J. Res. Technol. Eng. 6 (3), 2025, 313-328

Journal Of Research Technology & Engineering
www.jrte.org

A Comprehensive Review of Low-Cost Big Data Clusters Using Single Board Computers: Architectures, Performance, and Applications

W. W. S. Welikala, H. K. I. S. Lakmal

Faculty of Engineering & Science, NSBM Green University
wwswelikala@students.nsbm.ac.lk, isuru.l@nsbm.ac.lk

Received:28 June 2025; Revised: 30 June 2025; Accepted: 06 July 2025; Available online: 10 July 2025

Abstract: The explosive increase in data generation has resulted in an unprecedented need for the availability of scalable and cost-effective big data processing solutions. Although traditional high-performance computing clusters have rich functionality, the energy and cost requirements are high, which inhibits the ability of education facilities, small research groups, and organizations with limited resources to adopt such a solution. This paper is a thorough review of the new paradigm of inexpensive big data clusters that deploy Single Board Computers (SBCs), and specifically, a Raspberry Pi-based implementation. Numerous research papers and hands-on experiences across hardware systems and platforms, software platforms and frameworks, benchmarking, and field implementations are critically analyzed here. The results demonstrate that SBC-based clusters, especially those deployed with Apache Hadoop ecosystems, provide potential solutions in terms of education, research, and small-scale industrial use, and further cost savings up to 85 percent of the traditional solutions while maintaining adequate performance for specific workloads. This review analyzes some of the main challenges, such as scalability shortcomings, network bottlenecks, and fault tolerance, as well as exploring the potential future trends in edge computing, integrating IoT, and green computing ventures. The analysis can be of critical help to researchers, educators, and practitioners who are seeking SBC-based solutions for distributed computing applications.

Index Terms: Big Data, Distributed Computing, Edge Computing, Green Computing, Hadoop, Raspberry Pi, Single Board Computers.

1 INTRODUCTION

Device novelty and cloud technology have experienced unimaginable growth in the amount of generated data, which is estimated to grow up to 175 zettabytes of global data creation by the year 2025 [1]. This phenomenal growth has made big data analytics a flagship technology in many industries and has influenced the invention of scalable, efficient, and cost-effective processing solutions. Historically, big data processing has been done with the help of costly clusters of high-performance computers located in specialized data centers, which have posed serious challenges to schools and universities, small research groups, and organizations with low budgets.

The arrival of the Single Board Computers (SBCs), and especially the Raspberry Pi ecosystem, has resulted in a drastic change in the paradigm of distributed computing solutions [2], [3]. The energy-efficient small-sized computing platforms have the prospect of democratizing access to big data technologies by offering cost effective methods of cluster building and experimentation. The combination of SBC technology and existing distributed computing systems, such as Apache Hadoop has provided innovative research and

learning opportunities as well as deployment of the products in environments with limited resources.

Current SBC technology development has greatly improved their level of computing performance, and recent systems even support multi-core processors, large memory capacities and high networking bandwidths [4]. At the same time, the future technologies of containerization, edge computing paradigm, energy-efficient computing are evolving, thus opening the possibility to utilize recently developed innovative cluster infrastructures, which exploit the specific features of the SBC platforms.

This is a thorough examination of a much-needed systematic investigation of low-budget big data deployments subsequently deployed with SBCs. Individual research projects have already demonstrated the practicality of such systems, but a comprehensive examination of their capabilities and limitations, as well as their optimal applications, is still scattered throughout the literature. Based on the findings of over 50 research publications, practical implementations, and case studies, we synthesize evidence to present a set of conclusive recommendations regarding the strategies for SBC-based cluster design, implementation, and cluster optimization.

This analysis extends beyond the cost implications to promote larger concerns regarding sustainable computing, accessibility to education, and democratization of technology. Clusters based on SBCs present attractive solutions to those seeking greener computing solutions as environmental concerns encourage more energy awareness in computer systems [5], [6], [7]. Also, the availability of these platforms has groundbreaking effects in terms of computer science teaching and research capability development in the developing world.

2 BACKGROUND AND FUNDAMENTALS

This section introduces distributed computing, big data processing, and enabling technologies, laying the groundwork for understanding low-cost big data clusters using SBCs. The readers will get a background of the development as well as the cluster or distributed computing paradigms, the nature of single-board computers, and the software tools that are instrumental in driving the current big data applications.

2.1 Evolution of Distributed Computing Paradigms

The history of the development in distributed computing has undergone various stages, solving issues of scalability and performance. Centralized traditional computing models granted simplicity in the management and security of the big data workload but failed miserably in the volume, velocity, and variety properties of the big data workloads [8]. Cluster computing, which appeared in the 1990s, provided the parallel processing feature of interlinking multiple commodity computers with high-speed networking, and it was economically scalable relative to more customized supercomputing hardware [9].

The MapReduce programming model, brought about by Google in 2004, changed the world of distributed data processing, as it offered a simplified way of parallel computation over datasets whose size exceeded the capacity of individual computers [10], [11]. This technological revolution has enabled organizations to analyze petabyte-scale data running on clusters of commodity hardware, which has transformed the world of big data. The open-source implementation of MapReduce in Apache Hadoop, together with the Hadoop Distributed File System (HDFS), became the backbone of current big data processing systems [12], [13].

2.2 Single Board Computer Technology

Single Board Computers can be viewed as the result of miniaturization tendencies coupled with system-on-chip (SoC) technology, where all the necessary computing elements are integrated on the same circuit board [14], [15]. Launched in 2012, Raspberry Pi spearheaded the industry of modern SBCs by offering a full computing system at a fantastic price of $35 [16]. Each successive model has added more capabilities, and the Raspberry Pi 4 has quad-core ARM Cortex-A72 processors, up to 8GB RAM, and Gigabit Ethernet connectivity [4].

