

Critical Analysis of Parallel and Distributed Computing and Future Research Direction of Cloud Computing

Rimma Padovano

Department of Computer Science, Columbia University, USA.
rimmapadovano32@gmail.com

Article Info

Journal of Computing and Natural Science (<http://anapub.co.ke/journals/jcns/jcns.html>)

Doi: <https://doi.org/10.53759/181X/JCNS202101017>

Received 18 March 2021; Revised form 02 May 2021; Accepted 28 June 2021.

Available online 05 October 2021.

©2021 Published by AnaPub Publications.

Abstract - "Cloud computing" refers to large-scale parallel and distributed systems, which are essentially collections of autonomous. As a result, the "cloud organization" is made up on a wide range of ideas and experiences collected since the first digital computer was used to solve algorithmically complicated problems. Due to the complexity of established parallel and distributed computing ontologies, it is necessary for developers to have a high level of expertise to get the most out of the consolidated computer resources. The directions for future research for parallel and distributed computing are critically presented in this research: technology and application and cross-cutting concerns.

Keywords –Parallel and Distributed Systems, Artificial Intelligence, Machine Learning, Cloud Computing.

I. INTRODUCTION

The accurate and frequent upsurge in computing capabilities generated by conventional technology scalability patterns and optimized computer networks that leverage and disclose computational resources is the drumbeat of computer science progress. As a result of parallel and distributed computing, numerous computational capabilities, such as CPU cores and machineries, can be aggregated into a single powerful and effective system. Because they have evolved beyond the conventional high-performance computing applications, like scientific computation, parallel and distributed systems presently constitute a significant computer resource that is constantly improving and evolving over time Smart cities, large-scale global system for mobile communications [1] such as social networks, artificial intelligence (AI), and machine learning (ML) are just a few of the technologies that are available.

In order to meet the constantly rising supercomputing demand for new and larger distributed system applications, the sector of Parallel and Distributed Computing should overcome a number of traditional and future problems. The end of Dennard scaling and Moore's law would then presume that future increases in computing capability would not be procured simply by boosting the performance of independent cores and machineries in a system, and this is an immediate concern. On the other hand, future high-performance computing schemes will have to be specialized—customized in hardware & software algorithms and abstractions—to fit the accurate organization and prerequisites of a particular application. The suggested measures include a transition to sustainable supercomputing fabrics, which may provide future performance improvements.

A homogenous platform is made up of many different of peripherals, which can be very compelling but can complicate software development. However, as computer technology expands into new surroundings and contexts, such as cloud-based agribusiness (which involves regionalized sensing arithmetic operations on thousands or millions of low-power embedded systems as well as model - based computer models on an environment or global scale in big data), the need for software design becomes more pressing. As a result of such application areas, intelligent systems will have to compose and synchronize small or medium devices. A wide range of new abstractions and computational methods such as signalling, Machine learning, and pattern recognition will be enabled by these applications.

Parallel and distributed computing is now at an intersection because of these issues: new ontologies are required to manage the intricacies of developing a system at the spectrum and heterogeneity required by modern computing prospects. There are many disciplines (subfields of computer science) that contribute to every tier: It integrates both the application itself (conventionally called High Performance Computing (HPC), as well as its algorithms (Theory of Algorithms), the programming used to implement it as well as its platform (Programming Languages), along with any supporting hardware (Computer Architecture). The need for these inferences is also growing as conventional processing goals of scalability and flexibility (such as latency or bandwidth), energy consumption, and efficiency are pursued, as well as correctness, privacy and security which are now central system goals.

It is possible that the scientific method for constructing these frameworks will demand innovative automated evaluations and methodologies that address such objectives from the very beginning. In conclusion, this means that future

frameworks will be domain-specific, combining a diversified mix of supercomputing patterns, computational methods, and components in order to accomplish goals that go beyond conventional systems' objectives, including the need for more dependable, substantiated, and stable computer systems that are configurable and energy-efficient as well. The upcoming section (Section II) focuses on evaluating the recent works regarding parallel and distributed computing. Section III provides a critical analysis of the theme: technology and application; and cross-cutting concerns regarding parallel and distributed computing. Section IV concludes the paper.

