Fuzzy Logic System Using Graphics Processing Units - A Survey

[1]Gaurav Dhawan , [2]Sarabjeet Singh
[1,2]Department of CSE, SBS State Technical Campus,Moga Road, Ferozepur-152004, India
[1]dhawangaurav200@gmail.com

***Abstract*— In this paper, a comprehensive survey on various applications of Fuzzy Logic system has been carried out. GPU (Graphics Processing Units) is used for speed up purpose, otherwise commonly used for graphics applications. However, recent trends show use of GPU in various other general computational applications to run them parallel to reduce overall execution time. In this paper, it is discussed that how GPU and fuzzy logic can together be helpful in solving problems of different domains for faster responses.**

***Keywords*— GPU, CUDA, FLS, Fuzzy logic**.

I. Introduction

Latest advancements in parallel computing exploit GPGPUs that have multi-core architecture which supports parallel computations especially required for graphical processing. They devote more transistors for arithmetic and logical operations as compared to data caching and flow control compared to a CPU. Due to processing demand GPUs have advanced rapidly. And also beating the CPUs in terms of number of cores and hence, their computational power. As NVIDIA has launched CUDA software development kit in 2007, the use of GPU's computational powers for general purpose computing has become easy. It gives an API built upon the C language that can be used to write parallel computer programs. The GPU device operates as a coprocessor to the host i.e., CPU, running C program.

The paper is organized in six sections, in section II brief description of GPU is provided. In section III CUDA features are discussed, section IV is about GPU Computing, section V provides information about fuzzy logic system and finally section VI provides various applications where fuzzy logic can be used followed by conclusion.

II. Graphics Processing Units

GPU is a hardware specially designed for highly parallel applications. The GPU's rapid increase in both programmability and capability has spawned a research community that has successfully mapped a broad range of computationally demanding, complex problems to the GPU.

This effort in general purpose computing on the GPU, also known as GPU computing[1] .GPUs have been known to users since quite a long time as a graphics rendering coprocessor to the host PC, to render cool graphic effects in multimedia based applications such as gaming, animation etc. But now the technology inside the GPU has crossed that limit and being used also for many computing applications other than rendering graphics.

The research community has clearly demonstrated how non-graphics-related computing can be performed on the GPUs, with more than a thousands of papers published so far in this field.

So GPGPU is use of GPU computing for general purpose applications.GPU is invented by NVIDIA.GPU computing is the use of a GPU together with a CPU to accelerate general-purpose scientific and engineering applications.CPU controls all the computations.CPU sends tasks and data to GPU, GPU performs computations on data and sends back results to CPU. In this way, in context of GPU computing.GPU is called as DEVICE.CPU is called as HOST.GPUs consist of thousands of smaller, more efficient cores designed for parallel performance. CPUs consist of a few cores optimized for serial processing e.g. Intel Pentium Dual Core processors have 2 cores, Quad core have 4, which are very few .Serial portions of the code run on the CPU while parallel portions run on the GPU. The GPU devotes more transistors for computation. GPUs contain much larger number of dedicated ALUs then CPUs. Each processing unit on GPU contains local memory that improves data manipulation and reduces fetch time.

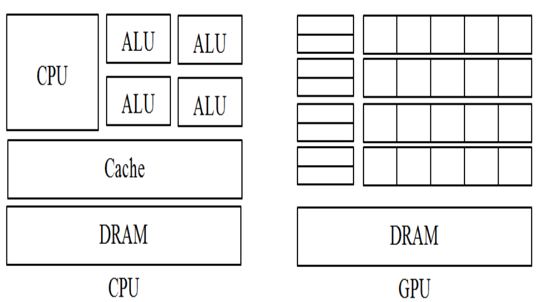

Fig. 1 Difference between CPU and GPU architecture.

III. Compute Unified Device Architecture (CUDA)

CUDA is NVIDIA's solution to access the GPU. Compute Unified Device Architecture (CUDA) is a data-parallel computing environment that does not require the use of a graphics API, such as OpenGL and a shadier language [2] [3]. To work on CUDA, we use the C language .A CPU and a GPU programs are developed in the same environment i.e. C language. In CUDA multiple kernels run simultaneously on a single GPU.CUDA refers to each kernel as grid. A grid is a collection of blocks. Each block runs the same kernel but independent of each other. A block contains threads, which are smallest divisible unit on a GPU. CUDA allows multiple programs, kernels; to run sequentially on a single GPU [4]. The architecture is shown below.

The NVIDIA CUDA architecture is an NVIDIA product that allows easy programming of NVIDIA based GPUs. The CUDA extensions for C allow programmers to access the extreme computational abilities of NVIDIA GPUs without having to map algorithms to graphics concepts such as shaders, vertices, and the like. As NVIDIA CUDA devices are distinctly single instruction multiple thread (SIMT) [5] devices, the CUDA Extensions for C also give programmers and scientists quick access to processors that focus on quickly speeding up data parallel applications. In order to program and utilize these devices well, a thorough knowledge of the hardware architecture and necessary software constructs are vital.

