0.00 3.56 0.00 1 0.00 0.00 \_GLOBAL\_\_sub\_I\_chunkId

0.00 3.56 0.00 1 0.00 0.00 \_\_static\_initialization\_and\_destruction\_0(int, int)

Call graph (explanation follows)

granularity: each sample hit covers 2 byte(s) for 0.28% of 3.56 seconds

index % time self children called name

<spontaneous>

[1] 100.0 0.00 3.56 main [1]

0.07 3.43 1/1 convolve(float\*, int, float\*, int, float\*, int) [2]

0.03 0.00 2/2 wavReader(float\*, char\*, int\*) [4]

0.02 0.00 1/1 scale(float\*, int) [6]

0.01 0.00 1/1 wavWriter(float\*, int, char\*) [7]

-----------------------------------------------

0.07 3.43 1/1 main [1]

[2] 98.3 0.07 3.43 1 convolve(float\*, int, float\*, int, float\*, int) [2]

3.41 0.00 3/3 four1(float\*, int, int) [3]

0.02 0.00 1/1 four1Scale(float\*, int) [5]

-----------------------------------------------

3.41 0.00 3/3 convolve(float\*, int, float\*, int, float\*, int) [2]

[3] 95.8 3.41 0.00 3 four1(float\*, int, int) [3]

-----------------------------------------------

0.03 0.00 2/2 main [1]

[4] 0.8 0.03 0.00 2 wavReader(float\*, char\*, int\*) [4]

0.00 0.00 2/3 std::operator|(std::\_Ios\_Openmode, std::\_Ios\_Openmode) [14]

0.00 0.00 2/2 std::fpos<\_\_mbstate\_t>::fpos(long) [15]

-----------------------------------------------

0.02 0.00 1/1 convolve(float\*, int, float\*, int, float\*, int) [2]

[5] 0.6 0.02 0.00 1 four1Scale(float\*, int) [5]

-----------------------------------------------

0.02 0.00 1/1 main [1]

[6] 0.6 0.02 0.00 1 scale(float\*, int) [6]

-----------------------------------------------

0.01 0.00 1/1 main [1]

[7] 0.3 0.01 0.00 1 wavWriter(float\*, int, char\*) [7]

0.00 0.00 1/3 std::operator|(std::\_Ios\_Openmode, std::\_Ios\_Openmode) [14]

-----------------------------------------------

0.00 0.00 1/3 wavWriter(float\*, int, char\*) [7]

0.00 0.00 2/3 wavReader(float\*, char\*, int\*) [4]

[14] 0.0 0.00 0.00 3 std::operator|(std::\_Ios\_Openmode, std::\_Ios\_Openmode) [14]

-----------------------------------------------

0.00 0.00 2/2 wavReader(float\*, char\*, int\*) [4]

[15] 0.0 0.00 0.00 2 std::fpos<\_\_mbstate\_t>::fpos(long) [15]

-----------------------------------------------

0.00 0.00 1/1 \_\_libc\_csu\_init [23]

[16] 0.0 0.00 0.00 1 \_GLOBAL\_\_sub\_I\_chunkId [16]

0.00 0.00 1/1 \_\_static\_initialization\_and\_destruction\_0(int, int) [17]

-----------------------------------------------

0.00 0.00 1/1 \_GLOBAL\_\_sub\_I\_chunkId [16]

[17] 0.0 0.00 0.00 1 \_\_static\_initialization\_and\_destruction\_0(int, int) [17]

-----------------------------------------------

Copyright (C) 2012-2016 Free Software Foundation, Inc.

Copying and distribution of this file, with or without modification,

are permitted in any medium without royalty provided the copyright

notice and this notice are preserved.

Index by function name

[16] \_GLOBAL\_\_sub\_I\_chunkId [6] scale(float\*, int) [15] std::fpos<\_\_mbstate\_t>::fpos(long)

[5] four1Scale(float\*, int) [2] convolve(float\*, int, float\*, int, float\*, int) [14] std::operator|(std::\_Ios\_Openmode, std::\_Ios\_Openmode)

[17] \_\_static\_initialization\_and\_destruction\_0(int, int) [4] wavReader(float\*, char\*, int\*)

[3] four1(float\*, int, int) [7] wavWriter(float\*, int, char\*)

**Regression Testing (Output from FC command with WAV files)–**

Comparing Change1Output.wav and Change2Output.wav...

