MULTI-CORE AND OPTICAL PROCESSOR RELATED APPLICATIONS RESEARCH AT OAK RIDGE NATIONAL LABORATORY

Jacob Barhen (Oak Ridge National Laboratory, Oak Ridge, Tn 37831 USA); Ryan Kerekes (ORISE, Oak Ridge, TN); Jesse St. Charles(ORISE); Mark A. Buckner (ORNL)

ABSTRACT

High-speed parallelization of common tasks holds great promise as a low-risk approach to achieving the significant increases in signal processing and computational performance required for next generation innovations in reconfigurable radio systems. Researchers at the Oak Ridge National Laboratory have been working on exploiting the parallelization offered by this emerging technology and applying it to a variety of problems. This paper will highlight recent experience with four different parallel processors applied to signal processing tasks that are directly relevant to signal processing required for SDR/CR waveforms. The first is the EnLight Optical Core Processor applied to matched filter (MF) correlation processing via fast Fourier transform (FFT) of broadband Doppler-sensitive waveforms (DSW) using active sonar arrays for target tracking. The second is the IBM CELL™ Broadband Engine applied to 2-D discrete Fourier transform (DFT) kernel for image processing and frequency domain processing. And the third is the NVIDIA graphical processor applied to document feature clustering.

EnLight Optical Core Processor. Optical processing is inherently capable of high-parallelism that can be translated to very high performance, low power dissipation computing. The *EnLight*™256 is a small form factor signal processing chip (5×5 cm^2) with a *digital optical core* that is being developed by an Israeli startup company. As part of its evaluation of foreign technology, ORNL's Center for Engineering Science Advanced Research (CESAR) had access to a precursor *EnLight*™64 *Alpha* hardware for a preliminary assessment of capabilities in terms of large Fourier transforms for matched filter banks and on applications related to Doppler-sensitive waveforms. This processor is optimized for array operations, which it performs in fixed-point arithmetic at the rate of 16 *Tera*OPS at 8–bit precision. This is approximately 1000 times faster than the fastest DSP available today. The optical core performs the matrix-vector multiplications, where the nominal matrix size is 256×256. The system clock is 125MHz. At each clock cycle, 128K multiply-and-add operations per second (OPS) are carried out, which yields a peak performance of 16 *Tera*OPS.

IBM Cell™ Broadband Engine. The Cell processor is the extraordinary resulting product of 5 years of sustained, intensive R&D collaboration (involving over $400M investment) between IBM, Sony, and Toshiba. Its architecture comprises one multithreaded 64-bit PowerPC processor element (PPE) with VMX capabilities and two levels of globally coherent cache, and 8 synergistic processor elements (SPEs). Each SPE consists of a processor (SPU) designed for streaming workloads, local memory, and a globally coherent direct memory access (DMA) engine. Computations are performed in 128-bit wide single instruction multiple data streams (SIMD). An integrated high-bandwidth element interconnect bus (EIB) connects the nine processors and their ports to external memory and to system I/O.

The Applied Software Engineering Research (ASER) Group at the ORNL is applying the Cell to a variety of text and image analysis applications. Research on Cell-equipped PlayStation3 (PS3) consoles has led to the development of a correlation-based image recognition engine that enables a single PS3 to process images at more than 10X the speed of state-of-the-art single-core processors.

NVIDIA Graphics Processing Units. The ASER group is also employing the latest NVIDIA graphical processing units (GPUs) to accelerate clustering of thousands of text documents using recently developed clustering algorithms such as document flocking and affinity propagation.

1. MATCHED FILTER PROCESSING WITH THE ENLIGHT OPTICAL CORE PROCESSOR

1.1 Background

Over the past few decades, a great deal of effort has been devoted to the extraction of spatio-temporal information from an array of spatially distributed sensors [1],[2]. In the area of anti-submarine warfare (ASW), much attention has focused on adaptive beamforming, primarily in the context of towed arrays [3][4]. The basic emphasis of such research was to achieve robust detection and direction-of-arrival (DOA) estimation under requirements for auto-calibration of the arrays [5],[6]. Notwithstanding the considerable progress reported over the years, today's leading paradigms still face substantial degradation in the presence of realistic ambient noise and clutter [7].

