Journal of Soft Computing Paradigm (JSCP) (2021)
Vol.03/ No.03
Pages: 135-148
http://irojournals.com/jscp/
DOI: https://doi.org/10.36548/jscp.2021.3.001

A Comprehensive Review on Power Efficient Fault Tolerance Models in High Performance Computation Systems

Nayana Shetty
Associate Professor, Department of Electrical and Electronics Engineering, NMAMIT, Nitte, Udupi, Karnataka, India
E-mail: nayanar401@gmail.com

Abstract

For the purpose of high performance computation, several machines are developed at an exascale level. These machines can perform at least one exaflop calculations per second, which corresponds to a billion billon or 108. The universe and nature can be understood in a better manner while addressing certain challenging computational issues by using these machines. However, certain obstacles are faced by these machines. As huge quantity of components is encompassed in the exascale machines, frequent failure may be experienced and also the resilience may be challenging. High progress rate must be maintained for the applications by incorporating certain form of fault tolerance in the system. Power management has to be performed by incorporating the system in a parallel manner. All layers inclusive of fault tolerance layer must adhere to the power limitation in the system. Huge energy bills may be expected on installation of exascale machines due to the high power consumption. For various fault tolerance models, the energy profile must be analyzed. Parallel recovery, message-logging, and restart or checkpoint fault tolerance models for rollback recovery are evaluated in this paper. For execution with failure, the most energy efficient solution is provided by parallel recovery when programs with various programming models are used. The execution is performed faster with parallel recovery when compared to the other techniques. An analytical model is used for exploring these models and their behavior at extreme scales.

ISSN: 2582-2640 (online)
Submitted: 11.06.2021
Revised: 10.07.2021
Accepted: 29.07.2021
Published: 07.08.2021

Journal of Soft Computing Paradigm (JSCP) (2021)
Vol.03/ No.03
Pages: 135-148
http://irojournals.com/jscp/
DOI: https://doi.org/10.36548/jscp.2021.3.001

Keywords: Exascale Machines, Parallel Recovery, Message Logging, Checkpoint, Fault Tolerance Model

1. Introduction

With the increase in complexity and size of high performance computing systems, the productivity of machines must be retained similar to the previous generations. However, several challenges emerge during this process [1]. Energy consumption and resilience are the major challenges that must be addressed under the prospect of exascale. The exascale machine is formed by a huge number of components leading to the the fundamental concern being resilience. Disks, routers and memory modules with millions of processors are incorporated in such a supercomputer [2]. Failure may occur every few minutes on such exascale machines with so many components involved. In designing exascale applications, systems and architectures, power management is the key element. In order to meet the power budget, all the layers of the system must be constrained which is a crucial factor in power-limited environment [3]. The relatively inexpensive energy contracts also may save up to one million USD every year by reducing one megawatt of power consumption.

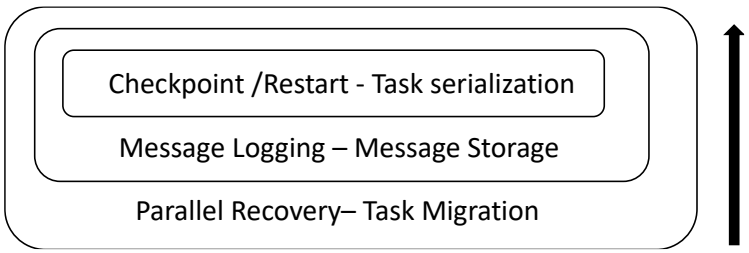

Figure 1. Fault Tolerance Protocol Framework

It is unavoidable to incorporate a fault tolerance model in these systems [4-5]. Ensuring the energy efficiency of the fault tolerance model used is also essential in the system. The exascale resilience research is driven by the energy consumption cost of these strategies. Based on the rate of energy consumption, the checkpoint-based fault tolerance schemes are compared

ISSN: 2582-2640 (online)
Submitted: 11.06.2021
Revised: 10.07.2021
Accepted: 29.07.2021
Published: 07.08.2021

