A Study of Massively parallel computer architecture and its application in neural networks.

Report By

Lakshmi Harshini Kuchibhotla

SUID: 230997383

Bhargav Rahul Surabhi

SUID: 264072521

Contents Page

ABSTRACT

Due to the advancements in the technology, the requirement for Massively Parallel Processors (MPPs) expanded quickly to enhance the execution and effectiveness of the implementations. Massively parallel computer architecture is a computer system, which can run a similar program on numerous processors utilizing different threads. Each processor has its own OS & memory. Due to the high level of parallelism shown by MPPs, this procedure is utilized in numerous innovative areas. One such emerging utilization of massively parallel PC designs is Neural systems. Neural networks, at real-time requires high speed computers which can provide very high efficiency and performance.

SpiNNaker - Spiking Neural Network Architecture, a massively parallel computer architecture design was made to mirror the actual working of the human brain. Its goal was to transform the regular conventional super computers into a vitality productive and energy efficient MPP systems. The primary objective of SpiNNaker was, to reproduce the real time behavior of human brain by reenacting various complex structured neurons that talk by means of spikes which are not reliable.

REMAP - Real-time Embedded Modular Adaptive Parallel processor, an exceptionally parallel architecture gives us a research area for future real-time action-oriented systems. Remap consists of communicating modules. These are generally the control systems which collaborate with the environment utilizing some high-speed actuators and sensors. Action-oriented systems request an abnormal level of parallelism so that they can communicate, learn and adjust to different environments.

In this study, we discuss two different computer architectures used for biologically inspired artificial neural networks: SpiNNaker and Remap and also compare the positives and negatives of both architectures. This study extends to the details of the hardware, the mapping algorithms used, and the inter-processor communication mechanism implemented.

1. INTRODUCTION

This paper provides a survey of the design and use of the ANN (artificial neural networks) using massively parallel computers. This paper identifies the importance of the architectural point of view during the design of such ANNs. The paper discusses the type of network used for communication and the algorithms used for it.

The paper is primarily organized into 3 parts. First part describes the problem that the architecture is designed to solve. We also discuss about the common characteristics of such architectures and their parallel implementation model. The second part discusses in detail about the SpiNNaker project with respect to the characteristics mentioned above and about the REMAP project that also is an ANN designed using massively parallel processors with high speed inter-processor communication. The third part discusses the comparison between these and conclusions we can draw from the two architectures.

1.1 BACKGROUND

An ANN is just a representation of biologically inspired algorithms its very far from the actual working of a brain. It's very important to understand the working of the human brain to simulate it and to understand the simplicity of our architectures and models as compared to the brain.

A lot of research and biological discovery was done on the brain to realize that the basic component is something called a neuron. The research suggests that there are huge number of such neurons in our brain. Brain imaging of an average human represents the activity of such neurons at various parts of the brain.

The basic parts of a neuron are dendrites (input), cell body, and an axon (output). The output, an axon is responsible to connect various neurons and the communication is done using synapses (like a chemical resistor).

The Artificial neuron is a very simple model consisting of levels/values that correspond to various impulse frequencies. Then a summation of such impulse frequencies is done to get the final information. Another aspect of the brain is its ability to learn new things. This happens by changing the structure of the brain to respond to such synapses or by changing the synapses. The former is a long-term adaptation e.g. language.

1.2 REMAP

As the evolution of hardware took place, continuous research on human-like capabilities for computers was also on the rise. The development of the semiconductors, ICs and transistors

resulted in a massive increase in the calculation speed. Despite such improvement, there was little progress in the fields related to AI and image processing etc.

It's only during the 80's that the AI actually exploded making it a very trendy topic for researchers from various fields. Also, at that time various commercial and massively parallel processors were readily available. But such devices also failed to recognize issues such as real-time determinism, power consumption, heterogenous communication and physical size.

The architecture discussed below aims towards eliminating such issues. REMAP (Real-time, Embedded, Modular, Adaptive, Parallel processor project) is a new system architecture that is highly parallel with communicating processing elements. REMAP is a reconfigurable bit-serial SIMD processor array. These PE's are reconfigurable to support different kinds of algorithms for mapping the ANN.

1.3 SpiNNaker

SpiNNaker, a massively parallel computer architecture design was created to replicate the actual working of the human brain. Its goal was to transform the regular conventional super computers into a vitality productive and energy efficient MPP systems. The primary objective of this design was to reproduce the real time behavior of a human brain by reenacting various complex structured neurons that talk by means of unreliable spikes.

Instead of using an artificial neural network spinnaker uses a SNN model (spiking neural network) to make it a very close abstraction of real neural networks. A lot of natural disclosure was done on the fundamental bit of the mind "the neuron". With brain imaging, we will in general, may discover how the movement is managed inside the mind. The brain is made from extensive scope of neurons and it is very intense to understand the natural model. So, we are required to recreate the virtual model to know it. A research group known as the Advanced Processor Technologies (APT) at the University of Manchester, basically focusses on the exploration for cutting edge and novel methodologies for registering & process. This group had a with progress built up the headsail venture. This group had developed this design for large scale computing in Dec 2009.

2. ANALYSIS

2.1 SpiNNaker

2.1.1 Hardware

The spinnaker has multiple nodes in an array these nodes are run in parallel for processing. Each of these nodes contains eighteen ARM 968 processor cores with a local memory of 96KB and shared memory of 128 MB. The shared memory is SDRAM. Each node is also referred to as a CMP (chip multi-processor) die. Each node is capable of spiking 1000 neurons. The architecture is extremely scalable with the smallest configuration having 1 chip and the largest with up to 65,536 chips[3].

