2022 IEEE 20th Jubilee World Symposium on Applied Machine Intelligence and Informatics (SAMI) | 978-1-6654-9704-6/22/$31.00 ©2022 IEEE | DOI: 10.1109/SAMI54271.2022.9780804

Fault detection in GPU-enabled Cloud Systems – An Overview

Farida Asadova*, Gábor Kertész†, Róbert Lovas‡ and Sándor Szénási§
*Doctoral School of Applied Informatics and Applied Mathematics, Óbuda University, Hungary
†‡Institute for Computer Science and Control (SZTAKI), Eötvös Loránd Research Network (ELKH), Hungary
†‡§John von Neumann Faculty of Informatics, Óbuda University, Hungary
*asadovafarida@stud.uni-obuda.hu, †gabor.kertesz@sztaki.hu, ‡robert.lovas@sztaki.hu, §szenasi.sandor@nik.uni-obuda.hu

Abstract—**Fault detection and handling are crucial tasks in cloud systems. As these infrastructures are growing and evolving, manual monitoring and interaction have become less feasible. To deal with this issue, monitoring systems are developed to track the behavior of the various components (e.g. nodes) in cloud systems, as well as the served applications in the virtual environment. Nowadays, most cloud environments provide graphics accelerators for their users leading to different problems. However, the application of GPUs in deep learning could also help the detection of incorrect behavior. In this paper, a short overview of cloud monitoring and fault detection methods is given focusing on GPU-enabled nodes.**

Index Terms—**Graphics Processing Unit, machine learning, cloud systems, fault detection, deep learning**

I. Introduction

As cloud technologies are developing rapidly, and more and more applications are migrated to cloud environments, stability and management of cloud systems became crucial. Despite cloud monitoring gets less attention, precise and continuous monitoring activities are vital to detect faults to accurately operate cloud processes. Cloud monitoring helps review, observe and manage the complex processes of cloud infrastructure [1]. Extended monitoring creates logs with several entries. These are difficult to process, identification of actual faults on time as well as providing possible solutions before serious errors occur are also challenging.

The logs produced and collected by a monitoring system could contain memory alerts, unreachable server alerts, application performance alerts, Service-Level Agreement (SLA) expiration. The effective and speedy mining of the collected huge amount of monitoring data by using Graphics Processing Unit (GPU) is bringing about rapid detection and prediction of the failures occurring in the cloud system.

The general concept of fault is defined in the manuscript by Farooq et al. [2] as ”A fault can be defined as abnormal condition that may cause a reduction in, or loss of, the capability of a functional unit to perform a required function.”. Failures are noticeable outcomes of the faults and are related to software executions [2]. To prevent failures the error behind the faults should be discovered before failures occur. These errors are usually due to program misinterpretation or inaccurate handling in software development. Errors are classified as network, software, permanent, transient, intermittent in the survey written by Kumari et al. [3]

Cloud monitoring helps collect continuous data with a large amount of information. To maintain instant and reliable service, fault detection is important. Generally, failure data is relatively small rather than the overall gathered logs from the cloud routine checks. That creates adversity to extract, evaluate and correct certain faults on the services. Based on the manuscript, written by Swetha et al. [4] types of faults are classified as network faults, physical faults, media faults, processor faults, process faults, service entry faults, permanent, transient, and intermittent faults. The classification of each type of fault is connected to the particular part of the cloud where erroneous logs are recorded [4].

An effective method introduced in the paper written by Gao et al. [5] named the auto-encoder model, which is more efficient than traditional methods of fault detection. In this model unsupervised learning is featured to extract essential characteristic data. Hereby deep auto-encoder neural network was introduced to extract erroneous data and provide predictive output as the result of the automatic learning of fault detection in cloud computing [5].

Fault detection requires the analysis of a large amount of data due to the complex infrastructure of cloud systems. Traditional Central Processing Unit (CPU) methodology, which is good at general computations, is not efficient for the complex and time-consuming Big Data mining tasks. Therefore, GPUs are extensively used to accelerate data analysis, split data into parallel processes and perform predictions.

A GPU is a specialized electronic circuit designed to speed up the production of rendered images. GPUs are used ingame consoles, personal computers, research workstations, and nowadays in mobile phones as well.