The SBC ecosystem is not limited to Raspberry Pi and has broadened out to include solutions based on a wide variety of platforms (BeagleBone, ODROID, NVIDIA Jetson series), specialized by application domain [17]. The platforms differ materially in processing power and memory capacity, as well as the connectivity available and the power consumption characteristics, allowing different configurations of the cluster and a broader range of use cases. Industrial-grade SBCs exhibit great reliability, wider temperature ranges, and longer lifecycle support, thus making them appropriate for production deployments [18], [19].

2.3 Apache Hadoop Ecosystem

The Apache Hadoop ecosystem serves as an integrated system of distributed storage and processing of large amounts of data in the form of clusters of commodity hardware [12], [20]. Its main building blocks are the Hadoop Distributed File System (HDFS) providing fault-tolerant storage, MapReduce providing massive parallel processing of data, and Yet Another Resource Negotiator (YARN), which manages resources across the cluster [12], [21]. HDFS uses a master-slave architecture where MetaData related activities are served by NameNode services and real data storage services are served by DataNode services on cluster nodes [13], [22].

In Hadoop 2.0, YARN overcomes such drawbacks of the original JobTracker-TaskTracker model presenting the division of resource management functions and job scheduling functions [21], [23]. The architecture allows a variety of processing frameworks other than MapReduce to share the common cluster and allow diverse workloads such as real-time stream processing, interactive queries, and machine learning algorithms. ResourceManager is used to determine the sharing of resources across the entire cluster and ApplicationMasters are used to control individual application lifecycle and NodeManagers are used to execute containers on worker nodes [12], [24].

2.4 Containerization and Orchestration Technologies

Another technology that has become a vital part of the current SBC clusters implementation is the containerization technology, which places specific emphasis on Docker containerization technology [25], [26]. Containers offer a lightweight virtualization that creates packages containing an application and all of its dependencies to provide the same execution on a wide range of hardware. This is especially useful in SBC scenarios, where the heterogeneity of the hardware, as well as resource limitations, needs to be taken into consideration with regard to deployment strategies of application use.

Kubernetes has established itself as the de facto solution in the container orchestration space, or more specifically, as the automated solution to deploy, scale, and manage containerized applications [27], [28]. Whereas the implementation of traditional Kubernetes systems would require significant resources,

applications such as K3s are lightweight releases that have been adapted for edge computing and resource-limited systems, suggesting that such an application is ideal for SBC cluster deployment procedures [29]. The pairing of containerization and orchestration technologies can allow complex cluster management functionalities once only offered on expensive platforms.

3 HARDWARE PLATFORMS AND ARCHITECTURES

Section 3 explores the physical infrastructure and design considerations for building SBC-based clusters, comparing different hardware platforms, network topologies, and storage options. The discussion identifies the significant cost, scale, and performance trade-offs that guide hardware selection at the time of design and implementation of the low-cost clusters.

3.1 Single Board Computer Comparison and Selection

When deciding on possible SBC platforms to support big data cluster implementation, the criteria of computational demands, cost restrictions, and scalability targets ought to be taken into consideration. The Raspberry Pi ecosystem is the most widely used platform, as it itself provides an evolution since the initial single-core ARM11 processor to the later one with much more performance gains in every generation, the quad-core ARM Cortex-A72 [2], [4]. The new Raspberry Pi 4 can offer up to 8GB LPDDR4 memory and a Gigabit Ethernet port, which is already a significant improvement in the cluster computation power in comparison to the previous models.

Other SBC solutions have different strengths in application. The ODROID series includes more powerful ARM processors and increased memory options at a higher price, and increased amount of power consumption, but offers better calculation power [17]. NVIDIA Jetson systems are built to support GPU-accelerated workloads, which carry out custom tasks, such as those required in machine learning and computer vision. BeagleBone systems focus on real-time processing and wide I/O selections so that they are appropriate for industrial automation and IoT gateway systems [30].

The characteristics of performance differ greatly between SBC platforms and have implications on the design of clusters and workload requirements. Benchmark analysis made on the newer generation SBCs shows that they yield 2-4x improvements in CPU-intensive workloads; furthermore, memory bandwidth increases have immense advantages in data-intensive applications [4], [20]. Performance of the network has become of paramount importance and the addition of Gigabit Ethernet support facilitates more effective distribution of data and lower overhead requirements in cluster design.

3.2 Network Architecture Considerations

The choice of the design of the Network is a critical aspect of SBC clusters that can affect performance, scale, and fault-tolerance features. Conventional star topology implementations employing commodity Ethernet switches are an inexpensive approach for small to medium-sized clusters, but bandwidth may become bottlenecked in large cluster sizes [31], [32]. Gigabit Ethernet has become the new normal in the current SBC implementations; it offers an adequate level of bandwidth to most big data applications and also retains its affordability.

Advanced network architectures have hierarchical designs and have the ability to distribute the network load among several switches, reducing the chances of contention and improving the aggregate bandwidth

[9]. By applying the concept of Software-Defined Networking (SDN) to the SBC clusters, dynamic configuration of the network, along with optimization of the network, can be performed, but it necessitates greater complexity in cluster management systems. Management and data transfer operations can be separated by means of Network segmentation strategies, avoiding some common systems issues regarding reliability and security.

Power-over-Ethernet (PoE) enabled builds can be of great benefit in terms of vast deployments of large-scale SBCs since their power and network connections are integrated into a single cable connection [33]. This increases the chances of remote power cycling, enabling troubleshooting shooting, and hence, in fault recovery, it reduces the complexity of cable management and enhances the reliability of power management systems since power management is centralized. PoE implementations do necessitate specialized switches and have the potential to raise the overall cost of the system when compared to standard power distribution methods.

