Research on paralleling compute efficiency of FIR filter

Jinbo Wang [a], Lin Mao [b]

College of Electronic Enginering, Naval Univ. of Engineering, Wuhan 430033, China

[a]wjbnavy@126.com, [b]maolin@126.com

Keywords: paralleling compute, FIR, GPU, CUDA.

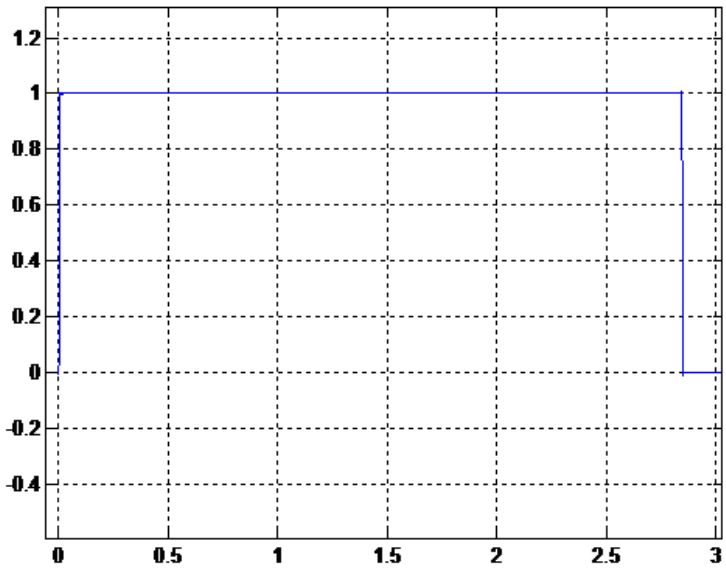
Abstract. Software radar is a hot direction of radar system research, and FIR filter is the basic part of the software radar system. Be restricted by the CPU speed, most of the FIR filter is achieved on the DSP. Following the development of new technology, GPU perhaps can replace the role of the DSP. This paper designs a band pass filter, then uses the GPU to fulfill that algorithm, Finally process some typical signals to test the speed and research the efficiency of using GPU to accomplish real time compute, it's useful for further research on software radar system.

Introduction

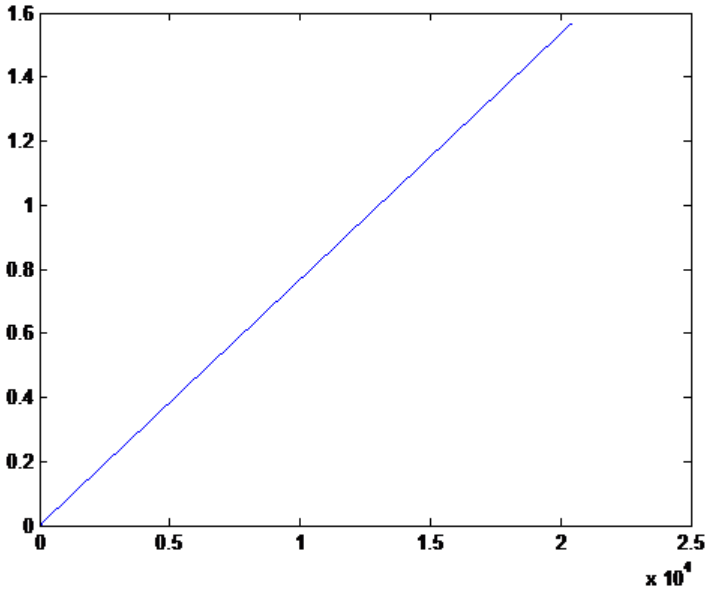
IT has promoted the great changes in military technology and weaponry. Software Radar uses software platform to support a new radar system. FIR filter is the basic unit of software radar, it has high real-time computing needs. Subject to CPU computing speed, it is generally achieved on the DSP, but with the development of microelectronics technology, GPU can be used to implement the FIR filter. This paper uses GPU parallel processing capability to test 4096-tap FIR filter operation speed. It is surprised that the speed exceed 30000 times than CPU.

Linear phase FIR filter

In this paper, we design a sound filter for the parallel computing test, depending on the frequency range of the sound signal, the FIR filter's[2,4] sampling frequency is 44.1kHz, initial frequency is 20Hz, the cutoff frequency is 20KHZ, the filter order is 4096. Its amplitude frequency response and phase frequency response curve are shown in Figure1 (a), Figure1 (b).



(a) Amplitude frequency response



(b) Phase frequency response

Fig1. FIR filter's amplitude frequency and phase frequency response

Effectiveness analysis of parallel computing

The paper calls functions of cables inside the CUDA library [1-6]. The CPU and GPU's operational time are shown in Table1.

Table1. CPU and GPU's computing time

	CPU	GPU
Time(ms)	1044.792578	435.860779

From Tab1 we can see the 4096 points convolution operation speed. The operation time of the GPU is 2.3971 times than the CPU. Although GPU's speed is faster, but the advantage of GPU computing speed is not obvious.

Analyses the reasons, the 435.860779ms includes the initialization's time, allocation of memory space's time, the data transfer's time, the calculated time, and etc. Each of the parts are respectively listed in the Tab2.