Computer processors are designed to handle a large number of instructions. However, extensive processing time, highly complex combinations of these commands are not practicable but the GPUs are very useful and consume less compilation power and time. That is why we used GPU instead of the traditional processor to compile the huge amount of complex calculations [6]. NVIDIA in 1999 popularized the term ”GPU” by presenting the GeForce 256 as ”the world’s first GPU”. It was shown as a ”single-chip CPU with built-in transform, lighting, triangle setup/clip, and auto rendering” [7].

The GPUs can perform various scientific tasks with high computational acceleration. Up-to-date computers are integrated with GPU to take over graphics part from CPU processor. In the beginning, GPUs were designed to proceed with graphics tasks. Nevertheless, nowadays it is equipped to perform complex calculations that are parallel programmable. GPUs are programmable processors with a streaming processing model. The speed of this kind of processor exceeded 100 teraflops (TFLOPs) [8].

Currently, calculations on GPU are being applied to the listed fields such as: image/sound processing [9], molecular dynamics simulation [10], numerical computation [11], computational biology [12], financial calculations [13], database [14], cryptography [15], adaptive radiation therapy (ART) [16], bioinformatics [12], [17], computer vision [18], Big data mining [19], geoinformation [20], military target detection [21], magnetic resonance imaging (MRI) [22], neural network [23], artificial intelligence [24], surgery simulation [25], ultrasound [26], video conferences [27] and etc.

A simple method of understanding the basic difference between a GPU and a CPU is that the CPU consists of a few cores which are optimized for serial processing but the GPU has an architecture based on several processing cores which are smaller. The architecture of GPU is designed to execute data-parallel computations.

II. Fault detection in cloud systems

While cloud systems are growing and evolving, system failures remain unavoidable. Fault detection within the cloud system requires speedy processing of monitoring data to achieve prediction of system failures to prevent the possible obstacles on time. Not all faults occurrence result in failure, the severity of them should be classified by their priorities. The failure terminology defines the concept of reliability, which is defining the service quality of the cloud providers.

Fault detection in cloud systems is performed by using the following approaches based on the article by Smara et al. [28]. The first one is called Intrusion or anomaly detection system [28]. The main focus of this type of detection is the network or host intrusions. Anomalies are detected based on behavior analysis. Signature-based and anomaly-based detection are the subgroups of intrusion detection. Signature-based detection contains a predetermined database of already experienced anomalies with particular priority. But the anomaly-based detection observing for atypical patterns inside the log data. An effective anomaly detection system for fault-tolerant network management in cloud data centers is presented particularly in the article by Abbasi et al. [29]. Software-defined networking provides easier administration, network control features, and a programmable console. This feature could be scheduled for the reading network by subnets, helps to utilize the best available path in the network and fault management of cloud network [29].

In cloud computing statistical, data mining, and machine learning anomaly-based detection methods are available. Statistical type of detection [28] compares the data with the stored ideal conditions and is able to detect unpredictable faults.

Compared to that, the data mining detection method [28] alerts faults with the help of rule-based technologies, such as Classification, Clustering, and Association. This method can distinguish between ordinary and erroneous actions. Hence, it is a widely used technique since it does not require preparatory knowledge to process the data. The main disadvantage of this method is, it can produce many false alerts.

In machine learning detection method, the system detects the faults and stores their values. The system improves its performance by the ability to learn from the prior values of the faults. Machine learning can detect faults by itself, but it is much more efficient when it is combined with statistical and data mining methods. Therefore, hybrid methods are being used in cloud systems, which merely requires computational expenses.

The second approach is heartbeat and pinging method [28]. In the heartbeat method monitoring device continuously checks the fault detector to whether the fault is available. If the detector does not respond before timeout the heartbeat method considers the device erroneous. The way around process conducts in pinging method, the fault detector sends a message to the monitoring device to confirm the faulty device. Both methods are used for persistent hardware fault detection.

