SCCTS Transactions on Reconfigurable Computing, ISSN: 3049-1533 Vol. 2, No. 1, 2025 (pp. 33-41)
RESEARCH ARTICLE ECEJOURNALS.IN

Reconfigurable FPGA Algorithms for Advancing Big Data Processing

Hassan Jaber[1], Ali A. Mahrooqi[2], Khalid Mansoori[3]*

[1-3]Department of Computer Engineering, University of Bahrain, Sakhir P.O. Box 32038, Bahrain

Keywords:
Big Data Processing;
Data Acceleration;
FPGA Algorithms;
Parallel Computing;
Reconfigurable Systems;
System Optimization

Corresponding Author Email:
mansoorikhal@uob.edu.bh

DOI: 10.31838/RCC/02.01.05

Received : 15.09.24
Revised : 11.11.24
Accepted : 21.12.24

Abstract

Due to the exponential growth of data in our increasingly digital world, there are both huge opportunities and big challenges. Artificial intelligence is slowly seeping into our lives, but big data analytics are quickly rising up to meet this growing demand without the help of traditional computing architectures that are struggling with the volume, velocity and variety of the data. Big data is solved using Field Programmable Gate Arrays (FPGAs), a reconfigurable hardware that can be optimized for a given big data workload. The article takes a deeper look at how reconfigurable algorithms for FPGA based big data processing are changing analytics capabilities in industries across the board. Together, the convergence of big data and reconfigurable computing represents a paradigm shift in the type of analysis we perform on large scale data. Research and engineers are developing novel algorithms using the flexibility and parallelism of FPGAs to be able to adapt in real time to changing data patterns and processing requirements. By adopting an adaptive approach, new performance levels are released and organizations can gain actionable insight from massive datasets at unprecedented speed and accuracy. This dive into this emerging field will cover fundamental reconfigurable algorithm concepts, state of the art FPGA architectures for big data, and real applications in fields such as finance, healthcare, and scientific computing. Not only will we also be looking ahead to what future trends and directions that will continue to further revolutionise how we process and analyse the expanding digital universe.

How to cite this article: Jaber H, Mahrooqi AA, Mansoori K (2025). Reconfigurable FPGA Algorithms for Advancing Big Data Processing. SCCTS Transactions on Reconfigurable Computing, Vol. 2, No. 1, 2025, 33-41

Reconfigurable Computing for Big Data: Understanding

Reconfigurable computing is a shift in the way how we process data and analyze it. Unlike fixed architecture processors, reconfigurable systems can change their hardware structure according to the different algorithms and workloads requirements. In the realm of big data, it's particularly nice to know that you can have this kind of flexibility about the scale and nature of processing tasks.

Parallelism: Because of their ability to parse parallel streams on multiple data, FPGAs excel at parallel processing. It's important for dealing with the massive volumes of data we find in big data problems.

Adaptability: FPGAs have reconfigurable nature, which makes them perform better in different stage of a data processing pipeline or reconfigured dynamically to accommodate changing data characteristics.

Energy Efficiency: For large scale data centers, custom hardware implementations on FPGAs can be far more energy efficient than software running on general purpose processors (Figure 1).

Low Latency: FPGAs offer extremely low latency when algorithms are implemented directly in hardware, required for real time big data applications such as financial trading or network security.

To derive maximum benefit from these possibilities, new algorithmic approaches for the

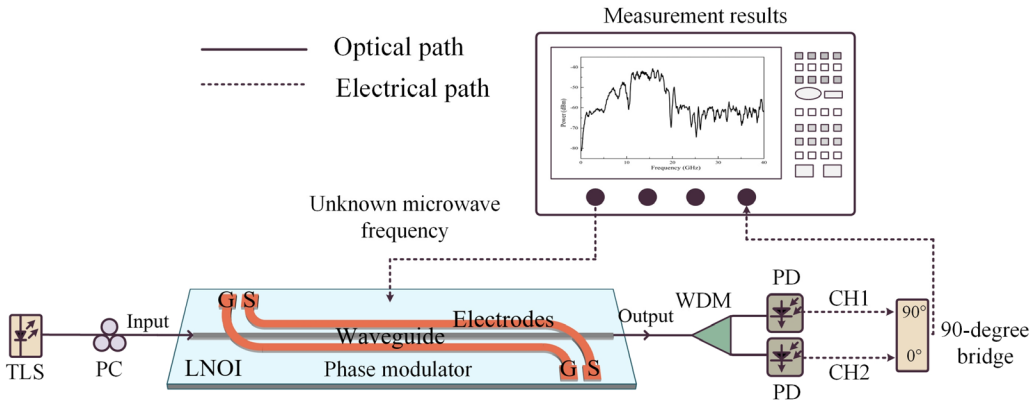

Fig. 1: Reconfigurable Computing for Big Data: Understanding

exploitation of the special features of reconfigurable hardware are being developed by researchers and engineers. Unlike simple porting of existing software to FPGAs, these reconfigurable algorithms go beyond that and actually reexamine what data processing tasks can be structured in order to take full advantage of parallelism and adaptability. A better understanding of the interplay for hardware architecture, algorithm design, and application requirement is important in exploring what we might call the landscape of reconfigurable algorithms for big data. A holistic approach that brings together all these factors into one cohesive whole to create synergistic systems that are at the edge of what's possible in big data analysis, these are the most effective solutions.[1-5]