Journal of Soft Computing Paradigm (JSCP) (2021)
Vol.03/ No.03
Pages: 135-148
http://irojournals.com/jscp/
DOI: https://doi.org/10.36548/jscp.2021.3.001

in this paper. After failure, the tasks are allowed to migrate by the system in the parallel recovery approach [6]. The recovery time is reduced by this model to a minor fraction of re-execution time from checkpoint. In case of failure, global rollback is avoided while message storage is ensured using message-logging strategy [7]. Several libraries make use of the local storage based restart or checkpoint scheme. The three fault tolerance schemes are compared in terms of their energy efficiency [8]. The composition of these schemes are represented in figure 1. Multiple programming models are used for evaluation of the results. The system is obtains the understanding of the fault strategies, their response and performance during recovery from failure. Further, at exascale and other simulated scenarios, the energy efficiency of the models are predicted.

2. Literature Review

This paper presents an outline of three fault tolerance models and their evaluation. A large chunk of data is held while preforming computation by each task using parallel programming, where a collection of tasks are involved [9]. Message passing is a crucial technique for information sharing. Machines that involve several nodes can make use of the parallel programing scheme. Here, the nodes are assigned with task by the runtime system decisions enabling one or more tasks to be executed on a single node [10]. When the nodes turns non-functional and cannot return to the functional mode, as per the fail-stop model, the system nodes fail. The failed node may be replaced by other nodes. A checkpoint is used for recreating the lost tasks that were running on the failed node [11]. The failures of a node are tolerated using the fault tolerance models.

Periodic checkpointing of the state by the system helps in tolerating the failure when traditional approaches are used [12]. The entire system rolls back and restarts from the most recent checkpoint if any node crashes. In high performance computation, the checkpoint or restart scheme is commonly used to achieve fault tolerance. The implementation of this technology is straightforward as it operates on a simple underlying principle. Based on the

ISSN: 2582-2640 (online)
Submitted: 11.06.2021
Revised: 10.07.2021
Accepted: 29.07.2021
Published: 07.08.2021

Journal of Soft Computing Paradigm (JSCP) (2021)
Vol.03/ No.03
Pages: 135-148
http://irojournals.com/jscp/
DOI: https://doi.org/10.36548/jscp.2021.3.001

protocol design decision, several variants are adopted [13]. The parallel program tasks or the whole system must be selected as a checkpoint. Determination of this checkpoint is a significant decision [14]. A disk is used for storing the entire machine state in system level checkpoint. BLCR is a popular library used for executing this function. In case of failure, the program execution may be resumed by the task state using the application-level checkpoint. This approach is used by the SCR library [15]. The checkpointed memory is reduced dramatically using the application level checkpoint, which is a major advantage of the system. The various points of execution and the variables to be stored in the program are to be identified by the program. For the HPC applications programmer, this will not be a prominent burden. In this paper, the application-level checkpoint is considered [16, 17].

In case of a failure, with restart/checkpoint, roll back of all nodes is essential. This is a significant drawback in this model [18, 19]. When devices with voluminous nodes are used, the failure of one node will cause the roll back of all the available nodes. The efficiency would increase by several folds if roll back of only the crashing node occurs [20, 21]. Until completion of the crashed node recovery, execution of the remaining nodes may continue. Meanwhile, the failed node catches up with the other nodes [22]. The energy consumption of an idle node is extremely low and hence, from an energy point of view, the appeal of the last alternative is high. During operation, re-execution of only the failed node is sufficient. Rollback ability is provided to the failed node using the checkpoint or restart extension termed as message logging [23]. Messages are stored at the memory of the sender in this process. In case of failure, these messages are resent. Along with the rest of the system, a consistent state is reached by the recovering node. This is ensured by means of message logging protocol along with logging the messages [24]. Storage of the non-deterministic decisions enables this process. The non-deterministic nature of message reception contributes to this process. An appropriate recovery is guaranteed by storage of the determinants generated by every non-deterministic event [25]. Based on the process of handling determinants, different message logging flavors exist. The speed of recovery process is improved when the failed node rolls back with message-logging. Parallel recovery of failed nodes is made possible by migrating the recovering node with some

ISSN: 2582-2640 (online)
Submitted: 11.06.2021
Revised: 10.07.2021
Accepted: 29.07.2021
Published: 07.08.2021

Journal of Soft Computing Paradigm (JSCP) (2021)
Vol.03/ No.03
Pages: 135-148
http://irojournals.com/jscp/
DOI: https://doi.org/10.36548/jscp.2021.3.001

live tasks if migratable tasks are allowed by the system [26]. Parallel recovery scheme is an extension of the message logging process. When compared to the normal rework time, the recovery time is reduced to a minor portion using this model. In comparison to the checkpoint period, the mean time to interrupt is significantly low and high failure rates may be tolerated using this speedup.