Another technique of handling the gathered data is by analyzing them. In such a system with extensive data, it is not possible to rely upon a human supervisor to identify all issues, some level of automation is vital. Server interruptions cause huge financial losses to organizations. For instance, Amazon lost $72 million during a one-hour interruption on Prime Day [30]. In addition, collective remote access to the virtual machines causes insufficient memory problems, interruption of services usage in the cloud systems. In the paper by Legashev et al. [31] collective access issues in educational institutions are investigated and an effective scheduling method with Simulated Annealing and Genetic Algorithm is provided.

Cloud monitoring as a service measures the infrastructure and continuously checks the behavior of the applications [32]. Such a tool consists of a data-gathering module and a data processing module, as well as user and application interfaces [33]. Data gathering continuously collects and stores the messages from the environment. Data processing may consist of a visualization tool, to help the explanation of the current status.

Zhang et al. [34] identified, that there are two main categories in cloud fault detection: rule-based detection and detection based on statistics. Rule-based detection methods can be based on simple rulesets on the error message and record components, or basic decision trees can be built using multiple rules and queries. The resemblance of previously found faults are based on other methodologies.

Lovas [35] proposed a method and a framework that not only monitors but also actively controls the cloud-based deployment processes leveraging on the so-called macrostep [36] based execution. The approach allows the manual rule-based evaluation by each macrostep, i.e. by each collective breakpoint sets.

Statistics base methods are nowadays based on machine learning [34]: Neural Networks (NN) and Support Vector Machines (SVM) are often used as classificators based on input observations [37]–[41]. In the article of Zuzana et al. [42], statistical-based functionalities are developed to extract features from the sample speech data in the detection of dysphonia.

There are few manuscripts [43] which describe the possibility to apply GPUs in cloud fault detection. The paper by Abusitta et al. [44] proposes a very effective GPU-enabled deep learning model which is relevant in Cloud-based Intrusion Detection Systems. During the training of a deep neural network, GPUs can be applied with great effect.

Ozaki et al. [45] stated, that system failures can be detected by Black-box monitoring and White-box monitoring.

1) **Black-box monitoring**- cloud system is being monitored by external vendors. This method is not effective in providing detailed system information.
2) **White-box monitoring**- monitoring systems function inside the operating system. This method is more accurate on failure detection by having internal knowledge of the system.

Ozaki et al. [45] also introduced the GPUSentinel development environment to perform white-box monitoring and detect the system failures [45]. GPUs are able to investigate various symptoms of system failures in parallel, several examinations have been performed such as the detection of CPU anomalies, detection of out-of-memory, detection of deadlocks.

III. Applications of Graphical Accelerators

Accelerating the process of fault analysis and prediction of certain failures can be done in a short time frame in GPU enabled cloud environments. In this section, we provide an overview on cloud environments and machine learning technologies with GPU. Software fault detection models can be applied to cloud environments as well. ML techniques are listed according to Pandey et al. [46] as Bayesian learners, Decision Tree, Evolutionary Algorithm, Ensemble Learners, Neural Networks, Support Vector Machine, Rule-based learning, miscellaneous. Fault detection ML-based model defines the faulty and non faulty partition over training set and validated the outcomes of test or development set. The development of fault detection system is a sequential process. First, datasets for various data repositories. After that, they do data preprocessing such as data cleaning, filling missing values, normalization of information, removal of duplicate data, etc.

A. GPU usage in Cloud Computing

In recent years many cloud computing related GPU solutions were introduced. GPUs provide high-speed computation and graphical processing capabilities using Cloud servers.

- **Google Cloud GPUs**- Google Compute Engine provides GPUs that can be added to virtual machine instances. The particular workloads on the instances can be accelerated via GPU usage, such as machine learning and data processing. Many NVIDIA GPU machine types can be chosen for the Compute Engine. All the details regarding the GPU models is specified in the documentation [47].
- **Tencent Cloud GPU Cloud Computing**(GCC)- It is a GPU-based computing solution appropriate for various scenarios including content and breadth of the profound learning/detection methods, processing of the graphics, and various scientific computations. GCC may be managed quickly and easily as the standard instance for the Cloud Virtual Machine (CVM). GCC's higher computer performance ensures a powerful, single-instance floating points operation with GPU and CPU to produce 125.6 single-accuracy teraflops and 62.4 double-accuracy teraflops. A secure and reliable network environment and extensive safety services are provided by GCC: GCC instances are located in a 25 Gbps (10 Gbps for select instances) network environment with low private network latency and great processing capacity. CVM, Virtual Private Cloud (VPC), Cloud Load Balancer enables GCC interconnects (CLB). Full security group and network ACL settings provide controls of incoming and outgoing network traffic and security filters from and to instances and subnetworks. [48].
- **Lambda GPU Cloud Deep Learning**- is providing AI, ML, and deep learning models. Every Lambda Stack Virtual machine is preinstalled. Lambda Stack contains major deep learning frameworks and CUDA drivers. Furthermore, direct SSH access possibility maintained [49].