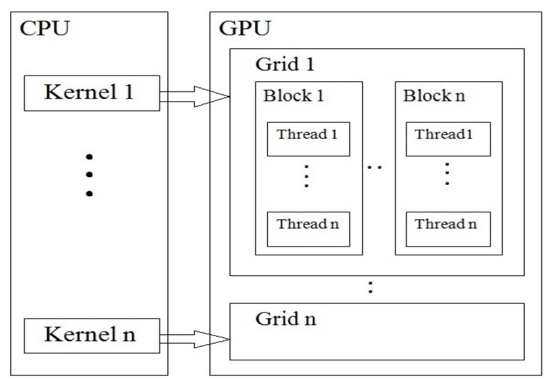

Fig.2 CUDA processing model design. [6][7]

IV. GPU COMPUTING

GPU computing is a promising technology that has become a staple of main stream computational science in the last few years. Worldwide there have been many large installations of GPUs along with plans for even larger installations. The GPU's advantage over the CPU is due to its highly parallel computing architecture with a strong focus on massive data parallelism. This aspect of the GPU originally evolved in order to produce high quality graphics but has also proven to be highly beneficial to scientific computation. Due to the nature of GPUs and their focus on graphics processing, it has only recently become possible for scientists and researchers to use these computer devices through APIs and SDKs such as CUDA and OPENCL. In this work we concern ourselves with the CUDA architecture and its use on NVIDIA GPUs. [8]

V. FUZZY LOGIC SYSTEM

A Fuzzy Logic System (FLS) is able to handle the numerical data and linguistic knowledge simultaneously. FLS can be explained in form of mathematics a linear combination of fuzzy basis function and is a nonlinear universal function approximation. The fuzzy basis function expansion is very useful because its basis functions can be derived from either objective knowledge or subjective knowledge, both of which can be assigned into the forms of IF-THEN rules. Both type of knowledge can be expressed in mathematical manner. There are two types of problem knowledge which can be solved by the Fuzzy Logic System.

1) Objective Knowledge which is used for mathematical models. For example solve the formula, equations of motion for a submarine, spacecraft etc.

2) Subjective knowledge which represents linguistic information that is usually impossible to calculate value using mathematical formulas, the rules that might be valid for tracking a submarine or any other slowly moving large object etc.

FLS is a non-Linear mapping of an input data (feature) vector into a scalar output [9]. Fuzzy logic is a process that tries to simulate the “fuzzy” decision making of a person, by using fuzzy sets. Fuzzy logic often only requires a small number of fuzzy sets and a small collection of simple rules to solve the same problem. In fact, when dealing with a fuzzy problem, computers that operate using fuzzy logic often perform tasks more quickly, efficiently and in many cases better than normal computers which use traditional crisp logic. An example which illustrates the difference between fuzzy and crisp logic is the way in which a computer controls an air conditioner. The normal crisp-logic computer has a sensor that measures the temperature, after which this number is fed into a computer that has some built-in logical rules under which it operates. A fuzzy logic system has three main components:

1) Fuzzifier: A fuzzifier that takes in numbers (in this case the temperature) and transforms them into fuzzy sets.

2) Logic Control Center: A logic control center that uses rules, which are activated by fuzzy sets, and, produces fuzzy sets at its output.

3) Defuzzifier: A defuzzifier that takes the fuzzy output sets and transforms them back into numbers that indicate what action should take place, or decision should be made. Our simple fuzzy air conditioner is governed by two basic rules (real air conditioners would probably be governed by more than two rules), which use the two fuzzy input sets cold and hot. These fuzzy sets describe the temperature. The rules are associated with two fuzzy output sets, high and off which describe the settings for the air conditioner [10].

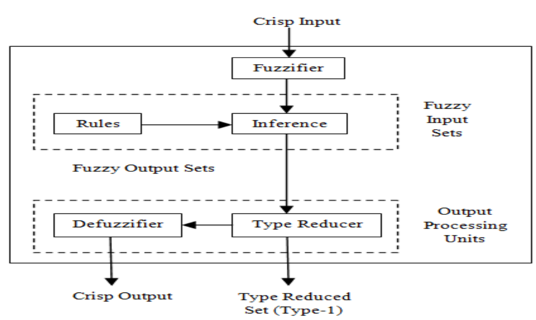

Fig. 2 Block Diagram Representation of an FLS [11] [12]

A. *Type-1 Fuzzy systems.*

Type-1 fuzzy systems consist of inputs fuzzified using fuzzy sets, expert knowledge extracted in the form of fuzzy rule base, inference engine, and defuzzifier as shown in Fig. 2. Type-1 fuzzy sets are incapable of handling uncertainties over uncertainties, i.e., second ordered uncertainty. So keeping in mind another type of fuzzy sets were introduced by Zadeh known as Type-2 fuzzy sets [13].