FPGA Architectures for Big Data Workloads

Big data processing requirements are largely responsible for the evolution of FPGA architectures. The massive parallelism, high bandwidth requirements, and complexity of the computations inherent in big data workloads are exactly the things modern FPGAs were designed to tackle. The development of effective reconfigurable algorithms requires an understanding of these architectural innovations. The efficient movement of large volumes of data is one of the significant challenges in big data processing. In order to do this, FPGA manufacturers have implemented HBM interfaces into their devices. At these interfaces, algorithms gain orders of magnitude more bandwidth to access and process data at orders of magnitude greater rates than external memory. Generally, FPGAs with HBM have multiple memory channels that move data at rates in excess of 100 GBs per second. By providing this massive bandwidth, streaming data can be processed in real time, large datasets can run in memory analytics, and memory intensive algorithms including graph processing and deep learning can be elegantly implemented. Today's FPGAs aren't just arrays of lookups and flip-flops. They now incorporate a diverse set of computing resources optimized for different types of operations:

Table 1: FPGA Algorithm Characteristics for Big Data Processing

Characteristic	Relevance to Big Data Processing
Parallel Execution	Parallel execution enables FPGA algorithms to handle multiple data streams simultaneously, speeding up the overall data processing tasks.
Customizable Logic Blocks	Customizable logic blocks allow FPGA designs to be tailored for specific big data tasks, improving computational efficiency and flexibility.
Real-Time Processing	Real-time processing ensures that large datasets are analyzed and processed without delays, making FPGA an ideal solution for time-sensitive applications.
Scalability	Scalability allows FPGA systems to grow with the size of the data, handling increasingly complex computations as big data volumes expand.
Dataflow Optimization	Dataflow optimization ensures efficient data transfer and management, reducing bottlenecks in data handling and improving the performance of big data applications.
Low-Latency Operations	Low-latency operations enable faster response times for data processing tasks, critical for applications requiring immediate decision-making or real-time analytics.

- **DSP Blocks:** Digital signal processing units are dedicated u lerating machine learning workloads so your edge devices can run inference at high performance.

This heterogeneous architecture allows algorithm developers to map pieces of their algorithmsto those that provide the best computing resources, resulting in maximum system efficiency.

Network-on-Chip Interconnects

But as FPGAs grow larger and more complex, the need for efficient on chip communication is essential. High speed, low latency data movement between different parts of the device has become possible on Advanced FPGAs, where sophisticated network-on-chip (NoC) architectures are present. A variety of topologies are supported by these NoCs, and they can reconfigure in response to an application's communication pattern. Big data applications pose the challenge that the data flow patterns may change dynamically based on the input or the stage of the processing and therefore, NoC interconnects provide the flexibility for them. With this structure, reconfigurable algorithms can exploit it to reduce data movement bottlenecks and optimize data movement. Partial reconfiguration is one of the most powerful features of modern FPGAs – the ability to reconfigure a subset of the FPGA while leaving the rest operating. This allows for truly adaptive systems which can change in and out of different processing modules on-the-fly, based on changing data characteristics or application requirements. Such advanced architectural features can be leveraged, but only with careful algorithm design and system level thought. The best reconfigurable algorithms for big data exploit best the specific features of modern FPGAs to deliver tightly integrated hardware-software solutions at the very limit of performance and efficiency. To develop effective reconfigurable algorithms for big data processing we need to understand the design principles of algorithms as well as the unique hardware characteristics of FPGA hardware. In this section, we present a number of key concepts that serve as the basis for designing high performance, flexible algorithms for reconfigurable platforms.[6-7]

Dataflow-Oriented Design

Reconfigurable algorithms for FPGAs are different from traditional algorithms: they generally employ a form of a data flow paradigm as opposed to control flow. Computation of pipelines is organized as a network of processors through data streams in this model. This naturally corresponds to the parallel nature of the FPGA hardware, and hence the operations can naturally be pipelined. By adopting dataflow principles, FPGAs allow us to create algorithms that can exploit massively parallelism orders of magnitude faster than sequential implementations. By demonstrating FPBAs on the FPGA, they show that FPGAs enable a spatial computing paradigm, where computations can be mapped to physical hardware resources distributed across the chip. However, this is in contrast with the temporal computing model of traditional processors, where an execution unit operates sequentially in time over a single task manipulation space. Often, traditional approaches are being rethought in order to design algorithms with spatial computing in mind. Some operations (e.g., random memory access) which are trivial in software may require a restructure to efficiently operate in the spatial paradigm.