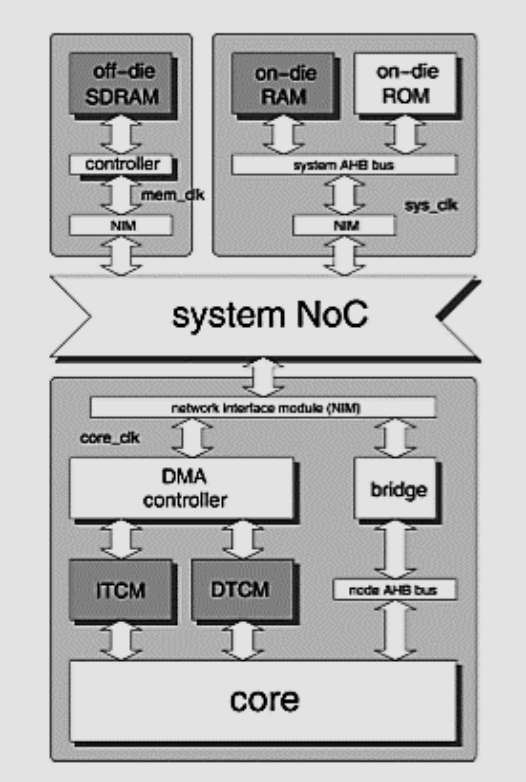
The design of SpiNNaker makes the assumption that the cost of the processors is basically free in comparison to the costs of keeping them at a low temperature. This is the reason for the inclusion of the ARM986 embedded chip. As it delivers decent performance while consuming low energy. Also, self-timed communication channel is preferred for its energy efficiency. The total estimation for the system (with million processors) is around 90kW[6].

The Spinnaker CMP is a GALS system with the 9-series ARM processor, each inside a synchronous island that use an asynchronous packet-switched communications infrastructure. A CHAIN technology is implemented that takes care of the on-off communications. Each of the CMP is also capable of connecting to the world using a 100Mbit ethernet interface.

The communications are handled by different systems altogether called the Comms NoC and the The communications are handled by different systems altogether called the Comms NoC and the System NoC. The Comms provides communication from any processor to any other processor in the system. The systems NoC takes care of chip wide communication using a 32kB RAM, 32kB boot ROM, a timer and the ethernet interface. It also provides a channel to communicate to the 128MB private storage assigned to each CMP[5].

The ARM968E-S is a 32bit processor that is designed for low power, data intensive applications. Every processor has a dual banded 64kB data TCM and a 32kB ITCM (Instruction Tightly coupled memory).

Using SpiNNaker to make a simulation of 10^9 neurons and 10^{12} synapses, we require around 10^6 processing cores running at 200Mhz. The size of the network also varies depending on various neuronal models with different complexities such as the Izhikevich model and Hodgkin-Huxley model. The advantage of a Neural reenactments expect memory to store neuron models, state data, and the consequences of neural calculation.



SpiNNaker memory organization

As the neurons are refreshed occasionally, neuron state is kept in a quick access memory at the processor centers. Synaptic information, then again, is related with spike sources and is required just when a spike arrives. Given that synaptic occasions are not prepared instantly, the synaptic data can be recovered from a bigger, slower memory utilizing DMA. This bigger memory is regular to all centers in a chip, be that as it may, neural recreations needn't bother with memory soundness, subsequently maintaining a strategic distance from the requirement for shared memory and the comparing memory-access instruments.

2.1.2 Network on Chip Simulators

Usually the Soc /processor development involved using dual and quad core processors chips but recently the trend moved towards usage of chips with up to 100s of cores which are inter-connected using a NoC. The network on chip has to manage the connections efficiently and effectively.

They used the NS-2 chip for simulating as it very closely resembles actual computer networks. In SNN, one neuron is normally associated with numerous others. Accordingly, immense measures of one-to numerous correspondences must happen between handling hubs. In this way, multicast-empowered directing looks very productive for the reenactment of SNN, which is additionally demonstrated by recent researches and advances in on-chip communications. For instance, Ginosar and Vainbrand demonstrated that multicast work NoC gives the most astounding execution/cost

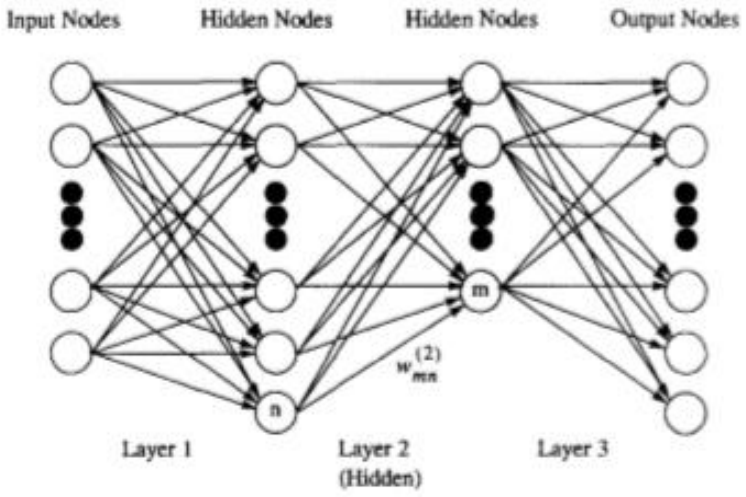
proportion between normal topologies, and thusly it is a standout amongst the most appropriate interconnect structures for configurable neural system usage. What's more, to make one to many communications for SpiNNaker, a multicast component is accommodated.