3.3 Storage Architecture and Data Distribution

The design of storage architecture plays a pivotal role in influencing the performance as well as reliability in SBC composed big data clusters. Although it is affordable and convenient, microSD cards are not a very good choice due to their limited performance and reliability portraits, which have the capability to limit cluster performance [4], [34]. Options such as solid-state storage (USB 3.0 drives, SATA SSD) offer high performance gain at the expense of cost per-node and complexity. Trade-offs in terms of cost, performance and reliability of storage are to be thoroughly considered on an application specific basis.

SBC clusters are known to have special requirements in HDFS implementation, owing to the fact that each node has small storage and may present reliability issues with user grade storage devices [22], [35]. This demands optimization to cater to special requirements in these systems. The configuration of replication factor is especially significant because the greater the replication factor, the more fault tolerant the system becomes at the cost of the consumption of more storage space and network bandwidth. Several studies show that replication factors of 2-3 are the best in terms of reliability in resource consumption in environmental SBC.

Clusters of SBC implementations have been demonstrated on strategies of distributed storage outside of HDFS. The application of object storage systems and distributed file systems of edge computing configurations may have increased the ability to scale and the fault-tolerant properties [36]. Network-attached storage (NAS) systems can be integrated to centralize high capacity storage with distributed processing ability but can also create more network bottlenecks and points of failure.

4 Software Frameworks and Technologies

This part discusses the main frameworks of software that assist in conducting distributed processing on SBC drawers, such as Hadoop adaptation and containerization approaches. It addresses optimization methods and management procedures that make the work of clusters efficient in a resource-limited environment.

4.1 Apache Hadoop Implementation Strategies

The deployment of Apache Hadoop on SBC clusters should be closely examined in terms of resource

limitations and approach optimization on ARM-based systems. Standard Hadoop builds that target x86_64 will not run optimally on ARM and may require ARM-specific builds or some form of cross compilation [4], [31]. Hadoop configuration options must be adjusted to suit the memory and processing constraints of SBC environments, especially the size of heap spaces assigned, the number of tasks allowed to run concurrently, and I/O buffer settings.

Memory handling is one of the areas that should be optimized in an SBC Hadoop implementation. The Java Virtual Machine (JVM) default settings of Hadoop services tend to imply significantly larger memory allocations than can be given on SBC systems [4], [31]. Optimization techniques involve decreasing the size of the JVM heap, tuning garbage collector settings, and incorporating memory-related swapping space management in cases of memory-straining conditions. Container-based deployments can enable other isolation and resource management capabilities, but with an overhead that has to be optimized versus the benefits.

Due to the configuration requirements of the YARN resource management framework that would be used on an SBC environment, reduced container memory allocations, altered scheduling policies, and updated heartbeat intervals are necessary circumstances [21], [23]. Research shows that the best-suited YARN configurations of SBC cluster will usually assign 512MB-1GB containers as opposed to 2 - 8GB containers assigned to a traditional cluster due to the memory limitations those platforms have [4], [31].

4.2 Container Orchestration and Management

The use of containerization as a deployment mechanism of SBC clusters has manifested as a desirable strategy, offering uniform application environments and cross-cutting administration advantages [25], [26]. Docker containerized Hadoop deployment makes a portable Hadoop cluster, which is supported across different SBC platforms with constant performance behavior. Arm-optimized container images will also be required because regular x86 containers will not run on ARM-based SBC platforms.

The Kubernetes orchestration offers advanced cluster management services that were complex to deploy on resource-limited platforms [27], [29]. Kubernetes distributions K3s and MicroK8s are particularly optimized to run in edge and IoT settings, with the goal of minimizing resource usage without losing key orchestration features. They have automated deployment, expansion, and restoration capacities achievable through these platforms, which improve the cluster's fault tolerance and work efficiency.

Kubernetes has numerous possibilities for connecting with Hadoop services, to which the issue of the distribution of resources and the service locating mechanism must be paid particular attention. Stateful applications like HDFS NameNodes need consistent identification through the network and persistent storage, and this can be difficult in an environment with dynamic containers. SBC clusters are capable of accommodating complex distributed systems with the simplicity of their deployment and management with the help of Helm charts and operators specifically geared towards big data workloads.

4.3 Monitoring and Management Tools

SBC clusters require extensive monitoring facilities since failures in the hardware can be frequent, and systems may experience variable performance on low-cost platforms [31], [37]. Prometheus and Grafana have become widely used monitoring systems, which in real-time collect and visualize metrics on

distributed systems. These tools are able to monitor performance in the system, such as CPU utilization, memory usage, network throughput, and storage performance on all cluster nodes.

Zabbix is also another monitoring solution that can offer the enterprise thorough infrastructure monitoring at the cost of fewer resources than the Prometheus-based solutions [31]. The agent-based architecture predetermines a detailed monitoring of single nodes without a withdrawal of the centralized management capabilities and alerting. SBC specific monitoring metrics like temperature sensors, power usage, and GPIO status can be monitored using custom-written monitoring scripts and used to give insights to the health of the hardware and the environmental conditions.

SBC-specific cluster management frameworks have the potential to automatically provide deployment, configuration management, and fault recovery. Infrastructure as Code solutions are possible through Ansible playbooks and Terraform configurations, so that a repeatable deployment into a cluster can be achieved and the upgrade developed with ease. These resources are especially useful when the size of the cluster gets big and it is no longer viable to be manually managed.

5 PERFORMANCE ANALYSIS AND BENCHMARKING

Section 5 discusses the means and techniques to determine and examine the abilities of SBC clusters, detailing the usage of typical assessment tools, performance trials and metrics to monitor. Readers will also get a detailed explanation on how these systems scale as well as the comparison of their effectiveness with traditional architecture.