II. LITERATURE REVIEW

P. Saariluoma and A. Oulasvirta in [2] argue that discipline boundaries should be re-evaluated. When it comes to computer science, the society's goals must be met through synchronised progress in numerous areas of the field, particularly across hardware-oriented and applications. As a result, collaborative relationship with system administrators outside of computer science will be necessary to develop domain-specific remedies for challenging emerging software.

A. Kahng in [3] posit that to the climax of Moore's Law and Dennard scaling comes the need for more investigation into other initiatives to maximize success - including the use of dedicated hardware accelerators and contemplating supercomputing systems that prioritize precision or serviceability in favor of performance. In order to design algorithms, languages, and systems that are domain-specific and platform-specific, innovative techniques are necessary that straightforwardly incorporate highly specialised abstractions.

C. FU and S. XU in [4] argue that determining how to build appropriate large-scale parallel and distributed infrastructures is already a tough prospect, and it will become much more complicated as the structures grow in size and complexity at the hardware and software dimension. All facets of planning and evaluation must take into account the convenience of advancement and the need to ensure accuracy. A broad definition of correctness includes both proper operation and other prerequisites essential for many growing distributed systems. It is expected that new methodologies for testing as well as troubleshooting and verifying systems would become important technologies that will allow for the development of operational processes.

M. Zhu and H. Pham in [5] posit that evaluating all aspects of a system is essential. If certain parts of the hardware or software stack are improved, it may have a cascading effect and result in changes across the whole structure. Researchers must therefore work at multiple levels, from proof-of-concept to deciding how the environment must transform, and new technologies have to be reviewed in an actual context with plausible workflows. When evaluating an overall system performance, configurability, and durability should be taken into account as well as other factors such as precision and system efficiency and safety,

As early as a month before the workshop, planners requested every respondent to submit one or even more suggestions regarding possible improvements for parallel and distributed systems which they believed should really be discussed. Participants were asked to submit ideas for concepts and discussion topics. These were made available beforehand of the conference so that participants could start preparing their opinions on them ahead of time. The workshop's planners also welcomed attendees to submit lightning discussion propositions for consideration. As part of the workshop's morning plan, there were spotlight discussions. During the afternoon, two breakout sessions were conducted, each preceded by an overview briefing from the first breakout session, which was organized around a cross-cutting concept, with each group focusing on a different theme. During the second breakout session (within computer science), each team focused on how their discipline could contribute to addressing the cross-cutting concepts and issues that were discussed. This article is arranged as a break-out. Section III is divided into two sections that evaluate the future trends for parallel and distributed computing: technology and application, and the cross-cutting concept.

III. CRITICAL ANALYSIS

Technology and Application

However, despite the growing technological challenges, the usage for computer technology will keep growing in all fields of science, technology, entertainment, administration, and business applications. As exploratory and comparative data are merged with modeling, data processing applications will be more intricate and distributed.

Applications

Scientists and engineers utilised parallel computing to model a wide range of physical system applications, from microstructures and hydrodynamics, to terrestrial ecosystem and astrophysics. However, due to the increase of socioeconomic, medicine, promotional, and pharmacogenomics datasets, the need for parallel processing has spread further than the natural sciences. With machine learning techniques becoming more prevalent, more output and parallel processing will be required. On existing systems, Deep Instructional strategies are among the most algorithmically-rigorous. They frequently perform on large sets of data or browse massive governmental sites.

Market for cost-effective, energy-efficient, and competent computer networks will increase in scientific research, technology, government and business as well as the arts and entertainment due to society's ambitious goals of uncovering highly complicated concepts in physical and human-constructed system applications. Parallel and distributed computing at existential level is driven by three diverse kinds of applications.

Machine Learning (ML)

Deriving configurations from massive data (yielded or discovered) using machine learning has revolutionized much advancement in engineering, government and business [6]. Machine learning techniques will continue expanding as more data is available, cataloged, and distributed across more realms. Technological advances in the field of interoperable data processing will allow machine learning models such as deep learning to function on data sources that are not completely communicated and might be encoded. A three-order-of-magnitude increase in the volume of data and simulation needed for the most effective image processing tasks has increased over the past eight years.