Dynamic Reconfiguration Strategies

This allows for new algorithms with adaptiveness at runtime using reconfigurable FPGA hardware. On the other hand, dynamic reconfiguration strategies allow algorithms to change their structure or the way in which they behave in response to changing data characteristics or changing processing requirements. Careful consideration is required for the tradeoffs between flexibility and overhead needed to effectively use dynamic reconfiguration. Reconfiguration is time consuming and will always demand power and the benefits must outweigh these costs. Co-design is successful only if the entire system is viewed holistically across the hardware software boundary including resource utilization, synchronization as well as data movement. With these skeletons developers can quickly prototype and optimize algorithms for certain big data tasks while remaining flexible to run customized versions for each application. If without these fundamental concepts we can not understand, how to apply these concepts to create reconfigurable algorithms which can fully take the advantages of FPGA platforms for the processing of big data. These principles will give us an idea of the basis of a wide array of powerful techniques and applications in many applications.[8-11]

Parallelization Techniques of Big Data Algorithms on FPGAs

Big data processing requires obtaining high performance with the help of massive parallelism

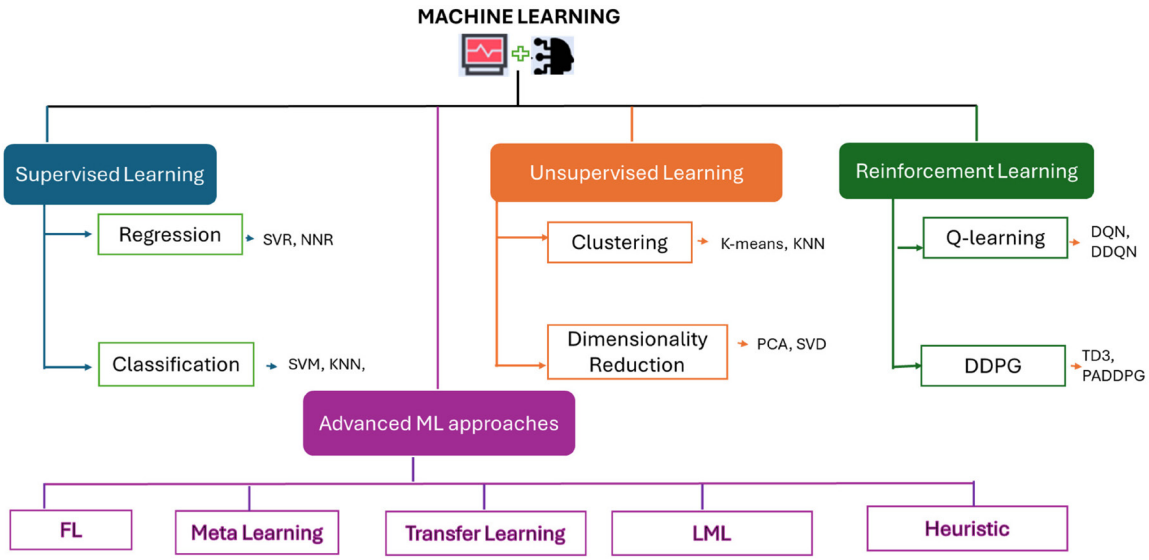

Fig. 2: Fundamental Principles of Fault Tolerant Design

from FPGAs. This section looks into advanced ways of parallelizing algorithms so that they can perform in the scale and complexity of the modern datasets.

Fine-Grained Parallelism

Parallelism at the fine grain level corresponds to breaking down computations into tiny, perpendicular units of computation perform in parallel. In general, this means creating sparse arrays of very simple processing elements doing a little bit of processing on a subset of data. Examples of coarse grained parallelism are to divide the entire algorithm into more complex, large scale tasks that can be run independently. This is a particularly useful approach for algorithms with in irregular computation patterns or over huge data structures. However, effective coarse grained parallelism often requires careful consideration of data partitioning and load balancing to meet the constraint of making every FPGA resource utilized efficiently. By matching this parallelization strategy to the memory architecture of the FPGA, developers can minimize data movement bottlenecks and maximize total system throughput (Table 2).