B. *Type-2 Sets & Fuzzy Systems*

Fuzzy sets models words that are being used in rule base and inference engine. However, word mean different thing to different people and, therefore, are uncertain. Membership degree of a Type-1 fuzzy set cannot capture uncertainties about the words. Hence, another type of fuzzy set, i.e., Type-2 fuzzy Sets, came into existence which is capable of handling such uncertainties. For such a fuzzy set membership value corresponding to some crisp input is not a crisp value rather a Type-1 fuzzy set called secondary membership [14][15]. This concept can be extended to Type-*n* fuzzy sets. Computations based on Type-2 fuzzy sets are very intensive, however, when secondary membership is assumed unity the computational burden reduces drastically. This is another variant to fuzzy set representation and is known as Interval Type 2 fuzzy sets [16] [17] [15].

VI. Fuzzy logic System Application with GPU.

A. *Fuzzy Inference System (FIS) on GPU*

Here GPU is reviewed for speedup up of FLSs which is one of the non-graphics based applications. Derek T. Anderson, *et al*. along with his team investigated this by exploiting inherent parallel nature of FLSs. 128 processing units were operated in parallel thus making intense calculations of constructing rule base and inference process faster as compared to that of CPU [18].

B. *Mamdani FIS*

Derek Anderson, *et al*. here exploited the HPC power of GPU to speedup the inference process inside Mamdani FIS [19]. Various steps of FIS, i.e. fuzzification, implication, aggregation and defuzzification are executed as separate CUDA kernels on GPU.

C. *Fuzzy TSK tuning*

Artificial Intelligence (AI) techniques are too slow to be computed on CPU in real-time. In 2012, Ferreira and Cruz have introduced special approach to offload parts of the AI computations, i.e., automatic training of fuzzy TSK tuning, of a game on to a GPU [20]. In TSK systems consequents for an output which are N-order polynomials are tuned using Batch Least Square (BLS) method and input fuzzy sets are tuned using gradient method.

D. *Fuzzy arithmetic library on GPU*

Fuzzy arithmetic library is introduced by David and Marin as solution to the problems which deals with the uncertainty and complex data representation in the form of integer and floating point [21]. Here with the use of CUDA based GPGPU execution time for basic operations (addition and multiplication) has been improved tremendously.

E. *Fuzzy Logic Based Image Processing*

The real time image processing using simple algorithm is computationally intensive task even with the moderate size images. With further increase in image size it becomes really a difficult task. Anderson *et al.* introduced parallelization of fuzzy logic based image processing where edge computation for each pixel being independent of all other pixels calculation is made parallel. GPGPU implementation using CUDA consisted of two CUDA kernels, one for rule firing and another for defuzzification [22].

F. *Fuzzy Clustering Algorithms*

Fuzzy clustering is one of the unsupervised learning procedures which are helpful in pattern recognition applications. As the number of various clustering parameters increases its computation becomes more and more hard. Anderson *et al*. investigated GPGPU in order to speed up clustering algorithm as it involves various stages and components that are data independent. In this implementation arrays of input data sets are passed from CPU to GPU as a texture [19].

G. *Interval type-2 FLS for robotic navigation*

Type-2 FLSs are comprised of fuzzy sets whose membership values are Type-1 membership functions and called secondary membership functions. Fuzzy computations such as rule implications, aggregation, and defuzzification, etc., become very intensive for ordinary computers [23].

VII. Conclusion

In this paper a representative and concise review of Fuzzy Logic System of various applications was presented. Now's days Fuzzy Logic has got famous for using in various applications as discussed above, due to its ability to handle complexity. In this paper, a representative review of the most recent application of Fuzzy Logic System was given. Fuzzy Logic is gaining popularity due to handling of various uncertainties in the various fields. Fuzzy logic is also helpful to gain the data from the object knowledge and to predict the next result on the bases of previous knowledge.

References

[1] J. Sanders and E. Kandrot, CUDA by Example: An Introduction to General-Purpose GPU Programming. Addison-Wesley Professional, 2010.

[2] C.-F. Juang, R.-B. Huang, and Y.-Y. Lin, "A Recurrent Self-evolving Interval Type-2 Fuzzy Neural Network for Dynamic System Processing," *IEEE Transactions on Fuzzy Systems*, vol. 17, no. 5, pp. 1092–1105, 2009.

[3] P. Melin and O. Castillo, "A Review on Type-2 Fuzzy Logic Applications in Clustering, Classification and Pattern Recognition," *Applied Soft Computing*, vol. 21, pp. 568–577, 2014.