Data Analysis

Increased resolution, faster speeds, and lower costs of camera systems, detection systems as well as other sensors will support the growth of large datasets; but so too will the greater connectivity of individuals and organisations, making it much easier to accumulate and incorporate data sets in a more efficient manner. This has also been made possible by government programs that promote the open sharing of government subsidized scientific research. The advantages of incorporating data from different sources are well-known, whether you are display advertisement or evaluating microorganisms interplay in the surrounding. Big data issues also occur in minimally obvious sectors; for instance, existing practices to the opiate addiction dilemma include the development of biosignatures for addictive behavior across-omic (genomic information, etc.), output, detector, and patient records statistics for huge populations, with assessment flows conceivably leveling into zettabytes.

Simulation

High-throughput simulation advertisements, like the Materials Genome Initiative and weather forecasting experiments, have widened the definition of scientific simulation further than conventional HPC. Large-scale multi-faceted simulations of the complete environments are required to assess the recommended ecological sustainability technologies. Simulations might be integrated with optimization to mitigate more inverse issues within the application domain integrating the clinical imaging, larger scientific experiment, and distributed sensor in smartphones. Inverse issues that focus on calculations from a collection of observation, and the causal elements producing them, enhance the computational requirements by the elements of thousands of modern simulations. These pave way for novel numerical patterns to uncertainty and science qualifications.

Inverse issues are utilized to restructure three-dimensional molecular structures out of the x-ray imaging of massive experimental structures, define MRI dataset from clinical experiments and enhance our comprehension of the critical physical laws by matching simulation to cosmology and experiments. These forms of application are not different but will be combined and blurred. Domain professionals from the social science field might utilize natural languages to interface effectively with the provided software elements, which scrape information out of multiple websites around the globe, develop statistical frameworks of complex social actions and utilize those frameworks to make forecasts about the transitions in populations and transforming requirements for medical services, housing and infrastructure.

Urban environments will be instrumented with sensors, which are capable of detecting energy usage in construction and will potentially optimize complex energy grids of dynamic renewable sources of energy to effectively manage the creation of energy, its distribution and storage. Urban environments will also utilize detailed frameworks of the transportation model, developed from wide-range streams of data, to effectively optimize for conveniences, costs and energy effectiveness. Driverless motors will gather and assess some datasets in situ, but would also share information with the closer cars and stationary city services. Information from vibrational sensors, carbon sensors, biosensors will gather data with significantly lower signal-to-noise ratio and could be extracted to comprehend the safety of earthquakes and ecological threats, and to assist in the designing of safer structures, digital agriculture and pollution management.

Digitalized agriculture will aid the community to focus on massive challenges of providing food for the globe's growing populations. It will be entirely data-driven and integrate heterogeneous data from different forms of on-ground and in-ground sensors (airflow, mechanical, electrochemical, etc.) and the drone sensors, and satellites. It will utilize the aspect of machine learning and information analysis to effectively control plans for harvesting, fertilizing, watering, planting and apply these actions automatically. These digital systems have to be predictive of typical actions, adapting to transitions in day time, seasons and social patterns of application and have to adapt to more extreme occasions brought by cyber-attacks, systems failures, and natural attacks.

Deep - learning and optimization methodologies will be used to direct the discovery of a big state space of simulation and experimental results to make forecasts using simulation models which integrate first-principles physical laws with empirical evidence or modeling techniques discovered from statistics. In the future, labs will become highly automated, with automaton "cloud" labs for experimentations, and researchers will progressively reuse data from other research teams and specializations. Certain challenges will require combining noisy sets of data from different sources to gain new insights. These challenges will require that the data be easily discoverable and freely accessible while also being tagged and searchable by fully automated metadata computation. In the future, a trainee from a small university or rural junior high ought to be able to access the data from global experimentations and then use virtual computer technology to make scientific discovery.

When faced with huge datasets, data rates, and costly selection techniques, applications will be using relatively large as well as progressively inhomogeneous catalogs of computer resources as well as software applications. As distributed and parallel applications move beyond their homes in supercomputer hubs and data centres, they will be able to incorporate

devices ranging from flawed IoT technology endpoints to small locators. A big challenge in cloud computing is trying to decide where to place and when to transmit files and data processing in such system applications.