Compare error at OFFSET 1992

file1 = F9

file2 = FA

Compare error at OFFSET 1FAE

file1 = 3

file2 = 2

Compare error at OFFSET 272C

file1 = 1C

file2 = 1B

Compare error at OFFSET 4630

file1 = D9

file2 = DA

Compare error at OFFSET 4680

file1 = D6

file2 = D7

Compare error at OFFSET 4D96

file1 = 18

file2 = 19

Compare error at OFFSET 4DE8

file1 = 13

file2 = 14

Compare error at OFFSET 675C

file1 = D9

file2 = D8

Compare error at OFFSET A5A0

file1 = 7A

file2 = 79

Compare error at OFFSET A606

file1 = 5F

file2 = 5E

10 mismatches - ending compare

n

**Summary -**

In this change, I implemented the FFT solution. Because of how much faster that algorithm is, my runtime has been improved to 3.56 seconds. Since the FFT is so much faster, we spent much less time in the convolution stage compared to last time, therefore we can see that 0.28% of our time is spent inside of the wavWriter function (an increase in percentage). When regression testing and comparing the output wav files from the previous version, I notice many differences in my wav files but when I play them they still produce the same sounds, so I move along.

1. **FFT Optimized Program + Hand Tuning**

**Profile –**

Flat profile:

Each sample counts as 0.01 seconds.

% cumulative self self total

time seconds seconds calls s/call s/call name

95.59 3.74 3.74 3 1.25 1.25 four1(float\*, int, int)

1.79 3.81 0.07 1 0.07 3.85 convolve(float\*, int, float\*, int, float\*, int)

1.03 3.85 0.04 1 0.04 0.04 four1Scale(float\*, int)

0.51 3.87 0.02 2 0.01 0.01 wavReader(float\*, char\*, int\*)

0.51 3.89 0.02 1 0.02 0.02 scale(float\*, int)

0.26 3.90 0.01 1 0.01 0.01 wavWriter(float\*, int, char\*)

0.00 3.90 0.00 3 0.00 0.00 std::operator|(std::\_Ios\_Openmode, std::\_Ios\_Openmode)

0.00 3.90 0.00 2 0.00 0.00 std::fpos<\_\_mbstate\_t>::fpos(long)

0.00 3.90 0.00 1 0.00 0.00 \_GLOBAL\_\_sub\_I\_chunkId

0.00 3.90 0.00 1 0.00 0.00 \_\_static\_initialization\_and\_destruction\_0(int, int)

Call graph (explanation follows)

granularity: each sample hit covers 2 byte(s) for 0.26% of 3.90 seconds

index % time self children called name

<spontaneous>

[1] 100.0 0.00 3.90 main [1]

0.07 3.78 1/1 convolve(float\*, int, float\*, int, float\*, int) [2]

0.02 0.00 2/2 wavReader(float\*, char\*, int\*) [5]

0.02 0.00 1/1 scale(float\*, int) [6]

0.01 0.00 1/1 wavWriter(float\*, int, char\*) [7]

-----------------------------------------------

0.07 3.78 1/1 main [1]

[2] 98.7 0.07 3.78 1 convolve(float\*, int, float\*, int, float\*, int) [2]

3.74 0.00 3/3 four1(float\*, int, int) [3]

0.04 0.00 1/1 four1Scale(float\*, int) [4]

-----------------------------------------------

3.74 0.00 3/3 convolve(float\*, int, float\*, int, float\*, int) [2]

[3] 95.9 3.74 0.00 3 four1(float\*, int, int) [3]

-----------------------------------------------

0.04 0.00 1/1 convolve(float\*, int, float\*, int, float\*, int) [2]

[4] 1.0 0.04 0.00 1 four1Scale(float\*, int) [4]

-----------------------------------------------

0.02 0.00 2/2 main [1]

[5] 0.5 0.02 0.00 2 wavReader(float\*, char\*, int\*) [5]

0.00 0.00 2/3 std::operator|(std::\_Ios\_Openmode, std::\_Ios\_Openmode) [14]

0.00 0.00 2/2 std::fpos<\_\_mbstate\_t>::fpos(long) [15]

-----------------------------------------------

0.02 0.00 1/1 main [1]

[6] 0.5 0.02 0.00 1 scale(float\*, int) [6]

-----------------------------------------------

0.01 0.00 1/1 main [1]

[7] 0.3 0.01 0.00 1 wavWriter(float\*, int, char\*) [7]