Adaptation of Dynamic Parallelism

FPGAs have reconfigurable nature and this allows for dynamic adaptation of parallelization strategies with respect to runtime conditions. This is particularly useful when the workloads have varying characteristics and are large data scale workloads. Dynamic adaptation becomes effective only if the hardware archicture

Table 2: Performance Metrics for FPGA-Based Big Data Processing Algorithms

Metric	Significance
Processing Speed	Processing speed is a critical metric, as FPGA-based algorithms can significantly reduce the time needed to process large volumes of data.
Power Consumption	Power consumption is an important metric for optimizing FPGA systems, especially when used for big data tasks in energy-sensitive environments.
Data Throughput	Data throughput measures the rate at which data is processed, with FPGA systems offering high throughput for large-scale data tasks.
Hardware Utilization	Hardware utilization indicates how efficiently FPGA resources are used for computations, impacting the overall effectiveness of big data processing.
Scalability Efficiency	Scalability efficiency measures how well the FPGA system adapts to growing data volumes without a proportional increase in resource consumption or delays.
Latency Reduction	Latency reduction is crucial in applications where real-time decision-making is required, and FPGA-based algorithms excel at minimizing delays in data processing.

and the control algorithms managing reconfiguration are carefully designed. A hierarchical approach to parallelization – of combining techniques at different granularities – proves useful for many complex big data algorithms. • A high level of coarse grained task parallelism to partition the algorithm into major stages. Effective coarse-grained parallelism often requires careful consideration of data partitioning and load balancing to ensure all FPGA resources are utilized efficiently.

Memory-Centric Parallelism

Given the memory-intensive nature of many big data algorithms, parallelization techniques that focus on optimizing memory access patterns can yield significant performance gains. By aligning the parallelization strategy with the memory architecture of the FPGA, developers can minimize data movement bottlenecks and maximize overall system throughput. The reconfigurable nature of FPGAs allows for dynamic adaptation of parallelization strategies based on runtime conditions. This can be particularly valuable for big data workloads with varying characteristics. Implementing effective dynamic adaptation requires careful design of both the hardware architecture and the control algorithms that manage reconfiguration. Many complex big data algorithms benefit from a hierarchical approach to parallelization, combining techniques at different granularities. The parallelization strategy can be matched to the natural structure of the algorithm and the characteristics of the data being processed with this multi level approach. The use of these advanced parallelization techniques allows for creation of reconfigurable algorithms that leverage massive computational resources of modern FPGA platforms. Success hinging on optimization of parallelization strategies given the structure of the algorithm, the properties of the incoming data and the capabilities of the specific target FPGA architecture being used.[12-14]

FPGA-Based Big Data Processing, Data Management and Memory Optimization.

To achieve high performance in FPGA based big data processing, data management and memory optimization are incredibly important. In particular, the massive data volumes of the big data applications, along with the unique memory architecture of FPGAs, make the consideration of data flow, storage strategies, and access patterns for the data critical. In this section, some advanced methods for data handling performance optimizations on FPGA platforms are presented.

Hierarchical Memory Architectures

Today, modern FPGAs provide us with various memory possibilities with different capacity, latency and bandwidth. To get the most out of these options, we need to design effective hierarchical memory architectures. To achieve good performance, effective algorithms must efficiently stream externally accessed data in from external memory, and at the same time very carefully orchestrate data between levels to keep frequently used data close to processing elements. Yet in big data workloads, irregular access patterns are common, and there may be no reason to force these to adhere to traditional cache architectures. Custom caching strategies for certain data structures and access patterns can drive FPGA based algorithms. The developer can achieve huge memory access latencies reduction as well as improvement of the system throughput by implementing these smart caching strategies.

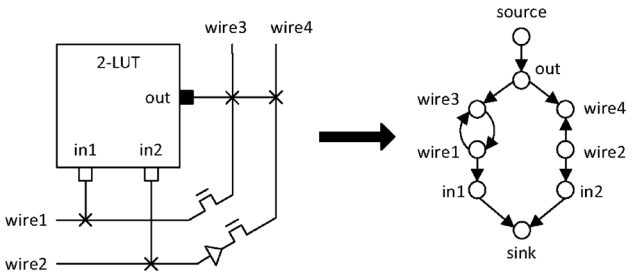

Fig. 2: Hierarchical Memory Architectures

Data Compression and Encoding

In environments with limited available on chip memory resources, data compression and efficient encoding can have profound impact on the achievable effective capacity and bandwidth of the memory subsystem. The issue is to find the balance between the increased performance of compression/decompression, and reduce the data movement needed, and increase effective memory capacity. For many big data applications, processing data in a stream rather than from static blocks can yield strong performance improvements. Finally, FPGA based streaming architecture helps to minimize the requirement of large intermediate storage, and to allow efficient pipelining of operations. Developers can develop algorithms to process huge datasets with minimal latency and

minimum usage of FPGA resources by taking streaming approach. Memory centric algorithm design views the data movement and access patterns as fundamental and not a passive storage resource. As a dynamic adaptive device, FPGAs facilitate the reconfigurable nature of memory management strategies with the changing runtime conditions and the changing workload characteristics. Dynamic memory management allows FPGA based big data processing systems to be more efficient and flexible. For developers looking to create reconfigurable algorithms that can efficiently deal with data sets of any sizes, leveraging these advanced data management and memory optimization alternatives, can enable developers to efficiently manage and process the massive numbers of data, which are characteristic to big data applications. One key is to take a holistic view of the entire system by concurrently considering data flow, access patterns, processing requirements together with the specific capabilities and constraints of FPGAs.[15-19]