Technology

The end of Moore's Law and Dennard scalability is an important technological pattern that will continue to affect the configuration of distributed and parallel computing environments. The outcome is a motivation to enhance efficiency in other contexts, such as by expanding parallel processing at all thresholds and by utilizing different hardware incubators, such as digital signal processors (DSPs), graphics processing unit (GPUs), image signal processing units, etc. As the percentage of embedded processors increases, parallel processing becomes more feasible. This is a hybrid method that utilizes advantage of both. Following the end of Moore's Law, there is a novel opportunity to review new computing capabilities which may result in orders of amplitudes advancements in obtainable computational power by trying to break the modern computing extrapolation wherein simulations are hypothesised to generate specific values.

However, the trade-off is that these computational methods can only quantify results to a predefined accuracy or possibility. Evolving computational models, like computing device, probability computer, and artificial neural computer, guarantee intense arithmetic operations under modern computing constrictions e.g., power or energy. It is true that each one of these systems has its own advantages, but the release of new numerical methods, such as enhanced parallel processing by allowing data racing events with satisfactory constrained probabilities, is one of the biggest advantages of adopting estimated mathematical calculations (including establishing novel abstract concepts to understand and subvert them).

Cross-cutting Concerns

There will be new challenges and directions for research that will cross areas of study to interactive software and hardware components to identify complete software alternatives. The conventional Moore's Law improved performances have flatlined, so grid computing has become the principal factor for sustained and improved efficiency [8]. Due to the increased parallel processing, specialized computing techniques as well as increasingly complex memory, computation and system hierarchies, it will be necessary to revisit the software application abstract concepts. To keep up with this trend of increasing complexity in applications and their heterogeneity, data/computing centers and smart infrastructural facilities and the surroundings will be outfitted with complex systems of computational resources.

The future directions will integrate: (1) new designs of efficiency and accuracy (Computational Designs; Domain knowledge, knowledge-based applications, computational methods, and structures, and; platform-based structures that incorporate divergent software and hardware into compact and resourceful system applications.

Software Abstraction

It has become extremely advanced to use parallel or distributed hardware and software components to accomplish tasks. Because of this, programmers use reconfigurable and recyclable operating systems with several different levels of abstraction such as Software Development Kits (SDKs), development tools, software library resources and framework libraries, language middleware structures and traverse language interactions, among others [7]. For example, while existing abstract concepts have been immensely valuable in minimizing total complexity and increasing coder productive capacity, new research is required to investigate a number of issues. It becomes increasingly difficult to hide certain functionalities as the underlying code becomes more intricate and there is a greater need to have high degree of control. It is possible to create abstract concepts with a variety of objectives in mind, such as efficiency, expandability, ease of use and versatility, parallelism and re-usability (declaration).

As a result, users may find it extremely difficult to use infrastructures that allow them to manage the specifics of data movement and power consumption, while a platform that instantaneously handles these factors for them may incur substantial time or bandwidth consumption, or even fail to do so for some applications. There are a number of factors that can be taken into account for each of these requirements: application workloads, system scale, or energy budget. We have to do more investigations about how to develop abstract concepts for distributed and parallel solutions that satisfy such objectives, how much they can be concealed, as well as how to reveal them when needed in a magnificent and accessible functionality. Among overall abstract concepts and domain-specific abstract concepts, there is a constant struggle for dominance over the other. It is possible to perform formulaic load adjustment based on the general system load, or it is possible to use an application - layer catalog that takes advantage of the unique datatypes and computational methods. The use of distributed blocks, for instance, can maximize both routing protocols and geographical region, but the software tool is not transferable to any other domains.

IoT systems which use ml algorithms on a highly dynamic set of heterogeneous distributed nodes, including small network edge and potent cloud resources, are isolated examples coordinates in the parallel/distributed design process. Where can the paradigms for constructing such various applications and systems be brought closer together? Because of prospective technical problems arising from the interaction between the elements, content of different levels of abstraction could also be difficult, particularly in exceptional systems (e.g., resource contention). Multifaceted challenges arise when building libraries, such as differences among methodology and threading modeling techniques, ranging from the horizontal system design used by the prevalent message-passing interface known as MPI to more complex and dynamic multiprocessor and activator parallelism. Additionally, data model portrayals must be consistent across libraries, particularly for sparsity or hierarchies' datatypes, which are adapted to specific applications. Today, this indicates that

applications are intended to use a solitary library (such as PETSc or Trilinos) and the data formats and distribution patterns supported by that library (such as XML).