0.00 0.00 1/3 std::operator|(std::\_Ios\_Openmode, std::\_Ios\_Openmode) [14]

-----------------------------------------------

0.00 0.00 1/3 wavWriter(float\*, int, char\*) [7]

0.00 0.00 2/3 wavReader(float\*, char\*, int\*) [5]

[14] 0.0 0.00 0.00 3 std::operator|(std::\_Ios\_Openmode, std::\_Ios\_Openmode) [14]

-----------------------------------------------

0.00 0.00 2/2 wavReader(float\*, char\*, int\*) [5]

[15] 0.0 0.00 0.00 2 std::fpos<\_\_mbstate\_t>::fpos(long) [15]

-----------------------------------------------

0.00 0.00 1/1 \_\_libc\_csu\_init [23]

[16] 0.0 0.00 0.00 1 \_GLOBAL\_\_sub\_I\_chunkId [16]

0.00 0.00 1/1 \_\_static\_initialization\_and\_destruction\_0(int, int) [17]

-----------------------------------------------

0.00 0.00 1/1 \_GLOBAL\_\_sub\_I\_chunkId [16]

[17] 0.0 0.00 0.00 1 \_\_static\_initialization\_and\_destruction\_0(int, int) [17]

-----------------------------------------------

Copyright (C) 2012-2016 Free Software Foundation, Inc.

Copying and distribution of this file, with or without modification,

are permitted in any medium without royalty provided the copyright

notice and this notice are preserved.

Index by function name

[16] \_GLOBAL\_\_sub\_I\_chunkId [6] scale(float\*, int) [15] std::fpos<\_\_mbstate\_t>::fpos(long)

[4] four1Scale(float\*, int) [2] convolve(float\*, int, float\*, int, float\*, int) [14] std::operator|(std::\_Ios\_Openmode, std::\_Ios\_Openmode)

[17] \_\_static\_initialization\_and\_destruction\_0(int, int) [5] wavReader(float\*, char\*, int\*)

[3] four1(float\*, int, int) [7] wavWriter(float\*, int, char\*)

**Regression Testing (Output from FC command with WAV Files) –**

Comparing Change2Output.wav and Change3Output.wav...

Files compare OK

n

**Changes (Next 5 Pages)–**

1. **Convolve Clearing loops (Loop Unrolling)**

while (i < newArrSize){

newInput[i] = 0;

i++;

}

while (i < newArrSize){

newIR[i] = 0;

i++;

}

**Become**

while (i < newArrSize){

newInput[i] = 0;

newInput[i+1] = 0;

i+=2;

}

while (i < newArrSize){

newIR[i] = 0;

newIR[i+1] = 0;

i+=2;

}

In this improvement, I unroll the loops that clear the newIR[] and newInput[] arrays. This speeds up my code because the loops iterate much less and the increment instruction is called half as much as before.

1. **Remove power operation (Strength Reduction)**

short sam;

sig = new float[wavSize];

for (int i = 0; i < wavSize; i++){

sam = wavData[i];

sig[i] = (sam\*1.0) / (pow(2.0, 15.0) -1);

if (sig[i] < -1.0){

sig[i] = -1.0;

}

}

**BECOMES**

short sam;

sig = new float[wavSize];

for (int i = 0; i < wavSize; i++){

sam = wavData[i];

sig[i] = (sam\*1.0) / (32767);

if (sig[i] < -1.0){

sig[i] = -1.0;

}

}

In this improvement I replace the pow(2.0, 15.0)-1 operation with 32767. Because that power operation will always result in the same number, there is no point in executing it over and over in each iteration of my for loop. I can use a constant value instead resulting in a more efficient program.

1. **Convolve Clearing output loop (Loop Unrolling)**

for (i = 0; i < newArrSize; i++) {

newOutput[i] = 0;

}

**Becomes**

for (i = 0; i < newArrSize; i+=2) {

newOutput[i] = 0;

newOutput[i+1] = 0;

}

In this improvement, I unroll the loop that clears the newOutput[] array. This refactoring is very similar to the first and it improves the speed for the same reasons the first hand tune improved it.