FPGA Real-Time Analytics and Stream Processing

As a growing plethora of big data applications require the ability to process and analyze data in real time, big data storage systems must become increasingly better at dealing with streaming data. Real time analytics and stream processing tasks can be performed well with FPGAs due to it's low latency and high throughput capability. In this section, we investigate some advanced techniques to efficiently and adaptively implement real-time processing algorithms on FPGA platforms. Because there are many real time analytics applications that require ability to query over continuous streams of data, analyzers have the capability for complex querying within the case. Specialized hardware for the common query operations can speed these operations using FPGAs. Using these techniques in an FPGA based system can provide order of magnitude throughput improvement to standard software based techniques. Many streaming algorithms also work on sliding windows of data, where temporal data structures must be efficiently managed. For window management and processing, specialized hardware for FPGAs can be implemented. The context is maintained for complex analytics even with high velocity data stream and FPGA implementations can handle such high velocity data stream effectively using window-based processing. FPGAs combined with real-time data streams allow new approaches to online machine learning where models are updated online as new data arrives. Equipping FPGA based systems with these approaches enables performing sophisticate machine learning tasks on high velocity data streams with low latency.

1.1 Complex Event Processing

For many streaming analytics applications, the requirement is to identify and respond to complex patterns of events in real time. Highly parallel pattern matching engines can be implemented in FPGAs accelerating complex event processing. Thanks to these techniques, in real-time even in bursts of high speed data, FPGA based systems are able to detect and respond to sophisticated event patterns. Data rates and processing demands are highly variable for real time analytics workloads. Adaptive load balancing mechanisms are implemented using FPGAs to maintain optimal performance under changing conditions. The use of these adaptive mechanisms enables FPGA based real time analytics systems to operate at high performance with low latency despite highly variable workloads. Mission critical real time analytics applications must be reliable and fault tolerant. In streaming contexts, FPGAs are good at specializing hardware to detect and isolate faults and recover from them. Fault tolerant approaches presented here enable FPGA based systems to deliver the high reliability needed for reliable real time analytics. These advanced techniques in real time analysis and stream processing enable FPGA base system developers to develop image quad FPGA systems able to run the most demanding big data applications. The trick to achieving this is to include the high throughput and low latency characteristics of FPGAs, and sophisticated algorithms that make use of the dynamic nature of the real time data stream [20]-[23].

Accelerating Machine learning on FPGA Platforms

Machine learning and what to do with all that big data processing has significant opportunity and a lot

of challenges. FPGAs, with their reconfigurability and a massive amount of parallelism, present unique advantages for accelerating the machine learning workloads. This section discusses techniques for employing machine learning algorithms on FPGA platforms that bring added value to big data applications. In many big data analytics applications, deep neural networks (DNNs) have played a big role. By implementing DNN inference & training on specialized hardware(FPGAs), they can provide a great acceleration power. These adaptive approaches enable FPGA-based machine learning systems to maintain high accuracy even under evolving data distributions, characteristic of big data applications. Most machine learning algorithms, especially in large data setting, become equivalent to operations on sparse large matrices. FPGAs may be used to give special hardware implementations of efficient sparse matrix computations. A great many of the machine learning algorithms are forgiving of some amount of approximation, allowing for a trade off between accuracy and better performance or energy efficiency (Figure 3).[24-27]

ML Algorithms Hardware-Software Co-Design

Much of the effort to accelerate machine learning workloads relies on tight integration across hardware and software components. Because of this, FPGAs enable novel ways to approach the hardware-software design separation process of ML algorithms. Through a hardware so software co-design one will be able to go beyond to create ML acceleration systems that make optimized use of FPGA hardware and flexible software implementations. In a lot of big data applications, federated learning, where models are trained over distributed datasets decentralized, is becoming important. Federated learning algorithms can benefit