B. GPU usage in Machine learning

GPUs are mainly utilized in deep learning technologies, which is the subset of machine learning and it is having more capabilities by using multiple hidden layers in artificial neural networks. Moreover, deep learning is one of the key in-demand technology with its abilities nowadays, among the activities considerably benefiting from parallel processing. Deep learning algorithms imitate activity in neuronal layers in the human brain that enable robots to understand language, detect patterns or make music [50].

The existing framework Theano [51] is written in Python language. The purpose of the library is to further develop both the time consumption for the execution and the development phase of machine learning, in particular algorithms of deep learning. It is a symbolic engine designed to optimize and execute tensor expressive graphs. A repetitive calculation of a complex mathematical expression is fundamental to many machine-learning applications. This expression can often be stated succinctly in the context of the matrix or tensor operations, even though it is not the optimal technique for calculating the complete expression.

For such mathematical expressions, Theano provides a high standard language description and a compiling device that employs traffic tricks, heuristics, supplementary libraries and even GPU to evaluate such maths as fast and accurately as feasible. Theano makes it simpler to use the Python language's rapid development pattern, define mathematical methods as you like, and benefit from incredibly quick code.

Theano is open source, BSD licensed software. The work by Huzmiev et al. [51] introduces Theano through logistical regression and describes the functional language mathematical methods and enjoys high-speed coding. Also, it explains the content and scope of the Deep Learning Tutorials, which demonsrate how Theano can be applied for deep learning [51].

Another suggested library is GPUMLib (GPU Machine Learning Library), which is proposed in the article [52].

TensorFlow is open-source platform for machine learning. It features an extensive and flexible ecosystem of tools, libraries, and communities that enable academics to advance the latest ML and developers to create and implement ML-powered applications effortlessly [53]. TensorFlow code and Keras models run transparently with no code changes on a single GPU. Distribution Strategies are the easiest technique to execute on several GPUs, on one or many PCs [54].

Keras is an industry-strength framework that can scale to large clusters of GPUs. Keras models can be exported to JavaScript, to run directly from the browser [55].

Pytorch is an open-source machine learning library built on the Torch library, mainly developed by Facebook's AI Research lab. PyTorch offers several tools to facilitate distributed training, including DataParallel for single-process multi-thread data-parallel training using multiple GPUs on the same machine, DistributedDataParallel for multi-process data-parallel training across GPUs and machines, and framework for general distributed model parallel training [56].

GPUs are mainly utilized for deep learning, particularly during training, since they offer a better speed of performance compared with similar investments in CPUs. The GPU hardware and software optimization for expediting the operations of tensors present in deep learning has been focused in particular. In Volta architecture, NVIDIA introduced a specific functional unit in the Tensor Core. Tensor cores are also found on NVIDIA's more recent Turing architecture and NVIDIA's T4 Turing-base GPUs are further optimized for inference tasks. The tensor cores on the Tesla V100 GPU deliver three times more speedup during the accurate training by statements of NVIDIA. Five out of Six 2018 tensor cores were used by the Gordon Bell Award Finalists to improve applications and three especially to accelerate machine learning [57].

This is not surprising that GPUs, which are extensively utilized in autonomous path detection, medical imaging, super-computing, and machine learning, have the capacity to process 10 to 100 times faster than conventional CPUs. That is practiced in one of the foremost factors that Graphic cards are used to power some of the most powerful, deep learning neural networks [6].