Noxim 2 is created in c++ library that allows the design of digital systems. It enables the client to give unique parameters and create his own 2D mesh NoC. The assessment of NoCs as far as throughput, dormancy and power utilization are done in Noxim 2. ORION, a simulator made specifically and essentially to the estimation of intensity and space for Network on chip structures. New semiconductor innovation is helped by models of capacitances and transistors taken from the industry.

Existing neural framework simulators bolster definite or basic portrayals of the neurons/neurotransmitters to shape the weighted and coordinated diagram. They likewise give programming interfaces to clients to create SNN models[4]. Typically, the model portrayal works on the populace (gathering of similar/same neurons)/projection (heap of mono associations between populaces) level instead of on the single neuron/association. This technique isn't only for improvement simplification. It additionally agrees with the genuine association of human minds. On close inspection, the cerebrum is a uniquely organized framework, made of various zones that contain neurons which are recognized for their utilitarian, anatomical and physiological properties.

2.1.3 Learning Algorithm

The simplest algorithm, back-propagation algorithm is a very popular method for solving real-world problems. One limitation of this algorithm is, it takes a very long time to adjust all the weights in order to respond to the required input pattern. That is, it takes a larger push for the system to learn and adapt. [3]



Back Propagation Algorithm

Learning is an essential part of any Parallel Distributed Processing (PDP) algorithm and specifically the BackProp type. The premise of advancements throughout the years was due to the straightforward 'delta' learning rule owing to Widrow and Hoff in 1960. Tragically, this standard necessitates that the yield of a unit be contrasted and its normal esteem, giving a mistake factor. In any case, when a framework has concealed units it ends up difficult to apply these tenets, as there are no normal qualities for the yield of the shrouded units. Therefore, a summed-up delta rule has been produced by Rumelhart, Hinton and Williams,15 which takes into account shrouded units in a feedforward arrange (no input association). This strategy is gotten back to spread (BackProp) and is connected here to systems whose unit info and yield esteems lie somewhere in the range of 0 and 1.

The DPN is extremely basic in its task. The present state (or instruction) directs the state machine which forms a data stream. Changes to the condition of the DPN (instructions) are inserted in the data stream and are discernable from data components. In the light of this method of activity, there is no requirement for any put away program system.

Control of the DPNs is an imperative part of the usage of the design. The algorithm for playing out a cycle of a BackProp simulation is settled for a given size of neural system, henceforth the undertakings to be performed by the DPNs are settled and deterministic. The issue of control inside a DPN has been considered in view of this deterministic nature. The control work is given by sprinkling instructions amidst the data stream.

At the point when a DPN gets another instruction it stores it and afterward over and again complies with that instruction on each new data thing it gets until the point when another instruction is gotten. This implies for instance that the DPN does not have to know what number of data things are required to be prepared; this is controlled by the customary processor, which sends another instruction at the right time. So as to reproduce the BackProp algorithm, it isn't constantly alluring to have each hub comply with each instruction (for instance when requesting that a hub restore an esteem).

The component to accomplish this is to give each DPN a novel personality. The personality won't be a settled trait of the DPN however will be allocated toward the beginning of the simulation by the utilization of a 'set character' (RSET) instruction. This likewise effectively informs the customary processor of the quantity of DPNs as of now designed in the ring.[3]

2.1.4 Communication

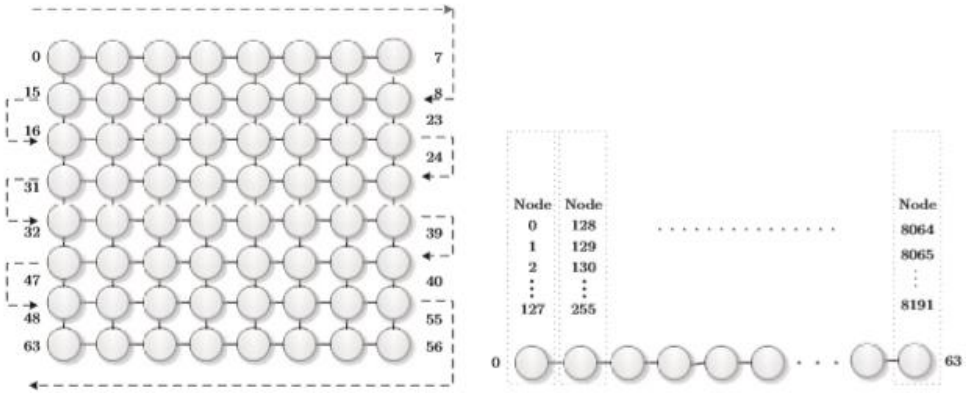
NoC is a typical and developed innovation that interfaces IPs on chip altogether[4]. Accordingly, we need to concentrate on few highlights which are specific to Spiking neural networks. We know that a single neuron is commonly associated with numerous others. In this manner gigantic measures of one-to many interchanges must occur between preparing hubs. Consequently, multicast-empowered routing seems to be exceedingly efficient for the reproduction of neural systems. This is explained in the recent works. For instance, Ginosar and Vainbrand demonstrated

that out of all the basic topologies, the multicast work NoC gives the most noteworthy execution/cost proportion and thus it is a standout amongst the most reasonable interconnect models for configurable neural system usage.