Demanding calculations need to be performed to achieve source localization, and their complexity is known to increase dramatically with the size of the sensor array. The same observation applies to tracking using active

Proceeding of the SDR 08 Technical Conference and Product Exposition. Copyright © 2008 SDR Forum. All Rights Reserved

sensors and specially designed waveforms. Both applications require substantial processing power that cannot readily be met with standard, off-the-shelf computing hardware.

First, we overview a matched filter framework for target tracking using active sonars and then describe its implementation on a terascale optical core processor, *EnLight*™, recently introduced by Lenslet Laboratories. This revolutionary *digital optical core* processor is optimized for array operations and provides *tera-scale* computing capabilities with native 8-bit fixed-point precision.

1.2 Tracking Underwater Threats

The algorithm we are implementing involves matched filter (MF) correlation processing via fast Fourier transform (FFT) of broadband Doppler-sensitive waveforms (DSW). The keys to reliable target tracking are proper waveform selection, accurate signal and system modeling, and efficient real-time signal processing using an MF bank implementation. The common waveforms used in underwater threat detection via sonar signal analysis have diverse and complimentary characteristics. For example, constant frequency (CF) pulses provide superior range-rate estimation but poor range resolution capabilities. The reverberation clutter power vs Doppler shift of a CF pulse is also more concentrated than that of linear frequency modulated (LFM) signals, another common candidate waveform in sonar tracking systems. To resolve the inherent conflict between reliable detection and good range resolution, signals other than the simple CF pulses have to be used.

The MF is central to ASW applications. Fundamentally, the MF is a correlator, which compares the received signal with a hypothesized signal. The output of the matched filter gives a measure of how well the hypothesized signal matches the received signal as function of a set of parameters, usually the range and velocity of targets. The estimated velocity is that for which the correlation peak magnitude of the filter output has the maximum value. The output of the correlator is calculated via FFT, followed by an inverse FFT. The vectors representing the discrete replica and echo signals can have considerable size in the case of broadband signals and the size of the FFT can easily exceed *100K* complex samples. Hence, it was anticipated that the very computationally expensive implementation of broadband matched filtering would be expedited on the *EnLight* optical processor.

1.3 The EnLight Optical Core Processor

Recently, Lenslet Inc. introduced the novel *EnLight*™ processing platform. The *EnLight*™256 is a small form factor digital signal processing chip (5×5 cm^2) with an *optical core*. The processor is optimized for array operations. It can perform 8–bit fixed point arithmetic at 16 *Tera*OPS. This is substantially faster than the fastest FPGA or DSP processors available today. The optical core performs matrix-vector multiplications (MVM), with a nominal matrix size of 256×256. The system clock is 125MHz. At each clock cycle, 128K multiply-and-add operations per second (OPS) are carried out, which yields the peak performance of 16 *Tera*OPS. Before starting production of the *EnLight*™256 processor, Lenslet built the *EnLight*™64α board, shown in Figure 1, a reduced size 64×64 optical core, as a prototype demonstrator of their optical processing technology. Our proof-of-concept effort used the 64α for all hardware tests. Our scale-up projections were based on the *EnLight*™256 bit-exact simulator.

The EnLight™64α clock operates at 60 MHz. The optical core has 64 input channels (configured as 256 vertical cavity surface emitting lasers, bundled in groups of 4 per channel). The optical core performs the MVM function at the rate of 60 $10^6 \times 64^2 \times 2$ = 492 GOPS. Each of the 64 data components in the input and output channels has an 8-bit accuracy, which results in a data stream of 60 10^6 × 64 × 8 bits/s = 30.7 Gbps.

Figure 1 The *EnLight*™ 64α

1.4 Results

We are interested in demonstrating the ability of the *EnLight* computing platform to robustly track an underwater threat source. For the purpose of the numerical simulations and hardware implementation, the following operational simplifications are made: 1) only a single target is present during the detection process; 2) the speed of sound is constant along the propagation path; and 3) the active sonar location is known. In active sonar systems, the proper selection of the transmitted waveform is crucial for target detection and parameter estimation, especially with the existence of reverberation. Several new classes of DSW pulses have been proposed, which theoretically provide superior reverberation processing to CF pulses by virtue of