The efficiency of parallel execution depends on the bandwidth of memory. CPUs can swiftly pick up little memory packages, while GPUs have a large delay that slows them down in this kind of work. However, GPUs are ideal when it comes to fetching exceptionally large amount of data from memory and the best GPUs can achieve up to 750 GB/s, which is significantly higher then the typical 50 GB/s memory bandwidth of typical CPUs [6], [58].

Latency problems in processors require more than one processing unit to be used. GPUs, unlike CPUs, consist of thousands of cores and perform large-scale memory and matrices tasks. The unloading process takes a lot of time to queue up for the GPU to begin the download process, every consecutive fetching becomes apparent sooner. In order to enable the GPU to manage high bandwidth, latency is masked with so much processing power. The second reason is that GPUs exceed traditional CPUs in the field of deep learning which is called thread parallelism [6], [58].

IV. Conclusion

This paper investigates the overall GPU-enabled cloud concept in the scope of fault detection. Fault detection methods are classified. Furthermore, widely used GPUs in cloud computing are investigated. The most well-known cloud GPU options have been gathered.

GPUs are widely used in cloud computing. Many cloud companies have GPU machine types to accelerate the processes. Hereby, the most prominent GPU solutions are discussed. Google Cloud GPUs, Tencent Cloud GPU, Lambda GPU Cloud for Deep Learning are some of them. There is still room for the investigation to conduct a comparison between them and collect statistics.

In the future, fault detection and prediction could be broadened to include more varieties of faults and associated specifications, along with monitoring technologies. A larger dataset and the creation of a new categorization model would be required. The proposed fault diagnosis method's runtime could be shortened thanks to the GPU's comprehensive parallel implementation. Additionally, expand the use of GPUs in multiple technologies to boost cloud resource management.

In this paper, we focused on the broad overview by referring to various materials and documentation. The effect of GPUs in fault detection and prediction in cloud systems should be analyzed in real-life monitoring processes, for example using the significantly upgraded ELKH Cloud infrastructure that has been serving a large number of machine learning and GPU accelerated scientific projects of the Hungarian research community [59]. The prepared overview opens new areas of research for investigation, and might serve some valuable contributions to the research related to debugging of orchestrated cloud applications [35].

Acknowledgment

The authors thankfully acknowledge the support of the Doctoral School of Applied Informatics and Applied Mathematics, Óbuda University. The presented work was supported by the ÚNKP-21-5 New National Excellence Program of the Ministry For Innovation And Technology from the source of the National, Research, Development and Innovation Fund. We gratefully acknowledge the financial support of the Hungarian Scientific Research Fund (OTKA K 132838). The presented work of R. Lovas was also supported by the János Bolyai Research Scholarship of the Hungarian Academy of Sciences.

REFERENCES

[1] G. Aceto, A. Botta, W. De Donato, and A. Pescapè, "Cloud monitoring: Definitions, issues and future directions," in *2012 IEEE 1st International Conference on Cloud Networking (CLOUDNET)*. IEEE, 2012, pp. 63–67.

[2] S. U. Farooq, S. Quadri, and N. Ahmad, "Metrics, models and measurements in software reliability," in *2012 IEEE 10th international symposium on applied machine intelligence and informatics (SAMI)*. IEEE, 2012, pp. 441–449.

[3] P. Kumari and P. Kaur, "A survey of fault tolerance in cloud computing," *Journal of King Saud University-Computer and Information Sciences*, vol. 33, no. 10, pp. 1159–1176, 2021.

[4] S. Swetha and S. K. Venkatesh, "Fault detection and prediction in cloud computing," *International journal of trend in scientific research and development*, vol. 2, no. 12, p. 24, 2018.

[5] W. Gao, Y. Zhu *et al.*, "A cloud computing fault detection method based on deep learning," *Journal of Computer and Communications*, vol. 5, no. 12, p. 24, 2017.

[6] P. Ongsulee, "Artificial intelligence, machine learning and deep learning," in *2017 15th International Conference on ICT and Knowledge Engineering (ICT&KE)*. IEEE, 2017, pp. 1–6.

[7] NVIDIA, "What's the difference between a cpu and a gpu?" https://blogs.nvidia.com/blog/2009/12/16/whats-the-difference-between-a-cpu-and-a-gpu/, [Online] Accessed 29 September2021.