Also, for SpiNNaker, a multicast component is accommodated that is efficient to numerous interchanges. Besides, as a contextual analysis, we center around the work, that is done by NoCs in light of the fact that the topology is generally utilized by CMP items. Likewise, the tree-based disseminated routing technique is taking as a base for multicasting. Contrasted and the source routing system, it was shown that less capacity over heads was achieved using the disseminated routing. Besides, there are many surely understood multicast routing strategies having a place with this class. For tree-based routing, we perform multicast using a typical way and then sends (recreates) the message when important to accomplish a negligible course to every goal. At each jump, the switch will finish comparing activities dependent on the source ID of the approaching parcel. Naturally, the X-Y routing technique is utilized for each single message to maintain a strategic distance from stop. Uncommonly, a two-level routing technique is utilized, and we give the layout here. As referenced in Subsection 3.1, SNN models of programming test systems ordinarily speak to groups of single associations (projections) between populaces. Along these lines it is beneficial to disseminate however many as could be expected under the circumstance’s neurons of a populace into one center, or into a few adjacent centers on the off chance that one can't involve every one of them. This methodology can diminish correspondence overheads.

As needs be, we accept the populace ID which acts as the lookup key for the routing tables. Then, the spike at that point can be coordinated which means copied first and coordinated later, to the objective arrangement of hubs means, from the part of SNN, to a minimum of one center that possess the neurons from the associated populace. Center is taken as our second dimension. When a spike is received, the neurons around the objective center will check whether this spike ought to be managed without anyone else's input or not, since a single hub contains neurons from various populaces. This can be accomplished by seeing into a neighborhood table. The ID of the source-neurons acts as the key for this table which is conveyed by the approaching parcel as a payload. Finally, we need to fill the routing tables, which relies on the procedure of mapping SNN onto NoC. A linear mapping calculation is utilized as a strategy of the SpiNNaker. Initially, we renumber all the neurons such that all the IDs of these neurons will be constant for a given solitary populace. Further, all the neurons are consistently assigned to NoC hubs all together. Therefore, all neurons of a populace will be conveyed into one center or into a few close-by centers.

Likewise, the lookup key of routing tables is taken from the populace ID. A spike at that point can be coordinated - first copied then coordinated- to the objective arrangement of hubs (or from the part of SNNs, to at least one centers that possess the neurons of the associated populace). The center consists of a second which. After receiving a spike, neurons in the objective center will verify if it ought to be managed without anyone else or not, as one hub may contain neurons from numerous populaces.



Mapping neuron nodes to NoC.

It is accomplished by seeing into a neighborhood table. The approaching parcel as the payload conveys the source-neuron ID which is the key for this table. Finally, we have to fill in routing tables, that vastly relies upon the procedure of mapping SNN onto NoC. Similar to SpiNNaker's strategy, a linear mapping calculation is utilized. Firstly, we re-number all the neurons such that the IDs of all neurons in a independent populace will remain persistent. Furthermore, neurons are consistently assigned to NoC hubs all together. Subsequently, all neurons of a populace will be conveyed into one center or into a few close-by centers.

2.2 REMAP

2.2.1 Hardware

The larger part of massively parallel processors use bit-serial arithmetic, and that is additionally the essential method of activity for our own exploration machine, REMAP. Accordingly, we might want to investigate the calculations down to bit-level. For the larger part of tasks utilizing bit-serial PEs, the preparing times develop straightly with the information length utilized. E.g. an opportunity to complete a bit-serial expansion is indistinguishable time from to peruse the operands and store the outcome (3 cycles). This might be viewed as a genuine disservice (example, when utilizing floating point values of either a 32-bit or 64-bit), or as an appealing component (utilizing low accuracy information accelerates the calculations as needs be). Regardless, bit-serial information ways streamline correspondence in massively parallel PCs.

The REMAP architecture is more like a guideline of what an ANN should be like rather than an implementation. Although, there is an implementation of the REMAP architecture that was explored. This architecture used FGPAs to implement the parallel computer.

The processors are implemented in a Xilinx XC4005 circuits and the serial/parallel I/O device in Xilinx XC3020(8 parallel and 8 serial I/O each). These Control units use an AMD 28331 and

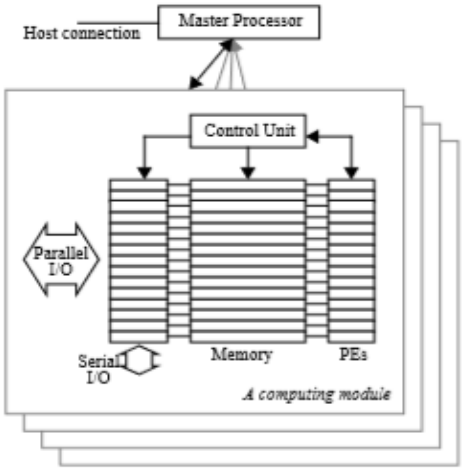
AMD 28332 as they are micro-programmable. Although the unit is more towards general purpose it serves the need of our computational requirements well enough. [9]

The processors implemented a counter instead of a multiplier that used a carry-save adder to reduce the multiplication time. The control unit has a space of 8kB to store the micro programs assigned to it.