Proceedings of the SDR '08 Technical Conference and Product Exposition. Copyright © 2008 SDR Forum, Inc. All Rights Reserved
Proceeding of the SDR 08 Technical Conference and Product Exposition. Copyright © 2008 SDR Forum. All Rights Reserved

their comb-like spectra. The simplest case of a signal with comb-like spectrum is the SFM signal where the modulating waveform itself is also a sinusoid. Thus,

$$s(t) = w(t)\exp^{j2\pi f_o t + j\beta\sin(2\pi f_m t)}.$$

The function $w(t)$ is the amplitude window of the pulse, f_0 is the center frequency, f_m is the modulation frequency, and β is the modulation index, which defines the bandwidth of the SFM signal as $B = 2 f_m (1+\beta)$. The power spectrum of the SFM waveform is a symmetrical comb centered on f_0 with a frequency spacing of f_m. By comparison with a CF pulse, where the energy is concentrated in a single mainlobe, the signal energy of the SFM pulse is distributed between several peaks. The result is a corresponding reduction in reverberation interference. For this benchmark study, the source was assumed to transmit a SFM waveform at a center frequency f_0=1200 *Hz*, with a 400 *Hz* bandwidth *B*. The pulse duration used was T = 1, the frequency was modulated at 5 *Hz*, and the Doppler scale ranged from -5 to +5 m/s. The sampling frequency f_s assumed was 5000 *Hz*, resulting in a sampling interval length of T_s = 0.0002 s.

To benchmark the *EnLight* performance, two computer codes were written, one using the Intel Visual FORTRAN-95 compiler, the other in MATLAB. The former enables the fastest possible execution on an Intel IA-32 dual *Xeon* processor system. The latter interfaces with the *EnLight*^{TM}*256* simulator, which is used to design the actual algorithm that was run both on the *EnLight*^{TM}*64*α hardware platform or is used to project the scaled performance for the *EnLight*^{TM}*256*.

For the MF simulation, a synthetic echo is generated for a particular target range and velocity. The echo signal is correlated with a bank of replicas. Spectral techniques are used. The correlation with the highest magnitude provides an estimate of the Doppler velocity bin. The location of the maximum within that correlation yields the time delay of the echo, and thus provides an estimate of the range. For this benchmark study, the assumed target range and (incoming) velocity were 3 km and -5 m/s respectively. The speed of sound was taken to be 1500 m/s. A MF bank with 32 Doppler bins was implemented, with each filter performing an 80K-sample complex FFT to calculate the cross-power spectrum and an 80K-sample inverse FFT to obtain the cross-correlation output of the filter. As can be seen in Figure 2, the output of the first filter has the closest velocity match to the received signal. The estimated target delay is 4 s and hence the estimated target range is 3 Km, with correct incoming direction and target velocity.

In this proof-of-concept study, an *EnLight*$^{TM}\alpha$ hardware prototype was compared to a *dual*-processor Intel *Xeon* (2GHz) system. A speed-up factor of over 13,000 per processor (for a series of 80K-sample complex FFTs corresponding to 32 Doppler cell banks and 1 target echo)

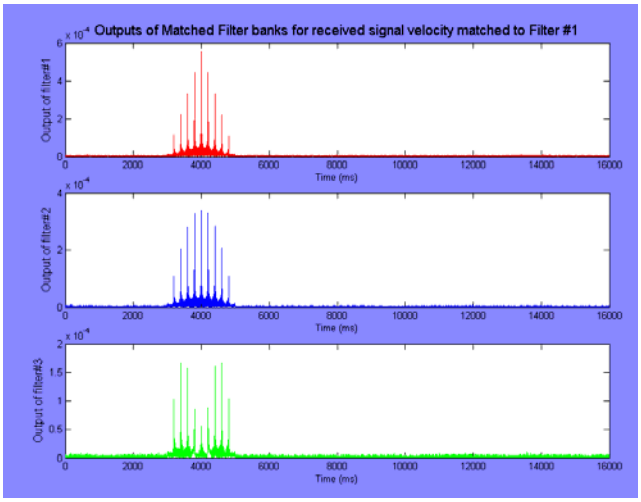

Figure 2 Matched filter back output for a filter bank of 32 Doppler bins. Not all filters are shown.

was achieved. Figure 3 compares the MATLAB simulation of the first filter output with the EnLight$^{TM}\alpha$ hardware runs.

The numerical accuracy of the hardware compares very favorably with the high precision MATLAB simulation and with the theoretical results (see Figure 3).