Table2. GPU's computing time schedule

	Initialization's time	Memory space allocation's time	Data transfer's time	Function calculated time	Free up memory's time
Time(ms)	268.306366	0.792498	194.9171	0.032478	0.477593

It was found that CUBLAS has long time-consuming on the initialization and data transfer from memory to memory, it consumes the 99.72% of the total time. The real parallel computing is only used 0.032478ms.

In order to more clearly compare the different of GPU and CPU's speed, we can do different point's convolution, as shown in Table3.

The data in table 3 can draw the Figure 2.

It can be very intuitive to see the curve's beginning part, due to the smaller amount of data, the speed of the GPU is a little bit faster than the CPU, but when N (Taps number) increases, such as

when N = 1024, CPU computing time basically did not change much, However, the GPU computing time decreases exponentially, the advantage of GPU computing obviously reflected.

It is undeniable that the data transfer is inevitable, if we coupled with a delivery time, The GPU parallel computing speed is not fast, and this obviously restricts the practical application of parallel computing.

Table3. Different number taps operation speed

	CPU	GPU	Ratio of GPU and CPU's speed
2	0.001717	0.03002	0.0572
4	0.002565	0.030315	0.0846
8	0.005122	0.030516	0.1678
16	0.016050	0.030408	0.5278
32	0.059340	0.031356	1.8925
64	0.243172	0.030585	7.9507
512	15.135652	0.048447	312.4167
1024	60.918068	0.046899	1298.9
2048	253.418513	0.049074	5164
4096	1049.065185	0.032427	32352

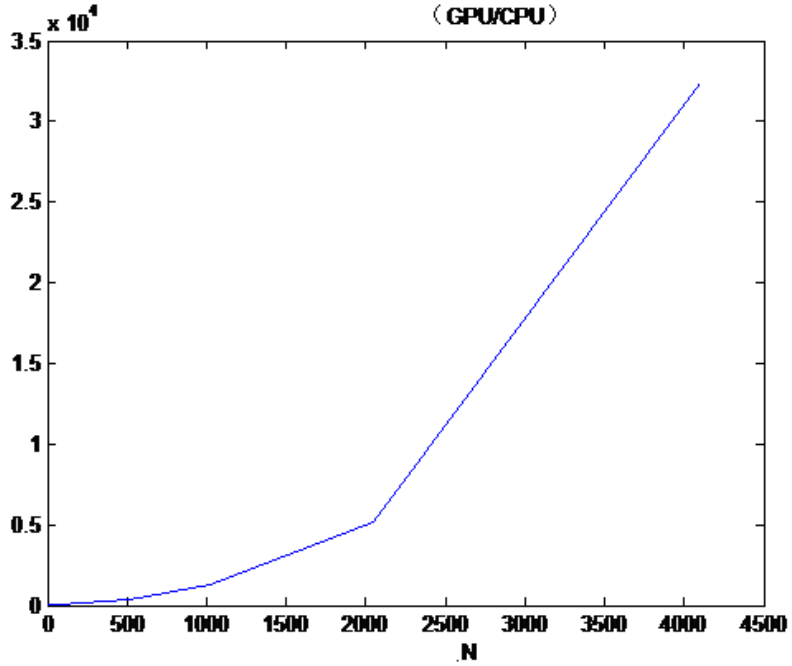


Fig2. Comparison of GPU and CPU speed

Conclusion

Although we can see that the results of the precision calculations on the GPU is worse than on the CPU, but the speed can be increased to more than five times. Further analysis of GPU program, we find that the main time-consuming is the data transfer between GPU and CPU memory, but if we only compare GPU and CPU speed, then the speed up is very impressive, when N = 4096 you can improve 30,000 times. Thus, the transfer of data has become the most time-consuming part, significantly driving down the overall speed of operation of the filter, for the present, it do not meet the real-time requirements. But with the rapid development of hardware technology [5,6], this bottleneck will gradually be overcome.

References

[1] Cublas-library. [Z]. (2007-09-11). http://developer.nvidia.com/object/cuda.html.

[2] Cheng PeiQing. Digital signal processing tutorial (third edition) [M]. Beijing: Tsinghua University Press, 2007.2.

[3] Zhang Shu, etc GPU of high-performance computing CUDA [M]. Beijing: China Water Power Press, 2009.10.

[4] Hu GuangShu. Digital Signal Processing- Theory, Algorithms and Implementation (Second Edition) [M]. Beijing: Tsinghua University Press, 2003.

[5] Pang Bo, Lin XinDang, etc. Software radar vision [J]. Radar and confrontation in March 2010, Vol. 30(1).

[6] CUDA—toward a new era of GPGPU [J]. Programmer, 2008, 10.

[7] Cheng Xiaoxu, etc. C language algorithm Quick Reference [M]. Beijing People's Posts and Telecommunications Press, 2009.1.

[8] Jason Sanders, Edward Kandrot (US). CUDA paradigm fine solution-General GPU programming (photocopy edition) [M]. Beijing Tsinghua University Press, 2010.10.