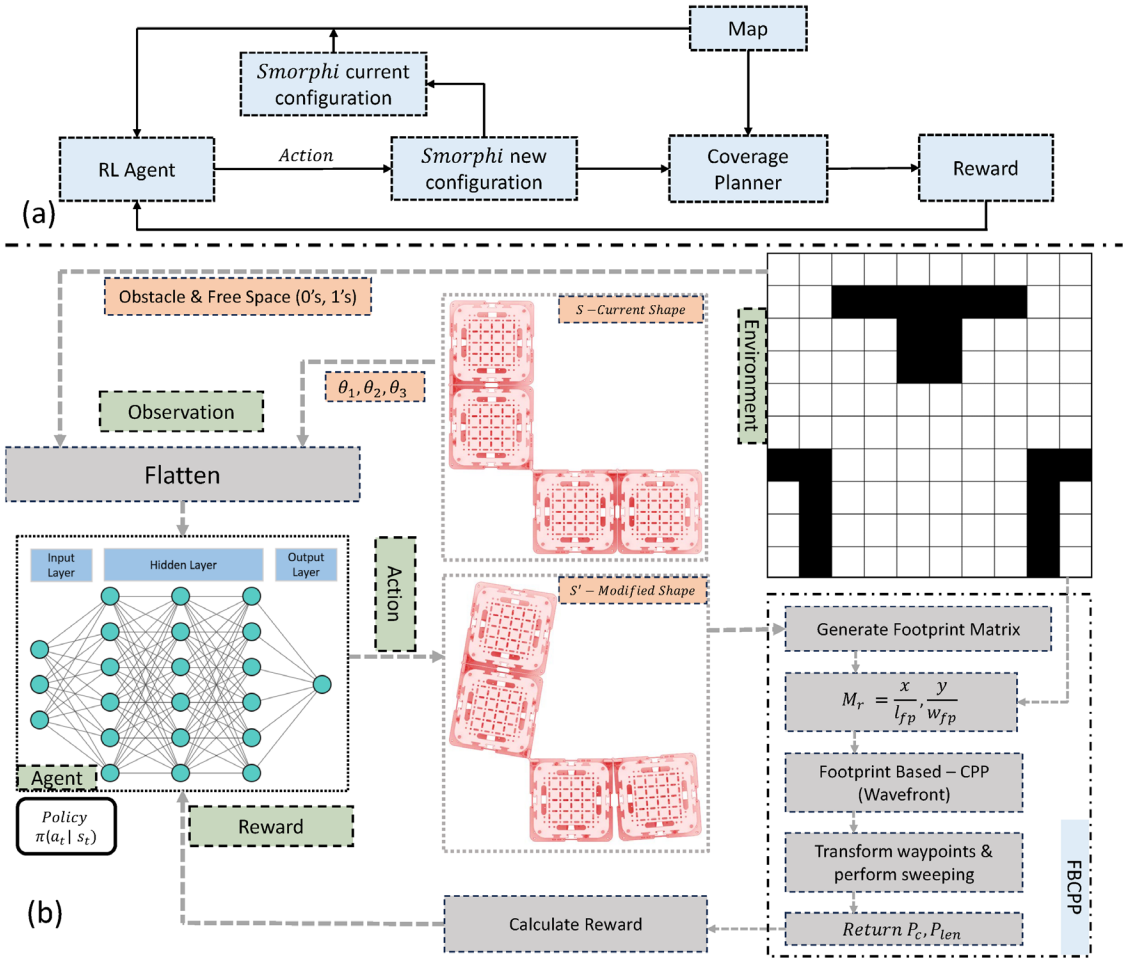

Fig. 3: Accelerating Machine learning on FPGA Platforms

from the acceleration that can be achieved in an FPGA. FPGA based systems are enabled to play a useful role within federated learning scenarios by taking advantage of the privacy and distribution strengths of Federated methods with the acceleration benefits of hardware. With these advanced techniques for machine learning acceleration, FPGA based systems can be created for the most demanding big data ML workloads. It is in combining the massive parallelism and reconfigurability of FPGAs with sophisticated algorithms that can change (reconfigure) to the specific characteristics of the different machine learning tasks and the different data distribution. An approach to optimization and benchmarking of reconfigurable algorithms on FPGA platforms is required to achieve optimal performance. In this section, we explore advanced techniques for achieving the highest possible efficiency of FPGA based big data processing systems, and the methodologies needed to accurately measure and compare both the performance and implementation overheads of each system. Through systematic exploration of the design space, developers are able to explore different configurations and understand what is a feasible choice for each workload, ultimately finding configurations maximizing performance for these workloads.

Conclusion

Performance models can result in accurate predictions of impact on system performance of design changes and can be used to guide optimization efforts reducing the need for time consuming implementation cycles. Finally, these modeling techniques allow for fast cycle iteration and optimization of FPGA based big data processing systems. By working systematically through the identification and resolution of bottlenecks, developers can realize a significant improvement in the overall performance of FPGA based big data processing systems. Many of the big data applications are energy efficiency that is very important. Fine grained power optimization on FPGAs presents unique opportunities. An application of these power optimization techniques can greatly improve the energy efficiency of FPGA based big data processing systems, especially for large scale deployments. A standardized benchmarking approach aids the community in better evaluating relative merits of alternative FPGA based solutions for big data processing. It is necessary to understand the scaling of performance with bigger data sizes or FPGA resources in order to find out strike good algorithms for big data applications.