1.5 Conclusions and Future Research

To achieve the real-time performance required for underwater threat source localization and tracking, many existing algorithms need to be revised and ported to emerging revolutionary computing technologies. These include FPGAs, multicore processors, and optical processors. The *EnLight* terascale optical core processor represents one such revolutionary advance. In that context, our future efforts will focus on demonstrating computational speed and parallel implementation efficiency on multicore architectures for relevant maritime sensing

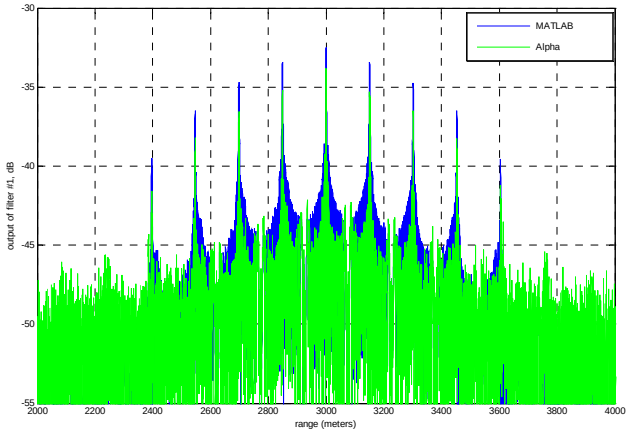

Figure 3 High resolution plot of matched filter bank output (for filter 1) obtained via hardware run on the EnLight$^{TM}\alpha$ demonstration board.

Proceedings of the SDR '08 Technical Conference and Product Exposition. Copyright © 2008 SDR Forum, Inc. All Rights Reserved
Proceeding of the SDR 08 Technical Conference and Product Exposition. Copyright © 2008 SDR Forum. All Rights Reserved

applications; quantifying *speed-up* achieved per processor as compared to a leading-edge conventional processor or DSP; determining *scaling properties* per processor as function of the number of sensors in the detection, tracking, and discrimination network; and characterizing *SNR gain* and *detection improvement* as function of array size and geometry.

2. IMAGE PROCESSING ON THE CELL BROADBAND ENGINE

2.1 Background

The 2-D discrete Fourier transform (DFT) is an important component of image filtering and frequency domain analysis and is typically carried out by the fast Fourier transform (FFT) algorithm. Recent research has focused on improving the speed of FFT both algorithmically and on different processor architectures [8],[9]. The Cell Broadband Engine (BE) processor stands to provide significant performance gains for the 2-D DFT kernel. The PS3 version of the chip offers a PowerPC-type processor along with six single-instruction, multiple-data (SIMD) vector processors which, together, provide over 20 times the single-precision floating-point computational power of conventional processors.

A major challenge arises, when the images are too large to fit into the local memory of the SIMD processor, which is only 256 KB; specifically, a matrix transpose must be carried out in main memory rather than in the local store, necessitating a scheduling algorithm to coordinate memory transfers and manage the workflow. In this section, we expose the overall performance of the PS3 as a 2-D DFT engine. In order to place the problem into context, we focus on the application of correlation-based automatic target recognition (ATR), in which each incoming image in an image stream needs to be searched for one or more known targets (e.g., a T-72 tank) [10]. In this application, a set of "correlation filters", each one designed to locate a specific target, is applied to every image; each filter requires that two 2-D DFTs be carried out on each image. The output image is then searched for peaks, indicating likely target locations.

2.2 Technical approach

The 2-D discrete Fourier transform (DFT) takes a 2-D signal (e.g., an image) as input and produces a 2-D frequency domain signal. Let the 2-D discrete signal $x[m,n]$ be defined for $m = 1 \ldots M$ and $n = 1 \ldots N$. The 2-D DFT $X[k,l]$ of $x[m,n]$ is given by the following equation:

$$X[k,l] = \sum_{m=1}^{M} \sum_{n=1}^{N} x[m,n] e^{-imk} e^{-inl}$$

An image can be filtered by multiplying its DFT by the DFT of the filter and then inverse transforming the result back to the space domain. The 2-D DFT can be decomposed into 1-D DFTs on the rows and columns. It is important to note, however, that in a software implementation of the above algorithm, a full matrix transpose must be carried out between the processing of the rows and columns. This requirement can cause problems when implementing a 2-D DFT on the Cell architecture. If the image is larger than the amount of memory available to the SPE, the transpose needs to be carried out on the PPE. As we show in our experimental results, this can cause bottlenecking and decrease performance.