[8] M. Romero Hortelano, "Applications of gpu computing to control and simulate systems," *Actas de las XXXIX Jornadas de Automática, Badajoz, 5-7 de Septiembre de 2018*, 2018.

[9] B. Cowan and B. Kapralos, "Spatial sound for video games and virtual environments utilizing real-time gpu-based convolution," in *Proceedings of the 2008 Conference on Future Play: Research, Play, Share*, 2008, pp. 166–172.

[10] J. Yang, Y. Wang, and Y. Chen, "Gpu accelerated molecular dynamics simulation of thermal conductivities," *Journal of Computational Physics*, vol. 221, no. 2, pp. 799–804, 2007.

[11] V. Kindratenko, *Numerical computations with GPUs*. Springer, 2014.

[12] M. S. Nobile, P. Cazzaniga, A. Tangherloni, and D. Besozzi, "Graphics processing units in bioinformatics, computational biology and systems biology," *Briefings in bioinformatics*, vol. 18, no. 5, pp. 870–885, 2017.

[13] X. Tian and K. Benkrid, "High-performance quasi-monte carlo financial simulation: Fpga vs. gpp vs. gpu," *ACM Transactions on Reconfigurable Technology and Systems (TRETS)*, vol. 3, no. 4, pp. 1–22, 2010.

[14] S. Breß, M. Heimel, N. Siegmund, L. Bellatreche, and G. Saake, "Gpu-accelerated database systems: Survey and open challenges," in *Transactions on Large-Scale Data-and Knowledge-Centered Systems XV*. Springer, 2014, pp. 1–35.

[15] O. Harrison and J. Waldron, "Gpu accelerated cryptography as an os service," in *Transactions on computational science XI*. Springer, 2010, pp. 104–130.

[16] C. Men, X. Jia, and S. B. Jiang, "Gpu-based ultra-fast direct aperture optimization for online adaptive radiation therapy," *Physics in Medicine & Biology*, vol. 55, no. 15, p. 4309, 2010.

[17] A. Welivita, I. Perera, D. Meedeniya, A. Wickramarachchi, and V. Mallawaarachchi, "Managing complex workflows in bioinformatics: An interactive toolkit with gpu acceleration," *IEEE Transactions on NanoBioscience*, vol. 17, no. 3, pp. 199–208, 2018.

[18] Y. Allusse, P. Horain, A. Agarwal, and C. Saipriyadarshan, "Gpucv: an opensource gpu-accelerated framework forimage processing and computer vision," in *Proceedings of the 16th ACM international conference on Multimedia*, 2008, pp. 1089–1092.

[19] H. Jiang, Y. Chen, Z. Qiao, T.-H. Weng, and K.-C. Li, "Scaling up mapreduce-based big data processing on multi-gpu systems," *Cluster Computing*, vol. 18, no. 1, pp. 369–383, 2015.

[20] S. Discher, R. Richter, M. Trapp, and J. Döllner, "Service-oriented processing and analysis of massive point clouds in geoinformation management," in *Service-Oriented Mapping*. Springer, 2019, pp. 43–61.

[21] Y. Tarabalka, T. V. Haavardsholm, I. Kåsen, and T. Skauli, "Real-time anomaly detection in hyperspectral images using multivariate normal mixture models and gpu processing," *Journal of Real-Time Image Processing*, vol. 4, no. 3, pp. 287–300, 2009.

[22] M. Hernández, G. D. Guerrero, J. M. Cecilia, J. M. García, A. Inuggi, S. Jbabdi, T. E. Behrens, and S. N. Sotiropoulos, "Accelerating fibre orientation estimation from diffusion weighted magnetic resonance imaging using gpus," *PloS one*, vol. 8, no. 4, p. e61892, 2013.

[23] K.-S. Oh and K. Jung, "Gpu implementation of neural networks," *Pattern Recognition*, vol. 37, no. 6, pp. 1311–1314, 2004.

[24] W. J. Dally, C. T. Gray, J. Poulton, B. Khailany, J. Wilson, and L. Dennison, "Hardware-enabled artificial intelligence," in *2018 IEEE Symposium on VLSI Circuits*, 2018, pp. 3–6.