5.1 Benchmarking Methodologies and Metrics

Benchmarking of big data clusters using SBC necessitates the formulation of appropriate benchmarking techniques that are standard and take into consideration the peculiarities and limitations of the platforms. Large-scale applications like TeraSort, TestDFSIO, and HiBench are standard benchmarks of big data with full performance assessment capabilities [4], [31], [38]. Nevertheless, the size and resources needed to meet such benchmarks need to be scaled to the capabilities of SBC clusters, where data are usually managed at the level of hundreds of megabytes up to low gigabytes instead of the terabyte level with enterprise systems.

The TeraSort benchmarking has been specifically useful when assessing SBC cluster performance because it challenges several system components, such as CPU, memory, network, and storage subsystems [4], [33]. Research shows that HDFS has linear scaling properties when it comes to execution time in relation to data volume, which proves that it is efficient even in environments where resources are restricted [31]. TeraGen, TeraSort, TeraValidate processes in three phases will give a whole performance observation, such as data generation rates, sorting throughput, and validation overhead.

TestDFSIO is designed to benchmark the performance of distributed file systems exclusively, and in particular, measure read and write performance characteristics with different numbers of nodes in a "cluster" [4], [31]. These benchmarks indicate that network bandwidth has proven to be the bottleneck of SBC clusters, especially in cases using shared Ethernet. The results show that there is an improvement in throughput as more nodes are added, but the improvement comes at diminishing returns as the network becomes saturated.

5.2 Performance Scaling Characteristics

Performance scaling studies point out intricate correlations between the size of clusters, the nature of workload, and the usage of the resources in SBC environments. Linear scaling can be achieved on CPU-intensive workloads, including computational algorithms and data transformations on up to 8-16 nodes with near-optimal ratios of speedup [4], [31]. Scaling is, however, expected to be sublinear in I/O intensive workloads because of the bandwidth limitations of the network used and possibly of storage devices used in SBC implementations.

Memory-demanding applications have serious problems in SBC, potentially because of the scarcity of RAM per node. Workloads involving significant amounts of in-memory working sets or caches have to be well-partitioned to prevent performance penalties due to swapping [4], [34]. The deployment of distributed memory systems like Apache Spark on SBC clusters needs intensive optimization of configuration to meet reasonable performance features.

Scaling involves the possibility that overall cluster performance is a bottleneck due to a saturated network infrastructure, as demonstrated by various studies [31], [33]. Gigabit Ethernet is adequate with small cluster sizes (4-8 nodes), and as cluster size increases, the scalability of a hierarchical network or higher bandwidth interconnect becomes necessary. There is a trade-off between cost-performance and cluster expansion, which requires consideration of the cost-performance trade-offs of network infrastructure upgrades.

5.3 Comparative Analysis with Traditional Systems

Research shows direct comparisons of the performance of the SBC cluster against a traditional big data infrastructure, and the trade-offs made between cost, energy efficiency, and computing ability are fascinating. On computational throughput, SBC clusters usually have a performance scaling of 10-30 percent compared to that of similar traditional clusters [4], [32]. The difference in cost is, however much higher, SBC implementation resulting in a reduction of up to 80-90 percent of costs incurred in equivalent traditional systems with the same number of nodes.

The results of energy efficiency comparisons of SBC implementations are favorable, with power consumption per node being 5-10 watts on average versus 100-300 watts in the case of typical server hardware [5], [32]. This is an impressive variance in power usage that ensures that SBC clusters are especially influential in instances when operational expenses and environmental effects are the major concerns. These operational savings should be reflected in the calculations of the total cost of ownership over the course of years of deployment.

The metrics of performance per dollar indicate the value offering of SBC clusters in a discrete application domain. Such fields as educational facilities, research prototyping, and development activities can be created to meet the learning and experimentation needs at a considerable cost with minimum performance adequacy [16], [37]. Traditional infrastructure still has an advantage when it comes to production applications that use high-performance computing resources, but SBC clusters can effectively be used to develop and test such applications.

6 APPLICATIONS AND USE CASES

This section reviews the areas of practical use where SBC-based big data clusters can play an impressive role, such as education, research, edge computing, IoT, and sustainability-oriented projects. It classifies various real-world applications, illustrating the strengths and limitations of SBC clusters across diverse use cases.

6.1 Educational and Research Applications

The educational sector has become a major implementer of SBC-based big data clusters due to the lower cost of such systems coupled with the pedagogical usefulness of the systems [16], [37]. Hands-on learning would be cost-prohibitive with traditional cluster infrastructure, but this can be achieved through SBC platforms. Students can acquire hands-on contact with ideas in distributed computing, cluster management, and big data technologies, without the need to have their institutions make major investments in computing infrastructure.

The use of SBC clusters in research has proved to be viable in algorithm development, proof of concept implementation, and small-scale data analysis projects [37], [39]. Its fast setup and easy access allow researchers to easily prototype distributed algorithms and test against their performance characteristics and scale to larger production systems more quickly than ever before. This has been a very useful manner, especially in academic settings where scientific funds are limited yet demands for innovativeness are high.

Standardized educational curricula that revolve around SBC cluster technologies have helped them proliferate into almost all computer science and engineering curricula [16]. Laboratory assignments using cluster setup, configuration, and programming give a practical education that supplements theoretical knowledge on distributed systems. The ability to move and the ease of installation of SBC clusters facilitate adaptable laboratory design capabilities that adjust with the change in class size and learning goals.