3. Research Design

The runtime system based on Charm++ [27] is used for implementing the three fault tolerance schemes. Among migratable objects, an asynchronous type invocation is performed using the parallel programming language Charm++. An object collection is involved in the Charm++ parallel program. The total nodes used for the execution of the program is independent of the total objects. Virtualization ratio is the total objects being executed per logical node. During load balancing, the objects are migrated and nodes are assigned with objects by the runtime system. Message passing is the only possible mechanism that enables communication in Charm++ between two objects. During message transmission, there is a lack of synchronization among objects as active messages alone form the basis of Charm++. Adaptive Message Passing Interface (AMPI) [28, 29] is the extension of Charm++ that enables execution of MPI programs on the runtime system of Charm++. Similar to a Charm++ object, handling of every rank is performed in AMPI enabling its migration from one node to another. A 256 core 64 node system is used for testing the protocols used for fault tolerance. The Ryzen 3 1300X chip with 3.5GHz and 4 cores is used where a single socket is available for each core. A gigabit ethernet switch with 48 ports is used for connecting the cluster nodes. On a per-node basis, at an interval of 1 second each, the machine power is measured by installing a power distribution unit over the rack that hosts the cluster. For every experiment, the readings are gathered and integrated over the time of execution to estimate the energy consumption of the overall machine. Simulation as well as hardware is used for performing the experimentation and the results are compiled in the following section.

ISSN: 2582-2640 (online)
Submitted: 11.06.2021
Revised: 10.07.2021
Accepted: 29.07.2021
Published: 07.08.2021

Journal of Soft Computing Paradigm (JSCP) (2021)
Vol.03/ No.03
Pages: 135-148
http://irojournals.com/jscp/
DOI: https://doi.org/10.36548/jscp.2021.3.001

Multiple applications from diverse programming schemes are chosen for evaluation of the presented fault tolerance schemes. The Jacobi 3D program based on Charm++ is the primary code on a 3D space performing successive over relaxation using a 7-point stencil. The communication pattern is non-trivial in this computation bound program. The MPI programming framework includes NAS parallel benchmark applications, BT, LU, SP, CG and FT. Various techniques are used for solving nonlinear partial differential equation systems using these programs. Some examples include scalar penta-diagonal algorithm used by SP and block tri-diagonal technique used by BT. AMPI is used for execution of these benchmarks. Along with the faulty case, the overhead is also estimated for various approaches. In the testbed, the process corresponding to a physical core logical node is destroyed for injecting failure. For every logical node, one process is created by the runtime system using Charm++. The process ID kill command is used for simulation of the node failure.