[25] K. Sase, A. Fukuhara, T. Tsujita, and A. Konno, "Gpu-accelerated surgery simulation for opening a brain fissure," *Robomech Journal*, vol. 2, no. 1, pp. 1–16, 2015.

[26] H. So, J. Chen, B. Yiu, and A. Yu, "Medical ultrasound imaging: To gpu or not to gpu?" *IEEE micro*, vol. 31, no. 5, pp. 54–65, 2011.

[27] A. R. Brodtkorb, T. R. Hagen, and M. L. Sætra, "Graphics processing unit (gpu) programming strategies and trends in gpu computing," *Journal of Parallel and Distributed Computing*, vol. 73, no. 1, pp. 4–13, 2013.

[28] M. Smara, M. Aliouat, A.-S. K. Pathan, and Z. Aliouat, "Acceptance test for fault detection in component-based cloud computing and systems," *Future Generation Computer Systems*, vol. 70, pp. 74–93, 2017.

[29] A. Abbasi, S. Band, M. A. A. Al-qaness, A. Abbasi, N. Al-Jallad, and A. Mosavi, "Resource-aware network topology management framework," *Acta Polytechnica Hungarica*, 02 2020.

[30] F. A. April Berthene, Josh Neblett, "Digital commerce 360. the potential cost of amazon's prime day," https://www.digitalcommerce360.com/2018/07/17/the-potential-cost-of-amazons-prime-day-miss-72-million/, Aug 2018, [Online] Accessed 9 October 2021.

[31] L. V. Legashev and I. P. Bolodurina, "An effective scheduling method in the cloud system of collective access, for virtual working environments," *Acta Polytechnica Hungarica*, vol. 17, pp. 179–195, 2020.

[32] G. Aceto, A. Botta, W. De Donato, and A. Pescapè, "Cloud monitoring: A survey," *Computer Networks*, vol. 57, no. 9, pp. 2093–2115, 2013.

[33] J. Montes, A. Sánchez, B. Memishi, M. S. Pérez, and G. Antoniu, "Gmone: A complete approach to cloud monitoring," *Future Generation Computer Systems*, vol. 29, no. 8, pp. 2026–2040, 2013.

[34] P. Zhang, S. Shu, and M. Zhou, "An online fault detection model and strategies based on svm-grid in clouds," *IEEE/CAA Journal of Automatica Sinica*, vol. 5, no. 2, pp. 445–456, 2018.

[35] B. Ligetfalvi, M. Emődi, J. Kovács, and R. Lovas, "Fundamentals of a novel debugging mechanism for orchestrated cloud infrastructures with macrosteps and active control," *Electronics*, vol. 10, no. 24, 2021.

[36] P. Kacsuk, "Systematic macrostep debugging of message passing parallel programs," *Future Generation Computer Systems*, vol. 16, no. 6, pp. 609–624, 2000.

[37] Y. Tamura, Y. Nobukawa, and S. Yamada, "A method of reliability assessment based on neural network and fault data clustering for cloud with big data," in *2015 2nd International Conference on Information Science and Security (ICISS)*. IEEE, 2015, pp. 1–4.

[38] T. Wang, W. Zhang, J. Wei, and H. Zhong, "Fault detection for cloud computing systems with correlation analysis," in *2015 IFIP/IEEE International Symposium on Integrated Network Management (IM)*. IEEE, 2015, pp. 652–658.

[39] Q. Liu, F. Zhang, M. Liu, and W. Shen, "A fault prediction method based on modified genetic algorithm using bp neural network algorithm," in *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE, 2016, pp. 004 614–004 619.

[40] P. Zhang, S. Shu, and M. Zhou, "Adaptive and dynamic adjustment of fault detection cycles in cloud computing," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 1, pp. 20–30, 2019.

[41] X. Zhang and Y. Zhuang, "A fault detection algorithm for cloud computing using qpso-based weighted one-class support vector machine," in *International Conference on Algorithms and Architectures for Parallel Processing*. Springer, 2019, pp. 286–304.