Traditionally, an API for a distributed/parallel ADT or provider is designed to resemble the API for the centrally-controlled variant as accurately as possible, e.g. by authorizing new theory. Developing inherently simultaneous performance requirements, as well as methodologies to constitute, authenticate, and validate them is an important research orientation. Conventionally, performance-oriented abstract concepts have focused on execution times and sometimes spatial dimensions. Energy-efficient abstract concepts are becoming a progressively active research area. Just as it is in cloud data centers, portable devices, and the Internet of Things, such ontologies are required. A datacenter's energy efficiency may be even more important economically than its data processing speed. Because of the advent of enhanced Non-Volatile Random-Access Memory (NVRAM), collection ontologies such as binary transitory disk space and block-oriented file system are being re-examined. It may well be necessary to reconsider this deeply embedded structure in order to get the most out of this innovation. Using various data portrayals for volatile and nonvolatile ram, should systems continue?

Distributed and parallel systems are affected by this in a big way, as well as sequenced and asynchronous applications in general [9]. Researchers must rethink the abstract concepts for offsite backup (both temporary and quasi) as well as file system, beginning with the inquiry of whether to use the same or distinct abstract concepts for them. Cloud services are increasingly relying on remote direct memory access (RDMA); and what were the appropriate abstract concepts? Due to the rapid expansion of the standard setters at all levels of experience, it is crucial to formulate abstract concepts that are useful to a wide range of individuals. Could we layout abstract concepts that are beneficial both for beginner programmers are operating on application areas with lackadaisical performance standards and specialist software developers operating on applications with stringent performance expectations?

Ontologies should be designed with real users, software and task scheduling in mind, so that they meet real needs. Contrarily, we could perhaps strive to create abstract concepts that are widely applicable, but not too tightly bound with particular purposes. Benchmark sets reflecting an adequately broad range of plausible operating modes for a specific application category can direct researchers to develop abstract concepts with a promising prospect for actual impact. For adaptation, rigorous configurations are essential to success. For regression testing and performance stereotyping purposes, the deployments should endorse resource APIs.

Computational Frameworks

To endorse structured rationalization about such qualities at the tier of classifiers, as well as to endorse interpreting such methodologies to installations which run on pragmatic computational resources while still maintaining these formal characteristics, potential developments will develop new first-order apprehensions which need to be enveloped in new computer modelling. Radically different computations are required for these application areas, which must be conveyed throughout the entire digital stack, from computational characterizations to cloud computing and components. As part of our technical requirements of a mathematical computation, we endorse rationalization about computational load and also workable appropriateness.

To reconsider computer simulations, nevertheless, there are three major problems to overcome. There will be a wide range of potential developments from small devices to large data centres or computing systems; 2) Intelligent devices would be diverse, incorporating different components with different algorithmic characteristics (such as effectiveness, consistency, and confidentiality); and 3) Applications in highly distributed environments will assistance participatory, practical applications. Intractable problems exist at all levels of complexity. In particular, designs and methodologies that are assertive on multiple levels at once are of interest to researchers. Classifiers and implementation methods have to be synchronized to every level's variables as with traditional multi-level or multi-scale data processing fabrics. If you'd like to maximize your software for the system memory of the console you're working on, you can design a program for circuit transitions such as unwrapping or tiled paving or obstructing.

As a result of this, many questions remain on how to design effectual algorithmic structures with deep memory hierarchical structures. Humongous multi-scale structures will also have a diverse collection of tools with a variety of interfacing options. In order to integrate evolving devices into asset or asset frameworks, innovative simulation approaches for such devices will need to be developed. Acknowledging future NVRAM innovations is of particular importance. Because of the high cost of publishes in Intel's Optane NVRAM, publish frequency band is approximately six to ten times greater than write throughput, while DRAM's ratio is generally 1.2 to 1.7. Reading and writing data from DRAM has traditionally been treated as separate operations. In order to take advantage of the much larger price gap between NVRAM and DRAM, computational methods and methodologies will need to be developed.

There will be a variety of numerical simulations for various types of devices. Some devices may be more accurately described by their worst-case error or ordinary error, which differs from the greatest categorization of another device in terms of reliability or precision. The challenge of creating computer simulations that consolidate distinct portrayals is still unsolved. Many potential developments will continue operating in real-time or immersive data processing configurations, which must be taken into account when reifying device characteristics in computer modelling (latency, bandwidth, energy, authority, etc.). As a result, designers will be looking for assurances about response time and processing times in these modes of interaction. A worst-case responsiveness or latency may be needed by various applications. Uncertainty is intrinsic to very complicated geometries: supercomputers may be adaptively planned on an array of diverse devices, each one with various performance characteristics and serviceability levels. There are boundaries on numerous points in system

performance distributions, along with tail limits and variability, because of the size of these processes and arbitrary nature in computational efficiency at scale.