4. Results

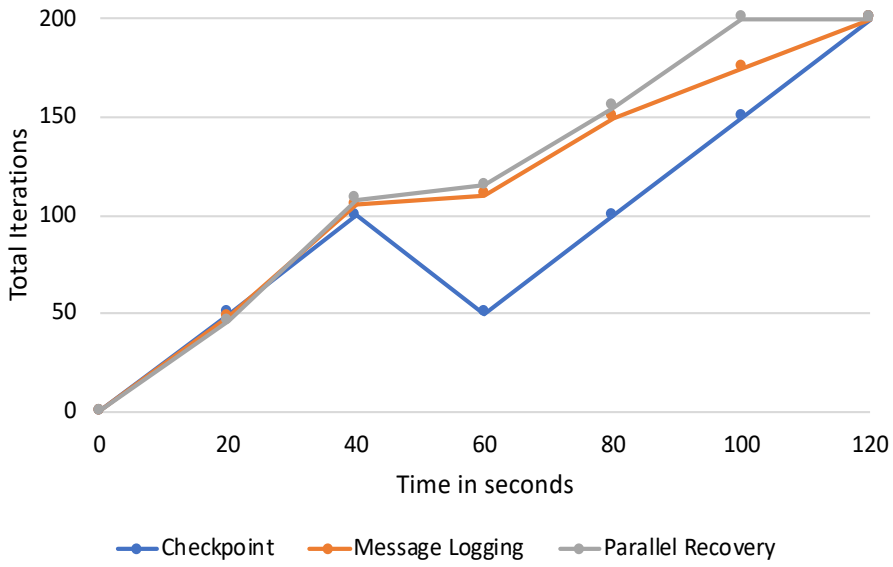

Figure 2. Various Fault Tolerance Schemes and their Corresponding Progress Rate

ISSN: 2582-2640 (online)
Submitted: 11.06.2021
Revised: 10.07.2021
Accepted: 29.07.2021
Published: 07.08.2021

Journal of Soft Computing Paradigm (JSCP) (2021)
Vol.03/ No.03
Pages: 135-148
http://irojournals.com/jscp/
DOI: https://doi.org/10.36548/jscp.2021.3.001

A 10243 space is used for executing the Jacobi3D for the presentation of results. The entire space is categorized into 4096 blocks contained in 643 blocks overall. The Charm++ objects are represented by each block. 64 is the virtualization ratio for the entire cluster with 256 logical nodes. Iterations ranging 50 to 200 are evaluated and plotted graphically. Figure 2 provides the various fault tolerance schemes and their corresponding progress rate. The storage of checkpoint at the local disk with checkpoint or restart while executing Jacobi3D and the power drawn at each node is also evaluated. Across the entire program execution duration, the variation in power levels is estimated in this experiment. An average of 45 W power is drawn by the system in idle mode. The power drawn rises to 105 W during the start of execution. As the checkpoint is started by the application, there is a power drop to 50 W. During checkpoint, the process in which power drop occurs to the base power is highlighted in the experiment.

When compared to the base power, during checkpoint, the average power is higher by 9% and corresponds to 50 W. When compared to computation, during checkpointing, the energy consumption is lesser. Smaller optimum checkpoint period is observed rather than execution time during optimization of energy consumption. Instead of re-computation of the missing data, frequent checkpointing enables the system to operate more efficiently with less energy consumption. When memory checkpointing is performed, identical results are observed. In comparison with power meter frequency, the precision is faster in this experimentation while for a specific problem size, memory checkpointing is performed. However, the time consumption is less than one second on an average. The rest of the experiments are also conducted using memory checkpoint. All the fault tolerance schemes are compared using the same approach. When compared to checkpoint or restart the overhead of message logging is very low. For this purpose, same failure time and checkpoint frequency is retained. Failure is injected at around 30 seconds in between the second interval between two checkpoints. A total of 100 iterations are performed. Figure 3 represents the comparison between the various fault tolerance schemes and their corresponding power consumption.

ISSN: 2582-2640 (online)
Submitted: 11.06.2021
Revised: 10.07.2021
Accepted: 29.07.2021
Published: 07.08.2021

Journal of Soft Computing Paradigm (JSCP) (2021)
Vol.03/ No.03
Pages: 135-148
http://irojournals.com/jscp/
DOI: https://doi.org/10.36548/jscp.2021.3.001