The Xilinx circuits composes of input-output blocks, combinatorial logic blocks and an interconnection network. These CLB, IOB and ICN are configurable and reprogrammable. The RAM can be stacked from an external memory or from a microchip.

Tragically, not very many bit-serial PCs have bolstered for bit-serial augmentation, and without such, duplication time develops quadratic with the information length. Be that as it may, with an incorporation of a bit-serial multiplier [9, 44, 55] the duplication of useless numbers can be performed in cycles (i.e. an opportunity to peruse the operands and store the outcome).

The fundamental work for the control unit is to pass instructions together with PE memory that delivers to the PE cluster. In the meantime, it registers new location esteems (normally augmentations and decrements). The control unit presently being used [3] has been structured around a micro-programmable sequencer and a 32bit ALU (AMD 28331, 28332). The control unit is equipped for conveying another location together with another guidance each 100ns. The controller is more universally useful than typically required, however until the point that we comprehend what is required it fills our need.



Remap System – Control Unit

The microprograms to be executed by the control unit are put away in a 8K words control store. The operations can either be straightforward field operations, such as including two fields, or entire calculations like an ANN calculation. For the minute just, a miniaturized scale code constructing agent is accessible to program the control unit.

2.2.2 Learning Algorithms

The algorithms that can be used are discussed below with their usage and drawbacks when necessary

A Feedback algorithm which is a straightforward processing element exhibits with communicate or ring correspondence might be utilized productively additionally for feedback systems (Hopfield nets, Boltzmann machines, repetitive backpropagation nets, and so forth.). The MCPS measures are, obviously, equivalent to above. 100 emphases of a 1024-byte input design takes 106ms on a 1024 processing element sample that clocks at 25 MHz [1].

Nordström, describes distinctive approaches to execute Kohonen's Self-Organizing Maps on MPPs [1]. The Self-Organizing Maps calculation needs an info vector that will be circulated to the around hubs and contrasted with their vector weights. These will be efficiently executed by using the communicated and straightforward processing elements structures. The resulting look for least is amazingly efficient on the serial-bit processor clusters. Deciding the area of final refresh part should again be possible by communicated and separation computations. Hence, additionally for this situation, communicate is sufficient as the methods for correspondence. Hub parallelism is, once more, easy to use. Efficiency proportions of over 80 percent are gotten – which is the quantity of tasks every second separated by most extreme no. of activities every second accessible on the PC).

Meager Distributed Memory (SDM), is a two-layer feedforward arrangement created by Kanerva [Kanerva, 1988][9]. However, it is all the more frequently – and all the more advantageously – depicted as a PC memory. It has an immense location space (ordinarily 10300 conceivable areas) which is just scantily (obviously) populated by real memory areas. Keeping in touch with one area that influences other regions in it (e.g. in the Hamming-remove regard) and, , a few neighboring areas add to the outcome when perusing from memory.

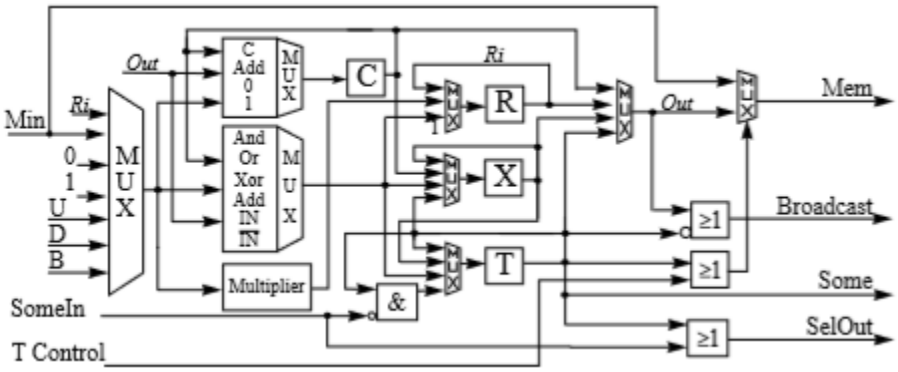
The SDM calculation necessitates appropriation of the relative reference address, refresh, or readout, correlation and separation figuring, and summation, of counters at those specified regions. Nordström [Nordström, 1991a] identified a "blended" mapping (exchanging among hub and weight parallelism in various parts of the count) that is particularly efficient which in-turn is a pre-requisite for these activities.

In the bit-serial-PE based design depicted above, there is a counter instead of a multiplier that makes the sample specifically productive for SDM. A 256-processing element REMAP model with counters instead of multipliers is found to run SDM at a speed 10 - multiple times quicker than a 8K Processing Elements CM-2 (clock frequencies evened out). As of now without counters (at that point the PEs turn out to be greatly basic) 256 Processing elements REMAP beats multiple times greater CM-2 by a factor of 4– 10. One clarification of this is the more created control unit of REMAP that makes the blended mapping conceivable to utilize [2].

2.2.3 Communication

The definite investigations of artificial neural system calculations have brought about a proposition for a PE that is appropriate for this area. Vital highlights include – 1) a bit-serial multiplier 2) broadcast connection. Strikingly, we don't need any between PE associations which communicates with the closest neighbor are required. The Processing Element is very broadly useful, and we are confident this is a valuable PE plan additionally in a few other application regions. In this variant it comprises of four flipflops namely C, R, X and T, 8 multiplexers, a rationale and an increase unit. In general, these units gain the power flags specifically from the small-scale guidance word which is sent from the control unit. In general PEs, without the help for augmentation, the duplication time develops quadratically with the information length. A technique dependent on carry and save adders can diminish the augmentation time required to an opportunity to stack the operands and store the outcome.