If you are interested in a resource-agnostic strategy, you'll have the same opportunity as with multi-level recollection hierarchical structures, in that a single strategy accomplishes relatively close resource consumption irrespectively of the variables of the resource management. It's also vital to build straightforward model that can predict the major characteristics of these recollection hierarchical structures, and to use these designs to design algorithms that work on actual machineries with such recollection hierarchical structures. Combining these efforts offers the incentive to optimize programability by incorporating resource requirements into evolving abstract concepts (language groups) and automating techniques that instantaneously traverse the space of optimization techniques and configuration options.

Signalling and ensure timing characteristics can be formalized in a way that is easy to understand, which a great opportunity is for researchers. Factors include navigating duration as an alternative measure in the optimal control space, such as trading resources or focusing on durability or reliability in order to meet specific timing goals. Many numerical methods are large or rely on presumptions from large data sets, so it will be vital to analyze how and where to execute them efficaciously under time restraints. It may be necessary in mathematical modelling for real-time interactive content by integrating broadcast arithmetic operations that facilitate configurable changes and updates to theoretical calculations as datasets, workloads and environments alternate.

Domain-Specific Designs

When it comes to Domain-Specific Design (DSD), this is all about developing comprehensive optimization techniques solutions that enhance our goals for any particular domain. We want a domain-specific, as well as conceivably platform-specific abstract concept, which contrasts with the Heterogeneity subject in which the primary objective is to construct adaptable abstract concepts that patch throughout all systems. DSD can affect hardware, libraries, programming languages, and models. Experimentation has shown a positive correlation between the complete system (optimization techniques) and a given application's supercomputing abstract concepts. Take into account, for instance, CNN (convolutional neural networks). It was thanks to the contract on sharable elements that designers and engineers from industry and academia across the data processing pile were able to innovate at both system level (TPU, convolutional core GPU on a microprocessor), as well as at the system level (a pervasiveness of architectures constructed on common libraries, such as PyTorch, Machine learning, Horovod and others). Identifying the correct abstract concepts for a particular domain is challenging, but the potential reward is enormous because they can activate advancement along multiple integrations [10].

DSD's greatest obstacle is attempting to reach the maturity and intelligence necessary for wider adoption. If any portion of the ecological system is changed, whether it be the way software is developed or whether it be the software or the hardware, it is indeed likely to cause a cascade of problems vertically and horizontally the pile. As a result, there is a natural aversion to modify in the sector. The majority of accomplishments have come from long-term investments, which are generally backed by another industry or market segments. Because of a large gamers and graphic elements market for GPUs, the GPU is now extensively recognised as an effective and valuable architectural style in computer technology but this was the outcome of fifteen years of operating system investment. Correspondingly, the integrated applications industry was primarily responsible for the growth of Field Programmable Gate Arrays (FPGAs) before they would be widely adopted in other locations.

Establishing how ecosystems really have to change requires multi-scale study and research development. That would be a lot to take on for an only one academic analysis. In order for advancements to occur at a specific point in the hardware/software array, a portfolio of data analysis should be taken into account, allowing them to be evaluated in a bigger framework and, when acceptable, carried through to attainment. Promising major developments require years of research & innovation before they can be launched on the market, potentially over a period of a decade or more. A co-design procedure needs purchase from community members as well as other neighbourhoods up or down the layer. Optimizer communities will need to implement different server APIs in order to create commands that manipulates them..

DSD has a wide range of potential applications. Due to the simplicity of the basic information in genomics, DSD has the potential for greater improvements in efficiency. Smart objects and IoT applications have a strong motivation of using DSD in order to attain the intended features and functions with restricted community resources, which is another strong motivation (power, processing energy, etc.). DSD gives substantial research programs, even at the highest possible level of software development. A wide range of research areas have improved at around half the percentage of maximum improving performance over the last decade, according to a new study. It was lately used to compare and contrast two NSF centerpiece computing mechanisms: Blue Waters and Frontera - using the NSF's Sustained Petascale Performance (SPP) reference point consisting of a combination of eleven full applications and reflective large scientific research challenges.

