PARALLELIZED CONVOLUTION

-Venkata Nanda Kishore Buddhiraju
3764-8659

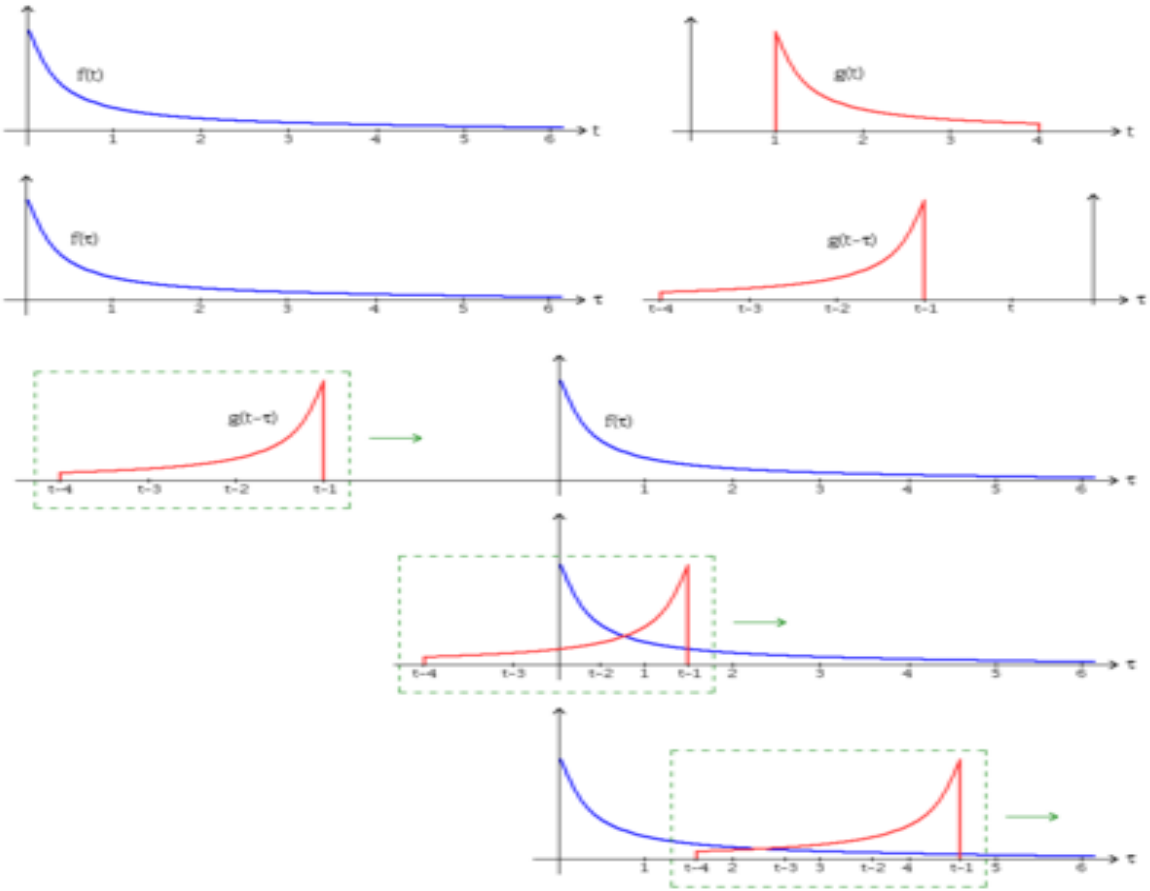
CSE 633

Fall, 2011

Convolution

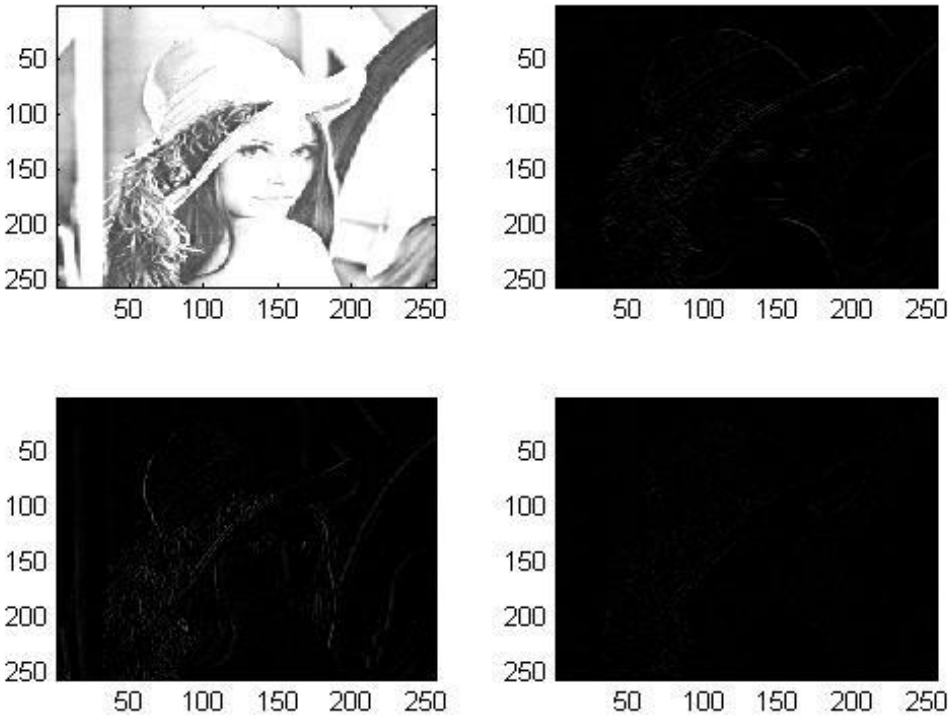
- Extent of overlap of 2 functions.
- Used to smooth the images and other functions.
- Also,

causes Blurring.



Example: Convolution of an image at various levels

Original Image



Top left: level 1, Top right: level 2
Bottom left: level 3, Bottom right: level 4

Technically: Let O be convolution matrix, I is Image matrix

I_{11}	I_{12}	I_{13}	I_{14}	I_{15}	I_{16}	I_{17}	I_{18}	I_{19}
I_{21}	I_{22}	I_{23}	I_{24}	I_{25}	I_{26}	I_{27}	I_{28}	I_{29}
I_{31}	I_{32}	I_{33}	I_{34}	I_{35}	I_{36}	I_{37}	I_{38}	I_{39}
I_{41}	I_{42}	I_{43}	I_{44}	I_{45}	I_{46}	I_{47}	I_{48}	I_{49}
I_{51}	I_{52}	I_{53}	I_{54}	I_{55}	I_{56}	I_{57}	I_{58}	I_{59}
I_{61}	I_{62}	I_{63}	I_{64}	I_{65}	I_{66}	I_{67}	I_{68}	I_{69}

K_{11}	K_{12}	K_{13}
K_{21}	K_{22}	K_{23}

$$O_{57} = I_{57}K_{11} + I_{58}K_{12} + I_{59}K_{13} + I_{67}K_{21} + I_{68}K_{22} + I_{69}K_{23}$$

$$O(i,j) = \sum_{k=1}^{m} \sum_{l=1}^{n} I(i+k-1, j+l-1)K(k,l)$$

eg. :2048×1536 = 3,145,728 pixels or 3.1 megapixels

Difficulties

- Have to perform padding
- Huge number of multiplications
- Becomes much larger with Kernel size

Solution: Parallelize!

- Parallelism should reduce the time to compute
- Different approaches available
- Master node has the Input matrix and the kernel
- Master node decides the extent of padding and pads the input matrix

Parallelize(2)

- Master node then distributes chunks of input matrix to other nodes
- These nodes/processors compute local convolution and sends them to the Master node
- Master, then fixes the output matrix together
- Because of independent convolutions, distributed parallelism can be implemented