The usage of a counter is better than a multiplier in the PE configuration as it would satisfy well while actualizing the SDM neuromorphic system. A 256PE REMAP acknowledgment with counters is found to run SDM at paces 10– multiple times of a 8K Processing Element CM-2, (with frequencies standardized and on a 8K issue). As of now without counters (at that point the PEs turn out to be amazingly straightforward) a 256PE REMAP beats multiple times greater CM-2 by a factor of 4– 10[8].



PE Model

Processing element can communicate with other units using 3 different ways - closest neighbor and communicate correspondence. PE(n) peruses from PE(n-1) and PE(n+1) that means, the closest neighbor correspondence organize enables every PE to peruse its neighbor's. The former and the latter PEs are viewed as their neighbors. Whenever any of these PEs communicate, an incentive to every other PE or to the control unit. This control unit could likewise communicate an incentive to the PEs. It has additionally a probability to check if any of the PEs has the movement bit (T-flip-flop) set. On the off chance that few PEs are dynamic in the meantime and the control

unit needs one PE to communicate, the control unit basically completes a choose the first task, that selects the initial dynamic Processing Element and deselects the remaining. This correspondence and assertion tasks shall be utilized to perform productively the grid calculations and additionally scan and test activities sufficient for some application regions, particularly artificial neural systems. To be helpful progressively applications which incorporate connecting with an evolving domain, levels of popularity are put on the I/O system. To accomplish all these requests, the MPP cluster is furnished with 2 I/O-channels, 1 for 8-bit correspondence and second one to exhibit wider interchanges. This interface has a capacity to keep running at velocities up to 80MHz (burst) which, for a 256PE cluster, infers a most extreme exchange rate of 20Gbit/s. Because of confinements in the control unit the I/O-interface as of now keeps running at 10MHz which diminishes the exchange rate to 2.5Gbit/s[7].

3. CONCLUSION

3.1 Parallel Computers and it's relation to ANN

For all intents and purposes any ANN calculations that are dependent on training examples for parallelism can be done very well on every parallel computer. This sort of computation gets the performance level to a very high level. This obviously gets intriguing for all the computer architecture enthusiasts who want to perform research on training and learning algorithms. In any case, performing this sort of a training cannot be done in real-time as the performance upgrades would be bottle-necked by other places. If the communication hardware meets the expectations and escape the bottle-neck we can achieve real-time like performance, A noteworthy end from this overview is that the SMID structures fit flawlessly for consistent ANN computations. The greater part of ANN calculations following the most prevalent models of today can be mapped rather proficiently onto existing structures. In any case we require very massive and and exceptionally fit processing models for a custom fitted application Eg. SDM.

3.2 Communication

Communication and ring correspondence could be used in ANN calculations in a productive way. Communication or ring correspondence alone has turned out to be adequate in massively parallel machines. For massively parallel machines, without utilizing preparing model parallelism it is hard to utilize just communication channels. On a two-dimensional work machine, communication in one course moment might be utilized and hub and weight parallelism might be reduced to a great extent. Accordingly, the communication channels introduced is extremely valuable to determine the performance of such machines as their parallel processing nature demands such inter connectivity. The "highways" of the DAP design fill this need.

3.3 Mapping ANN to the Algorithms and their respective Architectures

As we can understand from the above two implementations it is clear that identifying the best algorithm for mapping depends heavily on the kind of parallelism that required. As we can see both of the above discussed algorithms are very close to each other. And they depend on a concept of node-weight parallelism. This is very natural to most of the researchers as each node represents every neuron, but this basic mapping becomes inefficient I case of varied sized node layers.

4. FUTUREWORK

The rise of multi-modular concept for development and design of ANN architectures helps the use of various SIMD and VLSI processors. Also, the above implementations are among the very first in their respective application. Further survey on such architectures will definitely reveal much more information about how an ANN should behave.

The University of California Berkley is as of now dealing with demonstrating a connection oriented super computer that displays different neural computations, that can be used as a significant apparatus for numorous exploration attempts to be done by the analysts [5].

As a critical part of things to come. the SpiNNaker research engineers could perform more research and study could be done on assessing the framework under different training models (cut up framework, restricted models with high chance of failure), and different activity models that would be great for the spatial interfacing framework, which can support high rate communication among different spikes.

A great platform to start for future work is to provide a neural system machine that can hold more than a million processors with very high bandwidth communication channels. All of this while keeping in mind the application of such to be a more extensive neural network. The applications for such massive devices include real-time data analytics and machine learning to provide great and useful insights on how learning can be approached.

5. REFERENCES

[1] Nordström T. B. Svensson and "Using and designing massively parallel computers for artificial neural networks." (Research Report No. TULEA 1991:13), Luleå University of Technology, Sweden, 1991.

[2] B. Svensson, Nordstrom. T, P.-A Wiberg, K. Nilsson. “Towards modular,massively parallel neural computers”, Connectionism in a Broad perspective: Selected papers from the Swedish conference on connectionism – 1992, L. F. Niklasson and M.B. Boden Eds. Ellis Horwood, pp. 213-226, 1994.

[3] Wei-Min Zheng, Yu Ji, You-Hui Zhang, “Modelling Spiking Neural Network from the Architecture Evaluation Perspective” Journal of Computer Science and Technology Jan-2016.