1. **FFT Scaling (Loop unrolling)**

// From notes in class

void four1Scale (float signal[], int N)

{

int i;

int j;

for (i = 0, j = 0; i < N; i++, j+=2) {

signal[j] /= (float)N;

signal[j+1] /= (float)N;

}

}

**Becomes**

// From notes in class

void four1Scale (float signal[], int N)

{

int i;

int j;

for (i = 0, j = 0; i < N; i++, j+=4) {

signal[j] /= (float)N;

signal[j+1] /= (float)N;

signal[j+2] /= (float)N;

signal[j+3] /= (float)N;

}

}

In this improvement, I unroll the FFT scaling function given to us in class. Although the loop was already partially unrolled, I decided to unroll it even further resulting in less iterations and a faster code.

1. **Convolve Declaring Array Sizes (Strength Reduction)**

newInput = new float[2 \* newArrSize];

newIR = new float[2 \* newArrSize];

newOutput = new float[2 \* newArrSize];

**Becomes**

newInput = new float[newArrSize + newArrSize];

newIR = new float[newArrSize + newArrSize];

newOutput = new float[newArrSize + newArrSize];

In this improvement, when creating my newInput[], newIR[], and newOutput[] arrays, when deciding on the size, instead of multiplying newArrSize by 2 every time, I just add the same newArrSize together. This is because addition is a much faster operation than multiplication, so this would also result in a quicker program.

This can be optimized even further by creating an integer than equals newArrSize + newArrSize and creating those arrays with that variable instead of performing the same addition instruction three times.

**Summary –**

After all of my hand tunings, although many loops are iterating less times and I am saving time when calling some functions, the run time is longer as if I never optimized it in the first place. With a current run time of 3.9 seconds, this puts me almost 0.3 seconds slower than the non hand tuned version. When comparing the wav files from the previous FFT version and this current optimized FFT version, they are the exact same.

1. **FFT Optimized Program + Hand Tuning + Compiler Optimization**

**Profile –**

Flat profile:

Each sample counts as 0.01 seconds.

% cumulative self self total

time seconds seconds calls ms/call ms/call name

97.29 2.69 2.69 3 898.32 898.32 four1(float\*, int, int)

1.81 2.75 0.05 convolve(float\*, int, float\*, int, float\*, int)

0.36 2.76 0.01 scale(float\*, int)

0.36 2.77 0.01 wavReader(float\*, char\*, int\*)

0.36 2.78 0.01 wavWriter(float\*, int, char\*)

0.00 2.78 0.00 1 0.00 0.00 \_GLOBAL\_\_sub\_I\_chunkId

Call graph (explanation follows)

granularity: each sample hit covers 2 byte(s) for 0.36% of 2.78 seconds

index % time self children called name

<spontaneous>

[1] 98.9 0.05 2.69 convolve(float\*, int, float\*, int, float\*, int) [1]

2.69 0.00 3/3 four1(float\*, int, int) [2]

-----------------------------------------------

2.69 0.00 3/3 convolve(float\*, int, float\*, int, float\*, int) [1]

[2] 97.1 2.69 0.00 3 four1(float\*, int, int) [2]

-----------------------------------------------

<spontaneous>

[3] 0.4 0.01 0.00 scale(float\*, int) [3]

-----------------------------------------------

<spontaneous>

[4] 0.4 0.01 0.00 wavReader(float\*, char\*, int\*) [4]

-----------------------------------------------

<spontaneous>

[5] 0.4 0.01 0.00 wavWriter(float\*, int, char\*) [5]

-----------------------------------------------

0.00 0.00 1/1 \_\_libc\_csu\_init [21]

[13] 0.0 0.00 0.00 1 \_GLOBAL\_\_sub\_I\_chunkId [13]

-----------------------------------------------

Copyright (C) 2012-2016 Free Software Foundation, Inc.

Copying and distribution of this file, with or without modification,

are permitted in any medium without royalty provided the copyright

notice and this notice are preserved.

Index by function name

[13] \_GLOBAL\_\_sub\_I\_chunkId [3] scale(float\*, int) [4] wavReader(float\*, char\*, int\*)

[2] four1(float\*, int, int) [1] convolve(float\*, int, float\*, int, float\*, int) [5] wavWriter(float\*, int, char\*)

**Regression Testing (Output from FC command with WAV Files) –**

Comparing Compiler.wav and Change3Output.wav...

Files compare OK

n

**Summary –**

After using the compiler optimization with setting -O3

Example command –

G++ -O3 -pg convolveFFT1.cpp -o convolveFFT1

My run time improves to 2.78 seconds, more than 1 whole second from the previous version. When comparing my WAV files the output is also the same so I know that introducing the compiler optimization did not introduce any new bugs.