Image filtering (and, in general, data/signal processing) on the Cell processor may be abstracted into three types of tasks: 1) file I/O (moving data from the remote storage into main memory), 2) PPE computation (for operations that require concurrent access to large amounts of data), and 3) SPE computation. It is desirable to offload as much computation as possible onto the SPEs, provided that they will be given enough work to cover the time needed to transfer the data to and from the SPE local store (which can be done simultaneously with computation). In our application, the Fourier transforms and frequency-domain filtering are all carried out on the SPE. Additionally, transposes are carried out on the SPE for images of size 128x128 or smaller, i.e., "small images".

We designed three software schedulers to carry out the above three tasks to maximize processor utilization. These schedulers run simultaneously as threads. The operation of each scheduler is queue-based, so that when a particular task completes, it can notify another scheduler by placing the next task in its queue. We implemented notification using unnamed pipes shared between threads. While we use these schedulers in a specific manner for image filtering, this general scheduling framework could be used for a variety of signal processing applications on the Cell processor.

2.3 Results

To simulate the performance of our proposed image processing scheme in a real-world scenario, we connected the PS3 to an NFS file server containing numerous images and filters of various sizes. On the NFS server, we used two different storage schemes: 1) the images and filters were stored on the hard disk of the NFS server, and 2) the images and filters were stored in NFS RAM. We also stored the images and filters on the local disk of the PS3. We compare the performance of the architecture across these three storage schemes. As a performance baseline, we implemented a similar image filtering engine on a Dell Optiplex 745 with a 2.8-GHz Intel Pentium D dual-core processor. Our implementation uses two separate threads to

Proceedings of the SDR '08 Technical Conference and Product Exposition. Copyright © 2008 SDR Forum, Inc. All Rights Reserved
Proceeding of the SDR 08 Technical Conference and Product Exposition. Copyright © 2008 SDR Forum. All Rights Reserved

handle processing and file I/O in parallel. We used the well-known FFTW C library [9] for the FFT code, which automatically tunes itself for optimal performance on a given processor. We compare the performance of the PS3 to the FFTW engine to illustrate the potential performance gain of Cell over conventional processor architectures as well as to validate the quality of our Cell implementation.

For both small and large images, we varied the number of filters applied to each image in order to control the amount of computation per data. We measured performance in terms of the number of filtering operations carried out per second. Performance is expected to increase with the number of filters; this is because the ratio of computation to overhead (i.e., file I/O) increases with the number of filters since more filtering needs to be done on any given image. A performance comparison for images of size 128x128 is shown in Figure 4. Reported values are the average of 10 separate runs. We observe that, when all 6 SPUs are employed and many filters are used, Cell outperforms the desktop by a factor of more than 12. This speedup results from a combination of both SPE parallelism and vectorized floating point arithmetic.

The performance improvement observed on small images could not be achieved on large images; rather, we observed that performance was approximately the same as that of the desktop machine. This is due to the somewhat surprising fact that the transpose operation on the PPE takes roughly the same amount of time as one 2-D FFT operation on the SPE, making the transpose a significant bottleneck. Since all transposes must be carried out on the PPE sequentially, the SPEs are forced to spend most of their time waiting on the PPE. The performance curves show that this scheduling inefficiency effectively negates any speedup gained by vector arithmetic.

A direct comparison between small and large images is shown in Figure 5. We normalize by image size, measuring performance in pixels processed per second, in order to put all image sizes on an equal basis. These results emphasize not only the performance improvement achievable with small images on Cell, but also the stark difference in performance when a major part of the computation is moved from the SPE to the PPE.