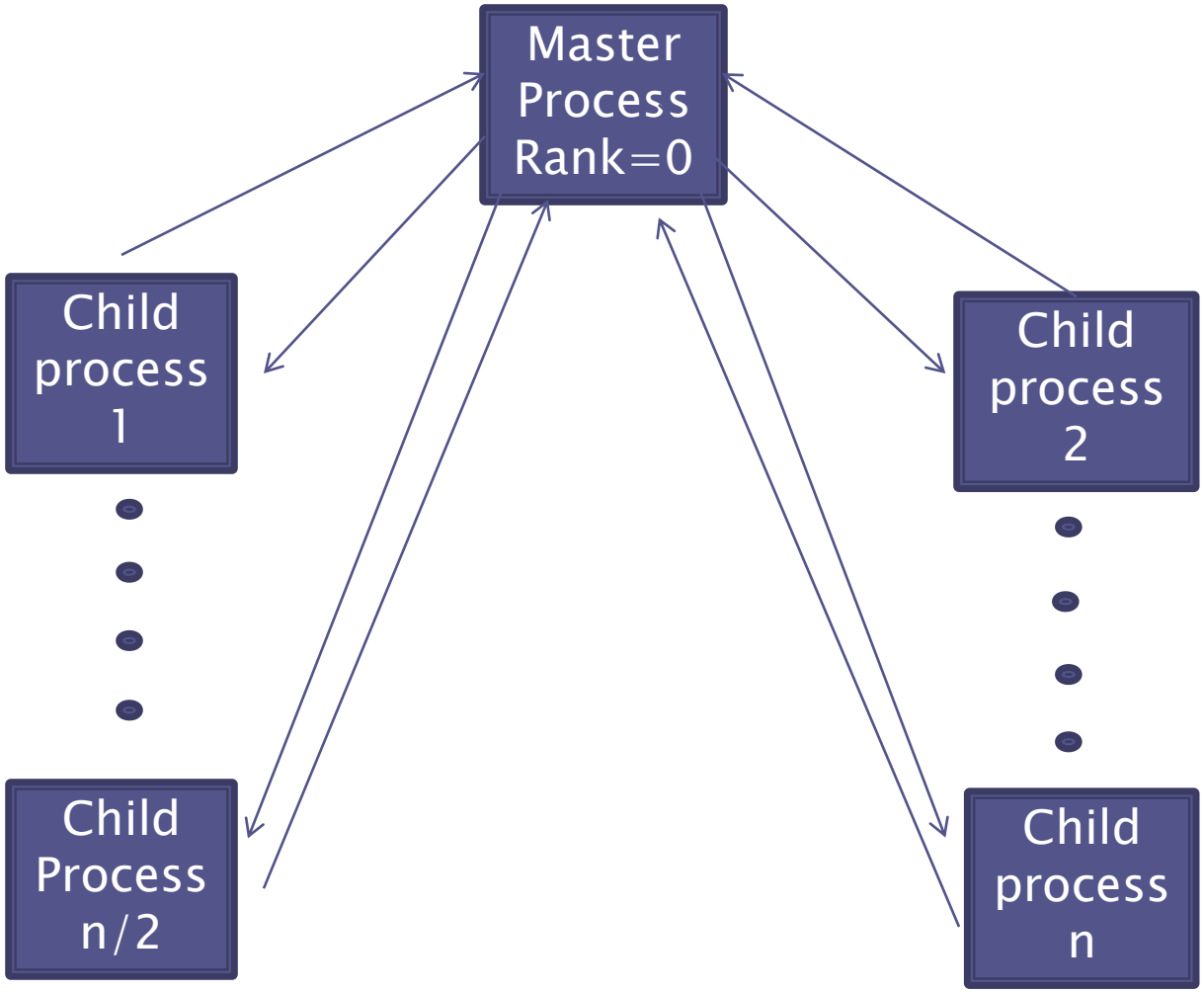
But–

- Communication between the nodes to align the right borders of local convolutions to facilitate merging
- This is communication overhead
- Or, append border pixels to the chunk of input matrix pixels for adjacent nodes
- This is redundancy of data!

Focus

- Levels of convolution
- Will be on the no. of chunks
- Time taken to compute
- Overhead vs. Redundancy
- Will be implementing on C and then using MPI