6.2 Edge Computing and IoT Integration

The potential unification of SBC cluster technology and edge computing paradigm has opened new prospects to perform distributed data processing at the edge of the network [40], [41], [42]. SBC clusters at edge sites are able to offer processing and analytics services locally to reduce network utilization and, consequently faster response times with IoT applications. This is to accommodate the emerging trend of moving even processing power closer to the sources of data in the form of distributed computing architectures.

The SBC clusters proved to be successful in industrial IoT apps where real-time measurements, predictive maintenance, and cycle optimization are present [14], [18], [30]. SBCs are compact in form factor and with low power consumption, hence their ability to be deployed in industrial applications with limited space and power. The ability to integrate sensor networks and industrial communication protocols provides the possibility of collecting and analyzing all the data without the use of centralized cloud resources.

The number of smart city applications is a growing field where SBC clusters can serve as intelligent nodes in traffic monitoring, environmental monitoring, and security applications [14]. The interconnected deployment of processing throughout urban infrastructure will provide the possibility to respond to changing conditions in real-time and also minimize the use of centralized data centers. Sensitivity to

privacy and security issues tends to support local data processing-based data analytic solutions that make little use of external systems.

6.3 Green Computing and Sustainability Initiatives

Environmental advantages of SBC-based clusters have been drawing the interest of organizations that have been seeking sustainability goals and carbon footprint minimization targets [5], [6], [43]. The sheer decrease in power usage over conventional infrastructure results in less greenhouse gas emissions and expenditure. Research indicates that SBC clusters may also perform the same kind of computing tasks at 80-95 percent reduced energy cost in comparison to server-based systems.

The use of SBC clusters in renewable energy is more viable because they have low power needs as well as the capacity to work in battery backup systems [6]. SBCs powered by solar energy have been successfully implemented in remote monitoring and data collection applications when there are no or unreliable grid power supply. This capacity to run whole clusters using renewable energy sources is a great leap forward as far as sustainable computing is concerned.

Since SBC-based computing is associated with having a low impact with regard to environmental duties of IT infrastructure, more corporations are seeing the potential value of emphasizing sustainability [7], [43]. By introducing SBC clusters into the organization to reduce carbon footprints through the development, testing, and non-critical production workloads, it will be possible to achieve significant reductions in carbon footprints of organizations while sustaining the provision of essential computing capabilities. The practice is in line with corporate social responsibility and governmental control of the reduction of environmental impact.

7 Challenges and Limitations

The major barriers during the deployment and management of low-cost SBC clusters in big data are critically discussed in Section 7. Areas covered involve reliability of the hardware, compatibility with software, restrictions on the network, and scalability issues, thereby putting a candid evaluation of what needs to be conquered, in order to achieve greater adoption, on record.

7.1 Performance and Scalability Constraints

Cluster scalability and application viability according to the SBC platform have built-in restrictions in their fundamental performance limits. Typical nodes in individual SBC setups are 1-4 CPU cores with minimal cache memory, meaning the computation resources are orders of magnitude less than the normal server hardware [4], [17]. Although cluster scaling can be used to balance out node constraints to an extent, overhead imposed by scaling up distributed coordination and communication may constrain productive ability to scale with selected performance profiles.

Memory bottlenecks are possibly one of the greatest constraints in the way SBC cluster is applied. Given normal RAM sizes of 1-8GB per node, most memory-intensive applications can meet much of their performance requirements but it comes at a huge cost [4], [34]. Java-based Hadoop necessitates the concerted management of memory, to prevent wasting excessive numbers of garbage collection overhead and subsequent out-of-memory errors. The effectiveness of caching mechanisms used in many applications of big data is also limited by the capacity of memory that is low in these devices.

The bandwidth limitations of networks can be witnessed once the size of the clusters goes large beyond a range of 8-16 nodes with any standard Gigabit Ethernet infrastructure [31], [33]. Distributed applications can saturate shared network resources, often leading to serious bottlenecks on the network, which restrict cluster performance. Some of these limitations can be overcome with hierarchical network designs but this increases complexity and cost.

7.2 Reliability and Fault Tolerance Issues

The practice of employing consumer-level hardware components in SBC clusters is a practice that comes with its own set of reliability problems, which must be cautiously handled during production [44], [45], [46]. MicroSD cards, which are often used as primary storage, experience greater failure rates than enterprise level storage devices and need to be replaced more often. The various potential malfunctions that power supply reliability, thermal accidents, and component lifespan possess have the ability to negatively affect the availability of a cluster.

There is a challenge that fault tolerance mechanisms used in conventional cluster environments cannot be effectively applied to SBC environments, where resources are limited and the architecture is quite different [44], [46]. Traditional fault tolerance methods may not be effective because they require a number of resources (in processing and memory) to be used in redundancy and error recovery. Container based deployments are able to offer some isolation and backup purposes, but come at the cost of increased complexity and resource overheads.

SBC clusters have problems associated with network partition scenarios because the links that are shared among the nodes introduce a possible single point of failure [45]. In contrast to the legacy clusters with redundant network routes and special-purpose clustering networks, SBC implementations often rely on commodity networking components that lack enterprise-class reliability assurances. It becomes significant to use the right monitoring and recovery mechanisms in order to sustain cluster availability.

7.3 Software Compatibility and Optimization Challenges

The majority of SBC platforms are based on ARM architecture, a platform that poses challenges in terms of software compatibility, which are absent in x86_64 [4], [34]. Most large-scale data software stacks are generally built and optimized on x86 platforms, so they must be cross-compiled, or built specifically on ARM, which are not necessarily easy to find. JVM tuning parameters developed to improve performance on the x86 architecture might not be easily adaptable to ARM architecture.

The availability of container images on ARM-based architectures is also much better now, but it is less widespread than the corresponding x86 versions [25], [28]. Deployment processes are also complicated by the requirement to have platform-specific builder images of containers and by the decisions on the distribution of certain software packages. Some of these issues can be resolved with multi-architecture container builds, though they necessitate extra development and upkeep.