References:

1. Bobda, C., Mbongue, J. M., Chow, P., Ewais, M., Tarafdar, N., Vega, J. C., ... & Tessier, R. (2022). The future of FPGA acceleration in datacenters and the cloud. *ACM Transactions on Reconfigurable Technology and Systems (TRETS)*, *15*(3), 1-42.
2. Miliadis, P., Theodoropoulos, D., Pnevmatikatos, D. N., & Koziris, N. (2022, September). VenOS: A virtualization framework for multiple tenant accommodation on reconfigurable platforms. In *International Symposium on Applied Reconfigurable Computing* (pp. 181-195). Cham: Springer Nature Switzerland.
3. Rodríguez, A., Valverde, J., Portilla, J., Otero, A., Riesgo, T., & De la Torre, E. (2018). Fpga-based high-performance embedded systems for adaptive edge computing in cyber-physical systems: The artico3 framework. *Sensors*, *18*(6), 1877.
4. Vieira, J., Malkowsky, S., Nieman, K., Miers, Z., Kundargi, N., Liu, L., ... & Tufvesson, F. (2014, December). A flexible 100-antenna testbed for massive MIMO. In *2014 IEEE Globecom Workshops (GC Wkshps)* (pp. 287-293). IEEE.
5. He, Z., Parravicini, D., Petrica, L., O'Brien, K., Alonso, G., & Blott, M. (2021, November). Accl: Fpga-accelerated collectives over 100 gbps tcp-ip. In *2021 IEEE/ACM International Workshop on Heterogeneous High-performance Reconfigurable Computing (H2RC)* (pp. 33-43). IEEE.
6. Bustio-Martínez, L., Cumplido, R., Letras, M., Hernández-León, R., Feregrino-Uribe, C., & Hernández-Palancar, J. (2021). FPGA/GPU-based acceleration for frequent itemsets mining: A comprehensive review. *ACM Computing Surveys (CSUR)*, *54*(9), 1-35.
7. Zhu, Q., & Agrawal, G. (2010, June). Resource provisioning with budget constraints for adaptive applications in cloud environments. In *Proceedings of the 19th ACM International Symposium on High Performance Distributed Computing* (pp. 304-307).
8. Islam, S., Keung, J., Lee, K., & Liu, A. (2012). Empirical prediction models for adaptive resource provisioning in the cloud. *Future Generation Computer Systems*, *28*(1), 155-162.
9. Vallabhuni, R. R., Yamini, G., Vinitha, T., & Reddy, S. S. (2020, September). Performance analysis: D-Latch modules designed using 18nm FinFET Technology. In *2020 International Conference on Smart Electronics and Communication (ICOSEC)* (pp. 1169-1174). IEEE.
10. Calheiros, R. N., Masoumi, E., Ranjan, R., & Buyya, R. (2014). Workload prediction using ARIMA model and its impact on cloud applications' QoS. *IEEE transactions on cloud computing*, *3*(4), 449-458.
11. Murphy, P., Shepard, C., Zhong, L., Dick, C., & Sabharwal, A. (2014). FPGAs Help Characterize Massive-MIMO Channels. *XceLL Journal*, *89*, 19-25.