Implementation

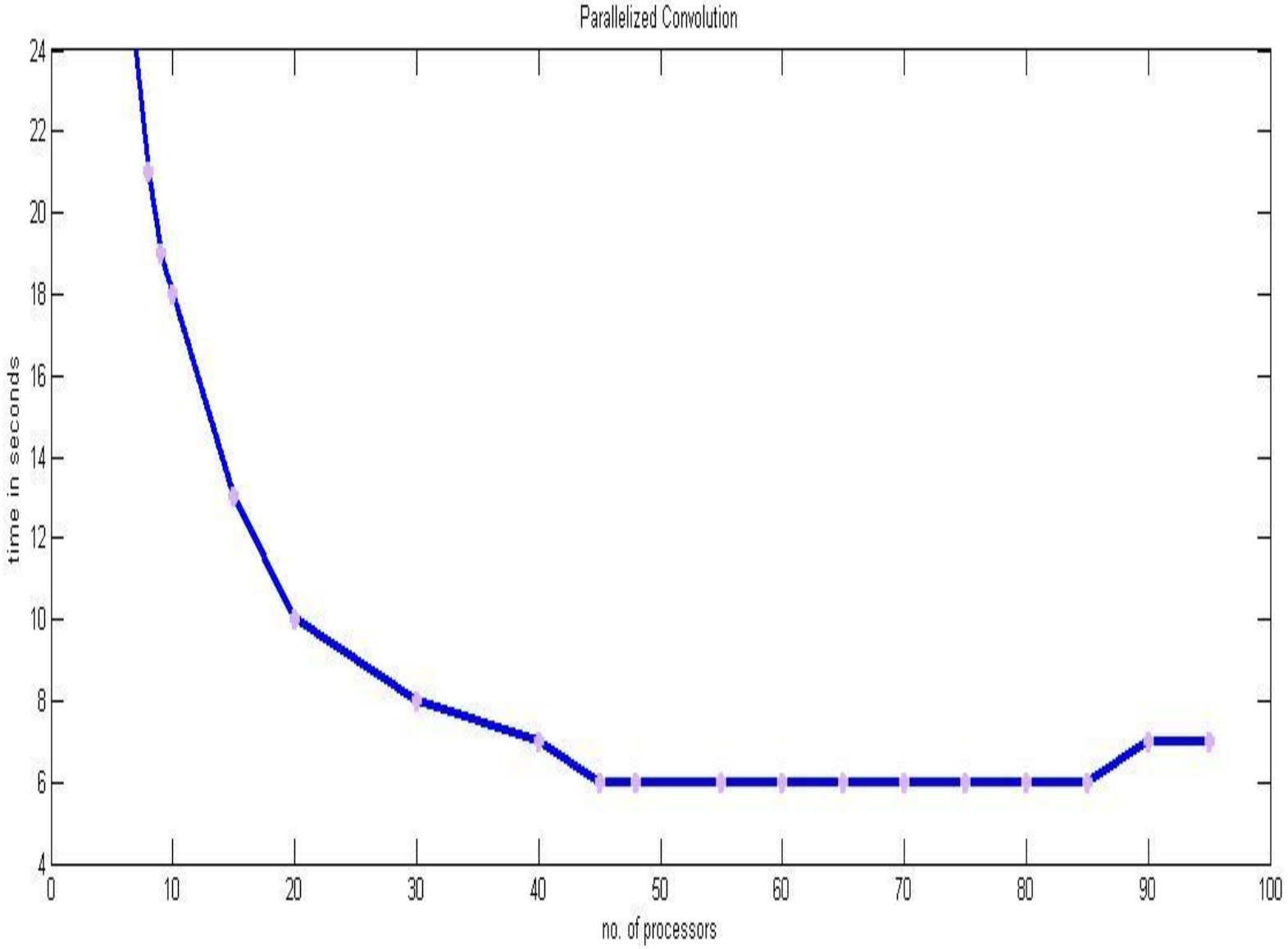


Problem Domain

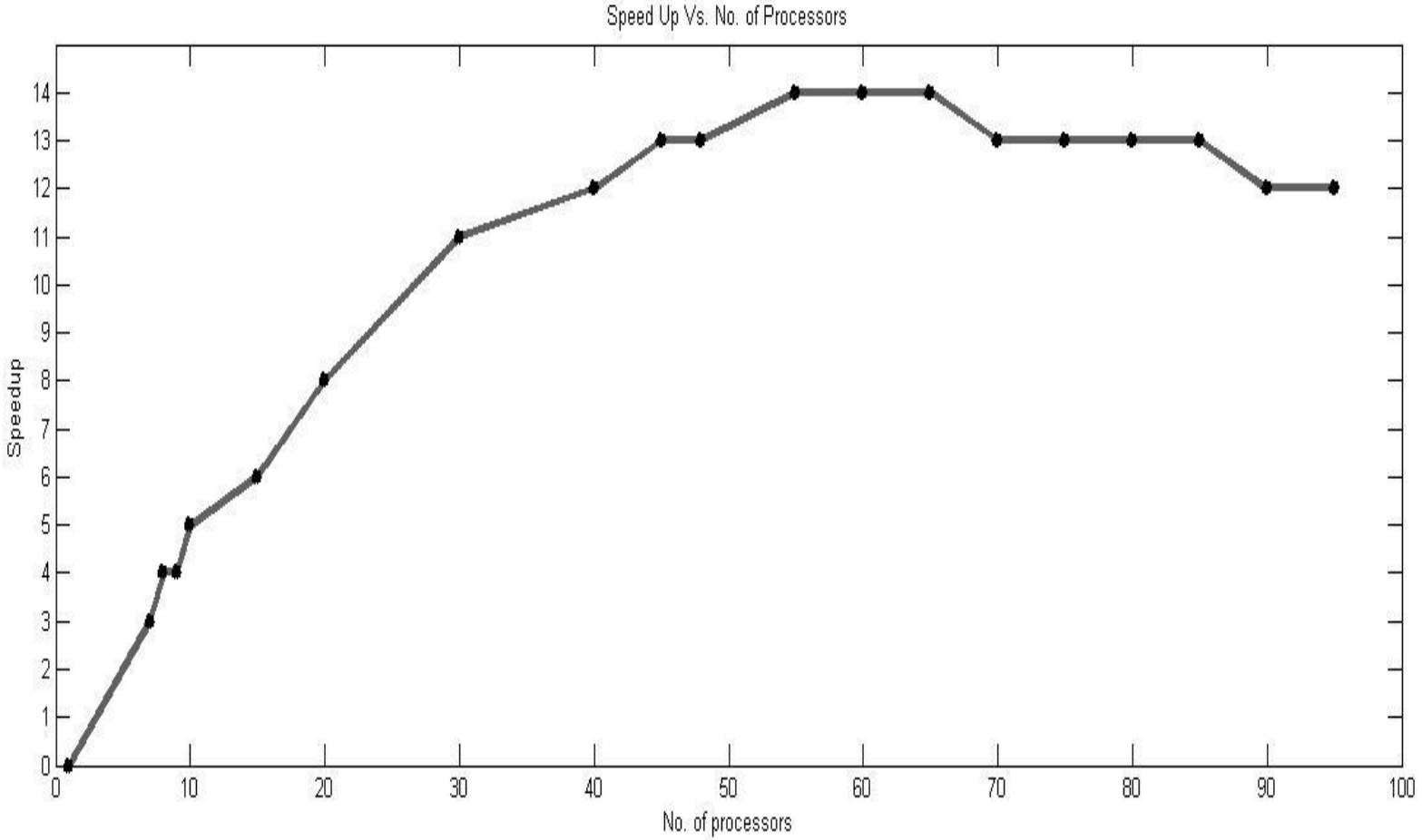
- 1200 x 1200 image
- 80 x 80 kernel
- Around 8 billion Multiplication operations
- And a similar no. of Additions
- No. of Processors subscribed: 2 to 90

Note: The test cases mentioned here are changed after the presentation in the class, for a better throughput.