It is a constant work to optimize the Java Virtual Machine to run on ARM hardware, and the vast majority of advice on tuning the JVM presupposes an understanding of the hardware model based on x86 [4]. The various memory hierarchy, cache behavior, and instruction set peculiarities of the ARM processors dictate that it needs certain methods of optimization that are not so comprehensively documented and understood.

Such a gap in knowledge can lead to poor performance, which would be hard to notice in the absence of detailed profiling and analysis.

8 Future Directions and Emerging Trends

This part describes the direction in which low-cost big data clustering is going, including hybrid frameworks, hardware progress, and green efforts. The discussion also gives an outlook on future research directions and technologies that will probably dominate the field.

8.1 Edge-Cloud Integration Architectures

The trend of hybrid edge-cloud computing has the potential to open up huge opportunities where SBC-based clusters can become an intelligent edge processing node [40], [41]. It can be expected that more work will be dedicated to the trouble-free interplay of edge-deployed SBC clusters, on the one hand, and cloud infrastructure, on the other hand, so that the distribution of workloads across the two environments can be performed regarding the latency thresholds, bandwidth limitations, and data security concerns. This combination of specialties can be used to get the best usage of available resources and still ensure the responsiveness necessary in real-time applications.

Applications of federated learning on the SBC clusters are a new area of application that harnesses the advantages of using distributed process power and overcoming issues of data privacy and bandwidth distribution [47]. Such systems may support distributed training of machine learning models on distributed datasets without centralization of data and thus find specific application to privacy-sensitive tasks in healthcare, financial, and personal data processing.

The trend of shifting towards edge-native software frameworks purposefully developed to operate under resource-constrained scenarios is one likely to increase the use of SBC clusters in production systems [41], [42]. Such frameworks can be optimized to take advantage of the nature of edge deployments, with intermittent connectivity, limited resources, and various hardware configurations of equal frequency in SBC cluster deployments.

8.2 Advanced Hardware Technologies

The next-generation properties of SBC-based systems with complex processing features, including AI acceleration, GPU, and self-managed co-processors will broaden the application scope of the cluster deployment [14]. Already, platforms such as NVIDIA Jetson are showing that GPU-accelerated computing is applicable to SBC form factors and bringing possibilities of computer vision, machine learning, and scientific computing to SBC platforms that were otherwise unattainable.

Functional neuromorphic computing power on future SBC platforms may make ultra-low-power AI inference and learning applications possible [7]. Such dedicated processors have the potential to deliver orders of magnitude improvements in the energy efficiency of specific classes of computational tasks, and these could be especially attractive in battery powered as well as renewable energy applications.

Despite remaining much in the realm of theory, quantum computing elements added to SBC platforms could unlock revolutionary potential in special areas of computation [7]. An ability to combine the quantum and classical processing capabilities into affordable, easy to access, and deliver platforms would

democratize access to quantum computers and research and development.

8.3 Sustainable Computing Initiatives

The increasing importance of sustainable computing practice is also likely to encourage more organizations to implement SBC-based solutions in order to make their operations less environmentally damaging [5], [7], [43]. More specialized SBC platforms oriented towards optimal energy efficiency, incorporation of renewable energy sources, and reduction of the carbon footprint of the built environment may come in the future. Such platforms would include sophisticated power management, energy harvesting, and optimized-cooling.

By SBC cluster design involving circular economy principles, the lifecycles of devices may be prolonged, recycling processes may be enhanced, and electronic waste minimized [6]. Modular designs potentially allow upgrade and repair of components and would result in a better sustainability profile of SBC clusters, in addition to lowering the total cost of ownership over the long-term deployment.

Carbon-smart computing systems that perform work optimally and execute it in a carbon-efficient manner may be developed, which would further enable SBC clusters to be even more environmentally friendly [5]. These systems may also be able to automatically transfer computational loads to renewable energy systems or delay non-critical processing to the time of low carbon intensity grid power.

9 Conclusion

This review paper has discussed in detail how low-cost big data cluster systems based on Single Board Computer technology can be built and used, based on the review that has been conducted using the findings of more than 50 research articles and practical implementations. This analysis indicates that SBC-based clusters, especially involving Apache Hadoop ecosystems are a potentially viable and increasingly mature way of tackling distributed computing with specific domains of application.

The main findings indicate that the current SBC units, in particular Raspberry Pi 4 and similar systems, have enough computational power to be used in educational, research, and small-scale production tasks. Although the performance shortcomings in comparison to more traditional cluster hardware cannot be overstated, the cost savings of 80-90% in capital cost, in combination with drastic energy cost savings, produce strong value propositions to applications that make good use of them.

Performance evaluation instances indicate that SBC clusters can scale linearly with CPU-intensive jobs to 8-16 nodes, with network bandwidth appearing as the scalability bottleneck in larger systems. Benchmarking studies indicate that these systems can be efficient with datasets in the hundreds of megabytes to low gigabytes range, and therefore, they are suitable for a variety of educational and research applications.

The areas of application with the most promising potential are the distributed systems learning platforms in education, edge computing, and integration points with Internet of Things, and green optimization projects with sustainability missions. Applications like the ones described can fit very easily within the capabilities of SBC platforms without necessarily having the drawbacks of SBCs when handling higher-performance applications in the realm of high-performance computing.

Nevertheless, some important hurdles still exist, such as reliability issues of consumer-grade parts, software challenges in working with the ARM processor, and limits in scaling out networking infrastructure. Future research needs to work on the elimination of these limitations by making improvements on the hardware components, software, and architectural designs, which could be used to better leverage the capabilities of SBC platforms.