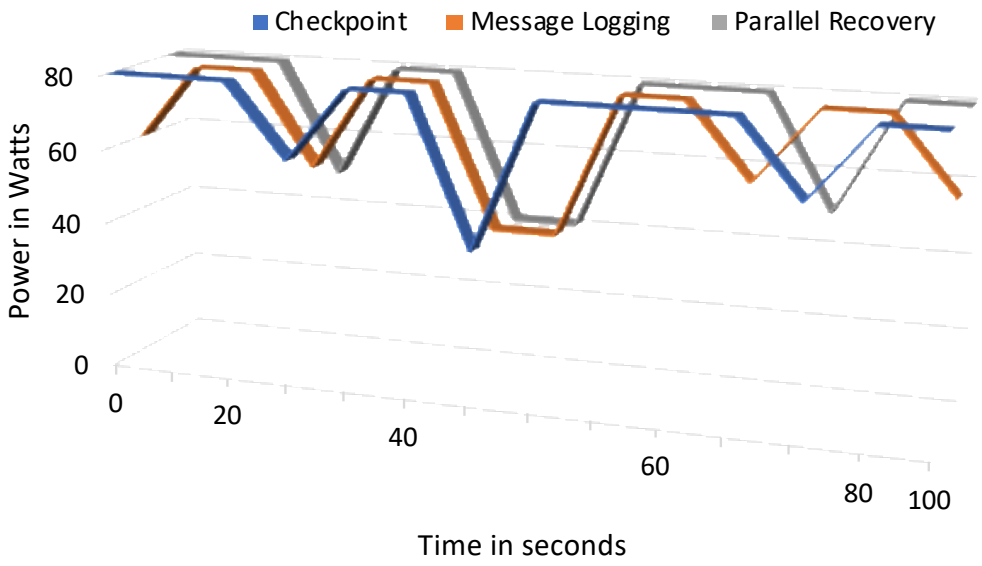

Figure 3. Various Fault Tolerance Schemes and their Corresponding Power Consumption

With the analytical and experimental results obtained, certain significant factors must be discussed. Primarily, in a system with resilient runtime, the concerns with respect to power management must be addressed. Among the applications analyzed in this paper, the maximum power is not increased by the fault tolerance models when compared to the applications without the fault tolerance model as seen in the experimental results. The power drawn is not increased even during storage of messages in handles and memory determinants during message logging. During the specified execution time, due to the overhead, energy consumption is also impacted in this model. The overhead and energy consumption are directly proportional in parallel recovery, message logging and fault free cases. While modifying the base power and maximum ratio, the process impacts the model. Parallel recovery and message logging can save huge amount of energy when this ratio is smaller. The value of the base power is attained by the non-faulty nodes during recovery as seen in the experimental results. During recovery, substantially small base power is attained when compared to the maximum power.

ISSN: 2582-2640 (online)
Submitted: 11.06.2021
Revised: 10.07.2021
Accepted: 29.07.2021
Published: 07.08.2021

Journal of Soft Computing Paradigm (JSCP) (2021)
Vol.03/ No.03
Pages: 135-148
http://irojournals.com/jscp/
DOI: https://doi.org/10.36548/jscp.2021.3.001

Over a duration of 3 years, various architectures and their power comparison is estimated. Wave2D, a Charm++ application with a 5 point stencil is used for obtaining results of each architecture on a single node. As time increases, the maximum power or the base ratio decreases. Lowest ratio of 2.1 is observed in Sandy Bridge, a recent machine and highest ratio of 0.48 is observed in Intel Xenon, one of the oldest machine. When parallel recovery and message logging is used, the energy consumption is saved largely as observed in this trend point. The energy consumption may be further reduced by dividing the computation into minor units through decomposition. The runtime system is empowered by this degree of freedom. Along with the applications load balance, recovery parallelism available is also a fundamental factor that contributes to over decomposition. The factor of parallelism is represented by the term P. The variation in the model with change in parameter P is as represented in the graph below. Parallel recovery is performed in a scalable manner with improved energy saving with the increase in the value of P. Better system utilization is achieved and system recovery is accelerated by encouraging over decomposition by programming models.