Contrary to the per-node comparison, Frontera outshone Blue Waters in terms of SSP achievement by a percentage greater than 8 to 1. However, the optimum efficiency between both the different schemes was 16 to one. Domain-specific models pledge software performance improvements that are closer to the maximum level of performance than generic models. Such study aims to look at ways to improve performance by modifying mainline design concepts to effectively suit a particular field of study. "Temporarily reprogrammable" architectures have the potential to speed up numerous domains, but little consideration has been devoted to this field of computing technology. Despite the fact that most DSD study is domain-specific, a few related literature and studies promote the implementation of DSD along a wide range of fields and industries. Researchers are focusing on methodologies that make it much easier to specialize and broaden an

infrastructure, which helps DSD distinctively. It is possible to add domain-specific hardware compatibility by developing computer languages and optimizations that are expandable.

IV. CONCLUSION

With the proliferation of data processing, the landscape of computer technology has transformed. Compilation takes place across a broader range of domains than ever before, and innovative applications in artificial intelligence, machine learning, and autonomy are among the most computationally expensive ones yet. Distributed and Parallel Computer technology is at the heart of this advancement, providing the fundamental abstract concepts, computational methods, and mechanisms for computing across multiple devices at scale. As a result, scientific investigations in parallel and distributed computing will need to proceed for these new applications' success to be assured. There are new opportunities to produce software/hardware solutions further easily because of previous advances in open-sourced programmable logic architecture designs and embedded system implementation methods. At all tiers of the data processing stack, from bare metal to the cloud, more research is required for devices to be more easily specialized and extensible.

References

- [1]. K. Siau and Z. Shen, "Mobile communications and mobile services", *International Journal of Mobile Communications*, vol. 1, no. 12, p. 3, 2003. Doi:10.1504/ijmc.2003.002457.
- [2]. P. Saariluoma and A. Oulasvirta, "User Psychology: Re-assessing the Boundaries of a Discipline", *Psychology*, vol. 01, no. 05, pp. 317-328, 2010. Doi: 10.4236/psych.2010.15041.
- [3]. A. Kahng, "Scaling: More than Moore's law", *IEEE Design & Test of Computers*, vol. 27, no. 3, pp. 86-87, 2010. Doi: 10.1109/mdt.2010.71.
- [4]. C. FU and S. XU, "WAPM: A parallel programming model in large scale Internet distributed computing environments", *Journal of Computer Applications*, vol. 29, no. 8, pp. 2161-2166, 2009. Doi:10.3724/sp.j.1087.2009.02161.
- [5]. M. Zhu and H. Pham, "A Novel System Reliability Modeling of Hardware, Software, and Interactions of Hardware and Software", *Mathematics*, vol. 7, no. 11, p. 1049, 2019. Doi:10.3390/math7111049.
- [6]. C. Meillier, F. Chatelain, O. Michel and H. Ayasso, "Nonparametric Bayesian Extraction of Object Configurations in Massive Data", *IEEE Transactions on Signal Processing*, vol. 63, no. 8, pp. 1911-1924, 2015. Doi: 10.1109/tsp.2015.2403268.
- [7]. N. Altintas, S. Cetin, A. Dogru and H. Oguztuzun, "Modeling Product Line Software Assets Using Domain-Specific Kits", *IEEE Transactions on Software Engineering*, vol. 38, no. 6, pp. 1376-1402, 2012. Doi: 10.1109/tse.2011.109.
- [8]. A. Lo, "Moore's Law vs. Murphy's Law in the Financial System: Who's Winning?", *SSRN Electronic Journal*, 2016. Doi: 10.2139/ssrn.2720724.
- [9]. C. Woodside and G. Monforton, "Fast allocation of processes in distributed and parallel systems", *IEEE Transactions on Parallel and Distributed Systems*, vol. 4, no. 2, pp. 164-174, 1993. Doi: 10.1109/71.207592.
- [10]. M. Severino and Y. Peng, "Machine learning algorithms for fraud prediction in property insurance: Empirical evidence using real-world microdata", *Machine Learning with Applications*, vol. 5, p. 100074, 2021. Doi: 10.1016/j.mlwa.2021.100074.