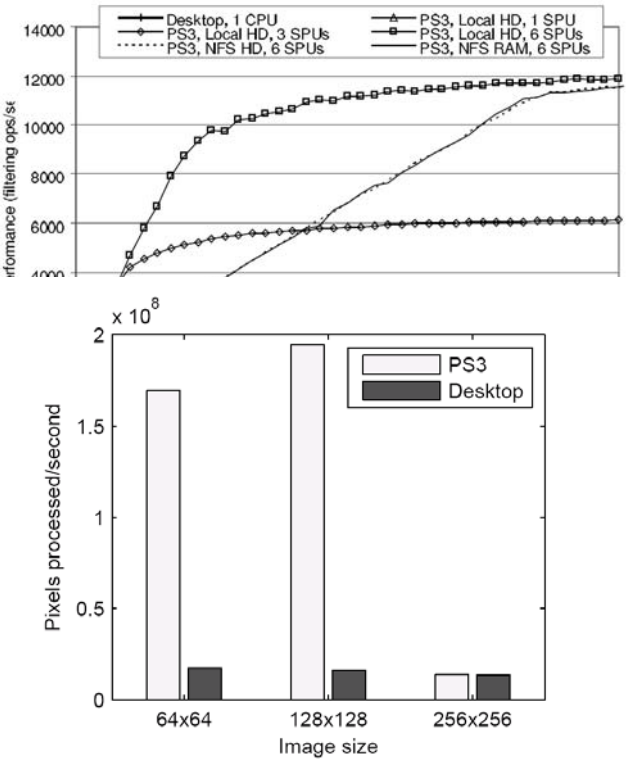

Figure 5 Filtering performance vs. image size for PS3 and desktop machines in terms of pixels processed per second (for an equal basis of comparison).

3. DOCUMENT CLUSTERING ON THE GPU

3.1 Background

The Graphics Processing Unit (GPU) is a specialized processor that is tailored to make extremely fast, data-parallel graphics calculations. Modern GPU hardware has a theoretical performance of over 100 times more floating point operations per second than the current top-of-the-line desktop CPU [12]. This difference comes from the fact that GPU development has centered on highly parallel, computationally intensive calculations rather than data caching or flow control.

To take advantage of this hardware, the graphics hardware company NVIDIA created a C-like language that allows programmers to easily write programs to be run directly on certain NVIDIA® GPUs; this language is called CUDA. In this new platform, execution is thread-based, with threads organized into blocks, which are in turn organized into grids. Inter-thread communication is only allowed between threads of the same block. CUDA follows the Single Program Multiple Data (SPMD) programming paradigm, using a kernel as a blueprint for all threads to be run on the GPU at the same time [5].

In our research we work to exploit the GPU using CUDA and apply its strengths to document clustering. Document clustering is a descriptive data mining task, which involves dividing a set of documents into numerous clusters to find inherent structure. This task is a fundamental operation used in unsupervised document organization, automatic topic extraction, and information retrieval; however, current techniques are slow or require cluster computers to analyze large sets of documents (thousands to millions).

The clustering method we focused on in our research was flocking-based clustering [11]. The flocking model is a biologically inspired computational model for simulating the animation of a flock of entities [13]. In this model, individuals make movement decisions without communicating with other individuals in the system. Each 'boid' acts using a few simple rules: don't get too close to neighboring boids; don't get too far away from neighboring

Proceedings of the SDR '08 Technical Conference and Product Exposition. Copyright © 2008 SDR Forum. All Rights Reserved
Proceeding of the SDR 08 Technical Conference and Product Exposition. Copyright © 2008 SDR Forum. All Rights Reserved

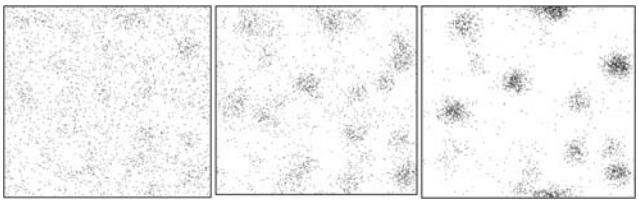

Figure 6 Documents flocking at time step 5, 200, and 400 respectively.

boids; and fly in the same general direction as neighboring boids. By including a similarity rule, each boid only flocks with similar boids. When we extend these rules to compare documents rather than birds, we observe documents clustering into groups of similar content (Figure 6).

Our document flocking tests were run on sets of documents ranging in size from 200 to 3400 documents in increments of 200 [13]. These experiments show a near sixty-fold increase in performance for the GPU over the CPU (Figure 7) for document flocking. To put this difference in more relevant terms, a month's worth of work becomes a day's worth. Our results highlight the GPU's impressive general use performance. We believe that with continued development, document flocking on the GPU would be an extremely versatile data clustering solution that allows for a practical, small-scale implementation of document clustering, leaving behind traditional limitations of cluster computers.