[4] David R. Lester , Steve B. Furber, Jim D. Garside, Luis A. Plana, Steve Temple, and Andrew D. Brown, Eustace Painkras, “Overview of the SpiNNaker System Architecture” IEEE transactions on computers, vol. 62, no. 12, December 2013.

[5] Xavier Lagorce, Evangelos Stromatias, Francesco Gallupi, Luis A. Plana, Shih-Chii Liu, Steve B. Furber and Ryad B. Benosman. “Breaking the millisecond barrier of SpiNNaker: implementing the asynchronous event-based plastic models with microsecond resolution”.

[6] Evangelos Stromatias, Francesco Gallupi, Cameron Patterson and Steve Fuber, “Power analysis of large-scale, real time neural networks on SpiNNaker”.

[7] Rumelhart, D. E. and J. L. McClelland. Parallel Distributed Processing; Explorations in the Microstructure of Cognition. Vol I and II, MIT Press. Cambridge.

[8] Svensson, B. ”Implementation and application of a software configurable massively parallel computer.” Second Swedish Workshop on Computer Systems Architecture, Bålsta, Sweden.

[9] Obermayer, K., H. Ritter and K. Schulten. "Large-scale simulations of self-organizing neural networks on parallel computers: application to biological modelling." Parallel Computing. Vol. 14(3): pp. 381-404.

[10] Svensson, B. and T. Nordström, “Execution of neural network algorithms on an array of bit-serial processors,” in 10th International Conference on Pattern Recognition, Computer Architectures for Vision and Pattern Recognition, Atlantic City, NJ, USA, vol. II, pp. 501-505.

[11]Linde, A., T. Nordström and M. Taveniku, “Using FPGAs to implement a reconfigurable highly parallel computer,” Field-Programmable Gate Array: Architectures and Tools for Rapid Prototyping; Selected papers from: Second International Workshop on Field-Programmable Logic and Applications (FPL'92), Vienna, Austria, H. Grünbacher and R. W. Hartenstein Eds. New York: Springer-Verlag, pp. 199-210.

Appendix

A Study of Massively parallel computer architecture and its application in neural networks.

REPORT BY

LAKSHMI HARSHINI KUCHIBHOTLA

SUID: 230997383

BHARGAV RAHUL SURABHI

SUID: 264072521

ABSTRACT

Due to the advancements in the technology, the requirement for Massively Parallel Processors (MPPs) expanded quickly to enhance the execution and effectiveness of the implementations. Massively parallel computer architecture is a computer system, which can run a similar program on numerous processors utilizing different threads.

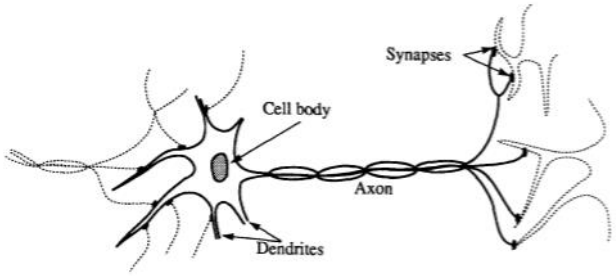
Each processor has its own OS & memory. Due to the high level of parallelism shown by MPPs, this procedure is utilized in numerous innovative areas. One such emerging utilization of massively parallel PC designs is Neural systems. Neural networks, at real-time requires high speed computers which can provide very high efficiency and performance.

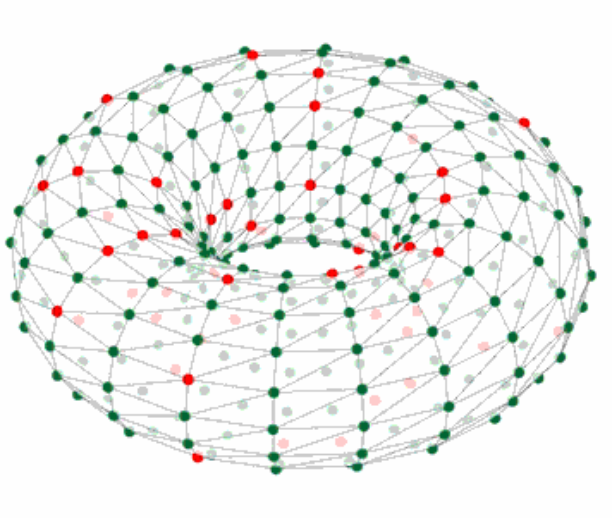
In this study, we discuss two different computer architectures used for biologically inspired artificial neural networks: SpiNNaker and Remap and also compare the positives and negatives of both architectures. This study extends to the details of the hardware, the mapping algorithms used, and the inter-processor communication mechanism implemented.

Introduction

The paper is organized largely into three parts.

- The first part describes about the problem that the architecture is designed to solve. We also discuss about the common characteristics of such architectures and their parallel implementation model.
- The second part discusses in detail about the SpiNNaker project with respect to the characteristics mentioned above and about the REMAP project that also is an ANN designed using massively parallel processors with high speed inter-processor communication.
- The third part discusses the comparison between these and conclusions we can draw from the two architectures.





Background

An ANN is just a representation of biologically inspired algorithms its very far from the actual working of a brain. It's very important to understand the working of the human brain to simulate it and to understand the simplicity of our architectures and models as compared to the brain.

A lot of research and biological discovery was done on the brain to realize that the basic component is something called a neuron. The research suggests that there are huge number of such neurons in our brain. Brain imaging of an average human represents the activity of such neurons at various parts of the brain.