The development of edge-cloud integration solutions, the improvement of hardware technologies, such as AI acceleration capabilities, and the eco-friendly computing programs indicate that SBC-based clusters will not only maintain their current state but also extend their degree of applicability. The potential impact of democratization of distributed computing power by the availability of low-cost platforms to education, research, and adoption of technology in resource-constrained settings is enormous.

This review gives critical information to researchers, educators, and practitioners who plan to use SBC-based solutions for distributed computing applications. Proper analysis of existing possibilities, constraints, and directions can provide the basis of proper decision-making on how these technologies are supposed to be utilized. With the SBC ecosystem still developing and new software applications being created, this field of research and development will be crucial for optimizing the use of the inexpensive distributed computing solutions.

REFERENCES

[1] J. Gantz and D. Reinsel, Extracting Value from Chaos, IDC IView, vol. 1142, pp. 1–12, 2011.

[2] M. Mishra, "Big Data Cluster Environment Powered by Raspberry Pi 4 and Hadoop," LinkedIn, [Online]. Available: https://www.linkedin.com/pulse/big-data-cluster-environment-powered-raspberry-pi-4-hadoop-mishra-

[3] A. Manam, “raspberry-pi4-hadoop-spark-cluster,” GitHub, [Online]. Available: https://github.com/aimanamri/raspberry-pi4-hadoop-spark-cluster

[4] H. Yazdani, H. Heidari and A. Ejlali, “Energy-efficient scheduling in cloud computing using a reinforcement learning model,” Computers & Electrical Engineering, vol. 102, 2022, Art. no. 108278. [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/S0045790622006206. doi: 10.1016/j.compeleceng.2022.108278

[5] K. A. Hussain and P. A. S. Reddy, "Green Computing in Hadoop Clusters: Approaches for Energy Efficiency," NeuroQuantology, vol. 20, no. 10, pp. 1057–1065, 2022. [Online]. Available: https://www.neuroquantology.com/article.php?id=14872

[6] L. Žlajpah, “Green and energy efficient computing in cloud environment,” in Proc. MakeLearn & TIIM Joint Conf., Bari, Italy, 2015, pp. 377–384. [Online]. Available: https://toknowpress.net/ISBN/978-961-6914-13-0/papers/ML15-377.pdf

[7] Gartner, “Green computing,” Gartner, [Online]. Available: https://www.gartner.com/en/articles/green-computing

[8] V. S. A. Anusha and M. Lakshmi, “Efficient Resource Utilization in Hadoop,” International Journal of Distributed and Parallel Systems (IJDPS), vol. 3, no. 1, pp. 131–140, 2012. [Online]. Available: https://airccse.org/journal/ijdps/papers/0112ijdps13.pdf

[9] D. P. Acharjya and A. Kauser, “A survey on big data analytics: Challenges, open research issues and tools,” CiteSeerX, [Online]. Available: https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=a2b09be4e98eb1fef81b81ec2245405daa64b554

[10] C. Kozyrakis et al., “CMP-based Systems: Performance and Power Analysis,” IEEE 13th International Symposium on High Performance Computer Architecture, 2007. [Online]. Available:

http://csl.stanford.edu/~christos/publications/2007.cmp_mapreduce.hpca.pdf

[11] B. Moon, "Challenges in Big Data Systems," SIGMOD Record, vol. 40, no. 4, pp. 4–5, Dec. 2011. [Online]. Available: https://www.cs.arizona.edu/~bkmoon/papers/sigmodrec11.pdf

[12] Apache, “YARN: Yet Another Resource Negotiator,” Apache Hadoop, [Online]. Available: https://hadoop.apache.org/docs/current/hadoop-yarn/hadoop-yarn-site/YARN.html

[13] Apache, “HDFS Architecture Guide,” Apache Hadoop, [Online]. Available: https://hadoop.apache.org/docs/r1.2.1/hdfs_design.html

[14] LattePanda, “Building a Raspberry Pi Cluster,” LattePanda Blog, [Online]. Available: https://www.lattepanda.com/blog-308796.html

[15] Maxtang, “Mini PCs for Industrial Applications,” MaxtangPC, [Online]. Available: https://www.maxtangpc.com/industrynewsen/42.html

[16] LattePanda, “AI Cluster Computing with Single Board PCs,” LattePanda Blog, [Online]. Available: https://www.lattepanda.com/blog-311021.html

[17] A. Naveed et al., "Multi-Objective Optimization for Energy Efficiency in Data Centers," Future Generation Computer Systems, vol. 98, pp. 135–151, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167739X1833142X. doi: 10.1016/j.future.2019.03.019

[18] Gateworks, “Applications for Embedded Systems,” Gateworks, [Online]. Available: https://www.gateworks.com/applications/

[19] CompuLab, “SBC-IoT,” CompuLab, [Online]. Available: https://www.compulab.com/products/sbcs/sbc-iot-link-nxp-i-mx93-internet-of-things-single-board-computer/

[20] K. B. Bhosale, “Big Data Analytics Using Hadoop,” International Journal of Computer Applications, vol. 178, no. 42, pp. 9–14, Jun. 2019. [Online]. Available: https://ijcaonline.org/archives/volume178/number42/30820-2019919328/. doi: 10.5120/ijca2019919328

[21] GeeksforGeeks, “Hadoop YARN Architecture,” GeeksforGeeks, [Online]. Available: https://www.geeksforgeeks.org/big-data/hadoop-yarn-architecture/

[22] IBM, “HDFS (Hadoop Distributed File System),” IBM Think, [Online]. Available: https://www.ibm.com/think/topics/hdfs

[23] CelerData, “YARN (Yet Another Resource Negotiator),” CelerData Glossary, [Online]. Available: https://celerdata.com/glossary/yarn-yet-another-resource-negotiator