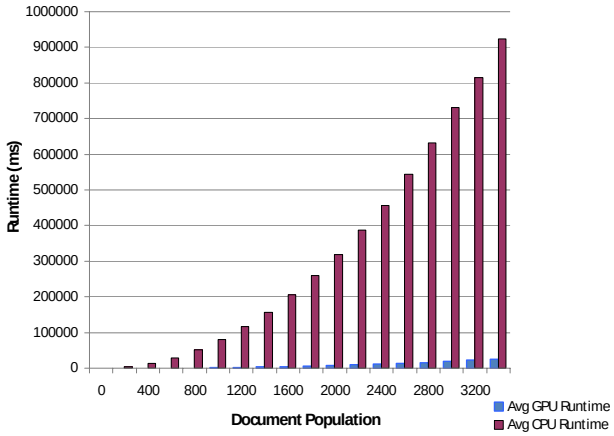

Figure 7 GPU vs. CPU Document Flocking Runtime.

3. REFERENCES

[1] R. Klemm, *Space – Time Adaptive Processing*, The Institution of Electrical Engineers (UK) Press (1998).

[2] R. Klemm, ed., *Applications of Space – Time Adaptive Processing*, The Institution of Electrical Engineers (UK) Press (2004).

[3] W. Burdick, *Underwater Acoustic System Analysis*, Prentice Hall (1984).

[4] P. Tichavsky and K.T. Wong, "Quasi-fluid-mechanics-based quasi-Bayesian Cramer-Rao bounds for deformed towed-array direction finding", *IEEE Transactions on Signal Processing*, **52**(1), 36-47 (2004).

[5] A. Van Buren, "Near-field transmitting and receiving properties of planar near-field calibration arrays", *Journal of the Acoustical Society of America*, **89**(3), 1423-1427 (1991).

[6] M. Viberg and A.L. Swindlehurst, "A Bayesian approach to auto-calibration for parametric array signal processing", *IEEE Transactions on Signal Processing*, **42**(12), 3495-3507 (1994).

[7] A. Nuttall and J. Wilson, "Adaptive beamforming at very low frequencies in spatially coherent, cluttered noise environments with low signal-to-noise ratio and finite-averaging times", *Journal of the Acoustical Society of America*, **108**(5), 2256-2265 (2000).

[8] A.C. Chow, G.C. Fossum, and D.A. Brokenshire, "A programming example: Large Fast Fourier Transform on the Cell Broadband Engine," *IBM*, http://www-01.ibm.com/chips/techlib/techlib.nsf/pages/main, Oct. 2005.

[9] M. Frigo and S.G. Johnson, "The design and implementation of FFTW3," *Proc. IEEE*, vol. 93, no. 2, pp. 216-231, 2005.

[10] R.A. Kerekes and B.V. Kumar, "Multiple target detection in video using quadratic multi-frame correlation filtering," (invited paper) *Proc. of SPIE: Optical Pattern Recognition XIX*, vol. 6977, pp. 697705-1-697705-15, 2008.

[11] X. Cui, T. Potok, "A Distributed Flocking Approach for Information Stream Clustering Analysis," snpd-sawn, pp. 97-102, Seventh ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing (SNPD'06), (2006)

[12] NVIDIA, "NVIDIA CUDA: Compute Unified Device Architecture" NVIDIA, http://developer.NVIDIA.com/cuda, Version 1.1, (2007)

[13] C.W. Reynolds. "Flocks, Herds, and Schools: A Distributed Behavioral Model," Computer Graphics (ACM) 21, pp. 25-34 (1987).

[14] J.St. Charles, T.E. Potok, R.M. Patton, X. Cui, "Flocking-based Document Clustering on the Graphics Processing Unit," *Proceedings of 2nd Workshop on Nature Inspired Cooperative Strategies for Optimization*, book chapter in Springer's Studies in Computational Intelligence (SCI), November, 2007, Acireale, Sicily (Italy).

Proceedings of the SDR '08 Technical Conference and Product Exposition. Copyright © 2008 SDR Forum, Inc. All Rights Reserved

Proceeding of the SDR 08 Technical Conference and Product Exposition. Copyright © 2008 SDR Forum. All Rights Reserved

Proceeding of the SDR 08 Technical Conference and Product Exposition. Copyright © 2008 SDR Forum. All Rights Reserved