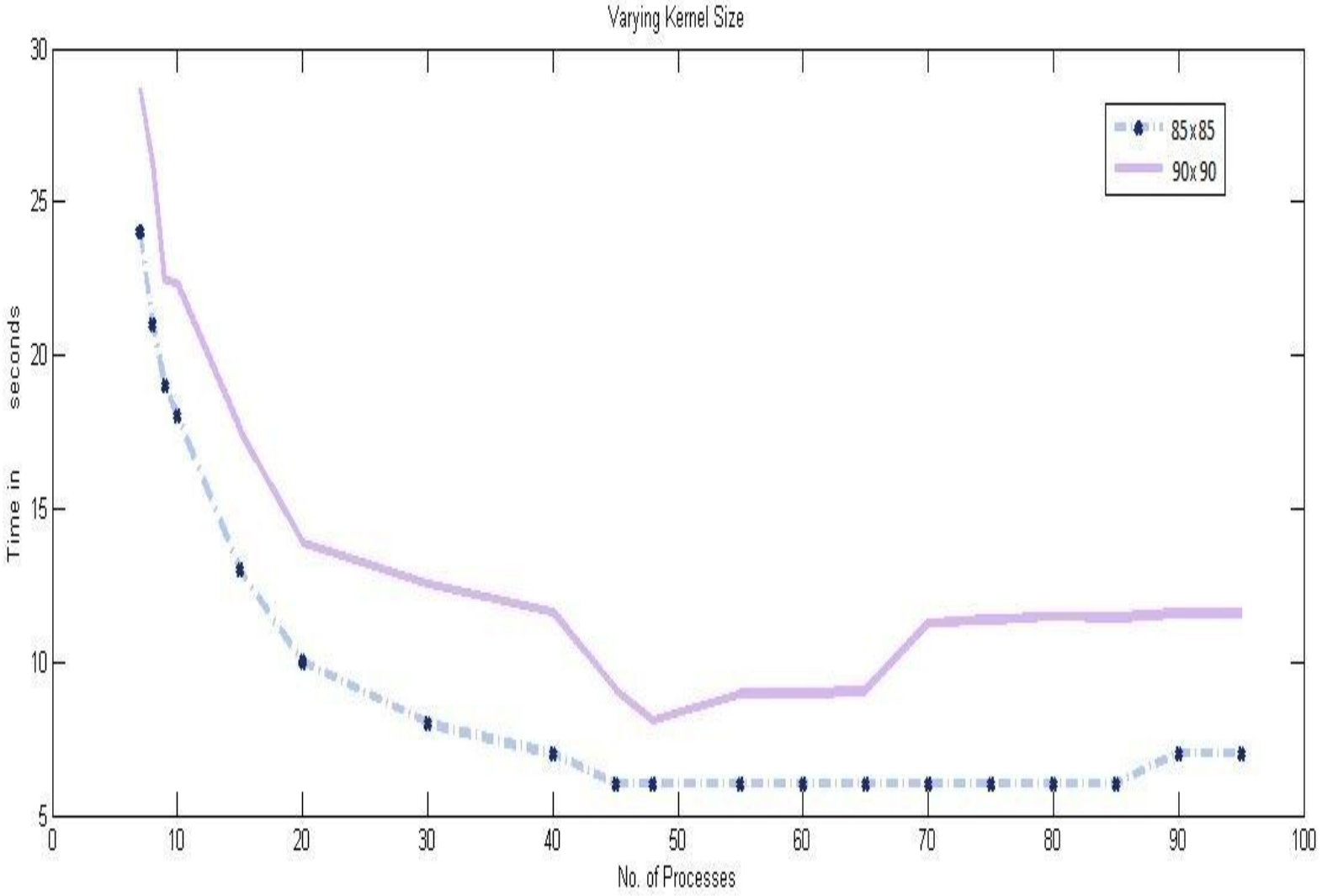
Results: Parallel Convolution



Speed Up vs. No. of Processors



Change of Kernel Size



Effects of Under subscription

(level 2 convolution)

Max. No. of processors (‘x’) (subscribed)	Processors requested(‘y’)	Time taken for ‘y’ processes on ‘x’ processors	Original time for ‘y’ processes on ‘y’ processors
72	144	88.451	71.046
72	115	82.869	70.847
72	100	81.734	73.646
72	90	76.170	71.783
72	88	75.146	71.978
48	80	87.732	73.568
48	72	81.088	70.512
48	60	78.836	73.291
48	55	77.356	73.779

Observations

- Dealing with master process was tricky. It either is slacking off or overloaded.
- Significant increase in the size of data did not result in a proportionate increase in time taken. Only a small factor of increase in time was observed.
- Under subscribing enabled the usage of the processors with more efficiency.

Questions put during the in- class presentation

- Was I able to distribute the data equally among sub- processes?

Answer: No. It is not always possible to divide the data equally. I made the master process compute convolutions for these outliers.

- Wont the sub- processes have a copy of the matrix as it is initialized in the program itself? If so, why distribute the data again?

Answer: Assuming that in a distributed environment, data is not known prior to the computation, I re- distributed the data through master process to other processes.

Future Works

- Implementation of separable convolution
- Implementing the same work on GPUs
- Implementing on OpenMP to see the execution time patterns on shared memory

Acknowledgements

- Dr. Russ Miller
- Dr. Matt Jones
- Cynthia D. Cornelius

References:

http://www.ccr.buffalo.edu/download/attachments/65681/Mpi-advanced-handout-2x2.pdf?version=2

http://www.ccr.buffalo.edu/download/attachments/65681/Mpi-intermed-handout-2x2.pdf?version=2

http://www.scribd.com/doc/58013724/10-MPI-Programmes

http://heather.cs.ucdavis.edu/~matloff/mpi.html

THANK YOU !!