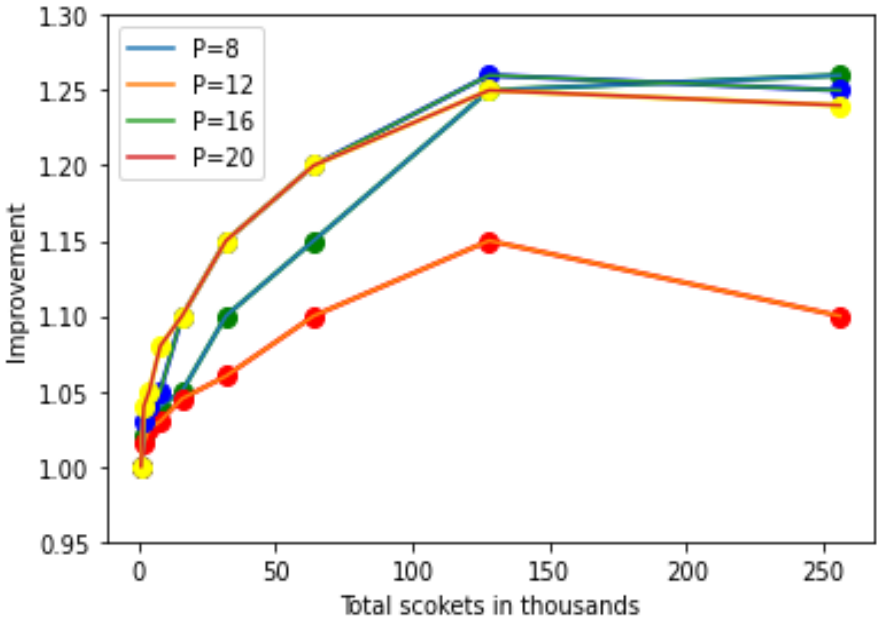

Figure 4. Variation in the model with the change in parameter P

ISSN: 2582-2640 (online)
Submitted: 11.06.2021
Revised: 10.07.2021
Accepted: 29.07.2021
Published: 07.08.2021

Journal of Soft Computing Paradigm (JSCP) (2021)
Vol.03/ No.03
Pages: 135-148
http://irojournals.com/jscp/
DOI: https://doi.org/10.36548/jscp.2021.3.001

5. Conclusion

The parallel recovery, message logging and checkpoint or restart based fault tolerance models are successfully compared based on their energy efficiency in this paper. Multiple programming models are used as benchmarks for evaluation of these protocols. At large scales, the behavior of the protocols are predicted using an analytical model that is built on the guidance of experimental results. When the fault tolerance model based on message logging is used, the increase in power drawn is not proved with any empirical evidences so far. The application progress rate is imposed with an overhead due to the failure-free scenario increasing the energy consumption of the system. In faulty execution, the energy consumption as well as execution time factors are more efficient with parallel recovery according the experimental results. Using task migration, the node recovery process is accelerated. The minimum energy consumption with respect to the system administrators and minimum execution time with respect to the users is satisfied with parallel recovery. The applications energy consumption and execution time is reduced at extreme scales using parallel recovery predicting analytical models. The energy consumption and execution time is saved by a factor of 15%. The operation of full-fledged applications with regard to the fault tolerance protocols has to be understood in a detailed manner. Particle-interaction simulation applications particularly benefit by this analysis. Parallel recovery is attained and over decomposition may be attained at a high degree using these programs.

References

[1] Beechu, N. K. R., Harishchandra, V. M., & Balachandra, N. K. Y. (2017). High-performance and energy-efficient fault-tolerance core mapping in NoC. Sustainable Computing: Informatics and Systems, 16, 1-10.

[2] Karuppusamy, Dr P. "Performance Analysis of Multiple Pico Hydro Power Generation." Journal of Electrical Engineering and Automation 2, no. 2: 92-101.

ISSN: 2582-2640 (online)
Submitted: 11.06.2021
Revised: 10.07.2021
Accepted: 29.07.2021
Published: 07.08.2021

Journal of Soft Computing Paradigm (JSCP) (2021)
Vol.03/ No.03
Pages: 135-148
http://irojournals.com/jscp/
DOI: https://doi.org/10.36548/jscp.2021.3.001

[3] Bautista-Gomez, L., Tsuboi, S., Komatitsch, D., Cappello, F., Maruyama, N., & Matsuoka, S. (2011, November). FTI: High performance fault tolerance interface for hybrid systems. In Proceedings of 2011 international conference for high performance computing, networking, storage and analysis (pp. 1-32).

[4] Vijayakumar, T., and Mr R. Vinothkanna. "Efficient Energy Load Distribution Model using Modified Particle Swarm Optimization Algorithm." Journal of Artificial Intelligence 2, no. 04 (2020): 226-231.