[24] DataFlair, “Hadoop YARN Resource Manager,” Data-Flair, [Online]. Available: https://data-flair.training/blogs/hadoop-yarn-resource-manager/

[25] S. Khan, M. Ali and A. Khan, “Performance and Fault Tolerance Analysis of Hadoop Clusters: A Comparative Study,” Computers, Materials & Continua, vol. 73, no. 3, pp. 5551–5568, 2022. [Online]. Available: https://www.techscience.com/cmc/v73n3/49039. doi: 10.32604/cmc.2022.024903

[26] Curity, “Clustering Using Docker Compose,” Curity.io, [Online]. Available: https://curity.io/resources/learn/clustering-using-docker-compose/

[27] Docker, “How to Set Up a Kubernetes Cluster on Docker Desktop,” Docker Blog, [Online]. Available: https://www.docker.com/blog/how-to-set-up-a-kubernetes-cluster-on-docker-desktop/

[28] Kubernetes, “Container Runtimes,” Kubernetes Docs, [Online]. Available: https://kubernetes.io/docs/setup/production-environment/container-runtimes/

[29] Reddit, “K8s Cluster Using Single Board Computers,” Reddit, [Online]. Available: https://www.reddit.com/r/kubernetes/comments/t3bvoe/k8s_cluster_using_single_board_computers/

[30] Application Nexus, “IoT Single Board Computer Development,” Application Nexus, [Online]. Available: https://www.applicationnexus.com/services/iot-development/iot-single-board-computer-development/

[31] Elsevier RefHub, “Computers & Electrical Engineering,” RefHub, [Online]. Available: http://refhub.elsevier.com/S0045-7906

[32] D. Kutscher, “Affordable HPC: Using Commodity Hardware for High Performance Computing,” [Online]. Available: https://dirk-kutscher.info/publications/affordable-hpc/

[33] S. S. Marshall, “Building a Raspberry Pi Hadoop + Spark Cluster,” DEV Community, [Online]. Available: https://dev.to/awwsmm/building-a-raspberry-pi-hadoop-spark-cluster-8b2

[34] A. Goyal et al., “Edge Analytics Using Raspberry Pi Cluster for Real-Time Applications,” ACM Trans. Internet Things, vol. 5, no. 3, pp. 1–23, Jun. 2024. [Online]. Available: https://dl.acm.org/doi/fullHtml/10.1145/3626203.3670618. doi: 10.1145/3626203.3670618

[35] A. P. M. Kovačević, “Implementation of Big Data Architecture Using Raspberry Pi Cluster,” TEM Journal, vol. 10, no. 2, pp. 806–814, May 2021. [Online]. Available: https://www.temjournal.com/content/102/TEMJournalMay2021_806_814.pdf. doi: 10.18421/TEM102-48

[36] R. H. Bhatt et al., “IoT-Edge Cluster with Lightweight ML for Smart Agriculture,” arXiv preprint, Dec. 2023. [Online]. Available: https://arxiv.org/html/2312.17524v1

[37] M. Giger, K. Srikugan and A. Persaud, “Distributed Data Analytics with Raspberry Pi Cluster,” University of Zurich MSc Project Report, [Online]. Available: https://www.ifi.uzh.ch/dam/jcr:9c6065e2-10aa-442b-915c-57246020c23c/ReportMScProjektGigerSrikuganPersaud.pdf

[38] N. Poggi, “Big Data Benchmark Compendium: SPEC, RGBD, and TPC,” UPC, [Online]. Available: http://www.npoggi.site.ac.upc.edu/publications/SPEC-RGBD-TPCTC-2015-Big%20Data%20Benchmark%20Compendium.pdf

[39] J. T. Chang and Y. Chen, “Real-Time Processing of Data Streams in IoT Using Edge Computing,” Journal of Systems Architecture, vol. 147, 2024, Art. no. 102782. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0743731524001606. doi: 10.1016/j.sysarc.2024.102782

[40] Rackspace, “Edge Computing Primer,” Rackspace Blog, [Online]. Available: https://www.rackspace.com/blog/edge-computing-primer

[41] SUSE, “Distributed Edge Computing: Unlocking Innovation,” SUSE.com, [Online]. Available: https://www.suse.com/c/distributed-edge-computing-unlocking-the-power-of-decentralized-networks-to-drive-innovation/

[42] GeeksforGeeks, “What is Edge Computing in Distributed Systems?” GeeksforGeeks, [Online]. Available: https://www.geeksforgeeks.org/cloud-computing/what-is-edge-computing-in-distributed-system/

[43] R. Ombuya and J. Waweru, “Performance Optimization of Hadoop Clusters Using Edge Computing,” African Journal of Computing & Engineering, vol. 4, no. 2, 2022. [Online]. Available: https://ajpojournals.org/journals/index.php/AJCE/article/view/1905

[44] T. Roscoe, “Fault-Tolerant Cluster Computing,” UCLA Technical Report, [Online]. Available: https://web.cs.ucla.edu/~tamir/papers/ftct_tr.pdf

[45] S. J. Park and B. Lee, “Lightweight Cluster Setup for Educational Use,” CEUR Workshop Proceedings, vol. 3374, 2023. [Online]. Available: https://ceur-ws.org/Vol-3374/paper02.pdf

[46] M. Zaharia et al., “Resilient Distributed Datasets: A Fault-Tolerant Abstraction for In-Memory Cluster Computing,” CiteSeerX, [Online]. Available: https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=e6ce908b2b5e5a2871f9e80b5ddc35002f953e9c

[47] N. Akhtar et al., “AI-Powered Edge Systems for Smart City Applications,” arXiv preprint, Dec. 2024. [Online]. Available: https://arxiv.org/abs/2412.04687