The basic parts of a neuron are dendrites (input), cell body, and an axon (output). The output, an axon is responsible to connect various neurons and the communication is done using synapses (like a chemical resistor).

The Artificial neuron is a very simple model consisting of levels/values that correspond to various impulse frequencies. Then a summation of such impulse frequencies is done to get the final information. Another aspect of the brain is its ability to learn new things. This happens by changing the structure of the brain to respond to such synapses or by changing the synapses. The former is a long-term adaptation e.g. language.

Analysis - SpiNNaker

SpiNNaker (Spiking Neural Network Architecture) is a massively parallel computer architecture created to reflect the working of a human brain. Objective of SpiNNaker was to turn the conventional super computers into advantageous energy-efficient massively parallel computers.

The spinnaker has multiple nodes in an array these nodes are run in parallel for processing. Each of these nodes contains 18 ARM968 processor cores

They used the NS-2 chip for simulating as it very closely resembles actual computer networks. In SNN, one neuron is normally associated with numerous others. Accordingly immense measures of one-to numerous correspondences must happen between handling hubs. In this way, multicast-empowered directing looks very efficient for the reenactment of neural networks

Analysis - REMAP

The REMAP architecture is more like a guideline of what an ANN should be like rather than an implementation. Although, there is an implementation of the REMAP architecture that was explored. This architecture used FGPAs to implement the parallel computer.

The processors are implemented in a Xilinx XC4005 circuits and the serial/parallel I/O device in Xilinx XC3020(8 parallel and 8 serial I/O each).

Meager Distributed Memory (SDM), created by Kanerva [Kanerva, 1988], is a two-layer feedforward arrange, however is all the more frequently – and all the more advantageously – depicted as a PC memory. It has an immense location space (ordinarily 10300 conceivable areas) which is just scantily (obviously) populated by real memory areas.

Conclusion and Futurework

- For all intents and purposes any ANN calculations that are dependent on training examples for parallelism can be done very well on every parallel computer. This sort of computation gets the performance level to a very high level.
- Communication and ring correspondence could be used in ANN calculations in a productive way. Communication or ring correspondence alone has turned out to be adequate in massively parallel machines. For massively parallel machines, without utilizing training model parallelism it is hard to utilize just communication channels.
- From the analysis of the above two implementations it is clear that identifying the best algorithm for mapping depends heavily on the kind of parallelism that required

A great platform to start for future work is to provide a neural system machine that can hold more than a million processors with very high bandwidth communication channels. All of this while keeping in mind the application of such to be a more extensive neural network. The applications for such massive devices include real-time data analytics and machine learning to provide great and useful insights on how learning can be approached.

REFERENCES

Nordström, T. and B. Svensson. "Using and designing massively parallel computers for artificial neural networks." (Research Report No. TULEA 1991:13), Luleå University of Technology, Sweden, 1991.

Nordstrom. T, B. Svensson, K. Nilsson, P.-A Wiberg “Towards modular,massively parallel neural computers”, Connectionism in a Broad perspective: Selected papers from the Swedish conference on connectionism – 1992, L. F. Niklasson and M.B. Boden Eds. Ellis Horwood, pp. 213-226, 1994.

Yu Ji, You-Hui Zhang, Wei-Min Zheng, “Modelling Spiking Neural Network from the Architecture Evaluation Perspective” Journal of Computer Science and Technology Jan-2016.

Steve B. Furber, David R. Lester, Luis A. Plana, Jim D. Garside, Eustace Painkras, Steve Temple, and Andrew D. Brown, “Overview of the SpiNNaker System Architecture” IEEE transactions on computers, vol. 62, no. 12, December 2013.

Xavier Lagorce, Evangelos Stromatias, Francesco Gallupi, Luis A. Plana, Shih-Chii Liu, Steve B. Furber and Ryad B. Benosman. “Breaking the millisecond barrier of SpiNNaker: implementing the asynchronous event-based plastic models with microsecond resolution”.

Evangelos Stromatias, Francesco Gallupi, Cameron Patterson and Steve Fuber, “Power analysis of large-scale, real time neural networks on SpiNNaker”.

Rumelhart, D. E. and J. L. McClelland. Parallel Distributed Processing; Explorations in the Microstructure of Cognition. Vol I and II, MIT Press. Cambridge.

Svensson, B. ”Implementation and application of a software configurable massively parallel computer.” Second Swedish Workshop on Computer Systems Architecture, Bålsta, Sweden.

Obermayer, K., H. Ritter and K. Schulten. "Large-scale simulations of self-organizing neural networks on parallel computers: application to biological modelling." Parallel Computing. Vol. 14(3): pp. 381-404.

Svensson, B. and T. Nordström, “Execution of neural network algorithms on an array of bit-serial processors,” in 10th International Conference on Pattern Recognition, Computer Architectures for Vision and Pattern Recognition, Atlantic City, NJ, USA, vol. II, pp. 501-505.

Linde, A., T. Nordström and M. Taveniku, “Using FPGAs to implement a reconfigurable highly parallel computer,” Field-Programmable Gate Array: Architectures and Tools for Rapid Prototyping; Selected papers from: Second International Workshop on Field-Programmable Logic and Applications (FPL'92), Vienna, Austria, H. Grünbacher and R. W. Hartenstein Eds. New York: Springer-Verlag, pp. 199-210.