12. Huang, P., Tonnemacher, M. J., Du, Y., Rajan, D., & Camp, J. (2020). Towards massive MIMO channel emulation: Channel accuracy versus implementation resources. *IEEE Transactions on Vehicular Technology*, *69*(5), 4635-4651.
13. He, Z., Korolija, D., & Alonso, G. (2021, August). Easynet: 100 gbps network for hls. In *2021 31st International Conference on Field-Programmable Logic and Applications (FPL)* (pp. 197-203). IEEE.
14. Hoozemans, J., Peltenburg, J., Nonnemacher, F., Hadnagy, A., Al-Ars, Z., & Hofstee, H. P. (2021). Fpga acceleration for big data analytics: Challenges and opportunities. *IEEE Circuits and Systems Magazine*, *21*(2), 30-47.
15. Sun, J., Zhang, Y., Wu, Z., Zhu, Y., Yin, X., Ding, Z., ... & Plaza, A. (2019). An efficient and scalable framework for processing remotely sensed big data in cloud computing environments. *IEEE Transactions on Geoscience and Remote Sensing*, *57*(7), 4294-4308.
16. Vallabhuni, R. R., Sravana, J., Pittala, C. S., Divya, M., Rani, B. M. S., & Vijay, V. (2021). Universal shift register designed at low supply voltages in 20 nm FinFET using multiplexer. In *Intelligent Sustainable Systems: Proceedings of ICISS 2021* (pp. 203-212). Singapore: Springer Singapore.
17. He, D., Nakandala, S., Banda, D., Sen, R., Saur, K., Park, K., ... & Interlandi, M. (2022). Query processing on tensor computation runtimes. *arXiv preprint arXiv:2203.01877*.
18. Ebrahim, A., & Khalifat, J. (2023). Fast approximation of the top-k items in data streams using FPGAs. *IET Computers & Digital Techniques*, *17*(2), 60-73.
19. Ebrahim, A., & Khlaifat, J. (2020, October). An efficient hardware architecture for finding frequent items in data streams. In *2020 IEEE 38th International Conference on Computer Design (ICCD)* (pp. 113-119). IEEE.
20. Mbongue, J. M., Shuping, A. M. I., Bhowmik, P., & Bobda, C. (2020, July). Architecture support for FPGA multi-tenancy in the cloud. In *2020 IEEE 31st International Conference on Application-specific Systems, Architectures and Processors (ASAP)* (pp. 125-132). IEEE.
21. Innova, M. (2). *Flex Open Programmable SmartNIC*.
22. Vallabhuni, R. R., Sravya, D. V. L., Shalini, M. S., & Maheshwararao, G. U. (2020, July). Design of Comparator using 18nm FinFET Technology for Analog to Digital Converters. In *2020 7th International Conference on Smart Structures and Systems (ICSSS)* (pp. 1-6). IEEE.
23. Microsoft Azure. Deploy ML models to FPGAs with Azure Machine Learning. https://azure.microsoft.com/enus/pricing/details/virtual-machines/windows/ [Last accessed: April 29, 2021].
24. Papaphilippou, P., & Luk, W. (2018, August). Accelerating database systems using FPGAs: A survey. In *2018 28th International Conference on Field Programmable Logic and Applications (FPL)* (pp. 125-1255). IEEE.
25. JFang, J., Mulder, Y. T., Hidders, J., Lee, J., & Hofstee, H. P. (2020). In-memory database acceleration on FPGAs: a survey. *The VLDB Journal*, *29*, 33-59.
26. Wang, N., Chen, F., Yu, B., & Qin, Y. (2020). Segmentation of large-scale remotely sensed images on a Spark platform: A strategy for handling massive image tiles with the MapReduce model. *ISPRS journal of photogrammetry and remote sensing*, *162*, 137-147.
27. Lee, C. A., Gasster, S. D., Plaza, A., Chang, C. I., & Huang, B. (2011). Recent developments in high performance computing for remote sensing: A review. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, *4*(3), 508-527.
28. Rahim, R. (2024). Quantum computing in communication engineering: Potential and practical implementation. *Progress in Electronics and Communication Engineering, 1*(1), 26–31. https://doi.org/10.31838/PECE/01.01.05
29. Rahim, R. (2024). Adaptive algorithms for power management in battery-powered embedded systems. *SCCTS Journal of Embedded Systems Design and Applications, 1*(1), 25-30. https://doi.org/10.31838/ESA/01.01.05
30. Sadulla, S. (2024). Optimization of data aggregation techniques in IoT-based wireless sensor networks. *Journal of Wireless Sensor Networks and IoT, 1*(1), 31-36. https://doi.org/10.31838/WSNIOT/01.01.05
31. Ariunaa, K., Tudevdagva, U., & Hussai, M. (2025). The need for chemical sustainability in advancing sustainable chemistry. *Innovative Reviews in Engineering and Science, 2*(2), 33-40. https://doi.org/10.31838/INES/02.02.05
32. Abdullah, D. (2024). Design and implementation of secure VLSI architectures for cryptographic applications. *Journal of Integrated VLSI, Embedded and Computing Technologies, 1*(1), 21-25. https://doi.org/10.31838/JIVCT/01.01.05
33. Prasath, C. A. (2023). The role of mobility models in MANET routing protocols efficiency. *National Journal of RF Engineering and Wireless Communication, 1*(1), 39-48. https://doi.org/10.31838/RFMW/01.01.05
34. El-Saadawi, E., Abohamama, A. S., & Alrahmawy, M. F. (2024). IoT-based optimal energy management in smart homes using harmony search optimization technique. *International Journal of Communication and Computer Technologies, 12*(1), 1-20. https://doi.org/10.31838/IJCCTS/12.01.01
35. Vijay, V., Pittala, C. S., Usha Rani, A., Shaik, S., Saranya, M. V., Vinod Kumar, B., Praveen Kumar, R. E. S., & Vallabhuni, R. R. (2022). Implementation of Fundamental Modules Using Quantum Dot Cellular Automata. Journal of VLSI Circuits and Systems, 4(1), 12–19. https://doi.org/10.31838/jvcs/04.01.03
36. Soh, H., & Keljovic, N. (2024). Development of highly reconfigurable antennas for control of operating frequency, polarization, and radiation characteristics for 5G and 6G systems. *National Journal of Antennas and Propagation, 6*(1), 31–39.