[5] Ansari, M., Salehi, M., Safari, S., Ejlali, A., & Shafique, M. (2020). Peak-Power-Aware Primary-Backup Technique for Efficient Fault-Tolerance in Multicore Embedded Systems. IEEE Access, 8, 142843-142857.

[6] Kamel, Khaled, and Eman Kamel. "Process Control Ladder Logic Trouble Shooting Techniques Fundamentals." IRO Journal on Sustainable Wireless Systems 1, no. 4 (2019): 206-241.-1

[7] Jahanpour, H., Barati, H., & Mehranzadeh, A. (2020). An Energy Efficient Fault Tolerance Technique Based on Load Balancing Algorithm for High-Performance Computing in Cloud Computing. Journal of Electrical and Computer Engineering Innovations (JECEI), 8(2), 169-182.

[8] Wang, Haoxiang. "Flexibility Management in Renewable Energy Source Operated Power Systems using Decision Support System." Journal of Electrical Engineering and Automation 2, no. 1 (2020): 35-42.

[9] Meneses, E., Sarood, O., & Kalé, L. V. (2012, October). Assessing energy efficiency of fault tolerance protocols for HPC systems. In 2012 IEEE 24th International Symposium on Computer Architecture and High Performance Computing (pp. 35-42). IEEE.

[10] Sathesh, A. "Assessment of Environmental and Energy Performance Criteria for Street Lighting Tenders using Decision Support System." Journal of Electronics and Informatics 2, no. 2: 72-79.

[11] Goundar, S., & Bhardwaj, A. (2018). Efficient fault tolerance on cloud environments. International Journal of Cloud Applications and Computing (IJCAC), 8(3), 20-31.

ISSN: 2582-2640 (online)
Submitted: 11.06.2021
Revised: 10.07.2021
Accepted: 29.07.2021
Published: 07.08.2021

Journal of Soft Computing Paradigm (JSCP) (2021)
Vol.03/ No.03
Pages: 135-148
http://irojournals.com/jscp/
DOI: https://doi.org/10.36548/jscp.2021.3.001

[12] Karuppusamy, Dr P. "Synchronization of Reactive Power in Solar Based DG and Voltage Regulated Elements Using Stochastic Optimization Technique." Journal of Electrical Engineering and Automation 2, no. 1 (2020): 50-59.

[13] Yu, S., Tang, Z., Ye, X., Zhang, Z., Fan, D., & Jiang, Z. (2018, December). High-Performance and Energy-Efficient Fault Tolerance Scheduling Algorithm Based on Improved TMR for Heterogeneous System. In 2018 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Ubiquitous Computing & Communications, Big Data & Cloud Computing, Social Computing & Networking, Sustainable Computing & Communications (ISPA/IUCC/BDCloud/SocialCom/SustainCom) (pp. 207-214). IEEE.

[14] Losada, N., González, P., Martín, M. J., Bosilca, G., Bouteiller, A., & Teranishi, K. (2020). Fault tolerance of MPI applications in exascale systems: The ULFM solution. Future Generation Computer Systems, 106, 467-481.

[15] Bansal, Malti, Harmandeep Singh, and Gaurav Sharma. "A Taxonomical Review of Multiplexer Designs for Electronic Circuits & Devices." Journal of Electronics 3, no. 02 (2021): 77-88.

[16] Wang, K., Louri, A., Karanth, A., & Bunescu, R. (2019, March). High-performance, energy-efficient, fault-tolerant network-on-chip design using reinforcement learning. In 2019 Design, Automation & Test in Europe Conference & Exhibition (DATE) (pp. 1166-1171). IEEE.

[17] Bashar, Abul, and S. Smys. "Integrated Renewable Energy System for Stand-Alone Operations with Optimal Load Dispatch Strategy." Journal of Electronics 3, no. 02 (2021): 89-98.

[18] Hengjinda, P., Joy Iong Zong Chen, and Joy Iong Zong. "Renewable Energy Production from Agricultural Waste and Hydrogen Battery Formation." Journal of Electrical Engineering and Automation 2, no. 4: 151-155.