[4] C.-H. Lee, F.-Y. Chang, and C.-M. Lin, "An Efficient Interval Type-2 Fuzzy CMAC for Chaos Time-Series Prediction and Synchronization," *IEEE Transactions on Cybernetics*, vol. 44, no. 3, pp. 329–341, 2014.

[5] Aisbett, J.; Rickard, J.T., "Centroids of Type-1 and Type-2 Fuzzy Sets When Membership Functions Have Spikes," *Fuzzy Systems, IEEE Transactions on* , vol.22, no.3, pp.685,692, June 2014
doi: 10.1109/TFUZZ.2014.2306973

[6] Owens, J.D.; Houston, M.; Luebke, D.; Green, S.; Stone, J.E.; Phillips, J.C., "GPU Computing," *Proceedings of the IEEE* , vol.96, no.5, pp.879,899, May2008 doi: 10.1109/JPROC.2008.917757

[7] R. H. Luke III, D. Anderson, J. M. Keller, and S. Coupland, "Fuzzy logic-based image processing using graphics processor units." In *IFSA/EUSFLAT Conference*, 2009, pp. 288–293.

[8] Owens, John D., et al. "GPU computing." *Proceedings of the IEEE* 96.5 (2008): 879-899.

[9] Mendel, Jerry M. "Fuzzy logic systems for engineering: a tutorial." *Proceedings of the IEEE* 83.3 (1995): 345-377.

[10] M. A. Martin and J. M. Mendel, "Flirtation: A Very Fuzzy Prospect: A Flirtation Advisor," Journal of Popular Cult., XI (1), pp. 1–41, 1995.

[11] A. Khosravi, S. Nahavandi, D. Creighton, and D. Srinivasan, "Interval Type-2 Fuzzy Logic Systems for Load Forecasting: A Comparative Study," *IEEE Transactions on Power Systems*, vol. 27, no. 3, pp. 1274–1282, 2012.

[12] E. A. Jammeh, M. Fleury, C. Wagner, H. Hagras, and M. Ghanbari, "Interval Type-2 Fuzzy Logic Congestion Control for Video Streaming Across IP Networks," *IEEE Transactions on Fuzzy Systems*, vol. 17, no. 5, pp. 1123–1142, 2009.

[13] L. A. Zadeh, "The Concept of a Linguistic Variable and Its Application to Approximate Reasoning". *Information Sciences*, vol. 8, no. 3, 1975, pp.199-249.

[14] N. N. Karnik and J. M. Mendel, "Operations on Type-2 Fuzzy Sets, " *International Journal on Fuzzy Sets & Systems*, vol. 122, pp. 327–348, 2001

[15] J. M. Mendel, R. I. John and F. Liu, "Interval Type-2 Fuzzy Logic Systems Made Simple", *IEEE Transactions on Fuzzy Systems*, vol. 14, no. 6, 2006, pp.808–821.

[16] O. Castillo and P. Melin, *3 Type-2 Fuzzy Logic*. Springer, 2008.

[17] M. Khosla, R. K. Sarin, M. Uddin, and S. Singh, A. Khosla, "Realizing Interval Type-2 Fuzzy Systems with Type-1 Fuzzy Systems", in *Cross-Disciplinary Applications of Artificial Intelligence and Pattern Recognition: Advancing Technologies*, IGI Global, Hershey, Pennsylvania, USA, 2012, pp. 412--427.

[18] N. Harvey, R. Luke, J. M. Keller, and D. Anderson, "Speedup of Fuzzy Logic Through Stream Processing on Graphics Processing Units, " in *IEEE Congress on Evolutionary Computation, 2008*, pp.3809–3815.

[19] D. T. Anderson, R. H. Luke, and J. M. Keller, "Speedup of Fuzzy Clustering Through Stream Processing on GraphicsProcessing Units, " *IEEE Transactions on Fuzzy Systems*, vol. 16, no. 4, pp. 1101–1106, 2008.

[20] B. B. Ferreira and A. J. Cruz, "A Parallel Method for Tuning Fuzzy TSK Systems with CUDA, " *SBC–Proceedings of SB Games, Brazilian Computer Society (SBC)*, pp. 5–8, 2012.

[21] D. Defour and M. Marin, "Fuzzy GPU: A Fuzzy Arithmetic Library for GPU, " in Parallel, Distributed and Network-Based Processing (PDP), 2014 22nd Euromicro International Conference on. IEEE, 2014, pp.624–631.

[22] R. H. Luke III, D. Anderson, J. M. Keller, and S. Coupland, "Fuzzy Logic based Image Processing using Graphics Processor Units", in *IFSA/EUSFLAT Conference*, 2009, pp.288–293

[23] T. Kumbasar and H. Hagras, "A Type-2 Fuzzy Cascade Control Architecture for Mobile Robots," in *2013 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2013, pp. 3226–3231.