[42] Z. Dankovičová, D. Sovák, P. Drotár, and L. Vokorokos, "Machine learning approach to dysphonia detection," *Applied Sciences*, vol. 8, no. 10, p. 1927, 2018.

[43] A. N. Sisyukov, O. S. Yulmetova, and V. A. Kuznecov, "Gpu accelerated industrial data analysis in private cloud environment," in *2019 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus)*. IEEE, 2019, pp. 348–352.

[44] A. Abusitta, M. Bellaiche, M. Dagenais, and T. Halabi, "A deep learning approach for proactive multi-cloud cooperative intrusion detection system," *Future Generation Computer Systems*, vol. 98, pp. 308–318, 2019.

[45] Y. Ozaki, S. Kanamoto, H. Yamamoto, and K. Kourai, "Detecting system failures with gpus and llvm," in *Proceedings of the 10th ACM SIGOPS Asia-Pacific Workshop on Systems*, 2019, pp. 47–53.

[46] S. K. Pandey, R. B. Mishra, and A. K. Tripathi, "Machine learning based methods for software fault prediction: A survey," *Expert Systems with Applications*, p. 114595, 2021.

[47] Google Cloud, "Gpus on compute engine — compute engine documentation — google cloud," https://cloud.google.com/compute/docs/gpus, [Online] Accessed 10 September 2021.

[48] Tencent Cloud, "Gpu cloud computing — tencent cloud," https://intl.cloud.tencent.com/product/gpu, 2013-2019, [Online] Accessed 1 October 2021.

[49] Lambda GPU Cloud, "Lambda gpu cloud for deep learning," https://lambdalabs.com/service/gpu-cloud, [Online] Accessed 15 September 2021.

[50] S. Pouyanfar, S. Sadiq, Y. Yan, H. Tian, Y. Tao, M. P. Reyes, M.-L. Shyu, S.-C. Chen, and S. Iyengar, "A survey on deep learning: Algorithms, techniques, and applications," *ACM Computing Surveys (CSUR)*, vol. 51, no. 5, pp. 1–36, 2018.

[51] I. Huzmiev and Z. A. Chipirov, "Energy consumption powered by graphics processing units (gpu) in response to the number of operating computing unit," in *2016 2nd International Conference on Industrial Engineering, Applications and Manufacturing (ICIEAM)*. IEEE, 2016, pp. 1–4.

[52] J. Bergstra, F. Bastien, O. Breuleux, P. Lamblin, R. Pascanu, O. Delalleau, G. Desjardins, D. Warde-Farley, I. Goodfellow, A. Bergeron *et al.*, "Theano: Deep learning on gpus with python," in *NIPS 2011, BigLearning Workshop, Granada, Spain*, vol. 3. Citeseer, 2011, pp. 1–48.

[53] C. Meng, M. Sun, J. Yang, M. Qiu, and Y. Gu, "Training deeper models by gpu memory optimization on tensorflow," in *Proc. of ML Systems Workshop in NIPS*, vol. 7, 2017.

[54] TensorFlow, "Use a gpu in tensorflow," https://colab.research.google.com/github/tensorflow/docs/blob/master/site/en/guide/gpu.ipynb#scrollTo=IqR2PQG4ZaZ0, [Online] Accessed 11 October 2021.

[55] Keras, "Deep learning for humans," https://keras.io, [Online] Accessed 11 October 2021.

[56] S. Li, Y. Zhao, R. Varma, O. Salpekar, P. Noordhuis, T. Li, A. Paszke, J. Smith, B. Vaughan, P. Damania *et al.*, "Pytorch distributed: Experiences on accelerating data parallel training," *arXiv preprint arXiv:2006.15704*, 2020.

[57] N. Lopes and B. Ribeiro, "Gpumlib: An efficient open-source gpu machine learning library," *International Journal of Computer Information Systems and Industrial Management Applications*, vol. 3, pp. 355–362, 2011.

[58] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural networks*, vol. 61, pp. 85–117, 2015.

[59] E. Nagy, R. Lovas, I. Pintye, A. Hajnal, and P. Kacsuk, "Cloud-agnostic architectures for machine learning based on apache spark," *Advances in Engineering Software*, vol. 159, p. 103029, 2021.