[19] Ranganathan, Dr G. "Energy Storage Capacity Expansion of Microgrids for a Long-Term." Journal of Electrical Engineering and Automation 3, no. 1 (2021): 55-64.

ISSN: 2582-2640 (online)
Submitted: 11.06.2021
Revised: 10.07.2021
Accepted: 29.07.2021
Published: 07.08.2021

Journal of Soft Computing Paradigm (JSCP) (2021)
Vol.03/ No.03
Pages: 135-148
http://irojournals.com/jscp/
DOI: https://doi.org/10.36548/jscp.2021.3.001

[20] Rai, Ashok Kumar, and A. K. Daniel. "An Energy-Efficient Routing Protocol Using Threshold Hierarchy for Heterogeneous Wireless Sensor Network." In Intelligent Data Communication Technologies and Internet of Things: Proceedings of ICICI 2020, pp. 553-570. Springer Singapore, 2021.

[21] Sampaio, A. M., & Barbosa, J. G. (2018). A comparative cost analysis of fault-tolerance mechanisms for availability on the cloud. Sustainable Computing: Informatics and Systems, 19, 315-323.

[22] Velu, Karthika, Pramila Arulanthu, and Eswaran Perumal. "Energy Reduction Stratagem in Smart Homes Using Association Rule Mining." In International Conference on Innovative Data Communication Technologies and Application, pp. 188-193. Springer, Cham, 2019.

[23] Chen, C. A., Won, M., Stoleru, R., & Xie, G. G. (2014). Energy-efficient fault-tolerant data storage and processing in mobile cloud. IEEE Transactions on cloud computing, 3(1), 28-41.

[24] Balasubramanian, M., V. Rajamani, and S. Puspha. "Enhancing Spectrum Efficiency and Energy Harvesting Selection for Cognitive Using a Hybrid Technique." In International Conference on Inventive Computation Technologies, pp. 556-568. Springer, Cham, 2019.

[25] Li, S., Li, H., Liang, X., Chen, J., Giem, E., Ouyang, K., ... & Chen, Z. (2019, November). FT-iSort: efficient fault tolerance for introsort. In Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis (pp. 1-17).

[26] Shafique, M., Rehman, S., Kriebel, F., Khan, M. U. K., Zatt, B., Subramaniyan, A., ... & Henkel, J. (2016). Application-guided power-efficient fault tolerance for H. 264 context adaptive variable length coding. IEEE Transactions on Computers, 66(4), 560-574.

[27] Karthikeyan, M. M., and G. Dalin. "Dynamic Congestion Control Routing Algorithm for Energy Harvesting in MANET." In Inventive Computation and Information Technologies, pp. 15-25. Springer, Singapore, 2021.

ISSN: 2582-2640 (online)
Submitted: 11.06.2021
Revised: 10.07.2021
Accepted: 29.07.2021
Published: 07.08.2021

Journal of Soft Computing Paradigm (JSCP) (2021)
Vol.03/ No.03
Pages: 135-148
http://irojournals.com/jscp/
DOI: https://doi.org/10.36548/jscp.2021.3.001

[28] van Dam, H. J., Vishnu, A., & De Jong, W. A. (2011). Designing a scalable fault tolerance model for high performance computational chemistry: A case study with coupled cluster perturbative triples. Journal of chemical theory and computation, 7(1), 66-75.

[29] Sivapriyan, R., D. Elangovan, and Kavyashri SN Lekhana. "Review of Python for Solar Photovoltaic Systems." In Evolutionary Computing and Mobile Sustainable Networks, pp. 103-112. Springer, Singapore, 2021.

Author's biography

Nayana Shetty, is working as an associate professor in the Department of Electrical and Electronics Engineering, NMAMIT, Nitte, Udupi, Karnataka.The author's area of research are power electronics, instrumentation, electrical machines and drives, smart grids, power sources, renewable energy, intelligent systems, automation, control theory, circuits and systems, power systems, EHV, electric machines, future energy systems, autonomous and distributed energy systems, smart energy storage and management.

ISSN: 2582-2640 (online)
Submitted: 11.06.2021
Revised: 10.07.2021
Accepted: 29.07.2021
Published: 07.08.2021